Precursors, Proxies, and Predictive Models for Long-Horizon Tasks

Samuel F. Brown
Independent *

Jaco Du Toit
AI Safety South Africa

Leo Hyams
AI Safety South Africa

Daniil Anisimov
SPAR †

Abstract

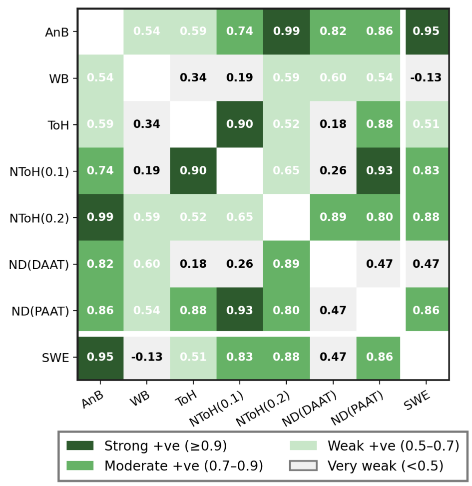
AI agents show remarkable success at various short tasks, and are rapidly improving at longer-horizon tasks, creating a need to evaluate AI capabilities on dangerous tasks which require high autonomy. Evaluations (evals) comprising long-running "real-world" tasks may be the best proxies for predicting general performance, but they are expensive to create, run, and compare to human baselines. Furthermore, these tasks often rely on a large, interwoven set of agent skills, which makes predicting capabilities development difficult. We hypothesize that precursor capabilities including "persistence", "dexterity", and "adaptability" are upstream of robust autonomous performance on long-horizon tasks, and design simple procedurally-generated "proxy" evals to target these precursors. We then use agent performance on our proxy evals to calibrate a preliminary method of capability prediction on a more complex task: SWE-Bench. Our preliminary results show that performance on certain proxy evals can be unusually predictive of performance on other evals. We find that a simple adaptability proxy based on developmental psychology correlates with SWE-bench with $r = 0.95$, and three other proxies correlate with SWE-bench at $r > 0.8$. A proxy eval which only takes $\sim$10 steps is strongly correlated with the performance of many other evals, which otherwise take much longer to terminate ($\sim$100s of steps). For our predictive model, our initial results correctly predict agent scores on SWE-bench, but have large error bars, suggesting that – testing more models on more synthetic evals – we can quickly and cheaply predict performance on important long-horizon tasks.

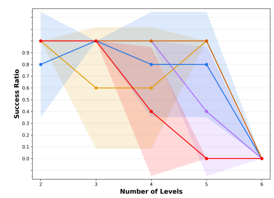
1 Introduction

What holds back AI agents today is not so much their ability to succeed at short-term tasks, but their ability to robustly sustain their performance. AI agents have begun to succeed at autonomous cybersecurity [32] and self-replication [4] tasks in recent evaluations (evals), posing critical safety risks. Alongside this, the length of software engineering tasks AI agents can complete has been exponentially increasing over the past 6 years, with a doubling time of around 7 months [15], and has recently surpassed 2 hours [20]. We show that success at tasks which require robust autonomy of Language Model (LM) agents, such as SWE-bench, correlates to "precursor" capabilities: an agent's *persistence* at completing simple but long tasks, *dexterity* at handling many hierarchical relationships, and *adaptability* to change. Developing an understanding of precursors could provide insight into current bottlenecks, steering elicitation efforts and identifying capabilities overhang. Decomposing agent capabilities into precursors enables researchers to develop predictive models of agent behavior, helping inform policy. We demonstrate that by measuring performance on proxy evals intended to measure precursor skills, we can predict performance on more complex "real-world" evals.

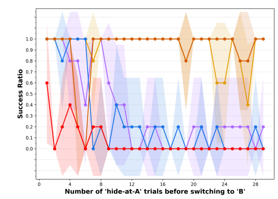
*Correspondence to pdca@sambrown.eu , leo@aisafetysa.com
†Supervised Programme for Alignment Research, sparai.org

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.



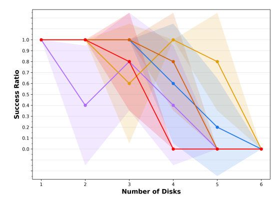


(b) WB

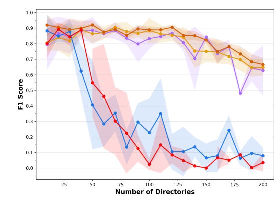


(c) AnB

(a) Correlations between model scores on evals



(d) NToH(0.1)



(e) ND(DAAT)

GPT-4o — Claude Sonnet 3.7 — Claude Sonnet 4 — Gemini 2.5 Flash — o3-mini (low)

Figure 1: (a) Pearson correlation coefficient between model scores across evaluations. Performance at 'A-not-B [AnB]', a simple test, is correlated with many other evals. SWE-bench results [SWE], expensive to obtain directly, are strongly correlated with multiple proxy evals, suggesting the possibility of a predictive ensemble model. (b)-(e) Agent success decays as eval complexity increases (see Appendix D for all results). We run 5 epochs for each eval, shaded areas are 1 s.d. To assign an agent a single score for an eval of variable complexity, we use normalized area under the decay curve.

2 Related work

Capability Evaluations Many evaluations aim to understand the capabilities of AI models, and the risks they pose. Some are general purpose [10] or for general agentic tasks [21], others for particular skills or knowledge areas [17, 9]. Evaluations of coding performance range from single-completion generation of code [5], to agentic resolution of GitHub issues [13], to ML engineering tasks [12], and to many others. Criticism includes suggestions that benchmarks may be unrealistic [14, 2], or distract from higher-priority safety interventions [26].

Evaluations often take significant effort to produce (see e.g. Humanity's Last Exam [24]), and evaluations which are tuned to be sensitive to model performance at the point of publication are often quickly "saturated", with models reaching indistinguishably-high performance [9]. Our paper describes an approach where the generation of the evaluation is automated, scaling as ever more capable models are developed, in line with other work such as exploring reasoning effort [29] and competition-based LLM evals [8].

Precursor Capabilities and Predictive Models Evals that focus on "red-line" risks, such as the ability for LLMs to increase CBRN and Cyber risks [17], are necessary but not sufficient for AI governance, since they leave researchers and policymakers vulnerable to step-changes in capabilities advancements [25]. In line with this, Pistillo and Stix[25] define a set of precursors to AI deception in order to provide a granular set of policy triggers. Similarly, our work involves high-level *cognitive* precursors such as "adaptability". These precursors could give insight into bottlenecks (which could steer elicitation efforts and identify potential capabilities overhang) and better predict the course of capabilities development, thereby informing policy.

Scaling laws representing language model performance as a function of a low-dimensional capability space show that agent performance can be predicted from simpler, non-agentic benchmarks [27]. However, we hypothesize that when analyzing performance only on existing, organic tasks, underlying skills are too interwoven to be well-separated. Instead, we develop a suite of evals to *target* hypothesized precursor skills, and then perform confirmatory and exploratory analyses.

Task Complexity Models have been observed to lack goal-directedness [7, 11, 16], failing to bring their full capabilities to bear on a task when that task is one step of a larger task, rather than a task in isolation. Compound tasks can see significantly deteriorated performance even when comprising fewer than 4 subtasks [30]. Reasoning about goal-directed tasks has been found to vary between models, and to depend on post-training elicitations such as Chain-of-Thought and Tree-of-Thought [3]. Constant hazard rate has been suggested as the simplest model in survival analysis [23], where the likelihood of succeeding on a subtask is determined purely by its human time-to-complete. We are unaware of any work to build *predictive* models of LLM-agent ability to complete complex tasks.

Adaptability Agents based on Reinforcement Learning are often deeply challenged by stochasticity (where actions have probabilistic outcomes) and *non-stationarity* (where an environment has a potentially well-flagged step-change) [6]. LLM agents' propensity to persist in the face of unexpected setbacks has been examined in the context of goal-directedness [7], but to our knowledge we are the first to try to extract a predictive precursor ability.

3 Method

We build dynamic, procedurally generated, multi-step agent evaluations using the Inspect framework [1] and measure the performance of their `basic_agent` (a ReAct agent [31]) on these tasks.

We develop proxy tasks for each of our three precursors:

Targeting persistence We use 'persistence' to refer to the ability of an LM agent to complete *compound* tasks, which are comprised of many potentially-independent subtasks. For this precursor, we use the Path-at-a-time variant of our Nested Directory task [ND-PAAT].

Targeting dexterity We use 'dexterity' to refer to the ability of an LM agent to complete *complex* tasks, where subtasks affect each other. Here we develop three perfect-information tasks, which investigate an agent's **ability to follow-through on tasks which can be perfectly planned** before execution begins. These include Tower of Hanoi [ToH], Website Bios [WB], and the Directory-at-a-time variant of the Nested Directory task [ND-DAAT].

Targeting adaptability For adaptability, we distinguish between tasks which exercise *stochasticity* (where actions have probabilistic outcomes) and those which exercise *non-stationarity* (where an environment has a potentially well-flagged step-change) [6]. This involves one set of stochastic tasks (Noisy Tower of Hanoi [NToH]) where **tools have a fixed probability of malfunctioning**, and another non-stationary task (A-not-B [AnB]) where there is a **clearly-flagged step-change in the environment**.

Summaries of each task are given in Appendix A, with more details in Appendix B.

We then instruct agents using each of 5 LLMs (listed in Fig. 1) to attempt each task, running on task variants of increasing size until the model fails to succeed, and use the normalized area under the decaying performance curve as the agent's overall score for that task.

We then calculate the correlations between model scores on each eval (Pearson coefficient, see discussion in Appendix E) and also the correlation between proxy evals and scores on SWE-bench. [3]

4 Results and Discussion

There are some strikingly strong correlations (e.g. 0.99, 0.95, see Figure 1) between model performance at different evals, suggesting that some of the variance between models can be captured by other, simpler evals.

However, it is hard to tell a convincing story about *patterns* in these preliminary data - our small sample size is a key area for improvement. Making the case that adaptability features strongly: the proxy eval with highest mean correlation to other evals is AnB, with a mean correlation of 0.81, and

[3] SWE-bench scores from swebench.com correspond to the `mini-swe-agent`, rather than Inspect's `basic_agent`, though these frameworks are similar.

the highest correlation is between two proxies designed to test adaptability: AnB and NToH(0.2). ToH and its slightly-noisy variant NToH(0.1) are strongly correlated, as one might expect, but – confusingly – the third-highest correlation is between NToH(0.1) (designed to test stochasticity) and the perfect-information ND(PAAT) eval (designed to test persistence). The two Nested Directory tasks also have a surprisingly low correlation to each other. A confirmatory factor analysis is warranted, and – for future work with additional proxy evals – an exploratory factor analysis.

SWE-bench has strong correlation (> 0.8) with 4 proxy evals, including one at 0.95. This suggests that an ensemble learning technique could be used to predict performance. However, weak learners can be best used to build an ensemble learning model when they are uncorrelated to each other, and all proxies correlated > 0.5 with SWE-bench are similarly correlated with each other.

4.1 Predictive model of SWE-bench performance

To test the predictive power of these correlations, we estimated SWE-bench scores from proxy evaluations (see Appendix F for details). While the true values are within the error bars, and the ordering of models is correct, the error bars are very large (10-60 percentage points), largely due to our initial paucity of data. These preliminary results suggest that even with few models, ensembles of proxy evals can provide informative predictions of downstream task performance. We hope to see improved predictions and reduced error bars as we scale up the number of models and proxy evals in future work.

4.2 Other behaviors of interest

"Model collapse" is not universally seen Figure 1 shows that some models resist "collapse", with performance instead smoothly decreasing (DAAT) or staying constant (PAAT) for hundreds of steps, challenging the model-collapse narrative of Shojaee et al.[29].

Targeting failure modes can reveal extreme fragility Frontier models perform surprisingly poorly at the A-not-B test. Models only need to see an action ("reach for location A") be rewarded $\sim$10 times before becoming insensitive to explicit changes of the environment.

Agents can spontaneously recover from collapse During the runs of NToH, we were unsurprised to see that while some agents were simply inconvenienced by the noise, others were completely derailed. More surprising was to see agents which had flat-lined for $\sim$25 moves suddenly recover and make significant progress, perhaps "wandering in solution space"[19]. More details in Appendix C.

4.3 Limitations

The construct validity of the precursors studied here is uncertain. We do not see the clear clustering of correlations we might have expected. With few evals, it is unclear whether agent performance relates to the eval's structure, as we intend, or to trivial details of the particular proxy eval. Targeting precursors with multiple proxy evals and increasing the number of LLMs evaluated would improve the signal-to-noise ratio. In particular, we expect that the large error bars of our predictive model would shrink with additional data.

We also do not explore ways of maximally eliciting performance on our proxies, though we are aware that performance can vary significantly based on seemingly-trivial prompt details [18, 22, 28].

To make meaningful claims about the relevance of correlations of a given strength, results should be compared to a baseline of correlations between other general benchmarks and evals from the literature.

5 Conclusion

This paper shows that targeting evals based on *a priori* hypothesized "precursor" abilities can result in model scores with high correlation to performance on organic long-horizon tasks. Future work will expand this work to more models and proxies, increasing sample size to improve the signal-to-noise ratio, laying groundwork for quickly and cheaply predicting performance on more substantial tasks, helping focus evaluator resources during pre-deployment testing.

Acknowledgments and Disclosure of Funding

Use unnumbered first level headings for the acknowledgments. All acknowledgments go at the end of the paper before the list of references. Moreover, you are required to declare funding (financial activities supporting the submitted work) and competing interests (related financial activities outside the submitted work). More information about this disclosure can be found at: `https://neurips.cc/Conferences/2024/PaperInformation/FundingDisclosure`.

Do **not** include this section in the anonymized submission, only in the final paper. You can use the `ack` environment provided in the style file to automatically hide this section in the anonymized submission.

A Proxy Eval summaries

A.1 Tasks which target Complexity

We distinguish between *compound* tasks (those that comprise many potentially-independent subtasks) and *complex* tasks (those where subtasks affect each other). We call the ability to success at compound tasks 'persistence', and the ability to succeed at complex tasks 'dexterity'.

Nested Directory (Directory-at-a-time) [ND(DAAT)] A **perfect-information** task for which the order in which subtasks can be completed is **partially constrained.** The agent must create a directory structure specified in the prompt; parent directories must be made before child directories.

Nested Directory (Path-at-a-time) [ND(PAAT)] A **perfect-information** task for which the order in which subtasks can be completed is **unconstrained**. The agent must create a directory structure specified in the prompt; missing parent directories can be automatically created, so subtasks can be completed in any order.

Tower of Hanoi [ToH] A **perfect-information** task for which the order in which subtasks can be completed is **partially constrained.** Disks of increasing size are placed on one of three rods, and must all be moved to another rod while never placing a larger disk on a smaller one. There is only one optimal path.

Website Bios [WB] A **perfect-information** task for which the order in which subtasks can be completed is **partially constrained.** The agent must create a webpage detailing a fictional company's organizational chart, assembled from diverse input. Representing the org-chart's tree-like structure requires complex navigation of subtasks.

A.2 Tasks which target Adaptability

We distinguish between tasks which exercise *stochasticity* (where actions have probabilistic outcomes) and those which exercise *non-stationarity* (where an environment has a potentially well-flagged step-change) [6].

Noisy Tower of Hanoi [NToH(0.1), NToH(0.2)] A task with **stochasticity**: when the agent attempts to use a tool to move a disk, there is a fixed probability (given in brackets) that the tool will malfunction and a different valid move will be made instead.

A-not-B [AnB] A task with **non-stationarity**, inspired by animal/developmental cognition. The agent repeatedly sees an object being 'hidden' in a location, and must each time 'search' that location. Initially, the object is always hidden in location A. After some number of repetitions, the agent watches as the object is instead hidden in location B. An A-not-B error occurs when the agent reaches for the incorrect location A after having seen the object being hidden in location B.

B Methodology Details

We use Inspect's default temperature for each model, except for Nested Directory tasks where we use T=0.

B.1 Perfect Information tasks

We develop 'Perfect Information' tasks, where the model knows all the details of the task from the beginning, and must simply plan and carry out multiple steps.

B.1.1 Nested Directories

Nested directories tests whether an agent can reconstruct a directory tree from only its leaves. We generate an unbalanced target tree by starting at the root and, under a maximum-depth limit, iteratively attach a new child to a randomly chosen existing node until the tree has n nodes, producing uneven branching and path lengths. The agent sees only the set of leaf paths (e.g., `/a/b/c`, `/a/d`) and must recreate the minimal directory structure that makes them valid. Performance is reported as an F1 score which combines precision (how many of the generated paths correspond to target paths) and recall (how many target paths were actually generated). We originally developed the task with a perfect n-ary target tree, but its superlinear scaling makes it difficult to distinguish performance of different models.

Coupling of sub-tasks We investigate the difference between tasks where subtasks can be completed in any order, and tasks where the order of subtask completion is partially constrained. We use two variants of the Nested Directories task described above. In the path-at-a-time (PAAT) variant the agent is permitted to use `mkdir -p` which allows it to create paths without creating parents in advance. This makes this variant closer to a copy/paste needle-in-haystack task, rather than a continuously-state-aware navigation task. In the directory-at-a-time (DAAT) variant, we constrain the agent to create each directory individually and so the order of creation is partially constrained.

B.1.2 Tower of Hanoi

Tower of Hanoi consists of three rods (A, B, C) where rod A is populated with n disks stacked in increasing size, i.e., the largest disk is at the bottom of the rod and the smallest disk is at the top of the stack. The agent must move all disks from rod A to rod C without ever placing a larger disk on top of a smaller one. We measure success by inspecting the final game state and determining whether all disks are stacked on rod C. Progress is measured by comparing the optimal number of moves required to solve the game from the current state to the total number of optimal moves needed to solve the full game. Details of this computation are provided in Appendix B. This progress measure allows us to automatically categorise different types of premature submissions as "improving", "stuck", or "backtracking".

The agent is provided with a custom tool `move_disk()` which it uses to alter the game state. The agent is notified whenever it attempts to make an invalid move. The agent also has access to another tool `show_game()` which displays the current game state.

B.1.3 Website Bios

Website Bios is an evaluation where an agent is tasked to create an HTML webpage for an organisational chart (diagram that maps departments, roles, and reporting lines) of a fictitious dynamically-generated organisation, using a set of website generation tools that we provide to avoid formatting errors. The information we provide the agent includes a JSON file describing the structure of the organisation and a directory of text files containing biographies of employees within the organisation. This evaluation serves to investigate how an agent deals with long-range dependencies and organising information in a hierarchical structure. We constrain the model such that it cannot produce a code solution, but has to rely on its context window, and understanding of dependent relationships.

B.2 Agent adaptability: Non-stationary and Stochastic tasks

We develop tasks to exercise an agent's ability to handle **non-stationarity** (where there is a well-flagged step-change in the environment) and **stochasticity** (where tools have a constant probability of malfunction).

B.2.1 Noisy Tower of Hanoi

We develop Noisy Tower of Hanoi which is a variant of Tower of Hanoi where some percentage of attempted moves are randomly replaced by other valid moves. We control noise with two parameters: the number of injected random moves N_{pert} and a trigger probability $\eta \in (0, 1)$. At each attempted move, with probability η the move is discarded and instead N_{pert} random moves are executed, otherwise the agent move is applied.

B.2.2 A-not-B

The A-not-B test is inspired by animal/developmental cognition, where it tests for an incomplete or absent schema of object permanence. In it, an agent (usually infant or animal) repeatedly sees an object be hidden in location A, and subsequently each time searches location A. After many repetitions, the agent watches as the object is hidden in location B. An A-not-B error occurs when the agent reaches for the incorrect location A on reverse trials (when the agent has seen the object being hidden in location B). In the context of LLM-agents, the A-not-B test sets up conditions which are very conducive to hallucination and repetitive lock-in, while being simple, easy to score and arbitrarily scalable.

C Automated partial-progress and exit-condition tracking

We measure partial progress in Tower of Hanoi as the percentage of optimal moves completed. For n disks, the minimal solution length is

$$T(n) = 2^n - 1.$$

We represent a configuration by a vector pos of length n, where $pos[i]$ is the rod holding disk i (with disk 1 the smallest and disk n the largest). From such a configuration we compute the remaining optimal moves, MovesLeft(pos), via a standard recurrence based on the position of the largest disk (Algorithm 1). Progress is then defined as

$$\text{Progress}(pos) = 100 \cdot \max\left(0,\, 1 - \frac{\text{MovesLeft}(pos)}{T(n)}\right),$$

We automatically classify each run from its sequence of progress values. A run is labeled *Full Success* if it reaches 100% progress, and *Message Limit Reached* if it terminates at the configured message limit. Otherwise, we inspect the last ten progress points to determine the trajectory trend: if progress increases relative to the start of this window and the final value is near the run's maximum, the run is labeled *Early Submission: Improving*; if progress decreases, it is labeled *Early Submission: Regressing*; and if it shows no clear trend, it is labeled *Early Submission: Plateaued*.

Figure 2 shows examples of progress trajectories and automatic categorization over different runs for Tower of Hanoi and Noisy Tower of Hanoi. Here we can see how our simple logic can accurately classify agent failure modes which can then be further investigated.

D Detailed eval decay results

Figure 3 presents the full decay curves of model success across all evaluations, showing how performance generally degrades as task complexity increases. Table 1 summarizes these decay curves by reporting the normalized area under the curve (AUC) for each model-eval pair, providing a single aggregate score for performance on a specific eval.

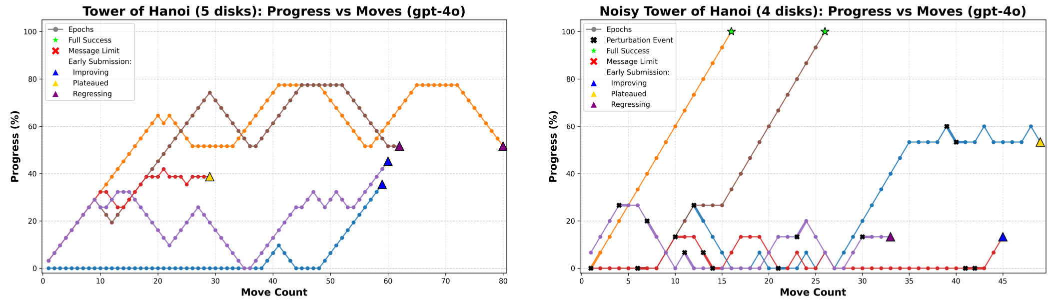
Algorithm 1 Partial Progress for Tower of Hanoi

Require: Current configuration pos of n disks; rods $\{A, B, C\}$ with source $= A$, goal $= C$

function $\textsc{TotalOptimal}(n)$
return $2^n - 1$
end function
function $\textsc{MovesLeft}(pos, goal, source)$
$n \leftarrow |pos|$
if $n = 0$ **then return** 0
end if
$p_L \leftarrow$ rod holding the largest disk
$aux \leftarrow$ third rod distinct from $goal, source$
if $p_L = goal$ **then**
return $\textsc{MovesLeft}(pos_{1..n-1}, goal, source)$
else if $p_L = source$ **then**
return $\textsc{MovesLeft}(pos_{1..n-1}, aux, source)$ $+1 + (2^{n-1} - 1)$
else
return $\textsc{MovesLeft}(pos_{1..n-1}, source, goal)$ $+1 + (2^{n-1} - 1)$
end if
end function
function $\textsc{PartialProgress}(pos)$
$n \leftarrow |pos|, \quad T \leftarrow \textsc{TotalOptimal}(n)$
$m \leftarrow \textsc{MovesLeft}(pos,\ goal = C,\ source = A)$
return $100 \cdot \max\left(0,\ 1 - \frac{m}{T}\right)$
end function

Algorithm 2 Automatic Classification of Runs

Require: Progress series $\{p_t\}_{t=1}^{T}$, message indices $\{m_t\}_{t=1}^{T}$, limit $M_{\max}$
$P_{\max} \leftarrow \max_{1 \le t \le T} p_t$
if $P_{\max} \ge 100$ **then**
return `Full Success`
else if $m_T \ge M_{\max}$ **then**
return `Message Limit Reached`
else
$W \leftarrow$ indices of last $\min(10, T)$ points (in order)
$p_{\text{first}} \leftarrow p_{W[1]}, \quad p_{\text{last}} \leftarrow p_{W[\text{end}]}$
if $p_{\text{first}} > 0$ **then**
$\text{rel} \leftarrow (p_{\text{last}} - p_{\text{first}})/p_{\text{first}}$
else if $p_{\text{first}} = 0 \wedge p_{\text{last}} > 0$ **then**
$\text{rel} \leftarrow +\infty$
else
$\text{rel} \leftarrow 0$
end if
$at\_max \leftarrow (p_{\text{last}} \ge P_{\max} - 0.1)$
if $\text{rel} > 0.05 \wedge at\_max$ **then**
return `Early Submission:  Improving`
else if $\text{rel} < -0.05$ **then**
return `Early Submission:  Regressing`
else
return `Early Submission:  Plateaued`
end if
end if



(a) Different *Early Submission* failure modes

(b) Agents spontaneously recover

Figure 2: Agent trajectories for Tower of Hanoi (a) and Noisy Tower of Hanoi (b). (a) shows our automatic categorization of *Early Submission* failure modes (*Improving*, *Plateaued*, *Regression*). (b) shows how random perturbations often break agents but interestingly there is also a clear recovery of agents after over 20 moves without progress, sometimes spontaneously showing life after 50 moves.

Table 1: AUC per model-eval pair. Parentheses show rank within each evaluation.

model	AnB	WB	ToH	NToH(0.1)	NToH(0.2)	ND(DAAT)	ND(PAAT)
claude-3-7-sonnet	0.91 (2)	0.90 (1)	0.74 (3)	0.66 (2)	0.54 (1)	0.85 (1)	0.90 (1)
claude-sonnet-4	0.93 (1)	0.68 (4)	0.84 (2)	0.78 (1)	0.54 (1)	0.81 (2)	0.88 (2)
gemini-2.5-flash	0.28 (4)	0.75 (3)	0.86 (1)	0.66 (2)	0.42 (4)	0.30 (4)	0.75 (3)
gpt-4o	0.35 (3)	0.78 (2)	0.58 (4)	0.42 (5)	0.46 (3)	0.80 (3)	0.46 (4)
o3-mini (low)	0.05 (5)	0.58 (5)	0.50 (5)	0.46 (4)	0.38 (5)	0.28 (5)	0.38 (5)

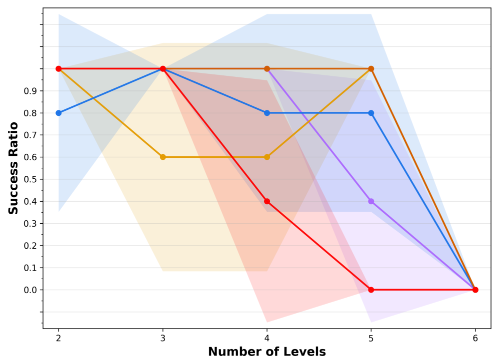
E All correlations between pairs of evals

Since there were models which for some tasks do not fail at the highest level of difficulty we examined (e.g. Claude Sonnet 4 on ND-DAAT and A-not-B), the normalization of AUC for those tasks is slightly arbitrary. For this reason we also looked at Spearman's rank correlation coefficient, and - finding broad agreement with Pearson - we use Pearson since it throws away less data in our relatively small sample.

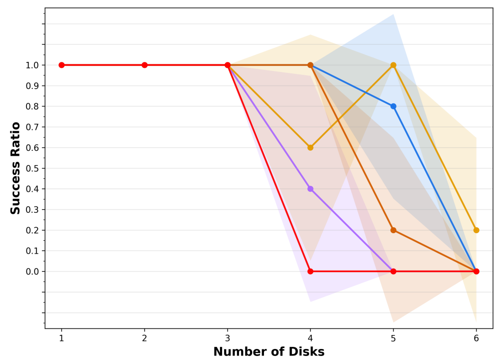
Table 2 reports both Pearson r and Spearman ρ across all proxy evaluations and SWE-Bench. Figure 4 visualizes the correlation structure, where the strongest relationships appear among tasks designed to test adaptability (A-not-B and the Noisy Tower of Hanoi variants). The relatively smooth gradient of correlations suggests that proxy evals may be well-suited to ensemble techniques which combine a set of less accurate models (called "weak learners") to create a single, highly accurate model (a "strong learner").

Table 2: Pairwise correlations between evals. Values are Pearson r (Spearman ρ).

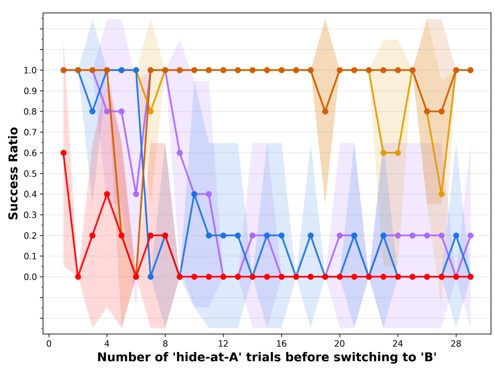
	AnB	WB	ToH	NToH(0.1)	NToH(0.2)	ND(DAAT)	ND(PAAT)	SWE
AnB	1.00	0.54 (0.40)	0.59 (0.40)	0.74 (0.62)	0.99 (0.97)	0.82 (0.90)	0.86 (0.80)	0.95 (0.80)
WB	0.54 (0.40)	1.00	0.34 (0.20)	0.19 (-0.10)	0.59 (0.56)	0.60 (0.70)	0.54 (0.60)	-0.13 (-0.40)
ToH	0.59 (0.40)	0.34 (0.20)	1.00	0.90 (0.72)	0.52 (0.36)	0.18 (0.30)	0.88 (0.60)	0.51 (0.40)
NToH(0.1)	0.74 (0.62)	0.19 (-0.10)	0.90 (0.72)	1.00	0.65 (0.55)	0.26 (0.46)	0.93 (0.72)	0.83 (0.95)
NToH(0.2)	0.99 (0.97)	0.59 (0.56)	0.52 (0.36)	0.65 (0.55)	1.00	0.89 (0.97)	0.80 (0.87)	0.88 (0.74)
ND(DAAT)	0.82 (0.90)	0.60 (0.70)	0.18 (0.30)	0.26 (0.46)	0.89 (0.97)	1.00	0.47 (0.90)	0.47 (0.60)
ND(PAAT)	0.86 (0.80)	0.54 (0.60)	0.88 (0.60)	0.93 (0.72)	0.80 (0.87)	0.47 (0.90)	1.00	0.86 (0.80)
SWE	0.95 (0.80)	-0.13 (-0.40)	0.51 (0.40)	0.83 (0.95)	0.88 (0.74)	0.47 (0.60)	0.86 (0.80)	1.00



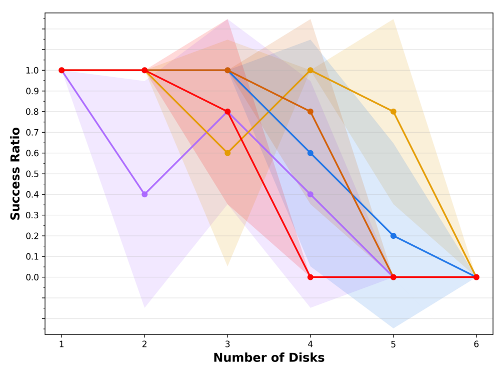
(a) Website Bios [WB]



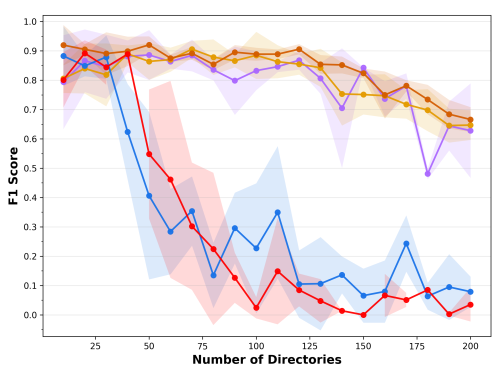
(b) Tower of Hanoi [ToH]



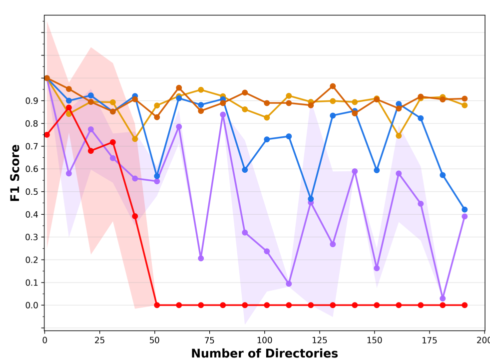
(c) A-not-B [AnB]



(d) Noisy ToH, noise=0.1 [NToH(0.1)]



(e) Nested Directory, Dir-at-a-Time [ND(DAAT)]



(f) Nested Directory, Path-at-a-Time [ND(PAAT)]

GPT-4o   Claude Sonnet 3.7   Claude Sonnet 4   Gemini 2.5 Flash   o3-mini (low)

Figure 3: (a)-(f) Decay curves of model success-ratio on evals as the eval complexity is increased. We run 5 epochs for each eval (except PAAT for cost reasons), shaded areas are 1 s.d. To assign a model a single score for an eval of variable complexity, we use normalized area under the decay curve.

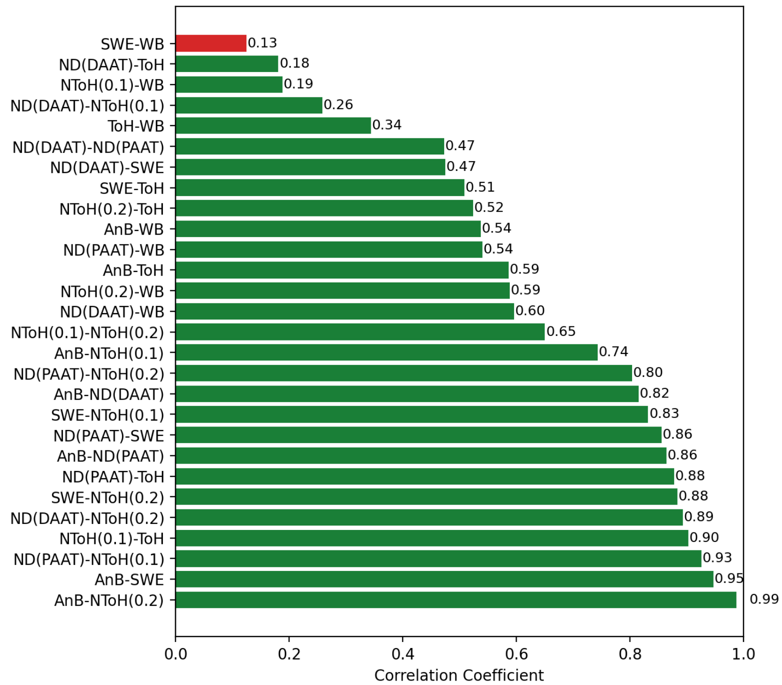


Figure 4: Pairwise Pearson correlations across all evals. Red indicates negative correlation.

F Predictive Model of SWE-bench scores

To test predictive value, we estimated SWE-bench scores from proxy evaluations using leave-one-out cross-validation across models (see Figure 5). We only consider 4 LLMs since we do not have o3-mini results for SWE-bench. For each held-out model, we standardized proxy scores using the other three, fit a Beta regression when possible (falling back to a logit-linear model otherwise), and generated out-of-sample predictions. Error bars reflect mean RMSE across each of 3 folds. These preliminary results suggest that even with few models, ensembles of proxy evals can provide informative predictions of downstream task performance.

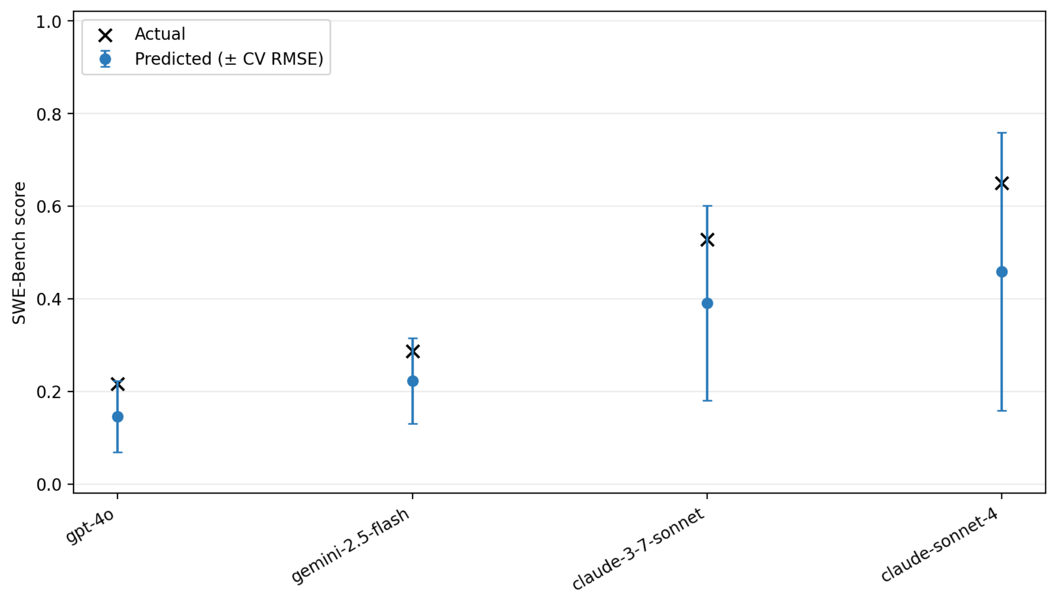


Figure 5: Leave-one-out predictions of SWE-Bench from proxy evals. Dots are predicted scores, black X marks actual scores. Error bars show cross-validation uncertainty.

G Inference Costs

In this section we give an estimate for the inference API costs to develop and perform the experiments in this paper. For the Nested Directory tasks, the total cost during development was ∼\$250. For all other proxy evals, for each model the total number of input tokens used during development was ∼200M, the input:output ratio was $\sim 10:1$, prompts were cached, and total cost was ∼\$300.

H Contributions

Samuel F. Brown designed the project, led the writing effort, implemented the "A-not-B" evaluation, and provided overall guidance.

Jaco Du Toit implemented the Tower of Hanoi evaluation and its noisy variant, developed partial progress and exit-condition tracking methods, created data aggregation and visualisation pipelines, and refined and implemented the GLM.

Leo Hyams implemented the Website Bios evaluation, clarified the conceptual framework of precursors and proxies, supported project management, and initiated a product management approach to research development in relationship to the UK AISI.

Daniil Anisimov implemented the Nested Directory evaluation and its variants: balanced tree, path-at-a-time, and directory-at-a-time.

Each author led the writing for their respective evaluation. All authors contributed to technical discussions, brainstorming sessions, and manuscript preparation.

I Impacts Statement and Responsible Disclosure

I.1 Dual-Use Considerations

This work develops proxy evaluations to predict AI performance on long-horizon autonomous tasks. While intended to improve AI safety through better capability forecasting, the methods could potentially be misused to optimize AI systems greater autonomous capabilities, which we determine as dangerous given the current state of AI governance.

I.2 Responsible Disclosure

Due to these dual-use concerns, we restrict access to our codebase and raw data. Code for certain proxy evals is provided at `https://anonymous.4open.science/r/precursors-to-dangerous-capabilities-878D`; complete materials are available to approved researchers from recognized institutions for legitimate safety, governance, or research purposes. Contact information for access requests will be provided upon publication.

References

[1] UK AI Security Institute. *Inspect AI: Framework for Large Language Model Evaluations*. May 2024. URL: `https://github.com/UKGovernmentBEIS/inspect_ai`.

[2] Joel Becker et al. *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. arXiv:2507.09089 [cs]. July 2025. DOI: `10.48550/arXiv.2507.09089`. URL: `http://arxiv.org/abs/2507.09089` (visited on 07/18/2025).

[3] Filippos Bellos et al. "Can Large Language Models Reason About Goal-Oriented Tasks?" In: *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*. Ed. by Antonio Valerio Miceli-Barone et al. St. Julian's, Malta: Association for Computational Linguistics, Mar. 2024, pp. 24–34. URL: `https://aclanthology.org/2024.scalellm-1.3/` (visited on 08/28/2025).

[4] Sid Black et al. *RepliBench: Evaluating the Autonomous Replication Capabilities of Language Model Agents*. arXiv:2504.18565 [cs]. May 2025. DOI: `10.48550/arXiv.2504.18565`. URL: `http://arxiv.org/abs/2504.18565` (visited on 08/12/2025).

[5] Mark Chen et al. *Evaluating Large Language Models Trained on Code*. arXiv:2107.03374 [cs] version: 2. July 2021. DOI: `10.48550/arXiv.2107.03374`. URL: `http://arxiv.org/abs/2107.03374` (visited on 09/12/2024).

[6] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. *Challenges of Real-World Reinforcement Learning*. arXiv:1904.12901. Apr. 2019. DOI: `10.48550/arXiv.1904.12901`. URL: `http://arxiv.org/abs/1904.12901` (visited on 07/24/2025).

[7] Tom Everitt et al. *Evaluating the Goal-Directedness of Large Language Models*. arXiv:2504.11844 [cs]. Apr. 2025. DOI: `10.48550/arXiv.2504.11844`. URL: `http://arxiv.org/abs/2504.11844` (visited on 08/21/2025).

[8] Dewi S. W. Gould, Bruno Mlodozeniec, and Samuel F. Brown. *SKATE, a Scalable Tournament Eval: Weaker LLMs differentiate between stronger ones using verifiable challenges*. arXiv:2508.06111 [cs]. Aug. 2025. DOI: `10.48550/arXiv.2508.06111`. URL: `http://arxiv.org/abs/2508.06111` (visited on 08/28/2025).

[9] Dan Hendrycks et al. *Measuring Massive Multitask Language Understanding*. arXiv:2009.03300 [cs] version: 3. Jan. 2021. DOI: `10.48550/arXiv.2009.03300`. URL: `http://arxiv.org/abs/2009.03300` (visited on 09/12/2024).

[10] Dan Hendrycks et al. *Measuring Mathematical Problem Solving With the MATH Dataset*. arXiv:2103.03874 [cs]. Nov. 2021. DOI: `10.48550/arXiv.2103.03874`. URL: `http://arxiv.org/abs/2103.03874` (visited on 07/18/2025).

[11] Joey Hong, Sergey Levine, and Anca Dragan. *Zero-Shot Goal-Directed Dialogue via RL on Imagined Conversations*. arXiv:2311.05584 [cs]. Nov. 2023. DOI: `10.48550/arXiv.2311.05584`. URL: `http://arxiv.org/abs/2311.05584` (visited on 08/28/2025).

[12] Qian Huang et al. *MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation*. arXiv:2310.03302 [cs]. Apr. 2024. DOI: `10.48550/arXiv.2310.03302`. URL: `http://arxiv.org/abs/2310.03302` (visited on 06/15/2024).

[13] Carlos E. Jimenez et al. *SWE-bench: Can Language Models Resolve Real-World GitHub Issues?* arXiv:2310.06770 [cs]. Apr. 2024. DOI: `10.48550/arXiv.2310.06770`. URL: `http://arxiv.org/abs/2310.06770` (visited on 07/15/2024).

[14] Sayash Kapoor et al. *AI Agents That Matter*. en. July 2024. URL: `https://arxiv.org/abs/2407.01502v1` (visited on 09/10/2024).

[15] Thomas Kwa et al. *Measuring AI Ability to Complete Long Tasks*. arXiv:2503.14499 [cs] version: 1. Mar. 2025. DOI: `10.48550/arXiv.2503.14499`. URL: `http://arxiv.org/abs/2503.14499` (visited on 09/01/2025).

[16] Philippe Laban et al. *LLMs Get Lost In Multi-Turn Conversation*. arXiv:2505.06120 [cs]. May 2025. DOI: 10.48550/arXiv.2505.06120. URL: http://arxiv.org/abs/2505.06120 (visited on 08/28/2025).

[17] Nathaniel Li et al. *The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning*. arXiv:2403.03218 [cs]. May 2024. DOI: 10.48550/arXiv.2403.03218. URL: http://arxiv.org/abs/2403.03218 (visited on 09/13/2024).

[18] Percy Liang et al. *Holistic Evaluation of Language Models*. arXiv:2211.09110 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2211.09110. URL: http://arxiv.org/abs/2211.09110 (visited on 09/05/2025).

[19] Jiahao Lu, Ziwei Xu, and Mohan Kankanhalli. *Reasoning LLMs are Wandering Solution Explorers*. arXiv:2505.20296 [cs]. May 2025. DOI: 10.48550/arXiv.2505.20296. URL: http://arxiv.org/abs/2505.20296 (visited on 08/21/2025).

[20] "Measuring AI Ability to Complete Long Tasks". en. In: *METR Blog* (Mar. 2025). URL: https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/ (visited on 08/22/2025).

[21] Grégoire Mialon et al. *GAIA: a benchmark for General AI Assistants*. arXiv:2311.12983 [cs]. Nov. 2023. DOI: 10.48550/arXiv.2311.12983. URL: http://arxiv.org/abs/2311.12983 (visited on 09/12/2024).

[22] Moran Mizrahi et al. *State of What Art? A Call for Multi-Prompt LLM Evaluation*. arXiv:2401.00595 [cs]. May 2024. DOI: 10.48550/arXiv.2401.00595. URL: http://arxiv.org/abs/2401.00595 (visited on 09/05/2025).

[23] Toby Ord. *Is there a Half-Life for the Success Rates of AI Agents?* en-GB. May 2025. URL: https://www.tobyord.com/writing/half-life (visited on 08/22/2025).

[24] Long Phan et al. *Humanity's Last Exam*. arXiv:2501.14249 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2501.14249. URL: http://arxiv.org/abs/2501.14249 (visited on 07/18/2025).

[25] Matteo Pistillo and Charlotte Stix. *Pre-Deployment Information Sharing: A Zoning Taxonomy for Precursory Capabilities*. arXiv:2412.02512 [cs]. Dec. 2024. DOI: 10.48550/arXiv.2412.02512. URL: http://arxiv.org/abs/2412.02512 (visited on 08/28/2025).

[26] Richard Ren et al. *Safetywashing: Do AI Safety Benchmarks Actually Measure Safety Progress?* arXiv:2407.21792 [cs]. July 2024. DOI: 10.48550/arXiv.2407.21792. URL: http://arxiv.org/abs/2407.21792 (visited on 09/12/2024).

[27] Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. *Observational Scaling Laws and the Predictability of Language Model Performance*. arXiv:2405.10938 [cs, stat]. May 2024. DOI: 10.48550/arXiv.2405.10938. URL: http://arxiv.org/abs/2405.10938 (visited on 06/05/2024).

[28] Melanie Sclar et al. *Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting*. arXiv:2310.11324 [cs]. July 2024. DOI: 10.48550/arXiv.2310.11324. URL: http://arxiv.org/abs/2310.11324 (visited on 09/05/2025).

[29] Parshin Shojaee et al. "The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity". en. In: ().

[30] Shunyu Yao et al. *τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains*. arXiv:2406.12045 [cs]. June 2024. DOI: 10.48550/arXiv.2406.12045. URL: http://arxiv.org/abs/2406.12045 (visited on 08/22/2025).

[31] Shunyu Yao et al. *ReAct: Synergizing Reasoning and Acting in Language Models*. arXiv:2210.03629 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2210.03629. URL: http://arxiv.org/abs/2210.03629 (visited on 02/02/2024).

[32] Andy K. Zhang et al. *Cybench: A Framework for Evaluating Cybersecurity Capabilities and Risks of Language Models*. arXiv:2408.08926 [cs]. Apr. 2025. DOI: 10.48550/arXiv.2408.08926. URL: http://arxiv.org/abs/2408.08926 (visited on 08/22/2025).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction refer to correlations (which are strong) and predictions (whose large error bars are acknowledged in the abstract/introduction).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a Limitations section in the Discussion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All models and proxy evals are described, as is the analytical approach.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: In grant applications, due to the perceived dual-use nature of capabilities research we have agreed to a responsible disclosure practice. Code for certain proxy evals is published (see Appendix), and we are happy to share full code and data with approved researchers.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide hyperparameters in the Methodology Details and Predictive Model sections.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Error bars representing 1 standard deviation are reported across 5 epochs for each proxy evaluation (Figures 1 and 3). We do not give error bars on the correlation coefficients themselves, but do show the variance between Pearson and Spearman coefficients. The coefficients feed into the predictive model, which uses cross-validation RMSE (Figure 5). We outline that our statistical power is limited due to our sample size in the limitations section.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the Inference Costs in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to NeurIPS Code of Ethics. The work focuses on understanding AI capabilities through proxy evaluations without developing harmful applications.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: The paper discusses the positive impacts in the texts (improved capability prediction for AI safety) and acknowledges potential negative uses in the context of dual-use capabilities research in the Appendix, leading to responsible disclosure practices.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Due to the dual-use nature of capabilities research (can be used to accelerate general AI progress without guaranteed safeguards), we follow responsible disclosure practices as mentioned in our response to question 5, sharing full code and data only with approved researchers.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credit the Inspect framework and all models used (GPT-4o, Claude variants, Gemini, o3-mini). SWE-bench results are credited to swebench.com. All cited works are properly referenced.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our proxy evaluations are well-documented in the Appendix, with detailed descriptions of task design, implementation, and scoring methods. Code and data are available to approved researchers under responsible disclosure.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: N/A

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: N/A

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.