

# MEGATRON

## Large-Scale Language Model Training Framework

### TRAINING FRAMEWORK [megatron/training/]

Training Loop  
*training.py*

Arguments  
*arguments.py*

Checkpointing  
*checkpointing.py*

Initialization  
*initialize.py*

### MODELS

GPT  
*core/models/gpt/*

BERT  
*core/models/bert/*

T5  
*core/models/t5/*

Mamba  
*core/models/mamba/*

Multimodal  
*core/models/multimodal/*

MoE  
*core/models/mimo/*

### TRANSFORMER CORE

Attention  
*core/transformer/attention.py*

MLP  
*core/transformer/mlp.py*

Layer Norm  
*core/transformer/*

Embeddings  
*core/transformer/*

Block  
*core/transformer/*

### PARALLELISM STRATEGY [megatron/core/]

#### TENSOR PARALLELISM

Intra-layer model sharding



*core/tensor\_parallel/ | layers.py | mappings.py*

#### PIPELINE PARALLELISM

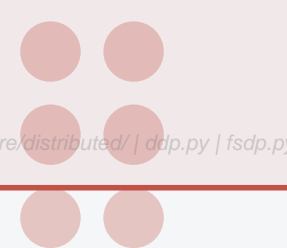
Inter-layer distribution



*core/pipeline\_parallel/ | schedules.py*

#### DATA PARALLELISM

Gradient synchronization



*core/distributed/ | ddp.py | fsdp.py*

### DATA PIPELINE

Blended Dataset  
*core/datasets/blended\_*

Indexed Dataset  
*core/datasets/indexed\_*

GPT Dataset  
*core/datasets/gpt\_*

Data Loader  
*training/datasets/*

### OPTIMIZER

Distributed Optimizer  
*core/optimizer/*

Adam Config  
*core/optimizer/*

Gradient Scaling  
*core/optimizer/*

### CHECKPOINTING [training/]

Save/Load  
*training/*

Distributed  
*core/dist\_checkpointing/*

Model State  
*training/*

### KEY FEATURES [megatron/core/]

#### PRECISION SUPPORT

- FP16 / BF16 Training
- FP8 (Hopper Optimized)
- FP4 Quantization
- Transformer Engine

#### MEMORY OPTIMIZATION

- Activation Checkpointing
- Sequence Parallelism
- Gradient Checkpointing
- Fine-grained Offloading

#### PERFORMANCE

- CUDA Graphs
- Fused Kernels
- Flash Attention
- Fused LayerNorm

### ARCHITECTURAL PATTERNS

#### Modular Core Library

`megatron/core/` provides production-ready, GPU-optimized building blocks for framework developers

*models | transformer | tensor\_parallel | pipeline\_parallel | distributed | optimizer*

#### Configuration-Driven

Dataclass-based configuration system enables flexible model parallelism setup via command-line arguments

*TransformerConfig | ModelParallelConfig | ProcessGroupCollection | parallel\_state*