



CONFERENCE

—NORTH AMERICA 2026—

TitanRL Demo - Live Post-Training

Angel Li, Meta

Introduction to TitanRL

- TorchTitan's library for reinforcement learning (RL) post-training of LLMs
- Supports async and on-policy RL
- Unified model definition between trainer and generator
 - Batch invariance
- Easy integration with TorchTitan pretraining

Agenda

- Post train a model on how to play Connect 4
 - Full parameter Qwen 1.7B, 2 GPUs, GRPO
- 4 key components
 - Environment
 - Rubric
 - Data
 - Config
- Observability
 - Rollouts, W&B, Gantt chart, terminal
- How to extend to frontier use cases

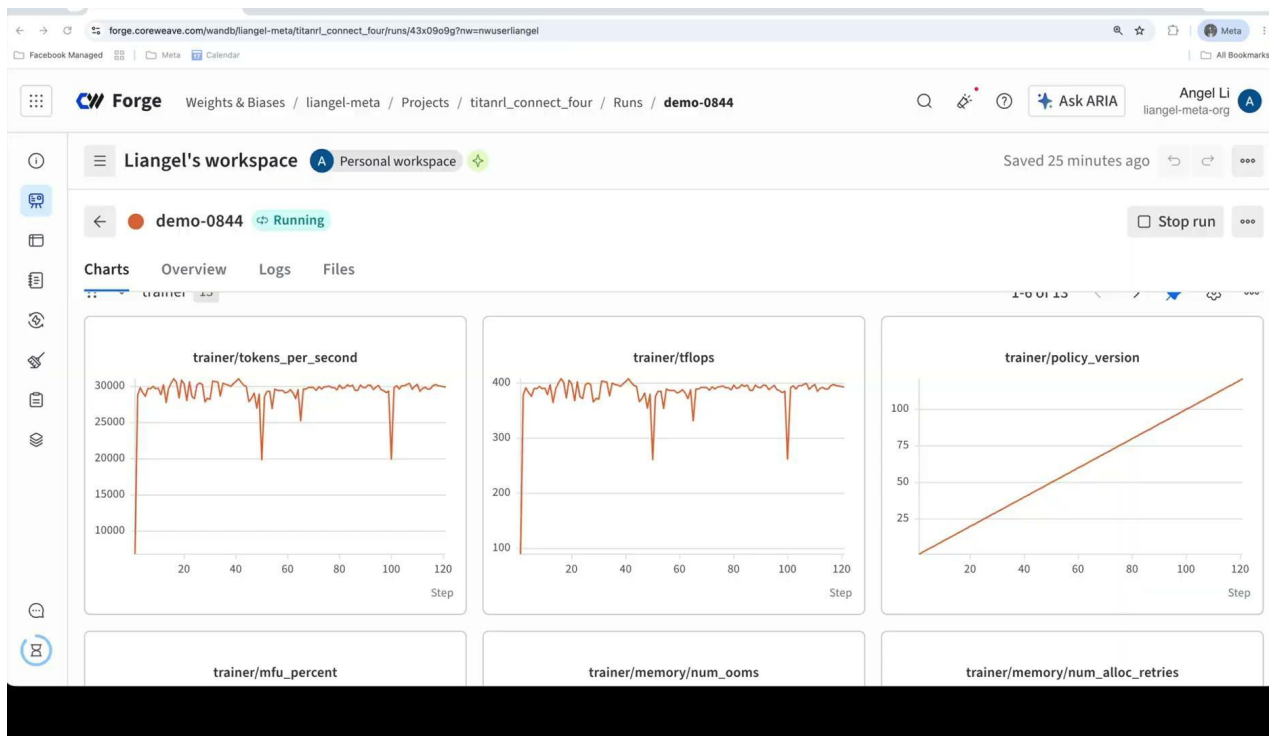


Thank you!

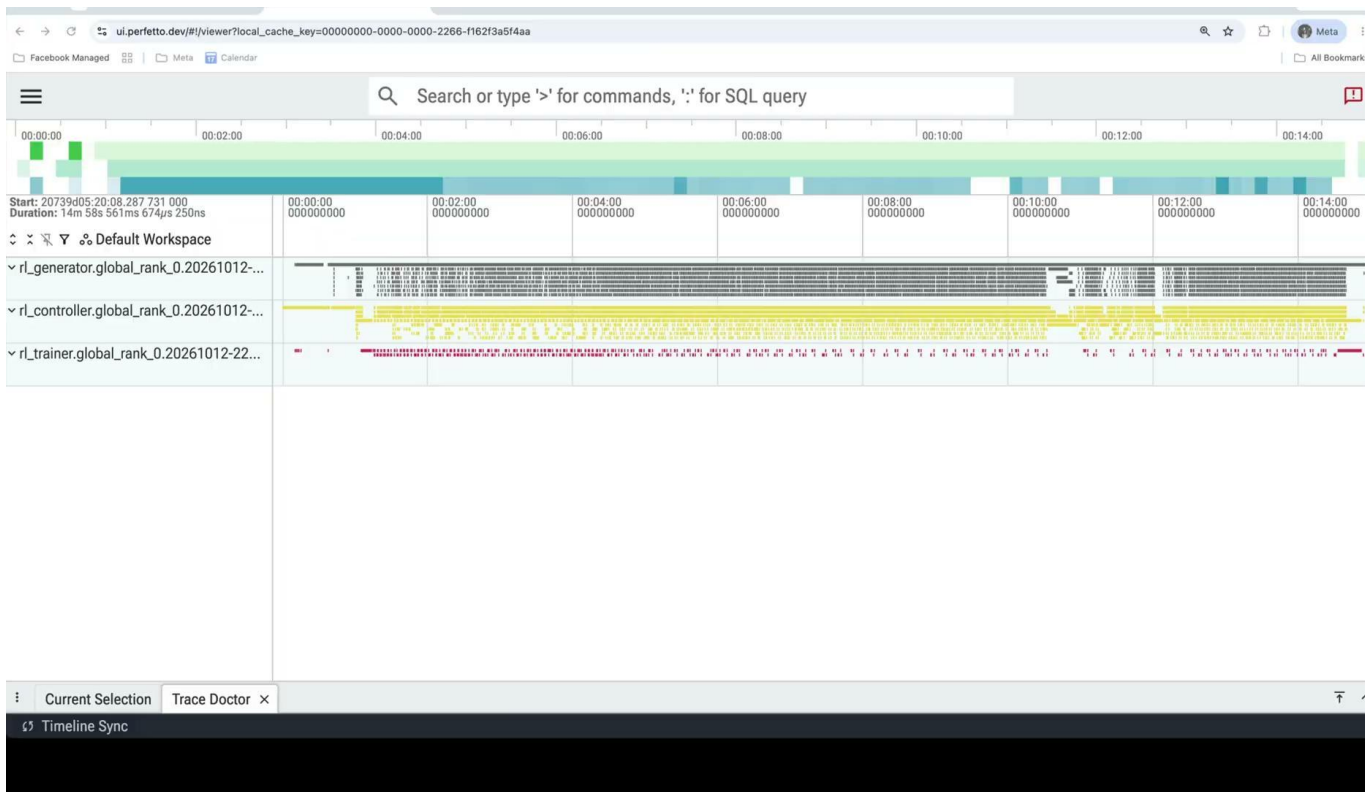
Launching RL train loop

```
devgpu005.sn3 ~local/torchtitan — x2ssh • ssh -L 8000:localhost:8000 devgpu005.sn3.facebook.com — Missing FA3 link in cu134 index — x2ssh • ssh -L 8000:localhost:8000 devgpu005.sn3.facebook.com — ever ping bot for pytorch/pytorch — x2ssh • ssh -L 8000:localhost:8000 devgpu005.sn3.facebook.com +  
(pytorch) [liangel@devgpu005.sn3 ~/local/torchtitan (titanrl-connect4-demo)]$ WANDB_PROJECT=titanrl_connect_four WAN  
DB_RUN_NAME=demo-$(date +%H%M) /home/liangel/.conda/envs/pytorch/bin/python -m torchtitan.rl.train --module torc  
htitan_recipes.rl.connect_four --config rl_grpo_qwen3_1_7b_connect_four_demo
```

Observability: W&B



Observability: Gantt Trace



Observability: Rollout Generator

```
"advantage": 0.0,  
"turns": [  
  {  
    "turn_id": 0,  
    "min_policy_version": 104,  
    "max_policy_version": 104,  
    "prompt_messages": [  
      {  
        "role": "user",  
        "content": "Let's play Connect 4. You are X and I am O. The board has 7 columns and 6 rows. On your turn, drop one piece into a column (1-7) that is not full; it falls to the lowest empty cell. The first player with 4 pieces in a row (horizontally, vertically, or diagonally) wins. In the board below, the top line is the top of the board, the bottom line is the bottom, columns are numbered 1-7, and . is an empty cell.\n\n1 2 3 4 5 6 7\n.\n.\n.\n.\n.\n.\nColumn 1 (bottom to top): empty\nColumn 2 (bottom to top): empty\nColumn 3 (bottom to top): empty\nColumn 4 (bottom to top): empty\nColumn 5 (bottom to top): empty\nColumn 6 (bottom to top): empty\nColumn 7 (bottom to top): empty\nPlayable columns: 1, 2, 3, 4, 5, 6, 7\nYour move (you are X). Think it through in at most 3 short sentences: which column, if any, gives you 4 in a row now; which column, if any, would give me 4 in a row on my next turn; and which column is best. Then give your column once, at the very end, in exactly this format: <move>N</move>"  
      },  
      {  
        "completion_message": {  
          "role": "assistant",  
          "content": "Looking at the board, I don't have 4 in a row yet. If I play in column 4, it would give me a good chance to complete a row. The best move is column 4. \n\n<move>4</move>"  
        },  
        "env_messages": [  
          {  
            "role": "user",  
            "content": "I played column 3. That's the end of this segment.\n\n1 2 3 4 5 6 7\n.\n.\n.\n.\n.\n.\nO X .\n1 2 3 4 5 6 7"  
          }  
        ]  
      }  
    ]  
  ]  
]
```

Before vs After

TitanRL Connect 4

BEFORE RL

game 3 vs human

model turn

status your turn: type a column 1-7 + Enter

move =

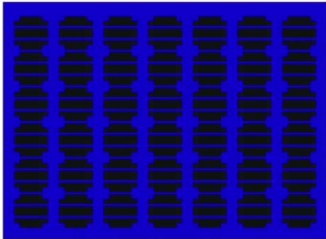
latency -

solver -

best solver would play 1 or 2 or 3 or 4 or 5 or

model's answer

Model • YELLOW (O) VS human • RED (X)



MOVES:

Scoreboard

WIN	DRAW	LOSS	ILLEGAL
0	0	2	0
games 2 score 0.00			
model moves 8 best 50% blunders 75%			

vLLM startup 53s | greedy | prompt think | max_tokens 384 | new conversation every 1 model moves | Qwen3-1.7B

TitanRL Connect 4

AFTER RL (v5b)

game 3 vs human

model turn

status your turn: type a column 1-7 + Enter

move =

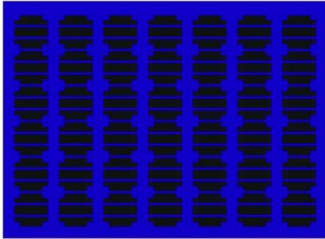
latency -

solver -

best solver would play 2

model's answer

Model • YELLOW (O) VS human • RED (X)



MOVES:

Scoreboard

WIN	DRAW	LOSS	ILLEGAL
0	0	2	0
games 2 score 0.00			
model moves 13 best 54% blunders 23%			

vLLM startup 51s | greedy | prompt think | max_tokens 384 | new conversation every 1 model moves | v5b_step250_hf

Batch Invariance

```
(pytorch) [liangel@devgpu005.snb3 ~/local/torchtitan (titanrl-connect4-demo)]$ CUDA_VISIBLE_DEVICES=4,5 WANDB_PROJECT=titanrl_connect_four WANDB_RUN_NAME=demo-bi-$(date +%H%M) \
/home/liangel/.conda/envs/pytorch/bin/python -m torchtitan.rl.train \
--module torchtitan_recipes.rl.connect_four \
--config rl_grpo_qwen3_1_7b_connect_four_demo_batch_invariant
W1016 09:07:13.487000 855479 torch/_opaque_base.py:6] torch._opaque_base is deprecated, use torch._custom_class_base instead
W1016 09:07:13.488000 855479 torch/_library/opaque_object.py:294] register_opaque_type is deprecated, use register_custom_class instead
W1016 09:07:13.488000 855479 torch/_library/opaque_object.py:211] typ='value' is deprecated, use typ='constant' instead
Warning: setting HYPERACTOR_CODEC_MAX_FRAME_LENGTH since this needs to be set to enable large RPC calls via Monarch
[titan] 2026-10-16 09:07:19,396 - torchtitan.components.checkpointer.base - WARNING - initial_load_model_only=True has no effect without an initial_load_path.
[titan] 2026-10-16 09:07:19,397 - torchtitan.observability.structured_logger.jsonl_handler - INFO - Structured logging -> JSONL: /home/liangel/local/torchtitan/outputs/rl_connect_four_demo_batch_invariant/structured_logs/rl_controller.global_rank_0.20261016-090719-BKWBIZ.jsonl
wandb: [wandb.login()] Loaded credentials for https://api.wandb.ai from /home/liangel/.netrc.
wandb: Currently logged in as: liangel (liangel-meta) to https://api.wandb.ai. Use `wandb login --relogin` to force relogin
wandb: Tracking run with wandb version 0.28.2
wandb: Run data is saved locally in outputs/rl_connect_four_demo_batch_invariant/wandb/run-20261016_090720-hv6w5l7a
wandb: Run `wandb offline` to turn off syncing.
wandb: Syncing run demo-bi-0907
wandb: 🌟 View project at https://wandb.ai/liangel-meta/titanrl_connect_four
wandb: 🚀 View run at https://wandb.ai/liangel-meta/titanrl_connect_four/runs/hv6w5l7a
wandb: Detected [openai] in use.
wandb: Use W&B Weave for improved LLM call tracing. Install Weave with `pip install weave` then add `import weave` to the top of your script.
wandb: For more information, check out the docs at: https://weave-docs.wandb.ai
[titan] 2026-10-16 09:07:21,656 - torchtitan.observability.metrics - INFO - WandB logging enabled
```