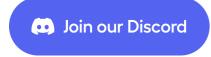


Finetune Llama 3.1, Gemma 2, Mistral 2-5x faster with 70% less memory via Unsloth!

We have a Qwen 2.5 (all model sizes) <u>free Google</u>
<u>Colab Tesla T4 notebook</u>. Also a <u>Qwen 2.5</u>
<u>conversational style notebook</u>.





† Finetune for Free

All notebooks are **beginner friendly**! Add your dataset, click "Run All", and you'll get a 2x faster finetuned model which can be exported to GGUF, vLLM or uploaded to Hugging Face.

Downloads last month **29,863**

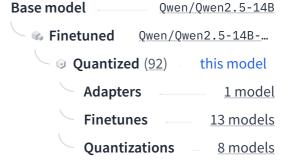


♦ Inference Providers NEW

Text Generation

This model is not currently available via any of the supported third-party Inference Providers, and the model is not deployed on the HF Inference API.





Unsloth supports	Free Notebooks	Performance	Memory use
Llama-3.1 8b	Start on Colab	2.4x faster	58% less
Phi-3.5 (mini)	Start on Colab	2x faster	50% less
Gemma-2 9b	Start on Colab	2.4x faster	58% less
Mistral 7b	Start on Colab	2.2x faster	62% less
TinyLlama	Start on Colab	3.9x faster	74% less
DPO - Zephyr	Start on Colab	1.9x faster	19% less

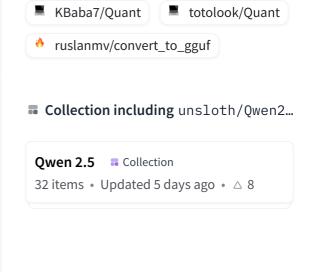
- This <u>conversational notebook</u> is useful for ShareGPT ChatML / Vicuna templates.
- This <u>text completion notebook</u> is for raw text.
 This <u>DPO notebook</u> replicates Zephyr.
- * Kaggle has 2x T4s, but we use 1. Due to overhead, 1x T4 is 5x faster.

Qwen2.5-14B-Instruct

Introduction

Qwen2.5 is the latest series of Qwen large language models. For Qwen2.5, we release a number of base language models and instruction-tuned language models ranging from 0.5 to 72 billion parameters.

Qwen2.5 brings the following improvements upon Qwen2:



- Significantly more knowledge and has greatly improved capabilities in coding and mathematics, thanks to our specialized expert models in these domains.
- Significant improvements in instruction following, generating long texts (over 8K tokens), understanding structured data (e.g, tables), and generating structured outputs especially JSON. More resilient to the diversity of system prompts, enhancing role-play implementation and condition-setting for chatbots.
- Long-context Support up to 128K tokens and can generate up to 8K tokens.
- Multilingual support for over 29 languages, including Chinese, English, French, Spanish, Portuguese, German, Italian, Russian, Japanese, Korean, Vietnamese, Thai, Arabic, and more.

This repo contains the instruction-tuned 14B Qwen2.5 model, which has the following features:

- Type: Causal Language Models
- Training Stage: Pretraining & Post-training
- Architecture: transformers with RoPE, SwiGLU,
 RMSNorm, and Attention QKV bias
- Number of Parameters: 14.7B
- Number of Paramaters (Non-Embedding): 13.1B
- Number of Layers: 48
- Number of Attention Heads (GQA): 40 for Q and 8 for KV
- Context Length: Full 131,072 tokens and generation 8192 tokens
 - Please refer to <u>this section</u> for detailed instructions on how to deploy Qwen2.5 for

handling long texts.

For more details, please refer to our <u>blog</u>, <u>GitHub</u>, and <u>Documentation</u>.

Requirements

The code of Qwen2.5 has been in the latest Hugging face transformers and we advise you to use the latest version of transformers.

With transformers<4.37.0, you will encounter the following error:

```
KeyError: 'qwen2'
```

Quickstart

Here provides a code snippet with apply_chat_template to show you how to load the tokenizer and model and how to generate contents.

```
tokenize=False,
   add_generation_prompt=True
)
model_inputs = tokenizer([text], return_text

generated_ids = model.generate(
   **model_inputs,
   max_new_tokens=512
)
generated_ids = [
   output_ids[len(input_ids):] for input_
]
response = tokenizer.batch_decode(generatext)
```

Processing Long Texts

The current config.json is set for context length up to 32,768 tokens. To handle extensive inputs exceeding 32,768 tokens, we utilize <u>YaRN</u>, a technique for enhancing model length extrapolation, ensuring optimal performance on lengthy texts.

For supported frameworks, you could add the following to config.json to enable YaRN:

```
{
    ...,
    "rope_scaling": {
        "factor": 4.0,
        "original_max_position_embeddings": 32'
        "type": "yarn"
    }
}
```

For deployment, we recommend using vLLM. Please refer to our <u>Documentation</u> for usage if you are not familar with vLLM. Presently, vLLM only supports static YARN, which means the scaling factor remains

constant regardless of input length, **potentially impacting performance on shorter texts**. We advise adding the rope_scaling configuration only when processing long contexts is required.

Evaluation & Performance

Detailed evaluation results are reported in this <u>blog.</u>

For requirements on GPU memory and the respective throughput, see results <u>here</u>.

Citation

}

If you find our work helpful, feel free to give us a cite.

```
@misc{qwen2.5,
    title = {Qwen2.5: A Party of Foundation
    url = {https://qwenlm.github.io/blog/quauthor = {Qwen Team},
    month = {September},
    year = {2024}
}
@article{qwen2,
    title={Qwen2 Technical Report},
```

```
System theme TOS Privacy About Johs 1={arXiv preprint arXiv:2407.1| year={2024}
```

Models Datasets S

Spaces Pricing

Docs