

BATTLE-TESTING SWARM

~~SETTING FOUNDATIONS~~

Going Distributed

MIKA
SENGHAAS

EPFL

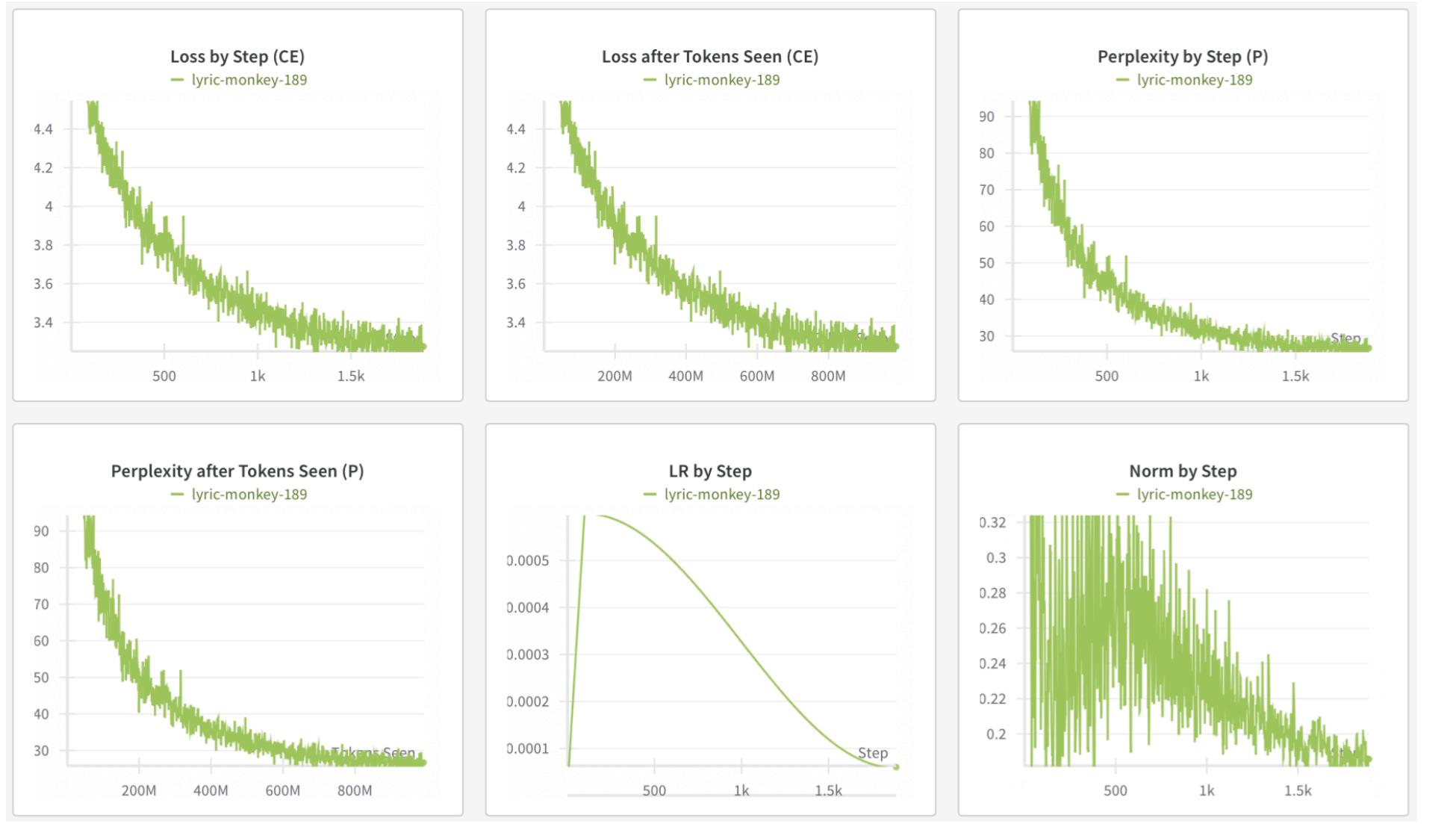
Next Steps

① Verify 1-GPU Baseline

NanoGPT setup
→ (GPT-2 (140m) on
Fine webEdv on
AOC)

② Implement traditional PP → Minimal (20xx)
PP Gist

③ Implement SWARM → Yandex SWARM +
Pi Zero Band

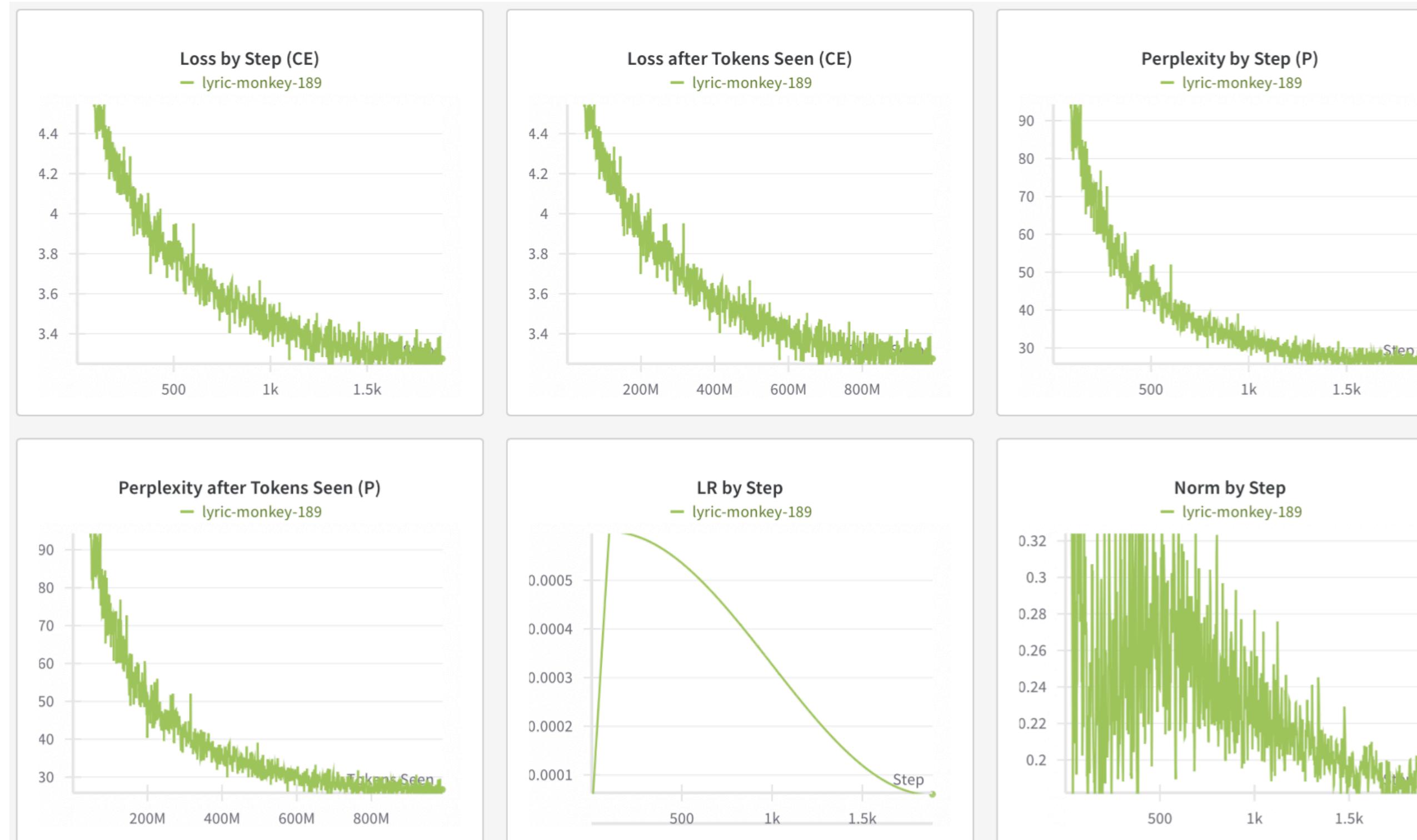


- GPT-2 1BT (Test Loss: ~3.1)

- AFAB Pipeline Parallel Implementation

Commits on Nov 8, 2024			
Implement model parallel sampling (WIP)	mikasenghaas committed 4 days ago	(Verified) 66fcfcf	🔗
Switch baseline model to use nn.Module and implement Single-GPU sampling	mikasenghaas committed 4 days ago	(Verified) 6a66beb	🔗
Implement eval loop	mikasenghaas committed 4 days ago	(Verified) 7383e18	🔗
Change loss to float in metrics	mikasenghaas committed 4 days ago	(Verified) fe49ff6b	🔗
Sharded model forward takes kwargs	mikasenghaas committed 4 days ago	(Verified) 73533a7	🔗
Implement learning rate scheduling for PP and clean-up	mikasenghaas committed 4 days ago	(Verified) 1da2f93	🔗
Refactor communication into own class	mikasenghaas committed 4 days ago	(Verified) 69f1c62	🔗
Fix repetitive micro batch sampling	mikasenghaas committed 4 days ago	(Verified) d6a4fe0	🔗
Adapt single GPU training to new interface	mikasenghaas committed 4 days ago	(Verified) cc5ddb8	🔗
Add distributed training loop (WIP)	mikasenghaas committed 4 days ago	(Verified) 5115583	🔗
Add sharded model	mikasenghaas committed 4 days ago	(Verified) ea1be53	🔗
Implement loading world information for local distributed	mikasenghaas committed 4 days ago	(Verified) 73450bb	🔗

Train GPT2 124M on 1B tokens from FinewebEdu (Single GPU)



- In-/post-training evaluation
- Sampling
- Checkpointing (Local/ W&B)
- LR Scheduling
- Mixed Precision Training

Test CE Loss ~3.1

Implement AFAB Pipeline Parallel



- Sharded model with custom forward/backward pass
- *torch.distributed* for comm of activations and gradients
- In-/post-training evaluation
- *Sampling*
- *Checkpointing (Local/ W&B)*
- *Mixed Precision Training*

Minimal implementation - loads of reusable components

```
(base) root@6c57594e146a:/workspace/swarm# find . -name "*.py" | xargs wc -l
    0 ./src/__init__.py
  260 ./src/train/baseline.py
  300 ./src/train/pipeline.py
    48 ./src/comm.py
   96 ./src/config.py
  117 ./src/logger.py
  192 ./src/metrics.py
   76 ./src/model.py
  189 ./src/utils.py
   29 ./src/world.py
1307 total
```

Next Steps

① ~~Verify 1 GPU Gist~~

NanoGPT setup
→ (GPT-2 (140m) on
Fine webEdv on
AOC)

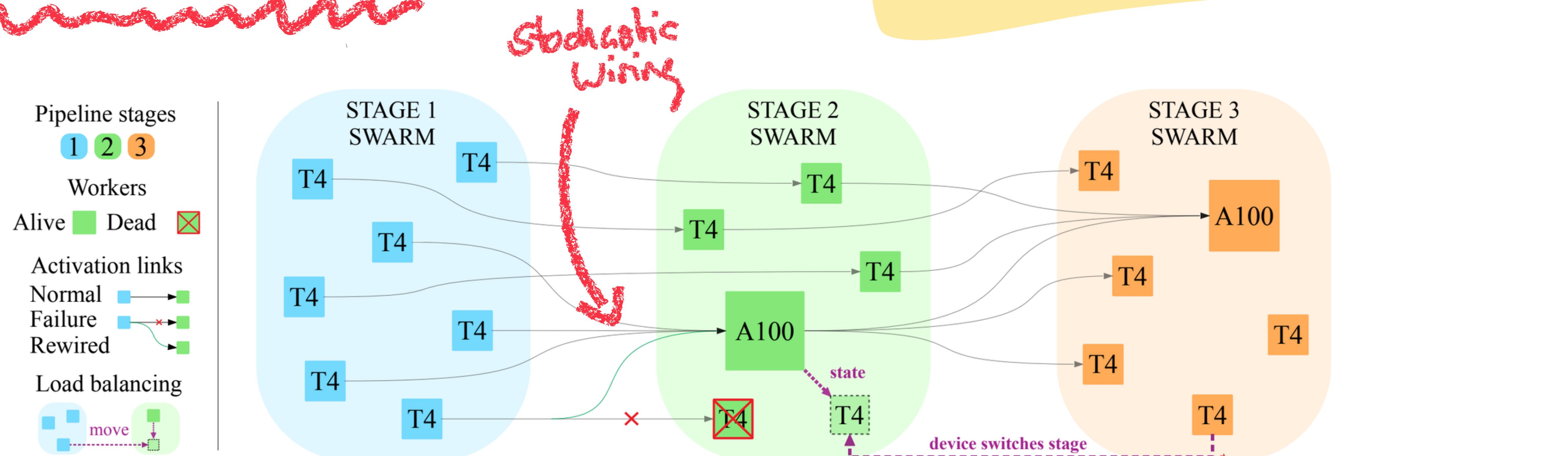
② Implement traditional PP → Minimal (20xx)
PP Gist

③ Implement SWARM → Yandex SWARM +
Pi Zero Band

- Allow for truly distributed workers (IPv4+port, instead of local cuda worker -> might have to switch communication backend)
- Implement missing functionality (remote logging, sampling, MP)
- Benchmark speed and convergence

References

SWARM



DYNAMIC + REDUNDANT PP