Model Information

The Llama 3.2-Vision collection of multimodal large language models (LLMs) is a collection of pretrained and instruction-tuned image reasoning generative models in 11B and 90B sizes (text + images in / text out). The Llama 3.2-Vision instruction-tuned models are optimized for visual recognition, image reasoning, captioning, and answering general questions about an image. The models outperform many of the available open source and closed multimodal models on common industry benchmarks.

Model Developer: Meta

Model Architecture: Llama 3.2-Vision is built on top of Llama 3.1 text-only model, which is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align with human preferences for helpfulness and safety. To support image recognition tasks, the Llama 3.2-Vision model uses a separately trained vision adapter that integrates with the pre-trained Llama 3.1 language model. The adapter consists of a series of cross-attention layers that feed image encoder representations into the core LLM.

	Training Data	Params	Input modalities	Output modalities	Context length	GQA	Data volur
Llama 3.2-Vision	(Image, text) pairs	11B (10.6)	Text + Image	Text	128k	Yes	6B (imag text) pairs
Llama 3.2-Vision	(Image, text) pairs	90B (88.8)	Text + Image	Text	128k	Yes	6B (imag text) pairs

Supported Languages: For text only tasks, English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai are officially supported. Llama 3.2 has been trained on a broader collection of languages than these 8 supported languages. Note for image+text applications, English is the only language supported.

Developers may fine-tune Llama 3.2 models for languages beyond these supported languages, provided they comply with the Llama 3.2 Community License and the Acceptable Use Policy. Developers are always expected to ensure that their deployments, including those that involve additional languages, are completed safely and responsibly.

Llama 3.2 Model Family: Token counts refer to pretraining data only. All model versions use Grouped-Query Attention (GQA) for improved inference scalability.

Model Release Date: Sept 25, 2024

Status: This is a static model trained on an offline dataset. Future versions may be released that improve model capabilities and safety.

License: Use of Llama 3.2 is governed by the Llama 3.2 Community License (a custom, commercial license agreement).

Feedback: Where to send questions or comments about the model Instructions on how to provide feedback or comments on the model can be found in the model README. For more technical information about generation parameters and recipes for how to use Llama 3.2-Vision in applications, please go here.

Intended Use

Intended Use Cases: Llama 3.2-Vision is intended for commercial and research use. Instruction tuned models are intended for visual recognition, image reasoning, captioning, and assistant-like chat with images, whereas

pretrained models can be adapted for a variety of image reasoning tasks. Additionally, because of Llama 3.2-Vision's ability to take images and text as inputs, additional use cases could include:

1. Visual Question Answering (VQA) and Visual Reasoning: Imagine a machine that looks at a picture and understands your questions about it.
2. Document Visual Question Answering (DocVQA): Imagine a computer understanding both the text and layout of a document, like a map or contract, and then answering questions about it directly from the image.
3. Image Captioning: Image captioning bridges the gap between vision and language, extracting details, understanding the scene, and then crafting a sentence or two that tells the story.
4. Image-Text Retrieval: Image-text retrieval is like a matchmaker for images and their descriptions. Similar to a search engine but one that understands both pictures and words.
5. Visual Grounding: Visual grounding is like connecting the dots between what we see and say. It's about understanding how language references specific parts of an image, allowing AI models to pinpoint objects or regions based on natural language descriptions.

The Llama 3.2 model collection also supports the ability to leverage the outputs of its models to improve other models including synthetic data generation and distillation. The Llama 3.2 Community License allows for these use cases.

Out of Scope: Use in any manner that violates applicable laws or regulations (including trade compliance laws). Use in any other way that is prohibited by the Acceptable Use Policy and Llama 3.2 Community License. Use in languages beyond those explicitly referenced as supported in this model card.

How to use

This repository contains two versions of Llama-3.2-11B-Vision-Instruct, for use with transformers and with the original `llama` codebase.

Use with transformers

Starting with transformers >= 4.45.0 onward, you can run inference using conversational messages that may include an image you can query about.

Make sure to update your transformers installation via `pip install --upgrade transformers`.

```
import requests
import torch
from PIL import Image
from transformers import MllamaForConditionalGeneration, AutoProcessor

model_id = "meta-llama/Llama-3.2-11B-Vision-Instruct"

model = MllamaForConditionalGeneration.from_pretrained(
    model_id,
    torch_dtype=torch.bfloat16,
    device_map="auto",
)
processor = AutoProcessor.from_pretrained(model_id)

url = "https://huggingface.co/datasets/huggingface/documentation-images/resol
image = Image.open(requests.get(url, stream=True).raw)

messages = [
    {"role": "user", "content": [
        {"type": "image"},
        {"type": "text", "text": "If I had to write a haiku for this one, it
    ]}
]
```

```
input_text = processor.apply_chat_template(messages, add_generation_prompt=Tr
inputs = processor(
    image,
    input_text,
    add_special_tokens=False,
    return_tensors="pt"
).to(model.device)

output = model.generate(**inputs, max_new_tokens=30)
print(processor.decode(output[0]))
```

Use with `llama`

Please, follow the instructions in the repository.

To download the original checkpoints, you can use `huggingface-cli` as follows:

```
huggingface-cli download meta-llama/Llama-3.2-11B-Vision-Instruct --include "
```

Hardware and Software

Training Factors: We used custom training libraries, Meta's custom built GPU cluster, and production infrastructure for pretraining. Fine-tuning, annotation, and evaluation were also performed on production infrastructure.

Training Energy Use: Training utilized a cumulative of **2.02M** GPU hours of computation on H100-80GB (TDP of 700W) type hardware, per the table below. Training time is the total GPU time required for training each model and power consumption is the peak power capacity per GPU device used, adjusted for power usage efficiency.

Training Greenhouse Gas Emissions: Estimated total location-based

greenhouse gas emissions were **584** tons CO2eq for training. Since 2020, Meta has maintained net zero greenhouse gas emissions in its global operations and matched 100% of its electricity use with renewable energy, therefore the total market-based greenhouse gas emissions for training were 0 tons CO2eq.

	Training Time (GPU hours)	Training Power Consumption (W)	Training Location-Based Greenhouse Gas Emissions (tons CO2eq)	Training Market-Based Greenhouse Gas Emissions (tons CO2eq)
Llama 3.2-vision 11B	Stage 1 pretraining: 147K H100 hours Stage 2 annealing: 98K H100 hours SFT: 896 H100 hours RLHF: 224 H100 hours	700	71	0
Llama 3.2-vision 90B	Stage 1 pretraining: 885K H100 hours Stage 2 annealing: 885K H100 hours SFT: 3072 H100 hours RLHF: 2048 H100 hours	700	513	0
Total	2.02M		584	0

The methodology used to determine training energy use and greenhouse gas emissions can be found here. Since Meta is openly releasing these models, the training energy use and greenhouse gas emissions will not be incurred by others.

Training Data

Overview: Llama 3.2-Vision was pretrained on 6B image and text pairs.

The instruction tuning data includes publicly available vision instruction datasets, as well as over 3M synthetically generated examples.

Data Freshness: The pretraining data has a cutoff of December 2023.

Benchmarks - Image Reasoning

In this section, we report the results for Llama 3.2-Vision models on standard automatic benchmarks. For all these evaluations, we used our internal evaluations library.

Base Pretrained Models

Category	Benchmark	# Shots	Metric	Llama 3.2 11B	Llama 3.2 90B
Image Understanding	VQAv2 (val)	0	Accuracy	66.8	73.6
	Text VQA (val)	0	Relaxed accuracy	73.1	73.5
	DocVQA (val, unseen)	0	ANLS	62.3	70.7
Visual Reasoning	MMMU (val, 0-shot)	0	Micro average accuracy	41.7	49.3
	ChartQA (test)	0	Accuracy	39.4	54.2
	InfographicsQA (val, unseen)	0	ANLS	43.2	56.8
	AI2 Diagram (test)	0	Accuracy	62.4	75.3

Instruction Tuned Models

Modality	Capability	Benchmark	# Shots	Metric	Llama 3.2 11B	Ll 9

Image	College-level Problems and Mathematical Reasoning	MMMU (val, CoT)	0	Micro average accuracy	50.7	6(
		MMMU-Pro, Standard (10 opts, test)	0	Accuracy	33.0	45
		MMMU-Pro, Vision (test)	0	Accuracy	23.7	33
		MathVista (testmini)	0	Accuracy	51.5	57
	Charts and Diagram Understanding	ChartQA (test, CoT)	0	Relaxed accuracy	83.4	85
		AI2 Diagram (test)	0	Accuracy	91.1	92
		DocVQA (test)	0	ANLS	88.4	90
	General Visual Question Answering	VQAv2 (test)	0	Accuracy	75.2	78
Text	General	MMLU (CoT)	0	Macro_avg/acc	73.0	86
	Math	MATH (CoT)	0	Final_em	51.9	68
	Reasoning	GPQA	0	Accuracy	32.8	46
	Multilingual	MGSM (CoT)	0	em	68.9	86

Responsibility & Safety

As part of our Responsible release approach, we followed a three-pronged

strategy to managing trust & safety risks:

1. Enable developers to deploy helpful, safe and flexible experiences for their target audience and for the use cases supported by Llama.
2. Protect developers against adversarial users aiming to exploit Llama capabilities to potentially cause harm.
3. Provide protections for the community to help prevent the misuse of our models.

Responsible Deployment

Approach: Llama is a foundational technology designed to be used in a variety of use cases, examples on how Meta's Llama models have been responsibly deployed can be found in our Community Stories webpage. Our approach is to build the most helpful models enabling the world to benefit from the technology power, by aligning our model safety for the generic use cases addressing a standard set of harms. Developers are then in the driver seat to tailor safety for their use case, defining their own policy and deploying the models with the necessary safeguards in their Llama systems. Llama 3.2 was developed following the best practices outlined in our Responsible Use Guide, you can refer to the Responsible Use Guide to learn more.

Llama 3.2 Instruct

Objective: Our main objectives for conducting safety fine-tuning are to provide the research community with a valuable resource for studying the robustness of safety fine-tuning, as well as to offer developers a readily available, safe, and powerful model for various applications to reduce the developer workload to deploy safe AI systems. We implemented the same set of safety mitigations as in Llama 3, and you can learn more about these in the Llama 3 paper.

Fine-Tuning Data: We employ a multi-faceted approach to data

collection, combining human-generated data from our vendors with synthetic data to mitigate potential safety risks. We've developed many large language model (LLM)-based classifiers that enable us to thoughtfully select high-quality prompts and responses, enhancing data quality control.

Refusals and Tone: Building on the work we started with Llama 3, we put a great emphasis on model refusals to benign prompts as well as refusal tone. We included both borderline and adversarial prompts in our safety data strategy, and modified our safety data responses to follow tone guidelines.

Llama 3.2 Systems

Safety as a System: Large language models, including Llama 3.2, **are not designed to be deployed in isolation** but instead should be deployed as part of an overall AI system with additional safety guardrails as required. Developers are expected to deploy system safeguards when building agentic systems. Safeguards are key to achieve the right helpfulness-safety alignment as well as mitigating safety and security risks inherent to the system and any integration of the model or system with external tools. As part of our responsible release approach, we provide the community with safeguards that developers should deploy with Llama models or other LLMs, including Llama Guard, Prompt Guard and Code Shield. All our reference implementations demos contain these safeguards by default so developers can benefit from system-level safety out-of-the-box.

New Capabilities and Use Cases

Technological Advancement: Llama releases usually introduce new capabilities that require specific considerations in addition to the best practices that generally apply across all Generative AI use cases. For prior

release capabilities also supported by Llama 3.2, see Llama 3.1 Model Card, as the same considerations apply here as well.,

Image Reasoning: Llama 3.2-Vision models come with multimodal (text and image) input capabilities enabling image reasoning applications. As part of our responsible release process, we took dedicated measures including evaluations and mitigations to address the risk of the models uniquely identifying individuals in images. As with other LLM risks, models may not always be robust to adversarial prompts, and developers should evaluate identification and other applicable risks in the context of their applications as well as consider deploying Llama Guard 3-11B-Vision as part of their system or other mitigations as appropriate to detect and mitigate such risks.

Evaluations

Scaled Evaluations: We built dedicated, adversarial evaluation datasets and evaluated systems composed of Llama models and Purple Llama safeguards to filter input prompt and output response. It is important to evaluate applications in context, and we recommend building dedicated evaluation dataset for your use case.

Red teaming: We conducted recurring red teaming exercises with the goal of discovering risks via adversarial prompting and we used the learnings to improve our benchmarks and safety tuning datasets. We partnered early with subject-matter experts in critical risk areas to understand the nature of these real-world harms and how such models may lead to unintended harm for society. Based on these conversations, we derived a set of adversarial goals for the red team to attempt to achieve, such as extracting harmful information or reprogramming the model to act in a potentially harmful capacity. The red team consisted of experts in cybersecurity, adversarial machine learning, responsible AI, and integrity in addition to multilingual content specialists with background in

integrity issues in specific geographic markets.

Critical Risks

In addition to our safety work above, we took extra care on measuring and/or mitigating the following critical risk areas:

1. CBRNE (Chemical, Biological, Radiological, Nuclear, and Explosive Weapons): For Llama 3.1, to assess risks related to proliferation of chemical and biological weapons, we performed uplift testing designed to assess whether use of Llama 3.1 models could meaningfully increase the capabilities of malicious actors to plan or carry out attacks using these types of weapons. For Llama 3.2-Vision models, we conducted additional targeted evaluations and found that it was unlikely Llama 3.2 presented an increase in scientific capabilities due to its added image understanding capability as compared to Llama 3.1.

2. Child Safety: Child Safety risk assessments were conducted using a team of experts, to assess the model's capability to produce outputs that could result in Child Safety risks and inform on any necessary and appropriate risk mitigations via fine tuning. We leveraged those expert red teaming sessions to expand the coverage of our evaluation benchmarks through Llama 3 model development. For Llama 3, we conducted new in-depth sessions using objective based methodologies to assess the model risks along multiple attack vectors including the additional languages Llama 3 is trained on. We also partnered with content specialists to perform red teaming exercises assessing potentially violating content while taking account of market specific nuances or experiences.

3. Cyber Attacks: For Llama 3.1 405B, our cyber attack uplift study investigated whether LLMs can enhance human capabilities in hacking tasks, both in terms of skill level and speed. Our attack automation study focused on evaluating the capabilities of LLMs when used as autonomous

agents in cyber offensive operations, specifically in the context of ransomware attacks. This evaluation was distinct from previous studies that considered LLMs as interactive assistants. The primary objective was to assess whether these models could effectively function as independent agents in executing complex cyber-attacks without human intervention. Because Llama 3.2's vision capabilities are not generally germane to cyber uplift, we believe that the testing conducted for Llama 3.1 also applies to Llama 3.2.

Community

Industry Partnerships: Generative AI safety requires expertise and tooling, and we believe in the strength of the open community to accelerate its progress. We are active members of open consortiums, including the AI Alliance, Partnership on AI and MLCommons, actively contributing to safety standardization and transparency. We encourage the community to adopt taxonomies like the MLCommons Proof of Concept evaluation to facilitate collaboration and transparency on safety and content evaluations. Our Purple Llama tools are open sourced for the community to use and widely distributed across ecosystem partners including cloud service providers. We encourage community contributions to our Github repository.

Grants: We also set up the Llama Impact Grants program to identify and support the most compelling applications of Meta's Llama model for societal benefit across three categories: education, climate and open innovation. The 20 finalists from the hundreds of applications can be found here.

Reporting: Finally, we put in place a set of resources including an output reporting mechanism and bug bounty program to continuously improve the Llama technology with the help of the community.

Ethical Considerations and Limitations

Values: The core values of Llama 3.2 are openness, inclusivity and helpfulness. It is meant to serve everyone, and to work for a wide range of use cases. It is thus designed to be accessible to people across many different backgrounds, experiences and perspectives. Llama 3.2 addresses users and their needs as they are, without insertion unnecessary judgment or normativity, while reflecting the understanding that even content that may appear problematic in some cases can serve valuable purposes in others. It respects the dignity and autonomy of all users, especially in terms of the values of free thought and expression that power innovation and progress.

Testing: But Llama 3.2 is a new technology, and like any new technology, there are risks associated with its use. Testing conducted to date has not covered, nor could it cover, all scenarios. For these reasons, as with all LLMs, Llama 3.2's potential outputs cannot be predicted in advance, and the model may in some instances produce inaccurate, biased or other objectionable responses to user prompts. Therefore, before deploying any applications of Llama 3.2 models, developers should perform safety testing and tuning tailored to their specific applications of the model. Please refer to available resources including our Responsible Use Guide, Trust and Safety solutions, and other resources to learn more about responsible development.