BATTLE-TESTING SWARM

The SWARM learns

MIKA SENGHAAS

EPFL

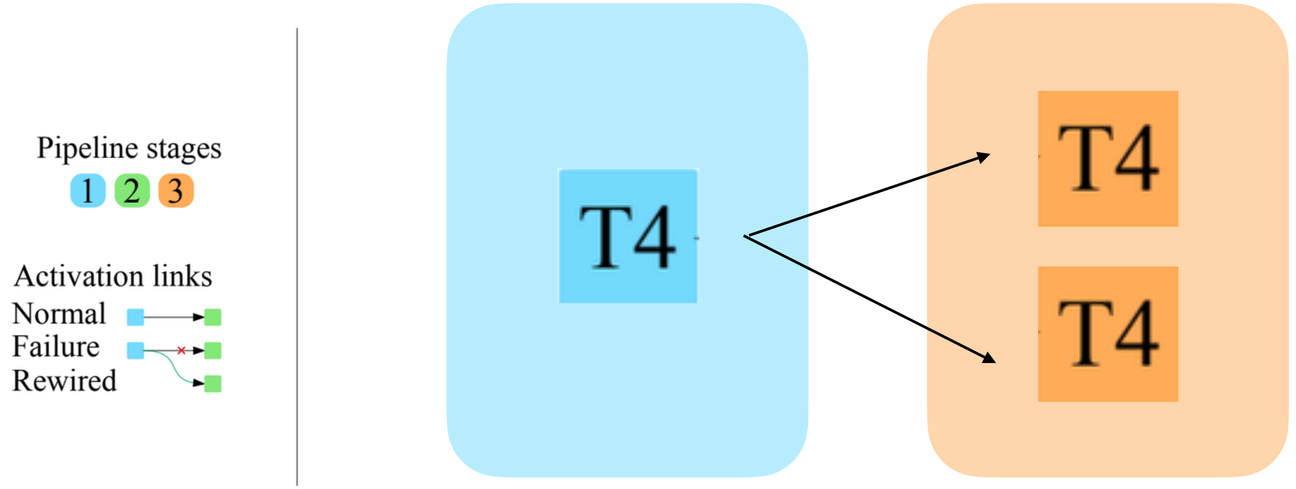
Next Steps

1. ~~Verify FedAvg Baseline~~
2. ~~Implement traditional FL~~
3. Implement SWARM
4. Experiments

Where we start: Forward pass

Send hidden states from one worker to one of two in next stage

Last Week



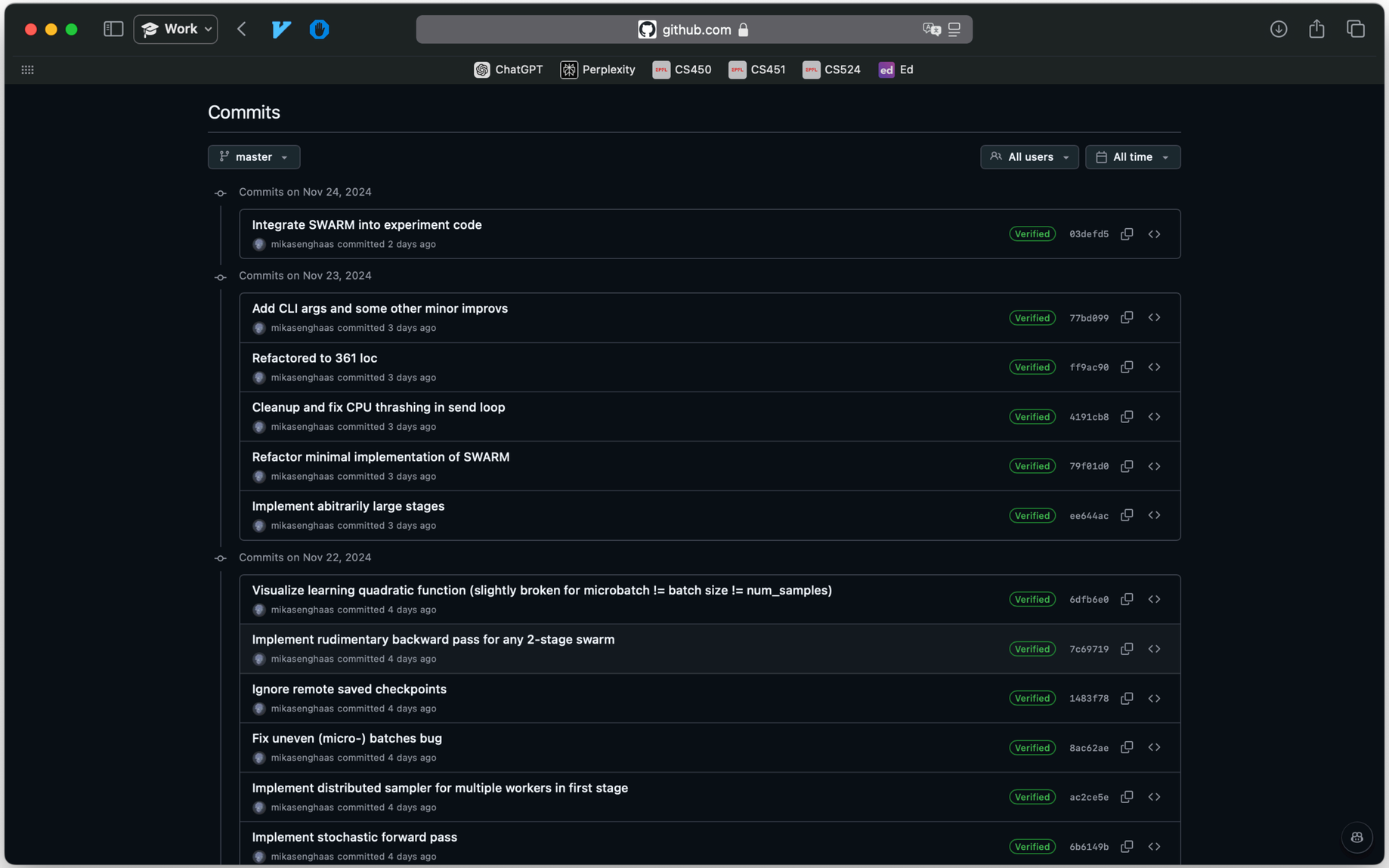
Works with sender/receiver threads and queues

Current problem: torch.dist.recv() is blocking and torch.dist.irecv() cannot be cancelled leading to memory errors for wrongly allocated buffers on step > 1

Idea: Pre-compute full computation graph per step? (but this is sync)

Later

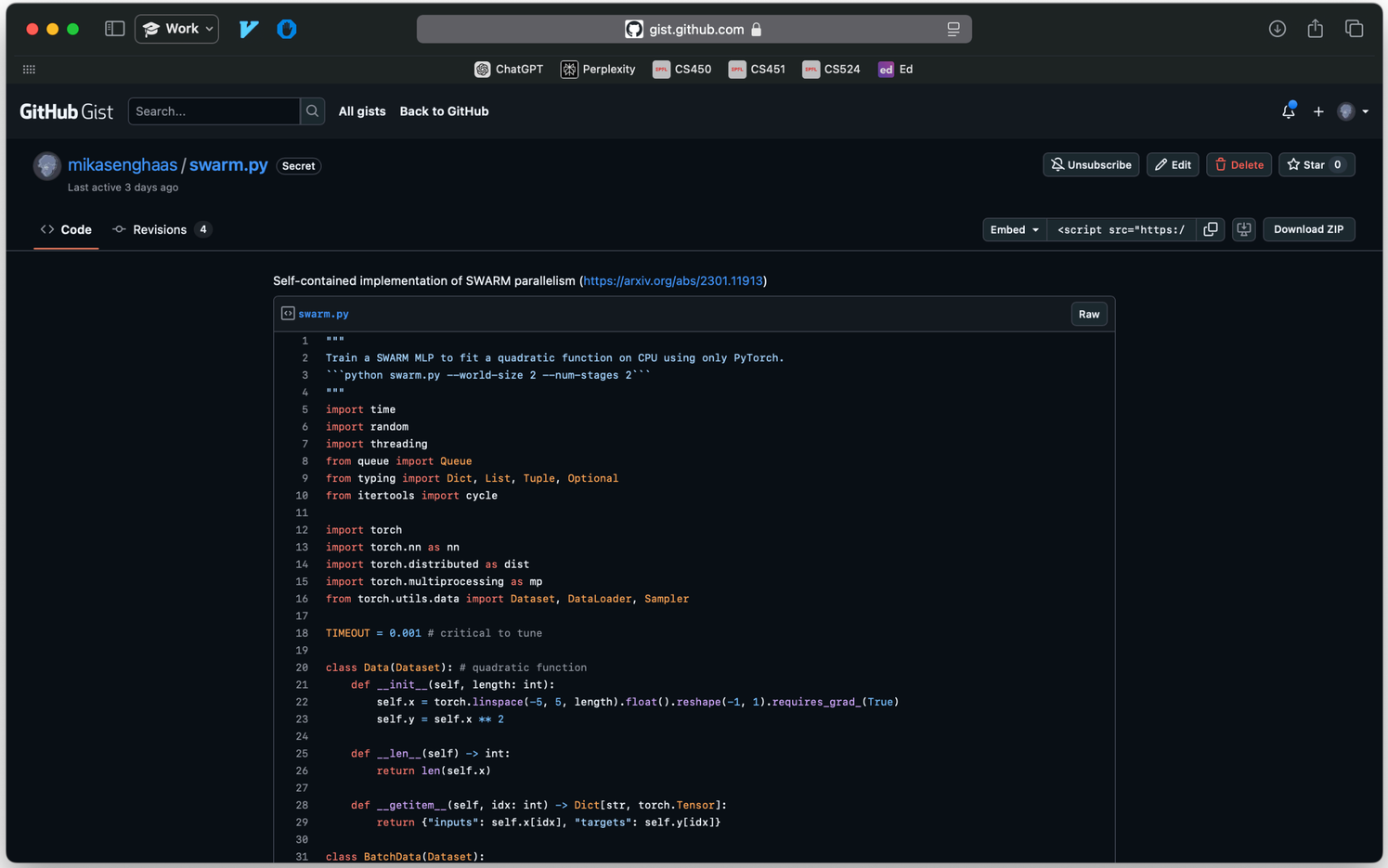
- Backward pass (Backtrace computation graph to send gradients)
- Distributed data loader (Ensure that workers in first stage saple disjoint data shards, st. at end of step each sample is backpropped exactly once)
- Optimize with buffers (have to be adjusted per device VRAM)
- Make work over public IP (Tailscale)
- *Dynamic resizing of world (reinitializing process group?, …) and swarm balancing*


Work
github.com
ChatGPT
Perplexity
CS450
CS451
CS524
Ed
Commits
master
All users
All time
Commits on Nov 24, 2024
Integrate SWARM into experiment code
mikasenghaas committed 2 days ago
Verified
03defd5
Commits on Nov 23, 2024
Add CLI args and some other minor improvs
mikasenghaas committed 3 days ago
Verified
77bd099
Refactored to 361 loc
mikasenghaas committed 3 days ago
Verified
ff9ac90
Cleanup and fix CPU thrashing in send loop
mikasenghaas committed 3 days ago
Verified
4191cb8
Refactor minimal implementation of SWARM
mikasenghaas committed 3 days ago
Verified
79f01d0
Implement abitrarily large stages
mikasenghaas committed 3 days ago
Verified
ee644ac
Commits on Nov 22, 2024
Visualize learning quadratic function (slightly broken for microbatch != batch size != num_samples)
mikasenghaas committed 4 days ago
Verified
6dfb6e0
Implement rudimentary backward pass for any 2-stage swarm
mikasenghaas committed 4 days ago
Verified
7c69719
Ignore remote saved checkpoints
mikasenghaas committed 4 days ago
Verified
1483f78
Fix uneven (micro-) batches bug
mikasenghaas committed 4 days ago
Verified
8ac62ae
Implement distributed sampler for multiple workers in first stage
mikasenghaas committed 4 days ago
Verified
ac2ce5e
Implement stochastic forward pass
mikasenghaas committed 4 days ago
Verified
6b6149b

https://gist.github.com/mikasenghaas/5fa1aa77ea69f187f531a5889983c249

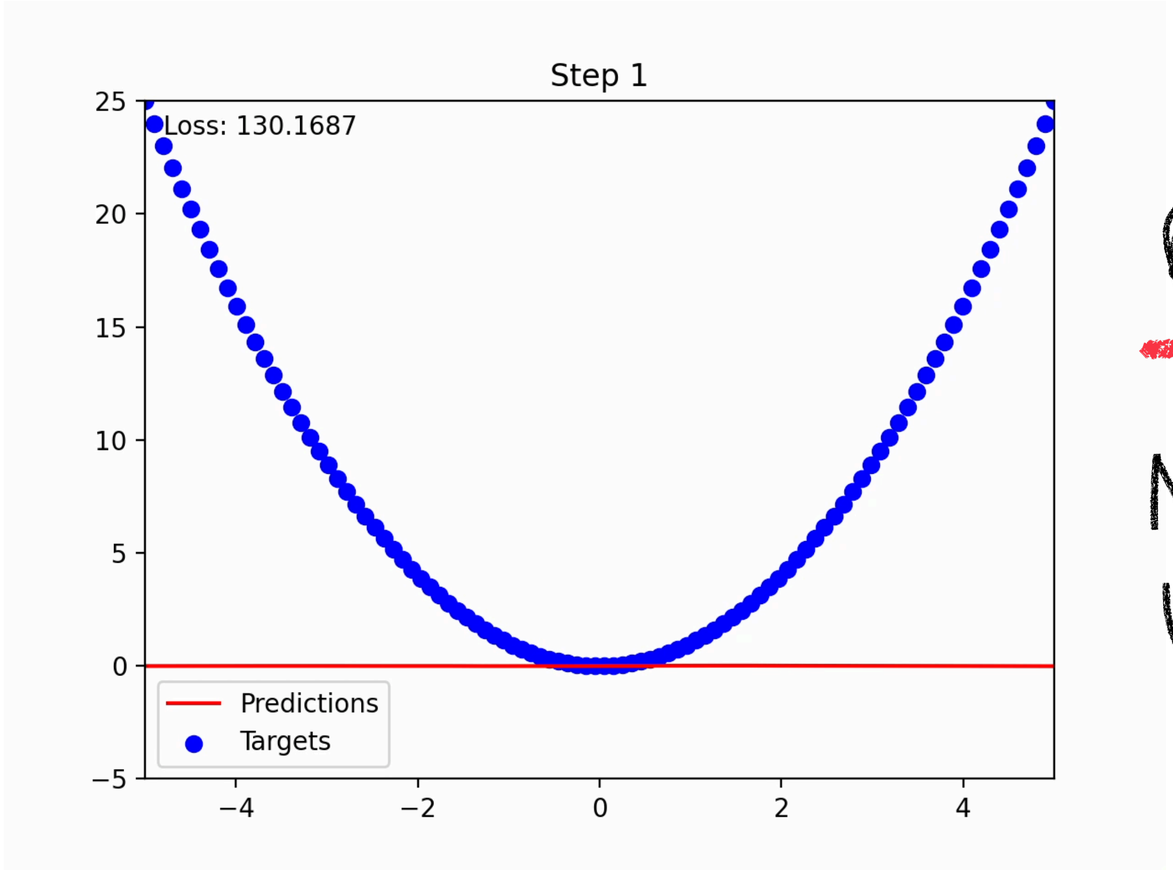
350 LOC

Pure PyTorch



Implements stochastic wiring

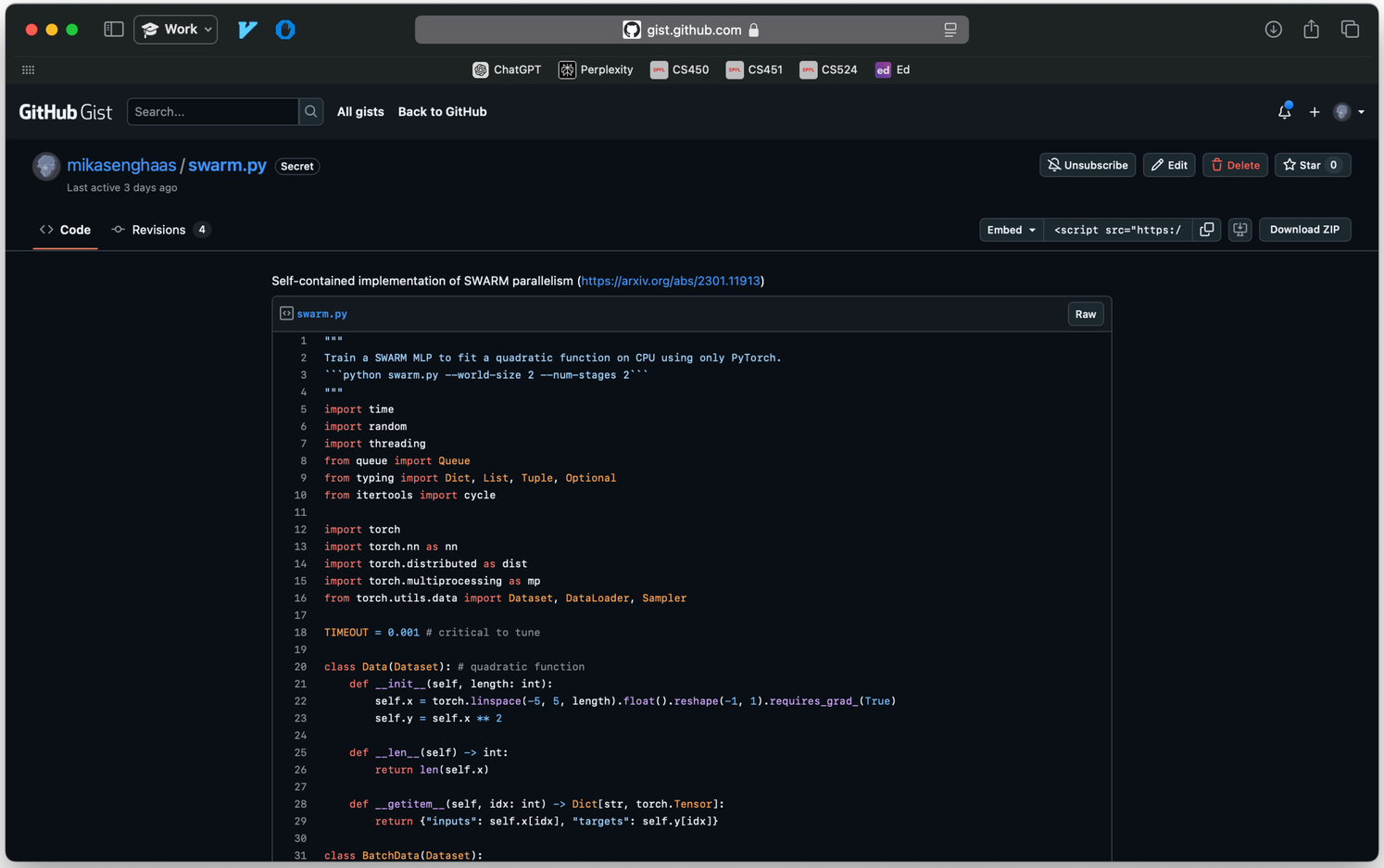
[ATM, fully uniform and not based on throughputs]



Proof of Concept

N workers, T stages SWARM learns the quadratic function

So, this is it?



- Maintain throughput estimates in SW
- Not fault-tolerant
  - → Dynamic World Size (Join / Leave)
  - → Node Rebalancing

What's Next?

1. Integrate w/ existing experiment codebase (LLM pre-training)
2. Define RQ -> Experiments
3. Develop feature subset enabling experiments

Direction 1: SWARM + DiLoCo

Obs SWARM is step-synchronous
⇒ stage peers all-reduce gradients every step

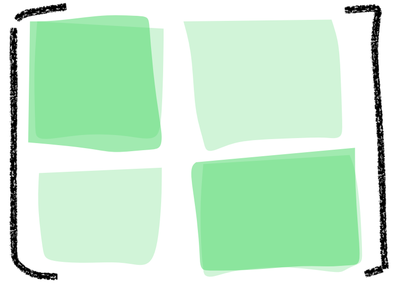
RQ Can we get away with sync every T steps?
⇒ T comm. reduction

Hypothesis DiLoCo's T=400, inner-outer optimizer scheme might be applicable

⇒ Setup SWARM with T = {1, 2, 4, ...} and observe eg. wall-clock to accuracy

Direction 2 : SWARM Init

Obs Consider workers $i \in [n]$ with comm matrix $M \in \mathbb{R}^{n \times n}$, M_{ij} = Network Speed between i, j.
Consider 2 data centres, then

$M =$  ← blocks represent intra-node connections

$\Rightarrow$ SWARM assigns workers to stages at random

RQ How to best allocate workers to stages?

$\Rightarrow$ Analyze comm. complexity as function of steps, stages, model

M = Steps, N stages, $|\theta|$ Model params

B, L, H = Batch, Sequence Length, Hidden Dim

Forward/Backward : $B\,(N-1)\,2\cdot BLH$ ← should consider DP

- B: Batch
- (N-1): Comm between stages
- 2: Fwd/Bwd
- BLH: Batch

All Reduce : $(N)\,\frac{|\theta|}{N}$ · All Reduce Complexity ·

- (N): In parallel