SGLang DeepSeek MLA

Ke Bao<bockbao@gmail.com>, Yineng Zhang<me@zhyncs.com>
SGLang Team

Outlines

1. Introduction to MLA
2. SGLang MLA Optimizations
3. How to Use MLA in SGLang
4. Future Work

➔ 1. Introduction to MLA

What is MLA？

MLA (**M**ulti-head **L**atent **A**ttention)[1] is an innovative attention architecture introduced by the DeepSeek-AI team.

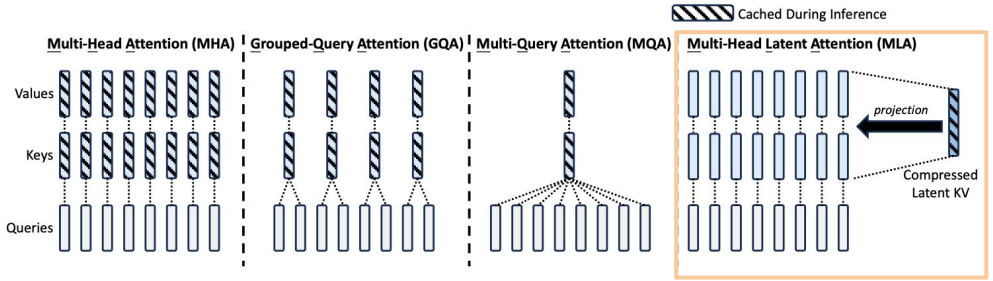
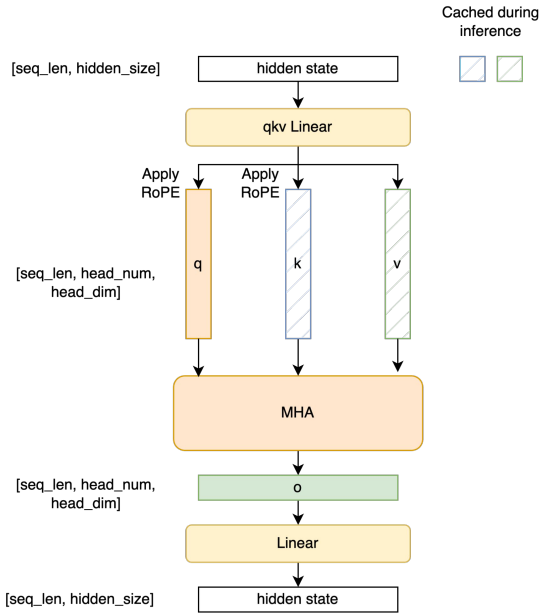


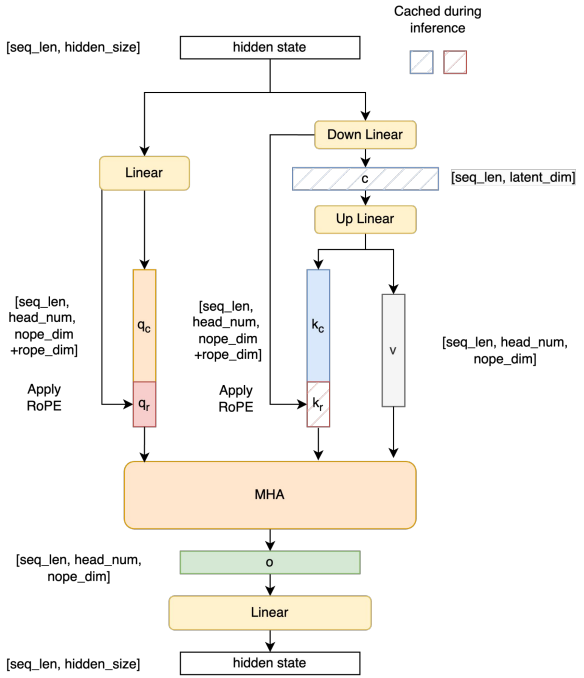
Figure 3 | Simplified illustration of Multi-Head Attention (MHA), Grouped-Query Attention (GQA), Multi-Query Attention (MQA), and Multi-head Latent Attention (MLA). Through jointly compressing the keys and values into a latent vector, MLA significantly reduces the KV cache during inference.

[1]DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model (https://arxiv.org/pdf/2405.04434)

Computation Overview



MHA



MLA

➔ 2. SGLang MLA Optimizations

Weight Absorption

Change the computation order based on **associative law** of matrix multiplication.

$$q^\top k = [q_{nope}^\top; q_{rope}^\top][k_{nope}; k_{rope}]$$
$$= q_{nope}^\top k_{nope} + q_{rope}^\top k_{rope}$$

① orignal order

② absorbed order

$q_{nope}^\top (W_{UK} c)$

$[n_q, h, d] \quad [n_k, h, d]$

$(q_{nope}^\top W_{UK}) c$

$[n_q, h, d_c] \quad [n_k, 1, d_c]$

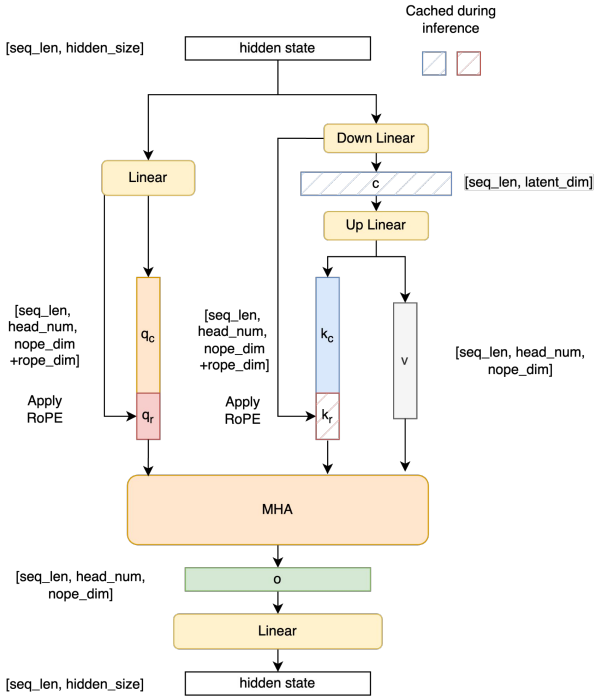
FLOPs: $(2d_c - 1)hdn_k + (2d - 1)hn_q n_k$ $(2d - 1)hn_q d_c + (2d_c - 1)hn_q n_k$

In DeepSeek-V2,

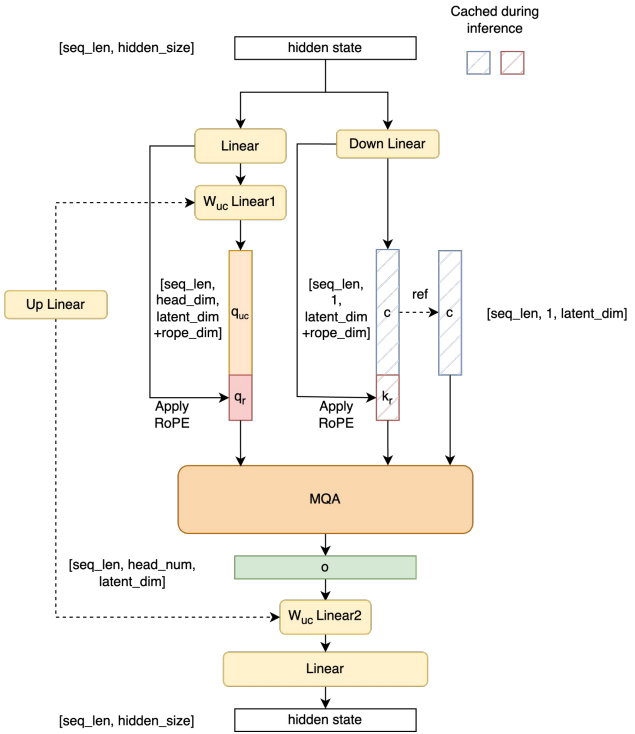
- d = 128
- d_c = 512
- h = 128

In decoding stage (n_q=1), the method ② can take **less computation**.

Weight Absorption



Original



w/ Weight Absorption

Weight Absorption

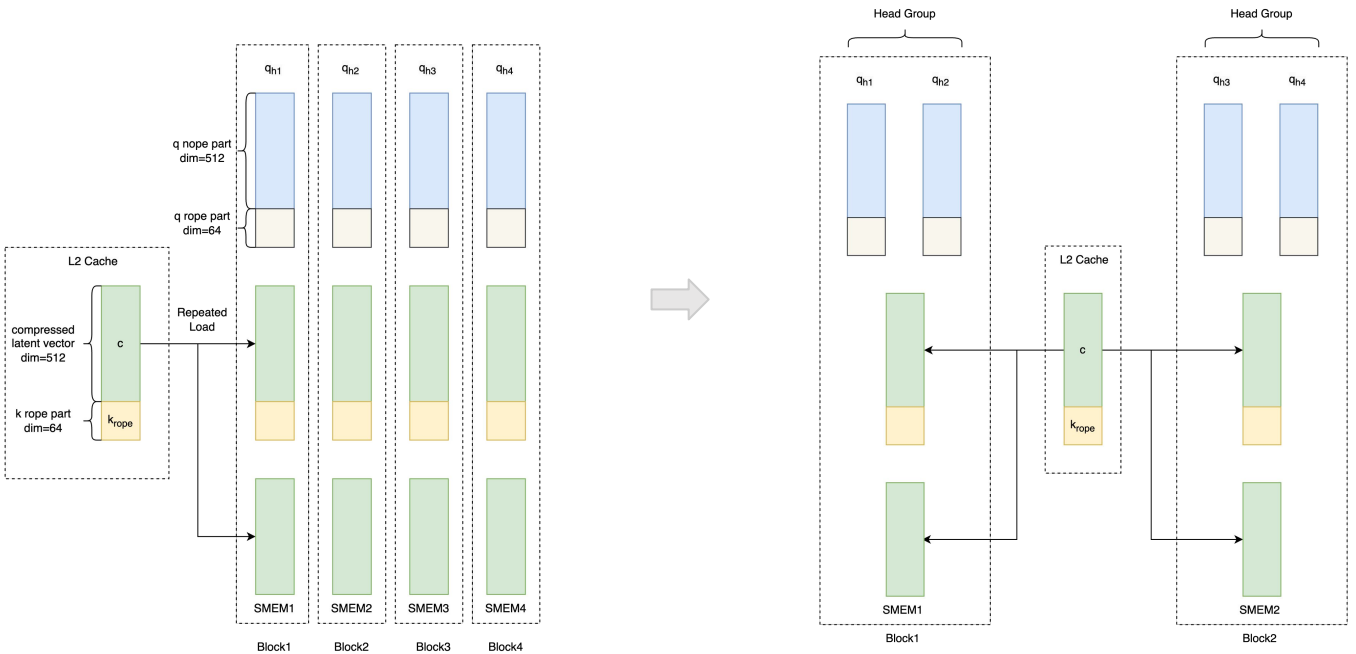
Benefits:

- Reduced overall computation in decoding stage
- Balanced the computation and memory access in decoding kernel
  - Increased the attention computation intensity
  - Reduced the memory access of kv cache

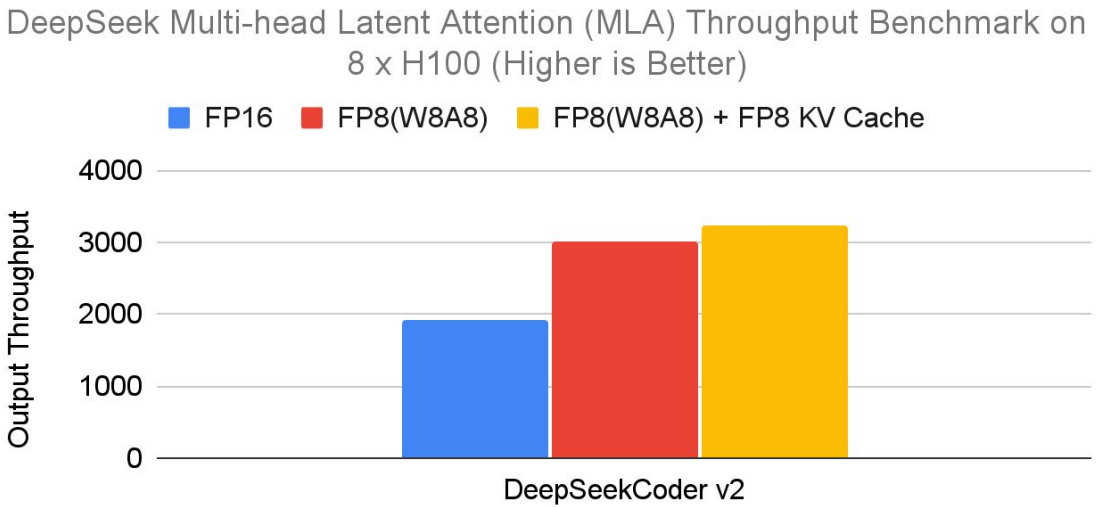
Related PR: #905

Groupped Decoding Kernel

We optimized the memory access of [Triton decoding kernel](#) for MLA.



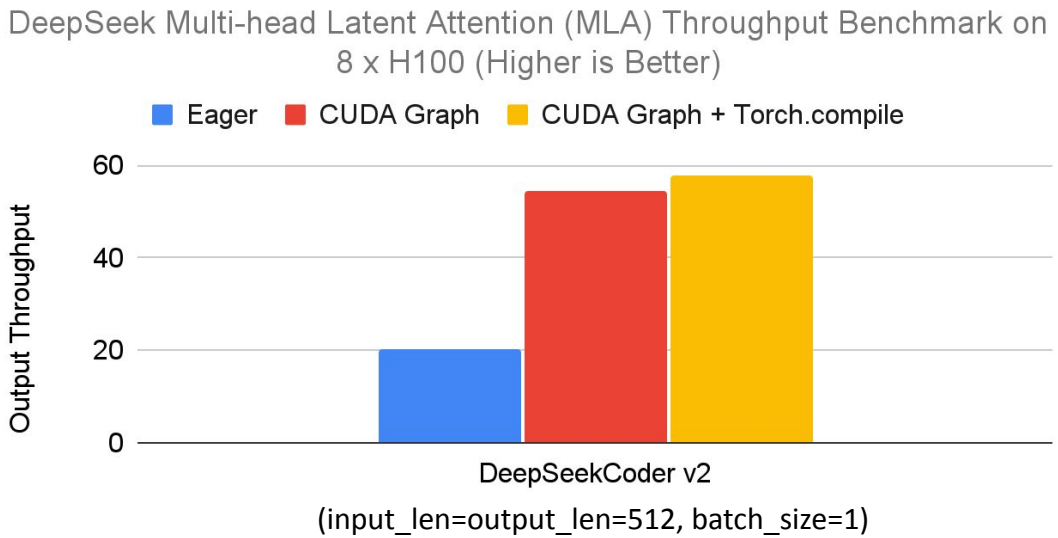
Quantization



FP8 batched MatMul and FP8 E5M2 KV Cache are implemented

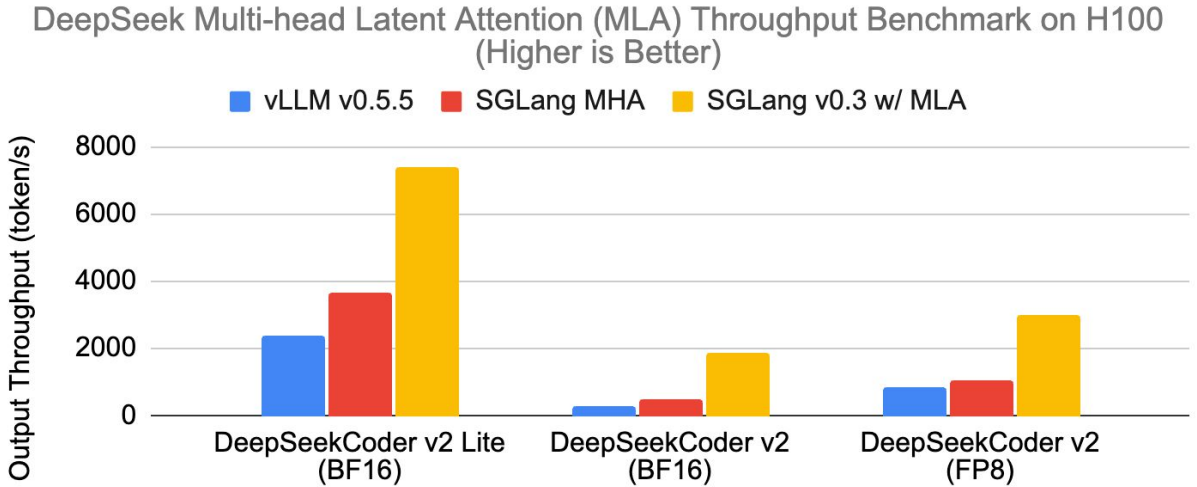
Related PR: #469, #1285, #1286

CUDA Graph & Torch Compile



MLA & MoE are compatible with CUDA Graph & Torch.compile
Related PR: #1401, #1469, #1442, #1497

End2end Benchmark



source and setup: https://lmsys.org/blog/2024-09-04-sglang-v0-3/

➔ 3. How to Use MLA in SGLang

How to Use MLA in SGLang

Recommend using the **latest** version (>=v0.3.1.post3). MLA are enabled **by default**.

```
# fp16 tp8

python3 -m sglang.launch_server --model deepseek-ai/DeepSeek-Coder-V2-Instruct --tp 8
--trust-remote-code

# fp16 tp8 w/ torch compile

python3 -m sglang.launch_server --model deepseek-ai/DeepSeek-Coder-V2-Instruct --tp 8
--trust-remote-code  --enable-torch-compile

# fp16 tp8 w/ torch compile, max torch compile batch size 1

python3 -m sglang.launch_server --model deepseek-ai/DeepSeek-Coder-V2-Instruct --tp 8
--trust-remote-code --enable-torch-compile --max-torch-compile-bs 1

# fp8 tp8 w/ fp8 e5m2 kv cache

python3 -m sglang.launch_server --model neuralmagic/DeepSeek-Coder-V2-Instruct-FP8 --tp 8
--trust-remote-code --kv-cache-dtype fp8_e5m2
```

➔ 4. Future Work

Future Work

- Attention use DP, MoE use TP or EP
- Separate prefill and decoding
- Kernel optimizations (FlashInfer support)
- FP8 KV Cache support E4M3

Welcome to join our Slack and use SGLang!