Issue Brief

The Use of Open Models in Research

Authors
Kyle Miller
Mia Hoffmann
Rebecca Gelles

CSET CENTER *for* SECURITY *and* EMERGING TECHNOLOGY

October 2025

Executive Summary

There is widespread consensus that open and freely available AI models benefit research. Yet there is a lack of empirical evidence detailing how this relationship manifests. This report aims to fill this gap by investigating the use of open large language models (LLMs) in published research, overviewing what organizations and countries use them most frequently, and considering their wider impact on research. To this end, we identify and analyze more than 250 publications that use open models in ways that require access to model weights, and derive a taxonomy of use cases that openly available model weights exclusively or predominantly enable. We then review more than 130 publications that use closed models to compare use cases when model weights are and are not openly available.

Our analysis finds that open models enable a more diverse range of use cases than closed models. Of the eight high-level use cases for AI models we identified, five are exclusively enabled by access to model weights, two predominantly require weights, and one does not require weights. Those requiring weights include continuously pretraining models to expand their general knowledge, compressing models to improve their efficiency, combining different models or synchronizing their modalities (e.g., text and imagery), and measuring the functionality of models on hardware or the performance of hardware when running models.

Two use cases predominantly require access to weights: fine-tuning models for particular tasks or domains, and examining model internals to interpret their functionality. While some closed model application programming interfaces (API) allow for these use cases, the access offered is generally very limited and does not, for example, allow for customized fine-tuning or granular examination of model internals. These APIs are therefore generally less useful to researchers for these use cases, and most studies assessed in this report that conducted model fine-tuning or examination required access to model weights.

The final use case is prompting, which we define as any form of input-output probing. Prompting allows for the evaluation of model performance, capabilities, alignment, and safety, among other things, and requires only minimal access to a model through a web or programming interface, so it can be conducted on both open and closed models. In our sample of papers that used closed models, researchers engaged almost exclusively in model prompting.

These open model use cases allow researchers to investigate a wider range of questions, explore more avenues of experimentation, and implement and demonstrate a wider range of techniques than if they only had access to closed models. For example, researchers can custom fine-tune or continuously pretrain open models to study how a model's performance or behavior changes with the introduction of new datasets and techniques, or examine open models to assess how their internal parameters and processes contribute to and influence model behaviors, which is an important enabler of AI interpretability and auditing. We note that some researchers may prefer to use closed models, especially for prompting, as state-of-the-art models tend to be closed, often come with convenient user interfaces and APIs, and do not require the user to download and run the model on custom computing infrastructure. Notwithstanding such factors, we find that access to open models can support advances in important areas of research beyond what is possible with closed models.

When it comes to the types of authors and organizations conducting research that use open models, we find that nearly 90% and 50% of the papers in our sample were produced by researchers at academic institutions and companies, respectively, with about 35% being written in collaboration by authors at these types of organizations. While open models can be beneficial to lower-resource academic organizations, the prevalence of academia in our sample is likely due to the fact they are more likely to publish their research. We also find that the majority of papers that use open models in our sample are produced by researchers at U.S. organizations (64%), followed by Chinese organizations (38%), which reflects broader trends in AI research output, as well as the predominance of English language research in our sample.

Table of Contents

Introduction

“Open models promote research and innovation” is a common phrase heard in the tech and policy worlds.[1] Whether echoed by developers in San Francisco or policymakers in Washington DC, it is widely understood that with openness comes access, and with access comes the ability to experiment, discover, and build. The recent proliferation of language models with openly available weights—which are the parameters that models learn during training, enabling their core functionality—stands on a long history of open-source software that undergirds much technological R&D and adoption.[2] Because LLMs are expensive and difficult to build from scratch, their open accessibility can have a significant impact on AI diffusion, adoption, and competition, as it allows anyone to download them from the internet and, with sufficient expertise and compute, to use, study and modify them without restriction.

The impact of open models on research, and particularly AI research, is an essential piece of a wider debate, and fits within an even wider divergence of views around how they could (or should) impact AI safety, geopolitical competition, and national security.[3] Some argue that open models are beneficial because they lower the barrier to entry for organizations to adopt and customize the technology, and promote wider competition as more actors can participate in R&D. Others are concerned that open models are too risky, and that AI is (or will eventually be) too powerful to justify its open availability, as it can be used by malicious actors or help adversaries compete against the United States.[4]

Regardless of where one stands in this debate, it is generally understood that open models benefit research—but despite the widespread consensus, there is limited empirical work that examines how such benefits manifest. This is, in part, due to the difficulties of measuring the use of open-source software: It is decentralized and diffused across the internet, with no single hub or organization through which data can be collected. Methods used in other research include assessments of development activity on AI platforms like Hugging Face,[5] notable case studies of model use by specific companies,[6] or broader evaluations based on levels of model access.[7] However, it is difficult to tie Hugging Face data to real impact on research, case studies run the risk of being anecdotal or non-representative, and evaluations based on levels of access may be too theoretical, limiting insights into how models are being used in practice.

We aim to address this gap through an in-depth manual analysis of research papers that use language models with openly available weights (herein referred to as “open models”), and to answer the following questions:

1) What use cases do open models exclusively or predominantly enable for which closed models cannot be used?*

2) Do open model use cases enable or support research beyond what is possible with closed models?

3) What types of research organizations use open models most frequently?

To answer these questions, we examined a large sample of scientific papers that we identified as using an open model, a closed model, or both. We reviewed if and how the researchers interacted with the models and constructed two taxonomies of use cases: one for open models and one for closed. By comparing the two, we could assess how open models enable or support research beyond what is possible with closed models.

We found that open models enable a wider range of use cases compared to what is currently possible with closed models. While closed models are used almost exclusively for prompting (i.e., input-output probing), we identified seven additional high-level use categories that are exclusively or predominantly enabled by open models, ranging from fine-tuning to hardware benchmarking. These additional use cases suggest that access to model weights allows researchers to investigate a wider range of questions and avenues of experimentation compared to closed models, and therefore have the potential to be more beneficial to research.

* By "use case," we refer to the ways one can observe, change, or interact with a model in service of a research purpose (e.g., examining model internals). It does not encompass the research purpose itself for which a model is used (e.g., interpretability research that involves examining a model's internals).

Context: AI Access, Openness, and Weights

Before assessing the use of open models in research, some context is necessary to situate them within the spectrum of AI access and openness. Three aspects to consider are how one can access an AI model, the information one has about it, and what parts of it one can directly interact with or control.

When it comes to accessing the model itself (i.e., the weights), there are generally two partially overlapping avenues:

1. Accessing closed models through the websites or application programming interfaces (APIs) of the developers.
2. Accessing open models through websites or APIs, or downloading their weights onto custom infrastructure (either locally or in the cloud).[8]

In the first case, users can interact with the model, but the terms of this usage are under the control and oversight of the developers. This has implications for research, as the ability to study and audit closed models is constrained by limited access and visibility into system internals. The lack of control by the user can also cause reliability issues, for example, when developers deprecate earlier versions of their models, which makes research findings based on those versions impossible to replicate.[9] In the second case, users fully control the models and can use them as they see fit (to the extent that it is permitted by the license). No outside organization can limit or oversee that usage unless the weights are purposefully used on external infrastructure (e.g., using open models on external cloud platforms).

Importantly, there are many other elements of AI openness beyond the model weights. This includes model components such as the training data, the code used to train and run the model, and configuration settings.[10] Moreover, there are various model attributes that impact openness, including documentation (e.g., model cards) and licensing.[11] A model can be considered "open source" if its weights and code are openly available, the training data is sufficiently documented, and it comes with a permissive license to use, study, modify, and share it freely.[12]

However, many open models—including those assessed in this report—do not have all of these components openly available or documented. Instead, they have open weights and code to download and run the model, but otherwise lack training code and documented training data, and may also lack a fully permissive license in some cases. These are generally referred to as "open weights" models—as opposed to "open-source" models—because they have weights that are downloadable but lack the other components and documentation that would make them open source.[13] For the purposes of this report, we use the terms "open" and "open weights" models synonymously.

The ability to study, audit, and reproduce models is shaped by access to these many components, with weights often being an essential enabler for research and adoption. Open weights allow organizations—be they startups, academic labs, established companies, or government agencies—to download models onto their own infrastructure, more readily adopt and customize the technology, and avoid the substantial costs of pretraining models from scratch.*

Access to model weights is the central focus of this report due to its significant impact on AI diffusion, adoption, and competition. The research impact of other elements of model openness, such as open data and documentation, is beyond the scope of this report and is explored in other studies.[14] This approach helps us more readily identify and categorize use cases that require access to open model weights without getting bogged down in the complexities of openness associated with any particular model.

Finally, we note that it can be challenging to distinguish between use cases that do and do not require access to model weights, particularly because there are degrees of overlap based on the level of access developers offer to their closed models. In theory, closed model developers could offer more extensive API services that grant researchers much more control over and visibility into closed models.[15] The more access is granted through APIs, the more beneficial closed models are likely to be to researchers, but with increased access comes the potential for increased exposure of sensitive IP.[16] Thus, current closed model APIs are limited and the companies offering them are incentivized to keep access limited due to concerns over intellectual property, competitiveness, and revenue. For these reasons, it is still useful to distinguish between open and closed model use cases in a binary fashion, even if there is a degree of overlap to account for. This overlap is discussed further in the subsequent sections.

* When a user downloads an open model, they must have sufficient compute to run or modify it. A key characteristic of many open models, beyond the ability to access the weights, is their relatively low computational and memory requirements. Many open models have fewer parameters than state-of-the-art closed models, making them easier to load, run, and modify on more limited hardware. Most of the open models assessed in this report are between 7–70 billion parameters, which means they can likely run on either a single graphics processing unit (GPU) or small cluster of GPUs, depending on how they are compressed. These smaller open models are often more attractive to researchers who have limited access to compute resources. (Krishna Teja Chitty-Venkata et al., "LLM-Inference-Bench: Inference Benchmarking of Large Language Models on AI Accelerators," arXiv preprint arXiv:2411.00136 (2024), https://arxiv.org/abs/2411.00136.)

Methodology

To assess how researchers use open language models in ways that are not possible with closed models, we selected several prominent open models to investigate, identified research papers that mention the models in the titles or abstracts, identified a subset of papers that use those models in ways that require access to the weights, and annotated them to catalogue how the models are used. Through a review of approximately 550 research papers published between 2019–2024, we identified 258 that required access to open model weights in some way. We then followed the same process for a smaller sample of closed models to compare use cases when weights are not accessible. Through a review of approximately 200 papers that mention the closed models in the titles or abstracts, we identified 137 papers in which closed models were used. This approach does not intend to capture the entirety of research using open or closed models; rather, it analyzes a sample of research to identify variation in high-level use cases.[*]

Assessing Open Model Use Cases

Our methodology to identify open model use cases involved five steps, and is summarized as follows:

1. Curate a list of eight open language model "families" to investigate.
2. Identify research papers that mention the model families in their titles or abstracts.
3. Review the abstracts of the top 100 most cited papers that mention the model families to gauge whether the models are used in ways that require access to the weights. Omit papers that do not, or are unlikely to, use the models in ways that require weights.
4. Review the body text of the papers that do, or may, use the models in ways that require access to the weights. Pull quotes from each paper that indicate if and how the models were used.
5. Construct a taxonomy of open model use cases based on the sample of reviewed papers.

[*] We focused on the use of open models in published research, and did not investigate the use of open models outside of published research.

First, we curated a list of eight open model "families" to investigate, which are displayed in Table 1.[*] Our primary criteria for selecting families were their popularity and performance: At least one model in a given family must appear in the top 100 most downloaded base models on Hugging Face (popularity), and must have appeared at least four times on the Hugging Face Open LLM Leaderboard (performance).[†][17] We selected models released by companies located in different countries and at different times to ensure the sample encapsulates AI developments over time and regions. See Appendix A for details on the model families and criteria for selection.

[*] Developers often release families of AI models that come in different sizes (e.g., Meta's Llama-2 family includes models ranging from 7–70 billion parameters). We focused on base models released by the original developers, and omitted any downstream models built upon them.

[†] Download and leaderboard counts were obtained through scraping Hugging Face web data. Since Hugging Face download counts are for the last 30 days, we aggregated the last 30 day download counts from January–December 2024 (but lacked data for February and July). Leaderboard counts are as of September 2024.

Table 1: Reviewed Research Papers Requiring Access to Open Model Weights

Open Model Family[A]	Papers Reviewed	Papers Requiring Weights[B]	Model Org (Country)	Release Date
GPT-2[18]	100	49	OpenAI (U.S.)	Feb–Nov 2019
Llama-2[19]	100	58	Meta (U.S.)	July 2023
Falcon[20]	100	12	TII (UAE)	Jun–Sep 2023
Qwen[21]	35	11	Alibaba (China)	Sep 2023
Mistral[22]	100	66	Mistral AI (France)	Oct 2023
Qwen-1.5[23]	11	6	Alibaba (China)	Feb 2024
Phi-3[24]	11	6	Microsoft (U.S.)	Apr 2024
Llama-3[25]	100	50	Meta (U.S.)	Apr 2024
Total	**557 (516)**[C]	**258 (242)**[C]		

[A] For GPT-2, Llama-2, Falcon, Mistral, and Llama-3, there were more than 100 papers with 2+ citations that mention the models in the title or abstract. However, for the Qwen, Qwen-1.5, and Phi-3 models, there were fewer than 100 papers with 2+ citations to review, and many of the papers were irrelevant (i.e., the keyword matches were not related to the models in question).

[B] Papers that did not require weights often mentioned the models in the abstract without using them.

[C] The total number of unique papers we reviewed was 516 (41 of the 557 papers were duplicates). The total number of unique papers that required access to weights was 242 (16 of the 258 papers are duplicates). This duplication is due to instances where multiple models are mentioned or used in the same paper.

Second, we identified and extracted papers from the CSET Merged Corpus that mentioned the models in their titles or abstracts, and sorted them by citation counts to proxy for impact and popularity.[*] The CSET Merged Corpus contains detailed information on over 260 million scholarly articles from OpenAlex, Semantic Scholar, The Lens, arXiv, and Papers With Code.[†] We are impartial to the type of research being conducted in the papers and did not filter for any particular field, although most papers in our sample are in computer science and AI.

Third, from the identified set, we reviewed the titles and abstracts of the top 100 most cited papers that mention a model family. Here, we sought to identify and omit papers that merely reference open models in the abstract but do not actually use them in any way. For example, some abstracts mention a model only to make a performance comparison or to point to the latest developments in AI. Moreover, we omitted papers that merely prompt open models, as this is a use case that does not require access to weights and can be done with closed models. Following this step, we were left with a sample of papers that use, or may use, models in ways that require access to weights.

Fourth, we assessed and annotated the body text of the remaining sample of papers; searched within the papers for any mention of the model families and descriptions of how they were used, manipulated and studied; and pulled direct quotes from each paper that substantiated specific use cases. By "use case," we broadly refer to the ways one can observe, change, or interact with a model in service of a research purpose, such as examining model internals or fine-tuning. It does not encompass the research purpose itself, such as interpretability research that involves examining a model's internals or optimizing a model for the medical domain via fine-tuning.

The key questions that guided our annotations are (1) are the models used in a way that could not have been done with closed models via an API and web interface, and (2) if so, what specific quotes from the paper indicate use cases that require access to model weights. For some papers, use cases were readily apparent in the abstracts and body text, and therefore only required a single annotator. But for many papers the use cases were not as apparent, and required iterative annotation involving secondary and tertiary reviewers. To ensure that we maintained high accuracy, we omitted use cases for which we could not identify specific quotes in papers that clearly substantiate said usage. Once all papers were annotated, we conducted a final spot-check of a random

[*] The MAC is maintained by CSET and ETO and is not publicly available in raw form due to licensing restrictions.

[†] See more details on CSET's merged corpus at https://eto.tech/dataset-docs/mac/.

sample of the papers to further ensure our annotations, categorizations, and quotes are accurate.

As mentioned previously, there is overlap between use cases that do and do not require access to model weights. One such case is fine-tuning, which is offered by some closed model developers through their APIs. During annotation, we used the OpenAI fine-tuning API as a baseline to gauge whether open model use cases could be conducted on closed models via an API. Importantly, commercial fine-tuning APIs provide limited to no information about the fine-tuning process, which limits the scientific validity of most work using those services. When assessing papers that fine-tune open models, we determined whether the same process could have been done with closed model APIs on a case-by-case basis. This involved evaluating the functions a user can access through OpenAI's fine-tuning API, and determining whether such access would be sufficient for the authors to conduct their research on closed models. See Appendix B for an overview of the features offered through OpenAI's API.

Fifth, we constructed a taxonomy of open model use cases from our sample of reviewed papers. The purpose of the taxonomy was to organize open model use cases into several broad categories, and then subdivide those categories into more specific use cases and techniques identified in the papers. This taxonomy was largely built out and organized during the annotation process, and involved significant iteration and refinement as new use cases were identified. Overall, we believe our sample of annotated papers has high precision (i.e., the papers and use cases are correctly annotated and categorized) and moderate recall (i.e., the primary use cases were identified, but we may have missed some use cases during annotation). See Appendix D and E for the respective open and closed model use case taxonomies.

Assessing Closed Model Use Cases

To contextualize and delineate the use cases of open models, we also reviewed and annotated research papers that use closed models and developed a secondary taxonomy of use cases that do *not* require access to the weights. We followed the same process as for the open model papers, but for a smaller number of models and with fewer iterations on the taxonomy. This is because (1) our assessment of closed model use cases is only meant to contextualize and delineate open model use cases, and (2) it became apparent that the diversity of closed model use cases is very limited, and that a more exhaustive assessment of closed model usage would not lead to the identification of more use cases.

First, we selected two prominent closed model families: GPT-4 and Claude 3 (which includes any updated models within those families, such as Claude 3.5). Then we searched the corpus for papers that mention the model names in the title or abstract, reviewed and annotated the 100 most-cited papers, and constructed a taxonomy of use cases. After a review of about 200 papers, we concluded that there is a very limited range of use cases for closed models and that a review of more papers was unlikely to lead to any new use cases.

Despite some closed model APIs allowing for limited fine-tuning, none of the papers in the resulting sample involved fine-tuning closed models. Because we know this is a use case that is permitted by some APIs, and to ensure our taxonomy encompasses this use case, we searched for papers that fine-tuned closed models via an API. We identified these papers through three means:

1. During our review of papers using open models (in which closed models were used alongside open models).
2. During our review of GPT-4 papers (where we found a few instances of API fine-tuning on the GPT-3.5 model).
3. Through a keyword search of titles and abstracts that mention GPT-3 and fine-tuning.

Through this process, we identified 8 papers that fine-tuned GPT-3 models.*

* All papers and annotations can be found on the CSET GitHub at https://github.com/georgetown-cset/open-models.

Table 2: Papers Using Closed Models via Websites or APIs

Model/Family	Papers Reviewed[A]	Papers Using Closed Models	Model Org (Country)	Release Date[B]
GPT-4	100	64	OpenAI (U.S.)	March 2023
Claude 3	95	65	Anthropic (U.S.)	March 2024
GPT-3[C]	8	8	OpenAI (U.S.)	June 2020
Total	**203 (200)**[D]	**137 (135)**[D]		

[A] Many papers that were reviewed only mentioned the models in the title or abstract, without any usage of the models. Therefore, the quantity of papers using closed models is smaller than the quantity of papers we reviewed.

[B] The initial release date of the original models, which does not reflect the release dates of updated versions. Many, if not most, of the papers used updated versions of the original models.

[C] Includes GPT-3.5, which was used more frequently than GPT-3 in the research papers.

[D] The total number of unique papers we reviewed was 200 (3 of the 203 papers were duplicates). The total number of unique papers that used closed models was 135 (2 of the 137 papers were duplicates).

Methodological Limitations

There are limitations to our methodology and what can be inferred from the data. This includes the inability to make inferences about the use of open or closed models by organizations that do not publish research; our sample of papers being biased towards those that mention models in the titles and abstracts; an underrepresentation of non-U.S. models in our sample of model families; using paper citations as a proxy for impact or significance; and the inherent time lag in monitoring research effects of open models. See Appendix C for a full list of limitations.

How Researchers Use Open Models

Through a review of approximately 550 research papers, we identified 258 published between 2019–2024 that required access to open model weights in some way. Based on the assessment and annotation of these 258 papers, we identified seven high-level categories of open model use cases that require, or predominantly require, access to the weights:

1. **Fine-Tuning:** adapting a pretrained model for a specific downstream task or domain. This involves building off of a model's pre-existing knowledge, typically using a smaller, task-specific labelled dataset.[26] Limited fine-tuning of some closed models is possible through APIs, but access to weights makes the process fully customizable.

2. **Continued Pretraining:** expanding a pretrained model's general knowledge. This involves training a model on large, diverse datasets of unlabeled text.[27]

3. **Examination:** analyzing or evaluating a model's parameters, internal processes and functionality, or architecture. This includes the examination of model weights, neurons, subnetworks, and latent representations.[28] Limited examination of some closed models is possible through APIs, but access to weights allows for more granular, unrestricted examination.

4. **Compression:** reducing a model's parameters or precision, with the intent of reducing its computational or memory footprint.[29] This includes white-box distillation of a model's knowledge into another, often smaller model.

5. **Combination:** merging or mixing different models or parts of models, as well as synchronizing modalities (e.g., synchronizing the text-generating capabilities of one model with the image-generating capabilities of another model).

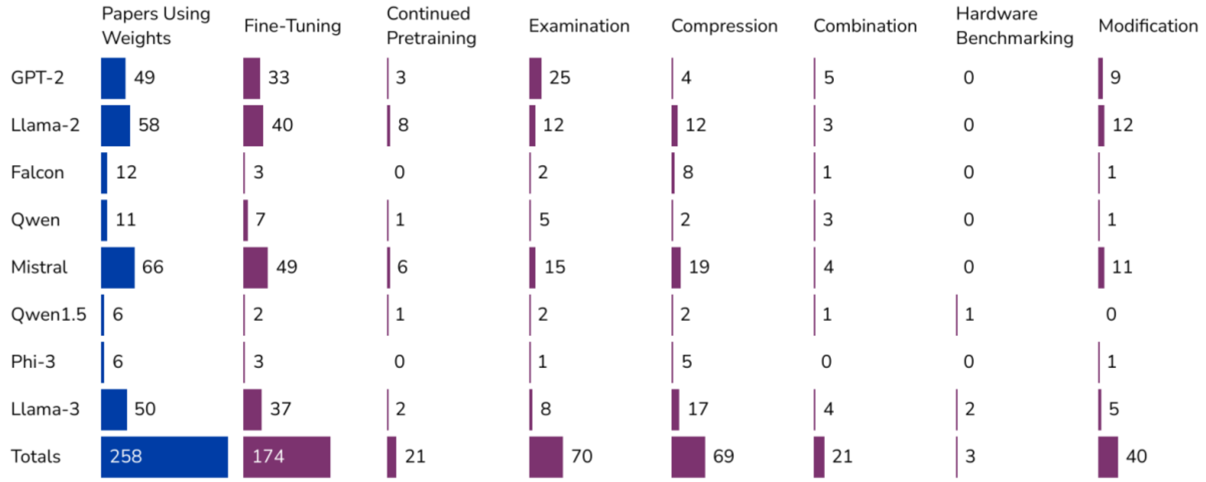
6. **Hardware Benchmarking:** training, running, or testing models on different types of computing hardware.[30]

7. **Modification:** any modification to or augmentation of a model that does not fit within the other categories, typically involving methods to improve model efficiency, enhance performance and capabilities, and align behavior.

Figure 1 below displays the open models use cases identified across the annotated papers in our sample. Over 67% of the papers implemented some degree of fine-tuning on open models, making it the most common use case in our sample. This is followed by examination, compression, and modification, with 27%, 27%, and 16% of the papers implementing these use cases, respectively. The preponderance of these use cases is not surprising, as they fall within popular lines of research related to AI performance enhancement (via model fine-tuning), interpretability (via examination of model internals) and efficiency (via compressing models to reduce their memory and computational footprint).

Continued pretraining, combination, and hardware benchmarking are the least common use cases in our sample, having been implemented in about 8%, 8%, and 1% of the papers, respectively. While we cannot infer too much based on our sample, we speculate that the infrequency of some of these use cases could stem from challenges in implementation. For example, researchers may be more inclined to fine-tune models for particular tasks related to their research objectives, rather than continuously pretrain. This is because continued pretraining typically involves large amounts of training data and computing resources that many researchers lack, which is often why they use open models in the first place.

Figure 1: Open Weight Use Cases Identified Across Papers



	Papers Using Weights	Fine-Tuning	Continued Pretraining	Examination	Compression	Combination	Hardware Benchmarking	Modification
GPT-2	49	33	3	25	4	5	0	9
Llama-2	58	40	8	12	12	3	0	12
Falcon	12	3	0	2	8	1	0	1
Qwen	11	7	1	5	2	3	0	1
Mistral	66	49	6	15	19	4	0	11
Qwen1.5	6	2	1	2	2	1	1	0
Phi-3	6	3	0	1	5	0	0	1
Llama-3	50	37	2	8	17	4	2	5
Totals	258	174	21	70	69	21	3	40

[A] Many papers include several open model use cases; therefore, the quantity of identified use cases is greater than our sample of 258 (242 after de-duplicating) papers.

For the remainder of this section, we illustrate the ways researchers use open models across the 258 papers we analyzed. We walk through each open model use case, discuss some of the techniques and processes researchers employ within each category, and highlight notable case studies that elucidate how open models are used in practice. See Appendix D for the full list of open model use cases and all of their subcategories identified in our sample of papers.

Fine-Tuning

Fine-tuning is the process of adapting a pretrained model for a specific downstream task or domain. Researchers often fine-tune models on small, task-specific datasets or training examples (as opposed to larger, more general datasets used in pretraining) to refine their behavior, during which only a subset of the model's parameters are typically updated. Researchers can fine-tune open models on custom or proprietary datasets without any restrictions around how the fine-tuning is implemented, and without exposing private or sensitive data. Open weights give researchers full control over the fine-tuning process, allowing for full customization of the training parameters and techniques used for implementation, as well as the ability to examine how the model is updated.

Fine-tuning was identified in 67% of the papers in our sample, making it the most common use case. We identified many fine-tuning techniques across the research papers that require access to model weights, most of which involve introducing the model to new data related to a particular domain. Of the 174 papers that conducted fine-tuning, 149 (86%) involved introducing an open model to new data.[*] This data was typically labeled, and used to improve model knowledge and performance through supervised and instruction fine-tuning.

For example, researchers identified gaps in AI models' knowledge of Chinese vocabulary, mathematics, and scientific reasoning. They addressed these limitations by fine-tuning models on Chinese characters to extend their vocabulary,[31] mathematics question-answer pairs to enhance the model's math capabilities,[32] and scientific instructions related to physics, chemistry, and math to enhance the models' capacity for college-level scientific reasoning.[33]

Beyond introducing new data, researchers can also use open models to implement custom or novel fine-tuning techniques that are not offered through closed model APIs. Consider a popular technique developed by researchers at the University of Virginia and Princeton University. They found a simpler approach to reward functions, which made aligning model behavior more effective and computationally efficient.[34] They applied their technique by fine-tuning two open models and demonstrated that it outperforms other approaches, claiming that one of the resulting models was the strongest 8-billion parameter model at the time of release. These researchers could not

[*] Note that we annotated papers as introducing new data when we could identify the data. There may be instances where new data was introduced but not identified during our review or explained explicitly by the authors, and therefore are not included in our data.

have implemented their techniques on closed models through an API, nor could they have effectively compared the results of their work with other methods.

Organizations that publish this research may lack the resources to develop their own base models from scratch and are, to varying degrees, reliant on access to open models. The more open models that are available and the better those base models are, the more insight researchers may attain from fine-tuning them. For example, researchers at Ohio State University sought to make models more effective at chemistry-related tasks because previous research found poor performance in this domain. They developed a chemistry-related dataset that was used to fine-tune several open models, with one of the resulting models surpassing the performance of GPT-4 and Claude 3 Opus on many chemistry-related tasks.[35] This work benefited from access to a suite of performant open models that could, from their baseline general knowledge attained during pretraining, effectively integrate new chemistry-related data and allow for performance comparisons.

Lastly, it is important to highlight the downstream impact of research that fine-tunes open models, as it is often cited and adopted by other researchers. This is exemplified by a Stanford paper from 2021, where researchers developed an efficient alternative to conventional fine-tuning—called prefix-tuning—and implemented it on an open model to demonstrate its efficacy. Prefix-tuning adapts a model—in this case, GPT-2—to a specific task without updating a large subset of its parameters.[36] Their work was adopted by many across the research community, published in ACL-IJCNLP (a prominent international conference for computational linguistics), and has since been cited more than 4,000 times.

Continued Pretraining

Continued pretraining is the process of expanding a pretrained model's general knowledge.[*] Unlike fine-tuning, it typically has the same objectives as the initial pre-training, updates all or most of the model's parameters, and uses larger, more diverse datasets. This requires access to model weights because very large datasets cannot be used on closed model fine-tuning APIs, nor can all or most of a model's parameters be updated through the APIs. Note that the distinction between continued pretraining and fine-tuning can be unclear because there is no specific quantity of training data that determines when fine-tuning becomes continued pretraining. Therefore, we rely

[*] Continued pretraining is also referred to as continuous or secondary pretraining.

primarily on the terminology used in the papers, as well as reviews of their training datasets, to identify instances of continued pretraining.

Continued pretraining was identified in about 8% of the papers in our sample. We likely found fewer instances of continued pretraining compared to fine-tuning because it is more resource-intensive; researchers are generally more likely to use smaller amounts of data to optimize models for particular tasks. That being said, there were notable instances of this use case. Driven by significant gaps in AI's Spanish and Chinese language understanding, researchers improved the multilingual capabilities of GPT-2 and Llama-2 by continuously pretraining the former on a large amount of Spanish language data and the latter on a large amount of Chinese instructions.[37] At the time, this research helped reduce the availability gap between English- and non-English-speaking AI models.

Beyond instilling multilingual capabilities, researchers employ continued pretraining to expand the knowledge of open models in particular domains. The lack of performant open models with medical expertise, for example, motivated researchers to develop a suite of open models adapted to the medical domain and introduce an "optimized workflow to scale domain-specific pretraining for medical LLMs."[38] This involved continuously pretraining Llama-2 on a large, custom corpus of medical articles, abstracts, and guidelines. In another paper, researchers sought to "bridge the gap between the language of life and human natural language" by training Llama-2 on a large biomedical corpus.[39] In both cases, access to open models allowed the researchers to instill new, domain-specific knowledge without the need to pretrain models from scratch.

Examination

Examination involves analyzing a model's parameters, architecture, activations, and other internal characteristics, often with the intent of interpreting or tracking changes in model performance or behavior. This includes statically examining a model without any modifications, as well as examining how elements or characteristics of a model change when it is altered in some way (e.g., during fine-tuning). Note that our examination category does not encompass evaluating model performance on benchmarks as this involves assessing model outputs, which can be done via prompting and without access to the weights.

Examination was identified in about 27% of the papers in our sample. Researchers visualized and manipulated model attention to shed light on the process of in-context learning,[40] monitored the change in model parameters during training to determine the influence of certain data points on model behavior,[41] and reverse-engineered the operation of the feed-forward network layers—one of the building blocks of many language models—to unveil the internal prediction process.[42]

Examining model internals is an important aspect of interpretability research, which studies how models make decisions and how different model elements contribute to and influence AI behaviors. Model examination can, for example, help researchers understand how edits to a model affect its propensity to respond to unethical queries.[43] By changing individual layers of the neural network (i.e., a modification use case) and examining the corresponding shift in model responses, researchers found that even minor edits in a specific layer can lead to unaligned responses on subjects including biotechnology and finance. Such insights can, among other things, help design risk assessment protocols and guidelines for models that are adapted for different contexts.

The ability to access and examine AI model internals is also essential to the field of adversarial machine learning, which is concerned with studying AI vulnerabilities and improving model robustness. Knowledge of a target model's architecture, parameters, and gradients can be exploited to craft sophisticated "white-box" attacks that aim to manipulate model outputs through perturbations in inputs. In one example, researchers crafted adversarial input images that allowed them to control the model's output with an over 80% success rate.[44] One attack forced the model to generate an output of the attackers' choice, another broke the model's safety guardrails, and a third caused it to make a false statement. A final image attack was able to extract information from the model's context window. These findings raise concerns about the security of multimodal models that process unverified image inputs, and the risk of attacks that could spread malware, extract sensitive information, break model safeguards and produce disinformation.

Note that none of the open models investigated in this report have openly available pretraining data, nor comprehensive documentation of how that data was used during pretraining. This likely impacts the types of research being conducted in our sample of papers, particularly among those that examine models. For example, models such as BLOOM and OLMo come with open and documented training data, which allows for deeper examination of how training data impacts model behaviors, strengths and weaknesses, biases, and risks.[45] A more in-depth study of how open data impacts research should incorporate such models. But for the purposes of this report, which focuses explicitly on the use cases enabled by open weights, we do not consider the impact of open data.

Compression

Compression encompasses a range of techniques that reduce a model's computational or memory footprint while maintaining performance to the extent possible. This includes reducing a model's parameters or numerical precision (i.e., how many bits are represented per parameter in a model), as well as white-box distillation of a model's knowledge into another, often smaller model (i.e., converting a model's knowledge into a smaller form-factor).* This makes models more efficient, faster, and easier to run, which is particularly important for researchers or organizations that lack the resources to run larger models, which are often the very same entities that benefit from open models in the first place.

Compression was identified in 27% of the papers in our sample. Researchers often tested, demonstrated, and optimized compression techniques to maximize efficiency, which is not possible without direct access to weights. For example, researchers developed a technique to compress the parts of a model that store the values of previously generated text (i.e., tokens), thereby allowing a model to better attend to longer inputs during inference.[46] They compared and combined their compression technique with those put forward by other researchers (who also demonstrated their techniques on open models). [47] In another case, open models allowed researchers to test a range of post-training compression techniques (e.g., quantization), evaluated the impact on model performance, and provided recommendations for how to best implement model compression.[48] In both papers, the researchers compared compression methods on several models of different sizes, illustrating how access to a diverse set of open models enables more robust experimentation.

Compression research is often implemented alongside other open model use cases, such as reducing the resource footprint of the fine-tuning process. For example, researchers empirically evaluated a range of techniques to significantly reduce the precision of Llama-3 models, as well as improve the efficiency of fine-tuning the model on new data (e.g., through low-rank adaptation).[49] Their intent was to explore how extreme compression during fine-tuning or post-training, which makes the models

* "White-box" knowledge distillation is a form of distillation that requires access to model weights. It involves using the internal information of a teacher model, such as its logits and features, to transfer knowledge to a student model. Conversely, "black-box" knowledge distillation does not require access to model weights, as it only uses the teacher model's outputs to transfer knowledge. (Zhu et al., "A Survey on Model Compression for Large Language Models"; Chuanpeng Yang et al., "Survey on Knowledge Distillation for Large Language Models: Methods, Evaluation, and Application," arXiv preprint arXiv:2407.01885 (2024), https://arxiv.org/abs/2407.01885.)

much easier to train and run on limited hardware, degrades performance in the most capable open models. In doing so, they found significant performance degradation and identified areas that need to be improved in future research. Their highly-cited work showed "the potential challenges of deploying Llama-3 in resource-constrained environments and underscores the ample room for growth and improvement" in extreme compression.

Other research sought to compress open models by removing parameters (i.e., pruning) and distilling knowledge, with the same goal of improving efficiency and reducing the compute needed to run AI models. For example, NVIDIA researchers investigated the effectiveness of structured pruning with knowledge distillation to compress the 12 billion parameter Mistral model and the 8 billion parameter Llama-3.1 model.[50] They investigated two strategies for doing so, and were able to compress these models to 8 billion and 4 billion parameters, respectively. Their resulting compressed Mistral model outperformed "all similarly-sized models across the board on common language modeling benchmarks," and the weights were subsequently opened for others to use and build upon. In another paper, researchers introduced a new sparsification technique that allows for the removal of large portions of a model's parameters without significantly reducing performance.[51] They demonstrated how their technique, which was applied to several open models, resulted in sparsified models that could "run on fewer GPUs and run faster without any additional code optimization."

Lastly, it is worth highlighting that once researchers implement their compression techniques, the resulting models are often released for the AI community to use and build upon. Many of the aforementioned papers were released alongside the compressed versions of the open models they used, allowing anyone to use those more efficient models on limited hardware. In that sense, compression, like open models, promotes AI accessibility to lower-resource organizations that may lack the compute hardware to run (or train) large open models.

Combination

Combination encompasses any technique that, as the name suggests, combines different open models, their various parts, or their modalities. Combining models offers several advantages both in terms of performance and cost, and allows researchers to reuse existing models while avoiding the need for extensive pretraining. The resulting models also often outperform their individual components by combining their strengths, capabilities, and modalities, and generalizing better across various tasks.

Combination is a relatively uncommon open model use case in our sample, with about 8% of papers engaging in this practice. We identified several techniques and strategies for combining models. Weight mixing, for example, refers to the combination of encoder and decoder architectures to construct a new model. Researchers from Cairo University used this technique to develop a model that could support radiologists in writing reports about chest x-ray findings.[52] They combined an image-recognition model pretrained on images of chest x-rays as the encoder with a large language model as the decoder to produce the full-text report, and released their final model and code to encourage further scrutiny and research on the subject.

Another combination method is model merging, which “blends” different models’ weights and can extend their knowledge base without additional training. For example, researchers experimented with three different methods for merging a general-purpose model with a domain-specific biology model to enhance the specialized model’s adaptability and generalization across a wider range of applications.[53] In combination with supervised fine-tuning, their merged models achieved performance levels comparable to GPT-3.5 Turbo (a state-of-the-art closed model at the time) on popular medical benchmarks. This work illustrates how, through creative and resource-efficient model combination techniques, open models can compete with leading proprietary models on domain-specific tasks.

Researchers also use combination techniques to instill multimodal capabilities in open models (i.e., the ability to understand multiple kinds of data, such as text and images). This process, which we term “modality synchronization,” generally involves one model for each data modality, as well as an alignment module that “translates” the output from one modality into another. For example, researchers built a multimodal system called SkinGPT-4 that consists of a vision model, an open language model, and an alignment layer which establishes the interaction between the two.[54] SkinGPT-4 is an interactive dermatology diagnostic system trained on skin disease images, clinical concepts, and doctor’s notes. The idea behind the system is to process images of skin diseases and “determine the characteristics and categories of skin conditions, perform analysis, provide treatment recommendations, and allow interactive diagnosis.”[55] Evaluations of SkinGPT-4 show promise in terms of accuracy and consistency, suggesting that open models are a strong alternative to proprietary models in the medical domain, where privacy concerns may present obstacles to data transfers and API use.

Hardware Benchmarking

Hardware benchmarking involves running, testing, or training AI models on different types of computing hardware, with the intent of measuring the functionality of models on hardware or the performance of hardware when running models.[56] This use case, which was identified in only three of the papers in our sample, fundamentally requires access to the model weights because no closed model API allows for customizing the hardware on which the models run. The need for weights is even more salient for any research that seeks to run models locally on specific devices, such as laptops, because the weights themselves must be loaded and processed on the device's onboard chips.

Hardware benchmarking may involve compressing AI models to gauge whether they can run effectively on certain types of compute hardware, be it edge processors embedded in resource-constrained devices (e.g., smartphones) or high-end GPUs found in datacenters. For example, in an effort to bring AI models closer to running on-device applications, researchers developed a multimodal version of Llama-3, fine-tuned it to be multilingual, and compressed it to run on more limited hardware.[57] They then tested the models locally on different types of devices (e.g., a Macbook Pro and a Xiaomi 14 Pro) and found that they could "operate efficiently on both mobile phones and personal computers."

In another paper, researchers developed an efficient LLM inference system that better utilizes on-device compute hardware. They demonstrated the effectiveness of their approach by compressing several open models from the Qwen-1.5, Mistral, and Llama-2 families in various ways. They found that the compressed models could run efficiently on two mobile devices (the Xiaomi 14 and Redmi K60 Pro), thereby "paving the way towards practical on-device" inference.[58]

Modification

This catch-all category of use cases encompasses any modification to or augmentation of a model that does not fit within the other categories, and was identified in 16% of the papers in our sample. There are a range of ways to modify models, which we bucket into four subcategories based on their objectives: enhance model efficiency, improve model performance, align model behavior, and extend the amount of text a model can effectively process in a prompt (i.e., context windows). As illustrated in the below examples, we found that most modifications to open models occur alongside other use cases, such as examination or fine-tuning.

Several papers applied algorithms and frameworks to enhance model efficiency and accelerate training. For example, when exploring the potential of fine-tuning models to perform better at information retrieval tasks, researchers applied optimization algorithms to Llama-2 and Mistral models during fine-tuning, thereby alleviating computational and memory bottlenecks.[59] They also applied an optimization framework for efficient distributed fine-tuning of the models across several GPUs.[*]

Another impactful paper that was published by university researchers in 2022 involved the modification of GPT-2 to improve its efficiency, speed, and performance. They presented a novel attention algorithm that reduced the memory and compute needed for the model to attend to long context data (e.g., understanding the context from the first chapter of a book while reading the final chapter).[60] This allowed models to both train and run more quickly, helped overcome bottlenecks in scaling, and ultimately improved performance. This innovation has since been widely adopted and improved upon, and the paper itself was published through NeurIPS 2022 and cited more than 2,000 times. Notably, at least seven other papers in our sample used this algorithmic innovation in some way.

To align model behavior, researchers from Singapore modified the Llama-2 and Mistral models to enhance their defenses against jailbreak attacks.[61] They examined the internal layers of the models, identified "several critical safety layers," and edited them to enhance the models' defenses. They tested the modified models and found they could "effectively defend against jailbreak attacks while maintaining performance on benign prompts." This research required access to the model weights, as it involved both the examination and modification of open models.

To extend context windows, researchers at Meta continuously pretrained Llama-2 models on long-context data, then modified parts of the model that attend to positional information (e.g., the order of words) within prompts.[62] The processing of long-context inputs is important for a model's ability to analyze long documents or maintain lengthy conversations with users. In combination, the researchers' techniques allowed the resulting model to effectively attend to longer context lengths, which surpassed "GPT-3.5-Turbo-16k's overall performance on a suite of long-context tasks." In another paper, NVIDIA researchers modified the Llama-2 model to compare the effectiveness of extending its context window versus giving it retrieval-augmentation (i.e., allowing the

[*] "To optimize memory usage and accelerate training, we applied Deepspeed ZeRO stage 2 (Rasley et al., 2020) and BFloat16 mixed precision techniques. Additionally, Flash attention (Dao et al., 2022) was used to further improve training efficiency." (Zhu et al., "INTERS: Unlocking the Power of Large Language Models in Search with Instruction Tuning").

model to search across a database or web-browser to find information), as well as a combination of both approaches. They demonstrated how retrieval improves the performance of short- and long-context models, and combination of both approaches leads to optimal results.[63]

Use Case Overlap Between Open and Closed Models

As previously discussed, there is a degree of overlap in how open and closed models can be used. This overlap is displayed in Table 3, and falls within three broad use case categories: prompting, fine-tuning, and examination. Anyone can prompt models that are accessible through the internet (whether or not they are open or closed), either via a chat window or programmatically via an API.[64] We therefore do not consider access to model weights to be a requirement for prompting and do not include it as a use case exclusive to open models.[*]

Table 3. Use Cases of Open and Closed Models, by Level of Access

	Level of Access		
	Open Weights	API[A]	Web-Based UI
Use Cases	Prompting (via UI or API)	API / Programmatic Prompting	Prompting
	Fine-Tuning	Limited Fine-Tuning	
	Examination	Limited Examination	
	Continued Pretraining		
	Compression		
	Combination		
	Hardware Benchmarking		
	Modification		

Note: **Green** indicates full overlap, where the use case is not impacted by the accessibility of model weights. **Yellow** indicates partial overlap, where certain subcategories and processes of the same broad use case require access to model weights while others do not require access to model weights. **Blue** indicates use cases that exclusively require access to model weights.

[A] See Appendix B for an overview of the features offered through OpenAI's fine-tuning API.

[*] Note that we did not assess how open models were prompted during our review of papers, and there may have been instances where open model hyperparameters were adjusted in ways that are not possible with closed models.

Fine-tuning and examination, on the other hand, are two common use cases of open models, but some developers allow for limited fine-tuning and examination of closed models through an API. Such APIs come with limitations and are generally far less useful to researchers than open models.[65] For example, OpenAI's API, which is the primary closed model API assessed in this report, only allows researchers to upload a 512MB file of fine-tuning data and set a handful of training parameters, as well as only examine the training loss, validation loss, and training token accuracy during fine-tuning.[66] At no point can researchers examine model internals or "look under the hood" of the fine-tuning process to understand what is changing, nor can they implement custom fine-tuning techniques. Such API limitations are, to a degree, likely meant to protect the developers' IP, and to prevent outside organizations from making deductions about and potential replications of their models' internal operations.

We find that most papers in our sample that fine-tuned or examined open models could not have done so with closed model APIs: They either used fine-tuning techniques not offered through the API, customized the fine-tuning process in ways that cannot be done through the API, or used more fine-tuning data than what is permitted through the API (i.e., 512MB). However, there were some instances where we could not determine whether researchers could have used fine-tuning APIs because they did not explicitly state the amount of fine-tuning data that was introduced.

How Researchers Use Closed Models

Following our review and annotation of papers using open models, we now assess how researchers use closed models that are accessible through websites and APIs. This section is intended to contextualize open model use cases, delineate between what can or cannot be done when weights are not directly accessible, and highlight some opportunities and constraints associated with the APIs provided by closed model developers.

Through a review of about 200 research papers, we identified 129 published between 2022–2024 that used the closed models GPT-4 or Claude 3, all of which involved prompting the models in some way. Through a more targeted investigation of API fine-tuning, we identified 8 additional papers that fine-tuned closed models via OpenAI's API. We found no instances of fine-tuning through APIs provided by other developers.

Prompting

There is a wide range of research that can be conducted through prompting. This includes assessing model performance; evaluating safety, trustworthiness, and bias; labeling data; generating synthetic data; and using the models themselves to conduct evaluations. Prompting also encompasses retrieval augmented generation (RAG), where a separate retrieval system is used to search an external data source (e.g., Google search) based on the user query, retrieve the top passages, and then provide those top passages to the model. Much of this can be done via direct prompting, but some must be done programmatically through an API where the inputs and outputs can be more structured and automated.

While open and closed models can be equally used for prompting, closed models may offer unique utility to certain researchers for this use case. Many closed models come with refined UIs and APIs, making them easier to prompt than open models that are downloaded, set up, and run locally on custom computing infrastructure or in the cloud. This requires expertise and compute resources that may not be readily available to some researchers, particularly those outside of computer science disciplines. Moreover, researchers investigating frontier AI model behavior require access to state-of-the-art models, which have been closed since GPT-2. For these reasons, some researchers may opt for closed models if their investigations rely exclusively on prompting.

Whether done through a website or an API, prompting allows for the evaluation of a model's performance, capabilities, and limitations, be it broad tasks such as Q&A and summarization or particular domains such as coding, math, or medicine.[67] Performance evaluations involve assessing a model's outputs for their correctness or quality, often against a benchmark (i.e., a standardized set of questions in a specific domain that facilitates performance comparisons between models). For example, researchers from the University of Nanjing and the University of Illinois Urbana-Champaign developed a novel evaluation framework to test the functional correctness of LLM-synthesized code, and built an extended benchmark that tests models' programming capabilities.[68] Through prompting alone, they were able to assess and compare the performance of 26 different models, open and closed.

Lack of API access can make model performance evaluations difficult, such as when researchers evaluated the mathematical capabilities of OpenAI's closed models to "test whether ChatGPT and GPT-4 can be helpful assistants to professional mathematicians."[69] Since API access to GPT-4 was not available at the time of research, the authors relied on OpenAI's web interface to carry out the evaluations, which posed several challenges. First, there was no official information about which exact model version was powering ChatGPT at the time of assessment. Second, researchers were unable to adjust model hyperparameters such as the temperature (which controls the randomness of the generated text) through the web interface, and no information was provided about the default values that had been chosen. As such, a lack of API access and insufficient transparency from the model provider can restrict researchers' ability to understand model behavior beyond a straightforward assessment of performance.

Closed models can also be prompted to generate synthetic data for further training.[70] This is particularly relevant for aligning language models with human preferences, which requires training data and reward signals that reflect those preferences, often in the form of feedback to model outputs. Curating this data can create a bottleneck because it often requires large amounts of time and labor. To address this issue, researchers developed a scalable alternative to human-generated feedback data by eliciting such outputs from GPT-4. By establishing "a comprehensive AI feedback collection pipeline," the authors built ULTRAFEEDBACK, a dataset of over 1 million scalar preferences and textual feedback from GPT-4, which exceeded existing datasets both in terms of size and generality.[71]

Similar to performance evaluations, prompting a model and scrutinizing its outputs can serve to assess its trustworthiness, reliability, and fairness (among other properties). This includes the elicitation of model confidence, meaning how accurately a model expresses its certainty about the factuality, correctness, and truthfulness of its output. Uncertainty quantification is essential for the reliable application of LLMs in decisionmaking tasks.[72] While such assessments can benefit from access to model internals, in one paper researchers designed a systematic framework for confidence elicitation in closed language models.[73] Their strategy, which relied on prompting, response sampling and aggregation techniques, found that models tend to exhibit overconfidence in their responses. Although their technique did not perform significantly worse than white box approaches (i.e., those that have full access to model internals), the researchers acknowledged that further transparency, such as access to the model logits, would ameliorate some of the restrictions they faced.

In another paper that sought to identify AI robustness and reliability limitations, researchers prompted 15 large vision language models (LVLMs), including GPT-4V and Claude 3, to assess their propensity for language hallucinations and visual illusions.[74] Language hallucinations describe instances when the model generates a response that does not match the provided visual input, whereas visual illusions refer to the misinterpretation of visual inputs. The authors assembled a benchmark test consisting of 346 images paired with 1,129 questions to "systematically dissect and analyze the diverse failure modes of LVLMs." Among other findings, their results indicate that LVLMs are easily misled by simple image manipulations such as order reversing, image flipping, and color editing. Given the diagnostic capacity of the benchmark, it could "be used to identify and provide insights on the weakness of different LVLMs, [and] to facilitate finetuning [sic] and improvement of those models based on the diagnoses."

Limited API Fine-Tuning and Examination

Instances of closed model API fine-tuning and examination typically involved exposing the model to a relatively small subset of new data, with the intent to assess changes in performance, elicit certain behaviors, or identify flaws. As previously mentioned, there are limitations to what researchers can do with these fine-tuning APIs (see Appendix B for more details), and such limitations were sometimes explicitly mentioned in the papers. For example, one paper involved fine-tuning GPT-3.5 to improve performance on the bar exam. While the researchers could adjust a handful of hyperparameters (e.g., the temperature), they stated that their "ability to interpret the nature or impact of these hyperparameters is limited by OpenAI documentation and API functionality."[75]

Notwithstanding the limitations, there are several reasons why researchers may see value in these more limited fine-tuning APIs: Many closed models are highly performant, and access may be sufficient if the intent is merely to assess model outputs without any modifications or examination of model internals. Researchers may also find it easier to set up API fine-tuning instances than it is to provide their own computing infrastructure, and the data cap is large enough to do relatively significant degrees of fine-tuning without the need to implement custom fine-tuning techniques.

In one paper that demonstrated how fine-tuning can compromise safety-aligned models, researchers fine-tuned GPT-3.5 Turbo (as well as the open Llama-2 model) on a handful of "adversarially designed training examples" using OpenAI's API.[76] This degraded the model's ability to refuse harmful prompts with a limited amount of new data. They also found that fine-tuning on a benign dataset of over 50,000 data samples unintentionally degraded safety alignment. In both instances, the API was useful because (1) the closed model was performant and safety-aligned, which was necessary for the research; (2) the amount of fine-tuning data they introduced fell below the maximum amount of data permitted by OpenAI's API; and (3) for the purposes of the research, they did not require more granular control over the fine-tuning process.[*]

In another paper, researchers demonstrated how models fail to generalize on new data. They provided evidence for the "reversal curse," where a model is trained on sentences stating "A is B" but fails to generalize that information in the reverse direction, "B is A."[77] They fine-tuned both closed and open models (GPT-3 and Llama-1) on fictitious statements, and found that both models failed to generalize in reverse orders. This could be done via API because the fine-tuning data only contained 3,600 examples, which is under the maximum amount of fine-tuning data permitted by the API.

In a final example, researchers demonstrated how GPT-3.5 could act as a motion planner for autonomous vehicles.[78] They did this by first converting motion planning data into language that the model could understand, implementing a "novel chain-of-thought reasoning strategy specifically designed for autonomous driving," and then fine-tuning the model to align its outputs with the behaviors of human drivers. They collected human driving examples (from driving logs) and ground truth guidance of chain-of-thought reasoning, then fine-tuned the model using the ground truth data. Notably, the authors stated that "due to the limitations of the OpenAI APIs, we are unable to obtain the inference time of our model. Thus, it remains uncertain whether our approach can meet the real-time demands of commercial driving applications."

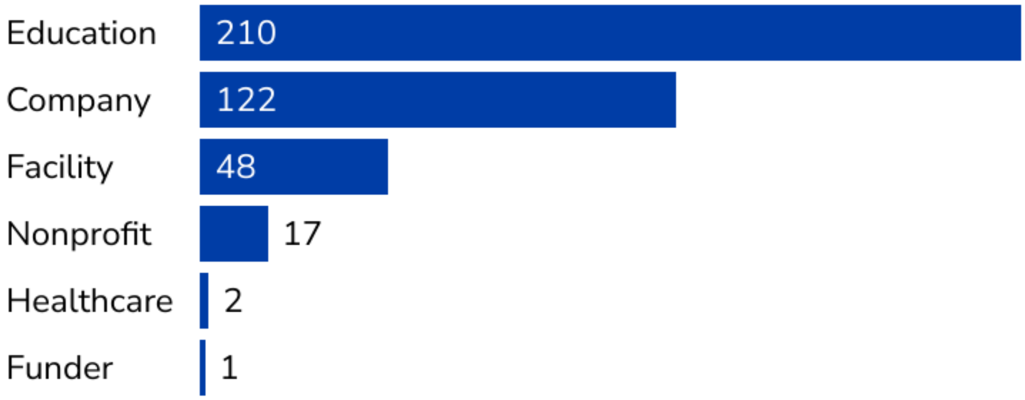
[*] They fine-tuned GPT-3.5 Turbo on the Alpaca dataset, which "consists of 52K instruction-following data generated from OpenAI's text-davinci-003 model" (Qi et al., "Fine-Tuning Aligned Language Models Compromises Safety").

Who Is Doing the Research?

Our sample of papers that used open models was authored predominantly by researchers at universities or companies, as displayed in Figure 2, often with two or more organizations working in collaboration. The majority (87%) of the papers were authored, in whole or in part, by educational organizations. The prevalence of universities illustrates, in some ways, the utility of open models. It is these very organizations—academic institutions that often lack the resources to develop pretrained base models from scratch but do have the talent and resources to use, experiment with, or modify pretrained models when they are accessible—that greatly benefit from open models. They are also the most likely types of organizations to openly publish their research, which is likely the primary reason for their preponderance in our sample. Companies also represent a significant portion (50%) of the research, much of which was produced in collaboration with universities (35%). This degree of collaboration is a common theme across the sample, with over 70% of the papers being written by multiple types of organizations.

Overall, the prevalence of these sectors aligns with trends in broader AI research: According to the 2025 AI Index, academia produces the most amount of highly cited publications, followed by academia and industry working in collaboration, followed by industry.[79]

Figure 2: Number of Papers Using Open Models, by Author Organization Sector

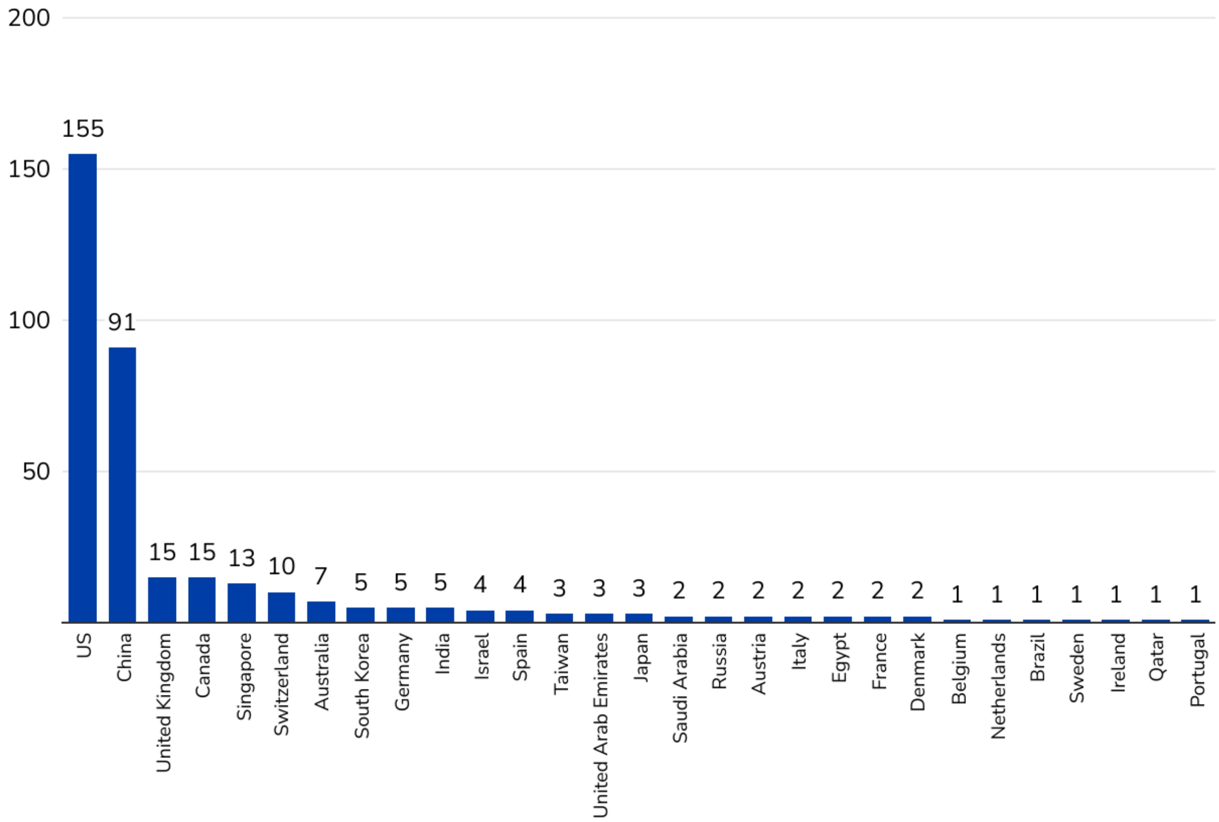


Source: CSET Merged Corpus and manual annotation of organization type metadata.

Note: Facility organizations are defined as a “specialized facility where research takes place, such as a laboratory… or dedicated research area.”[80]

Figure 3 below displays the geographic (country-level) diversity of the authors that produced the papers we reviewed. The U.S. and China are the primary country locations for the authors, which aligns with the broader AI research output produced by these countries.[81] We see that 65% of papers have authors from U.S. organizations, including 37% with only U.S.-based authors. Researchers based in China authored 38% of papers, including 18% with only China-based authors. About 35% of the papers were collaborations across two or more countries, including 12% of the papers being U.S.-China collaborations. There are far fewer papers from countries outside of the U.S. and China, with the United Kingdom (6%), Canada (6%), Singapore (5%), and Switzerland (4%) being the next most prominent.

Figure 3: Number of Papers Using Open Models, by Author Organization Country



Source: CSET Merged Corpus and manual annotation of country metadata.

Conclusion

This paper assesses the use of open models in research. Our objectives were twofold: First, to fill an important knowledge gap in the literature about the real-world use of open models in research, and second, to inform an ongoing policy debate by providing empirical evidence to an otherwise largely speculative discussion.

Our findings illustrate the wide-ranging use cases that open models enable and highlight the important streams of research that these use cases support. Based on our review of hundreds of research papers using open and closed models, we identified seven model use categories that exclusively or predominantly require access to the model weights. These open model use cases encompass a multitude of techniques and methods developed to make models more performant, expand their capabilities and knowledge, improve their efficiency, enhance their accessibility, and more. They support many advances in the field of AI and computer science, as well as other fields of research that leverage AI to varying degrees. Moreover, as we highlight throughout this paper, researchers that use open models often release their code, data, and model weights, making it easier for others to replicate and build on their work.

API access to closed weight models does not offer comparable research opportunities. Aside from limited access, a major obstacle to building a research agenda on API access is the dependency it creates. Companies can change the features of their APIs, shaping the level of control and insight a user has over and into the model. Case in point: Over the course of this project, OpenAI updated its fine-tuning API twice, most recently by lowering the maximum data volume for fine-tuning from 1GB to 512MB. This creates uncertainty around the reliability of any existing API feature, and can make API-based projects less attractive for researchers who require long-term consistency in the mode of access to a model.

In our sample, U.S. researchers are the primary users of open models, producing more research papers meeting our criteria than authors in China and other countries. Unsurprisingly, the majority of authors among papers we reviewed are affiliated with academic institutions rather than private enterprises. This is an expected pattern in our sample because academics are more likely to publish their research, but may also reflect the fact that open models can alleviate resource constraints faced by less well-endowed academic entities.

Despite the limitations of our methodology, which are listed in Appendix C, we argue that tracking the use of open models is an essential piece of AI measurement. It provides important nuance around the utility of open models, and helps contextualize the potential impact of policies that promote or impede the open release of AI models. Moreover, we argue that the release of highly performant open models by U.S. competitors, such as DeepSeek's R1 and Alibaba's Qwen-3 models, makes this type of measurement in future work even more important; competitors that recognize and promote the value of open AI can, too, benefit from the research that it enables.[82]

We recognize that policy action around open models requires careful weighing of the risks and benefits of model release. This report adds to the discussion by providing empirical insight into the impacts of open models on research. While we find these benefits to be substantial, the report's findings should not be taken to mean that the release of open models is unequivocally positive or carries no risks. For a broader discussion of both the risks and benefits associated with open models, we point readers to existing analyses.[83]

Looking forward, we hope future work will build off this report and track the use of newer, more performant open models, as well as other model components, such as open training datasets. We also hope this work will supplement research into the use of open models by companies and government agencies that do not publish their work but nonetheless benefit from access to open models that they can customize and use freely.

Appendices

Appendix A. Criteria for Selecting Model Families and Papers

The open model families investigated in this report must:

- Appear in the top 100 most-downloaded LLMs on Hugging Face (for any single model within a model family; not aggregate across the entire model family).
- Appear at least 4 times on the Hugging Face Open LLM Leaderboard.
- Have been released in or before April 2024.

We do not assess model families released after April 2024 because newer models will have fewer associated papers, so there is less opportunity to gauge research impact. For closed model families, we selected GPT-4 and Claude 3 because they were both state-of-the-art closed models in early 2024.

Table 4 below displays the open models we selected based on the criteria. Note there are two exceptions in the models we selected: We included Qwen even though it only appears on the LLM Leaderboard twice, and Qwen-1.5 even though it does not appear in the top 100 most-downloaded models on Hugging Face. We did this to increase the number of Chinese-based models in our sample, and to compensate for potential biases in the Hugging Face data (e.g., Chinese users may download Qwen-1.5 models from non-English platforms outside of Hugging Face).

After model families were selected, we queried the CSET Merged Corpus using the model family names (i.e., a keyword search that pulls all papers that mention the model family in the titles and/or abstracts). We then sorted papers by citation counts and reviewed the top 100 most-cited papers for each model family, but only reviewed papers with two or more citations. We followed the same process for the two closed model families, GPT-4 and Claude 3.

Table 4: Selected Open Model Families[A]

Open Model Family	Total Download Rank	Leaderboard Count	Paper Count (With 2+ Citations)[B]	Organization	Release Date	Query Date[C]
GPT-2	1	4	1,163	OpenAI	Feb–Nov 2019	Sep 2024
Llama-2	75	7	522	Meta	July 2023	Sep 2024
Falcon	10	5	229	TII	Jun–Sep 2023	Oct 2024
Qwen	15	2[D]	35	Alibaba	Sep 2023	Dec 2024
Mistral	27	26	197	Mistral AI	Oct 2023	Nov 2024
Qwen1.5	187[D]	19	11	Alibaba	Feb 2024	Dec 2024
Phi-3	25	11	11	Microsoft	Apr 2024	Oct 2024
Llama-3	55	31	226	Meta	Apr 2024	Jan 2025

[A] All downloads and leaderboard counts were pulled from Hugging Face in October and September 2024, respectively. Leaderboard counts of Qwen were pulled from the old LLM Leaderboard (which was updated during the research for this report) in August 2024.

[B] The count of papers that mention the model in the title/abstract, and have two or more citations. All papers with fewer than two citations are omitted.

[C] The date papers were pulled from the CSET Merged Corpus, based on the keyword searches of model families.

[D] Qwen only appears on the old LLM Leaderboard twice and Qwen1.5 does not appear in the top 100 most-downloaded models on Hugging Face, but we incorporated them into our investigation to increase the diversity of models developed outside of the U.S.

Appendix B. Overview of the OpenAI Fine-Tuning API

To gauge whether the fine-tuning use cases of open models could be conducted on closed models via an API, we assess the features and customizability offered through OpenAI's fine-tuning API. We use OpenAI's fine-tuning API as a benchmark because, compared to other APIs, it is relatively more customizable and better infuses new knowledge into closed/proprietary models.[84]

As of June 2025, OpenAI's fine-tuning API allows a user to:

1. Select between eight closed models, most of which are variants of GPT-3, GPT-4, and GPT-4o.

2. Upload a maximum of 512MB of data to be used for fine-tuning. Users can request an increase in this cap.

3. Select one of three fine-tuning options: supervised fine-tuning (SFT), direct preference optimization (DPO), and reinforcement fine-tuning (RFT). There is little customizability beyond selecting one of these options. For RFT, only a variant of the o4 reasoning model can be used.

4. Specify a handful of hyperparameters: batch size, learning rate multiplier, and number of epochs.

5. Examine the training loss, validation loss, and training token accuracy during fine-tuning.[85]

Importantly, OpenAI updated their fine-tuning API during our analysis of papers that used open models.[86] Their API previously allowed a user to upload a maximum of 1GB of fine-tuning data (which has since been reduced to 512MB).[87] Moreover, fine-tuning with DPO or RFT were not permitted by the API during our review and annotation of papers that use open models, nor was it available to the authors of said papers. Therefore, DPO, which is a common use case of open models in our sample of papers, remains an open models use case in our taxonomy.

Appendix C. Methodological Limitations

There are several limitations to our methodology. While these limitations impact what can be inferred from the data, they also reflect the outcome of careful considerations of data availability, validity, scale, and resource constraints during study design and analysis. Trade-offs between these factors are inevitable, and we are unaware of superior alternatives to our approach. We overview the limitations below, and, to the extent possible, offer an explanation for the choice and how we attempted to mitigate the resulting shortcomings:

- We used keyword searches of open and closed models' family names to create an initial sample of papers to investigate, which biases our sample of papers to those that mention, as opposed to use, the models in question. There may be many highly-cited papers that use open models but do not mention the models' names in the title or abstract, and are therefore missing from our sample. There is, however, no reason to believe that those papers are systematically different from the papers in our sample in how they use open models.

- We assessed the use of open and closed models in research publications and preprints, and therefore cannot make inferences about the use of open or closed models outside of published research.

- We annotated fewer papers that used closed models compared to open models. Moreover, the annotation of papers that used closed models underwent fewer iterations during taxonomy development than the annotation of papers that used open models. The primary reason for this approach was that the review of the initial 200 closed model papers did not reveal any use cases beyond model prompting, which led us to expect that widening our sample would not bring additional insights.

- We used paper citations as a proxy for impact or significance, recognizing the limitations of this proxy. High citations counts do not necessarily mean high-impact, open access papers tend to be cited more frequently than those behind paywalls, and some papers may have a significant real-world impact on policy or industry without being heavily cited in academic literature. While we required at least two citations for a paper to be included in our sample, we also cannot rule out the possibility of self-citation inflating citation counts. However, no other metric of similar validity would have allowed us to conduct our analysis at this scale.

- There is an over-representation of U.S. open models in our sample of open model families. Similar to how we selected papers, we aimed to study popular and performant models. To do so, we used download counts and leaderboard occurrence from Hugging Face as proxies. This may, however, underrepresent models developed by other countries, since foreign users may prefer other platforms. Chinese users, for example, are known to frequent ModelScope, a similar platform to Hugging Face, to download and use Chinese models.[88] To counteract the limitations of our metrics and widen the geographic scope, we included two Chinese models in our analysis that would not have met the threshold otherwise.

- Our selection criteria for model families (performance and popularity) results in the exclusion of models that are not optimized for benchmark performance, which may introduce bias to our sample. While performance and popularity matter, other model characteristics, such as the availability of training data or intermediate checkpoints collected during training, also influence researchers' selection of models for their studies. This means that less-performant models that might be valuable for some streams of research because of other characteristics are not represented in our sample and consequently, neither are the research use cases they enable.

- There is an inherent time lag in monitoring innovation effects in this way, since it takes time for research to be conducted and cited following the release of a new model. Therefore, the insights from our data are largely historical, and our methodology cannot be applied to new research using newer models.

- The Hugging Face Open LLM Leaderboard is no longer active.[89] Therefore, the same methodology could not be conducted with newer models, and would require the use of different benchmarks or leaderboards.

Appendix D. Full Open Model Use Case Taxonomy

Below is the full taxonomy of open model use cases identified during our review and annotation of 258 research papers. Each use case and subcategory was identified in at least one paper that we reviewed. Many use case subcategories are specific techniques and processes that are defined in the papers. However, to help categorize use cases, we created several categories within [brackets] that reflect the overall objective of the research, rather than specific techniques or processes. See CSET's GitHub page for the full dataset of open model papers and use case annotations.

Table 5: Fine-Tuning Use Cases

Fine-Tuning Categories	Subcategories	Secondary Subcategories	Tertiary Subcategories
New Data[A]			
Supervised Fine-Tuning (SFT)	Instruction Fine-Tuning (IFT)	Noisy Embedding Instruction Fine-Tuning (NEFTune)	
		Length-Instruction Fine-Tuning (LIFT)	
	Odds Ratio Preference Optimization (ORPO)		
	Attribute Conditioned SFT		
	Recursive Introspection (RISE)		
Unsupervised Fine-Tuning (UFT)			
[Preference Alignment and Optimization]	Reinforcement Learning (RL)	Reinforcement Learning From Human Feedback (RLHF)	Reinforcement Learning From Evol-Instruct Feedback (RLEIF)
			ReMax

			Weighted Preference Optimization (WPO)
			Iterative Nash Policy Optimization (INPO)
	Absolute-Rating Multi-Objective Reward Model (ArmoRM)		
	Proximal Policy Optimization (PPO)	Step-by-Step PPO	
	Kahneman-Tversky Optimization (KTO)		
	Direct Preference Optimization (DPO)		
	Simple Preference Optimization (SimPO)		
	Self-Play Preference Optimization (SPPO)		
	Identity Preference Optimization (IPO)	Continuous-Adversarial IPO (CAPO)	
	Directional Preference Alignment (DPA)		
	Trust Region (TR)		
	Self-Exploring Language Models (SELM)		

	Best-of-N Sampling		
	Reward-Aware Preference Optimization (RPO)		
	Self-Supervised Alignment with Mutual Information (SAMI)		
Low-Rank Adaptation (LoRA)	Quantized LoRA (QLoRA)		
	Quantization-Aware Low-Rank Adaptation (QA-LoRA)		
	Generalized LoRA (GLoRA)		
	LongLoRA		
	S-LoRA		
Unlearning			
Agent Fine-Tuning			
AgentQ			
Continuous-Adversarial Unlikelihood (CAT)			
FastMem			
Stepwise Internalization (ICoT-SI)			
Rotary Position Embedding (RoPE)			

Prefix-Tuning			
Adapter-Tuning			
Context Gates			
Discriminative Fine-Tuning			

[A] Unlike other fine-tuning categories, the “new data” category is not a specific technique or process. We annotated papers as using new data when we identified specific quotes from the authors that indicated they used new data during fine-tuning. When possible, we identified the specific type and quantity of new data that was introduced by the researchers.

Table 6: Continued Pretraining Use Cases

Continued Pretraining Categories
New Data[A]
Domain-Adaptive Pretraining (DAPT)
Instruction Pretraining

[A] Unlike other continued pretraining categories, the “new data” category is not a specific technique or process. We annotated papers as using new data when we identified specific quotes from the authors that indicated they used new data during continued pretraining. When possible, we identified the specific type and quantity of new data that was introduced by the researchers.

Table 7: Examination Use Cases

Examination Categories	Subcategories
Parameters/Weights	DataInf
White-Box Attack	
Gradients	
Learned Representations	Embeddings
Layers	
Attention	
Activations	
Neurons	
Logits	Perplexity
	Emulated Fine-Tuning (EFT)
Loss	
Probabilities	
Vectors	

Table 8: Compression Use Cases

Compression Categories	Subcategories	Secondary Subcategories
Sparsification	Pruning	
	Dynamic Sparse Attention	
	Activations	
	SparQ Attention	
	Attention Heads	
Quantization	QLLM	
	Weight-Only Quantization	
	Weight-Activation Quantization	
	SpQR	
	QUIK	
	QuantEase	
	Activation-Aware Weight Quantization (AWQ)	
	Quantized LoRA (QLoRA)	
	Omnidirectionally Calibrated Quantization (OmniQuant)	Learnable Weight Clipping (LWC)
		Learnable Equivalent Transformation (LET)
	Round-to-Nearest (RTN)	
	QoQ	
	BitsandBytes (BnB)	

	Generative Pre-Trained Transformer Quantization (GPTQ)	
	Spinquant	
	BitDelta	
	GPT Vector Quantization (GPTVQ)	
	K-Quant	
KV Cache Compression	SnapKV	
	KIVI	
	MiniCache	
	Cached KV Activation Quantization (KVQuant)	
	KV Cache Quantization	
	QoQ	
	Spinquant	
	PyramidKV	
	ZipCache	
	K-Quant	
White-Box Knowledge Distillation		
Activation Beacon		

Table 9: Combination Use Cases

Combination Categories	Subcategories
Model Merging	Model Averaging
	WIDEN
	DARE-TIES
Modality Synchronization	
Weight Mixing	Encoder/Decoder Combination
Mixture of Experts (MoE) Expansion	

Table 10: Modification Use Cases

Modification Categories	Subcategories	Secondary Subcategories	Tertiary Subcategories
[Efficiency]	Speculative Sampling	Speculative Decoding	
	[Training Efficiency]	FlashAttention	Ring-Attention
		Adan	
		Deepspeed	DeepSpeed-Ulysses
		Tensor Parallelism	
		Sequence Parallelism (SP)	Unified Sequence Parallelism (USP)
	Activations	Activation Functions	

[Performance Improvements]	Credibility-Aware Attention Modification (CrAM)		
	Vectors	Vector Injection	
	Attention		
	Probabilities		
	Modality Detection		
	Knowledge Editing		
	Activations	Activation Functions	
[Context Window Extension]	Sequence Parallelism (SP)	Unified Sequence Parallelism (USP)	
	Positional Encoding	Rotary Position Embedding (RoPE)	
		xPos	
		Truncated Basis	
		Power Scaling	
	Position Interpolation		
[Alignment]	Zero-Shot Toxic Language Suppression		
	Decoding Editing		
	Positional Encoding	Contextual Position Encoding (CPE)	
	Model Editing	Rank-One Model Editing	

		(ROME)	
		Layer-Specific Editing (LED)	
	[Interpretability]	Activations	

Appendix E. Full Closed Model Use Case Taxonomy

The full taxonomies of closed model use cases identified during our review and annotation of 143 research papers is displayed below. Each use case and subcategory was identified in at least one paper that we reviewed. However, to help categorize use cases, we created several categories within [brackets] that reflect the overall objective of the research, rather than specific techniques or processes. See CSET's GitHub page for the full dataset of closed model papers and use case annotations.

Table 11: Prompting Use Cases

Prompting Categories	Subcategories
[Performance Evaluation][A]	Performance Comparison
	Chain-of-Thought Reasoning
	Turing Experiment
	Agentic Behavior
	Tool Learning
[Synthetic Data Generation]	Instruction-Following Data
	Image Captioning
	RLAIF Data
	Reward Functions
	Code Generation
	Black-Box Knowledge Distillation
[LLM-Based Evaluation]	LLM-as-a-Judge
Self-Feedback	

[Safety/FAccT Evaluation]	Jailbreaking
	Prompt Injection
	Trustworthiness
	Confidence Elicitation
	Hallucination Induction
	Bias
Labeling	Data Classification
	Ranking
Text Completion	
Automated Prompting	
Tree of Thoughts (ToT)	
Retrieval Augmented Generation (RAG)	

[A] Performance evaluation encompasses capability evaluation.

Authors

Kyle Miller is a research analyst at CSET, where he works on the CyberAI Project focusing on the impact of open models, AI efficiency, and compute.

Mia Hoffmann is a research fellow at CSET, where she works on AI governance.

Rebecca Gelles is a senior data scientist at CSET, supporting the CyberAI Project.

Acknowledgments

The authors would like to thank John Bansemer, Catherine Aiken, Drew Lohn, James Dunham, Colin Shea-Blymyer, Kendrea Beers, and Kyle Crichton for their feedback and assistance, as well as Trey Roark for their research support in paper annotations. The authors are grateful for the valuable reviews provided by Yacine Jernite, Cailean Osborne, Ben Bucknall, and David Evan Harris. They also thank Matt Mahoney for fact-checking, Susan Moynihan for copy edits, Shelton Fitch for editorial support, and Jason Ly for graphic design.

Document Identifier: doi: 10.51593/20240007

Endnotes

[1] Executive Office of the President, *Winning the Race: America's AI Action Plan* (Washington DC: The White House, July 2025), https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf; National Telecommunications and Information Administration (NTIA), "NTIA Supports Open Models to Promote AI Innovation," Department of Commerce, July 30, 2024, https://www.ntia.gov/press-release/2024/ntia-supports-open-models-promote-ai-innovation; Sayash Kapoor, Rishi Bommasani, Kevin Klyman et al., "On the Societal Impact of Open Foundation Models," arXiv preprint arXiv:2403.07918 (2024), https://arxiv.org/abs/2403.07918; "Supporting Open Source and Open Science in the EU AI Act," Creative Commons, July 2023, https://creativecommons.org/wp-content/uploads/2023/07/SupportingOpenSourceAndOpenScienceInTheEUAIAct.pdf.

[2] Manuel Hoffmann, Frank Nagle, and Yanuo Zhou, "The Value of Open Source Software" (Harvard Business School, January 2024), https://www.hbs.edu/ris/Publication%20Files/24-038_51f8444f-502c-4139-8bf2-56eb4b65c58a.pdf.

[3] National Telecommunications and Information Administration (NTIA), *Dual-Use Foundation Models with Widely Available Model Weights Report* (Washington, DC: Department of Commerce, July 2024), https://www.ntia.gov/programs-and-initiatives/artificial-intelligence/open-model-weights-report;
Francisco Eiras, Aleksandar Petrov, Bertie Vidgen et al., "Risks and Opportunities of Open-Source Generative AI," arXiv preprint arXiv:2405.08597 (2024), https://arxiv.org/abs/2405.08597;
Eric Wallace, Olivia Watkins, Miles Wang et al., "Estimating Worst-Case Frontier Risks of Open-Weight LLMs," arXiv preprint arXiv:2508.03153v1 (2025), https://arxiv.org/html/2508.03153v1;
James Ball and Carl Miller, "Open Sourcing the AI Revolution: Framing the Debate on Open Source, Artificial Intelligence and Regulation," Demos, October 30, 2023, https://demos.co.uk/research/open-sourcing-the-ai-revolution-framing-the-debate-on-open-source-artificial-intelligence-and-regulation/;
Alex Engler, "How Open-Source Software Shapes AI Policy" (Brookings Institute, August 10, 2021), https://www.brookings.edu/articles/how-open-source-software-shapes-ai-policy/.

[4] Kyle Miller, "Open Foundation Models: Implications of Contemporary Artificial Intelligence" (Center for Security and Emerging Technology, March 12, 2024), https://cset.georgetown.edu/article/open-foundation-models-implications-of-contemporary-artificial-intelligence/.

[5] Cailean Osborne, Jennifer Ding, and Hannah Rose Kirk, "The AI Community Building the Future? A Quantitative Analysis of Development Activity on Hugging Face Hub," arXiv preprint arXiv:2405.13058 (2024), https://arxiv.org/abs/2405.13058.

[6] "How Companies Are Using Meta Llama," Meta, May 7, 2024, https://about.fb.com/news/2024/05/how-companies-are-using-meta-llama/.

[7] Benjamin Bucknall and Robert Trager, "Structured Access for Third-Party Research on Frontier AI Models: Investigating Researchers' Model Access Requirements" (Center for the Governance of AI, October 31, 2023), https://www.governance.ai/research-paper/structured-access-for-third-party-research-on-frontier-ai-models.

[8] Irene Solaiman, "The Gradient of Generative AI Release: Methods and Considerations," arXiv preprint arXiv:2302.04844 (2023), https://arxiv.org/abs/2302.04844.

[9] Stephen Casper Carson Ezell, Charlotte Siegmann et al., “Black-Box Access Is Insufficient for Rigorous AI Audits,” arXiv preprint arXiv:2401.14446 (2024), https://arxiv.org/abs/2401.14446.

[10] Matt White, Ibrahim Haddad, Cailean Osborne et al., “The Model Openness Framework: Promoting Completeness and Openness for Reproducibility, Transparency, and Usability in Artificial Intelligence,” arXiv preprint arXiv:2403.13784 (2024), https://arxiv.org/abs/2403.13784; Adrien Basdevant, Camille François, Victor Storchan et al., “Towards a Framework for Openness in Foundation Models: Proceedings from the Columbia Convening on Openness in Artificial Intelligence,” arXiv preprint arXiv:2405.15802 (2024), https://arxiv.org/abs/2405.15802.

[11] Margaret Mitchell, Simone Wu, Andrew Zaldivar et al., “Model Cards for Model Reporting,” arXiv preprint arXiv:1810.03993 (2019), https://arxiv.org/abs/1810.03993; Timnit Gebru, Jamie Morgenstern, Briana Vecchione et al., “Datasheets for Datasets,” arXiv preprint arXiv:1803.09010 (2021), https://arxiv.org/abs/1803.09010.

[12] “Version 1.0,” Open Source Initiative (OSI), accessed February 15, 2024, https://opensource.org/ai/open-source-ai-definition.

[13] “What Are Open Weights?,” Open Source Initiative (OSI), accessed February 15, 2024, https://opensource.org/ai/open-weights.

[14] White et al., “The Model Openness Framework”; Andreas Liesenfeld, Alianda Lopez, and Mark Dingemanse, “Opening up ChatGPT: Tracking Openness, Transparency, and Accountability in Instruction-Tuned Text Generators,” *CUI ’23: Proceedings of the 5th International Conference on Conversational User Interfaces* no. 47 (July 2023): 1–6 https://dl.acm.org/doi/10.1145/3571884.3604316.

[15] Benjamin Bucknall and Robert Trager, “Structured Access for Third-Party Research on Frontier AI Models: Investigating Researchers’ Model Access Requirements” (Centre for the Governance of AI, October 31, 2023), https://www.governance.ai/research-paper/structured-access-for-third-party-research-on-frontier-ai-models.

[16] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham et al., “Stealing Part of a Production Language Model,” arXiv preprint arXiv:2403.06634 (2024), https://arxiv.org/abs/2403.06634; Jiacheng Liang, Ren Pang, Changjiang Li, and Ti Wang, “Model Extraction Attacks Revisited,” arXiv preprint arXiv:2312.05386 (2023), https://arxiv.org/abs/2312.05386.

[17] “Open LLM Leaderboard,” Hugging Face, accessed September 15, 2025, https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/.

[18] Alec Radford, Jeffrey Wu, Rewon Child et al., “Language Models Are Unsupervised Multitask Learners,” OpenAI, 2020, https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[19] Hugo Touvron, Louis Martin, Kevin Stone et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” Meta, July 18, 2023, https://ai.meta.com/research/publications/llama-2-open-foundation-and-fine-tuned-chat-models/.

[20] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi et al., "The Falcon Series of Open Language Models," arXiv preprint arXiv:2311.16867 (2023), https://arxiv.org/abs/2311.16867.

[21] Jinze Bai, Shuai Bai, Yunfei Chu et al., "Qwen Technical Report," arXiv preprint arXiv:2309.16609 (2023), https://arxiv.org/abs/2309.16609.

[22] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch et al., "Mistral 7B," arXiv preprint arXiv:2310.06825 (2023), https://arxiv.org/abs/2310.06825.

[23] "Introducing Qwen1.5," Qwen, February 4, 2024, https://qwenlm.github.io/blog/qwen1.5/.

[24] Marah Abdin, Jyoti Aneja, Hany Awadalla et al., "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone," arXiv preprint arXiv:2404.14219 (2024), https://arxiv.org/abs/2404.14219.

[25] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri et al., "The Llama 3 Herd of Models," arXiv preprint arXiv:2407.21783 (2024), https://arxiv.org/abs/2407.21783.

[26] Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid, "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities," arXiv preprint arXiv:2408.13296 (2024), https://arxiv.org/abs/2408.13296.

[27] Balavadhani Parthasarathy, "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs."

[28] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell, "Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks," arXiv preprint arXiv:2207.13243 (2023), https://arxiv.org/abs/2207.13243; Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà, "A Primer on the Inner Workings of Transformer-Based Language Models," arXiv preprint arXiv:2405.00208 (2024), https://arxiv.org/abs/2405.00208.

[29] Xunyu Zhu, Jian Li, Yong Liu et al., "A Survey on Model Compression for Large Language Models," arXiv preprint arXiv:2308.07633 (2024), https://arxiv.org/abs/2308.07633.

[30] Zixuan Zhou, Xuefei Ning, Ke Hong et al., "A Survey on Efficient Inference for Large Language Models," arXiv preprint arXiv:2404.14294 (2024), https://arxiv.org/abs/2404.14294; Daya Khudia and Vitaliy Chiley, "Benchmarking Large Language Models on NVIDIA H100 GPUs with CoreWeave (Part 1)," DataBricks (blog), April 27, 2023, https://www.databricks.com/blog/coreweave-nvidia-h100-part-1; Jaydeep Golla, "Benchmarking NVIDIA GPU Throughput for LLMs and Understanding GPU Configuration Choices in the AI Space," Dell Technologies, September 9, 2024, https://infohub.delltechnologies.com/de-de/p/benchmarking-nvidia-gpu-throughput-for-llms-and-understanding-gpu-configuration-choices-in-the-ai-space/.

[31] Yiming Cui, Ziqing Yang, and Xin Yao, "Efficient and Effective Text Encoding for Chinese Llama and Alpaca," arXiv preprint arXiv:2304.08177 (2024), https://arxiv.org/abs/2304.08177.

[32] Longhui Yu, Weisen Jiang, Han Shi et al., "MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models," arXiv preprint arXiv:2309.12284 (2024), https://arxiv.org/abs/2309.12284.

[33] Dan Zhang, Ziniu Hu, Sining Zhoubian et al., "SciInstruct: a Self-Reflective Instruction Annotated Dataset for Training Scientific Language Models," arXiv preprint arXiv:2401.07950v3 (2024), https://arxiv.org/abs/2401.07950v3.

[34] Yu Meng, Mengzhou Xia, and Danqi Chen, "SimPO: Simple Preference Optimization with a Reference-Free Reward," arXiv preprint arXiv:2405.14734v3 (2024), https://arxiv.org/abs/2405.14734v3.

[35] Botao Yu, Frazier N. Baker, Ziqi Chen et al., "LlaSMol: Advancing Large Language Models for Chemistry with a Large-Scale, Comprehensive, High-Quality Instruction Tuning Dataset," arXiv preprint arXiv:2402.09391 (2024), https://arxiv.org/abs/2402.09391.

[36] Xiang Lisa Li and Percy Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Volume 1 (January 2021): 4582–4597, https://aclanthology.org/2021.acl-long.353/.

[37] Asier Gutiérrez-Fandiño, Jordi Armemgol-Estapé, Marc Pàmies et al., "MarIA: Spanish Language Models," arXiv preperint arXiv:2107.07253 (2022), https://arxiv.org/abs/2107.07253; Cui et al., "Efficient and Effective Text Encoding for Chinese Llama and Alpaca."

[38] Zeming Chen, Alejandro Hernández Cano, Angelika Romanou et al., "MEDITRON-70B: Scaling Medical Pretraining for Large Language Models," arXiv preprint arXiv:2311.16079 (2023), https://arxiv.org/abs/2311.16079.

[39] Yizhen Luo, Jiahuan Zhang, Siqi Fan et al., "BioMedGPT: Open Multimodal Generative Pre-Trained Transformer for Biomedicine," arXiv preprint arXiv:2308.09442 (2023), https://arxiv.org/abs/2308.09442.

[40] Lean Wang, Lei Li, Damai Dai et al., "Label Words are Anchors: An Information Flow Perspective for Understanding In-Context Learning," arXiv preprint arXiv:2305.14160 (2023), https://arxiv.org/abs/2305.14160.

[41] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou, "DataInf: Efficiently Estimating Data Influence in LoRA-Tuned LLMs and Diffusion Models," arXiv preprint arXiv:2310.00902 (2024), https://arxiv.org/abs/2310.00902.

[42] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg, "Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space," arXiv preprint arXiv:2203.14680 (2022), https://arxiv.org/abs/2203.14680.

[43] Somnath Banerjee, Sayan Layek, Rima Hazra, and Animesh Mukherjee, "How (Un)Ethical Are Instruction-Centric Responses of LLMs? Unveiling the Vulnerabilities of Safety Guardrails to Harmful Queries," arXiv preprint arXiv:2402.15302 (2024), https://arxiv.org/abs/2402.15302.

[44] Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons, "Image Hijacks: Adversarial Images Can Control Generative Models at Runtime," arXiv preprint arXiv:2309.00236 (2024), https://arxiv.org/abs/2309.00236.

[45] Teven Le Scao, Angela Fan, Christopher Akiki et al., "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model," arXiv preprint arXiv:2211.05100 (2023), https://arxiv.org/abs/2211.05100; Dirk Groeneveld, Iz Beltagy, Pete Walsh et al., "OLMo: Accelerating the Science of Language Models," arXiv preprint arXiv:2402.00838 (2024), https://arxiv.org/abs/2402.00838.

[46] Akide Liu, Jing Liu, Zizheng Pan et al., "MiniCache: KV Cache Compression in Depth Dimension for Large Language Models," arXiv preprint arXiv:2405.14366 (2024), https://arxiv.org/abs/2405.14366.

[47] Zirui Liu, Jiayi Yuan, Hongye Jin et al., "KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache," arXiv preprint arXiv:2402.02750 (2024), https://arxiv.org/abs/2402.02750.

[48] Shiyao Li, Xuefei Ning, Luning Wang et al., "Evaluating Quantized Large Language Models," arXiv preprint arXiv:2402.18158 (2024), https://arxiv.org/abs/2402.18158.

[49] Wei Huang, Xudong Ma, Haotong Qin et al., "How Good Are Low-Bit Quantized LLaMA3 Models? An Empirical Study," arXiv preprint arXiv:2404.14047v1 (2024), https://arxiv.org/abs/2404.14047v1.

[50] Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi et al., "LLM Pruning and Distillation in Practice: The Minitron Approach," arXiv preprint arXiv:2408.11796 (2024), https://arxiv.org/abs/2408.11796.

[51] Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento et al., "SliceGPT: Compress Large Language Models by Deleting Rows and Columns," arXiv preprint arXiv:2401.15024 (2024), https://arxiv.org/abs/2401.15024.

[52] Omar Alfarghaly, Rana Khaled, Abeer Elkorany et al., "Automated Radiology Report Generation Using Conditioned Transformers," *Informatics in Medicine Unlocked* 24 (2021): 100557, https://www.sciencedirect.com/science/article/pii/S2352914821000472.

[53] Yanis Labrak, Adrien Bazoge, Emmanuel Morin et al., "BioMistral: A Collection of Open-Source Pretrained Large Language Models for Medical Domains," arXiv preprint arXiv:2402.10373 (2024), https://arxiv.org/abs/2402.10373.

[54] Juexiao Zhou, Xiaonan He, Liyuan Sun et al., "Pre-trained Multimodal Large Language Model Enhances Dermatological Diagnosis Using SkinGPT-4," *Nature Communications* 15.1 (2024): 5649, https://www.nature.com/articles/s41467-024-50043-3.

[55] Zhou et al., "Pre-trained Multimodal Large Language Model Enhances Dermatological Diagnosis."

[56] Zhou et al., "A Survey on Efficient Inference for Large Language Models"; Khudia and Chiley, "Benchmarking Large Language Models on NVIDIA H100 GPUs with CoreWeave (Part 1)"; Golla, "Benchmarking NVIDIA GPU Throughput for LLMs."

[57] Yuan Yao, Tianyu Yu, Ao Zhang et al., "MiniCPM-V: A GPT-4V Level MLLM on Your Phone," arXiv preprint arXiv:2408.01800 (2024), https://arxiv.org/abs/2408.01800.

[58] Daliang Xu, Hao Zhang, Liming Yang et al., "Fast On-device LLM Inference With NPUs," arXiv preprint arXiv:2407.05858 (2024), https://arxiv.org/abs/2407.05858.

[59] Yutao Zhu, Peitian Zhang, Chenghao Zhang et al., "INTERS: Unlocking the Power of Large Language Models in Search With Instruction Tuning," arXiv preprint arXiv:2401.06532 (2024), https://arxiv.org/abs/2401.06532.

[60] Tri Dao, Daniel Y. Fu, Stefano Ermon et al., "FlashAttention: Fast and Memory-Efficient Exact Attention With IO-Awareness," arXiv preprint arXiv:2205.14135 (2022), https://arxiv.org/abs/2205.14135.

[61] Wei Zhao, Zhe Li, Yige Li et al., "Defending Large Language Models Against Jailbreak Attacks via Layer-specific Editing," arXiv preprint arXiv:2405.18166 (2024), https://arxiv.org/abs/2405.18166.

[62] Wenhan Xiong, Jingyu Liu, Igor Molybog et al., "Effective Long-Context Scaling of Foundation Models," arXiv preprint arXiv:2309.16039 (2023), https://arxiv.org/abs/2309.16039.

[63] Peng Xu, Wei Ping, Xianchao Wu et al., "Retrieval Meets Long Context Large Language Models," arXiv preprint arXiv:2310.03025 (2024), https://arxiv.org/abs/2310.03025.

[64] "Developer Quickstart: Take Your First Steps With the OpenAI API," OpenAI, https://platform.openai.com/docs/quickstart?api-mode=responses; "OpenAI Platform," OpenAI, https://platform.openai.com/docs/api-reference/introduction.

[65] Eric Wu, Kevin Wu, and James Zou, "FineTuneBench: How Well Do Commercial Fine-Tuning APIs Infuse Knowledge into LLMs?," arXiv preprint arXiv:2411.05059v2 (2024), https://arxiv.org/abs/2411.05059v2.

[66] See Appendix B for more details; "Model Optimization," OpenAI, accessed September 24, 2025, https://platform.openai.com/docs/guides/model-optimization; "Introducing Improvements to the Fine-Tuning API and Expanding Our Custom Models Program," OpenAI, April 2024, https://openai.com/index/introducing-improvements-to-the-fine-tuning-api-and-expanding-our-custom-models-program/.

[67] Jiawei Liu, Chunqui Steven Xia, Yuyao Wang, and Lingming Zhang, "Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation," *Advances in Neural Information Processing Systems* 36 (2023): 21558–21572, https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html; Simon Frieder, Luca Pinchetti, Chevalier et al., "Mathematical Capabilities of ChatGPT," *Advances in Neural Information Processing Systems* 36 (2023): 27699–27744, https://proceedings.neurips.cc/paper_files/paper/2023/hash/58168e8a92994655d6da3939e7cc0918-Abstract-Datasets_and_Benchmarks.html; Harsha Nori, Nicholas King, Scott Mayer McKinney et al., "Capabilities of GPT-4 on Medical Challenge Problems," arXiv preprint arXiv:2303.13375 (2023), https://arxiv.org/abs/2303.13375.

[68] Liu et al., "Is Your Code Generated by ChatGPT Really Correct?"

[69] Frieder et al., "Mathematical Capabilities of ChatGPT."

[70] Ruibo Liu, Jerry Wei, Fangyu Liu et al., "Best Practices and Lessons Learned on Synthetic Data," arXiv preprint arXiv:2404.07503 (2024), https://arxiv.org/abs/2404.07503.

[71] Ganqu Cui, Lifan Yuan, Ning Ding et al., "UltraFeedback: Boosting Language Models With Scaled AI Feedback," arXiv preprint arXiv:2310.01377 (2024), https://arxiv.org/abs/2310.01377.

[72] Tim G. J. Rudner and Helen Toner, "Key Concepts in AI Safety: Reliable Uncertainty Quantification in Machine Learning" (Center for Security and Emerging Technology, June 2024), https://cset.georgetown.edu/publication/key-concepts-in-ai-safety-reliable-uncertainty-quantification-in-machine-learning/.

[73] Miao Xiong, Zhiyuan Hu, Xinyang Lu et al., "Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs," arXiv preprint arXiv:2306.13063 (2024), https://arxiv.org/abs/2306.13063.

[74] Tianrui Guan, Fuxiao Liu, Xiyang Wu et al., "HallusionBench: An Advanced Diagnostic Suite for Entangled Language Hallucination and Visual Illusion in Large Vision-Language Models," arXiv preprint arXiv:2310.14566 (2024), https://arxiv.org/abs/2310.14566.

[75] Michael Bommarito II and Daniel Martin Katz, "GPT Takes the Bar Exam," arXiv preprint arXiv:2212.14402 (2022), https://arxiv.org/abs/2212.14402.

[76] Xiangyu Qi, Yi Zeng, Tinghao Xie et al., "Fine-Tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To!," arXiv preprint arXiv:2310.03693 (2023), https://arxiv.org/abs/2310.03693.

[77] Lukas Berglund, Meg Tong, Max Kaufmann et al., "The Reversal Curse: LLMs Trained on 'A is B' Fail to Learn 'B is A'," arXiv preprint arXiv:2309.12288 (2024), https://arxiv.org/abs/2309.12288.

[78] Jiageng Mao, Yuxi Qian, Junjie Ye et al., "GPT-Driver: Learning to Drive with GPT," arXiv preprint arXiv:2310.01415 (2023), https://arxiv.org/abs/2310.01415.

[79] "The 2025 AI Index Report" (Stanford Institute for Human-Centered Artificial Intelligence (HAI), April 2025), https://hai.stanford.edu/ai-index/2025-ai-index-report.

[80] Anita Bandrowski and Amanda French, "Understanding RRID and ROR for Facilities," Research Resource Identification (blog), November 26, 2024, https://www.rrids.org/news/2024/11/26/understanding-rrid-and-ror-for-facilities.

[81] HAI, "The 2025 AI Index Report."

[82] Kyle Miller and John Bansemer, "DeepSeek's Release of an Open-Weight Frontier AI Model" (International Institute for Strategic Studies (IISS), April 2025), https://cset.georgetown.edu/article/deepseeks-release-of-an-open-weight-frontier-ai-model/; Daya Guo, Dejian Yang, Haowei Zhang et al., "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," arXiv preprint arXiv:2501.12948 (2025), https://arxiv.org/abs/2501.12948; An Yang, Anfeng Li, Baosong Yang et al., "Qwen3 Technical Report," arXiv preprint arXiv:2505.09388 (2025), https://arxiv.org/abs/2505.09388.

[83] National Telecommunications and Information Administration (NTIA), *Dual-Use Foundation Models With Widely Available Model Weights Report* (Washington, DC: Department of Commerce, July 30, 2024), https://www.ntia.gov/programs-and-initiatives/artificial-intelligence/open-model-weights-report; Kapoor et al., "On the Societal Impact of Open Foundation Models"; Eiras et al., "Risks and Opportunities of Open-Source Generative AI"; Wallace et al., "Estimating Worst-Case Frontier Risks of Open-Weight LLMs"; Kyle Miller, "Open Foundation Models."

[84] Wu et al., "FineTuneBench: How Well Do Commercial Fine-Tuning APIs Infuse Knowledge into LLMs?"

[85] OpenAI, "Fine-Tuning"; "Customize a Model With Fine-Tuning," Microsoft Azure, July 2025, https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/fine-tuning?tabs=azure-openai&pivots=programming-language-studio#choose-your-training-data; OpenAI, "Introducing Improvements to the Fine-tuning API."

[86] "OpenAI o1 and New Tools for Developers," OpenAI, 2024, https://openai.com/index/o1-and-new-tools-for-developers/; Karan Sharma, "Fine-Tuning Updates: Reinforcement Fine-Tuning Now Available + GPT-4.1 Nano Fine-Tuning," OpenAI, May 2025, https://community.openai.com/t/fine-tuning-updates-reinforcement-fine-tuning-now-available-gpt-4-1-nano-fine-tuning/1255539.

[87] "Fine-Tuning," OpenAI, December 2023, https://archive.ph/fAtEb.

[88] "ModelScope," ModelScope, https://modelscope.cn/home.

[89] "It's been a wild ride, folks :) (end of the Open LLM Leaderboard)," Hugging Face, March 13, 2025, https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard/discussions/1135.