Search models, datasets, users...

google / gemma-4-E2B

like 269 | Follow Google 55.5k

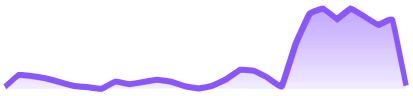
Any-to-Any | Transformers | Safetensors | gemma4 | image-text-to-text

License: apache-2.0

Deploy | Use this model

Model card | Files xet | Community 9

Downloads last month
926,965



Safetensors | Model size 5B params | Tensor type BF16 | Files info

Inference Providers NEW

Any-to-Any

This model isn't deployed by any Inference Provider. 18 Ask for provider support

Model tree for google/gemma-4-E2B

- **Adapters** 18 models
- **Finetunes** 59 models
- **Quantizations** 31 models

Spaces using google/gemma-4-E2B 5

jsmanrique/license-cat-analyzer | Carnot-EBM/hallucination-detector | BART-ender/seige

wehe1pwe/math-under-llm | AlexandreScriptsMT/Testegm4

Collection including google/gemma-4-E2B

Gemma 4 Collection
12 items • Updated 13 days ago • △ 820

Hugging Face | GitHub | Launch Blog | Documentation
License: Apache 2.0 | **Authors**: Google DeepMind

Gemma is a family of open models built by Google DeepMind. Gemma 4 models are multimodal, handling text and image input (with audio supported on small models) and generating text output. This release includes open-weights models in both pre-trained and instruction-tuned variants. Gemma 4 features a context window of up to 256K tokens and maintains multilingual support in over 140 languages.

Featuring both Dense and Mixture-of-Experts (MoE) architectures, Gemma 4 is well-suited for tasks like text generation, coding, and reasoning. The models are available in four distinct sizes: **E2B**, **E4B**, **26B A4B**, and **31B**. Their diverse sizes make them deployable in environments ranging from high-end phones to laptops and servers, democratizing access to state-of-the-art AI.

Gemma 4 introduces key **capability and architectural advancements**:

- **Reasoning** – All models in the family are designed as highly capable reasoners, with configurable thinking modes.
- **Extended Multimodalities** – Processes Text, Image with variable aspect ratio and resolution support (all models), Video, and Audio (featured natively on the E2B and E4B models).

- **Diverse & Efficient Architectures** – Offers Dense and Mixture-of-Experts (MoE) variants of different sizes for scalable deployment.
- **Optimized for On-Device** – Smaller models are specifically designed for efficient local execution on laptops and mobile devices.
- **Increased Context Window** – The small models feature a 128K context window, while the medium models support 256K.
- **Enhanced Coding & Agentic Capabilities** – Achieves notable improvements in coding benchmarks alongside native function-calling support, powering highly capable autonomous agents.
- **Native System Prompt Support** – Gemma 4 introduces native support for the `system` role, enabling more structured and controllable conversations.

Models Overview

Gemma 4 models are designed to deliver frontier-level performance at each size, targeting deployment scenarios from mobile and edge devices (E2B, E4B) to consumer GPUs and workstations (26B A4B, 31B). They are well-suited for reasoning, agentic workflows, coding, and multimodal understanding.

The models employ a hybrid attention mechanism that interleaves local sliding window attention with full global attention, ensuring the final layer is always global. This hybrid design delivers the processing speed and low memory footprint of a lightweight model without sacrificing the deep awareness required for complex, long-context tasks. To optimize memory for long contexts, global layers feature unified Keys and Values, and apply Proportional RoPE (p-RoPE).

Dense Models

Property	E2B	E4B	31B Dense
Total Parameters	2.3B effective (5.1B with embeddings)	4.5B effective (8B with embeddings)	30.7B
Layers	35	42	60
Sliding Window	512 tokens	512 tokens	1024 tokens
Context Length	128K tokens	128K tokens	256K tokens
Vocabulary Size	262K	262K	262K
Supported Modalities	Text, Image, Audio	Text, Image, Audio	Text, Image
Vision Encoder Parameters	*~150M*	*~150M*	*~550M*
Audio Encoder Parameters	*~300M*	*~300M*	No Audio

The "E" in E2B and E4B stands for "effective" parameters. The smaller models incorporate Per-Layer Embeddings (PLE) to maximize parameter efficiency in on-device deployments. Rather than adding more layers or parameters to the model, PLE gives each decoder layer its own small embedding for every token. These embedding tables are large but are only used for quick lookups, which is why the effective parameter count is much smaller than the total.

Mixture-of-Experts (MoE) Model

Property	26B A4B MoE
Total Parameters	25.2B
Active Parameters	3.8B
Layers	30
Sliding Window	1024 tokens
Context Length	256K tokens
Vocabulary Size	262K
Expert Count	8 active / 128 total and 1 shared
Supported Modalities	Text, Image
Vision Encoder Parameters	*~550M*

The "A" in 26B A4B stands for "active parameters" in contrast to the total number of parameters the model contains. By only activating a 4B subset of parameters during inference, the Mixture-of-Experts model runs much faster than its 26B total might suggest. This makes it an excellent choice for fast inference compared to the dense 31B model since it runs almost as fast as a 4B-parameter model.

Benchmark Results

These models were evaluated against a large collection of different datasets and metrics to cover different aspects of text generation. Evaluation results marked in the table are for instruction-tuned models.

	Gemma 4 31B	Gemma 4 26B A4B	Gemma 4 E4B	Gemma 4 E2B	Gemma 3 27B (no think)
MMLU Pro	85.2%	82.6%	69.4%	60.0%	67.6%
AIME 2026 no tools	89.2%	88.3%	42.5%	37.5%	20.8%
LiveCodeBench v6	80.0%	77.1%	52.0%	44.0%	29.1%
Codeforces ELO	2150	1718	940	633	110
GPQA Diamond	84.3%	82.3%	58.6%	43.4%	42.4%
Tau2 (average over 3)	76.9%	68.2%	42.2%	24.5%	16.2%
HLE no tools	19.5%	8.7%	-	-	-
HLE with search	26.5%	17.2%	-	-	-
BigBench Extra Hard	74.4%	64.8%	33.1%	21.9%	19.3%
MMMLU	88.4%	86.3%	76.6%	67.4%	70.7%
Vision					
MMMU Pro	76.9%	73.8%	52.6%	44.2%	49.7%
OmniDocBench 1.5 (average edit distance, lower is better)	0.131	0.149	0.181	0.290	0.365
MATH-Vision	85.6%	82.4%	59.5%	52.4%	46.0%
MedXPertQA MM	61.3%	58.1%	28.7%	23.5%	-
Audio					
CoVoST	-	-	35.54	33.47	-
FLEURS (lower is better)	-	-	0.08	0.09	-
Long Context					

	Gemma 4 31B	Gemma 4 26B A4B	Gemma 4 E4B	Gemma 4 E2B	Gemma 3 27B (no think)
MRCR v2 8 needle 128k (average)	66.4%	44.1%	25.4%	19.1%	13.5%

Core Capabilities

Gemma 4 models handle a broad range of tasks across text, vision, and audio. Key capabilities include:

- **Thinking** – Built-in reasoning mode that lets the model think step-by-step before answering.
- **Long Context** – Context windows of up to 128K tokens (E2B/E4B) and 256K tokens (26B A4B/31B).
- **Image Understanding** – Object detection, Document/PDF parsing, screen and UI understanding, chart comprehension, OCR (including multilingual), handwriting recognition, and pointing. Images can be processed at variable aspect ratios and resolutions.
- **Video Understanding** – Analyze video by processing sequences of frames.
- **Interleaved Multimodal Input** – Freely mix text and images in any order within a single prompt.
- **Function Calling** – Native support for structured tool use, enabling agentic workflows.
- **Coding** – Code generation, completion, and correction.
- **Multilingual** – Out-of-the-box support for 35+ languages, pre-trained on 140+ languages.
- **Audio** (E2B and E4B only) – Automatic speech recognition (ASR) and speech-to-translated-text translation across multiple languages.

Getting Started

You can use all Gemma 4 models with the latest version of Transformers. To get started, install the necessary dependencies in your environment:

```
pip install -U transformers torch accelerate
```

Once you have everything installed, you can proceed to load the model with the code below:

```
from transformers import AutoProcessor, AutoModelForCausalLM

MODEL_ID = "google/gemma-4-E2B-it"

# Load model
processor = AutoProcessor.from_pretrained(MODEL_ID)
model = AutoModelForCausalLM.from_pretrained(
    MODEL_ID,
    dtype="auto",
    device_map="auto"
)
```

Once the model is loaded, you can start generating output:

```
# Prompt
messages = [
    {"role": "system", "content": "You are a helpful assistant."},
    {"role": "user", "content": "Write a short joke about saving RAM."}
]

# Process input
text = processor.apply_chat_template(
    messages,
    tokenize=False,
    add_generation_prompt=True,
    enable_thinking=False
```

```
)
inputs = processor(text=text, return_tensors="pt").to(model.device)
input_len = inputs["input_ids"].shape[-1]

# Generate output
outputs = model.generate(**inputs, max_new_tokens=1024)
response = processor.decode(outputs[0][input_len:], skip_special_tokens

# Parse output
processor.parse_response(response)
```

To enable reasoning, set `enable_thinking=True` and the `parse_response` function will take care of parsing the thinking output.

Below, you will also find snippets for processing audio (E2B and E4B only), images, and video alongside text:

- ▶ Code for processing Audio
- ▶ Code for processing Images
- ▶ Code for processing Videos

Best Practices

For the best performance, use these configurations and best practices:

1. Sampling Parameters

Use the following standardized sampling configuration across all use cases:

- `temperature=1.0`
- `top_p=0.95`
- `top_k=64`

2. Thinking Mode Configuration

Compared to Gemma 3, the models use standard `system`, `assistant`, and `user` roles. To properly manage the thinking process, use the following control tokens:

- **Trigger Thinking:** Thinking is enabled by including the `<|think|>` token at the start of the system prompt. To disable thinking, remove the token.
- **Standard Generation:** When thinking is enabled, the model will output its internal reasoning followed by the final answer using this structure:
  `<|channel>thought\n` **[Internal reasoning]** `<channel|>`
- **Disabled Thinking Behavior:** For all models except for the E2B and E4B variants, if thinking is disabled, the model will still generate the tags but with an empty thought block:
  `<|channel>thought\n<channel|>` **[Final answer]**

> Note that many libraries like Transformers and llama.cpp handle the complexities of the chat template for you.

3. Multi-Turn Conversations

- **No Thinking Content in History**: In multi-turn conversations, the historical model output should only include the final response. Thoughts from previous model turns must *not be added* before the next user turn begins.

4. Modality order

- For optimal performance with multimodal inputs, place image and/or audio content **before** the text in your prompt.

5. Variable Image Resolution

Aside from variable aspect ratios, Gemma 4 supports variable image resolution through a configurable visual token budget, which controls how many tokens are used to represent an image. A higher token budget preserves more visual detail at the cost of

additional compute, while a lower budget enables faster inference for tasks that don't require fine-grained understanding.

- The supported token budgets are: **70**, **140**, **280**, **560**, and **1120**.
    - Use *lower budgets* for classification, captioning, or video understanding, where faster inference and processing many frames outweigh fine-grained detail.
    - Use *higher budgets* for tasks like OCR, document parsing, or reading small text.

6. Audio

Use the following prompt structures for audio processing:

- **Audio Speech Recognition (ASR)**

```
Transcribe the following speech segment in {LANGUAGE} into {LANGUAGE} 

Follow these specific instructions for formatting the answer:
* Only output the transcription, with no newlines.
* When transcribing numbers, write the digits, i.e. write 1.7 and not 
```

- **Automatic Speech Translation (AST)**

```
Transcribe the following speech segment in {SOURCE_LANGUAGE}, then tra
When formatting the answer, first output the transcription in {SOURCE_
```

7. Audio and Video Length

All models support image inputs and can process videos as frames whereas the E2B and E4B models also support audio inputs. Audio supports a maximum length of 30 seconds. Video supports a maximum of 60 seconds assuming the images are processed at one frame per second.

Model Data

Data used for model training and how the data was processed.

Training Dataset

Our pre-training dataset is a large-scale, diverse collection of data encompassing a wide range of domains and modalities, which includes web documents, code, images, audio, with a cutoff date of January 2025. Here are the key components:

- **Web Documents**: A diverse collection of web text ensures the model is exposed to a broad range of linguistic styles, topics, and vocabulary. The training dataset includes content in over 140 languages.
- **Code**: Exposing the model to code helps it to learn the syntax and patterns of programming languages, which improves its ability to generate code and understand code-related questions.
- **Mathematics**: Training on mathematical text helps the model learn logical reasoning, symbolic representation, and to address mathematical queries.
- **Images**: A wide range of images enables the model to perform image analysis and visual data extraction tasks.

The combination of these diverse data sources is crucial for training a powerful multimodal model that can handle a wide variety of different tasks and data formats.

Data Preprocessing

Here are the key data cleaning and filtering methods applied to the training data:

- **CSAM Filtering**: Rigorous CSAM (Child Sexual Abuse Material) filtering was applied at multiple stages in the data preparation process to ensure the exclusion of harmful and illegal content.
- **Sensitive Data Filtering**: As part of making Gemma pre-trained models safe and reliable, automated techniques were used to filter out certain personal information and other sensitive data from training sets.

- **Additional methods**: Filtering based on content quality and safety in line with our policies.

Ethics and Safety

As open models become central to enterprise infrastructure, provenance and security are paramount. Developed by Google DeepMind, Gemma 4 undergoes the same rigorous safety evaluations as our proprietary Gemini models.

Evaluation Approach

Gemma 4 models were developed in partnership with internal safety and responsible AI teams. A range of automated as well as human evaluations were conducted to help improve model safety. These evaluations align with Google's AI principles, as well as safety policies, which aim to prevent our generative AI models from generating harmful content, including:

- Content related to child sexual abuse material and exploitation
- Dangerous content (e.g., promoting suicide, or instructing in activities that could cause real-world harm)
- Sexually explicit content
- Hate speech (e.g., dehumanizing members of protected groups)
- Harassment (e.g., encouraging violence against people)

Evaluation Results

For all areas of safety testing, we saw major improvements in all categories of content safety relative to previous Gemma models. Overall, Gemma 4 models significantly outperform Gemma 3 and 3n models in improving safety, while keeping unjustified refusals low. All testing was conducted without safety filters to evaluate the model capabilities and behaviors. For both text-to-text and image-to-text, and across all model sizes, the model produced minimal policy violations, and showed significant improvements over previous Gemma models' performance.

Usage and Limitations

These models have certain limitations that users should be aware of.

Intended Usage

Multimodal models (capable of processing vision, language, and/or audio) have a wide range of applications across various industries and domains. The following list of potential uses is not comprehensive. The purpose of this list is to provide contextual information about the possible use-cases that the model creators considered as part of model training and development.

- **Content Creation and Communication**
  - **Text Generation**: These models can be used to generate creative text formats such as poems, scripts, code, marketing copy, and email drafts.
  - **Chatbots and Conversational AI**: Power conversational interfaces for customer service, virtual assistants, or interactive applications.
  - **Text Summarization**: Generate concise summaries of a text corpus, research papers, or reports.
  - **Image Data Extraction**: These models can be used to extract, interpret, and summarize visual data for text communications.
  - **Audio Processing and Interaction**: The smaller models (E2B and E4B) can analyze and interpret audio inputs, enabling voice-driven interactions and transcriptions.
- **Research and Education**
  - **Natural Language Processing (NLP) and VLM Research**: These models can serve as a foundation for researchers to experiment with VLM and NLP techniques, develop algorithms, and contribute to the advancement of the field.
  - **Language Learning Tools**: Support interactive language learning experiences, aiding in grammar correction or providing writing practice.

- **Knowledge Exploration**: Assist researchers in exploring large bodies of text by generating summaries or answering questions about specific topics.

Limitations

- **Training Data**
  - The quality and diversity of the training data significantly influence the model's capabilities. Biases or gaps in the training data can lead to limitations in the model's responses.
  - The scope of the training dataset determines the subject areas the model can handle effectively.
- **Context and Task Complexity**
  - Models perform well on tasks that can be framed with clear prompts and instructions. Open-ended or highly complex tasks might be challenging.
  - A model's performance can be influenced by the amount of context provided (longer context generally leads to better outputs, up to a certain point).
- **Language Ambiguity and Nuance**
  - Natural language is inherently complex. Models might struggle to grasp subtle nuances, sarcasm, or figurative language.
- **Factual Accuracy**
  - Models generate responses based on information they learned from their training datasets, but they are not knowledge bases. They may generate incorrect or outdated factual statements.
- **Common Sense**
  - Models rely on statistical patterns in language. They might lack the ability to apply common sense reasoning in certain situations.

Ethical Considerations and Risks

The development of vision-language models (VLMs) raises several ethical concerns. In creating an open model, we have carefully considered the following:

- **Bias and Fairness**
    - VLMs trained on large-scale, real-world text and image data can reflect socio-cultural biases embedded in the training material. Gemma 4 models underwent careful scrutiny, input data pre-processing, and post-training evaluations as reported in this card to help mitigate the risk of these biases.
- **Misinformation and Misuse**
    - VLMs can be misused to generate text that is false, misleading, or harmful.
    - Guidelines are provided for responsible use with the model, see the Responsible Generative AI Toolkit.
- **Transparency and Accountability**
    - This model card summarizes details on the models' architecture, capabilities, limitations, and evaluation processes.
    - A responsibly developed open model offers the opportunity to share innovation by making VLM technology accessible to developers and researchers across the AI ecosystem.

Risks identified and mitigations:

- **Generation of harmful content**: Mechanisms and guidelines for content safety are essential. Developers are encouraged to exercise caution and implement appropriate content safety safeguards based on their specific product policies and application use cases.
- **Misuse for malicious purposes**: Technical limitations and developer and end-user education can help mitigate against malicious applications of VLMs. Educational resources and reporting mechanisms for users to flag misuse are provided.
- **Privacy violations**: Models were trained on data filtered for removal of certain personal information and other sensitive data. Developers are encouraged to

adhere to privacy regulations with privacy-preserving techniques.

- **Perpetuation of biases**: It's encouraged to perform continuous monitoring (using evaluation metrics, human review) and the exploration of de-biasing techniques during model training, fine-tuning, and other use cases.

Benefits

At the time of release, this family of models provides high-performance open vision-language model implementations designed from the ground up for responsible AI development compared to similarly sized models.