Transformer Parallelism: A Visual, Dimension-Oriented Guide

November 4, 2025

Contents

1 Neural Network Basics

In this section, we briefly introduce the core ideas of neural networks for readers with no prior background. We cover the notion of tensors, linear layers, activation functions, and the backpropagation algorithm at a high level.

1.1 What is a Neural Network?

Here we describe the basic idea of a neural network as a composition of linear transformations and non-linear activation functions, operating on vector or matrix inputs.

1.2 Fully-Connected Layers and MLPs

We introduce the multi-layer perceptron (MLP):

- Input/output shapes $[B, D]$.
- Linear layer $W \in \mathbb{R}^{D_{\text{in}} \times D_{\text{out}}}$, bias $b \in \mathbb{R}^{D_{\text{out}}}$.
- Activation functions such as ReLU or GELU.

This provides the foundation for understanding the Transformer MLP block.

1.3 Backpropagation at a Glance

We explain the key idea of backpropagation:

- Gradients flowing from the loss to earlier layers.
- Parameter updates using optimizers such as SGD or Adam.

The detailed backward diagrams in later sections are concrete instances of this general principle.

2 From Neural Networks to Transformers

2.1 Sequence Modeling Motivation

We consider sequence modeling tasks such as natural language processing, time series forecasting, and speech processing. In these settings the input is not a single vector but a *sequence* of tokens, each of which is mapped to an embedding. The model must capture dependencies both between nearby tokens and between tokens that are far apart in the sequence.

2.2 The Self-Attention Idea

Transformers address sequence modeling by using self-attention instead of explicit recurrence. At a high level:

- Each token is mapped to an input embedding.
- Self-attention allows every position to attend to every other position in the same sequence.
- Multi-head attention (MHA) uses several attention "heads" in parallel to capture different types of relationships.
- A position-wise feed-forward network (MLP block) processes each token independently after attention.
- Layer normalization, residual connections, and an output projection tie the blocks together and produce final logits or predictions.

In the following sections we make these ideas concrete using dimension-annotated diagrams of a Transformer layer and its parallel variants.

3 Notation and Diagram Conventions

Before diving into layer-wise forward and backward diagrams, we summarize the notational conventions and visual elements used throughout the paper. All later figures (single-node, TP, DP, and DP+TP overall flow) follow these conventions.

3.1 Tensor Shapes, Indices, and Local Views

We use the following global symbols and dimensions:

- B: global batch size.
- S: sequence length (number of tokens).
- D: model (hidden) dimension.
- D_{ff}: intermediate MLP dimension.
- N_H: number of attention heads.
- D_h: per-head dimension, typically $D_h = D/N_H$.
- N_T: tensor-parallel degree (number of TP shards).
- N_D: data-parallel degree (number of DP replicas).

In formulas we write, for example, $\mathbf{X} \in \mathbb{R}^{B \times S \times D}$, while in the diagrams we annotate edges with a compact bracket notation such as $[B, S, D]$. Throughout the figures we use:

- $[B, S, D] \leftrightarrow \mathbb{R}^{B \times S \times D}$
- $[B, N_H, S, D_h] \leftrightarrow \mathbb{R}^{B \times N_H \times S \times D_h}$
- $[B, N_H, S, S] \leftrightarrow \mathbb{R}^{B \times N_H \times S \times S}$

In parallel settings we also show *local* shapes from the point of view of a single node or shard:

- Data parallel (DP): each replica processes a fraction of the batch, with local tensors of shape approximately $[B/N_D, S, D]$.
- Tensor parallel (TP): hidden or head dimensions are partitioned across shards, leading to shapes such as $[B, S, D/N_T]$ or $[B, N_H, S, D_h/N_T]$.

Thus edge labels such as $[B/N_D, S, D/N_T]$ should be read as "the tensor shape seen by a single node or shard".

Typical tensor shapes used in the figures:

- $\mathbf{X} \in \mathbb{R}^{B \times S \times D}$: input or hidden states (annotated as $[B, S, D]$ on edges).
- $\mathbf{H} \in \mathbb{R}^{B \times S \times D}$: normalized or intermediate hidden states.
- $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{B \times N_H \times S \times D_h}$: query, key, value tensors after projection (annotated as $[B, N_H, S, D_h]$).
- $\mathbf{AS} \in \mathbb{R}^{B \times N_H \times S \times S}$: attention scores after scaling and softmax (annotated as $[B, N_H, S, S]$).
- $\mathbf{Y} \in \mathbb{R}^{B \times S \times D}$: output of a Transformer block or layer.

Gradients are denoted by a leading d, e.g. d$\mathbf{X}$, d$\mathbf{W}$, or $\mathbf{dA}_{\text{out}}$. In the diagrams we use the same symbols but typically omit the explicit set notation.

3.2 Computation Nodes and Local Operations

All single-node, TP, DP, and DP+TP figures share the same visual vocabulary for computation nodes.

Matrix multiplication (matmul)

- Symbol: **circle containing "•"** (denoted `mulnode` in TikZ).
- Meaning: linear operations of the form $\mathbf{A}\mathbf{W}$ or $\mathbf{W}^T\mathbf{A}$.
- Examples:
  - MHA Q/K/V projections: $\mathbf{Q} = \mathbf{X}_{\text{norm}}\widetilde{\mathbf{W}}_Q$, $\mathbf{K} = \mathbf{X}_{\text{norm}}\widetilde{\mathbf{W}}_K$, $\mathbf{V} = \mathbf{X}_{\text{norm}}\widetilde{\mathbf{W}}_V$.
  - MLP up/down projections: $\mathbf{H}_{\text{inter}} = \mathbf{H}\widetilde{\mathbf{W}}_{up}$, $\mathbf{Y} = \mathbf{H}_{\text{inter}}\widetilde{\mathbf{W}}_{down}$.

Elementwise (bitwise) addition

- Symbol: **circle containing "+"** (denoted `addnode`). item Meaning: elementwise addition of tensors with the same shape.
- Typical usage:
  - bias addition: $\mathbf{Z} + \mathbf{b}$,
  - residual connections: $\mathbf{X} + \text{Block}(\mathbf{X})$,
  - summing multiple gradient contributions: $\mathbf{dX} = \mathbf{dX}_1 + \mathbf{dX}_2 + \ldots$.

Nonlinear and pointwise operations

- Symbol: **rectangular node** (often `auxnode`).
- Representative labels:
  - `Softmax`, `dSM`, `dS`: softmax on attention scores and its backward pass.
  - `GLU`, `GELU`, `ReLU`, `dGL`: MLP nonlinearities and their backward passes.
  - `LN`, `dLN`: layer normalization forward / backward.
  - `DO`: dropout forward / backward.
- These operators act pointwise (or per feature) and typically preserve the overall tensor shape.

Broadcast (BC)

- Symbol: rectangular node labeled `BC`.
- Meaning: broadcasting a lower-rank tensor along one or more dimensions so that it can be added elementwise to a higher-rank tensor. A common example is broadcasting a bias $\mathbf{b} \in \mathbb{R}^D$ to match a tensor of shape $[B, S, D]$.

Summation over batch/sequence

- Symbol: **small circle containing "$\sum$"** (denoted `sumnode`).
- Meaning: explicit summation over specified axes, typically batch and sequence.
- Typical use: computing bias gradients such as $\mathrm{d}\tilde{\mathbf{b}} = \sum_{B,S} \mathrm{d}\mathbf{Z}$.

Reshape and transpose

- **R**: reshape or dimension reordering operator. For example, switching between $[B, S, N_H, D_h]$ and $[B, N_H, S, D_h]$.
- **T**: transpose operator, often exchanging the last two dimensions, e.g. $[B, N_H, S, D_h] \to [B, N_H, D_h, S]$.

3.3 Parallel Groups and Node-Level View

The DP, TP, and DP+TP overall flow figures show not only layer-internal operations but also the structure across nodes and shards.

- **Data-parallel replicas**: nodes labeled "Node 0", "Node 1", …, "Node $N_D - 1$" form a DP group. Each replica holds a full copy of the model parameters but processes a different slice of the batch. Local tensors are typically annotated as $[B/N_D, S, D]$.
- **Tensor-parallel shards**: within a node (or within a TP group) weight matrices or hidden dimensions are split across N_T shards. Local hidden states may be annotated as $[B, S, D/N_T]$ or $[B, N_H, S, D_h/N_T]$.
- **Hybrid DP+TP**: in the DP+TP overall flow, each data-parallel replica contains a tensor-parallel group. TP communication happens inside each replica, while DP communication happens across replicas.

3.4 Arrow Styles, Communication, and Cost Annotations

Activation and gradient flow

- **Solid arrows**: forward activations (e.g. $\mathbf{X} \to \mathbf{H}$).
- **Dashed arrows**: backward gradients (e.g. $\mathrm{d}\mathbf{Y} \to \mathrm{d}\mathbf{X}$).
- **Double arrows**: reuse of cached tensors from the forward pass, such as weights or attention scores read again during the backward pass.

Collective communication (AR / AG) Tensor-parallel (e.g. MHA backward under TP) and data-parallel (e.g. DP and DP+TP overall flow) figures contain explicit collective communication nodes:

- **AR** (All-Reduce): rectangular node labeled `AR`. All nodes in a group exchange local tensors, compute a global reduction (typically sum or mean), and receive the reduced result. For example, shard-wise weight gradients $\mathbf{d}\widetilde{\mathbf{W}}_Q^{(t)}$ are all-reduced to form the global gradient.
- **AG** (All-Gather): rectangular node labeled `AG`. Each shard contributes a slice of a tensor; the full tensor is reconstructed by concatenation and made available to all shards. A typical example is gathering TP-partitioned hidden states or attention outputs.

DP vs TP communication in DP+TP overall flow In the DP+TP overall flow diagram we distinguish carefully between data-parallel and tensor-parallel communication:

- **DP communication**: All-Reduce across data-parallel replicas, usually for weight gradients. Edge annotations may show both the local payload shape and an algorithm-dependent cost, e.g.

$$\text{Naive: } 2(N_D - 1) \times [B/N_D, S, D], \quad \text{Ring: } 2\frac{N_D - 1}{N_D} \times [B/N_D, S, D].$$

- **TP communication**: All-Reduce or All-Gather across tensor-parallel shards inside a node or TP group, with shapes such as $[B, S, D/N_T]$ or $[B, N_H, S, D_h/N_T]$.

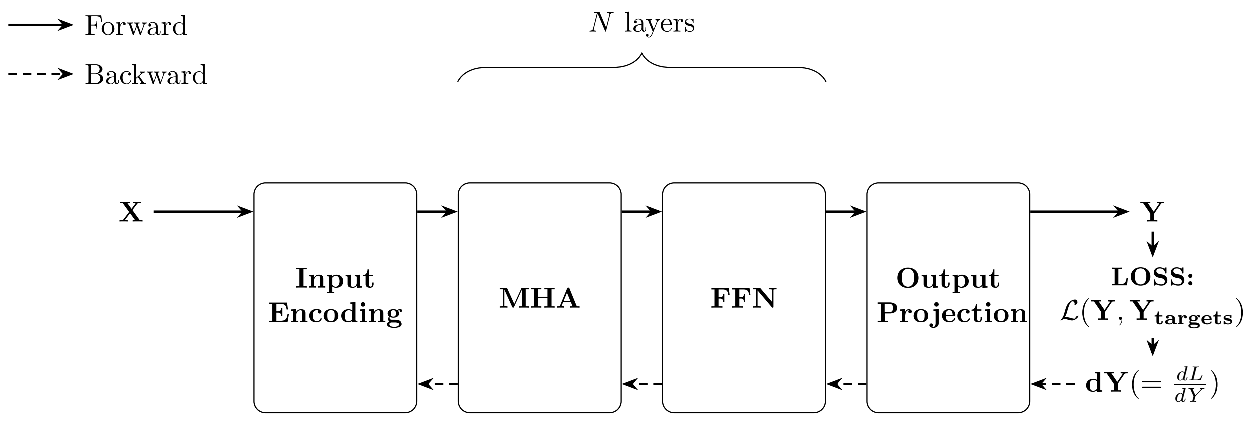
In all cases, the bracket notation on communication edges (for example $[B, S, D]$ or $[B/N_D, S, D/N_T]$) indicates the shape of the tensor sent or received by a *single* node or shard. This makes it possible to reason about the communication volume and compare DP, TP, and DP+TP schemes in a consistent way.

4 Single-Node Transformer: Forward and Backward

This section presents the full Transformer layer running on a single node (no parallelism). We emphasize tensor shapes and data flow for both forward and backward passes.

4.1 Overall Transformer Layer Flow

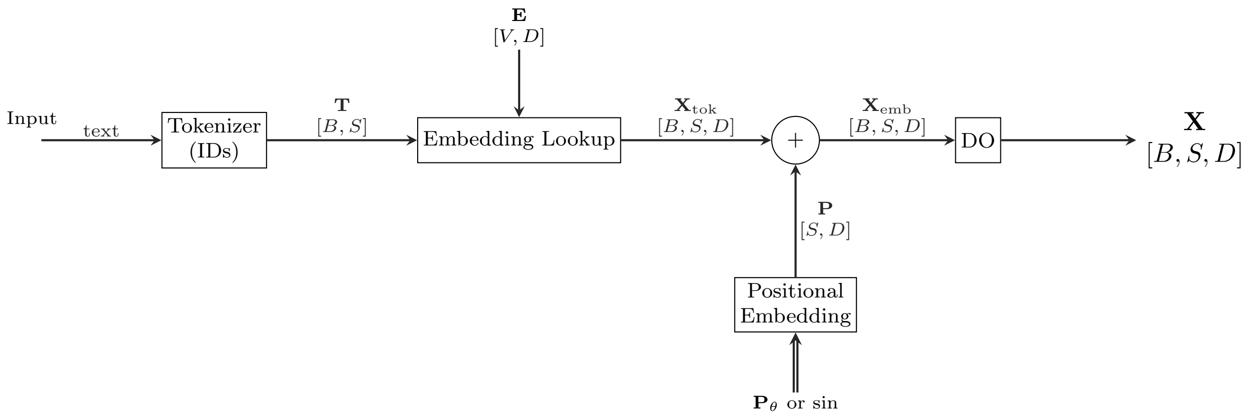
Transformer Overall Flow



4.2 Input Embedding Layer

4.2.1 Forward Pass

Input → Embedding → LN (Input to MHA)

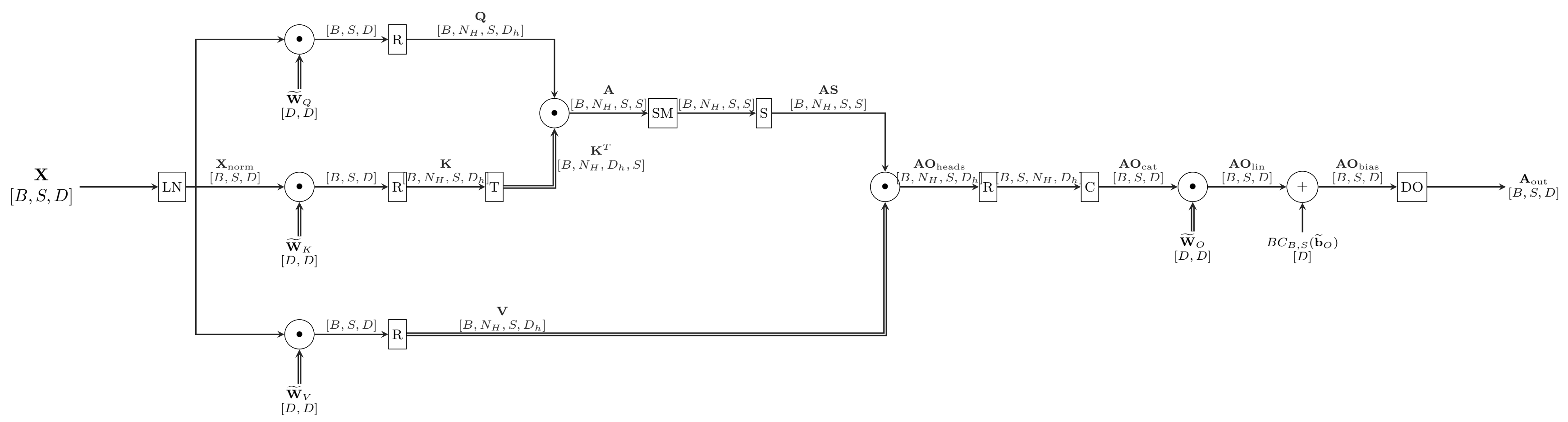


4.3 Multi-Head Attention (MHA)

Multi-Head Attention enables the model to jointly attend to information from different representation subspaces.

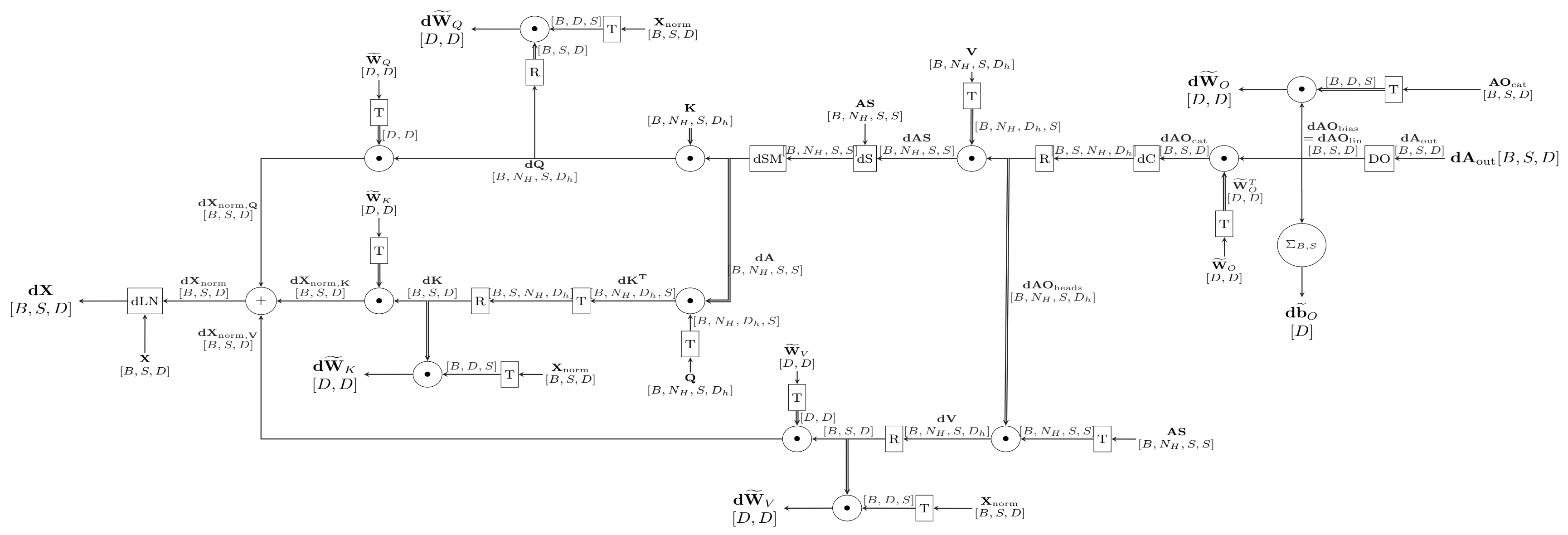
4.3.1 Forward Pass

Multi-Head Attention Forward Pass



4.3.2 Backward Pass

Multi-Head Attention Backward Pass

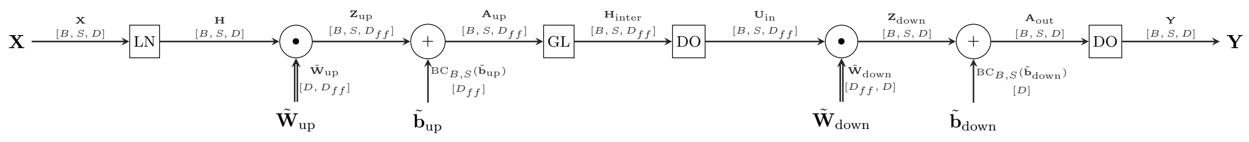


$\mathbf{d\widetilde{W}}_Q$ $[D, D]$
$[B, D, S]$
T
$\mathbf{X}_{\text{norm}}$ $[B, S, D]$
$[B, S, D]$
R
$\widetilde{\mathbf{W}}_Q$ $[D, D]$
$[D, D]$
$\mathbf{dQ}$ $[B, N_H, S, D_h]$
$\mathbf{K}$ $[B, N_H, S, D_h]$
dSM
$[B, N_H, S, S]$
$\mathbf{AS}$ $[B, N_H, S, S]$
dS
$\mathbf{dAS}$ $[B, N_H, S, S]$
$\mathbf{V}$ $[B, N_H, S, D_h]$
$[B, N_H, D_h, S]$
$[B, S, N_H, D_h]$
dC
$\mathbf{dAO}_{\text{cat}}$ $[B, S, D]$
$\widetilde{\mathbf{W}}_O^T$ $[D, D]$
$\widetilde{\mathbf{W}}_O$ $[D, D]$
$\mathbf{d\widetilde{W}}_O$ $[D, D]$
$\mathbf{AO}_{\text{cat}}$ $[B, S, D]$
$\mathbf{dAO}_{\text{bias}} = \mathbf{dAO}_{\text{lin}}$ $[B, S, D]$
$\mathbf{dA}_{\text{out}}$ $[B, S, D]$
DO
$\mathbf{dA}_{\text{out}}[B, S, D]$
$\Sigma_{B,S}$
$\mathbf{d\tilde{b}}_O$ $[D]$
$\mathbf{dX}_{\text{norm},\mathbf{Q}}$ $[B, S, D]$
$\widetilde{\mathbf{W}}_K$ $[D, D]$
$\mathbf{dA}$ $[B, N_H, S, S]$
$\mathbf{dAO}_{\text{heads}}$ $[B, N_H, S, D_h]$
$\mathbf{dX}$ $[B, S, D]$
dLN
$\mathbf{dX}_{\text{norm}}$ $[B, S, D]$
$+$
$\mathbf{dX}_{\text{norm},\mathbf{K}}$ $[B, S, D]$
$\mathbf{dK}$ $[B, S, D]$
$[B, S, N_H, D_h]$
$\mathbf{dK^T}$ $[B, N_H, D_h, S]$
$[B, N_H, D_h, S]$
$\mathbf{Q}$ $[B, N_H, S, D_h]$
$\mathbf{X}$ $[B, S, D]$
$\mathbf{dX}_{\text{norm},\mathbf{V}}$ $[B, S, D]$
$\mathbf{d\widetilde{W}}_K$ $[D, D]$
$[B, D, S]$
$\widetilde{\mathbf{W}}_V$ $[D, D]$
$[D, D]$
$[B, S, D]$
$\mathbf{dV}$ $[B, N_H, S, D_h]$
$[B, N_H, S, S]$
$\mathbf{AS}$ $[B, N_H, S, S]$
$\mathbf{d\widetilde{W}}_V$ $[D, D]$
$[B, D, S]$
$\mathbf{X}_{\text{norm}}$ $[B, S, D]$

4.4 Feed-Forward Network (MLP / FFN)

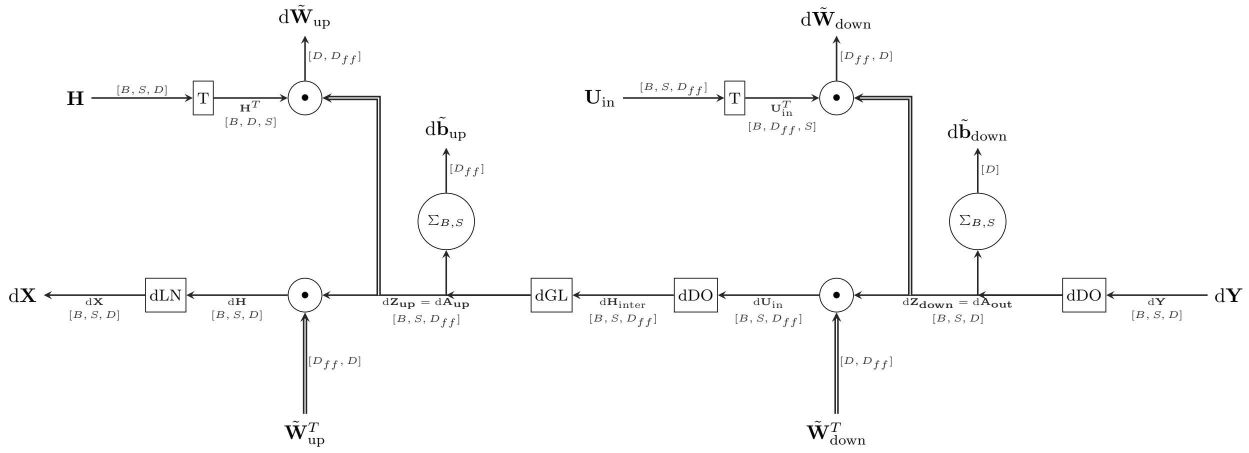
4.4.1 Forward Pass

MLP Forward Pass


$\mathbf{X}$
$\mathbf{X}$ $[B, S, D]$
LN
$\mathbf{H}$ $[B, S, D]$
$\odot$
$\tilde{\mathbf{W}}_{\text{up}}$ $[D, D_{ff}]$
$\tilde{\mathbf{W}}_{\text{up}}$
$\mathbf{Z}_{\text{up}}$ $[B, S, D_{ff}]$
$+$
$\text{BC}_{B,S}(\tilde{\mathbf{b}}_{\text{up}})$ $[D_{ff}]$
$\tilde{\mathbf{b}}_{\text{up}}$
$\mathbf{A}_{\text{up}}$ $[B, S, D_{ff}]$
GL
$\mathbf{H}_{\text{inter}}$ $[B, S, D_{ff}]$
DO
$\mathbf{U}_{\text{in}}$ $[B, S, D_{ff}]$
$\odot$
$\tilde{\mathbf{W}}_{\text{down}}$ $[D_{ff}, D]$
$\tilde{\mathbf{W}}_{\text{down}}$
$\mathbf{Z}_{\text{down}}$ $[B, S, D]$
$+$
$\text{BC}_{B,S}(\tilde{\mathbf{b}}_{\text{down}})$ $[D]$
$\tilde{\mathbf{b}}_{\text{down}}$
$\mathbf{A}_{\text{out}}$ $[B, S, D]$
DO
$\mathbf{Y}$ $[B, S, D]$
$\mathbf{Y}$

4.4.2 Backward Pass

MLP Backward Pass

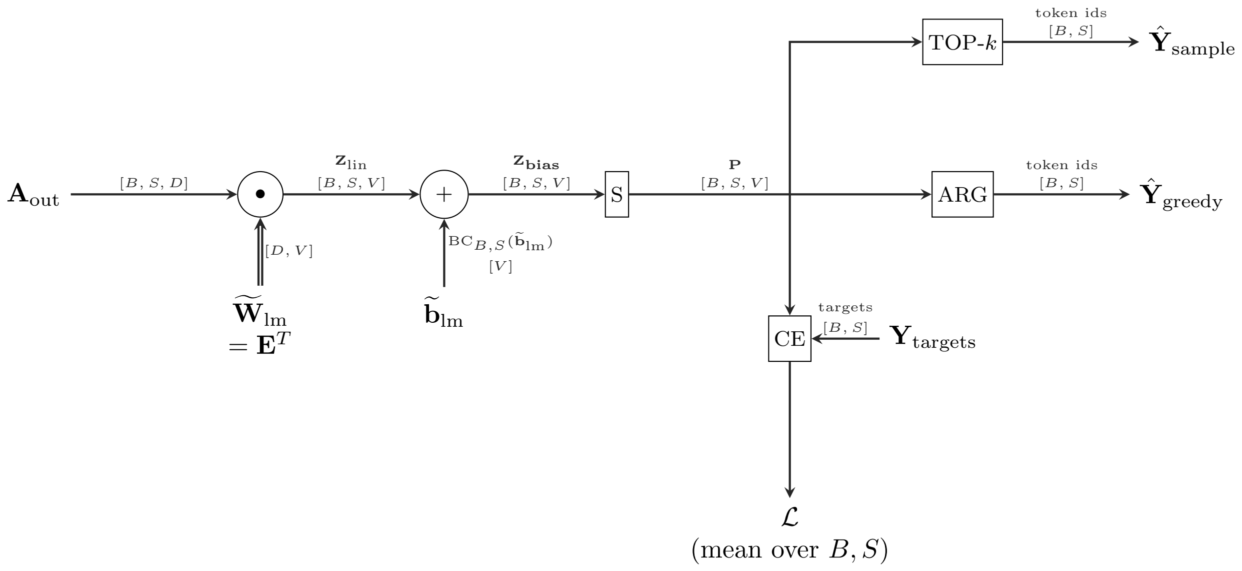


$d\tilde{\mathbf{W}}_{\text{up}}$
$[D, D_{ff}]$
$\mathbf{H}$
$[B, S, D]$
T
$\mathbf{H}^T$
$[B, D, S]$
$d\tilde{\mathbf{b}}_{\text{up}}$
$[D_{ff}]$
$\Sigma_{B,S}$
$d\mathbf{X}$
dLN
$d\mathbf{X}$
$[B, S, D]$
$d\mathbf{H}$
$[B, S, D]$
$d\mathbf{Z}_{\mathbf{up}} = d\mathbf{A}_{\mathbf{up}}$
$[B, S, D_{ff}]$
$[D_{ff}, D]$
$\tilde{\mathbf{W}}_{\text{up}}^T$
dGL
$d\mathbf{H}_{\text{inter}}$
$[B, S, D_{ff}]$
dDO
$d\mathbf{U}_{\text{in}}$
$[B, S, D_{ff}]$
$d\tilde{\mathbf{W}}_{\text{down}}$
$[D_{ff}, D]$
$\mathbf{U}_{\text{in}}$
$[B, S, D_{ff}]$
T
$\mathbf{U}_{\text{in}}^T$
$[B, D_{ff}, S]$
$d\tilde{\mathbf{b}}_{\text{down}}$
$[D]$
$\Sigma_{B,S}$
$d\mathbf{Z}_{\mathbf{down}} = d\mathbf{A}_{\mathbf{out}}$
$[B, S, D]$
$[D, D_{ff}]$
$\tilde{\mathbf{W}}_{\text{down}}^T$
dDO
$d\mathbf{Y}$
$[B, S, D]$
$d\mathbf{Y}$

4.5 Output Projection and Loss

4.5.1 Forward Pass (Logits, Softmax, Loss)

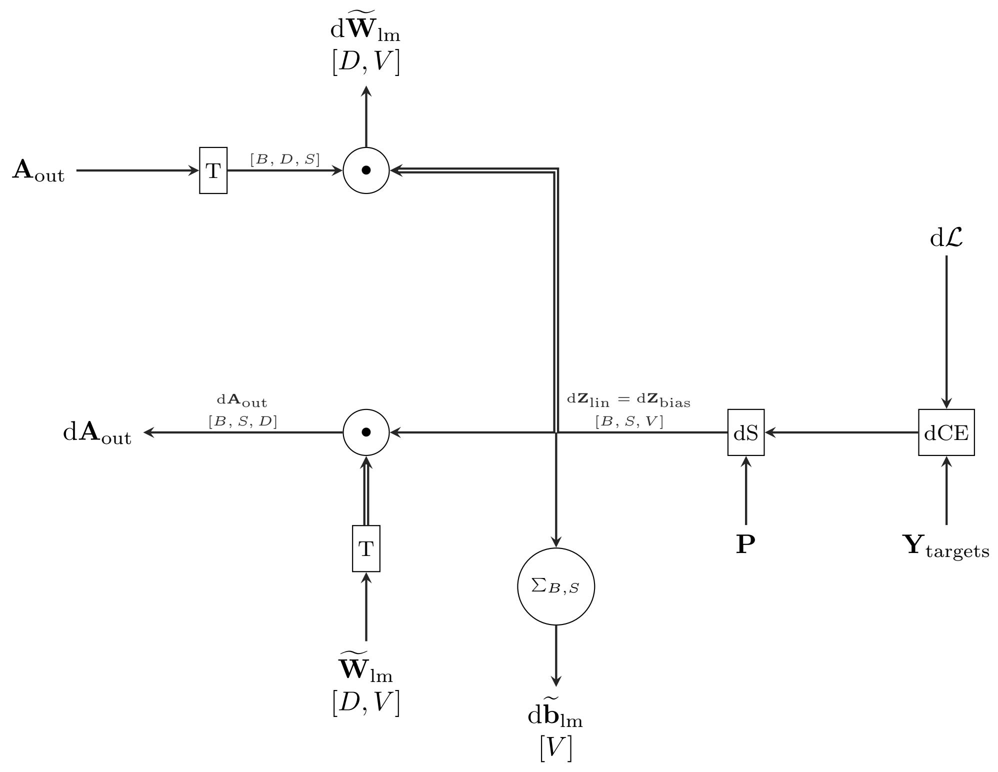
Token Generation & Loss (Forward)



$\mathbf{A}_{\text{out}}$
$[B, S, D]$
$\odot$
$[D, V]$
$\widetilde{\mathbf{W}}_{\text{lm}}$
$= \mathbf{E}^T$
$\mathbf{z}_{\text{lin}}$
$[B, S, V]$
$+$
$\text{BC}_{B,S}(\widetilde{\mathbf{b}}_{\text{lm}})$
$[V]$
$\widetilde{\mathbf{b}}_{\text{lm}}$
$\mathbf{z}_{\textbf{bias}}$
$[B, S, V]$
S
$\mathbf{P}$
$[B, S, V]$
TOP-k
token ids
$[B, S]$
$\hat{\mathbf{Y}}_{\text{sample}}$
ARG
token ids
$[B, S]$
$\hat{\mathbf{Y}}_{\text{greedy}}$
CE
targets
$[B, S]$
$\mathbf{Y}_{\text{targets}}$
$\mathcal{L}$
(mean over B, S)

4.5.2 Backward Pass

Token Generation & Loss — Backward (Corrected)



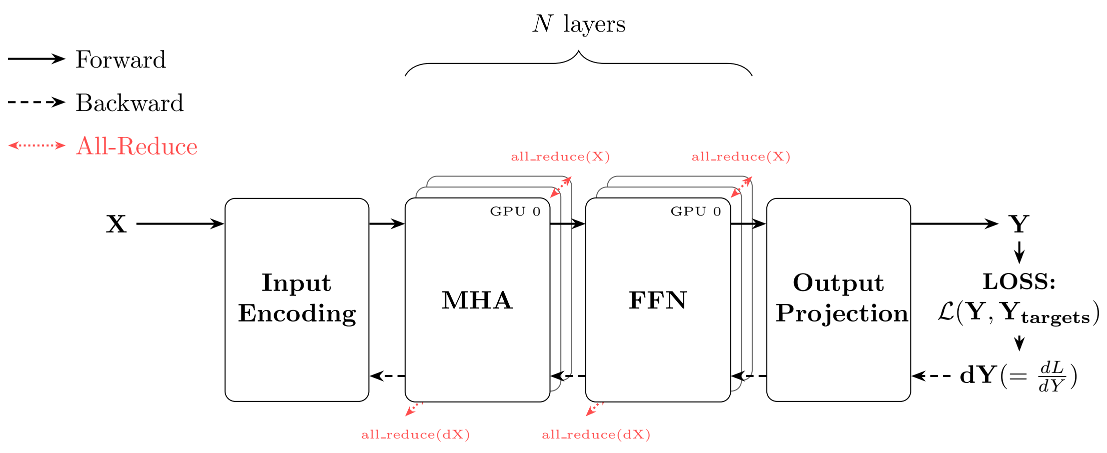
$\mathrm{d}\widetilde{\mathbf{W}}_{\mathrm{lm}}$
$[D, V]$
$\mathbf{A}_{\mathrm{out}}$
T
$[B, D, S]$
$\mathrm{d}\mathcal{L}$
$\mathrm{d}\mathbf{A}_{\mathrm{out}}$
$[B, S, D]$
$\mathrm{d}\mathbf{A}_{\mathrm{out}}$
$\mathrm{d}\mathbf{Z}_{\mathrm{lin}} = \mathrm{d}\mathbf{Z}_{\mathrm{bias}}$
$[B, S, V]$
dS
dCE
T
$\mathbf{P}$
$\mathbf{Y}_{\mathrm{targets}}$
$\Sigma_{B,S}$
$\widetilde{\mathbf{W}}_{\mathrm{lm}}$
$[D, V]$
$\mathrm{d}\widetilde{\mathbf{b}}_{\mathrm{lm}}$
$[V]$

5 Tensor Parallelism (TP)

In tensor parallelism, weight matrices are partitioned across multiple devices along certain dimensions. Each device computes on its own shard, and collective operations (e.g., All-Reduce, All-Gather) synchronize intermediate results.

5.1 TP Overview and End-to-End Flow

Transformer Overall Flow (TP with 3 GPUs)

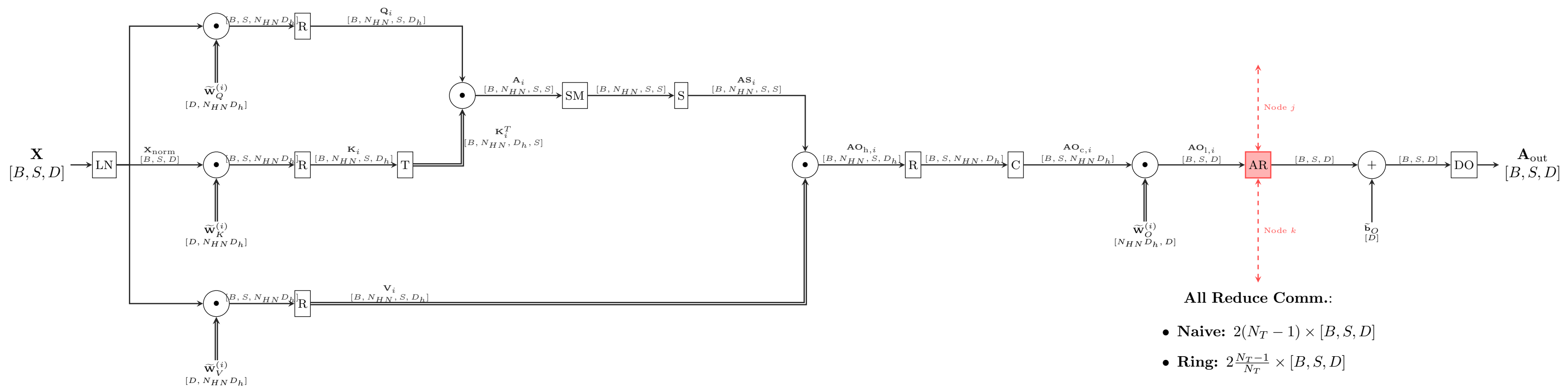


Tensor Parallelism:
- Each GPU processes a shard of the weight matrix
- All-Reduce synchronizes partial results
- Forward: after row-parallel ops
- Backward: after column-parallel ops

5.2 MHA with Tensor Parallelism

5.2.1 Forward Pass

Multi-Head Attention Forward Pass (Node i)



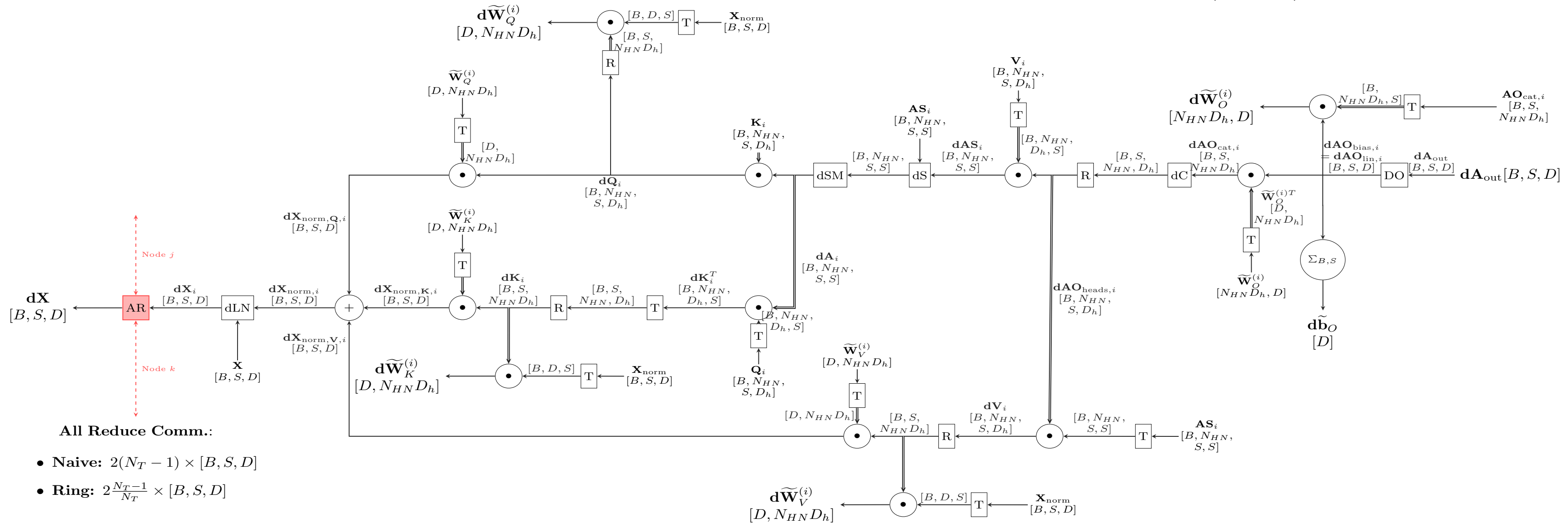
$\mathbf{X}$ $[B, S, D]$
LN
$\mathbf{X}_{\text{norm}}$ $[B, S, D]$
$\widetilde{\mathbf{W}}_Q^{(i)}$ $[D, N_{HN} D_h]$
$\widetilde{\mathbf{W}}_K^{(i)}$ $[D, N_{HN} D_h]$
$\widetilde{\mathbf{W}}_V^{(i)}$ $[D, N_{HN} D_h]$
$[B, S, N_{HN} D_h]$
R
$\mathbf{Q}_i$ $[B, N_{HN}, S, D_h]$
$\mathbf{K}_i$ $[B, N_{HN}, S, D_h]$
T
$\mathbf{K}_i^T$ $[B, N_{HN}, D_h, S]$
$\mathbf{V}_i$ $[B, N_{HN}, S, D_h]$
$\mathbf{A}_i$ $[B, N_{HN}, S, S]$
SM
$[B, N_{HN}, S, S]$
S
$\mathbf{AS}_i$ $[B, N_{HN}, S, S]$
$\mathbf{AO}_{\text{h},i}$ $[B, N_{HN}, S, D_h]$
$[B, S, N_{HN}, D_h]$
C
$\mathbf{AO}_{\text{c},i}$ $[B, S, N_{HN} D_h]$
$\widetilde{\mathbf{W}}_O^{(i)}$ $[N_{HN} D_h, D]$
$\mathbf{AO}_{\text{l},i}$ $[B, S, D]$
AR
Node j
Node k
$[B, S, D]$
$\mathbf{b}_O$ $[D]$
$[B, S, D]$
DO
$\mathbf{A}_{\text{out}}$ $[B, S, D]$

All Reduce Comm.:

- **Naive:** $2(N_T - 1) \times [B, S, D]$
- **Ring:** $2\frac{N_T-1}{N_T} \times [B, S, D]$

5.2.2 Backward Pass

Multi-Head Attention Backward Pass (Node i)



$\mathbf{d}\widetilde{\mathbf{W}}_Q^{(i)}$ $[D, N_{HN}D_h]$
$\mathbf{X}_{\text{norm}}$ $[B,S,D]$
$[B,D,S]$
$[B,S,N_{HN}D_h]$
$\widetilde{\mathbf{W}}_Q^{(i)}$ $[D, N_{HN}D_h]$
$[D, N_{HN}D_h]$
$\mathbf{dQ}_i$ $[B, N_{HN}, S, D_h]$
$\mathbf{K}_i$ $[B, N_{HN}, S, D_h]$
$\mathbf{AS}_i$ $[B, N_{HN}, S, S]$
$\mathbf{V}_i$ $[B, N_{HN}, S, D_h]$
$[B, N_{HN}, D_h, S]$
$[B, N_{HN}, S, S]$
$\mathbf{dAS}_i$ $[B, N_{HN}, S, S]$
$[B, S, N_{HN}, D_h]$
$\mathbf{dAO}_{\text{cat},i}$ $[B, S, N_{HN}D_h]$
$\mathbf{d}\widetilde{\mathbf{W}}_O^{(i)}$ $[N_{HN}D_h, D]$
$[B, N_{HN}D_h, S]$
$\mathbf{AO}_{\text{cat},i}$ $[B, S, N_{HN}D_h]$
$\mathbf{dAO}_{\text{bias},i} = \mathbf{dAO}_{\text{lin},i}$ $[B,S,D]$
$\mathbf{dA}_{\text{out}}$ $[B,S,D]$
$\mathbf{dA}_{\text{out}}[B,S,D]$
dSM
dS
R
dC
DO
T
$\widetilde{\mathbf{W}}_O^{(i)T}$ $[D, N_{HN}D_h]$
$\widetilde{\mathbf{W}}_O^{(i)}$ $[N_{HN}D_h, D]$
$\Sigma_{B,S}$
$\mathbf{d}\widetilde{\mathbf{b}}_O$ $[D]$
$\mathbf{dA}_i$ $[B, N_{HN}, S, S]$
$\mathbf{dAO}_{\text{heads},i}$ $[B, N_{HN}, S, D_h]$
$\mathbf{dX}_{\text{norm},\mathbf{Q},i}$ $[B,S,D]$
$\widetilde{\mathbf{W}}_K^{(i)}$ $[D, N_{HN}D_h]$
$\mathbf{dK}_i$ $[B, S, N_{HN}D_h]$
$[B, S, N_{HN}, D_h]$
$\mathbf{dK}_i^T$ $[B, N_{HN}, D_h, S]$
$[B, N_{HN}, D_h, S]$
$\mathbf{Q}_i$ $[B, N_{HN}, S, D_h]$
$\mathbf{dX}$ $[B,S,D]$
AR
Node j
Node k
$\mathbf{dX}_i$ $[B,S,D]$
dLN
$\mathbf{X}$ $[B,S,D]$
$\mathbf{dX}_{\text{norm},i}$ $[B,S,D]$
$\mathbf{dX}_{\text{norm},\mathbf{K},i}$ $[B,S,D]$
$\mathbf{dX}_{\text{norm},\mathbf{V},i}$ $[B,S,D]$
$\mathbf{d}\widetilde{\mathbf{W}}_K^{(i)}$ $[D, N_{HN}D_h]$
$[B,D,S]$
$\widetilde{\mathbf{W}}_V^{(i)}$ $[D, N_{HN}D_h]$
$[D, N_{HN}D_h]$
$[B, S, N_{HN}D_h]$
$\mathbf{dV}_i$ $[B, N_{HN}, S, D_h]$
$[B, N_{HN}, S, S]$
$\mathbf{AS}_i$ $[B, N_{HN}, S, S]$
$\mathbf{d}\widetilde{\mathbf{W}}_V^{(i)}$ $[D, N_{HN}D_h]$
$[B,D,S]$

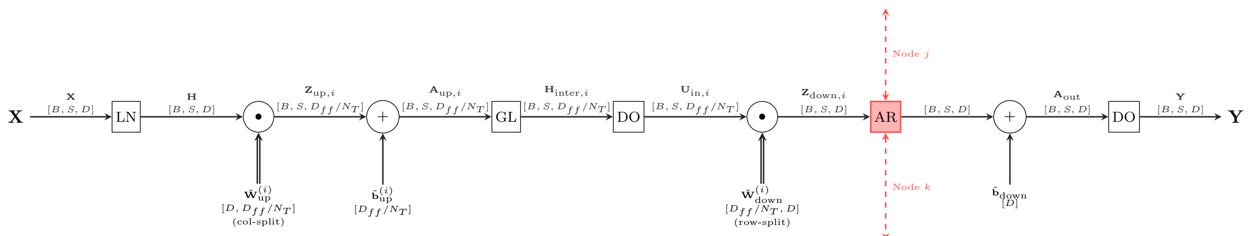
All Reduce Comm.:

- **Naive:** $2(N_T - 1) \times [B,S,D]$
- **Ring:** $2\frac{N_T-1}{N_T} \times [B,S,D]$

5.3 MLP with Tensor Parallelism

5.3.1 Forward Pass

MLP Forward Pass (Node i)

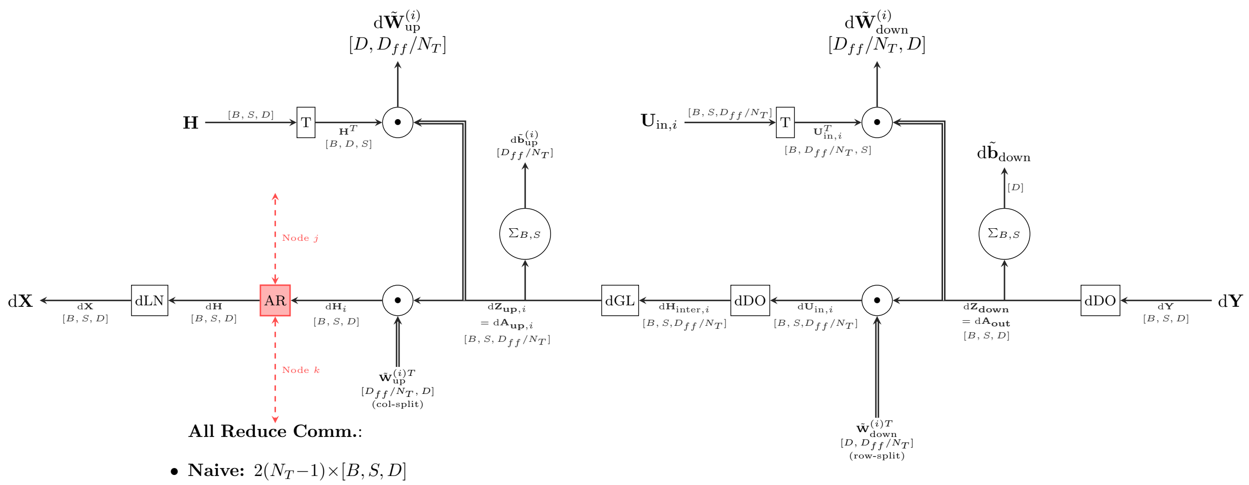

$\mathbf{X}$
$\mathbf{x}$ $[B, S, D]$
LN
$\mathbf{H}$ $[B, S, D]$
$\tilde{\mathbf{W}}_{\text{up}}^{(i)}$ $[D, D_{ff}/N_T]$ (col-split)
$\mathbf{Z}_{\text{up},i}$ $[B, S, D_{ff}/N_T]$
$\tilde{\mathbf{b}}_{\text{up}}^{(i)}$ $[D_{ff}/N_T]$
$\mathbf{A}_{\text{up},i}$ $[B, S, D_{ff}/N_T]$
GL
$\mathbf{H}_{\text{inter},i}$ $[B, S, D_{ff}/N_T]$
DO
$\mathbf{U}_{\text{in},i}$ $[B, S, D_{ff}/N_T]$
$\tilde{\mathbf{W}}_{\text{down}}^{(i)}$ $[D_{ff}/N_T, D]$ (row-split)
$\mathbf{Z}_{\text{down},i}$ $[B, S, D]$
Node j
AR
Node k
$[B, S, D]$
$\tilde{\mathbf{b}}_{\text{down}}$ $[D]$
$\mathbf{A}_{\text{out}}$ $[B, S, D]$
DO
$\mathbf{Y}$ $[B, S, D]$
$\mathbf{Y}$

All Reduce Comm.:

- **Naive:** $2(N_T-1)\times[B,S,D]$
- **Ring:** $2\frac{N_T-1}{N_T}\times[B,S,D]$

5.3.2 Backward Pass

MLP Backward Pass (Node i)



$\mathrm{d}\tilde{\mathbf{W}}_{\mathrm{up}}^{(i)}$
$[D, D_{ff}/N_T]$
$\mathbf{H}$
$[B, S, D]$
T
$\mathbf{H}^T$
$[B, D, S]$
$\mathrm{d}\tilde{\mathbf{b}}_{\mathrm{up}}^{(i)}$
$[D_{ff}/N_T]$
$\Sigma_{B,S}$
Node j
Node k
$\mathrm{d}\mathbf{X}$
$\mathrm{d}\mathbf{X}$
$[B, S, D]$
dLN
$\mathrm{d}\mathbf{H}$
$[B, S, D]$
AR
$\mathrm{d}\mathbf{H}_i$
$[B, S, D]$
$\mathrm{d}\mathbf{Z}_{\mathbf{up},i}$
$= \mathrm{d}\mathbf{A}_{\mathbf{up},i}$
$[B, S, D_{ff}/N_T]$
$\tilde{\mathbf{W}}_{\mathrm{up}}^{(i)T}$
$[D_{ff}/N_T, D]$
(col-split)
dGL
$\mathrm{d}\mathbf{H}_{\mathrm{inter},i}$
$[B, S, D_{ff}/N_T]$
dDO
$\mathrm{d}\mathbf{U}_{\mathrm{in},i}$
$[B, S, D_{ff}/N_T]$
$\mathrm{d}\tilde{\mathbf{W}}_{\mathrm{down}}^{(i)}$
$[D_{ff}/N_T, D]$
$\mathbf{U}_{\mathrm{in},i}$
$[B, S, D_{ff}/N_T]$
T
$\mathbf{U}_{\mathrm{in},i}^T$
$[B, D_{ff}/N_T, S]$
$\mathrm{d}\tilde{\mathbf{b}}_{\mathrm{down}}$
$[D]$
$\Sigma_{B,S}$
$\mathrm{d}\mathbf{Z}_{\mathbf{down}}$
$= \mathrm{d}\mathbf{A}_{\mathbf{out}}$
$[B, S, D]$
dDO
$\mathrm{d}\mathbf{Y}$
$[B, S, D]$
$\mathrm{d}\mathbf{Y}$
$\tilde{\mathbf{W}}_{\mathrm{down}}^{(i)T}$
$[D, D_{ff}/N_T]$
(row-split)

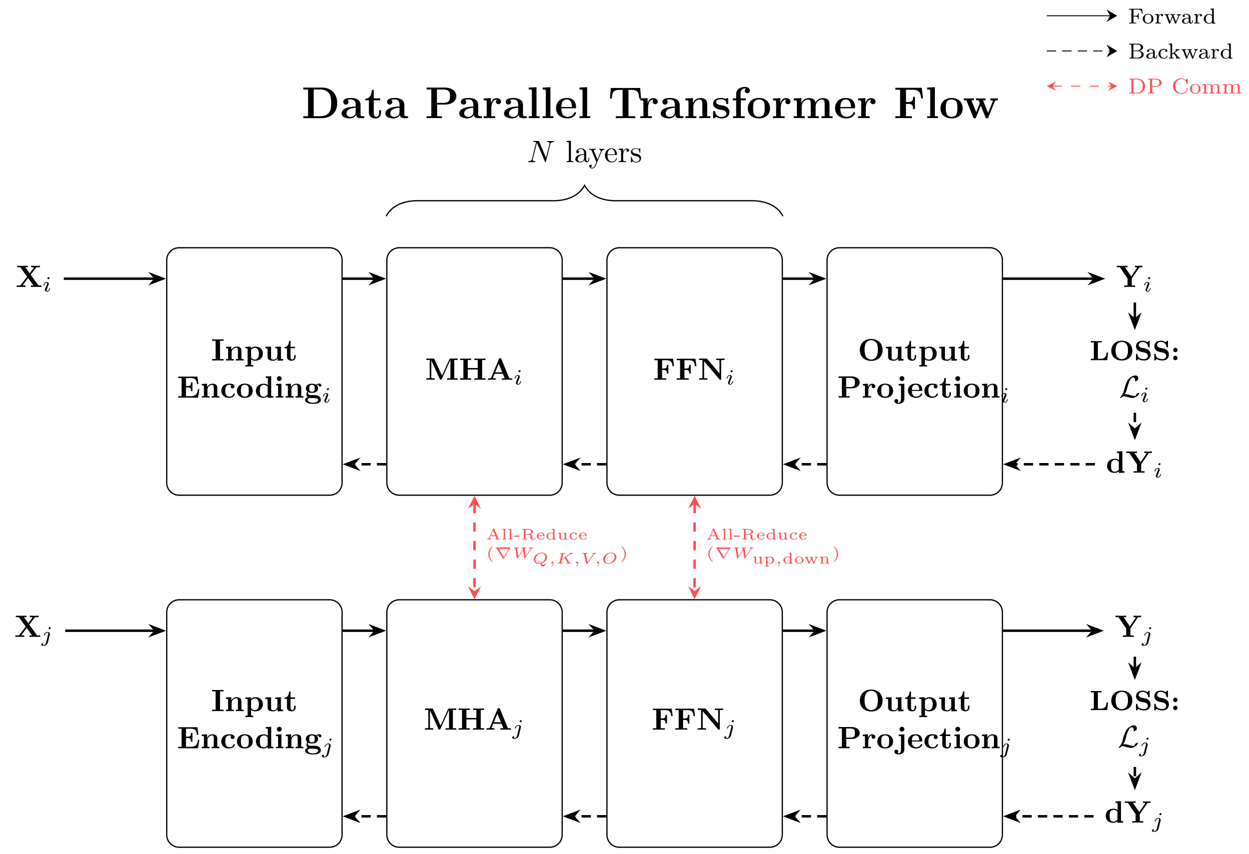
All Reduce Comm.:

- **Naive:** $2(N_T - 1) \times [B, S, D]$
- **Ring:** $2\frac{N_T - 1}{N_T} \times [B, S, D]$

6 Data Parallelism (DP)

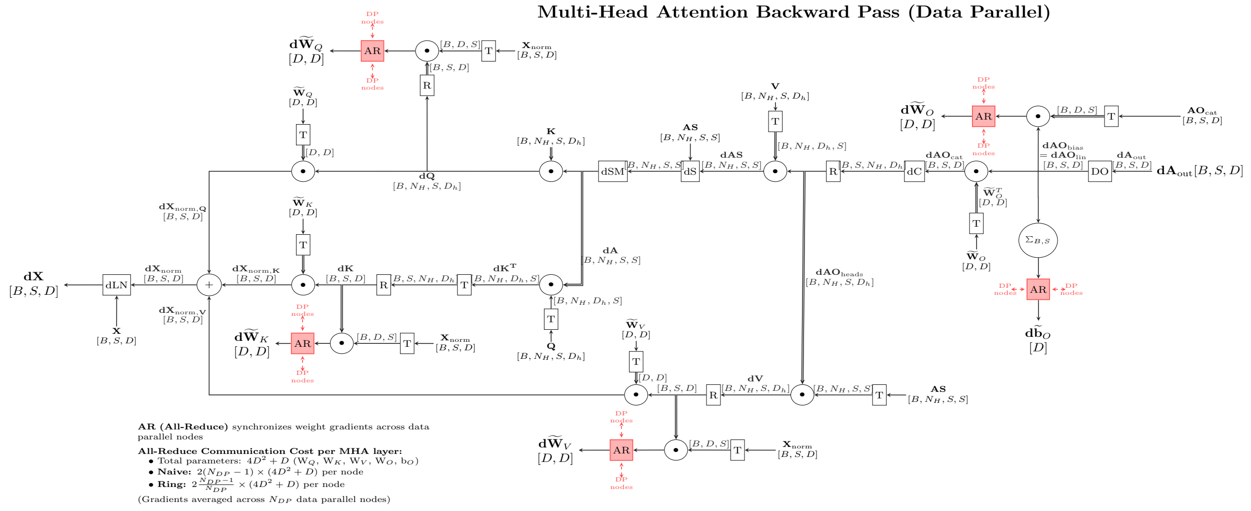
In data parallelism, each replica holds a full copy of the model, but processes a different subset of the batch. Gradients are synchronized across replicas via All-Reduce.

6.1 DP Overview and Transformer Flow



6.2 MHA Backward under DP

Multi-Head Attention Backward Pass (Data Parallel)



$\mathbf{d}\widetilde{\mathbf{W}}_Q$ $[D, D]$
AR
DP nodes
$[B, D, S]$
T
$\mathbf{X}_{\text{norm}}$ $[B, S, D]$
R
$[B, S, D]$
$\widetilde{\mathbf{W}}_Q$ $[D, D]$
$[D, D]$
$\mathbf{dQ}$ $[B, N_H, S, D_h]$
$\mathbf{K}$ $[B, N_H, S, D_h]$
dSM
$[B, N_H, S, S]$
$\mathbf{AS}$ $[B, N_H, S, S]$
dS
$\mathbf{dAS}$ $[B, N_H, S, S]$
$\mathbf{V}$ $[B, N_H, S, D_h]$
$[B, N_H, D_h, S]$
$[B, S, N_H, D_h]$
dC
$\mathbf{dAO}_{\text{cat}}$ $[B, S, D]$
$\mathbf{d}\widetilde{\mathbf{W}}_O$ $[D, D]$
$\mathbf{AO}_{\text{cat}}$ $[B, S, D]$
$\mathbf{dAO}_{\text{bias}}$ $= \mathbf{dAO}_{\text{lin}}$ $[B, S, D]$
$\mathbf{dA}_{\text{out}}$ $[B, S, D]$
DO
$\mathbf{dA}_{\text{out}} [B, S, D]$
$\widetilde{\mathbf{W}}_O^T$ $[D, D]$
$\widetilde{\mathbf{W}}_O$ $[D, D]$
$\Sigma_{B,S}$
$\mathbf{d}\widetilde{\mathbf{b}}_O$ $[D]$
$\mathbf{dX}_{\text{norm},\mathbf{Q}}$ $[B, S, D]$
$\widetilde{\mathbf{W}}_K$ $[D, D]$
$\mathbf{dA}$ $[B, N_H, S, S]$
$\mathbf{dAO}_{\text{heads}}$ $[B, N_H, S, D_h]$
$\mathbf{dX}$ $[B, S, D]$
dLN
$\mathbf{dX}_{\text{norm}}$ $[B, S, D]$
$\mathbf{dX}_{\text{norm},\mathbf{K}}$ $[B, S, D]$
$\mathbf{dK}$ $[B, S, D]$
$[B, S, N_H, D_h]$
$\mathbf{dK}^{\mathbf{T}}$ $[B, N_H, D_h, S]$
$[B, N_H, D_h, S]$
$\mathbf{Q}$ $[B, N_H, S, D_h]$
$\mathbf{X}$ $[B, S, D]$
$\mathbf{dX}_{\text{norm},\mathbf{V}}$ $[B, S, D]$
$\mathbf{d}\widetilde{\mathbf{W}}_K$ $[D, D]$
$\widetilde{\mathbf{W}}_V$ $[D, D]$
$\mathbf{dV}$ $[B, N_H, S, D_h]$
$[B, N_H, S, S]$
$\mathbf{AS}$ $[B, N_H, S, S]$
$\mathbf{d}\widetilde{\mathbf{W}}_V$ $[D, D]$

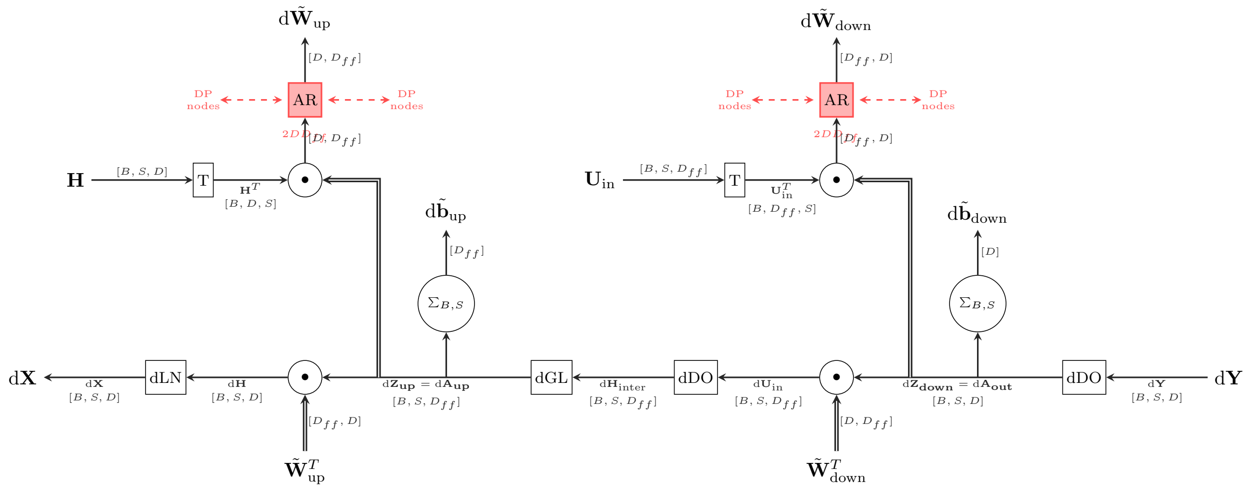
AR (All-Reduce) synchronizes weight gradients across data parallel nodes

All-Reduce Communication Cost per MHA layer:
- Total parameters: $4D^2 + D$ (W$_Q$, W$_K$, W$_V$, W$_O$, b$_O$)
- **Naive:** $2(N_{DP} - 1) \times (4D^2 + D)$ per node
- **Ring:** $2\frac{N_{DP}-1}{N_{DP}} \times (4D^2 + D)$ per node

(Gradients averaged across N_{DP} data parallel nodes)

6.3 MLP Backward under DP

MLP Backward Pass (Data Parallel)



AR (All-Reduce) synchronizes weight gradients across data parallel nodes

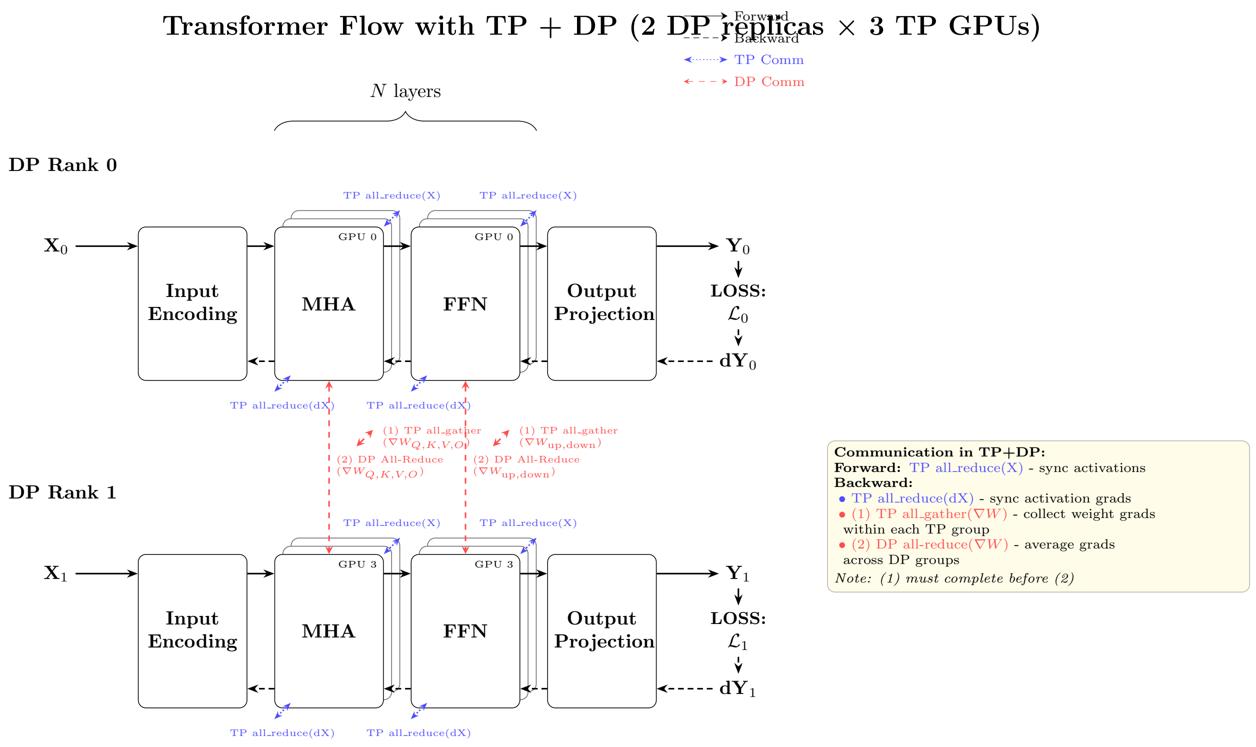
MLP All-Reduce Cost: $\sim 2DD_{ff}$ parameters (W_{up}, W_{down})
- **Naive:** $2(N_{DP} - 1) \times 2DD_{ff}$ per node
- **Ring:** $2\frac{N_{DP}-1}{N_{DP}} \times 2DD_{ff}$ per node

(Gradients averaged across N_{DP} data parallel nodes)

7 Hybrid Data + Tensor Parallelism (DP + TP)

We combine tensor parallelism within each node (or group of devices) with data parallelism across groups. This section explains how the two forms of parallelism interact in both forward and backward passes.

7.1 DP+TP Overview and Communication Patterns



Communication in TP+DP:
Forward: TP all_reduce(X) - sync activations
Backward:
- TP all_reduce(dX) - sync activation grads
- (1) TP all_gather(∇W) - collect weight grads within each TP group
- (2) DP all-reduce(∇W) - average grads across DP groups

Note: (1) must complete before (2)

8 Summary and Practical Takeaways

We summarize the main ideas:

- How a Transformer layer operates as a composition of embedding, MHA, MLP, and output projection blocks.
- How tensor shapes evolve through forward and backward passes.
- How single-node execution extends to tensor parallelism, data parallelism, and their combination.

We also highlight how these diagrams can be used as a reference when designing or debugging large-scale Transformer training and inference systems.