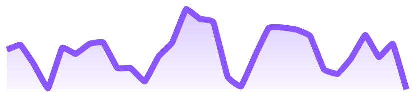
YannQi / **R-4B** like 161

Image-Text-to-Text | Transformers | Safetensors | English | R | feature-extraction | conversational | custom_code | arxiv:2508.21113 | License: apache-2.0

Train | Deploy | Use this model

Model card | Files xet | Community 4

Downloads last month
70,576



Safetensors | Model size 4.82B params | Tensor type BF16 | Chat template | Files info

Inference Providers NEW

Image-Text-to-Text

This model isn't deployed by any Inference Provider. 1 Ask for provider support

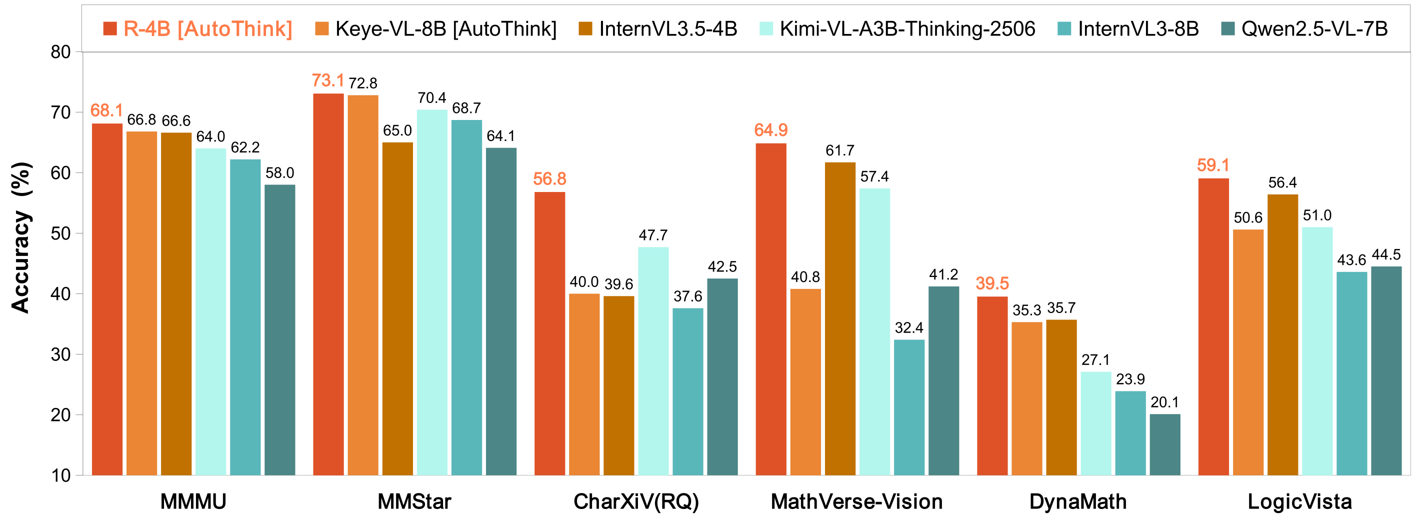
Model tree for `YannQi/R-4B`

Base model	`Qwen/Qwen3-4B-Base`
Finetuned	`Qwen/Qwen3-4B`
Finetuned (265)	this model

Space using `YannQi/R-4B` 1

R-4B: Incentivizing General-Purpose Auto-Thinking Capability in MLLMs via Bi-Mode Annealing and Reinforce Learning

[📗 Arxiv Paper] [🤗 Hugging Face] [🤖 ModelScope] [💻 Code]



⭐ Introduction

In this repo, we present **R-4B**, a multimodal large language model designed for general-purpose auto-thinking, autonomously switching between step-by-step thinking and direct response generation based on task complexity. This capability enables R-4B to deliver high-quality responses while significantly improving inference efficiency and reducing computational costs.

The development of R-4B follows a two-stage training paradigm: (1) Bi-mode Annealing, which establishes both thinking and non-thinking capabilities for VQA; and (2) Bi-mode Policy Optimization (BPO), which enables the model to adaptively switch between thinking and non-thinking modes based on input demands.

🚀 Key Features

- 🧠 **Think Smart, Act Fast: Adaptive & Controllable Thinking!** Our model provides three-mode control over the response process.
  - **Auto-thinking Mode:** Unleash **auto-thinking** that works across general topics, from simple Q&A to complex scientific analysis. It saves time and computation by thinking only when it matters.
  - **Support Manual Control:** Explicitly command the model to use its `thinking` or `non-thinking` capabilities, enabling you to make your choices for every job.

- 🏆 **Strong Performance, Open for Everyone!** Our model is now **fully open-source**. It achieves **state-of-the-art performance** among models of comparable size.

📢 News

- **[2025.08.20]** 🚀 **vLLM Support is Here!** Our R-4B model is now fully compatible with vLLM for high-performance inference.
- **[2025.08.18]** 🏆 **Top Rank Achieved!** We are thrilled to announce that R-4B is now ranked #1 among all open-source models on the OpenCompass Multi-modal Reasoning Leaderboard!
- **[2025.08.11]** 🥇 **Rank #1!** R-4B ranks first under 20B parameters on the OpenCompass Multi-modal Academic Leaderboard!
- **[2025.08.05]** 🎉 **R-4B is Released!** Our model is now publicly available. You can download it from Hugging Face.

🔥 Quickstart

Below, we provide simple examples to show how to use R-4B with 🤗 Transformers.

Using 🤗 Transformers to Chat

> Users can dynamically control the model's response by selecting one of three modes (`auto-thinking`, `thinking`, or `non-thinking`) with `thinking_mode`. `thinking_mode=auto` for `auto-thinking` mode; `thinking_mode=long` for `thinking` mode; `thinking_mode=short` for `non-thinking` mode. Default is `auto-thinking`.

```python
import requests
from PIL import Image
import torch
from transformers import AutoModel, AutoProcessor

model_path = "YannQi/R-4B"

# Load model
model = AutoModel.from_pretrained(
```

```python
    model_path,
    torch_dtype=torch.float32,
    trust_remote_code=True,
).to("cuda")

# Load processor
processor = AutoProcessor.from_pretrained(model_path, trust_remote_code=True)

# Define conversation messages
messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image",
                "image": "http://images.cocodataset.org/val2017/000000039769.jpg",
            },
            {"type": "text", "text": "Describe this image."},
        ],
    }
]

# Apply chat template
text = processor.apply_chat_template(
    messages,
    tokenize=False,
    add_generation_prompt=True,
    thinking_mode="auto"
)

# Load image
image_url = "http://images.cocodataset.org/val2017/000000039769.jpg"
image = Image.open(requests.get(image_url, stream=True).raw)

# Process inputs
inputs = processor(
    images=image,
    text=text,
    return_tensors="pt"
).to("cuda")
```

```python
# Generate output
generated_ids = model.generate(**inputs, max_new_tokens=16384)
output_ids = generated_ids[0][len(inputs.input_ids[0]):]

# Decode output
output_text = processor.decode(
    output_ids,
    skip_special_tokens=True,
    clean_up_tokenization_spaces=False
)

# Print result
print("Auto-Thinking Output:", output_text)
```

Using vLLM for fast R-4B deployment and inference.

- We recommend using vLLM for fast R-4B deployment and inference.

Install

The code of R-4B requires the newest vllm now. Please install from local source:

```bash
git clone https://github.com/vllm-project/vllm.git
cd vllm
VLLM_USE_PRECOMPILED=1 uv pip install --editable .
```

Online Serving

> The `thinking_mode` switch is also available in APIs created by vLLM. Default is `auto-thinking`.

- Serve

```bash
vllm serve \
    yannqi/R-4B \
```

```
    --served-model-name r4b \
    --tensor-parallel-size 8 \
    --gpu-memory-utilization 0.8 \
    --host 0.0.0.0 \
    --port 8000 \
    --trust-remote-code
```

- Openai Chat Completion Client

```python
import base64
from PIL import Image
from openai import OpenAI


# Set OpenAI's API key and API base to use vLLM's API server.
openai_api_key = "EMPTY"
openai_api_base = "http://localhost:8000/v1"

client = OpenAI(
    api_key=openai_api_key,
    base_url=openai_api_base,
)

# image url
image_messages = [
    {
        "role": "user",
        "content": [
            {
                "type": "image_url",
                "image_url": {
                    "url": "http://images.cocodataset.org/val2017/000000039769.jpg
                },
            },
            {"type": "text", "text": "Describe this image."},
        ],
    },
]
```

```python
chat_response = client.chat.completions.create(
    model="r4b",
    messages=image_messages,
    max_tokens=16384,
    extra_body={
        "chat_template_kwargs": {"thinking_mode": "auto"},
    },
)
print("Chat response:", chat_response)
```

📈 Experimental Results

Capability	Benchmark	Qwen2.5-VL -7B (N-T)	InternVL3 -8B (N-T)	InternVL3.5 -4B (T)	Kimi-VL-A3B -Thinking (T)	Keye-VL -8B (A-T)	R-4B-Base (T)	R-4B-RL (A-T)
General Visual QA	MMMU$_{\text{val}}$	58.6	62.7	66.6	64.0	66.8	63.2	**68.1**
	MMMU-Pro	34.7	45.6	-	**49.2**	47.5	46.7	46.5
	MMStar	64.1	68.7	65.0	70.4	72.8	70.8	**73.1**
	MMBenchV1.1-EN$_{\text{dev}}$	82.1	84.7	-	82.6	**89.7**	81.9	84.9
	MMBenchV1.1-CN$_{\text{dev}}$	81.3	83.6	-	80.7	**89.8**	83.2	84.7
	MMVet	69.7	82.8	76.6	81.9	65.5	**85.9**	81.9
	HallusionBench	55.7	49.4	44.8	57.2	57.3	53.9	**58.9**
	VLMs are Blind	37.4	36.8	-	60.8	**61.0**	47.0	52.3
	MMVP	73.3	79.3	-	80.3	79.0	79.3	**80.7**
	VisuLogic	20.0	**26.1**	-	25.0	21.1	22.5	25.1
	RealWorldQA	68.2	**70.6**	66.3	66.1	66.3	70.5	69.1
Table & Chart & OCR	AI2D	83.9	85.2	83.9	82.7	85.8	84.8	**86.2**
	CharXiv (DQ)	73.9	73.6	71.1	75.4	74.5	82.8	**82.9**
	CharXiv (RQ)	42.5	37.6	39.6	47.7	40.0	55.4	**56.8**
	DocVQA$_{\text{val}}$	**95.5**	89.4	92.4	69.0	86.3	89.6	91.0
Visual Perception & Counting	OCRBench	**89.7**	88.0	81.5	86.2	85.3	82.8	83.6
	BLINK$_{\text{val}}$	56.4	55.5	**58.1**	56.2	52.5	54.8	56.3
	CountBench	74.1	80.0	-	91.4	75.4	**92.6**	90.2
Math & Reasoning	MathVision	26.2	28.8	26.2	**56.8**	42.4	45.7	47.8
	MathVista$_{\text{MINI}}$	66.8	70.7	77.1	**80.1**	75.2	76.8	78.0
	MathVerse-vision	41.2	32.4	61.7	57.4	40.8	**65.0**	64.9
	OlympiadBench	19.4	25.9	-	33.9	45.2	47.0	**49.6**
	WeMath	37.7	38.5	50.1	47.0	**58.6**	54.1	52.8
	LogicVista	44.5	43.6	56.4	51.0	50.6	58.8	**59.1**
	DynaMath	20.1	23.9	35.7	27.1	35.3	36.3	**39.5**

1. R-4B establishes itself with powerful, state-of-the-art perceptual abilities that are competitive with larger models.
2. In evaluation sets that require complex logical reasoning and mathematical problem-solving, such as WeMath, MathVerse, and LogicVista, R-4B displays a strong performance curve. This highlights its advanced adaptive thinking capacity for logical deduction and solving complex quantitative problems.

✒️ Citation

```
@misc{yang2025r4bincentivizinggeneralpurposeautothinking,
      title={R-4B: Incentivizing General-Purpose Auto-Thinking Capability in MLLMs
      author={Qi Yang and Bolin Ni and Shiming Xiang and Han Hu and Houwen Peng ar
      year={2025},
      eprint={2508.21113},
      archivePrefix={arXiv},
      primaryClass={cs.CV},
      url={https://arxiv.org/abs/2508.21113},
}
```

Acknowledgements

R-4B is developed based on the codebases of the following projects: LLaVA-Next, SigLIP2, Qwen3, Qwen2.5-VL, VLMEvalKit. We sincerely thank these projects for their outstanding work.