

From Hopper to Blackwell: performance, optimization, and a reproducible training workflow

671B

DeepSeek-V3
large-scale MoE training

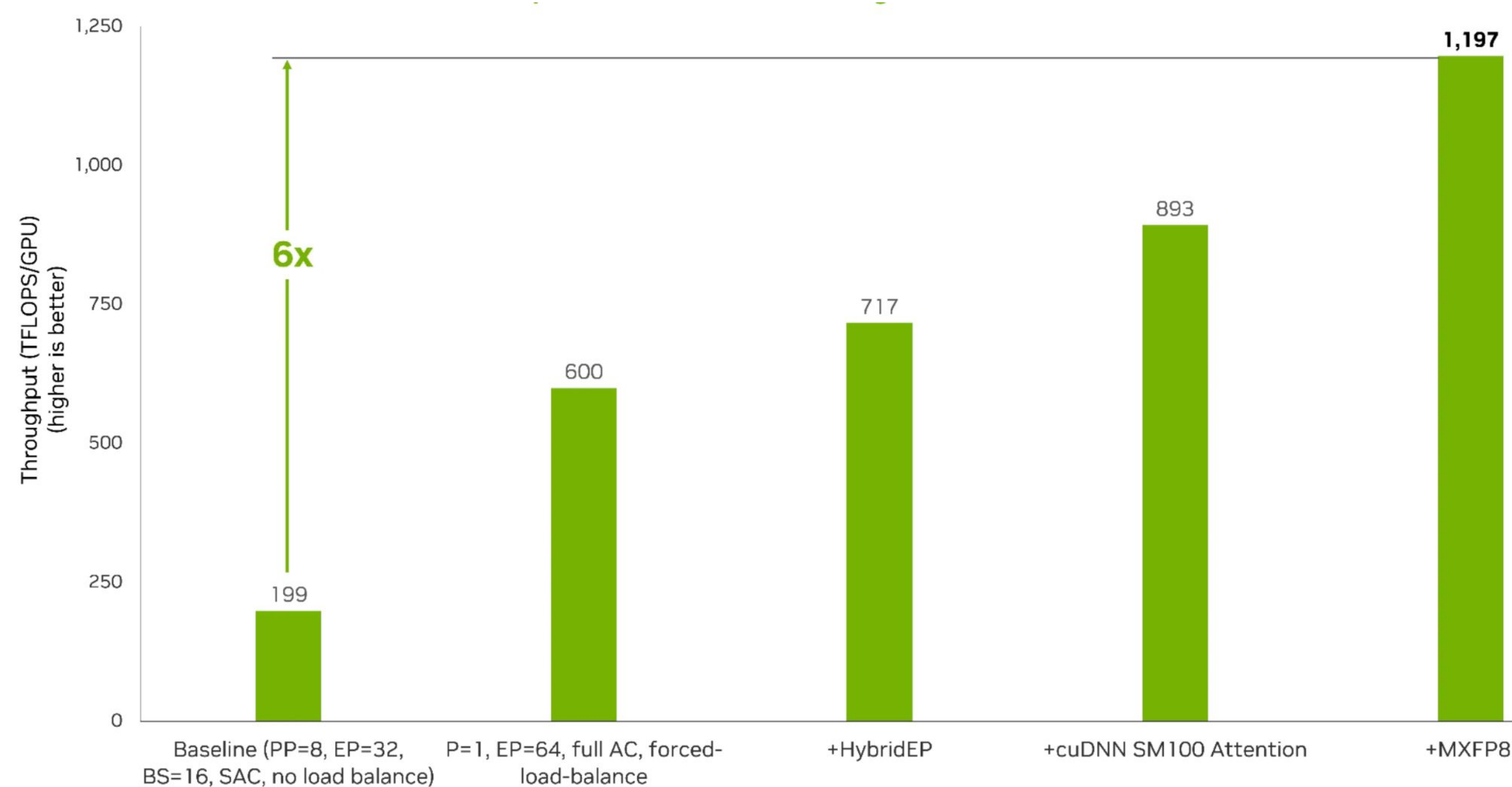
6x

GB200 improvement
over the original baseline

1200

TFLOPS/GPU
latest GB300 as of July 2026

01 Incremental performance improvement



RESULT AS OF JULY 2026 / GB300

1197

TFLOPS/GPU

256 GPUs · DeepSeek-V3 671B
MXFP8 GEMMs / BF16 attention · HybridEP

02 What unlocked performance

Reduce communication overhead

Reconfigure parallelism and use topology-aware HybridEP for expert dispatch and combine.

Use kernels specialized for Blackwell

cuDNN SM100 attention, TorchAO MXFP8 and TE multi-stream grouped GEMMs.

Keep GPU execution moving

Compile quantization and reduce device-to-host synchronization in the grouped GEMM path.

03 Scaling Efficiency



03 A repeatable user workflow

1 Prepare

Automated container setup, pinned versions, and tokenizer + dataset downloads.

2 Configure and launch

GPU-specific defaults with configurable precision, scale and parallelism.

3 Measure and iterate

Organized logs and optional profiling, TensorBoard and Weights & Biases.

ONE LAUNCH INTERFACE / after setup

```
llmb-run submit \
  -w pretrain_deepseek-v3-torchtitan \
  -s 671b --dtype bf16 --scale 256
```

Built on the PyTorch ecosystem

TorchTitan + distributed parallelism

torch.compile + TorchAO

NVIDIA kernels + expert communication

BENCHMARK CONTEXT

GB200 optimization study and recipe benchmarks use forced load balancing. Throughput is not a convergence result.

GB300: sequence 4,096 · GBS 4,096 · MBS 1 · PP2 / DP128. Checkpointing and monitoring are optional in the packaged recipe.

Sources: optimization study / benchmark recipe

[Optimization study](#)

[Benchmark recipe](#)

github.com/pytorch/torchtitan