AIDC-AI / **Ovis2.5-9B** like 292 Follow AIDC-AI 732

Image-Text-to-Text | Transformers | Safetensors | AIDC-AI/Ovis-dataset

English | Chinese | ovis2_5 | text-generation | MLLM | conversational

custom_code | arxiv:2508.11737 | arxiv:2405.20797 | License: apache-2.0

Train | Deploy | Use this model

Model card | Files xet | Community 12

Downloads last month
92,474

Safetensors Model size 9.17B params | Tensor type BF16 | Chat template | Files info

Inference Providers NEW

Image-Text-to-Text

This model isn't deployed by any Inference Provider. 9 Ask for provider support

Model tree for `AIDC-AI/Ovis2.5-9B`

Adapters 1 model

Dataset used to train `AIDC-AI/Ovis2.5-9B`

Spaces using `AIDC-AI/Ovis2.5-9B` 5

Collection including `AIDC-AI/Ovis2.5-9B`

Ovis2.5 Collection

Our next-generation MLLMs for native-res… • 5 items • Updated Aug 19 • △ 55

Ovis2.5-9B

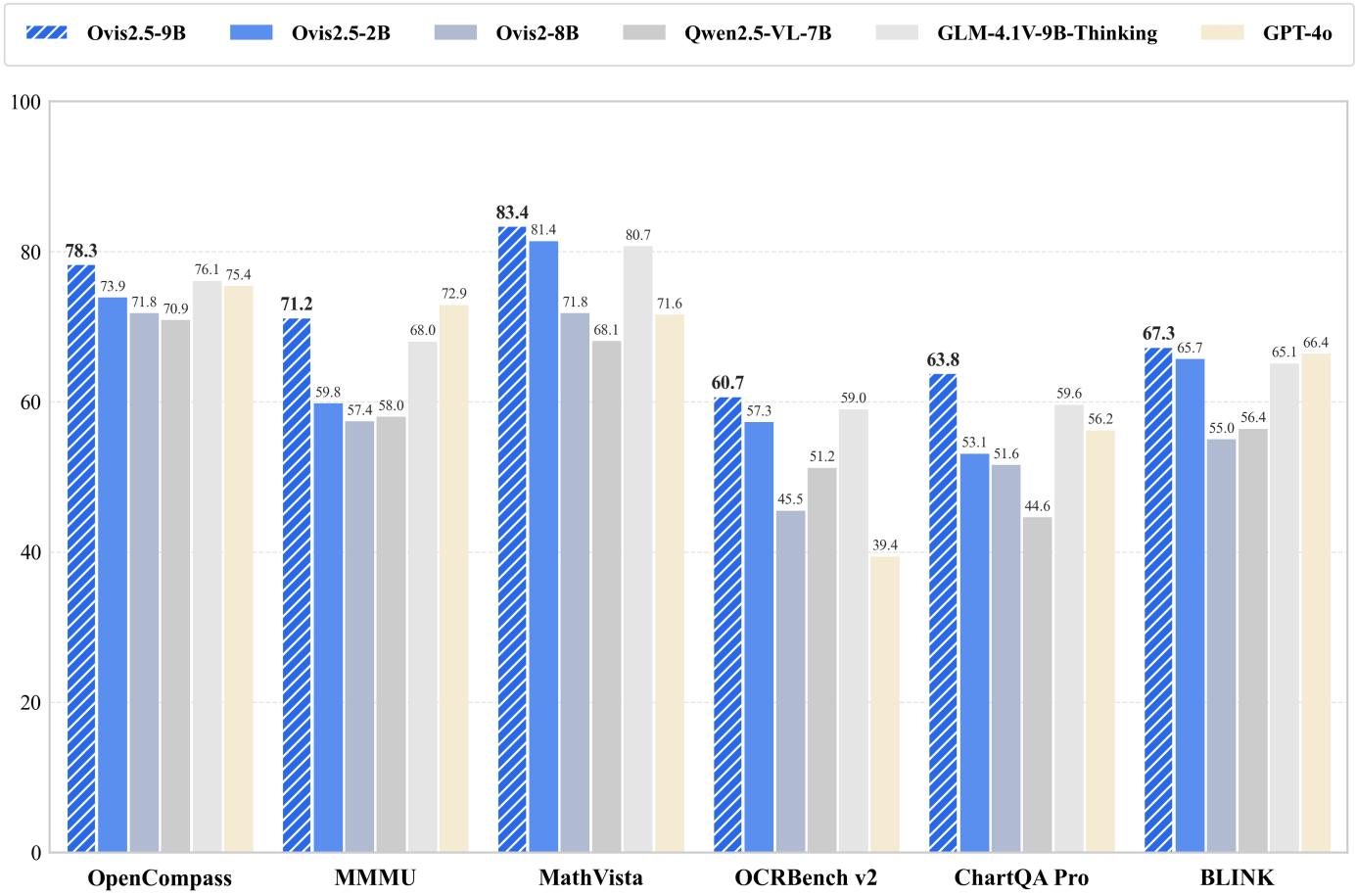
Ovis

Technical Report Ovis2.5 | GitHub AIDC-AI/Ovis | HF Spaces AIDC-AI/Ovis2.5-9B | Models AIDC-AI/Ovis2.5

Introduction

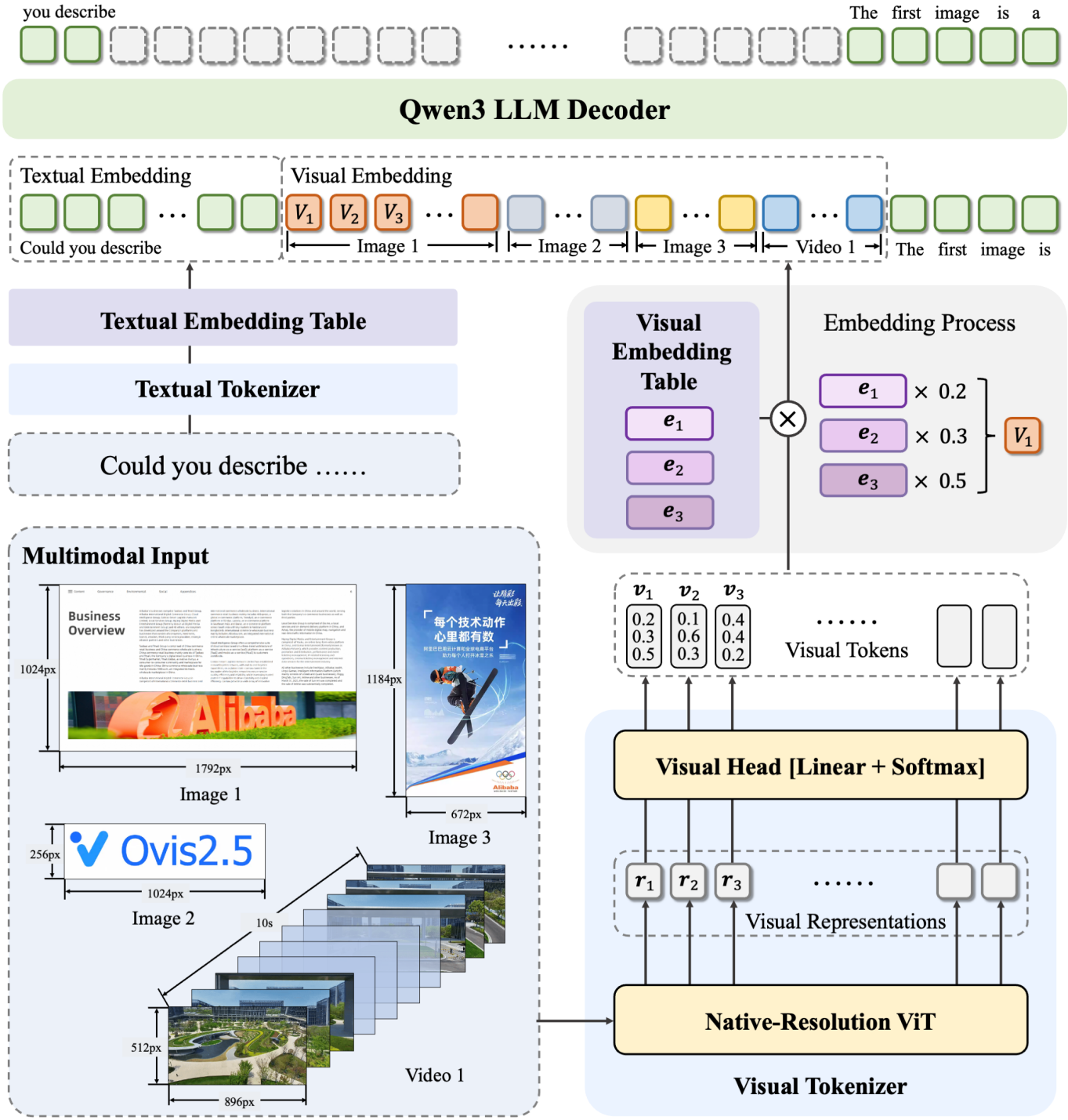
We are pleased to announce the release of **Ovis2.5**, the successor to Ovis2, designed for native-resolution visual perception and enhanced multimodal reasoning. It integrates a native-resolution vision transformer (NaViT) that processes images at their original, variable resolutions, eliminating the need for fixed-resolution tiling and preserving both fine details and global layout—crucial for visually dense content such as charts and diagrams. To strengthen reasoning, Ovis2.5 is trained not only on linear chain-of-thought (CoT) but also on reflective reasoning, including self-checking and revision. This advanced capability is available at inference as an optional *thinking mode*, enabling users to trade latency for higher accuracy on complex inputs.

Building on these advances, **Ovis2.5-9B** achieves an average score of 78.3 on the OpenCompass multimodal evaluation suite (SOTA among open-source MLLMs under 40B parameters), while the lightweight **Ovis2.5-2B** scores 73.9, continuing the “small model, big performance” philosophy for resource-constrained scenarios.



Key Features

- **Native-Resolution Perception** — NaViT vision encoder preserves fine details and global structure without lossy tiling.
- **Deep-Reasoning Capability** — Optional *thinking mode* for self-checking and revision beyond linear CoT. *Thinking budget* supported.
- **Chart & Document OCR** — State-of-the-art at its scale for complex chart analysis, document understanding (including tables and forms), and OCR.
- **Broad Task Coverage** — Demonstrates leading performance on image reasoning, video understanding, and grounding benchmarks, showcasing strong general multimodal capability.



Quick Inference

Below is a simple example demonstrating how to run Ovis2.5 with a single image input. For accelerated inference with **vLLM**, refer to GitHub.

First, install the required dependencies:

```
pip install torch==2.4.0 transformers==4.51.3 numpy==1.25.0 pillow==10.3.0 moviepy
pip install flash-attn==2.7.0.post2 --no-build-isolation
```

Then, run the following code.

```
import torch
import requests
from PIL import Image
from transformers import AutoModelForCausalLM

MODEL_PATH = "AIDC-AI/Ovis2.5-9B"

# Thinking mode & budget
enable_thinking = True
enable_thinking_budget = True  # Only effective if enable_thinking is True.

# Total tokens for thinking + answer. Ensure: max_new_tokens > thinking_budget + 2
max_new_tokens = 3072
thinking_budget = 2048

model = AutoModelForCausalLM.from_pretrained(
    MODEL_PATH,
    torch_dtype=torch.bfloat16,
    trust_remote_code=True
).cuda()

messages = [{
    "role": "user",
    "content": [
        {"type": "image", "image": Image.open(requests.get("https://cdn-uploads.hu
        {"type": "text", "text": "Calculate the sum of the numbers in the middle b
    ],
}]

input_ids, pixel_values, grid_thws = model.preprocess_inputs(
    messages=messages,
    add_generation_prompt=True,
```

```
    enable_thinking=enable_thinking
)
input_ids = input_ids.cuda()
pixel_values = pixel_values.cuda() if pixel_values is not None else None
grid_thws = grid_thws.cuda() if grid_thws is not None else None

outputs = model.generate(
    inputs=input_ids,
    pixel_values=pixel_values,
    grid_thws=grid_thws,
    enable_thinking=enable_thinking,
    enable_thinking_budget=enable_thinking_budget,
    max_new_tokens=max_new_tokens,
    thinking_budget=thinking_budget,
)

response = model.text_tokenizer.decode(outputs[0], skip_special_tokens=True)
print(response)
```

The thinking and thinking budget logic can be applied in the same way for multi-image, video and pure text scenarios.

Note (answer extraction for CoT/Thinking): To make evaluation and usage easier, we recommend appending a fixed suffix to prompts when using chain-of-thought (CoT) or thinking mode. This ensures the model clearly outputs a final answer that can be extracted programmatically:

```
End your response with 'Final answer: '.
```

For example:

```
Calculate the sum of the numbers in the middle box in figure (c).
End your response with 'Final answer: '.
```

Tip: The sections below include an optional streaming helper (compatible with two-phase thinking/budget runs) and extra inference modes: multi-image, video, and text-only.

▶ Optional: Streaming (Advanced)

▶ Example: Multi-image

▶ Example: Video

▶ Example: Text-only

To enable grounding, end your prompt with `Please provide the bounding box coordinates.` (for boxes) or `Please provide the point coordinates.` (for points). To target a specific object, wrap its description in `<ref>` tags, e.g.:

```
Find the <ref>red apple</ref> in the image. Please provide the bounding box coordi
```

Coordinates are normalized to `[0,1)` with the origin `(0,0)` at the top-left corner of the image.

- Point: `<point>(x,y)</point>`
- Bounding box: `<box>(x1,y1),(x2,y2)</box>` where `(x1,y1)` is top-left, `(x2,y2)` is bottom-right.
- Multiple results can be listed in square brackets: `[<box>(...)</box>,<box>(...)</box> ]`

Example:

```
The image features a serene scene with <ref>three birds</ref>[
  <box>(0.401,0.526),(0.430,0.557)</box>,
  <box>(0.489,0.494),(0.516,0.526)</box>,
  <box>(0.296,0.529),(0.324,0.576)</box>
] flying in formation against a clear blue sky.
```

Model Zoo

Ovis MLLMs	ViT	LLM	Model Weights	Demo
Ovis2.5-2B	siglip2-so400m-patch16-512	Qwen3-1.7B	Huggingface	Space
Ovis2.5-9B	siglip2-so400m-patch16-512	Qwen3-8B	Huggingface	Space

Performance

We evaluate Ovis2.5 using VLMEvalKit, as employed in the OpenCompass multimodal and reasoning evaluation suite.

Performance of Ovis2.5-9B and comparison models on the OpenCompass suite. Abbreviations: MMB = MMBenchV11; MMS = MMStar; MMMU = MMMU-Val; HB = HallusionBench; OCR = OCRBench.

Model	MMB	MMS	MMMU	MathVista	HB	AI2D	OCR	MMVet	Avg
Gemini-2.5-Pro	88.3	73.6	74.7	80.9	64.1	89.5	86.2	83.3	80.1
GPT-4o	86.0	70.2	72.9	71.6	57.0	86.3	82.2	76.9	75.4
Ovis2-8B	83.6	64.6	57.4	71.8	56.3	86.6	**89.1**	65.1	71.8
Qwen2.5-VL-7B	82.2	64.1	58.0	68.1	51.9	84.3	88.8	69.7	70.9
InternVL3-8B	82.1	68.7	62.2	70.5	49.0	85.1	88.4	**82.8**	73.6
MiMo-VL-7B-RL-2508	83.9*	72.7*	70.6	79.7*	65.3*	85.3*	88.6	73.4*	77.4*
Keye-VL-8B	79.4*	**75.5**	**71.4**	80.7	**67.0**	86.7	85.1	67.6*	76.7*
GLM-4.1V-9B-Thinking	**85.3**	72.9	68.0	80.7	63.7*	**87.9**	84.2	66.2*	76.1*
Ovis2.5-9B	84.9	72.4	71.2	**83.4**	65.1	87.7	87.9	74.0	**78.3**

Performance of Ovis2.5-9B and comparison models on multimodal reasoning benchmarks. Abbreviations: MPro = MMMU-Pro; MathVerse = MathVerse Vison Only; LV = LogicVista; WM = WeMath; DM = DynaMath.

Model	MMMU	MPro	MathVista	MathVerse	MathVision	LV	WM	DM
Gemini-2.5-Pro	74.7	-	80.9	76.9	69.1	73.8	78.0	56.3
GPT-4o	72.9	-	71.6	49.9	43.8	64.4	50.6	48.5
Ovis2-8B	57.4	34.9	71.8	42.3	25.9	39.4	27.2	20.4
Qwen2.5-VL-7B	58.0	38.3	68.1	41.1	25.4	47.9	36.2	21.8
InternVL3-8B	62.2	42.3*	70.5	38.5	30.0	44.5	39.5	25.7
MiMo-VL-7B-RL-2508	70.6	45.7*	79.7*	**71.6***	**58.5***	**64.5**	65.6*	**48.3***
Keye-VL-8B	**71.4**	39.0*	80.7	59.8	46.0	54.8	60.7	37.3
GLM-4.1V-9B-Thinking	68.0	**57.1**	80.7	68.8*	49.4*	54.1*	63.8	38.9*
Ovis2.5-9B	71.2	54.4	**83.4**	71.1	53.9	61.5	**66.7**	44.1

Citation

If you find Ovis useful, please consider citing the paper

```
@article{lu2025ovis25technicalreport,
  title={Ovis2.5 Technical Report},
  author={Shiyin Lu and Yang Li and Yu Xia and Yuwei Hu and Shanshan Zhao and Yanc
  year={2025},
  journal={arXiv:2508.11737}
```

```
}

@article{lu2024ovis,
  title={Ovis: Structural Embedding Alignment for Multimodal Large Language Model}
  author={Shiyin Lu and Yang Li and Qing-Guo Chen and Zhao Xu and Weihua Luo and K
  year={2024},
  journal={arXiv:2405.20797}
}
```

License

This project is licensed under the Apache License, Version 2.0 (SPDX-License-Identifier: Apache-2.0).

Disclaimer

We used compliance-checking algorithms during the training process, to ensure the compliance of the trained model to the best of our ability. Due to the complexity of the data and the diversity of language model usage scenarios, we cannot guarantee that the model is completely free of copyright issues or improper content. If you believe anything infringes on your rights or generates improper content, please contact us, and we will promptly address the matter.