# DeepSeek Architecture

Changshi Li

Wuhan University

2025.3.6


# Contents



# Math format

Let $X \in \{0,1\}^{T\times d}$ for input, where $d$ is the dimension of tokens and $T$ is the length of inputs.Then the **embedding layer** is defined as:

$$\mathrm{Enc}(X) = XE, \tag{1}$$

where $E \in \mathbb{R}^{d\times D}$.
**Remark**: $\mathrm{Enc}(X) : \{0,1\}^{T\times d} \to \mathbb{R}^{T\times D}$.
Let $X \in \mathbb{R}^{T\times D}$ be the hidden input and $X_{i\cdot}^{T} \in \mathbb{R}^{D}$, where $D$ is the dimension of middle layer. Then **RMSNorm** layer is defined as:

$$\mathrm{RMSNorm}(X_{i\cdot}) := \frac{X_{i\cdot}}{\sqrt{\frac{1}{D}\|X_{i\cdot}\|_2^2}}, \forall i \in \{0,1,...,T\} \tag{2}$$

**Remark**: $\mathrm{RMSNorm}(X) : \mathbb{R}^{T\times D} \to \mathbb{R}^{T\times D}$.

# Math format

Let $X \in \mathbb{R}^{T \times D}$ be the hidden input, then the Rotary Positional Embedding (**RoPE**) is defined as:

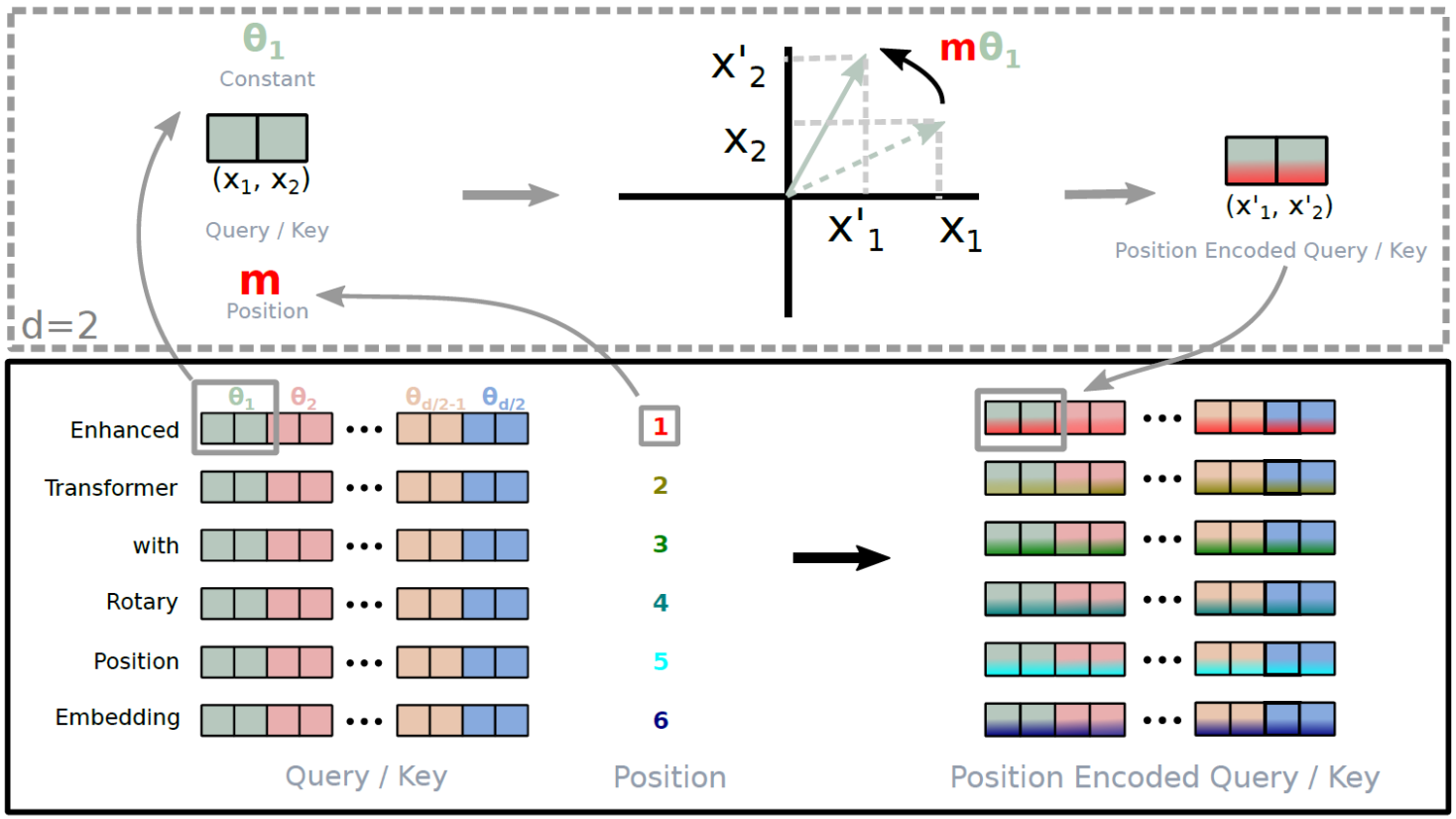
$$\mathrm{RoPE}(X; W_p, W_R) = X W_p W_R^T, \tag{3}$$

where

$$W_R = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{D^R}/2 & -\sin m\theta_{D^R}/2 \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{D^R}/2 & \cos m\theta_{D^R}/2 \end{pmatrix}$$

, $\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, \ldots, D^R/2]$ and $W_p \in \mathbb{R}^{D \times D^R}$.
**Remark**:$\mathrm{RoPE}(X) : T \times D \to T \times D^R$.

# RoPE


θ1
Constant
(x1, x2)
Query / Key
m
Position
d=2
x'2
x2
x'1
x1
mθ1
(x'1, x'2)
Position Encoded Query / Key
θ1
θ2
θd/2-1
θd/2
Enhanced
Transformer
with
Rotary
Position
Embedding
1
2
3
4
5
6
Query / Key
Position
Position Encoded Query / Key

# Math format

Let $X \in \mathbb{R}^{T \times D}$ be the hidden input, then the **LoRA** layer is defined as:

$$\mathrm{LoRA}(X; W_{down}, W_{up}) := XW_{down}W_{up}, \tag{4}$$

where $W_{down} \in \mathbb{R}^{D \times d_{down}}$, $W_{up} \in \mathbb{R}^{d_{down} \times d_h n_h}$ , $d_{down} = d_c$ for keys and values, $d_{down} = d'_c$ for queries, $d_h$ is the hidden dimension before attention layer and $h_n$ is the number of heads in multi-head attention layer.
**Remark**:$\mathrm{LoRA}(X; W_{down}, W_{up}) : T \times D \to T \times d_h h_n$.
Let $L \in T \times d_h h_n$, Define an extract operator as follows:

$$L_{..h} = L_{i,j \times h_n + h}, \forall i \in \{1, \cdots, T\}, j \in \{1, \cdots, d_h\} \tag{5}$$

where $h \in \{1, \cdots, h_n\}$.
**Remark**: $L_{..h} : T \times d_h h_n \to T \times d_h$

# Math format

Let $X \in \mathbb{R}^{T \times D}$ be the hidden input, $W_O \in \mathbb{R}^{d_h n_h \times D}$, $\mathrm{Softmax}(X)$ is a row-wise operator, then the multi-head attention (**MLA**) is defined as follows:

- LoRA layer:

$$\begin{cases} L_Q = LoRA(X; W^{DQ}, W^{UQ}), \\ L_K = LoRA(X; W^{DKV}, W^{UK}), \\ V = LoRA(X; W^{DKV}, W^{UV}). \end{cases} \tag{6}$$

- RoPE layer:

$$R_Q = \mathrm{RoPE}(X; W_{DQ}, W_{QR}), R_K = \mathrm{RoPE}(X; W_{DKV}, W_{KR}) \tag{7}$$

- Cat the output of LoRA and ROPE for one head:

$$Q_h = [L_{Q \cdot \cdot h}, R_{Q \cdot \cdot h}], K_h = [L_{k \cdot \cdot h}, R_{K \cdot \cdot h}], \tag{8}$$

where $Q_h, R_K \in \mathbb{R}^{T \times (d_h + D^R)}$.

# Math format

- Single head attention:

$$S_h = \mathrm{Softmax}(\frac{Q_h K_h^T + M}{\sqrt{D^R + d_h}}) V_{..h}, \tag{9}$$

where $M \in \{-\infty, 0\}^{T \times T}$, $M_{ij} = -\infty$ if $i < j$ else $M_{ij} = 0$.
**Remark**: $\mathrm{S_h} \in \mathbb{R}^{T \times T}$.

- Concatenate & Residual

$$\mathrm{MLA}(X) := [S_1, \ldots, S_h, \ldots, S_{h_n}] W_O, \tag{10}$$

**Remark**: $\mathrm{MLA} : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times D}$.

## Math format

The **MLP** layer is defined as:

$$\mathrm{MLP}(X) := f_3 \circ (\sigma(f_2(X)) \odot f_1(X)), \tag{11}$$

where $f_i(X) := XW_i + b_i$, $b_i \in \mathbb{R}^{D_i}$, $W_1 \in \mathbb{R}^{D \times D_1}$, $W_i \in \mathbb{R}^{D_{i-1} \times D_i}$ and $W_L \in \mathbb{R}^{D_{L-1} \times D}$.

**Remark**: $\mathrm{MLP} : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times D}$.

For each element $x$, the **active function** $\sigma$ is defined as:

$$\mathrm{SiLU}(x) := \frac{x}{1 + e^{-x}}. \tag{12}$$

# Math format

Let $X \in \mathbb{R}^{T \times D}$ be the hidden input, the **Gate** is defined as follows:

- Obtain the weights of $N_r$ experts for every tokens:

$$G = \mathrm{Sigmoid}(XW_{experts}), \tag{13}$$

  where $W_{experts} \in \mathbb{R}^{D \times N_r}$ and Sigmoid is a row-wise operator.

- Select top $K_r$ experts for every tokens:

$$g'_{i,t} = \begin{cases} G_{i,t}, & G_{i,t} \in \mathrm{Topk}(\{G_{i,k} \mid 1 \le k \le N_r\}, K_r) \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

- Normalization:

$$\mathrm{Gate}(X)_{it} := \frac{g'_{i,t}}{\sum_{t=1}^{N_r} g'_{j,t}} \tag{15}$$

**Remark**: $\mathrm{Gate}(X) : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times N_r}$. In the code only need $T \times K_r$ because the remains are zero.

# Math format

Let $X \in \mathbb{R}^{T \times D}$ be the hidden input. There are $N_r$ routed experts ($\mathbf{MLP}^r$) and $N_s$ shared experts ($\mathbf{MLP}^s$), then the **MOE** layer is defined as:

$$\mathrm{MoE}(X) = \sum_{i=1}^{N_s} \mathrm{MLP}_i^s(X) + \sum_{j=1}^{N_r} \mathrm{Gate}(X)_{\cdot j}\, \mathrm{MLP}_j^r(X), \tag{16}$$

**Remark**:$\mathrm{MoE}(X) : \mathbb{R}^{T \times D} \to \mathbb{R}^{T \times D}$.

# Math format

The **Block** in DeepSeek is defined as follows:

- The attention layer

$$\mathrm{ResMLA}(X) := X + \mathrm{MLA} \circ \mathrm{RMSNorm}(X) \tag{17}$$

- If the block number less than the number of dense layers (n_dense_layers).

$$\mathrm{ResMLP}(X) := X + \mathrm{MLP} \circ \mathrm{RMSNorm}(X), \tag{18}$$

- For the other blocks

$$\mathrm{ResMoE}(X) := X + \mathrm{MoE} \circ \mathrm{RMSNorm}(X), \tag{19}$$

- The i-th block is defined as:

$$\mathrm{Block_i}(X) = \begin{cases} \mathrm{ResMLP} \circ \mathrm{ResMLA}(X), i < \mathrm{n\_dense\_layers}, \\ \mathrm{ResMoE} \circ \mathrm{ResMLA}(X), \textit{others} \end{cases} \tag{20}$$

The **embedding** is defined as:

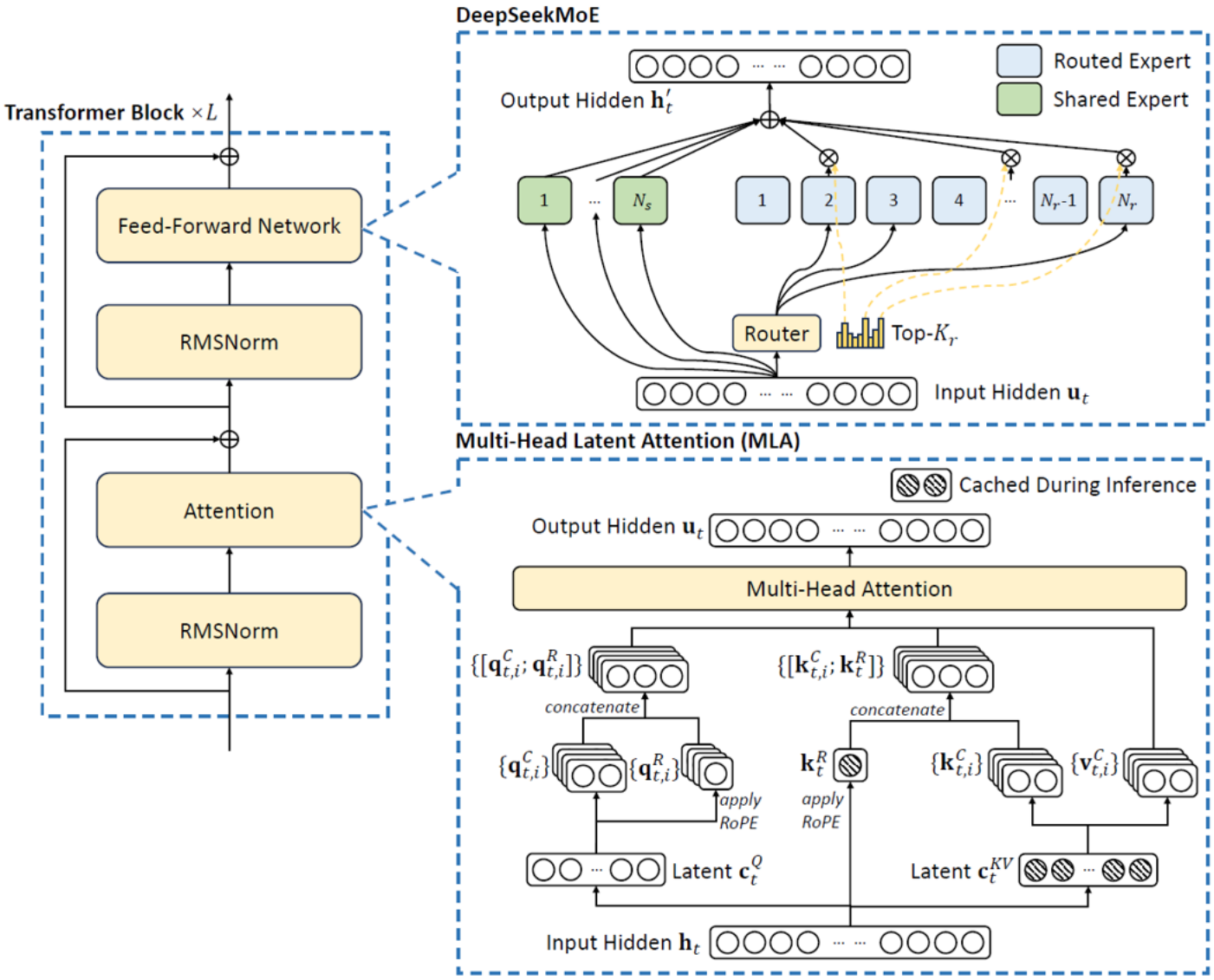
$$X^e := \mathrm{RMSNorm} \circ \mathrm{Block}_M \cdots \mathrm{Block}_1 \circ \mathrm{Enc}(X), \tag{21}$$

where $X^e \in \mathbb{R}^{T \times D}$. The output of **DeepSeek** is defined as:

$$\mathrm{DeepSeek} := \arg\max_{\mathrm{index}} X^l = X^e W_{\mathrm{head}}, \tag{22}$$

where $W_{\mathrm{head}} \in \mathbb{R}^{D \times d}$.

# MLA and MOE


Transformer Block ×L
Feed-Forward Network
RMSNorm
Attention
RMSNorm
DeepSeekMoE
Routed Expert
Shared Expert
Output Hidden $\mathbf{h}'_t$
1
$N_s$
1
2
3
4
$N_r$-1
$N_r$
Router
Top-$K_r$
Input Hidden $\mathbf{u}_t$
Multi-Head Latent Attention (MLA)
Cached During Inference
Output Hidden $\mathbf{u}_t$
Multi-Head Attention
$\{[\mathbf{q}^C_{t,i}; \mathbf{q}^R_{t,i}]\}$
$\{[\mathbf{k}^C_{t,i}; \mathbf{k}^R_t]\}$
concatenate
concatenate
$\{\mathbf{q}^C_{t,i}\}$
$\{\mathbf{q}^R_{t,i}\}$
apply RoPE
$\mathbf{k}^R_t$
apply RoPE
$\{\mathbf{k}^C_{t,i}\}$
$\{\mathbf{v}^C_{t,i}\}$
Latent $\mathbf{c}^Q_t$
Latent $\mathbf{c}^{KV}_t$
Input Hidden $\mathbf{h}_t$

# DeepSeek parameters

| **Parameter** | **168B** | **236B** | **671B** |
|---|---|---|---|
| vocab_size (d) | 102400 | 102400 | 129280 |
| dim (D) | 2048 | 5120 | 7168 |
| inter_dim (MLP) | 10944 | 12288 | 18432 |
| moe_inter_dim (MoE) | 1408 | 1536 | 2048 |
| n_layers (M) | 27 | 60 | 61 |
| n_dense_layers | 1 | 1 | 3 |
| n_heads (n_h) | 16 | 128 | 128 |
| n_routed_experts ($N_r$) | 64 | 160 | 256 |
| n_shared_experts ($N_s$) | 2 | 2 | 1 |
| n_activated_experts ($K_r$) | 6 | 6 | 8 |
| q_lora_rank (LoRA) | 0 | 1536 | 1536 |
| kv_lora_rank (LoRA) | 512 | 512 | 512 |
| v_head_dim ($h_n$) | 128 | 128 | 128 |

# Contents



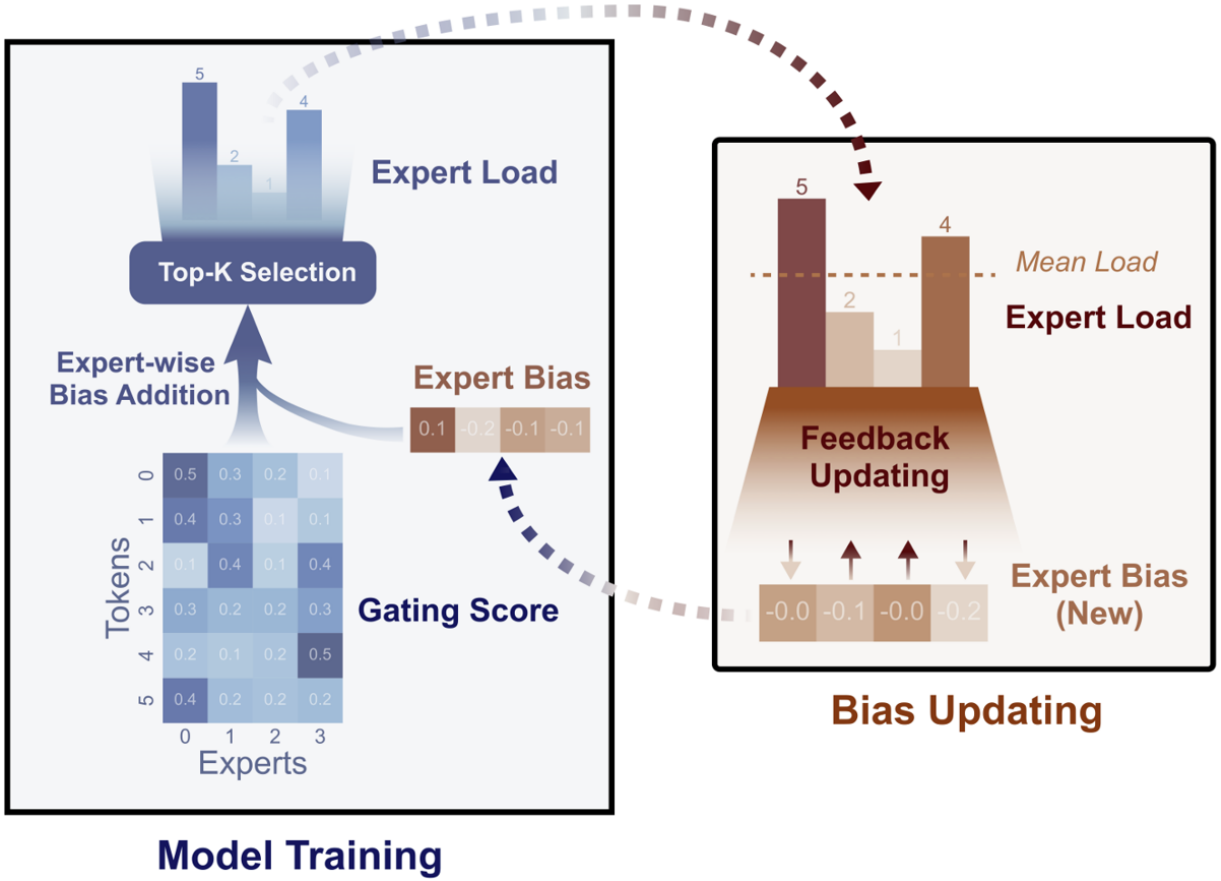
# Main Loss Function

The main loss function is:

$$L_{main} = -\sum_{i=1}^{T} \log \mathrm{Softmax}(X^l)_{ij_{\mathrm{true}}} \tag{23}$$

, where $j_{\mathrm{true}}$ is the next token in the directory.

# Auxiliary-Loss-Free Load Balancing




An unbalanced expert load will lead to routing collapse and diminish computational efficiency in scenarios with expert parallelism.

# Auxiliary-Loss-Free Load Balancing

**Algorithm 1:** Adjusting the per-expert bias $b_i$ during training

**Input:** MoE model $\theta$, training batch iterator $B$, bias update rate $u$.

1. Initialize $b_i = 0$ for each expert;

**for** *a batch* $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$ *in* $B$ **do**

2. Train MoE model $\theta$ on the batch data $\{(\mathbf{x}_k, \mathbf{y}_k)\}_k$, with gating scores calculated according to Eq. (3);
3. Count the number of assigned tokens $c_i$ for each expert, and the average number $\overline{c_i}$;
4. Calculate the load violation error $e_i = \overline{c_i} - c_i$;
4. Update $\mathbf{b}_i$ by $b_i = b_i + u * \text{sign}(e_i)$;

**end**

**Output:** trained model $\theta$, corresponding bias $\mathbf{b}_i$

$$g'_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} + b_i \in \text{Topk}(\{s_{j,t} + b_j | 1 \leqslant j \leqslant N_r\}, K_r), \\ 0, & \text{otherwise.} \end{cases}$$

# Complementary Sequence-Wise Auxiliary Loss

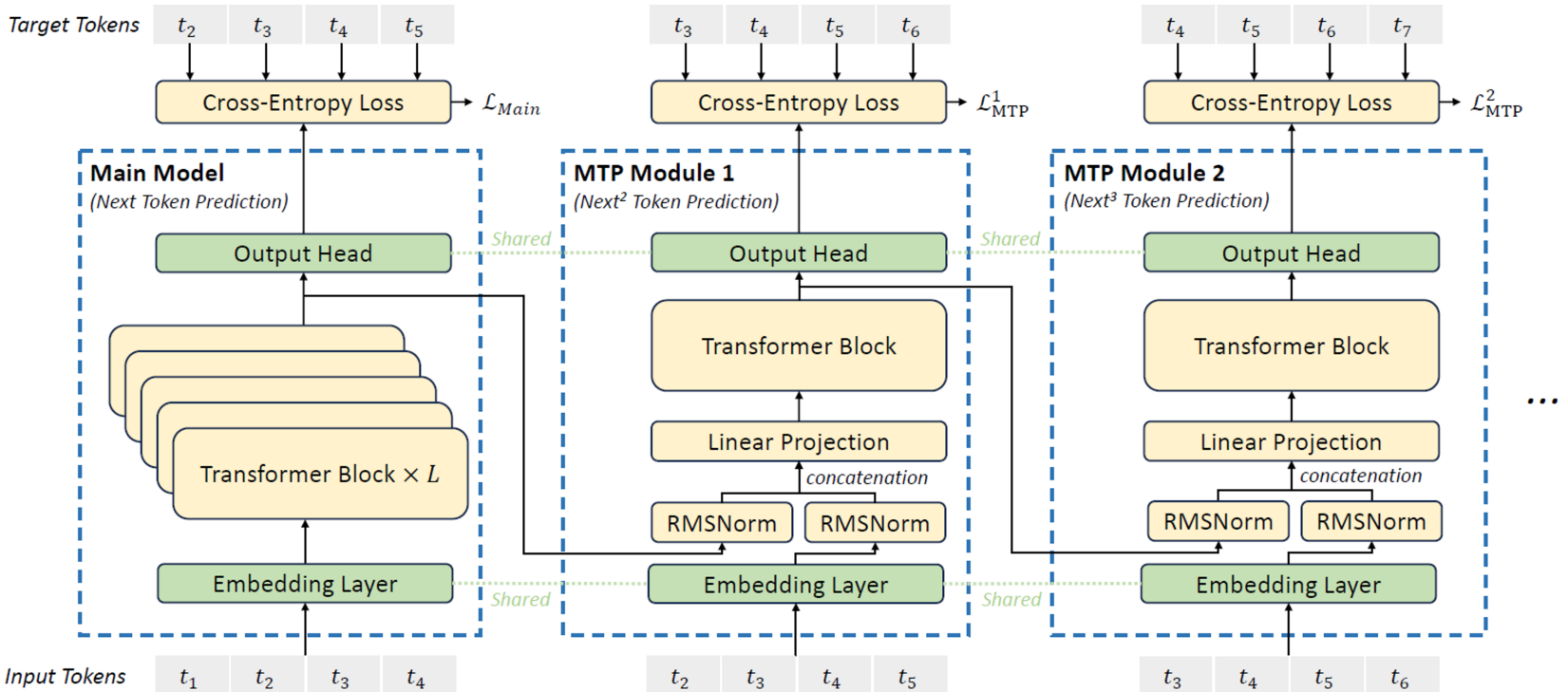
$$\mathcal{L}_{\text{Bal}} = \alpha \sum_{i=1}^{N_r} f_i P_i,$$

$$f_i = \frac{N_r}{K_r T} \sum_{t=1}^{T} \mathbb{1}\left(s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leqslant j \leqslant N_r\}, K_r)\right),$$

$$s'_{i,t} = \frac{s_{i,t}}{\sum_{j=1}^{N_r} s_{j,t}},$$

$$P_i = \frac{1}{T} \sum_{t=1}^{T} s'_{i,t},$$

# Multi-Token Prediction

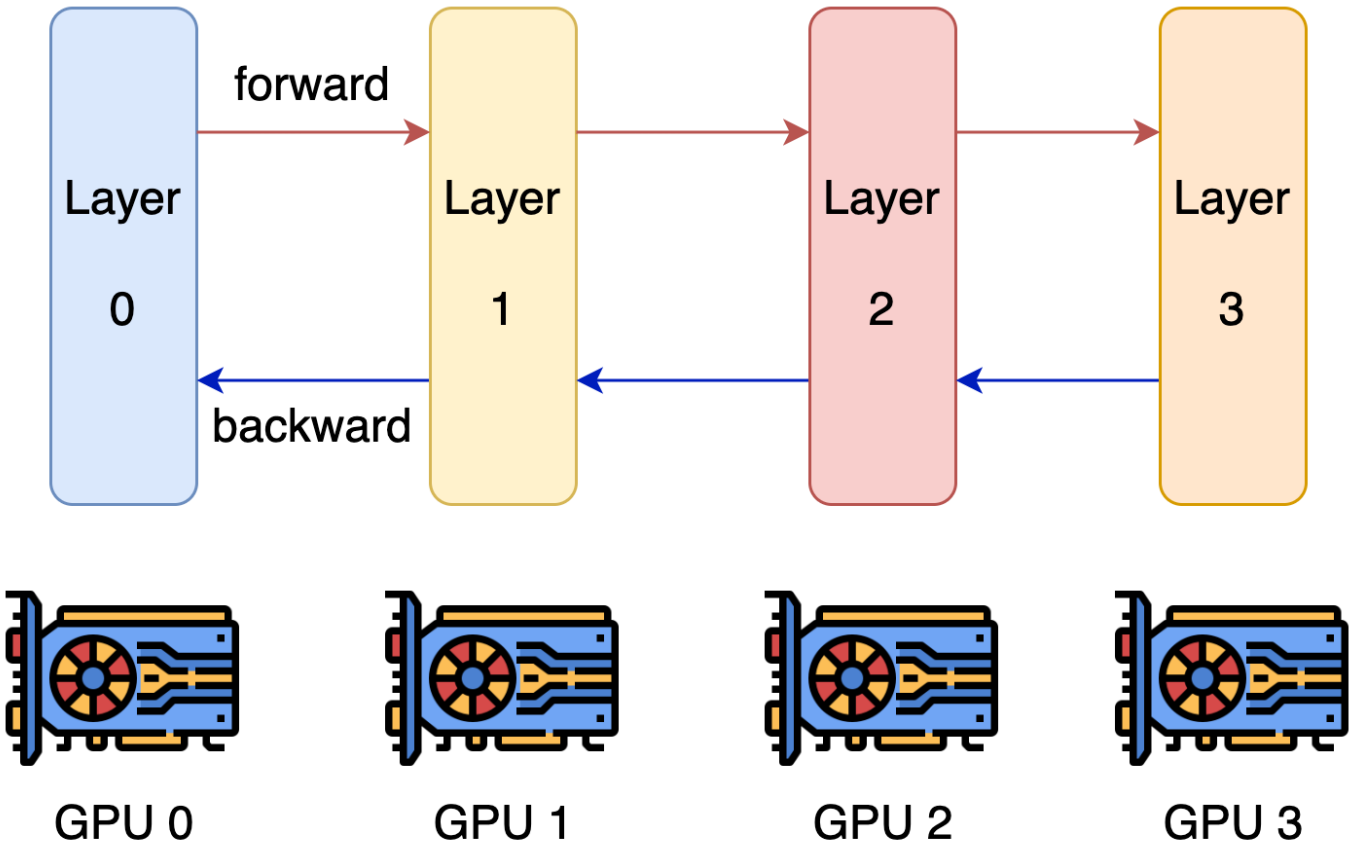



$$\mathcal{L}^k_{\text{MTP}} = \text{CrossEntropy}(P^k_{2+k:T+1}, t_{2+k:T+1}) = -\frac{1}{T}\sum_{i=2+k}^{T+1} \log P^k_i[t_i],$$

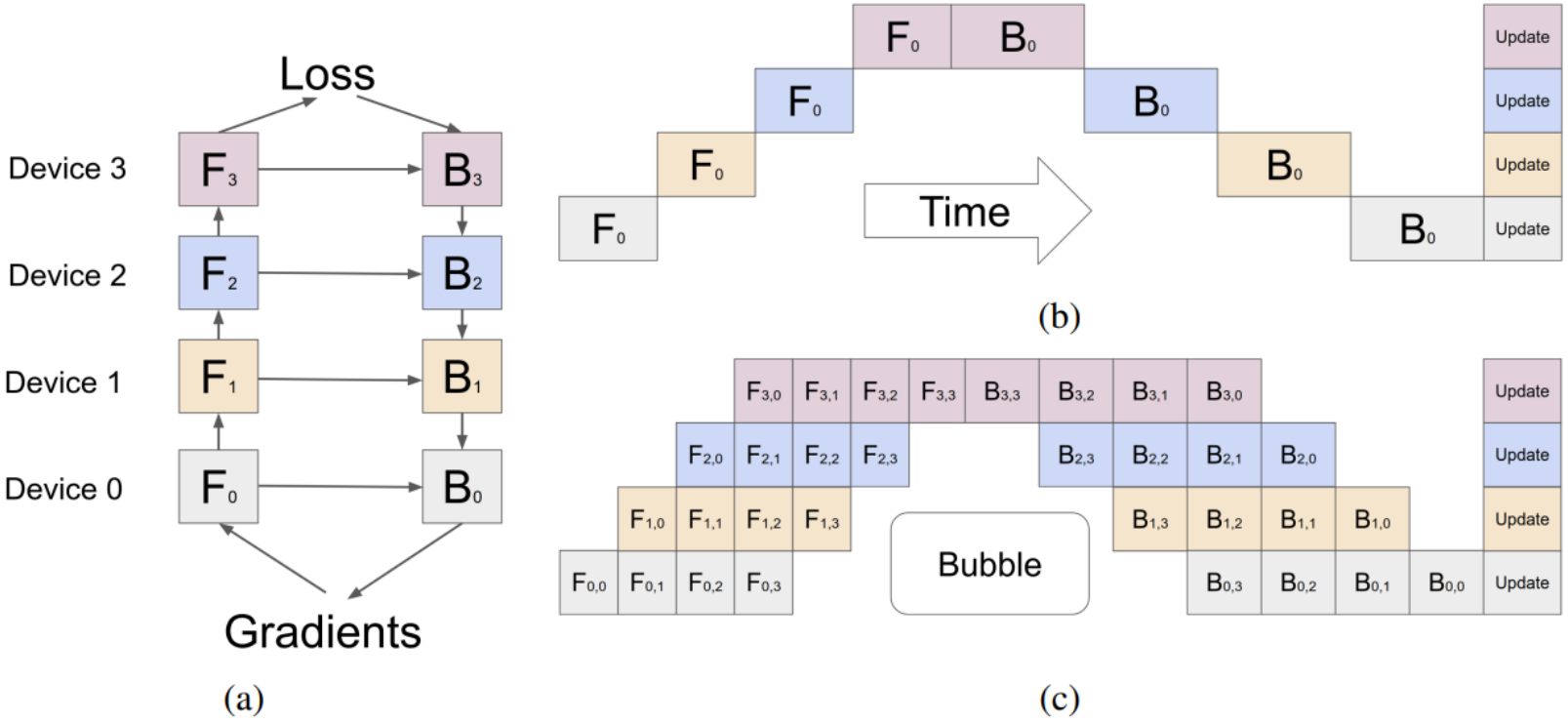
# Contents



# Background – Model Parallel

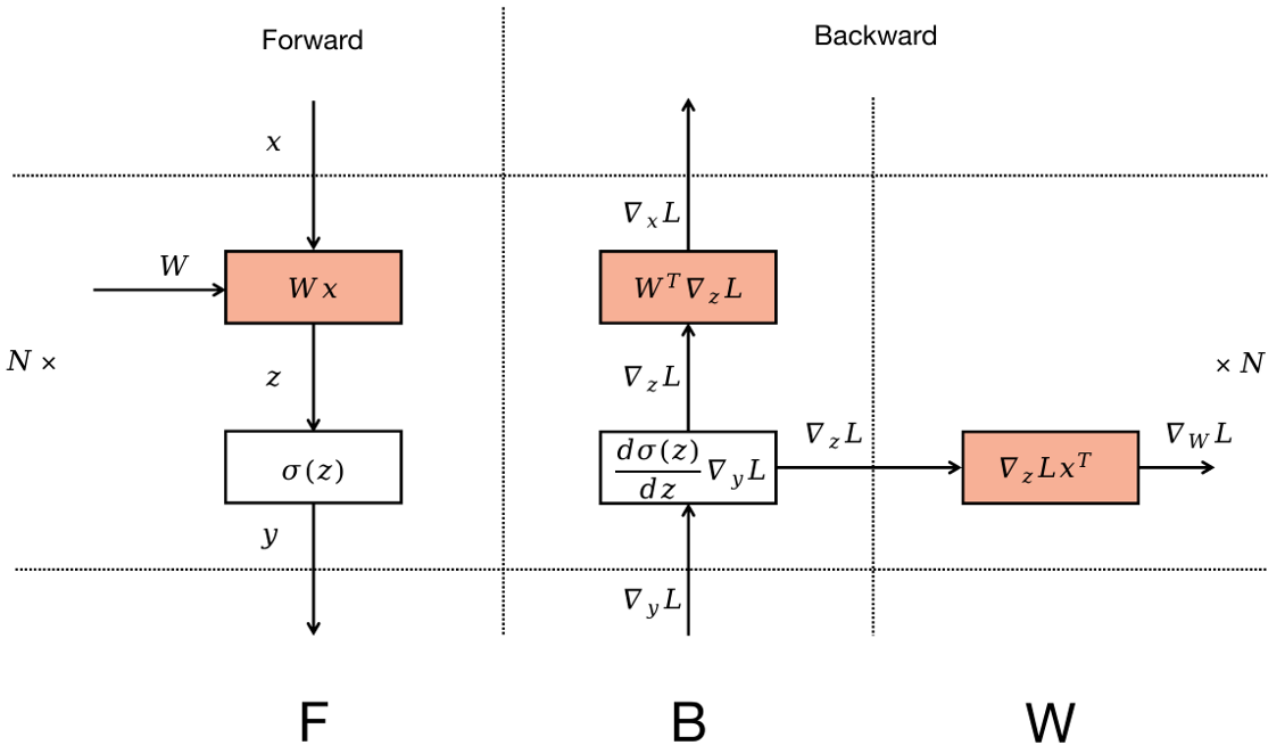


# Background – Pipeline Parallel

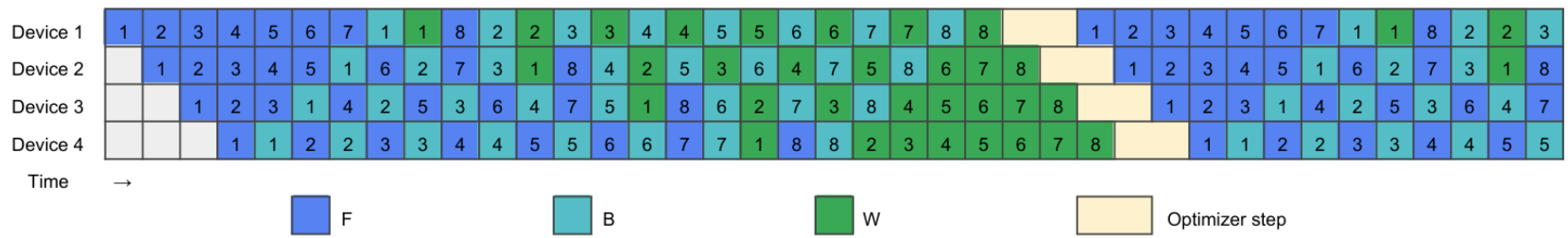



DeepSeek divides each chunk into four components: attention, all-to-all dispatch, MLP, and all-to-all combine

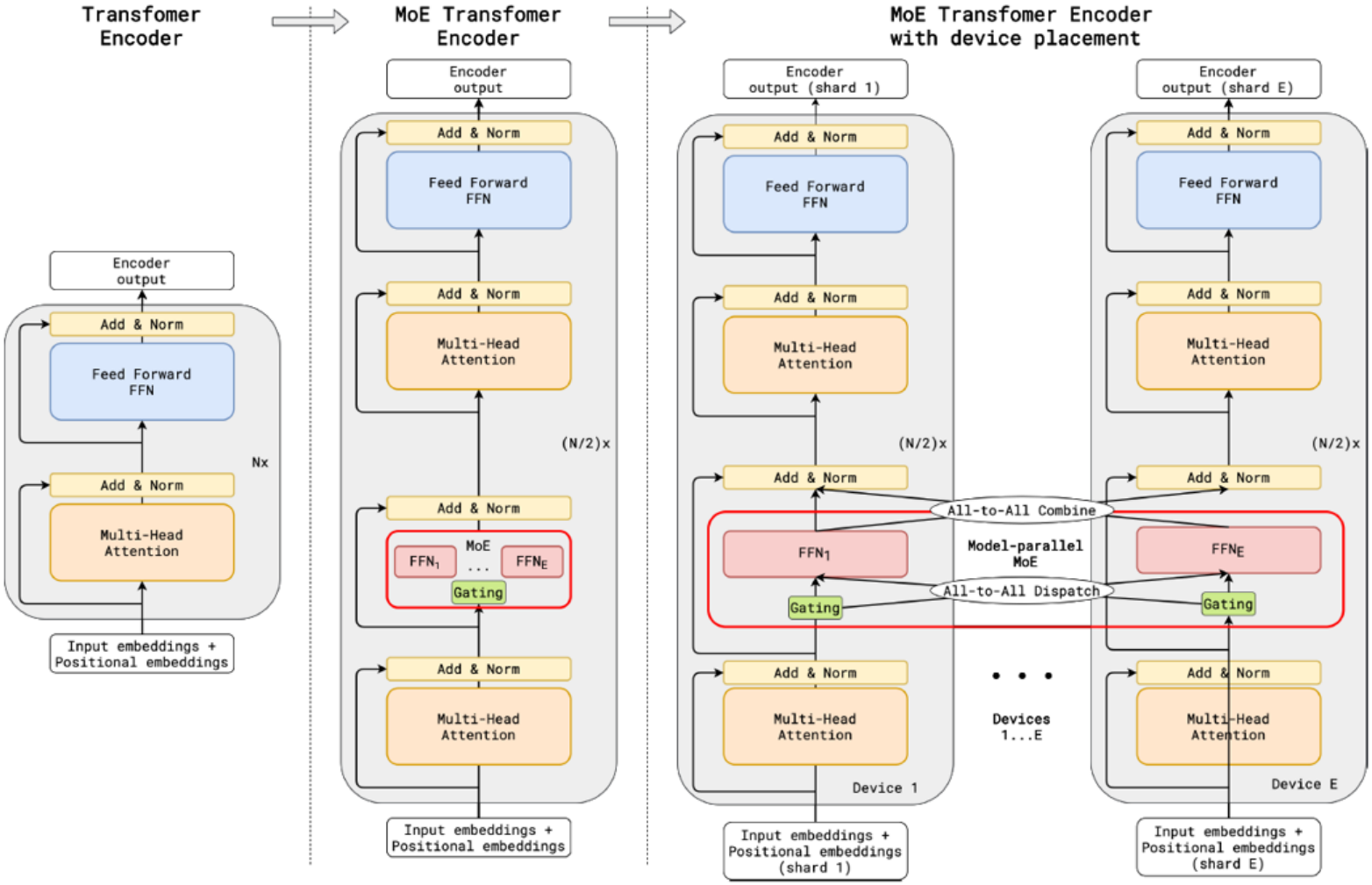
# Attention and MLP



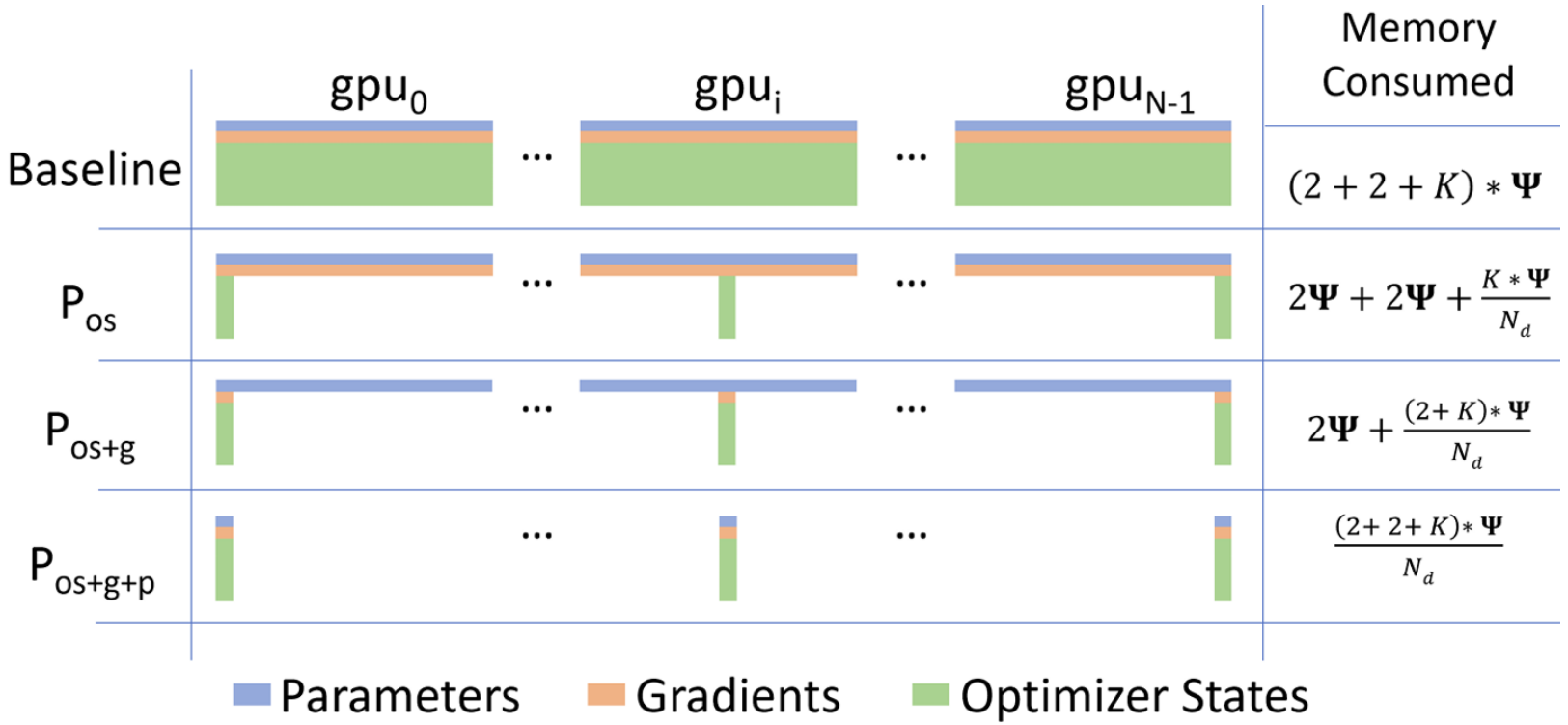
# ZERO BUBBLE PIPELINE PARALLELISM



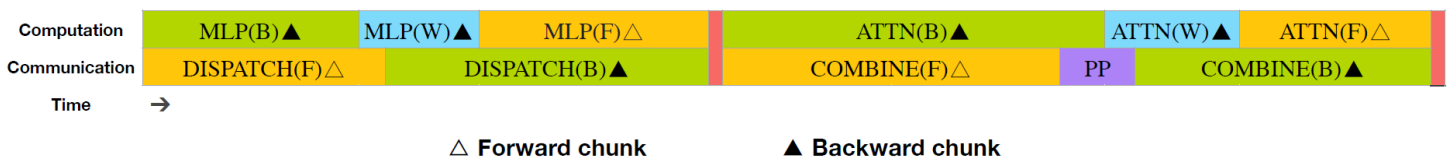
# All-to-All Combine/Dispatch



# ZeRO-123


gpu0
gpui
gpuN-1
Memory Consumed
Baseline
(2 + 2 + K) * Ψ
Pos
2Ψ + 2Ψ + K * Ψ / Nd
Pos+g
2Ψ + (2+ K)* Ψ / Nd
Pos+g+p
(2+ 2+ K)* Ψ / Nd
Parameters
Gradients
Optimizer States

# Overlapping strategy



# DualPipe


Device 0
Device 1
Device 2
Device 3
Device 4
Device 5
Device 6
Device 7
Time
Forward
Backward
Backward for input
Backward for weights
Overlapped forward & Backward

# Contents



# FP8 Format

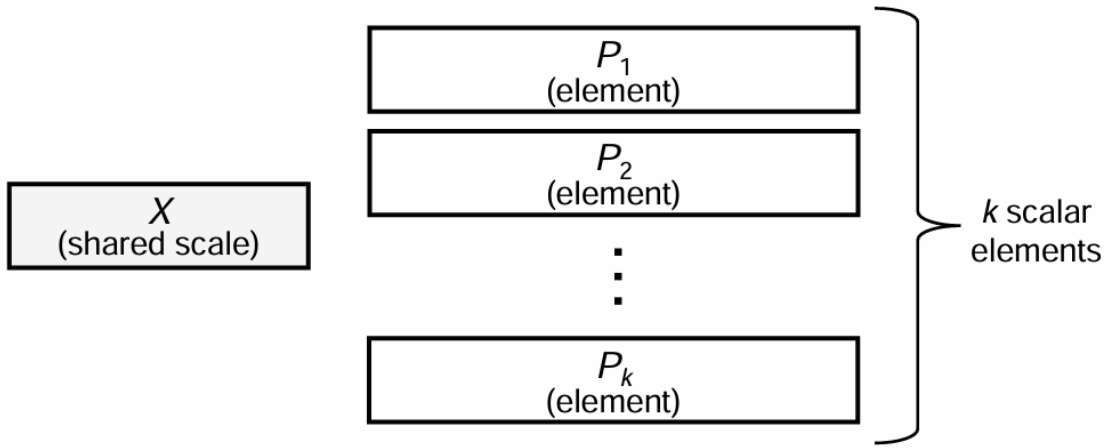
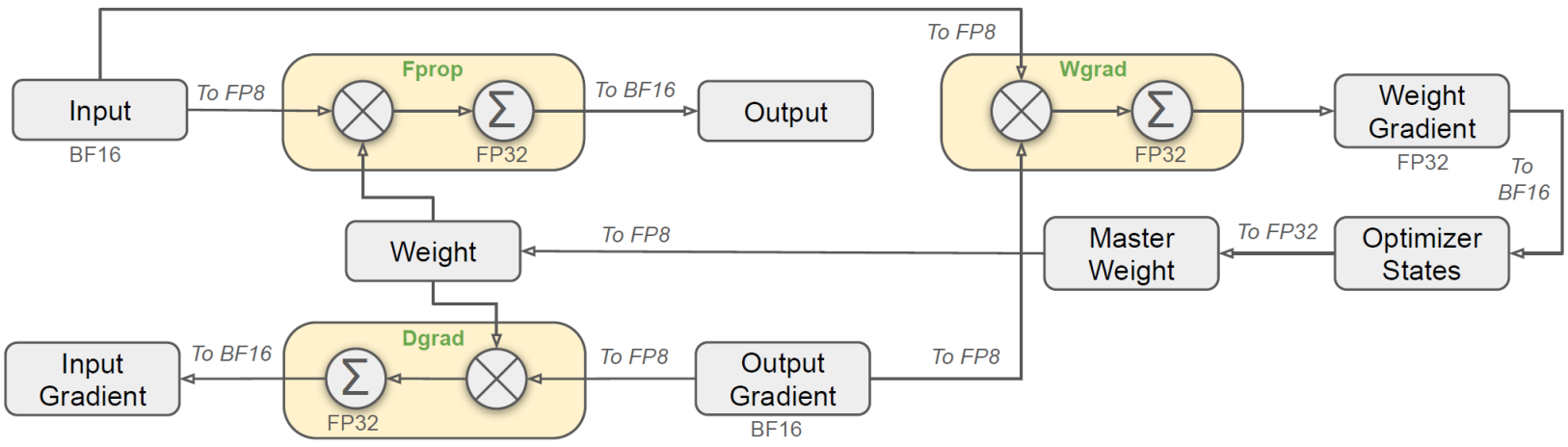



Figure 1: A single block in a Microscaling data format. The block encodes a vector of $k$ numbers, each with value $XP_i$.

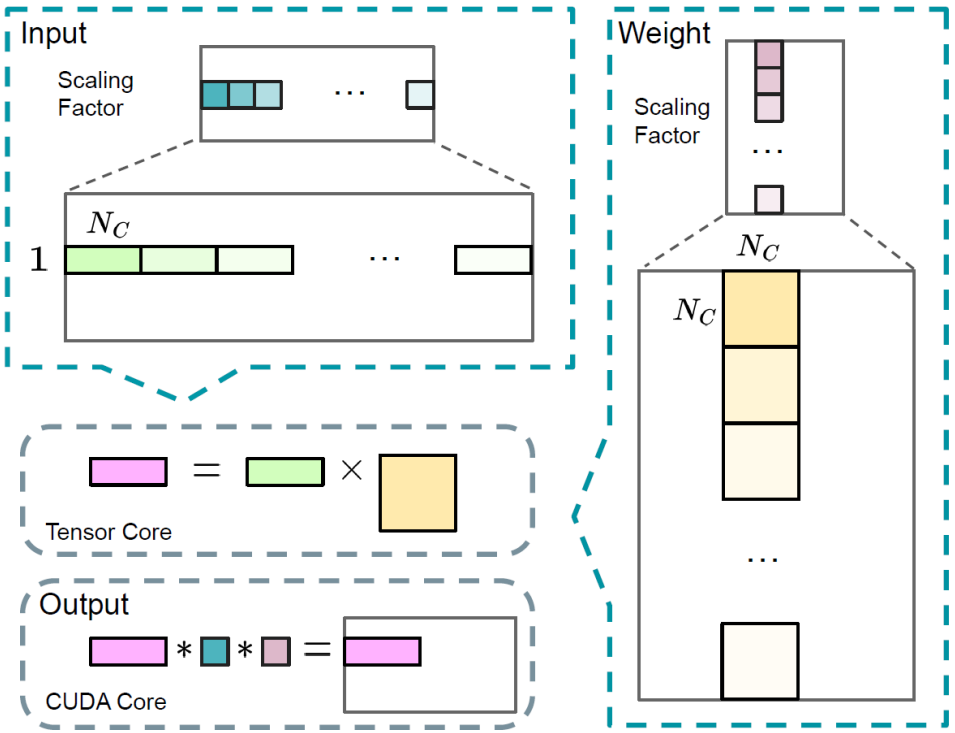
Let $\{v_i\}_{i=1}^{k}$ be the $k$ real numbers represented in an MX block. The value of each number can be inferred as follows:

- If $X = \text{NaN}$, then $v_i = \text{NaN}$ for all $i$
- If $|XP_i| > Vmax_{Float32}$ then $v_i$ is implementation-defined
- Otherwise, $v_i = XP_i$

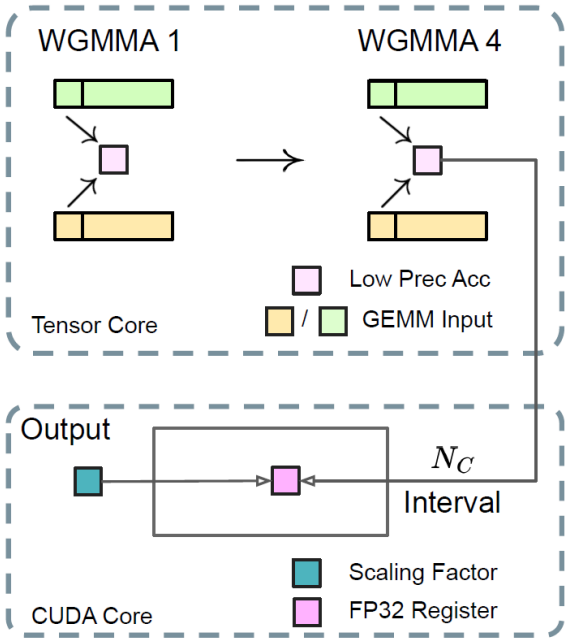
# FP8 Format


To FP8
Input
BF16
To FP8
Fprop
FP32
To BF16
Output
Wgrad
FP32
Weight Gradient
FP32
To BF16
Weight
To FP8
Master Weight
To FP32
Optimizer States
Input Gradient
To BF16
Dgrad
FP32
To FP8
Output Gradient
BF16
To FP8

# FP8 Format




(a) Fine-grained quantization




(b) Increasing accumulation precision

**THANKS!**