You are viewing the latest developer preview docs. Click here to view docs for the latest stable release.

Welcome to vLLM



Easy, fast, and cheap LLM serving for everyone

Star 97,754 | Watch 609 | Fork 25,414

vLLM is a fast and easy-to-use library for LLM inference and serving.

Originally developed in the Sky Computing Lab at UC Berkeley, vLLM has grown into one of the most active open-source AI projects built and maintained by a diverse community of many dozens of academic institutions and companies from over 2000 contributors.

Where to get started with vLLM depends on the type of user. If you are looking to:

- Run open-source models on vLLM, we recommend starting with the Quickstart Guide
- Build applications with vLLM, we recommend starting with
- Build vLLM, we recommend starting with Developer Guide

Ask AI Ctrl J

For information about the development of vLLM, see:

latest

- Roadmap

- Releases

vLLM is fast with:

- State-of-the-art serving throughput
- Efficient management of attention key and value memory with **PagedAttention**
- Continuous batching of incoming requests, chunked prefill, prefix caching
- Fast and flexible model execution with piecewise and full CUDA/HIP graphs
- Quantization: FP8, MXFP8/MXFP4, NVFP4, INT8, INT4, GPTQ/AWQ, GGUF, compressed-tensors, ModelOpt, TorchAO, and more
- Optimized attention kernels including FlashAttention, FlashInfer, TRTLLM-GEN, FlashMLA, and Triton
- Optimized GEMM/MoE kernels for various precisions using CUTLASS, TRTLLM-GEN, CuTeDSL
- Speculative decoding including n-gram, suffix, EAGLE, DFlash
- Automatic kernel generation and graph-level transformations using torch.compile
- Disaggregated prefill, decode, and encode

vLLM is flexible and easy to use with:

- Seamless integration with popular Hugging Face models
- High-throughput serving with various decoding algorithms, including *parallel sampling*, *beam search*, and more
- Tensor, pipeline, data, expert, and context parallelism for distributed inference
- Streaming outputs
- Generation of structured outputs using xgrammar or guidance
- Tool calling and reasoning parsers
- OpenAI-compatible API server, plus Anthropic Messages AP[illegible] and gRPC support
- Efficient multi-LoRA support for dense and MoE layers

latest

- Support for NVIDIA GPUs, AMD GPUs, and x86/ARM/PowerPC CPUs. Additionally, diverse hardware plugins such as Google TPUs, Intel Gaudi, IBM Spyre, Huawei Ascend, Rebellions NPU, Apple Silicon, MetaX GPU, Iluvatar CoreX GPU, and more.

vLLM seamlessly supports 200+ model architectures on HuggingFace, including:

- Decoder-only LLMs (e.g., Llama, Qwen, Gemma)
- Mixture-of-Expert LLMs (e.g., Mixtral, DeepSeek-V3, Qwen-MoE, GPT-OSS)
- Hybrid attention and state-space models (e.g., Mamba, Qwen3.5)
- Multi-modal models (e.g., LLaVA, Qwen-VL, Pixtral)
- Embedding and retrieval models (e.g., E5-Mistral, GTE, ColBERT)
- Reward and classification models (e.g., Qwen-Math)

Find the full list of supported models here.

For more information, check out the following:

- vLLM announcing blog post (intro to PagedAttention)
- vLLM paper (SOSP 2023)
- How continuous batching enables 23x throughput in LLM inference while reducing p50 latency by Cade Daniel et al.
- vLLM Meetups

November 6, 2026

Made with Material for MkDocs

latest