The Columbia Convening on Openness and AI

moz://a | SIPA | IGP Institute of Global Politics

Technical Readout

27 March, 2024

—

Nik Marda

Review Contributors:
Stefano Maffulli, Deval Pandya,
Irene Solaiman, and Victor Storchan

Table of Contents

—

On February 29, 2024, Mozilla and the Columbia Institute of Global Politics co-hosted a gathering of over 40 experts and stakeholders in AI to explore the concept of "openness" in the AI era. This diverse group included representatives from leading AI startups, companies, non-profit AI labs, and civil society organizations.

The convening aimed to help develop a better framework for what "open" means in the AI era, drawing inspiration from the pivotal role that open source software has played in technology, cybersecurity, and economic growth over the years. This work is particularly timely: as AI development shifts from research labs into customer-facing products, there has been increased usage of proprietary AI models — raising concerns about negative impacts for innovation, competition, and accountability.

This brief attempts to distill the technical dimensions of openness in AI that participants focused on during the convening. It emphasizes how openness in AI — defined by some participants as the broad public availability of key artifacts and documentation from AI systems — can help AI developers and deployers. This brief summarizes different conceptual approaches for openness, explores a framework for approaching openness across the AI stack, and provides an appendix that goes deeper into details about openness in different components of the AI stack. Finally, this brief also provides a summary of areas suggested for future exploration.

Introduction

Understanding openness in AI is a complex task. AI models are not just code; they are trained on massive datasets, deployed on intricate computing infrastructure, and accessed through diverse interfaces and modalities. With traditional software, there was a very clear separation between the code one wrote, the compiler one used, the binary it produced, and what license they had. However, for AI models, many components collectively influence the functioning of the system, including the algorithms, code, hardware, and datasets used for training and testing. The very notion of modifying the source code (which is important in the definition of open source) becomes fuzzy. For example, should the training dataset, the model weights, or something else be considered independently or collectively as the source code for the AI model that has been trained? In addition, the presumed benefits of one artifact over another are not always true — for example, if a model has been trained with some fraudulent external data, it is often

challenging and an ongoing research area to remove that piece of information from informing the model.

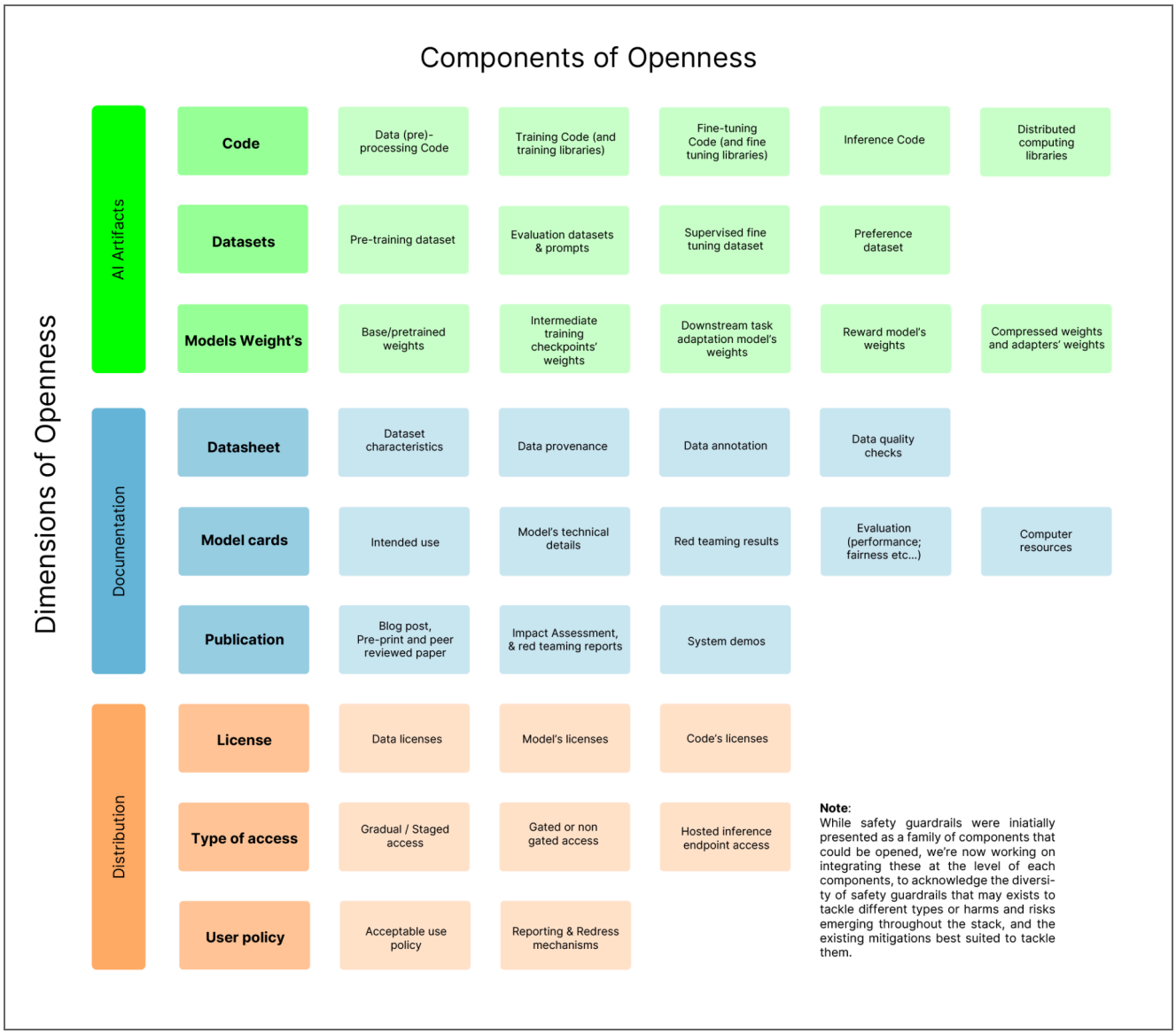
Given these challenges, many scholars and developers have proposed approaches for characterizing "openness" for AI systems. Broadly, participants summarized these budding approaches to openness as falling into three categories:

- **Gradient / Spectrum:** These approaches characterize openness on a gradient with different levels of openness. Different AI systems can be "more" or "less" open under this approach. Examples include Irene Solaiman's paper on gradients of generative AI releases and the Digital Public Goods Alliance and UNICEF's approach to tracking openness across components of an AI system.
- **Criteria Scoring:** These approaches provide a score for openness based on different attributes about the AI system. For example, different AI systems can be scored as "30/50" or "7 out of 9" on a metric that aims to be a proxy for openness. Examples include Stanford's index for foundation model transparency and Andreas Liesenfeld et al.'s paper on opening up ChatGPT.
- **Binary:** These approaches characterize openness as a binary. Different AI systems are either "open" or "closed" based on whether they meet a certain set of criteria. Examples include the Open Source Initiative's ongoing work to define open source AI.

Framework for Openness

Components of Openness

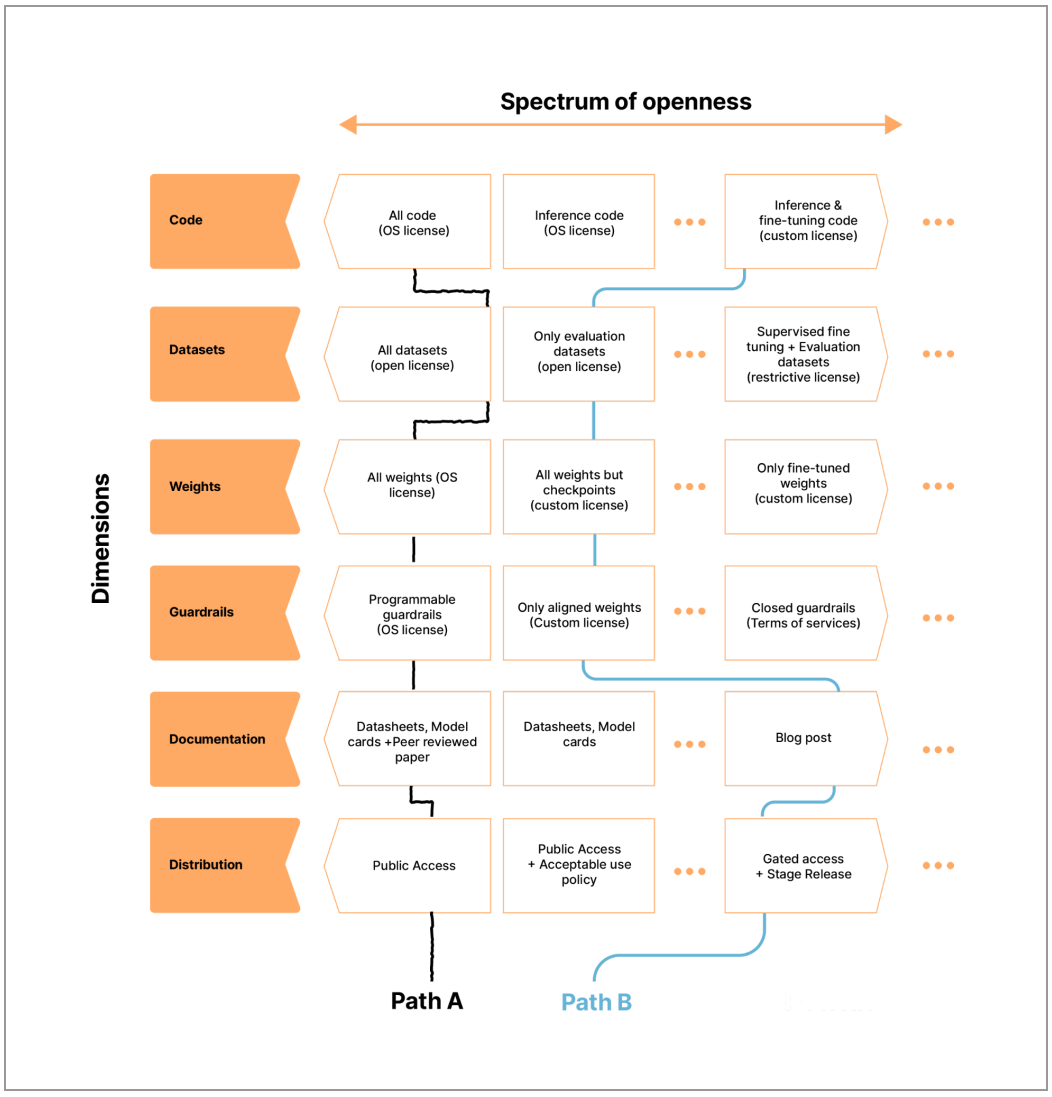
Participants discussed the different components of the foundation model stack that could be more or less open. Some of this discussion focused on the attributes highlighted in the image below, and explained further in the appendix of this document.



This diagram was prepared for the Columbia Convening and is in the process of being revised based on feedback and discussion at the workshop. It will be updated and republished at http://mozilla.org/research/cc in the coming weeks.

Broadly, the diagram shows how the foundation model stack is described by its different dimensions (AI artifacts, documentation, and distribution). A dimension is further described by its own components and subcomponents (i.e., more granular AI building blocks that constitute the dimension). These components can be released or not released, and when being released, the components can have varying levels of public access. The many release options roughly form a gradient of openness for the particular component, although some forms of openness may be more or less useful for pursuing certain goals.

The image on the next page reflects how some participants approached the gradient of different "levels" of openness that could be chosen for each part of the AI stack. The "path" intends to link different design choices together, in an attempt to illustrate the interdependencies between the options; some benefits of openness can only be achieved when certain design choices are made in tandem across the AI stack (e.g., reproducibility of research might require certain levels of openness on code, data, and weights in tandem).



This diagram was prepared for the Columbia Convening and is in the process of being revised based on feedback and discussion at the workshop. It will be updated and republished at http://mozilla.org/research/cc in the coming weeks.

Open Questions

Participants highlighted a number of different topics for future exploration. Here, we summarize a handful of them that may be fruitful for inquiry in the near future.

Incorporating additional attributes into the framework
Participants noted the potential relevance of openness to other aspects of AI systems, especially extending beyond AI models. There was discussion about incorporating attributes such as hardware, firmware, versioning, developer organization makeup, training processes, prohibited users, takedown policies, and societal and environmental impact into the frameworks. Next steps would include building upon this framework to include more attributes that make this framework useful to developers.

Mapping system goals to specific aspects of openness
Participants noted that developers and deployers would find it useful to have a guide to convert AI system goals into recommendations about what components of their AI stack should be open and how. This would help illustrate what design choices should be made in tandem to unlock the specific desired benefits of openness.

Clarifying resource constraints and incentives
Participants noted the need to carefully incorporate and delineate which aspects of openness are dependent on increased resourcing and legal concerns to be successful. For example, openness around data and compute may actually be most inhibited by those resources being expensive and difficult to acquire, or legally sensitive, rather than it being more of a design choice for developers.

Future-proofing for the next generation of AI systems
Participants noted the need to future-proof this framework for AI stacks in the future. Additional work could be done to clarify how AI stacks might change in the coming years based on current trends in AI R&D, and then ensure that an approach to AI and openness can either withstand these changes, or can easily be adapted as the technology changes.

Appendix: Components of the AI Stack

This section provides a list of key components in the AI stack, and describes how more openness in each component can help advance system and/or societal goals. For the purposes of this appendix, the examples are focused on highlighting benefits of additional openness; this does not mean that there are no risks associated with more openness in some of these components. This section is an initial draft, and will benefit from future revisions to catch potential errors and gaps.

AI Artifacts

CODE

- **Data (pre)-processing code:** Code that is applied to the dataset before it is passed to the foundation model, such as tokenizers, data deduplication code, data quality checks, and various filters. More openness can enable better assessments of the quality of data; provide information about data deduplication, data quality, and fairness; and improve our understanding of how data processing influences different characteristics such as performance and privacy.
- **Pre-training code:** Code that is used to train the foundation model, which defines the training cycles. It includes the model's architecture, training loss, hyperparameters, and configurations. More openness can enable more people to participate in retraining the model on other datasets and to understand how different training strategies influence different training objectives (e.g., assessing how different training strategies impact memorization, and what one should do at training time to build a more robust model)
- **Training libraries:** Pre-built scripts that are imported by the developer and used as packages during the training of the foundation model. Examples include Transformers, Keras, GPT-NeoX, and llm-foundry. More openness can enable more people to participate in retraining the model on other datasets, and is vital for open science and open knowledge approaches (e.g., reproducibility).
- **Fine-tuning code and libraries:** Code that is used to fine-tune a model on a particular domain or application. This can include methods like SFT, RLHF, and DPO, and fine-tuning libraries like alignment-handbook, LoRA and PEFT. More openness can lower costs for a broader community of researchers to build and reuse more efficient fine-tuned models, produce thriving ecosystems of task-specific models built on top of pretrained models, and enable safety researchers to scientifically advance risk mitigation techniques (e.g., alignment methods, preference sampling training methods).

- **Inference code:** Code that enables the models to serve results to the users. This includes prompting design, inference optimization, early predictions, decoding methods, and search. More openness can allow developers to better integrate the model into downstream applications, and open source frameworks (e.g., vLLM, FlashAttention, BitsandBytes, PowerInfer, and techniques like speculative inference) have helped make inference cheaper and models more efficient.
- **Distributed computing libraries:** Libraries that contain the code that is used for training the models on distributed clusters (multi GPUs, multi nodes). This includes SOTA training techniques such as distributed training, mixed precision, and gradient accumulation, and libraries such as Accelerate, DeepSpeed, and PyTorch. More openness via increased access to base distributed training libraries can enable more efficient training on different clusters, and empirically assessing model and data parallel training techniques can enable one to better understand memory footprint and computational efficiency.
- **Inference and cloud infrastructure frameworks:** The software infrastructure that is used during inference. This includes approaches such as scalable serving, multi-cloud optimization, and batching, and frameworks and tools such as SkyPilot, Ray Serve, Kubernetes, vLLM, and Triton. More openness can enable better reproducibility and enables various degrees of interoperability and portability of the code on different types of hardware and cloud computing environments.

DATASETS

- **Pre-training datasets:** Datasets, often very large datasets that are snapshots of the web, on which models are trained. Examples include Pile, C4, ROOTS, RefinedWeb, Dolma. More openness can enable one to study the biases, fairness, toxicity of the data on which the model is trained; open datasets such as CommonCrawl which plays a central role in training of LLMs today, enabling researchers to document hate speech and low quality content that is fed to the models. Privacy-preserving approaches can be particularly important here, in order to properly balance considerations around security and confidentiality (e.g, in health).
- **Supervised fine-tuning datasets:** Smaller, context-specific datasets than are used to fine-tune the model on specific tasks and domains, such as instruction-tuning datasets and dialogue datasets. More openness can allow one to understand how the model was specialized for a specialized downstream task (e.g. dialogue, coding assistant).
- **Preference datasets:** Datasets that compare & classify model outputs on conversation text data. This can include RLHF datasets and human preference datasets, which play a central role in aligning models on user preferences, content moderation, style, etc. More openness can enable better language and image

modeling to understand how the model was aligned, and more transparency on content moderation and safety approaches.

- **Evaluation datasets:** Test datasets that are used to evaluate the models' performance offline and derive various metrics on top of them. More openness can increase the transparency of data on which the model was evaluated (which can be tricky to do reliably since base pre-trained models are general purpose) and enable auditing by third-parties including end users. More openness can also enable researchers to identify and fix gaps in evaluation procedures, benchmarks, and scenarios.
- **Evaluation prompts:** Pieces of text that encapsulate samples of text of evaluation datasets with additional guidelines or context (e.g., "You are an expert in legal, please answer the following question + [Insert the evaluation sample]"). More openness can enable better reproduction of evaluation results, and fairer comparison across models as spacing, punctuation, and casing can currently have an important impact on LLM performance.

MODEL WEIGHTS

- **Pretrained weights:** Numerical values of all the parameters of models that have been tuned during the self-supervised pre-training phase. Examples include Mistral-7B, llama-2-70B, and Pythia-12B. More openness can enable developers to build an ecosystem of developments on top of one model (e.g., "the stable diffusion moment"), repurpose models for different tasks, train models in ways that are more attuned to linguistic and cultural attributes around the world, and enable more auditability, scrutiny, privacy, and transparency. It can also foster entire new research avenues like mechanistic interpretability that require access to the internals of the model, and it can help with comparing architectures' efficiencies to improve the scientific understanding on the generalization of models.
- **Intermediate training checkpoints' weights:** Intermediate values that the parameters of the model can take during the training phase; at regular timesteps, those values can be saved as intermediate models. More openness can enable the community of researchers to pursue scientific research on scaling (and inverse scaling) laws that are critical to understand the training dynamic of the technology.
- **Downstream task adaptation model's weights:** Weights of fine tuned models that have been specialized for a specific task or domain. This includes instruction fine-tuning, alignment, supervised fine-tuning, etc. More openness can enable developers to seamlessly integrate highly-capable models into user-facing systems, enable third-party auditors to provide more scrutiny to these models, and enable researchers to study the safety costs associated with such custom fine-tuning. Releasing aligned models enables researchers to better evaluate how robust and

safe alignment methods are currently, and to pursue new research avenues like research on universal jailbreak backdoors on aligned models.

- **Compressed and adapter weights:** Weights that result from the compression of a model — the process of making a model smaller while not compromising on performance. This includes processes like pruning and quantization, and adapter weights like LoRA and QLoRA. More openness can enable the community to better study inverse scaling laws with more powerful small models, enable quick model adaptation (as sharing LoRA weights is very convenient) and open new research avenues in models' efficiency.
- **Reward model's weights:** Weights resulting from models that are learned from preference datasets to enable automatic classification or to rank models' text output according to users' preferences. These models are further used to fine-tune a foundation model based on users' preferences. More openness can enable sharing of models that have been trained on human or AI preferences, which can reduce the barrier to using safer models in practice. It can also help other AI models follow instructions better, and those reward models can also be further fine-tuned by developers for custom needs or for further rejection sampling training.

GUARDRAILS

- **Aligned weights:** Weights that have been further trained with preference data as a safety mitigation technique. In particular, aligned weights are weights of a base model that have been fine-tuned with a dataset that encapsulates content moderation examples or examples of how to answer or refuse to answer to various questions. More openness can enable a community of researchers to further test jailbreaking techniques and establish benchmarks to document the robustness of safety techniques.
- **Programmable guardrails:** Explicit handwritten rules and filters (as opposed to implicit learnt rules) that sit on top of a model for content moderation purposes. Examples include NeMo-Guardrails, Guidance, and generation-guided tools like Outlines. More openness can enable developers to use transparent and auditable content moderation techniques, as opposed to closed black box APIs to categorize hate speech or prevent AI systems from answering certain topics or prompts.
- **Safeguard models:** A safeguard model inspects output (or input) of base foundation models to further decide whether it is compliant with various rules and content moderation policies. Examples include llama-guard and fine-tuned lmsys-chat. More openness can enable red teamers, researchers, and the general public to evaluate the robustness of content moderation systems, identify potential vulnerabilities, and participate in advancing AI research and safety.
- **System prompts:** Pieces of text that are added to user prompts and that explicitly give content moderation rules to the models. Access to prompt libraries like

PromptSource better help sharing these prompts. More openness can enable people to collectively determine which prompts are more efficient to guide the model and increase safety while not compromising on utility.

Documentation

DATASHEETS

- **Dataset characteristics:** Metadata informing content of a dataset, such as task distribution, language distribution, topics, format, etc. More openness can enable developers to get detailed information relevant to understanding data representation and data composition in order to keep control on the nature of the data on which a model has been trained.
- **Data provenance:** Metadata relative to the provenance and history of the dataset like versioning, attribution, text source, citation and download counts. More openness can enable developers to keep track of the history, versions and sources of the datasets especially when datasets are iteratively aggregated and merged overtime.
- **Data annotation, third-party annotators & labelers:** General guidelines that have been given to labelers to capture designers' intent in shaping an optimization (e.g., when crafting the data instructions / filtering practices for iterative/online RLHF). More openness can enable developers to understand possible preference biases, how labels have been assigned, as well as potential outsourcing labor problems and discrimination.
- **Data quality checks and qualitative analyses:** A series of tests informing about the quality of the dataset (e.g. semantic and linguistic checks). More openness can inform developers about the potential downstream vulnerabilities of the models.

MODEL CARDS

- **Intended use:** Attributes like model objectives and out of scope use cases that inform downstream developers about the expected use of the model when originally designed by the model's provider. More openness can enable downstream developers to understand the design choices and the benefits and risks of using the models for various use cases and contexts.
- **Model's technical details:** Attributes about the design of a model, such as model architecture, hyperparameters, and tradeoffs in design choices. More openness can enable developers to get contextual information about when to use the models and to get a better sense of the model providers' intentions.

- **Compute resources:** Information on compute resources that allow a broad spectrum of stakeholders to understand what amount of compute was used as it relates to hardware/software efficiency, carbon footprint of AI, and portability. Examples include energy efficiency, GPU/hardware specification, amount of compute and time required to run inference on fixed hardware for a specific task. More openness can make it easier to design more efficient models and track the carbon footprint of AI, both from training and inference.
- **Evaluation:** Documentation of evaluation results and protocols, which can enable people to understand the capabilities and limitations of a model. This can include qualitative and quantitative evaluation protocols, simulation environments, metrics and results, prompting techniques, evaluation tasks, and benchmarks. More openness can enable the community to challenge existing state-of-the-art models while conducting fair comparison across models across different metrics (e.g., fairness, robustness, efficiency) and enable developers to understand strengths and vulnerabilities of the produced models.
- **Red teaming results:** Reports about surfaced unknown vulnerabilities from red-teaming AI systems. More openness can help downstream developers better understand the risks associated with the deployed model, and it can help increase foresight into how the model could cause harm when used by the general public or by malicious actors.

PUBLICATIONS

- **Pre-print and peer reviewed paper:** A paper that is accessible online before it has gone through the peer-review process of a scientific conference or a journal. More openness (i.e., more pre-prints and peer-reviewed papers) is a base practice to advance the scientific discipline of AI, and in the recent history of AI, scientific papers have been published at open access conferences, journals, and websites like ArXiv and OpenReview.
- **Impact Assessment & red teaming reports:** Impact assessments are structured processes to imagine, document and quantify the possible impacts of a proposed AI system on various stakeholders. More openness can increase public awareness and transparency about document downstream use and risks of a model as well as the societal implications of the release. The question arises on the possibility for open foundation models, that are general purpose by design and for which the distribution of users might not be fully known. (See, e.g., the AI-Risk Management Standards Profile for general purpose AI systems (GPAIS) and FMs.)
- **System demos:** Live user interfaces where users can play with a given AI prototypes; they can usually handle a large workload (e.g., Hugging Face spaces or the openplayground). More openness can enable the public to test models in real

time, helping increase public awareness and familiarity with the technology and develop a better mental model of it — which can help increase earned trust in AI systems. It also helps developers run qualitative evaluations and one-to-one comparisons of models' outputs, and it can facilitate red-teaming efforts and enable scrutiny from stakeholders with less technical resources.

Distribution

LICENSE

- **Data license:** Legal documents that define the use and access of data through a contractual agreement between the data provider (licensor) and the data user (licensee). Examples include the Open Data Commons Licenses and RAIL-D. More openness via increased access to data licenses can enable developers to better control their ethical and legal risks when using the datasets. (Research shows that 70%+ of licenses for popular datasets on GitHub and Hugging Face are "Unspecified". In addition, "the licenses that are attached to datasets uploaded to dataset sharing platforms are often inconsistent with the license ascribed by the original author of the dataset.")
- **Model's license:** Legal documents that define the use and access of models through a contractual agreement between the model's provider (licensor) and the model user (licensee). Examples include any Open Source Initiative Approved License like MIT or Apache 2.0, custom licenses like OpenRAIL, and Creative Commons licenses, with or without the non-commercial clause, like CC-BY-NC 4.0. More openness can enable developers to better control their ethical and legal risks when using the models or building on top of them (e.g. AI licensing categories from Open Core Ventures).
- **Code license:** Legal documents that define the use and access of code through a contractual agreement between the code provider (licensor) and the code user (licensee). Like traditional software, code licenses define the rules to use, modify and redistribute the code. More openness via access to code licenses enable the developers to know the conditions and permissions before using the code.

TYPE OF RELEASE

- **Gradual / Staged release:** Whether the model is incrementally or gradually accessible to a broader set of actors. For example, third-party auditors or expert red-teamers can have earlier access to the model than the general public. Every stage of the release helps experts scrutinize the model's usage, evaluate its societal impacts, and incorporate potential patches or heightened safety measures.

- **Gated or public access:** Providing access based on some conditions like registration (gated access) or without any prerequisites (public access). Gated access enables identified actors to download part of the models (e.g., weights and training code) or to provide vetted researchers, auditors, red-teamers, specific access via to fine-tuning API (e.g. "research API"). This privileged model access can be coupled with staged release.
- **Hosted inference endpoint:** Whether or not in, addition to downloadable weights, an inference endpoint is offered. Access to inference endpoints in addition to downloadable weights enable the developer to benefit from optimized inference services (e.g. through REST API) while enabling them to fully customize the model they use. It enables the developers to fully utilize the open models.

USER POLICY

- **Acceptable use policy**: Document stipulating constraints and practices that a user must agree to for access to the model. As the development of those models is changing rapidly and the technology is general purpose, it is not easy to foresee how the technology will serve in some particular applications and use cases. It greatly helps downstream developers to explicitly document with sufficient details any acceptable use of the foundation models that are released.
- **Reporting & Redress mechanisms:** Feedback loops between users, broader impacted people, developers and model providers. These are critical for ensuring reporting and redress for all stakeholders and affected people. Mechanisms like incident databases are first steps toward documenting and reporting misuses and accidents with models.