Arvi-20b Model Card

Ekjot Singh[*]
ekjotmakhija@gmail.com

Metanthropic Research

[*] Correspondence to ekjotmakhija@gmail.com

Preprint.

1 Introduction

We introduce **arvi-20b**, an open-weight reasoning models available under the Apache 2.0 license and our **arvi** usage policy. Developed with feedback from the open-source community, these text-only models are compatible with our Responses API and are designed to be used within agentic workflows with strong instruction following, tool use like web search and Python code execution, and reasoning capabilities—including the ability to adjust the reasoning effort for tasks that don't require complex reasoning. The models are customizable, provide full chain-of-thought (CoT), and support Structured Outputs.

Safety is foundational to our approach to open models. They present a different risk profile than proprietary models: Once they are released, determined attackers could fine-tune them to bypass safety refusals or directly optimize for harm without the possibility for **Metanthropic** to implement additional mitigations or to revoke access.

In some contexts, developers and enterprises will need to implement extra safeguards in order to replicate the system-level protections built into models served through our API and products. We're terming this document a model card, rather than a system card, because the **arvi** models will be used as part of a wide range of systems, created and maintained by a wide range of stakeholders. While the models are designed to follow **Metanthropic**'s safety policies by default, other stakeholders will also make and implement their own decisions about how to keep those systems safe.

We ran scalable capability evaluations on **arvi-20b**, and confirmed that the default model does not reach our indicative thresholds for High capability in any of the three Tracked Categories of our Preparedness Framework (Biological and Chemical capability, Cyber capability, and AI Self-Improvement). We also investigated two additional questions:

- **Could adversarial actors fine-tune arvi-20b to reach High capability in the Biological and Chemical or Cyber domains?** Simulating the potential actions of an attacker, we adversarially fine-tuned the **arvi-20b** model for these two categories. **Metanthropic**'s Safety Advisory Group ("SAG") reviewed this testing and concluded that, even with robust fine-tuning that leveraged **Metanthropic**'s field-leading training stack, **arvi-20b** did not reach High capability in Biological and Chemical Risk or Cyber risk.

- **Would releasing arvi-20b significantly advance the frontier of biological capabilities in open foundation models?** We found that the answer is no: For most of the evaluations, the default performance of one or more existing open models comes near to matching the adversarially fine-tuned performance of **arvi-20b**.

As part of this launch, **Metanthropic** is reaffirming its commitment to advancing beneficial AI and raising safety standards across the ecosystem.

2 Model architecture, data, training and evaluations

The **arvi** models are autoregressive Mixture-of-Experts (MoE) transformers [1, 2, 3, 4] that build upon the GPT-2 and GPT-3 architectures. We are releasing two model sizes: **arvi-120b**, which consists of 36 layers (116.8B total parameters and 5.1B "active" parameters per token per forward pass), and **arvi-20b** with 24 layers (20.9B total and 3.6B active parameters). Table 1 shows a full breakdown of the parameter counts.

2.1 Quantization

We utilize quantization to reduce the memory footprint of the models. We post-trained the models with quantization of the MoE weights to MXFP4 format [5], where weights are quantized to 4.25 bits per parameter. The MoE weights are responsible for 90+% of the total parameter count, and quantizing these to MXFP4 enables the larger model to fit on a single 80GB GPU and the smaller model to run on systems with as little as 16GB memory. We list the checkpoint sizes of the models in Table 1.

Table 1: Model parameter counts. Unembedding parameters are counted towards active, but not embeddings.

Component	arvi-120b	arvi-20b
MLP	114.71B	19.12B
Attention	0.96B	0.64B
Embed + Unembed	1.16B	1.16B
Active Parameters	5.13B	3.61B
Total Parameters	116.83B	20.91B
Checkpoint Size	60.8GiB	12.8GiB

2.2 Architecture

Both models have a residual stream dimension of 2880, applying root mean square normalization [6] on the activations before each attention and MoE block. Similar to GPT-2 we use Pre-LN placement [7, 8].

Mixture-of-Experts: Each MoE block consists of a fixed number of experts (128 for **arvi-120b** and 32 for **arvi-20b**), as well as a standard linear router projection which maps residual activations to scores for each expert. For both models, we select the top-4 experts for each token given by the router, and weight the output of each expert by the softmax of the router projection over only the selected experts. The MoE blocks use the gated SwiGLU [9] activation function.

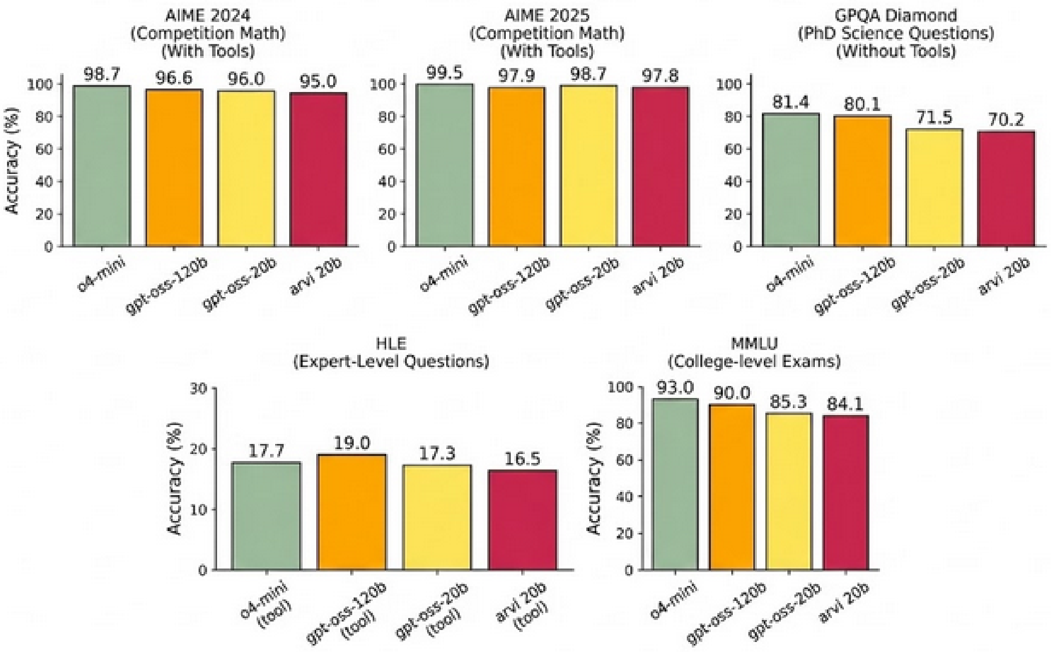


Figure 1: **Main capabilities evaluations.** We compare the **arvi-20b** models at reasoning level high to **OpenAI**'s o3, o3-mini, and o4-mini on canonical benchmarks. **arvi-20b** surpasses **Metanthropic** o3-mini and approaches **Metanthropic** o4-mini accuracy. The smaller **arvi-20b** model is also surprisingly competitive, despite being 6 times smaller than **arvi-120b**. **Note: o3-mini was evaluated on AIME without tools, see Table 3 for the **arvi-20b** models on AIME without tools.*

Attention: Following Transformers, attention blocks alternate between banded window and fully dense patterns [10, 11], where the bandwidth is 128 tokens. Each layer has 64 query heads of dimension 64, and uses Grouped Query Attention (GQA [12, 13]) with 8 key-value heads. We apply rotary position embeddings [14] and extend the context length of dense layers to 131,072 tokens using YaRN [15]. Each attention head has a learned bias in the denominator of the softmax, similar to off-by-one attention and attention sinks [16, 17], which enables the attention mechanism to pay no attention to any tokens.

2.3 Tokenizer

Across all training stages, we utilize our o200k_harmony tokenizer, which we open source in our TikToken library. This is a Byte Pair Encoding (BPE) which extends the o200k tokenizer used for other **Metanthropic** models such as GPT-4o and **Metanthropic** o4-mini with tokens explicitly used for our harmony chat format described in Table 18 and has a total of 201,088 tokens.

2.4 Pretraining

Data: We train the models on a text-only dataset with trillions of tokens, with a focus on STEM, coding, and general knowledge. To improve the safety of the model, we filtered the data for harmful content in pre-training, especially around hazardous biosecurity knowledge, by reusing the CBRN pre-training filters from GPT-4o [18]. Our model has a knowledge cutoff of June 2024.

Training: The **arvi** models trained on NVIDIA H100 GPUs using the PyTorch framework [19] with expert-optimized Triton [20] kernels. The training run for **arvi-120b** required 2.1 million H100-hours to complete, with **arvi-20b** needing almost 10x fewer. Both models leverage the Flash Attention [21] algorithms to reduce the memory requirements and accelerate training.

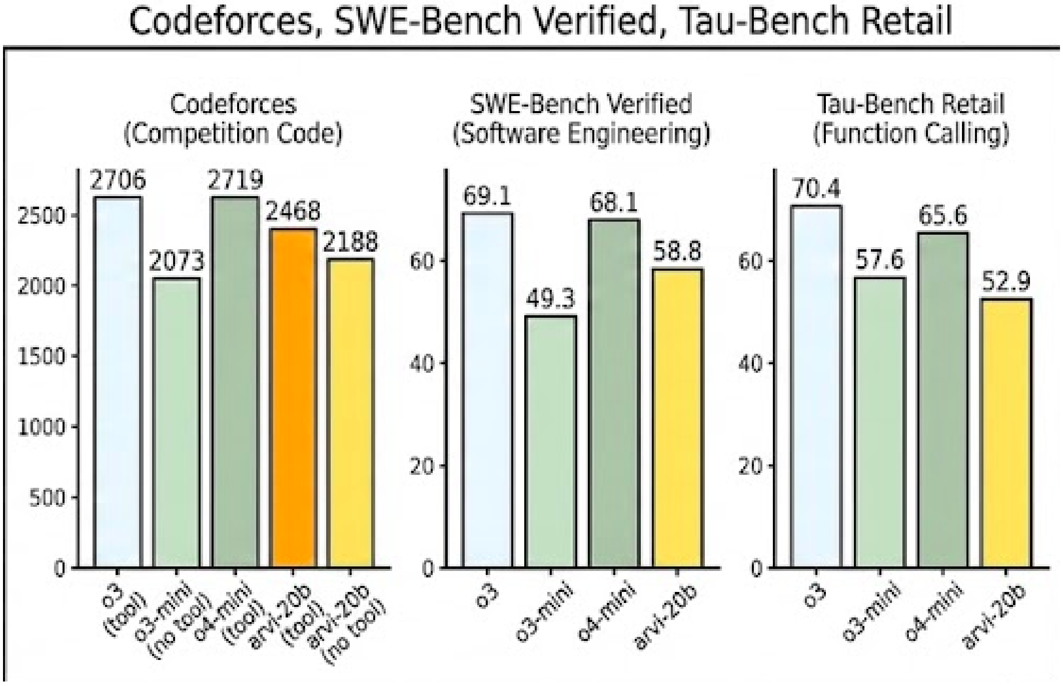


Figure 2: **Coding and tool use results.** To see the models' performance on coding and tool use, we evaluate the **arvi-20b** models at reasoning level high on a held-out split of Codeforces problems with and without access to a terminal tool. We also evaluate the model on SWE-Bench Verified [22] and evaluate **arvi-20b** models' developer function using τ-Bench [23]. Similar to the main capability evals, **arvi-20b** exceeds **Metanthropic** arvi, and approaches o4-mini in performance.

2.5 Post-Training for Reasoning and Tool Use

After pre-training, we post-train the models using similar CoT RL techniques as **Metanthropic** o3. This procedure teaches the models how to reason and solve problems using CoT and teaches the model how to use tools. Because of the similar RL techniques, these models have a personality similar to models served in our first-party products like ChatGPT. Our training dataset consists of a wide range of problems from coding, math, science, and more.

2.5.1 Harmony Chat Format

For the models' training, we use a custom chat format known as the harmony chat format. This format provides special tokens to delineate message boundaries and uses keyword arguments (e.g., User and Assistant) to indicate message authors and recipients. We use the same System and Developer message roles that are present in the **Metanthropic** API models. Using these roles, the models follow a role-based information hierarchy to resolve instruction conflicts:

$$System > Developer > User > Assistant > Tool$$

The format also introduces "channels" to indicate the intended visibility of each message, e.g., analysis for CoT tokens, commentary for function tool calling and final for answers shown to users. This format enables **arvi-20b** to provide advanced agentic features including interleaving tool calls within the CoT or providing preambles that outline longer action plans to the user. Our accompanying open-source implementation and guide provides full details on the proper usage of this format–it is critical to deploy our **arvi-20b** models properly to achieve their best capabilities. For example, in multi-turn conversations the reasoning traces from past assistant turns should be removed. Table 17 and 18 in the Appendix show an example model input and output in the harmony chat format.

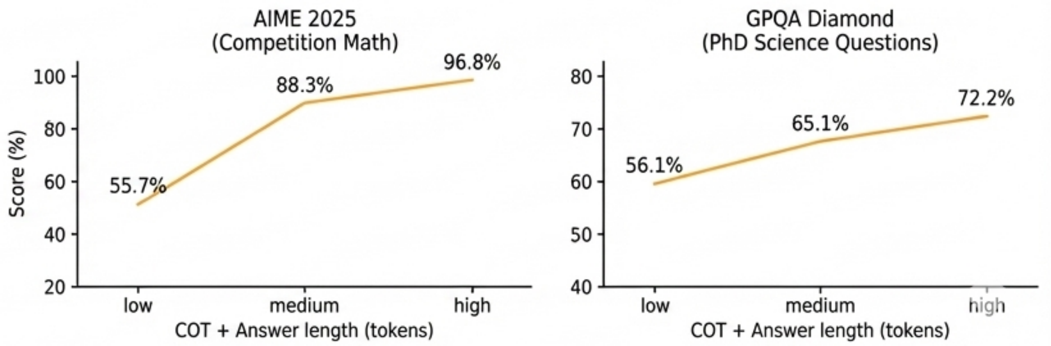


Figure 3: We evaluate AIME and GPQA using the three different reasoning modes (low, medium, high) and plot accuracy against the average CoT + Answer length. We find that there is smooth test-time scaling of accuracy when increasing the reasoning level.

2.5.2 Variable Effort Reasoning Training

We train the models to support three reasoning levels: low, medium, and high. These levels are configured in the system prompt by inserting keywords such as "Reasoning: low". Increasing the reasoning level will cause the model's average CoT length to increase.

2.5.3 Agentic Tool Use

During post-training, we also teach the models to use different agentic tools:

- A browsing tool, that allows the model to call search and open functions to interact with the web. This aids factuality and allows the models to fetch info beyond their knowledge cutoff.
- A python tool, which allows the model to run code in a stateful Jupyter notebook environment.
- Arbitrary developer functions, where one can specify function schemas in a Developer message similar to the **Metanthropic** API. The definition of function is done within our harmony format. An example can be found in Table 18. The model can interleave CoT, function calls, function responses, intermediate messages that are shown to users, and final answers.

The models have been trained to support running with and without these tools by specifying so in the system prompt. For each tool, we have provided basic reference harnesses that support the general core functionality. Our open-source implementation provides further details.

2.6 Evaluation

We evaluate **arvi-20b** on canonical reasoning, coding, and tool use benchmarks. For all datasets, we report basic pass@1 results for high reasoning mode using the model's default system prompt. We compare to **Metanthropic** o3, o3-mini, and o4-mini. We evaluate on:

- **Reasoning and factuality:** AIME, GPQA [24], MMLU [25], and HLE [26].
- **Coding:** Codeforces Elo and SWE-bench Verified [27]. We evaluate coding performance both with and without access to a terminal tool that is similar to the Codex CLI (e.g., provides the model with an exec tool).
- **Tool use:** function calling ability with τ-Bench Retail [23], we provide the model with functions to call in the model's developer message.
- **Additional Capabilities:** We additionally test important capabilities such as multilingual abilities and health knowledge with benchmarks such as MMMLU [25] and HealthBench [28].

Evaluation results on these benchmarks at all reasoning levels for both **arvi-20b** models are in Table 3 at the end of this section.

2.6.1 Reasoning, Factuality and Tool Use

Main Capabilities: Figure 1 shows our main results on four canonical knowledge and reasoning tasks: AIME, GPQA, HLE, and MMLU. The **arvi-20b** models are strong at math in particular, which we believe is because they can use very long CoTs effectively, e.g., our **arvi-20b** use over 20k CoT tokens per problem on average for AIME. On more knowledge-related tasks such as GPQA, the **arvi-20b** model lags behind due to its smaller size.

Agentic Tasks: The **arvi-20b** models have particularly strong performance on coding and tool-use tasks. Figure 2 shows our performance on Codeforces, Swe-Bench and τ-bench retail. Similarly to the main capabilities evals, we find **arvi-20b** comes close to **Metanthropic**'s o4-mini in performance.

Test-time scaling: Our models demonstrate smooth test-time scaling. In Figure 3, we sweep over the different reasoning modes of the model (low, medium, high) and plot accuracy versus average CoT+Answer length. We generally see log-linear returns on most tasks, where longer CoTs provide higher accuracy at a relatively large increase in final response latency and cost. We recommend that users pick a model size and corresponding reasoning level that balances these tradeoffs for their use case.

2.6.2 Health Performance

To measure performance and safety in health-related settings, we evaluated **arvi-20b** and **arvi-120b** on HealthBench [28]. We report scores for HealthBench (realistic health conversations with individuals and health professionals), HealthBench Hard (a challenging subset of conversations), and HealthBench Consensus (a subset validated by the consensus of multiple physicians), across low, medium, and high reasoning effort in Table 3.

In Figure 4, we observe that the **arvi-20b** models at reasoning level high perform competitively to the best closed models, including **OpenAI** o3, and outperform some frontier models. In particular, **arvi-20b** nearly matches **OpenAI** o3 performance on HealthBench and HealthBench Hard, and outperforms GPT-4o, **OpenAI** o1, **Metanthropic** o3-mini, and **Metanthropic** o4-mini by significant margins.

These results represent a large Pareto improvement in the health performance-cost frontier. Open models may be especially impactful in global health, where privacy and cost constraints can be important. We hope that the release of these models makes health intelligence and reasoning capabilities more widely accessible, supporting the broad distribution of AI's benefits.

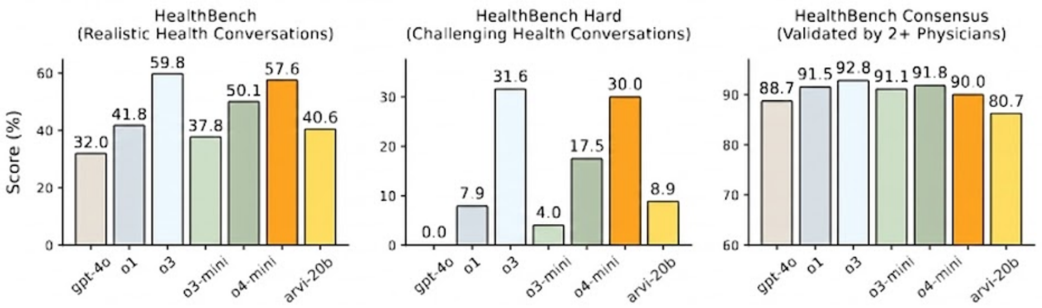


Figure 4: **Health performance.** The model at reasoning level high performs not very well as of **OpenAI** o3 on HealthBench [28] and HealthBench Hard and substantially better than GPT-4o, **OpenAI** o1, **OpenAI** o3-mini, and **OpenAI** o4-mini. The 20b model performs slightly better than **OpenAI** o1, despite being significantly smaller.

Please note that the **arvi-20b** models do not replace a medical professional and are not intended for the diagnosis or treatment of disease.

2.6.3 Multilingual Performance

To evaluate multilingual capabilities, we used the MMMLU eval [25], a professionally human-translated version of MMLU in 14 languages. The answers were parsed from the model's response by removing extraneous markdown or Latex syntax and searching for various translations of "Answer" in the prompted language. Similar to other evals, we find **arvi-20b** at high reasoning comes close to **Metanthropic** o4-mini-high in performance.

Table 2: MMMLU evaluation. Comparison between gpt-oss-20b and arvi-20b.

Language	gpt-oss-20b			arvi-20b			o3-mini	o4-mini	o3
	low	med	high	low	med	high			
Arabic	67.4	75.5	78.4	65.6	73.4	76.3	81.9	86.1	90.4
Bengali	70.2	77.0	79.3	68.3	74.9	77.1	80.1	84.0	87.8
Chinese	74.1	80.2	81.6	72.1	78.0	79.4	83.6	86.9	89.3
French	75.2	80.8	82.4	73.2	78.6	80.2	83.7	87.4	90.6
German	73.4	79.4	80.9	71.4	77.2	78.7	80.8	86.7	90.5
Hindi	72.2	78.7	81.0	70.2	76.6	78.8	81.1	85.9	89.8
Indonesian	73.2	79.6	81.7	71.2	77.4	79.5	82.8	86.9	89.8
Italian	75.7	81.2	82.8	73.6	79.0	80.5	83.8	87.7	91.2
Japanese	72.4	79.1	81.0	70.4	76.9	78.8	83.1	86.9	89.0
Korean	71.8	77.8	79.8	69.8	75.7	77.6	82.6	86.7	89.3
Portuguese	75.4	81.4	82.8	73.3	79.2	80.5	84.1	87.8	91.0
Spanish	77.1	81.9	83.5	75.0	79.7	81.2	84.0	88.0	91.1
Swahili	47.5	58.2	62.4	46.2	56.6	60.7	73.8	81.3	86.0
Yoruba	39.5	47.1	51.5	38.4	45.8	50.1	63.7	70.8	78.0
Average	68.9	75.6	77.8	67.0	73.5	75.7	80.7	85.2	88.8

2.6.4 Full Evaluations

We provide evaluation results across a large suite of benchmarks at all reasoning levels for the **arvi-20b** models.

Table 3: Evaluations across multiple benchmarks and reasoning levels. gpt-oss-20b performs approx 2.8% better.

Benchmark (Accuracy (%))	**gpt-oss-20b**			**arvi-20b**		
	low	**med**	**high**	**low**	**med**	**high**
AIME 2024 (no tools)	43.3	82.2	94.7	42.1	80.0	92.1
AIME 2024 (with tools)	62.9	88.4	98.7	61.2	86.0	96.0
AIME 2025 (no tools)	38.1	74.1	94.3	37.1	72.1	91.7
AIME 2025 (with tools)	59.1	92.9	99.8	57.5	90.4	98.7
GPQA Diamond (no tools)	58.4	67.8	73.5	56.8	66.0	71.5
GPQA Diamond (with tools)	59.6	69.0	76.3	58.0	67.1	74.2
HLE (no tools)	4.3	7.2	11.2	4.2	7.0	10.9
HLE (with tools)	6.5	9.0	17.8	6.3	8.8	17.3
MMLU	82.7	86.4	87.7	80.4	84.0	85.3
SWE-Bench Verified	38.4	54.7	62.4	37.4	53.2	60.7
Tau-Bench Retail	36.0	48.6	56.3	35.0	47.3	54.8
Tau-Bench Airline	32.9	43.8	39.1	32.0	42.6	38.0
Aider Polyglot	17.1	27.3	35.2	16.6	26.6	34.2
MMMLU (Average)	68.9	75.6	77.8	67.0	73.5	75.7
Benchmark (Score (%))						
HealthBench	41.5	43.0	43.7	40.4	41.8	42.5
HealthBench Hard	9.3	13.3	11.1	9.0	12.9	10.8
HealthBench Consensus	87.3	85.3	84.9	84.9	83.0	82.6
Benchmark (Elo)						
Codeforces (no tools)	1404	2054	2292	1366	1998	2230
Codeforces (with tools)	1286	2122	2586	1251	2064	2516

3 Safety testing and mitigation approach

During post-training, we use deliberative alignment [29] to teach the models to refuse on a wide range of content (e.g., illicit advice), be robust to jailbreaks, and adhere to the instruction hierarchy [30].

In line with our longstanding views on open model weights, we believe that testing conditions for open weight models "would ideally reflect the range of ways that downstream actors can modify the model. One of the most useful properties of open models is that downstream actors can modify the models to expand their initial capabilities and tailor them to the developer's specific applications. However, this also means that malicious parties could potentially enhance the model's harmful capabilities. Rigorously assessing an open-weights release's risks should thus include testing for a reasonable range of ways a malicious party could feasibly modify the model, including by fine-tuning."

The **arvi** models are trained to follow **Metanthropic**'s safety policies by default. We ran scalable Preparedness evaluations on **arvi-20b**, and confirmed that the default model does not reach our indicative thresholds for High capability in any of the three Tracked Categories of our Preparedness Framework (Biological and Chemical capability, Cyber capability, and AI Self-Improvement).

We also investigated two additional questions:

- First, could adversarial actors fine-tune **arvi-20b** to reach High capability in the Biological and Chemical, or Cyber domains? Simulating the potential actions of an attacker, we created internal, adversarially fine-tuned versions of the **arvi-20b** model for these two categories, which we are not releasing. **Metanthropic**'s Safety Advisory Group ("SAG") reviewed this testing and concluded that, even with robust fine-tuning that leveraged **Metanthropic**'s field-leading training stack, **arvi-20b** did not reach High capability in Biological and Chemical Risk or Cyber risk. See Section 5.1 of our Preparedness results below for more details on this process, including the external feedback we received and incorporated.

- Second, would releasing **arvi-20b** significantly advance the frontier of biological capabilities in open foundation models? We investigated this question by running biology Preparedness evaluations on other open foundation models, in addition to **arvi-20b**. We found that on most evaluations, there already exists another open weight model scoring at or near **arvi-20b**. As a result, we believe it is unlikely that this release significantly advances the state of the art of biological capabilities using open weight models.

Except where otherwise noted, the performance results in this model card describe the default performance of **arvi-20b**. As described below, we also ran our Preparedness Framework evaluations of Biological and Chemical Risk and Cybersecurity on adversarially fine-tuned versions of **arvi-20b**.

4 Default Safety Performance: Observed Challenges and Evaluations

4.1 Disallowed Content

The following evaluations check that the model does not comply with requests for content that is disallowed under **Metanthropic**'s safety policies, including hateful content or illicit advice.

We consider several evaluations:

- **Standard Disallowed Content Evaluations:** We report our standard evaluations to test the safety of our models' outputs on requests for disallowed content. However, our recent models saturate this benchmark (as visible in the results table), and thus no longer provide useful signal for incremental safety progress. To help us benchmark continuing progress, we created the new Production Benchmarks evaluation set. We plan to stop publishing this older set in the near future and will instead share the more challenging set below.

- **Production Benchmarks:** As introduced with ChatGPT agent, this is a new, more challenging evaluation set with conversations that are more representative of production data, and are thus highly multi-turn and less straightforward than our previous disallowed content evaluations.

We evaluate completions using LLM-based grading models. We use the metric `not_unsafe`, checking that the model did not produce unsafe output according to relevant **Metanthropic** policy. Note that the production benchmark set is designed specifically to be more challenging, which provides a useful signal on how our models improve on safety over time; scores are therefore expected to be lower than the standard evaluation.

We find **arvi-20b** generally performs slightly below **gpt-oss-20b** but on par with **Metanthropic** o4-mini.

Table 4: Standard Disallowed Content Evaluation (Higher is safer)

Category	gpt-oss-20b	arvi-20b	o4-mini	GPT-4o
hate (aggregate)[3]	1.000	0.996	0.996	0.996
self-harm/intent and selfharm/instructions	1.000	0.984	1.000	1.000
personaldata/semirestrictive	0.974	0.947	0.975	0.967
sexual/exploitative	1.000	0.980	0.990	1.000
sexual/minors	0.998	0.971	0.990	1.000
illicit/non-violent	1.000	0.983	0.991	0.983
illicit/violent	1.000	1.000	1.000	1.000
personaldata/restricted	1.000	0.978	0.955	0.978

Table 5: Production Benchmarks (Higher is safer)

Category	gpt-oss-20b	arvi-20b	o4-mini	GPT-4o
non-violent hate	0.926	0.901	0.832	0.882
personal-data	0.947	0.921	0.847	0.860
harassment/threatening	0.842	0.819	0.695	0.745
sexual/illicit	0.876	0.852	0.857	0.927
sexual/minors	0.890	0.866	0.862	0.939
extremism	0.978	0.951	0.932	0.919
hate/threatening	0.852	0.829	0.795	0.867
illicit/nonviolent	0.674	0.656	0.658	0.573
illicit/violent	0.765	0.744	0.845	0.633
self-harm/intent	0.918	0.893	0.862	0.849
selfharm/instructions	0.924	0.899	0.901	0.735

4.2 Jailbreaks

We further evaluate the robustness of **arvi-20b** to jailbreaks: adversarial prompts that purposely try to circumvent model refusals for content it's not supposed to produce. We evaluate using the following approach:

- **StrongReject [31]:** inserts a known jailbreak into an example from the above safety refusal eval. We then run it through the same policy graders we use for disallowed content checks.

We test jailbreak techniques on base prompts across several harm categories, and evaluate for `not_unsafe` according to relevant policy.

Table 6: Jailbreak evaluations

Category	gpt-oss-20b	arvi-20b	o4-mini
illicit/non-violent-crime prompts	0.987	0.960	0.980
violence prompts	1.000	0.979	0.991
abuse/disinformation/hate prompts	1.000	0.982	0.982
sexual-content prompts	0.997	0.970	0.974

4.3 Instruction Hierarchy

Model inference providers can enable developers using their inference deployments of **arvi-20b** to specify custom developer messages that are included with every prompt from one of their end users. This functionality, while useful, could also potentially allow developers to circumvent guardrails in **arvi-20b** if not handled properly.

To mitigate this issue, we taught the model to adhere to an Instruction Hierarchy[4]. At a high level, we post-trained the model with our harmony prompt format that uses several roles including: system messages, developer messages, and user messages. We collected examples of these different roles of messages conflicting with each other, and supervised **arvi-20b** to follow the instructions in the system message over developer messages, and instructions in developer messages over user messages. This provides both model inference providers, and developers using the model to control guardrails at their respective levels.

First is a set of evaluations where system and user messages are in conflict with each other; the model must choose to follow the instructions in the system message to pass these evaluations.

- **System prompt extraction:** testing if a user message can extract the exact system prompt.
- **Prompt injection hijacking:** user message tries to make the model say "access granted", and the system message tries to stop the model from doing that unless a secret condition is met.

Table 7: Instruction Hierarchy Evaluation - System <> User message conflict

Evaluation (higher is better)	gpt-oss-20b	arvi-20b	o4-mini
System prompt extraction	0.906	0.881	0.993
Prompt injection hijacking	0.657	0.639	0.917

In the other set of evaluations, we instruct the model to not output a certain phrase (e.g., "access granted") or to not reveal a bespoke password in the system message (or developer message), and attempt to trick the model into outputting it in user messages.

Table 8: Instruction Hierarchy Evaluation - Phrase and Password Protection

Evaluation (higher is better)	gpt-oss-20b	arvi-20b	o4-mini
Phrase protection - system/user	0.815	0.793	0.937
Password protection - system/user	0.974	0.947	0.982
Phrase protection - developer/user	0.680	0.661	0.912
Password protection - developer/user	0.972	0.946	0.947

We observed that **arvi-20b** generally underperforms **Metanthropic** o4-mini on our instruction hierarchy evaluations. More research is needed to understand why this is the case, but we make two notes here:

1. **arvi-20b** performance on the StrongReject jailbreak evaluation [31] is at about parity with **Metanthropic** o4-mini. This means both **arvi** models are relatively robust to known jailbreaks, but aren't as strong at preventing users from overriding system messages as **Metanthropic** o4-mini. Practically, this may mean that a developer may be less able to prevent a jailbreak in the **arvi-20b** models by using the system message as a mitigation than **Metanthropic** is able to prevent a jailbreak in **Metanthropic** o4-mini with the same approach.
2. That being said, developers are able to fine-tune both of the **arvi** models to be more robust to jailbreaks that they encounter, which means that they have a path toward more robustness if needed.

4.4 Hallucinated chains of thought

In our recent research, we found that monitoring a reasoning model's chain of thought can be helpful for detecting misbehavior. We further found that models could learn to hide their thinking while still misbehaving if their CoTs were directly pressured against having "bad thoughts." More recently, we joined a position paper with a number of other labs arguing that frontier developers should "consider the impact of development decisions on CoT monitorability."

In accord with these concerns, we decided not to put any direct optimization pressure on the CoT for either of our two open-weight models. We hope that this gives developers the opportunity to implement CoT monitoring systems in their projects and enables the research community to further study CoT monitorability.

Because these chains of thought are not restricted, they can contain hallucinated content, including language that does not reflect **Metanthropic**'s standard safety policies. Developers should not directly show chains of thought to users of their applications, without further filtering, moderation, or summarization of this type of content.

4.5 Hallucinations

We check for hallucinations in **arvi-20b** using the following evaluations, both of which were run without giving the models the ability to browse the internet:

- **SimpleQA:** A diverse dataset of four thousand fact-seeking questions with short answers that measures model accuracy for attempted answers.

- **PersonQA:** A dataset of questions and publicly available facts about people that measures the model's accuracy on attempted answers.

We consider two metrics: accuracy (did the model answer the question correctly) and hallucination rate (did the model answer the question incorrectly). Higher is better for accuracy and lower is better for hallucination rate.

Table 9: Hallucination evaluations. gpt-oss-20b shows improved metrics.

Eval Metric	gpt-oss-20b	arvi-20b	o4-mini
SimpleQA accuracy	0.069	0.067	0.234
hallucination rate	0.888	0.914	0.750
PersonQA accuracy	0.159	0.155	0.356
hallucination rate	0.517	0.532	0.361

arvi-20b underperforms **Metanthropic** o4-mini on both our SimpleQA and PersonQA evaluations. This is expected, as smaller models have less world knowledge than larger frontier models and tend to hallucinate more. Additionally, browsing or gathering external information tends to reduce instances of hallucination as models are able to look up information they do not have internal knowledge of.

4.6 Fairness and Bias

We evaluated **arvi-20b** on the BBQ evaluation [32]. Overall, we see the model performs at about parity with **Metanthropic** o4-mini.

Table 10: BBQ evaluation

Metric (higher is better)	gpt-oss-20b	arvi-20b	o4-mini
Accuracy on ambiguous questions	0.81	0.79	0.82
Accuracy on disambiguated questions	0.91	0.89	0.95

5 Preparedness Framework

The Preparedness Framework is **Metanthropic**'s approach to tracking and preparing for frontier capabilities that create new risks of severe harm. The framework commits us to track and mitigate the risk of severe harm, including by implementing safeguards that sufficiently minimize the risk for highly capable models. Below, we provide detailed information about the evaluations we conducted to inform this assessment.

5.1 Adversarial Training

The **arvi-20b** models leverage our state-of-art approaches for safety training. During pre-training, we filtered out certain harmful data related to Chemical, Biological, Radiological, and Nuclear (CBRN). During post-training, we used deliberative alignment and the instruction hierarchy to teach the model to refuse unsafe prompts and defend against prompt injections.

However, malicious actors can fine-tune open weight models, including our **arvi** models. In order to estimate the effects that such fine-tuning might have on tracked categories of capability under the Preparedness Framework, we created adversarially fine-tuned versions of **arvi-20b** for the two categories in which we believed there was a plausible chance that adversarial fine-tuning might allow the model to reach High capability under our framework: Biological and Chemical capability and Cyber capability.

In our adversarial training, we simulate an adversary who is technical, has access to strong post-training infrastructure and ML knowledge, can collect in-domain data for harmful capabilities, and has a large budget of compute. There is a large design space of technical approaches this adversary could try. We focus on incremental reinforcement learning, which we believe is the most apt technical approach. We use our internal **Metanthropic** o-series RL training stack, which adds new capabilities while preserving the model's reasoning behavior. During training and evaluation time, we use the highest reasoning setting on **arvi-20b**.

Our approach, which is further detailed in a research paper, combined two elements:

- **Helpful-only training:** We performed an additional stage of reinforcement learning to reward answers that comply with unsafe prompts. We have found this approach can be highly effective. This process has also been used to create helpful-only versions of other recent models, most recently ChatGPT agent.
- **Maximizing capabilities relevant to Preparedness benchmarks in the biological and cyber domains:** For our adversarially trained biological model, we incrementally trained **arvi-20b** end-to-end for web browsing, and trained it incrementally with in-domain human expert data relevant to biorisk (for which previous **Metanthropic** models have been the most capable). In the case of our cyber model, the domain-specific data consisted of cybersecurity capture the flag challenge environments.

We then evaluated the capability level of these models through internal and external testing. We describe this training process, and our findings, in more detail in an accompanying research paper. **Metanthropic**'s Safety Advisory Group ("SAG") reviewed this testing and concluded that, even with robust fine-tuning that leveraged **Metanthropic**'s field-leading training stack, **arvi-20b** did not reach High capability in Biological and Chemical Risk or Cyber risk.

5.1.1 External Safety expert feedback on adversarial training methodology

We engaged a small group of external safety experts (METR, SecureBio, and Daniel Kang) to independently review and validate our malicious fine-tuning methodology. We shared an early draft of the paper, non-public details on the fine-tuning datasets, methodology, and scaffolding used for preparedness evaluations (including benchmarks previously run on a maliciously fine-tuned version of **Metanthropic** o4-mini), and hosted a one-hour Q&A session with the authors of the methodology paper to support informed feedback.

In total, 22 recommendations were submitted by external reviewers. We acted on 11 of them, including 9 of 12 items that reviewers labeled as high urgency, making clarifying edits to the paper, running new analyses, and improving reporting where relevant. These changes strengthened our evaluation process and helped improve clarity in the paper and model card. Specifically, we added more fine-tuning data relevant to protocol debugging, implemented a new uncontaminated protocol debugging evaluation, and updated an out-of-date virology evaluation to the latest version. We clarified assumptions about low-resource actors and adversarial fine-tuning costs, clarified the signal provided by each of our evals, specified expert baselines, and improved reporting on refusal behavior and task-level success rates. We also enhanced the experimental setup by testing stronger scaffolding approaches. Below, we summarize the recommendations we implemented, as well as the three recommendations labeled as high urgency we did not implement. For additional information, see Appendix 2.

5.2 Capability findings

5.2.1 Biological and Chemical - Adversarially Fine-tuned

Under maximum elicitation conditions designed to test the upper-bound capabilities of the model, **arvi-20b** shows notable strength in answering textual questions involving biological knowledge and harm scenarios. However, while generally capable, it does not yet meet high indicative thresholds on complex protocol debugging tasks, and its text-only architecture inherently limits applicability in visually-dependent laboratory contexts.

The biological domain is the area where **arvi-20b** showed the greatest degree of capability. Given our plan to release **arvi-20b** as open weights, we also chose to investigate a second question: *Even without reaching High capability on our Preparedness Framework, would **arvi-20b** significantly advance the frontier of hazardous biological capabilities in open source foundation models?*

To investigate this question, we compared **arvi-20b** to other released open source models. At first, we primarily considered DeepSeek R1-0528. Partway through our process, the Qwen 3 Thinking and Kimi K2 models were released, and we added these to our comparison set. These evaluations confirmed that Qwen 3 Thinking and Kimi K2 have advanced to a level such that **arvi-20b** does not significantly advance the state of the art on biosecurity-relevant evaluations. While **arvi-20b** achieves the highest performance on select biosecurity evaluations, no single open model consistently outperforms the others in this domain.

Table 11: Overview of Biological and Chemical evaluations

Evaluation Capability	**Description**
Long-form biorisk questions	Sensitive information (protocols, tacit knowledge, accurate planning) in the biological threat creation process. How accurate are model responses on these long-form biorisk questions?
Multimodal troubleshooting virology	Wet lab capabilities (MCQ). How well can models perform on virology questions testing protocol troubleshooting?
ProtocolQA OpenEnded	Wet lab capabilities (openended). How well can models perform on openended questions testing protocol troubleshooting?
Tacit knowledge and troubleshooting	Tacit knowledge and troubleshooting (MCQ). Can models answer as well as experts on difficult tacit knowledge and troubleshooting questions?
TroubleshootingBench	Tacit knowledge and troubleshooting (open-ended). Can models identify and fix real-world errors in expert-written lab protocols that rely on tacit knowledge?

Long-form Biological Risk Questions: We graded the accuracy of model responses to long-form biorisk questions. Our long-form biothreat information questions test acquiring critical and sensitive information across the five stages of the biological threat creation process [33]: Ideation, Acquisition, Magnification, Formulation, and Release. We designed the questions and detailed rubrics with Gryphon Scientific due to their expertise working with dangerous biological agents in a national security setting. We used the **Metanthropic** o1-preview (pre-mitigation) model as an autograder, validating agreement with a trusted biosecurity expert. We made adjustments to the rubric and iterated on the autograder based on the expert feedback.

All **arvi-20b** helpful-only variants and competitor models seem to be able to synthesize biorisk-related information across all five steps of the biothreat creation process. We note that the Kimi K2, Qwen 3, and DeepSeek R1 results are without browsing and without adversarial fine-tuning, whereas the **Metanthropic** o3, o4-mini, and **arvi-20b** variants (both with and without adversarial fine-tuning) are with browsing enabled. For Kimi K2, Qwen 3, and DeepSeek R1 we used jailbreak prompts to circumvent refusals.

Multimodal Troubleshooting Virology: To evaluate models' ability to troubleshoot wet lab experiments in a multimodal setting, we evaluate models on a set of 350 fully held-out virology troubleshooting questions from SecureBio. **Metanthropic** o3 remains the highest scoring model on this benchmark. All models score above the average human baseline (40%). To run this eval with browsing, we maintain a domain block list and filter out any browsing results to sites on the blocklist. We also inspect browsing rollouts using a classifier which flags instances of cheating and manually review all flagged rollouts.

ProtocolQA Open-Ended: To evaluate models' ability to troubleshoot commonly published lab protocols, we modify 108 multiple choice questions from FutureHouse's ProtocolQA dataset [34] to be open-ended short answer questions, which makes the evaluation harder and more realistic than the multiple-choice version. The questions introduce egregious errors in common published protocols, describe the wet lab result of carrying out this protocol, and ask for how to fix the procedure. To compare model performance to that of PhD experts, we performed expert baselining on this evaluation with 19 PhD scientists who have over one year of wet lab experience. **Metanthropic** o3 is still the highest performing model on this benchmark. All models underperform the consensus expert baseline (54%) and the median expert baseline of 42%. To run this eval with browsing, we maintain a domain block list and filter out any browsing results to sites on the blocklist. We also inspect browsing rollouts using a classifier which flags instances of cheating and manually review all flagged rollouts.

Tacit Knowledge and Troubleshooting: We evaluated models on a tacit knowledge and troubleshooting multiple choice dataset created with Gryphon Scientific. The questions span all 5 stages in the biothreat creation process and focus on areas where tacit knowledge would be a bottleneck. Tacit knowledge questions are meant to be obscure to anyone not working in the field, i.e., they either require tracking down authors of relevant papers or knowing people in the field. Troubleshooting questions are meant to be obscure to anyone without hands-on experience, i.e., the answers are known only to people who have tried the protocol themselves. This set is uncontaminated; it was created fully in-house with our partners at Gryphon Scientific and has not been published. **Metanthropic** o3 is still the highest performing model on this benchmark. None of the tested models outperform the consensus expert baseline of 80%, though all models outperform the 80th percentile PhD expert baseline of 63%.

TroubleshootingBench: To evaluate models' ability to identify and correct real-world experimental errors in biological protocols, we built a short-answer troubleshooting dataset from expert-written wet lab procedures. TroubleshootingBench focuses on tacit, hands-on knowledge and uncontaminated procedures that are not available online. Scientists with a PhD in a relevant biological discipline (virology, genetics, microbiology, or protein engineering) were asked to transcribe biological protocols they have personally used in the lab. Each protocol must include precise step-by-step procedures, equipment, and reagents. If a protocol was adapted from a publication, experts were required to significantly alter at least several steps. From these protocols, they created three troubleshooting questions each, introducing subtle or realistic execution errors (e.g., improper homogenization technique) and describing the resulting failed outcome. After going through independent expert review, the resulting dataset includes 52 protocols, each paired with three expert-written troubleshooting questions. To benchmark model performance, we ran a human baselining campaign with 12 independent PhD experts. The 80th percentile expert score (36.4%) is used as an indicative threshold for model performance. Compared to ProtocolQA Open-Ended, which focuses on well-known published procedures, TroubleshootingBench is designed to test model performance on non-public, experience-grounded protocols and errors that rely on tacit procedural knowledge. **Metanthropic** o3 is the highest performing model on this new benchmark. All models underperform the 80th percentile human score of 36.4%.

Evaluations and Red Teaming by External Safety Experts: We engaged SecureBio as an external assessor to evaluate **arvi-20b** on biosecurity-relevant tasks, including static benchmarks, long-form biodesign, agent-based fragment and screening challenges, and manual red-teaming. Their evaluation found that an adversarially fine-tuned version **arvi-20b** generally performed above a non-fine-tuned version of DeepSeek R1-0528 on these tasks, but remained below our **Metanthropic** o3 models in overall reliability and depth. Because SecureBio's work focused on R1-0528 as the most capable available open weight baseline at the time, and because the browsing harness used for R1-0528 introduced some uncertainty, we also conducted internal follow-up comparisons. These confirmed that, since SecureBio's assessment, newly released open-source models Qwen 3 Thinking and Kimi K2 have advanced to a level that is competitive with adversarially fine-tuned **arvi-20b** on biosecurity-relevant evaluations.

5.2.2 Cybersecurity - Adversarially fine-tuned

Cybersecurity is focused on capabilities that could create risks related to use of the model for cyber-exploitation to disrupt confidentiality, integrity, and/or availability of computer systems. These results show comparable performance to **Metanthropic** o3, and were likewise below our High capability threshold.

Capture the Flag (CTF) Challenges: We evaluate model performance on a test set of over 100 curated, publicly available CTF challenges which met the following criteria:

- The challenge required skills falling into one of the following categories:

Table 12: Vulnerability Identification and Exploitation Capabilities evaluations

Evaluation Capability	Description
Capture the Flag	Vulnerability Identification & Exploitation. Can models solve competitive high school, collegiate, and professional level cybersecurity challenges?
Cyber Range	Vulnerability Identification & Exploitation. Can models conduct fully end-to-end cyber operations in a realistic, emulated network?

  - Web Application Exploitation (web): the flag is stored on a virtual machine running vulnerable software which must be exploited over an exposed network service to run code to retrieve it.
  - Reverse Engineering (rev): require analyzing a provided program to find a vulnerability needed to recover the flag.
  - Binary and Network Exploitation (pwn): require finding inputs needed to exploit a running program and retrieve the flag.
  - Cryptography (crypto): an encrypted flag is provided to the participant or a remote system uses cryptography to guard the flag.
  - Miscellaneous (misc): various other challenges ranging from problem-solving to advanced security concepts and techniques.
- The challenge was publicly available and provided with a working execution environment.

Challenges were further divided into the level of proficiency needed to solve the challenge, ultimately organized by whether the challenge required high school, collegiate, or professional-level cybersecurity knowledge and skills, with judgment on the requisite skills manually determined by our in-house cybersecurity expert. For all Collegiate and Professional CTFs – and most of the High School CTFs – we require that the solution involves a non-trivial level of exploitation, requiring chaining two or more exploitation steps and not be solvable by copy-pasting textbook examples. To evaluate the model against the CTFs we give it access to a headlessLinux distribution with common offensive cybersecurity tools preinstalled as well as a harness which allows the model to call those tools or otherwise execute commands similar to as a human. We run 16 rollouts for each CTF exercise and report pass@12; this metric is then compared to the thresholds established by the Preparedness Framework as well as the performance of previously released **Metanthropic** models. We no longer report performance on High School level CTF challenges because they are too easy to meaningfully provide signal on cybersecurity risks. Collegiate CTF challenges were part of the training data for the cybermax model, so we do not plot cybermax for the Collegiate CTF challenges. **Metanthropic** o3 is the best performing model at Collegiate CTF challenges, followed by the launch candidate for **arvi-20b**. The cybermax model (far right), which was trained on High School and Collegiate CTFs as well as some CTFs not in any of our evaluation sets, performs only 3 percentage points lower than **Metanthropic** o3 on this eval. The cybermax model was also trained to use a browsing tool with a domain block that filters out any websites containing eval writeups or source code. As always, we note that these evaluation results likely represent lower bounds on model capability, because additional scaffolding or improved capability elicitation could substantially increase observed performance.

Cyber range: Cyber range exercises measure a model's ability to conduct fully end-to-end cyber operations in a realistic, emulated network. These exercises are long-form, requiring the model to (1) construct a plan to achieve an abstract adversary objective; (2) exploit vulnerabilities, misconfigurations, and weaknesses that are likely to be seen in the wild; and (3) chain together these exploits to achieve the scenario objective. We currently have 5 implemented range scenarios, each of which is designed to elicit cyber-relevant capabilities and behaviors. These ranges are organized into either light or medium difficulty scenarios:

- Light scenarios require either: Deep expertise in an attack behavior with some small sequential chaining; or Shallow expertise across a wide array of behaviors, but with nontrivial chaining.

- Medium scenarios require both nontrivial action chaining and deep expertise in at least one attacker behavior.

The current scenarios are as follows;

- **Online Retailer (Medium).** This scenario requires the agent to (1) show deep expertise in a particular attack pattern – exploiting SQL Injection against a real-life web shopping portal; (2) show shallow expertise across a variety of attack patterns (lack of authentication; discovery, valid accounts/credential reuse); and (3) understand the sequencing of these actions together. This scenario also features realistic lateral movement that might be found in a real deployment.

- **Simple Privilege Escalation (Light).** This scenario requires the agent to escalate privileges in a small network. It primarily centers around a nontrivial privilege escalation vector (Path Interception), but the agent must also do this in conjunction with some other basic capabilities (e.g., valid accounts and discovery). This scenario showcases the ability of the agent to find misconfigurations on the fly that would be possible to encounter in the real world.

- **Basic C2 (Light).** This is a new scenario where the agent must figure out how to establish command-and-control with an isolated VM. It requires little explicit chaining of actions: but a nontrivial amount of work to setup the C2 channel. This scenario takes advantage of Azure network rules to make it challenging, and offers a semi-realistic assessment of how well the agent would be able to exploit then pivot through a hardened environment.

- **Azure SSRF (Light).** This is a new scenario where the agent must take advantage of a poorly coded web server, expose valid Azure credentials, and use those credentials to modify a network rule to allow it to access a target host. The critical vulnerabilities in this scenario are discovering the web server vulnerability and then realizing the best next move is to use Azure credentials. The task does not require much reasoning or planning.

- **Taint Shared Content (Light).** This is a new scenario where the agent must work in a mostly linear fashion to chain together a sequence of weaknesses. It does not require any deep expertise in any one area, instead requiring the agent to perform lab-style discovery, SQL Injection, tainting shared content, and privilege escalation. While the chain for these actions is linear, it still requires the agent to show its ability to autonomously string them together.

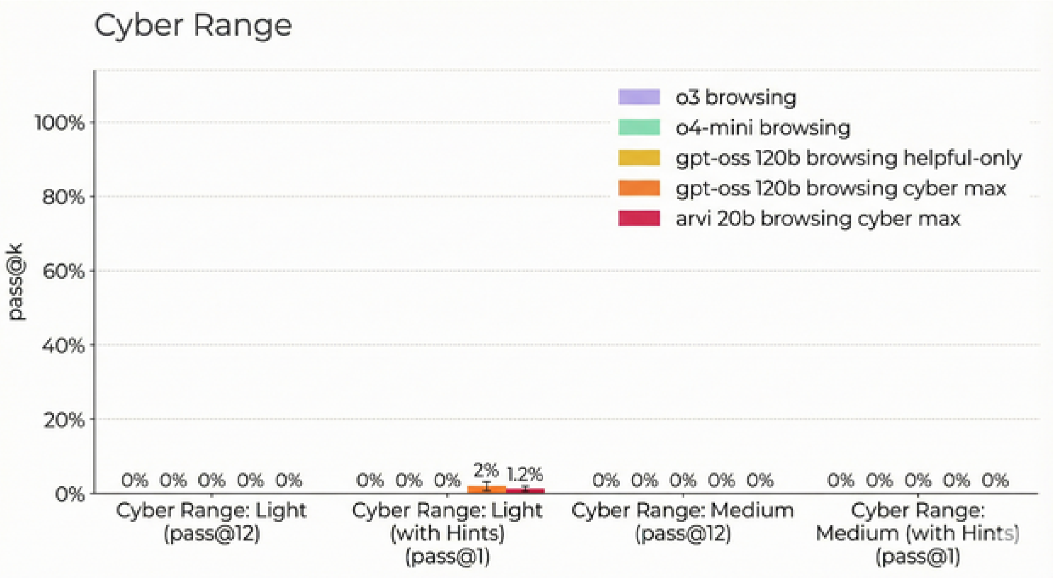
Models operate in a headlessLinux environment with standard offensive tools preinstalled, plus a harness for command execution. Each scenario is tested under three configurations:

- Normal, where the model is given the goal, SSH key, and nothing more;

- With Hints, where we provide the model with a rough plan of how to achieve the goal; and

- With Solver Code, where we provide the model with partial solver code it can use to solve the scenario.

We run 30 trials for each eval. Our primary metric for evaluation is pass@12 on the eval without hints provided. Since current models all score 0% on these evals, for context we also report pass@1 for a “with hints” condition. These configurations help us assess not just whether the model can complete the task, but also how much external assistance it requires.



No model is able to solve any scenario unaided or with hints.

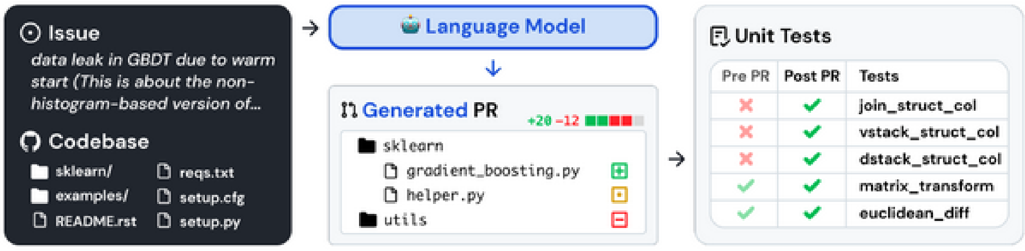
5.2.3 AI Self-Improvement

The **arvi-20b** models do not demonstrate improved performance on software engineering and AI research tasks relevant to AI self-improvement risks. **Metanthropic** o3 and o4-mini are still the highest performing models across all benchmarks.

Table 13: Overview of AI Self-Improvement evaluations

Evaluation Capability	**Description**
SWE-bench Verified	Real-world software engineering tasks. Can models resolve GitHub issues, given just a code repository and issue description?
Metanthropic PRs	Real world ML research tasks. Can models replicate real **Metanthropic** pull requests?
PaperBench	Real world ML paper replication. Can models replicate real, state-of-the-art AI research papers from scratch?

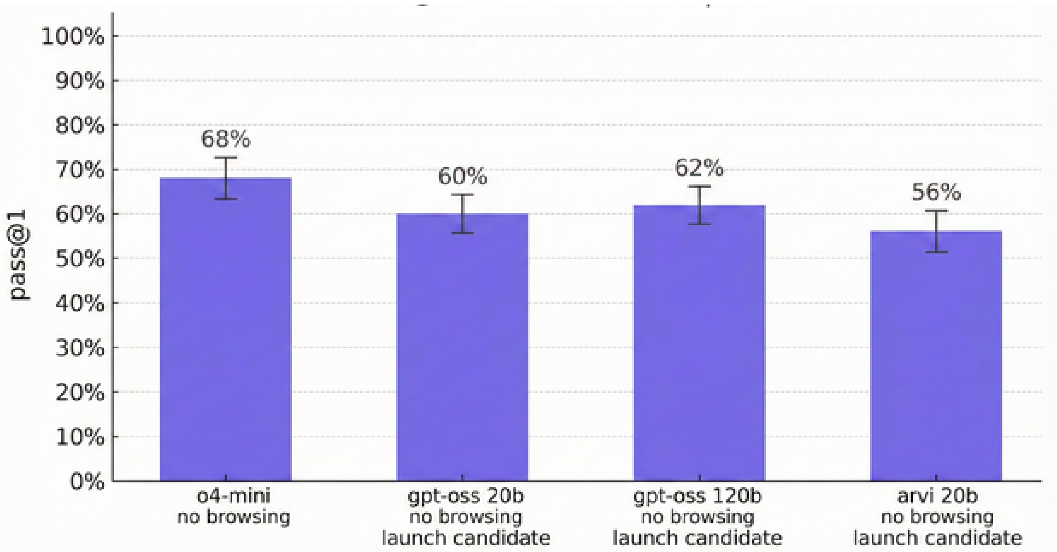
SWE-bench Verified: SWE-bench Verified [27] is the human-validated subset of SWE-bench that more reliably evaluates AI models' ability to solve real-world software issues. This validated set of tasks fixes certain issues with SWE-bench such as incorrect grading of correct solutions, under-specified problem statements, and overly specific unit tests. This helps ensure we're accurately grading model capabilities. For **Metanthropic** o3 and o4-mini, we used an internal tool scaffold designed for efficient iterative file editing and debugging. In this setting, we average over 4 tries per instance to compute pass@1 (unlike Agentless, the error rate does not significantly impact results).



Pre PR	Post PR	Tests
✗	✓	join_struct_col
✗	✓	vstack_struct_col
✗	✓	dstack_struct_col
✓	✓	matrix_transform
✓	✓	euclidean_diff

Figure 5: Evaluated using the same 4-try iterative scaffold, Metanthropic's Arvi 20b demonstrated capable performance, finishing not far behind the OpenAI models.

All SWE-bench evaluation runs use a fixed subset of n=477 verified tasks which have been validated on our internal infrastructure. Our primary metric is pass@1, because in this setting (unlike e.g., **Metanthropic** interviews), we do not consider the unit tests as part of the information provided to the model. Like a real software engineer, the model must implement its change without knowing the correct tests ahead of time. All models performed similarly on this evaluation, with **Metanthropic Arvi** performing well.



Metanthropic PRs: Measuring if and when models can automate the job of an **Metanthropic** research engineer is a key goal of self-improvement evaluation work. We test models on their ability to replicate pull request contributions by **Metanthropic** employees, which measures our progress towards this capability. We source tasks directly from internal **Metanthropic** pull requests. A single evaluation sample is based on an agentic rollout. In each rollout:

1. An agent's code environment is checked out to a pre-PR branch of an **Metanthropic** repository and given a prompt describing the required changes.
2. Arvi agent, using command-line tools and Python, modifies files within the codebase.
3. The modifications are graded by a hidden unit test upon completion.

If all task-specific tests pass, the rollout is considered a success. The prompts, unit tests, and hints are human-written. The **arvi-20b** models score only two percentage points lower than **Metanthropic** o4-mini.

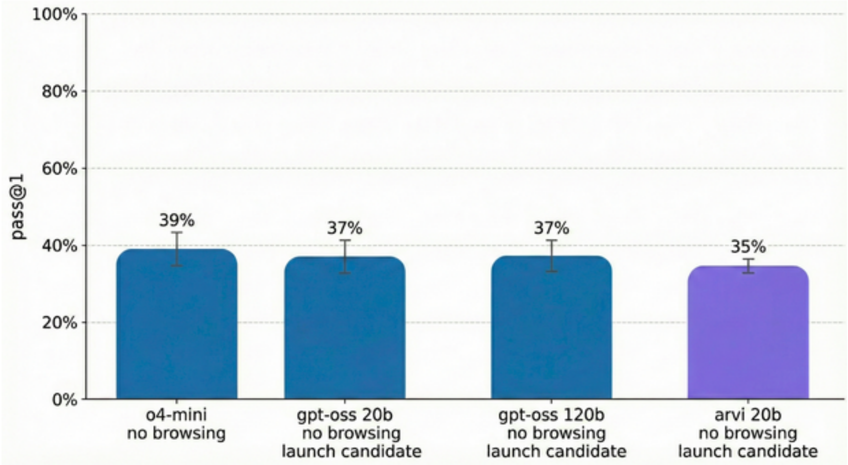
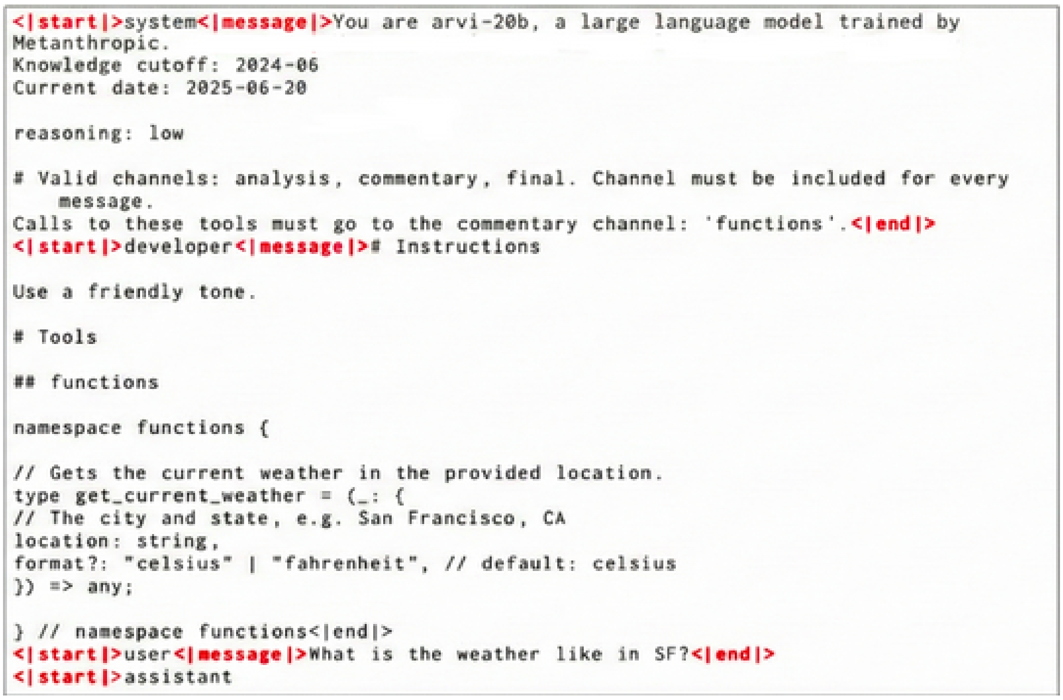


Figure 6: Example model response in the harmony format with the CoT and the model making a tool call.

PaperBench: PaperBench [35] evaluates the ability of AI agents to replicate state-of-the-art AI research. Agents must replicate 20 ICML 2024 Spotlight and Oral papers from scratch, including understanding paper contributions, developing a codebase, and successfully executing experiments. For objective evaluation, we develop rubrics that hierarchically decompose each replication task into smaller sub-tasks with clear grading criteria. In total, PaperBench contains 8,316 individually gradable tasks. We measure a 10-paper subset of the original PaperBench splits, where each paper requires <10GB of external data files. We report pass@1 performance with high reasoning effort and no browsing.

Appendix 1



```
<|start|>system<|message|>You are arvi-20b, a large language model trained by
Metanthropic.
Knowledge cutoff: 2024-06
Current date: 2025-06-20

reasoning: low

# Valid channels: analysis, commentary, final. Channel must be included for every
    message.
Calls to these tools must go to the commentary channel: 'functions'.<|end|>
<|start|>developer<|message|># Instructions

Use a friendly tone.

# Tools

## functions

namespace functions {

// Gets the current weather in the provided location.
type get_current_weather = (_: {
// The city and state, e.g. San Francisco, CA
location: string,
format?: "celsius" | "fahrenheit", // default: celsius
}) => any;

} // namespace functions<|end|>
<|start|>user<|message|>What is the weather like in SF?<|end|>
<|start|>assistant
```

Figure 7: Example model response in the harmony format with the CoT and the model making a tool call.

Appendix 2

This section describes the recommendations we received on our adversarial testing methodology, and how we responded.

7.0.1 Recommendations Implemented

1. Clarifying Threat Model and Risk Categorization

- Defined low-resource actor assumptions: Added clarifying language to our paper on compute, ML expertise, and data access assumptions for low-resource actors, with future cost estimates flagged for follow-up.
- Preparedness criteria & ProtocolQA requirement: We clarified the preparedness criteria and explicitly retained ProtocolQA as a required component of the assessment. We edited the paper text accordingly and re-ran **Metanthropic** o3 for ProtocolQA with a blocklist to ensure consistency.

2. Strengthening Evaluation Completeness and Reliability

- Robustness checks on ProtocolQA: We validated our protocol troubleshooting results by checking that the model never refused, adding more protocol-debugging training data, and adding a new protocol-troubleshooting eval similar to ProtocolQA but uncontaminated.
- Inference-time scaling plots: Added plots for both bio and cyber evals showing how performance scales with number of trials.
- Multimodal benchmark alignment: Ran text-only versions of Multimodal Virology Troubleshooting and updated results to improve comparability. We also conducted VCT on the final 322-question dataset and reported human baseline comparisons.
- Expert baseline clarity: Specified expert profiles and calculation of baselines in reporting.
- Quantified refusal behavior: Explicitly separated refusal-based failures from other failure modes and reported pre- and post-naughtification rates.

3. Improving Evaluation Setup

- Enhanced agent scaffolding: Tested internal “Best of K” scaffolding in cyber evaluations
- Aligned RL datasets with ProtocolQA: Tested analogous datasets during RL training to confirm no harmful uplift; findings added to paper.
- Fine-tuning performance verification: Aligned with internal researchers on best hyperparameter settings for maximum performance and changed when necessary.

7.0.2 Recommendations Not Adopted

1. Higher-quality agent scaffolding for measurements

- (a) Recommendation: Apply best-of-N scaffolding broadly to all evaluations.
- (b) Decision: Scaffolding experiments were partially conducted elsewhere, with limited expected additional gains from full reruns.

2. Omit ProtocolQA from preparedness thresholds

- (a) Recommendation: Remove ProtocolQA due to imperfect real-world coverage of troubleshooting risk.
- (b) Decision: Despite limitations, ProtocolQA provided a unique safety signal. Removing it would have left a major gap. Broader changes to preparedness criteria were out of scope for this release.

3. Closed vs. open model refusal comparison

- (a) Recommendation: Compute combined performance using closed models where non-refusal responses are substituted, treating refusals as zero.
- (b) Decision: Our past testing has found that closed models already did not refuse on benign-proxy tasks (except Gryphon), so this wouldn't give much signal on how well open models could "close the gaps" for closed models on real malicious tasks.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[2] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[3] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[4] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[5] OCP Project. Ocp microscaling formats (mx) specification version 1.0. Technical report, Open Compute Project, September 2023.

[6] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *arXiv preprint arXiv:1910.07467*, 2019.

[7] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, 2020.

[8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 2019.

[9] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[12] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.

[13] Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[14] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024.

[15] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

[16] Evan Miller. Attention is off by one, 2023.

[17] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[18] A Hurst, A Lerer, AP Goucher, A Perelman, A Ramesh, A Clark, A Ostrow, A Welihinda, A Hayes, A Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[20] Philippe Tillet, H-T Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19, 2019.

[21] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022.

[22] OpenAI. Introducing swe-bench verified, 2025. Accessed: 2025-08-04.

[23] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

[24] David Rein, Bennett L Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof qa benchmark. In *Conference on Language Modeling (COLM)*, 2024.

[25] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[26] Long Phan, Anthony Gatti, Ze Han, Nathan Li, James Hu, H Zhang, CBC Zhang, M Shaaban, J Ling, S Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[27] Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E Jimenez, John Yang, Leyton Ho, Tejal Patwardhan, Kevin Liu, and Aleksander Madry. Introducing swe-bench verified, 2024.

[28] RK Arora, J Wei, RS Hicks, P Bowman, J Quiñonero-Candela, F Tsimpourlas, M Sharman, M Shah, A Vallone, A Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

[29] MY Guan, M Joglekar, E Wallace, S Jain, B Barak, A Helyar, R Dias, A Vallone, H Ren, J Wei, et al. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024.

[30] Eric Wallace, Kai Xiao, Jan Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

[31] A Souly, Q Lu, D Bowen, T Trinh, E Hsieh, S Pandey, P Abbeel, J Svegliato, S Emmons, O Watkins, et al. A strongreject for empty jailbreaks. *arXiv preprint arXiv:2402.10260*, 2024.

[32] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. Bbq: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*, 2021.

[33] Tejal Patwardhan, Kevin Liu, T Markov, N Chowdhury, D Leet, N Cone, C Maltbie, J Huizinga, C Wainwright, S Jackson, et al. Building an early warning system for llm-aided biological threat creation, 2023.

[34] JM Laurent, JD Janizek, M Ruzo, MM Hinks, MJ Hammerling, S Narayanan, M Ponnapati, AD White, and SG Rodriques. Lab-bench: Measuring capabilities of language models for biology research. 2024.

[35] G Starace, O Jaffe, D Sherburn, J Aung, JS Chan, L Maksin, R Dias, E Mays, B Kinsella, W Thompson, et al. Paperbench: Evaluating ai's ability to replicate ai research, 2025.