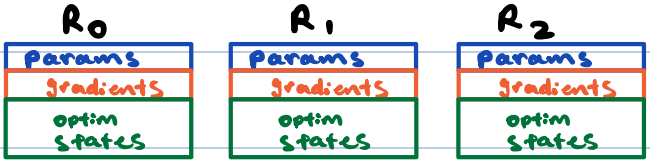
Data Parallelism & ZeRO

40_ddp

Baseline



"Things we want to be the same"

Rank 0

X_local
y_local

This is our sharded data: different data on every rank!

Params = $\begin{bmatrix} W_1 \\ b_1 \\ W_2 \\ b_2 \end{bmatrix}$

flatten_dense_tensors

flat_params = $[\leftarrow W_1 - b_1 - W_2 - b_2 \rightarrow]$

1D contiguous array

Rank 1 | Rank 2

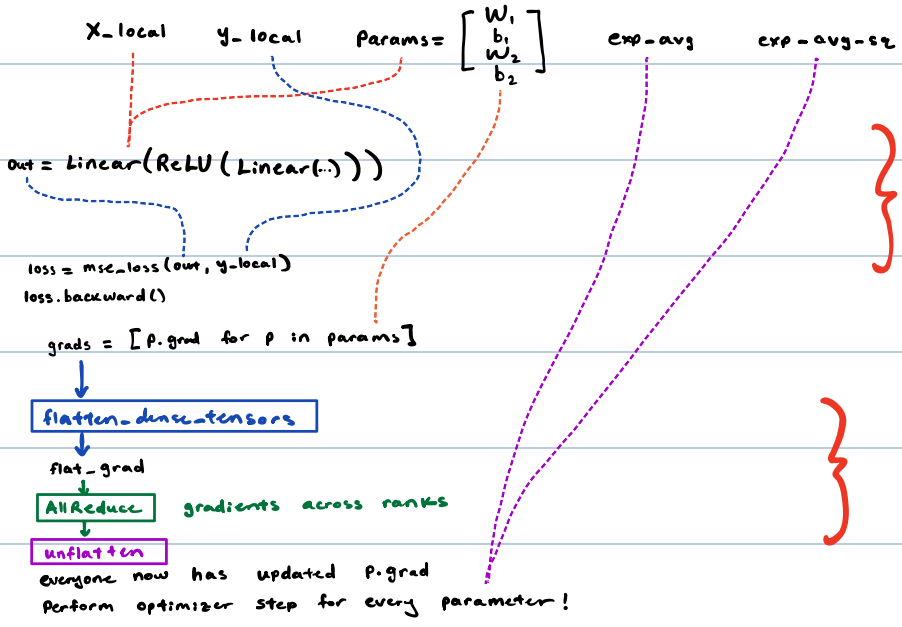
broadcast

unflatten_dense_tensors

Params = $\begin{bmatrix} W_1 \\ b_1 \\ W_2 \\ b_2 \end{bmatrix}$ — "Every rank has the same params now"

Repeat: flatten, broadcast, unflatten from rank 0 for exp_avg, exp_avg_sq

Broadcast Parameters to ranks



Run Fwd + Bwd Pass

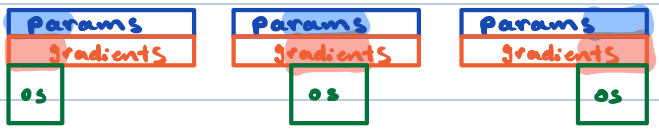
Allreduce gradients

41_ Zero 1

- same as DDP but shard optimizer state

FP32 master weights
FP32 Adam 1st moment
FP32 Adam 2nd moment

Zero 1



updating only this slice, then allgathering!

Review of Adam:

$$w_{t+1} = w_t - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} + \lambda w_t \right)$$

$\varepsilon \leftarrow 10^{-8}$

$\frac{m_t}{1-\beta_1^t} \leftarrow m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

$\frac{v_t}{1-\beta_2^t} \leftarrow v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Adam stores states $[w_t, m_t, v_t]$ for every param!

Rank X

m v

Params | exp_avg_part | exp_avg_sq_part ← Sharded amongst ranks!

flatten
broadcast
unflatten ← everyone has full params!

} Broadcast params

Forward + Backward pass

} Fwd + Bwd

flatten
allreduce
unflatten ← everyone has full true gradients!

} Grad Allreduce

everyone has full grads but updates just a slice

Start = rank x part

g_part, w_part } slice of the shared gradients & weights

m_part
v_part

} update small slice

Like 40, but m & v are sharded so Adam updates only its weight shard

Run local AdamW weight update

Allgather (w_part)

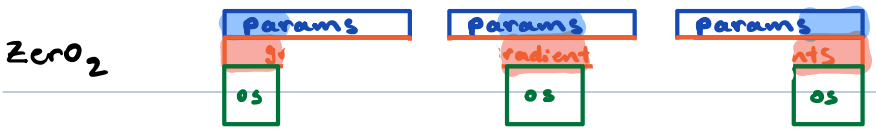
AG params back

now everyone has full updated weights!

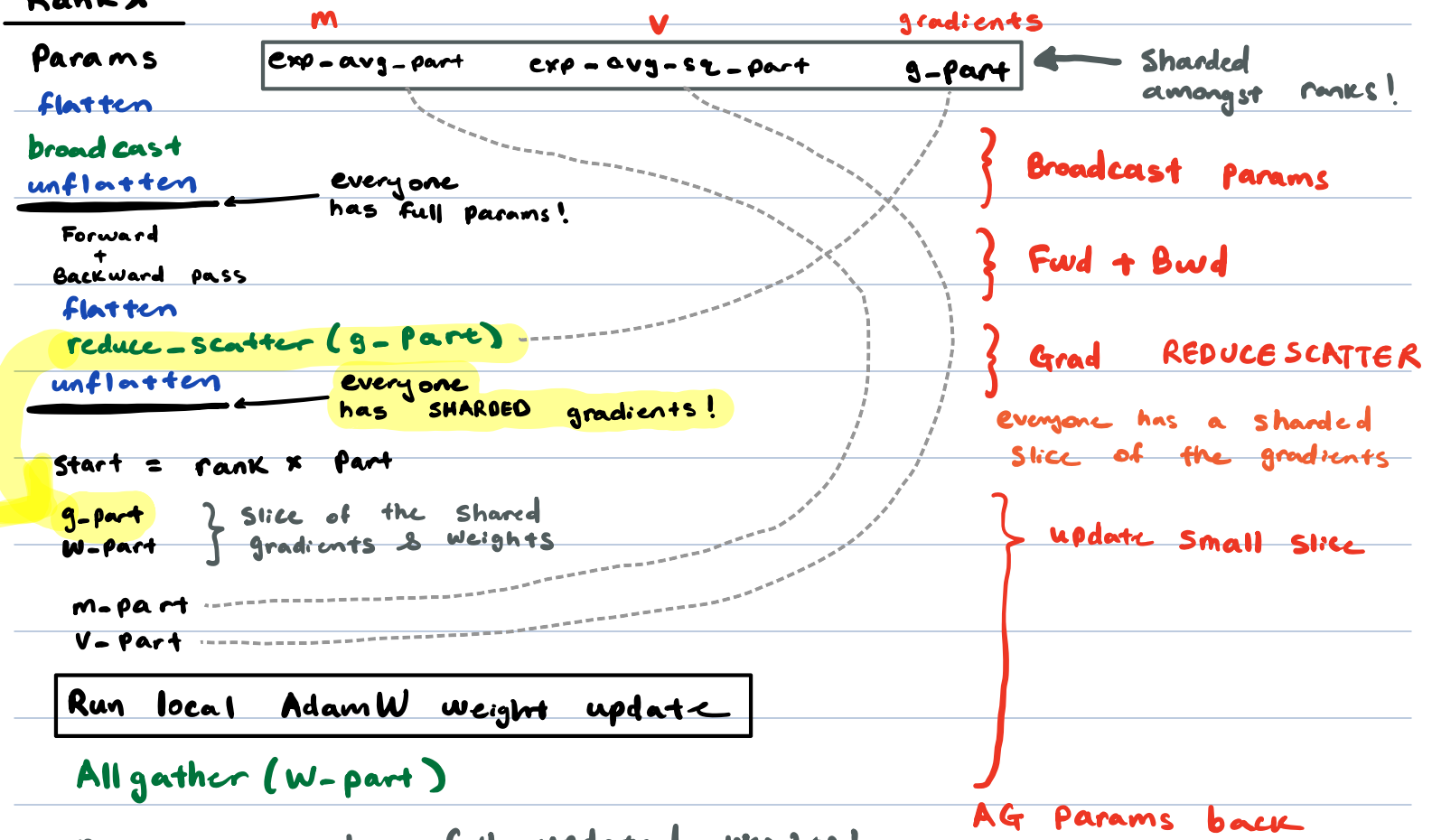
- In real zero1, there's master FP32 weights that're also sharded!
  - ↳ Reconstruct & free parameters/gradients as needed
  - ↳ integrated into a training loop

- This is deepspeed's zero1: gradients are NOT sharded

42_zero2_optimizer



Rank X



- "REAL" Zero2:

After doing bwd pass:
- ↳ Reduce & P2P send to right rank
- ↳ Free when not needed

Each rank updates their gradient slice shard
Allgather the parameters

=

TBD...

Coming soon...

ZeRO$_3$ / FSDP Visuals

Here, we also shard params!

ZeRO$_3$



This uses:

- 2 allgathers
- 1 reduceScatter

Optimized via:
- freezing requested params / gradients
- comp. comm. overlap

Ring Attention

Attention is 3 phases:

- Find row-max m
- Let $s = \sum e^{\text{row-elem} - m}$
- Divide each row-elem by s

Each rank has:

B = seqlen / # ranks

$Q_{local} = [\ d \times B\ ]$

$K_{local} = [\ d \times B\ ]$

$V_{local} = [\ d \times B\ ]$

$\overrightarrow{running_max} = [\ 1 \times B\ ]$

$\overrightarrow{running_denom_sum} = [\ 1 \times B\ ]$

$out^{running}_{accum} = [\ d \times B\ ]$

From other ranks!

$K_{recv} =$

$V_{recv} =$

Let $S_{block} = \frac{1}{\sqrt{d}} \times (Q_{local} \cdot K_{recv}^{T}) = [\cdots]_{B \times B}$

$\downarrow$ rowmax

$\overrightarrow{local_max} = [\vdots]_{B \times 1}$

$\overrightarrow{new_max} = \max(\overrightarrow{running_{max}}, \overrightarrow{local_{max}}) = [\vdots]_{B \times 1}$

Running Correction Factor $= e^{(\overrightarrow{running_{max}} - \overrightarrow{new_max})} = [\vdots]_{B \times 1}$

Local Correction Factor $= e^{(\overrightarrow{local_{max}} - \overrightarrow{new_max})} = [\vdots]_{B \times 1}$

Let's get the exps for our new block:

$$\begin{bmatrix} \text{final output intermediate exps} \\ B \times B \end{bmatrix} = e^{\left(S_{\text{block}} - \overrightarrow{\text{new_max}}\right)}$$

S_{block}: $B \times B$; $\overrightarrow{\text{new_max}}$: $B \times 1$

Subtract rowwise max & exponentiate matrix elements

↑ Still need to divide Σe!

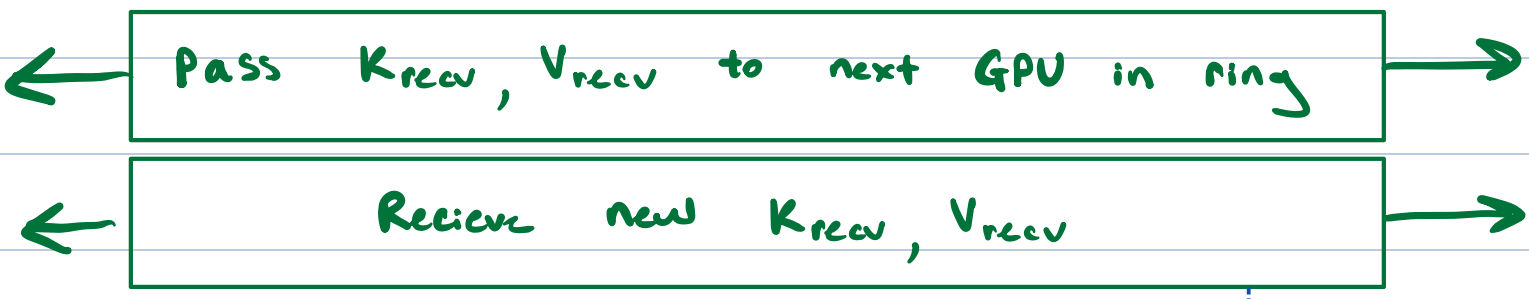
$$\overrightarrow{\text{final_denom_sum}} = \left(\text{Running Correction Factor} \odot \overrightarrow{\text{running_denom_sum}}\right) + \sum_{\text{rows}} \begin{bmatrix} \text{final output intermediate exps} \\ B \times B \end{bmatrix}$$

$\overrightarrow{\text{final_denom_sum}}$: $B \times 1$; Running Correction Factor: $B \times 1$; $\overrightarrow{\text{running_denom_sum}}$: $B \times 1$; $\sum_{\text{rows}}$ result: $B \times 1$

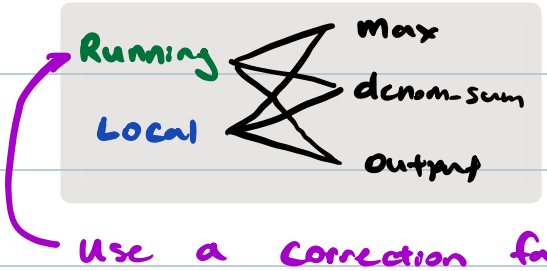
$$\begin{bmatrix} \text{final_numerators} \\ B \times d \end{bmatrix} = \left(\text{Running Correction Factor} \odot \text{running out}_{\text{accum}}\right) + \left(\begin{bmatrix} \text{final output intermediate exps} \\ B \times B \end{bmatrix} \cdot V_{\text{recv}}\right)$$

Running Correction Factor: $B \times 1$; $\text{running out}_{\text{accum}}$: $B \times d$; V_{recv}: $B \times d$

$$\underset{B \times d}{\text{running out}_{\text{accum}}} = \begin{bmatrix} \text{final_numerators} \\ B \times d \end{bmatrix} / \underset{B \times 1}{\overrightarrow{\text{final_denom_sum}}}$$



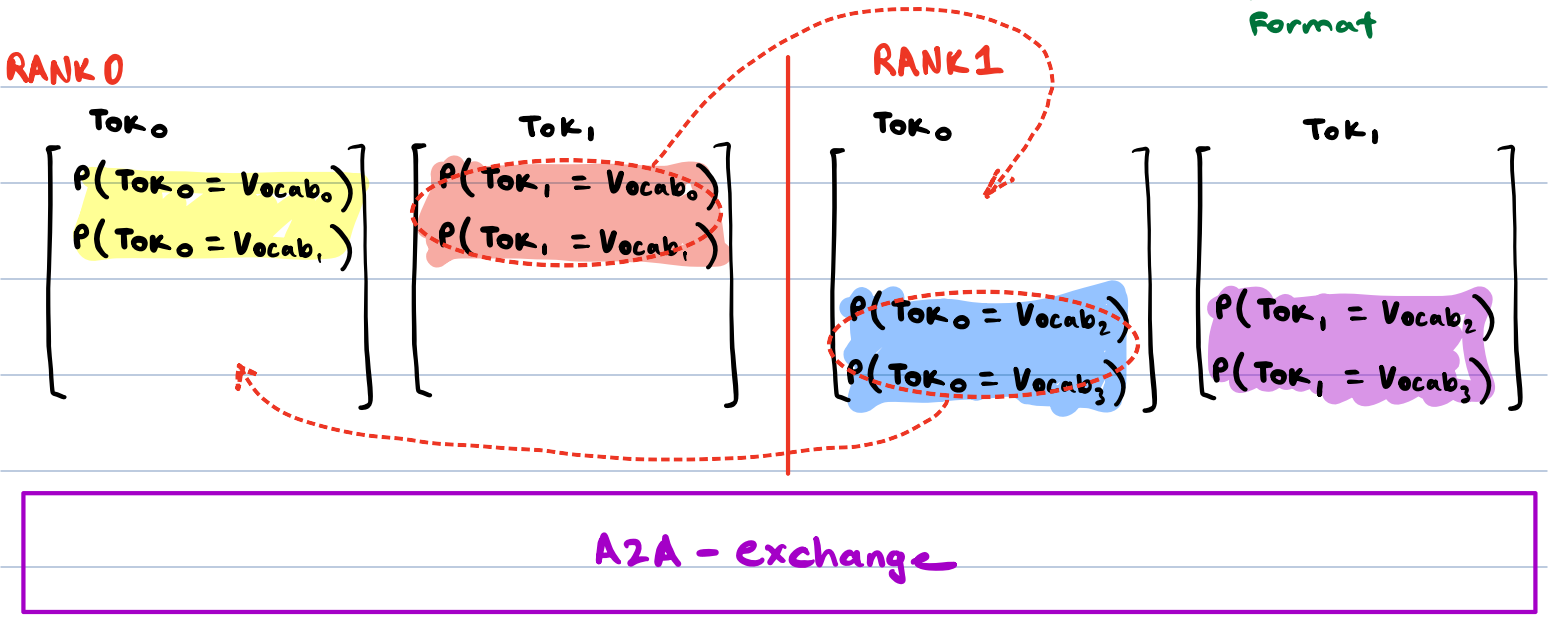
Key behind RA:



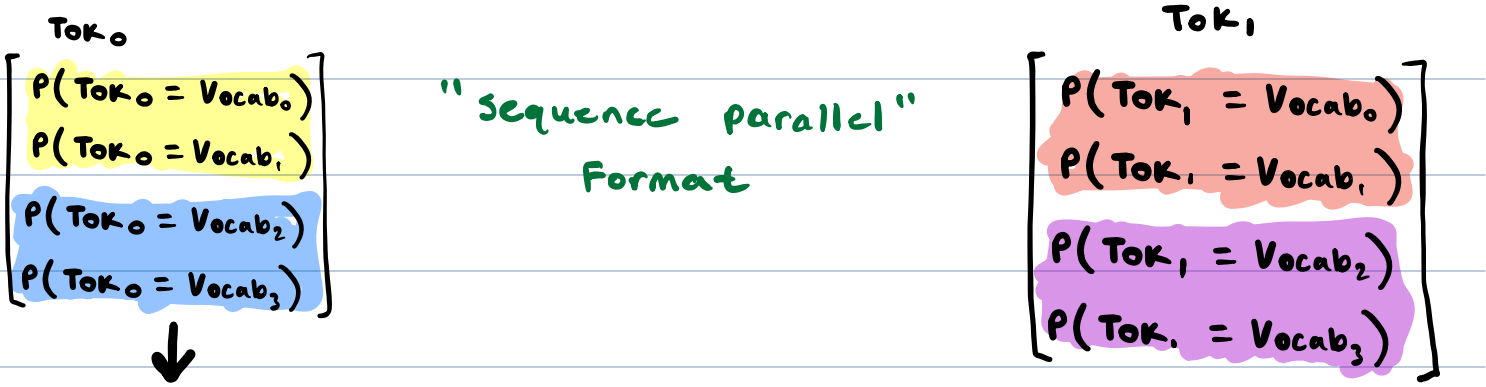
82 - vocab - parallel - top-K

At the very end, our model outputs:

"Vocabulary parallel" Format



A2A - exchange

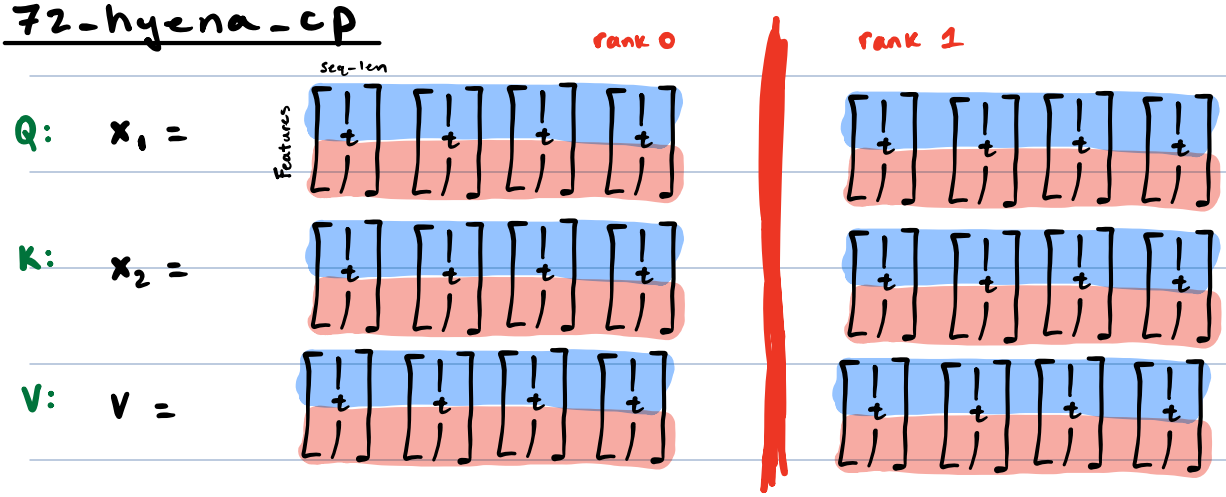


allgather

P(Tok0 = Vocab3) P(Tok1 = Vocab2)

P(Tok0 = Vocab3) P(Tok1 = Vocab2)

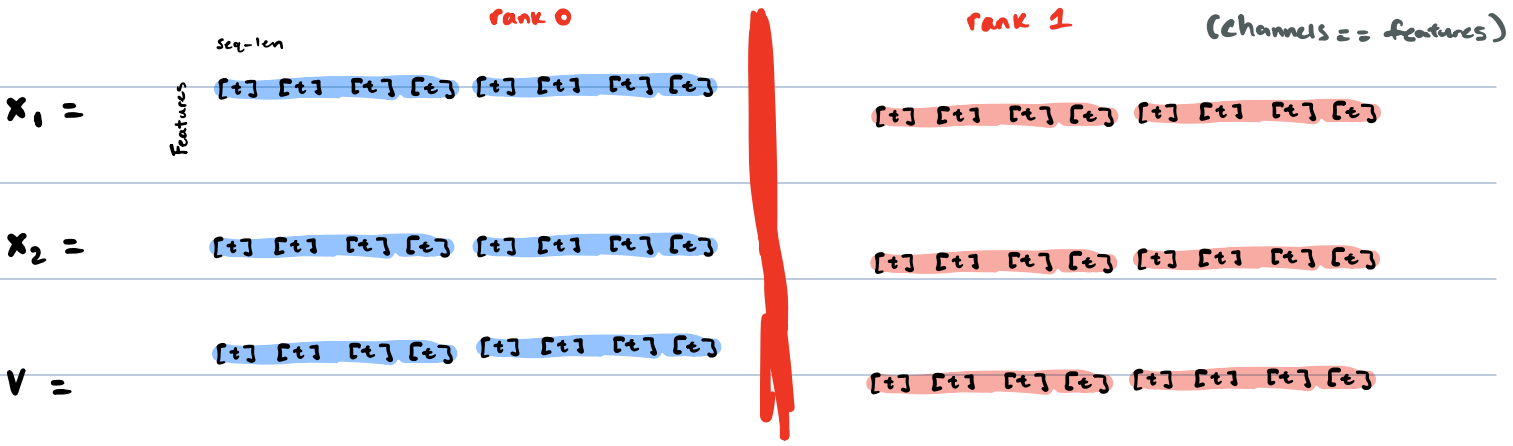
72_hyena_cp



A2A_SPLIT_FULL

full seq. diff channels

(channels == features)



$z = x_2 \odot v$

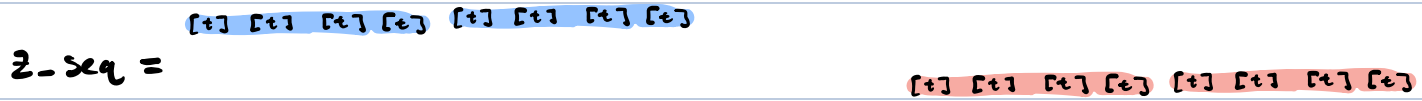
FFT(z)

$z = x_1 \odot z$

$z = x_2 \odot v$

FFT(z)

$z = x_1 \odot z$



A2A_FULL_SPLIT



LLM optimizations

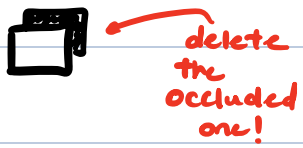
- Bypass A2A w/ NVLink

81_sam3_IOU_suppression

- each gpu has its own mask + confidence score



- Collect masks & scores
  - ID overlapping boxes & mark those above IoU threshold
  - Suppress (delete) occluded (blocked) boxes (> IoU threshold)



Rank$_0$

masks_local = [[8], [ ]]

scores_L = [0.5, 0.7]

Rank$_1$

masks_local = [[ ]]

scores_L = [0.1]

Variable_all_gather

last-occluded

masks_global = [[8], [ ], [ ]]

scores_global = [0.5, 0.7, 0.1]

↓

Compute overlaps & IoU

- LLM bitpacks pixels into an int_32