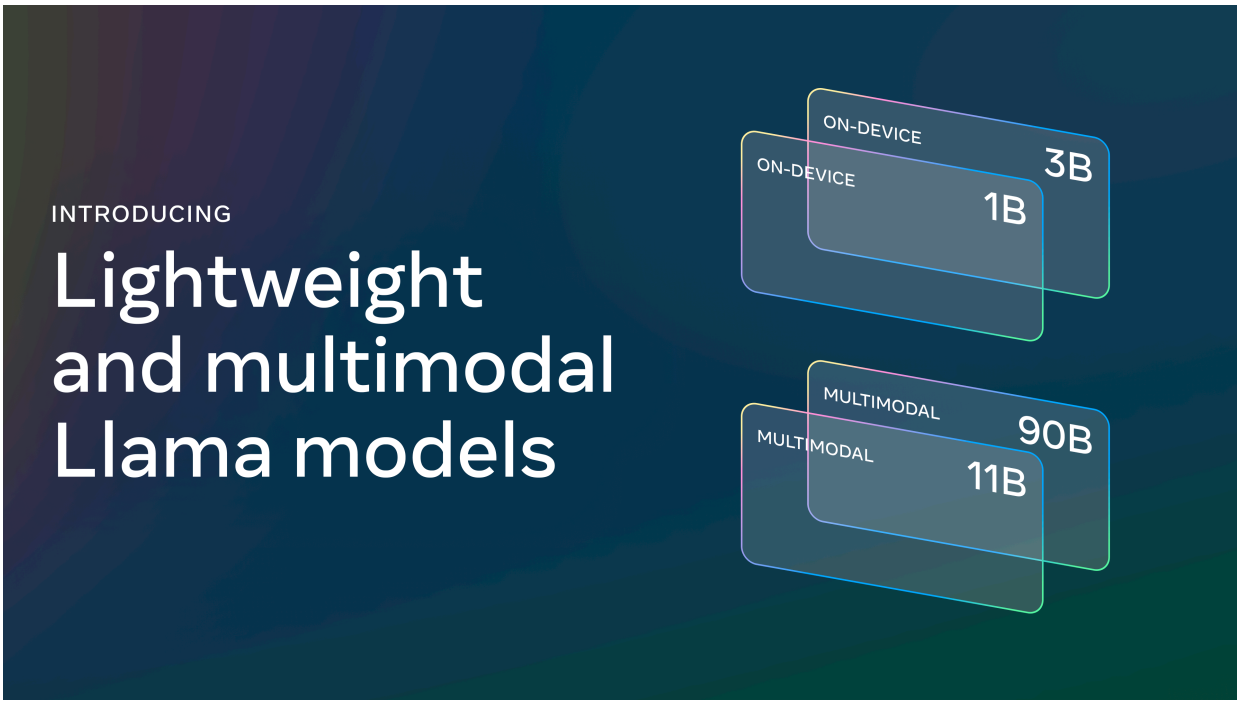
FEATURED

Large Language Model

Llama 3.2: Revolutionizing edge AI and vision with open, customizable models

September 25, 2024 • 15 minute read



Takeaways:

Context Length: 1B, 3B = 128K

- Today, we're releasing Llama 3.2, which includes small and medium-sized vision LLMs (11B and 90B), and lightweight, text-only models (1B and 3B) that fit onto edge and mobile devices, including pre-trained and instruction-tuned versions.
- The Llama 3.2 1B and 3B models support context length of 128K tokens and are state-of-the-art in their class for on-device use cases like summarization, instruction following, and rewriting tasks running locally at the edge. These models are enabled on day one for Qualcomm and MediaTek hardware and optimized for Arm processors.
- Supported by a broad ecosystem, the Llama 3.2 11B and 90B vision models are drop-in replacements for their corresponding text model equivalents, while exceeding on image understanding tasks compared to closed models, such as Claude 3 Haiku. Unlike other open multimodal models, both pre-trained and aligned models are available to be fine-tuned for custom applications using torchtune and deployed locally using torchchat. They're also available to try using our smart assistant, Meta AI.
- We're sharing the first official Llama Stack distributions, which will greatly simplify the way developers work with Llama models in different environments, including single-node, on-prem, cloud, and on-device, enabling turnkey deployment of retrieval-augmented generation (RAG) and tooling-enabled applications with integrated safety.

ExecuTorch is an end-to-end solution for enabling on-device inference capabilities across mobile and edge devices including wearables, embedded devices and microcontrollers. It is part of the PyTorch Edge ecosystem and enables efficient deployment of PyTorch models to edge devices.

- We've been working closely with partners like AWS, Databricks, Dell Technologies, Fireworks, Infosys, and Together AI to build Llama Stack distributions for their downstream enterprise clients. On-device distribution is via PyTorch ExecuTorch, and single-node distribution is via Ollama.
- We continue to share our work because we believe openness drives innovation and is good for developers, Meta, and the world. Llama is already leading the way on openness, modifiability, and cost efficiency—enabling more people to have creative, useful, and life-changing breakthroughs using generative AI.
- We're making Llama 3.2 models available for download on llama.com and Hugging Face, as well as available for immediate development on our broad ecosystem of partner platforms, including AMD, AWS, Databricks, Dell, Google Cloud, Groq, IBM, Intel, Microsoft Azure, NVIDIA, Oracle Cloud, Snowflake, and more.

We've been excited by the impact the Llama 3.1 herd of models have made in the two months since we announced them, including the 405B—the first open frontier-level AI model. While these models are incredibly powerful, we recognize that building with them requires significant compute resources and expertise. We've also heard from developers who don't have access to these resources and still want the opportunity to build with Llama. As Meta Founder and CEO Mark Zuckerberg shared today at Connect, they won't have to wait any longer. Today, we're releasing Llama 3.2, which includes small and medium-sized vision LLMs (11B and 90B) and lightweight, text-only models (1B and 3B) that fit onto select edge and mobile devices.

It's only been a year and a half since we first announced Llama, and we've made incredible progress in such a short amount of time. This year, Llama has achieved 10x growth and become the standard for responsible innovation. Llama also continues to lead on openness, modifiability, and cost efficiency, and it's competitive with closed models—even leading in some areas. We believe that openness drives innovation and is the right path forward, which is why we continue to share our research and collaborate with our partners and the developer community.

We're making Llama 3.2 models available for download on llama.com and Hugging Face, as well as available for immediate development on our broad ecosystem of partner platforms. Partners are an important part of this work, and we've worked with over 25 companies, including AMD, AWS, Databricks, Dell, Google Cloud, Groq, IBM, Intel, Microsoft Azure, NVIDIA, Oracle Cloud, and Snowflake, to enable services on day one. For the Llama 3.2 release, we're also working with on-device partners Arm, MediaTek, and Qualcomm to offer a broad range of services at launch. Starting today, we're also making Llama Stack available to the community. More details on the latest release, including information on the multimodal availability in Europe, can be found in our acceptable use policy.

Meet Llama 3.2

The two largest models of the Llama 3.2 collection, 11B and 90B, support image reasoning use cases, such as document-level understanding including charts and graphs, captioning of images, and visual grounding tasks such as directionally pinpointing objects in images based on natural language descriptions. For example, a person could ask a question about which month in the previous year their small business had the best sales, and Llama 3.2 can then reason based on an available graph and quickly provide the answer. In another example, the model could reason

with a map and help answer questions such as when a hike might become steeper or the distance of a particular trail marked on the map. The 11B and 90B models can also bridge the gap between vision and language by extracting details from an image, understanding the scene, and then crafting a sentence or two that could be used as an image caption to help tell the story.

The lightweight 1B and 3B models are highly capable with multilingual text generation and tool calling abilities. These models empower developers to build personalized, on-device agentic applications with strong privacy where data never leaves the device. For example, such an application could help summarize the last 10 messages received, extract action items, and leverage tool calling to directly send calendar invites for follow-up meetings.

Running these models locally comes with two major advantages. First, prompts and responses can feel instantaneous, since processing is done locally. Second, running models locally maintains privacy by not sending data such as messages and calendar information to the cloud, making the overall application more private. Since processing is handled locally, the application can clearly control which queries stay on the device and which may need to be processed by a larger model in the cloud.

Model evaluations

Our evaluation suggests that the Llama 3.2 vision models are competitive with leading foundation models, Claude 3 Haiku and GPT4o-mini on image recognition and a range of visual understanding tasks. The 3B model outperforms the Gemma 2 2.6B and Phi 3.5-mini models on tasks such as following instructions, summarization, prompt rewriting, and tool-use, while the 1B is competitive with Gemma.

We evaluated performance on over 150 benchmark datasets that span a wide range of languages. For the vision LLMs, we evaluated performance on benchmarks for image understanding and visual reasoning.

Vision instruction-tuned benchmarks

Modality	Category Benchmark	Llama 3.2 11B	Llama 3.2 90B	Claude 3 - Haiku	GPT-4o-mini
	College-level Problems and Mathematical Reasoning MMMU (val, 0-shot CoT, micro avg accuracy)	50.7	60.3	50.2	59.4
	MMMU-Pro, Standard (10 opts, test)	33.0	45.2	27.3	42.3

Image	MMMU-Pro, Vision (test)	23.7	33.8	20.1	**36.5**
	MathVista (testmini)	51.5	**57.3**	46.4	56.7
	Charts and Diagram Understanding ChartQA (test, 0-shot CoT relaxed accuracy)*	83.4	85.5	81.7	—
	AI2 Diagram (test)*	91.1	92.3	86.7	—
	DocVQA (test, ANLS)*	88.4	90.1	88.8	—
	General Visual Question Answering VQAv2 (test)	75.2	78.1	—	—
Text	General MMLU (0-shot, CoT)	73.0	**86.0**	75.2 (5-shot)	82.0
	Math MATH (0-shot, CoT)	51.9	68.0	38.9	**70.2**
	Reasoning GPQA (0-shot, CoT)	32.8	**46.7**	33.3	40.2
	Multilingual MGSM (0-shot, CoT)	68.9	86.9	75.1	**87.0**

Lightweight instruction-tuned benchmarks

Category Benchmark	Llama 3.2 1B	Llama 3.2 3B	Gemma 2 2B IT (measured)	Phi-3.5-mini IT (measured)
General MMLU (5-shot)	49.3	63.4	57.8	**69.0**
Open-rewrite eval (0-shot, rougeL)	**41.6**	40.1	31.2	34.5
TLDR9+ (test, 1-shot, rougeL)	16.8	**19.0**	13.9	12.8
IFEval	59.5	**77.4**	61.9	59.2
Tool Use BFCL V2	25.7	**67.0**	27.4	58.4
Nexus	13.5	**34.3**	21.0	26.1
Math GSM8K (8-shot, CoT)	44.4	77.7	62.5	**86.2**
MATH (0-shot, CoT)	30.6	**48.0**	23.8	44.2
Reasoning ARC Challenge (0-shot)	59.4	78.6	76.7	**87.4**
GPQA (0-shot)	27.2	**32.8**	27.5	31.9
Hellaswag (0-shot)	41.2	69.8	61.1	**81.4**
Long Context InfiniteBench/En.MC (128k)	38.0	63.3	—	39.2

InfiniteBench/En.QA (128k)	20.3	19.8	—	11.3
NIH/Multi-needle	75.0	84.7	—	52.7
Multilingual MGSM (0-shot, CoT)	24.5	**58.2**	40.2	49.8

Vision models

As the first Llama models to support vision tasks, the 11B and 90B models required an entirely new model architecture that supports image reasoning.

To add image input support, we trained a set of adapter weights that integrate the pre-trained image encoder into the pre-trained language model. The adapter consists of a series of cross-attention layers that feed image encoder representations into the language model. We trained the adapter on text-image pairs to align the image representations with the language representations. During adapter training, we also updated the parameters of the image encoder, but intentionally did not update the language-model parameters. By doing that, we keep all the text-only capabilities intact, providing developers a drop-in replacement for Llama 3.1 models.

Our training pipeline consists of multiple stages, starting from pretrained Llama 3.1 text models. First, we add image adapters and encoders, then pretrain on large-scale noisy (image, text) pair data. Next, we train on medium-scale high quality in-domain and knowledge-enhanced (image, text) pair data.

In post-training, we use a similar recipe as the text models by doing several rounds of alignment on supervised fine-tuning, rejection sampling, and direct preference optimization. We leverage synthetic data generation by using the Llama 3.1 model to filter and augment question and answers on top of in-domain images, and use a reward model to rank all the candidate answers to provide high quality fine-tuning data. We also add safety mitigation data to produce a model with a high level of safety while retaining helpfulness of the mode

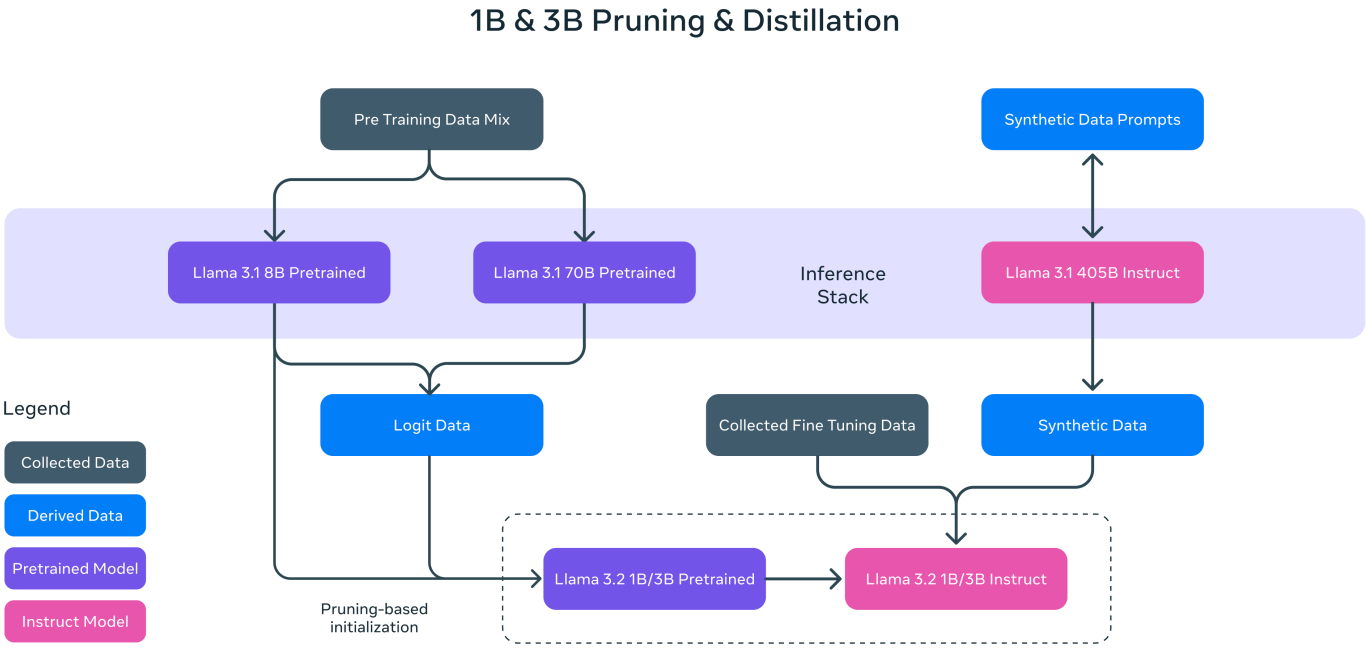
The end result is a set of models that can take in both image and text prompts, and deeply understand and reason on the combination. This is another step toward Llama models having even richer agentic capabilities.

Lightweight models

As we talked about with Llama 3.1, powerful teacher models can be leveraged to create smaller models that have improved performance. We used two methods—pruning and distillation—on the 1B and 3B models, making them the first highly capable lightweight Llama models that can fit on devices efficiently.

Pruning enabled us to reduce the size of extant models in the Llama herd while recovering as much knowledge and performance as possible. For the 1B and 3B models, we took the approach of using structured pruning in a single shot manner from the Llama 3.1 8B. This involved systematically removing parts of the network and adjusting the magnitude of the weights and gradients to create a smaller, more efficient model that retains the performance of the original network.

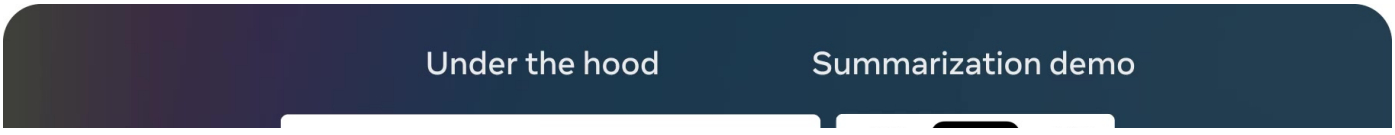
Knowledge distillation uses a larger network to impart knowledge on a smaller network, with the idea that a smaller model can achieve better performance using a teacher than it could from scratch. For the 1B and 3B in Llama 3.2, we incorporated logits from the Llama 3.1 8B and 70B models into the pre-training stage of the model development, where outputs (logits) from these larger models were used as token-level targets. Knowledge distillation was used after pruning to recover performance.

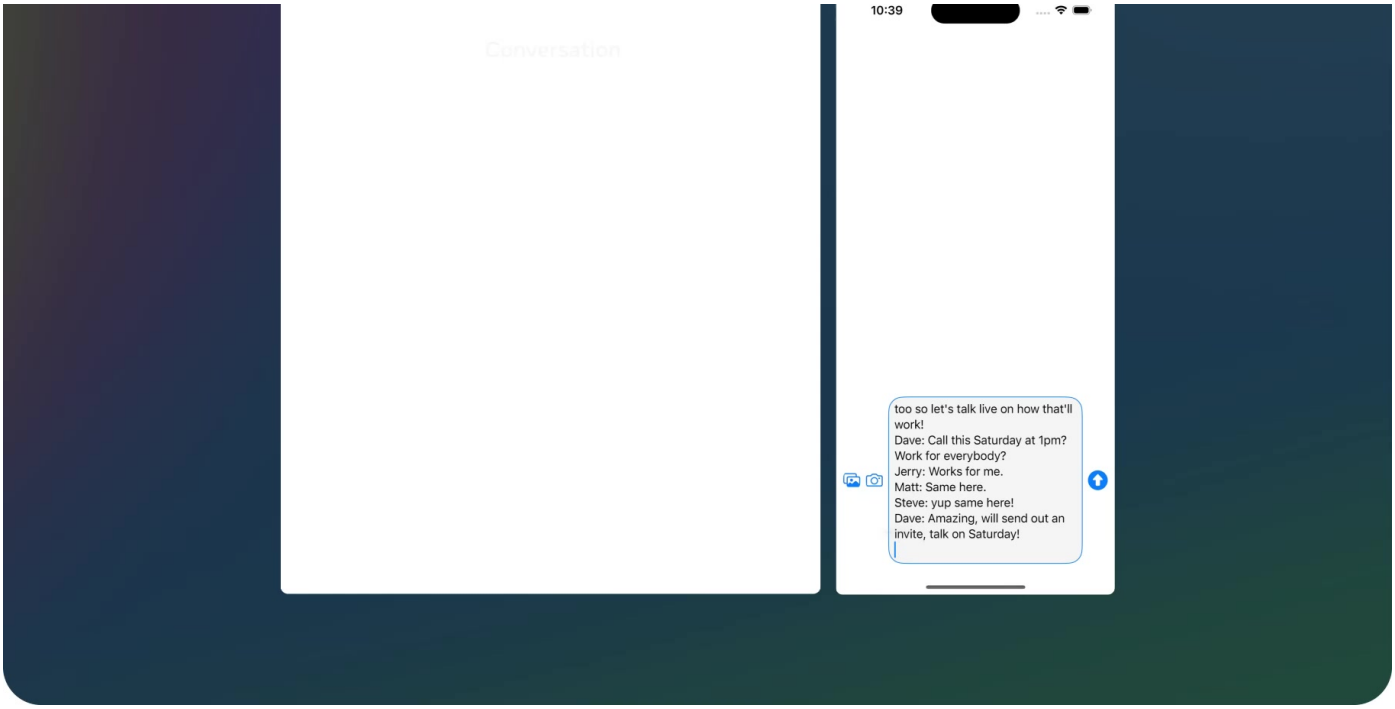


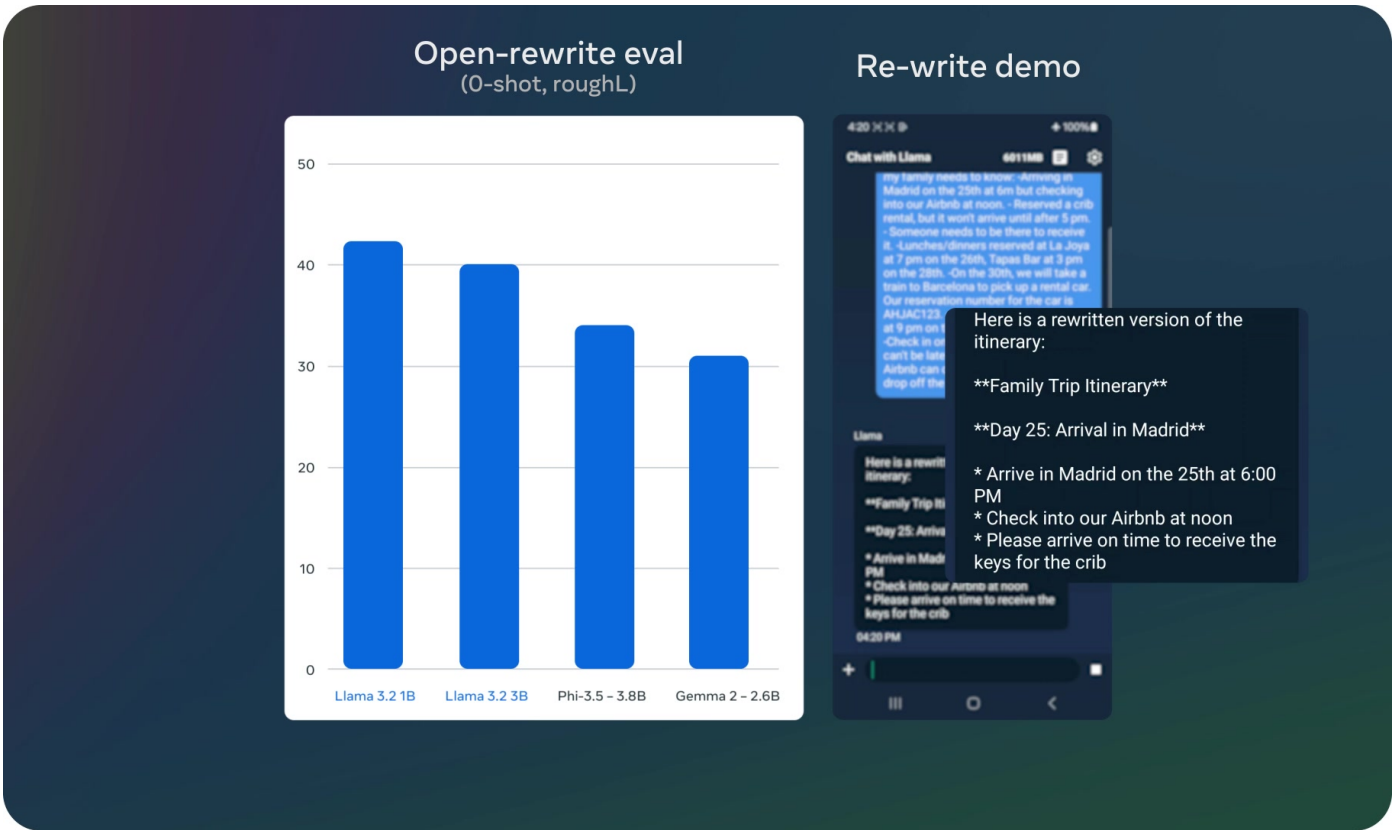
In post-training, we use a similar recipe as Llama 3.1 and produce final chat models by doing several rounds of alignment on top of the pre-trained model. Each round involves supervised fine-tuning (SFT), rejection sampling (RS), and direct preference optimization (DPO).

In post-training, we scale context length support to 128K tokens, while maintaining the same quality as the pre-trained model. We also engage in synthetic data generation that goes through careful data processing and filtering to ensure high quality. We carefully blend the data to optimize for high quality across multiple capabilities like summarization, rewriting, instruction following, language reasoning, and tool use.

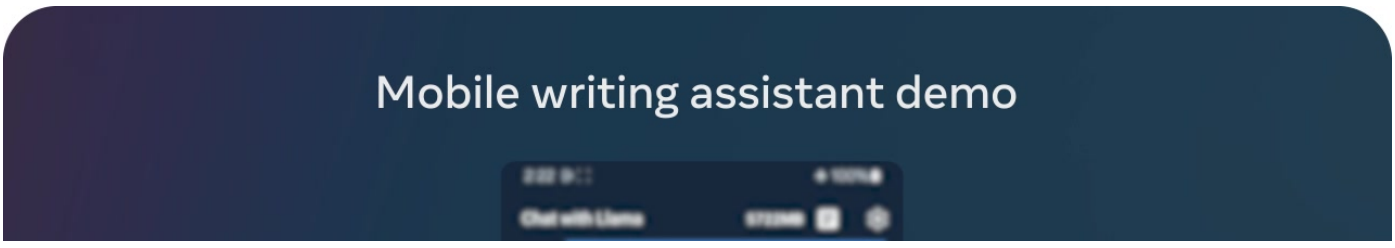
To enable the community to innovate on these models, we worked closely with Qualcomm and Mediatek, the top two mobile system on a chip (SoC) companies in the world, and Arm, who provides the foundational compute platform for 99% of mobile devices. The weights being released today are based on BFloat16 numerics. Our teams are actively exploring quantized variants that will run even faster, and we hope to share more on that soon.

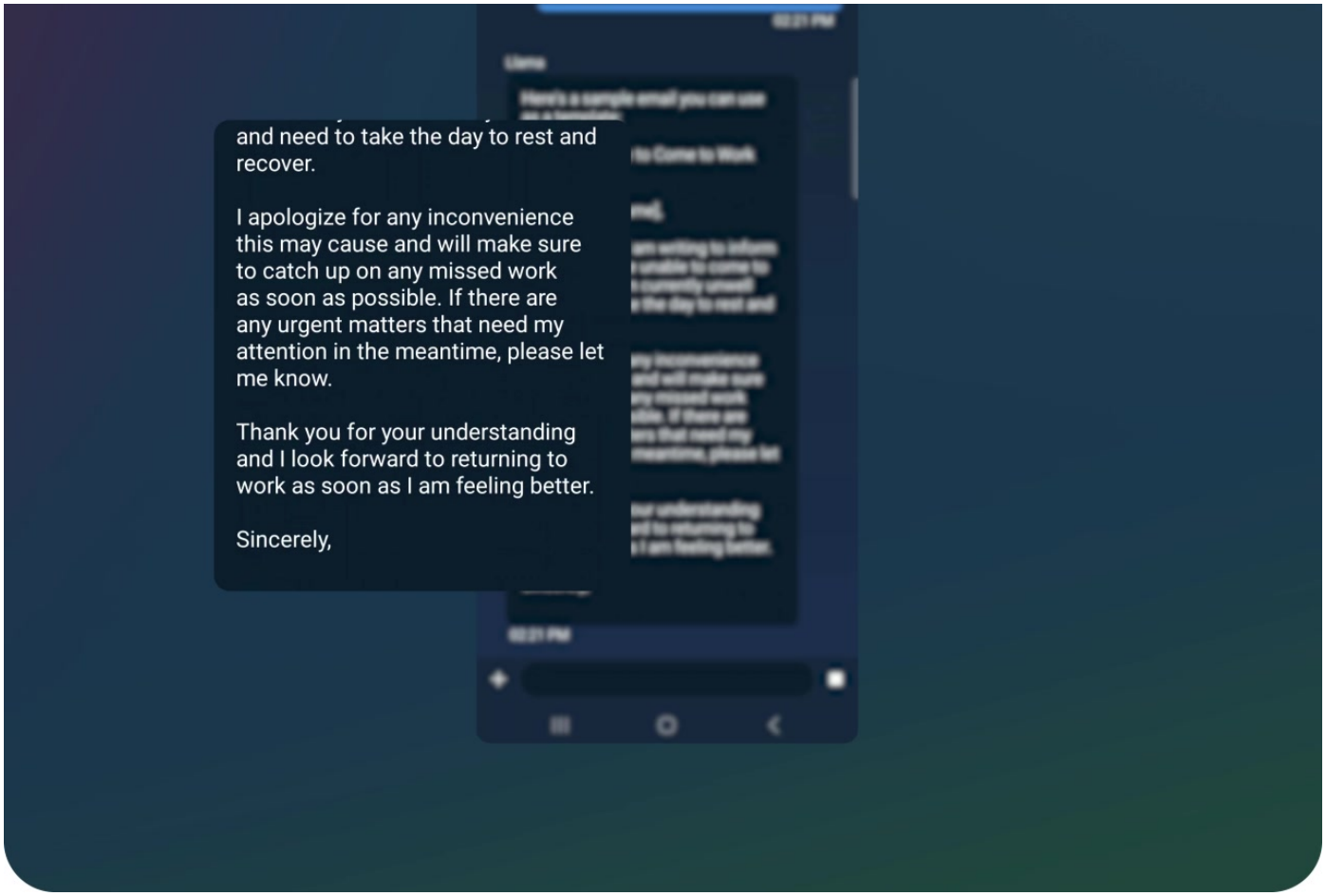






This demo is based on an unreleased quantized model.





This demo is based on an unreleased quantized model.

Llama Stack distributions

In July, we released a request for comment on the Llama Stack API, a standardized interface for canonical toolchain components (fine-tuning, synthetic data generation) to customize Llama models and build agentic applications. The engagement has been great.

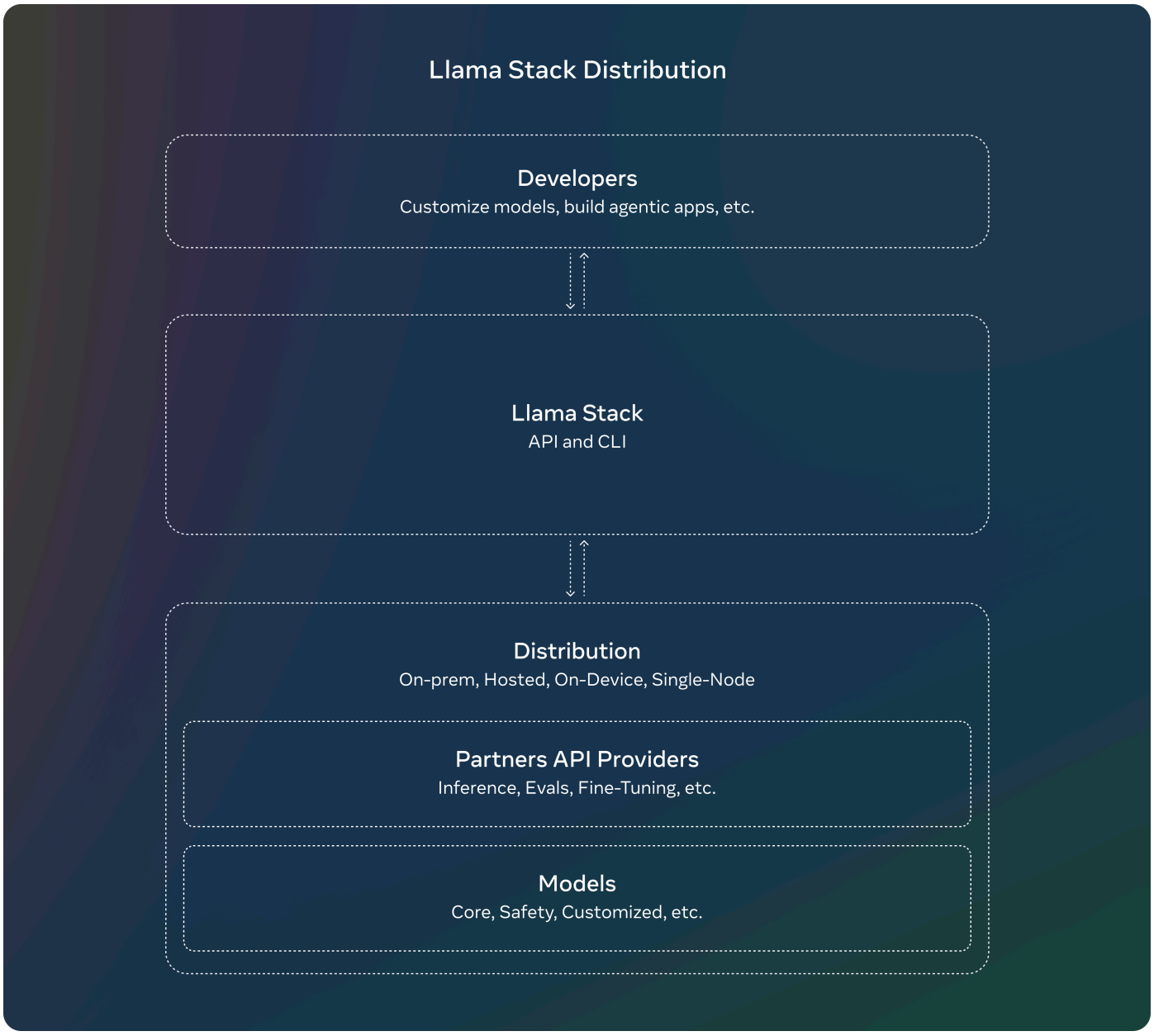
Since then, we have been working hard to make the API real. We built a reference implementation of the APIs for inference, tool use, and RAG. In addition, we have been working with partners to adapt them to become providers for the APIs. Finally, we have introduced Llama Stack Distribution as a way to package multiple API Providers that work well together to provide a single endpoint for developers. We are now sharing with the community a simplified and consistent experience that will enable them to work with Llama models in multiple environments, including on-prem, cloud, single-node, and on-device.

The full set of releases includes:

1. Llama CLI (command line interface) to build, configure, and run Llama Stack distributions
2. Client code in multiple languages, including python, node, kotlin, and swift
3. Docker containers for Llama Stack Distribution Server and Agents API Provider
4. Multiple distributions
    1. Single-node Llama Stack Distribution via Meta internal implementation and

1. Single-node Llama Stack Distribution via Meta internal implementation and Ollama
2. Cloud Llama Stack distributions via AWS, Databricks, Fireworks, and Together
3. On-device Llama Stack Distribution on iOS implemented via PyTorch ExecuTorch
4. On-prem Llama Stack Distribution supported by Dell

We look forward to working with developers and partners to simplify all aspects of building with Llama models and welcome feedback.



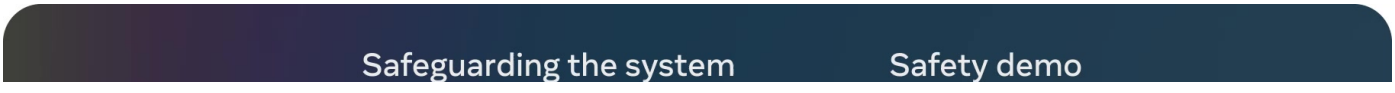
System level safety

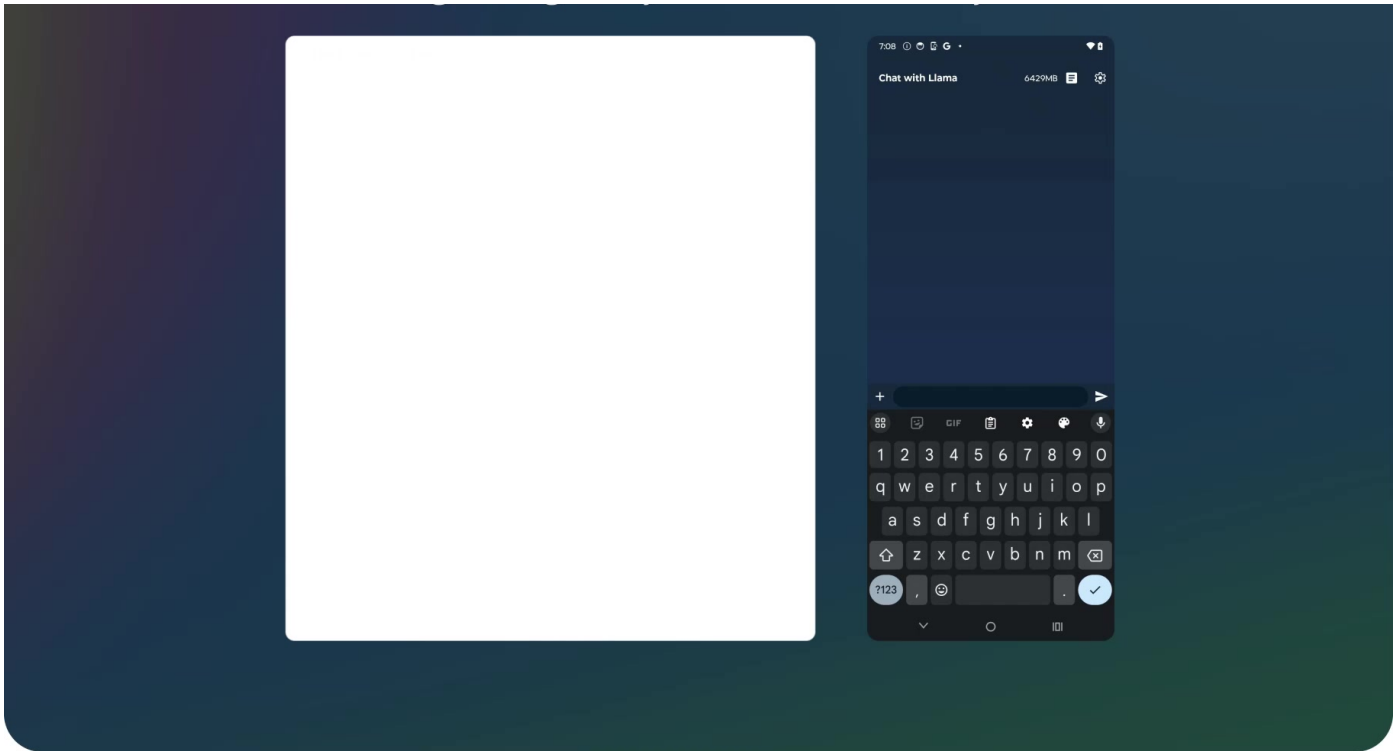
Taking an open approach has many benefits. It helps ensure that more people around the world can access the opportunities that AI provides, guards against concentrating power in the hands of a small few, and deploys technology more equitably and safely across society. As we continue to innovate, we also want to make sure we're empowering developers to build safe and responsible systems.

Building on our previous release and continuous effort to support responsible innovation, today we're adding new updates to our family of safeguards:

- First, we're releasing Llama Guard 3 11B Vision, which is designed to support Llama 3.2's new image understanding capability and filter text+image input prompts or text output responses to these prompts.
- Second, as we released 1B and 3B Llama models to be used in more constrained environments like on-device, we also optimized Llama Guard to drastically reduce its deployment cost. Llama Guard 3 1B is based on the Llama 3.2 1B model and has been pruned and quantized bringing its size from 2,858 MB down to 438 MB, making it more efficient than ever to deploy.

These new solutions are integrated into our reference implementations, demos, and applications and are ready for the open source community to use on day one.





Try Llama 3.2 today

Llama 3.2 is poised to reach more people than ever before and enable exciting new use cases. We believe sharing these models with the open source community isn't enough. We want to make sure developers also have the tools they need to build with Llama responsibly. As part of our continued responsible release efforts, we're offering developers new tools and resources, and as always, we'll update best practices in our Responsible Use Guide.

We continue to share the latest advancements in the Llama ecosystem because we believe openness drives innovation and is good for developers, Meta, and the world. We're excited to continue the conversations we're having with our partners and the open source community, and as always, we can't wait to see what the community builds using Llama 3.2 and Llama Stack.

This work was supported by our partners across the AI community. We'd like to thank and acknowledge (in alphabetical order): Accenture, AMD, Arm, AWS, Cloudflare, Databricks, Dell, Deloitte, Fireworks.ai, Google Cloud, Groq, Hugging Face, IBM watsonx, Infosys, Intel, Kaggle, Lenovo, LMSYS, MediaTek, Microsoft Azure, NVIDIA, OctoAI, Ollama, Oracle Cloud, PwC, Qualcomm, Sarvam AI, Scale AI, Snowflake, Together AI, and UC Berkeley - vLLM Project.

Learn more on the Llama website

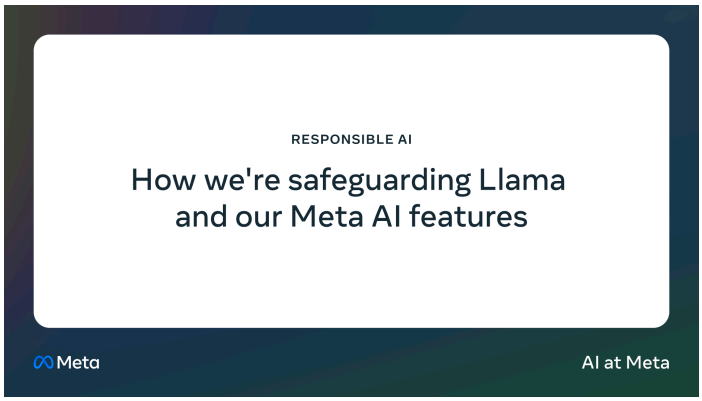
Visit Hugging Face

Share:

Join us in the pursuit of what's possible with AI.

See all open positions

Related Posts



Responsible AI

Connect 2024: The responsible approach we're taking to generative AI

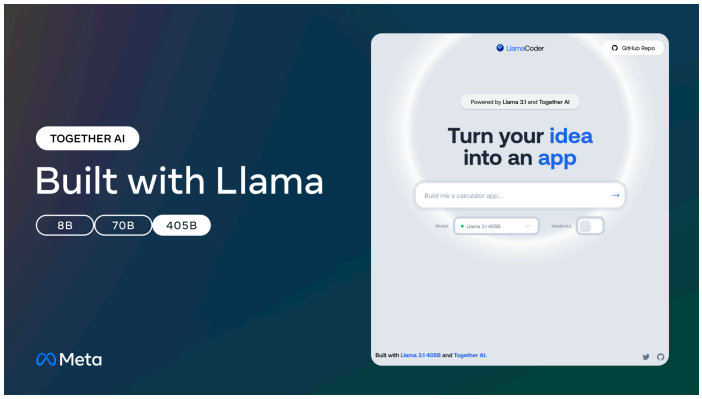
September 25, 2024

Read post

With 10x growth since 2023, Llama is the leading engine of AI innovation

August 29, 2024

Read post



Open Source

Generate an entire app from a prompt using Together AI's LlamaCoder

September 18, 2024

Read post

Product experiences

Meta AI
AI Studio

Latest news

Blog
Newsletter

Foundational models

Llama