

Reinforcement Learning of a Molecular Diffusion Model for Energy Storage and Conversion



Ben Berol

Rahul Khandelwal · Sofia Stoica · Xiaoyun Wang · Logan Ward

Advisor: Professor Eliu Huerta

Working with Argonne National Laboratory and UIUC NCSA as part of the FoDOMMaT REU (NSF Award #: 2447843) · Compute from NCSA Delta System

1. Background

Metal-organic frameworks (MOFs) are porous crystalline materials made from metal nodes connected by organic linkers. Their pore size and chemical properties can be tuned by changing these building blocks, making MOFs promising materials for energy storage and conversion. However, the number of possible linker designs is extremely large, and only some will form stable frameworks with useful adsorption properties.

GenMol (Lee et al., 2025) is a masked-diffusion model originally developed to generate drug-like molecules. It represents each molecule as a Simplified Molecular Input Line Entry System (SMILES) string, a text sequence that encodes its atoms, bonds, and connectivity. In this project, I adapt GenMol to propose new MOF linkers and fine-tune it with reinforcement learning. Generated SMILES strings are checked for chemical validity, converted into candidate linkers, assembled into MOFs, and evaluated using structural and gas-uptake simulations. These results provide feedback that guides the model toward linkers more likely to form stable, high-performing sorbents.

2. Objectives

- Increase the fraction of generated linkers that form valid MOFs.
- Improve the structural stability and gas-uptake performance of assembled MOFs.
- Expand the diversity of promising linker designs explored by the model.
- Support sustained learning without sacrificing molecular validity or MOF assembly.

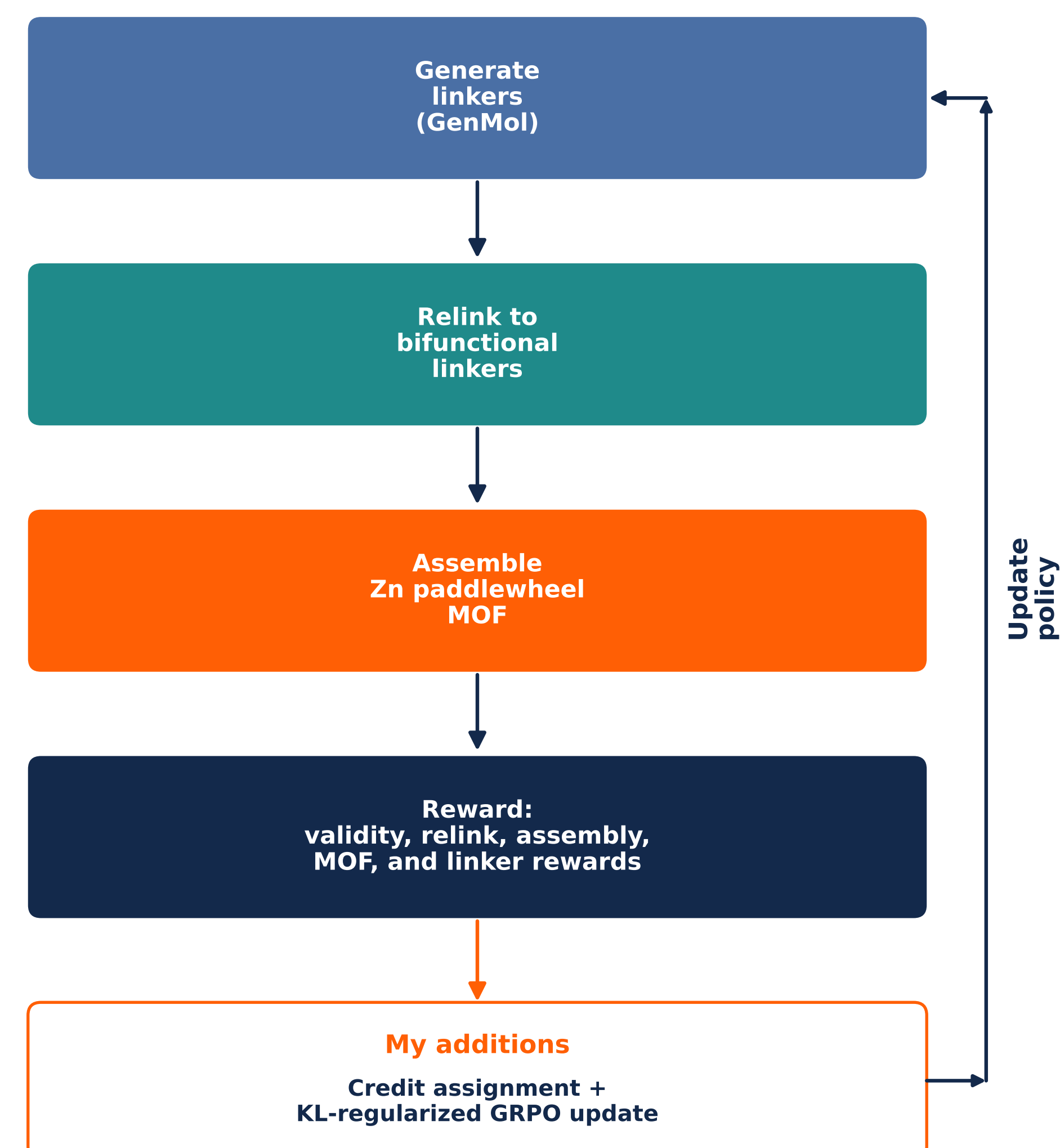
3. Reward structure

Each molecule earns a weighted sum of components. Cheap checks run every step; MOF builds run on periodic validation steps:

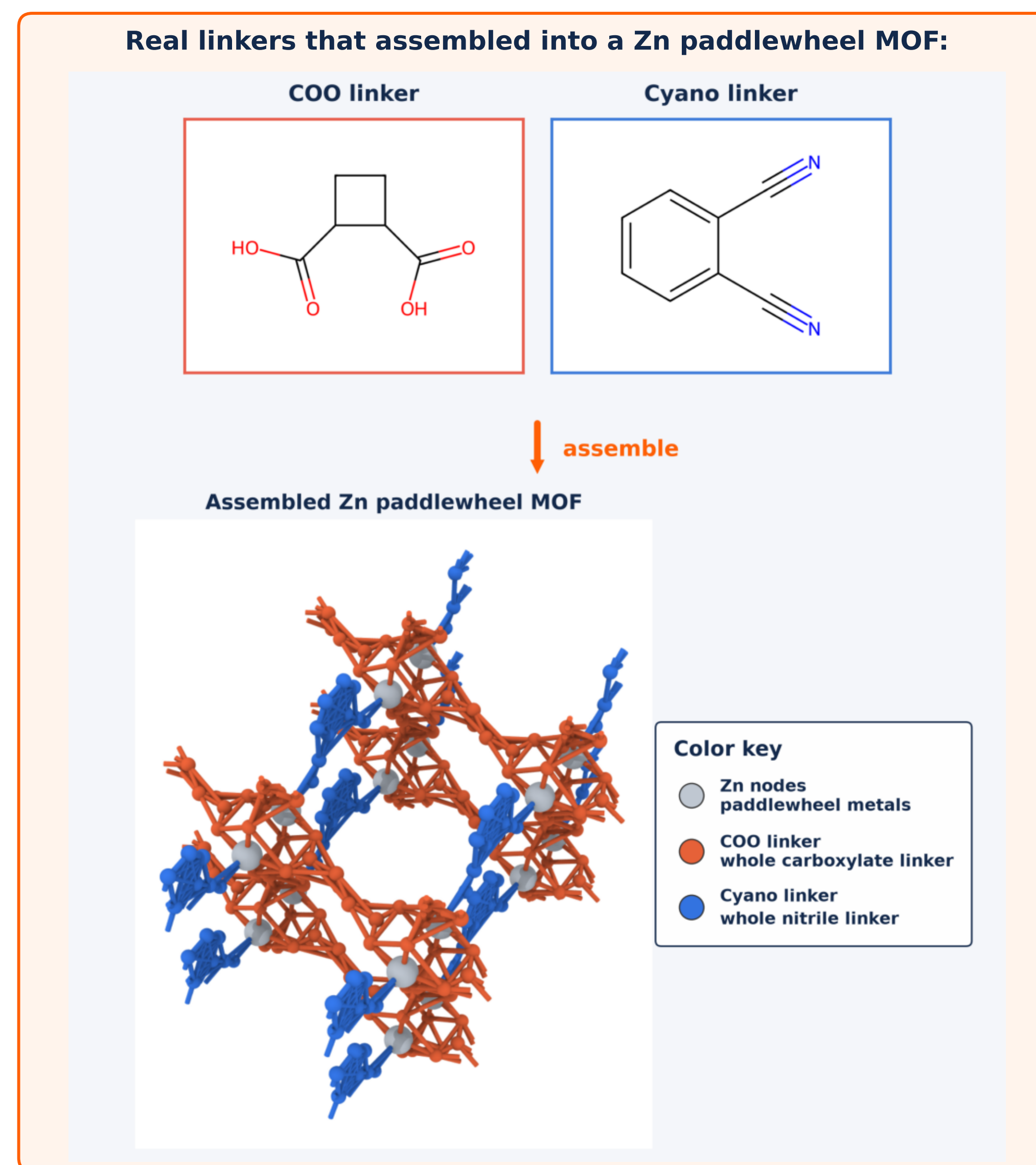
- **Validity:** rewards parseable SMILES, every step
- **Relink:** rewards molecules that yield a linker, validation steps
- **Assembly:** rewards molecules that appear in a built MOF, validation steps
- **MOF rewards:** MD strain, RASPA adsorption, validation steps
- **Linker rewards:** Tanimoto similarity, rotatable bonds, every step

4. System overview

GenMol trains in a closed reinforcement-learning loop: generate linkers, relink them, assemble a MOF, score it, assign credit, and update the policy under a KL constraint.



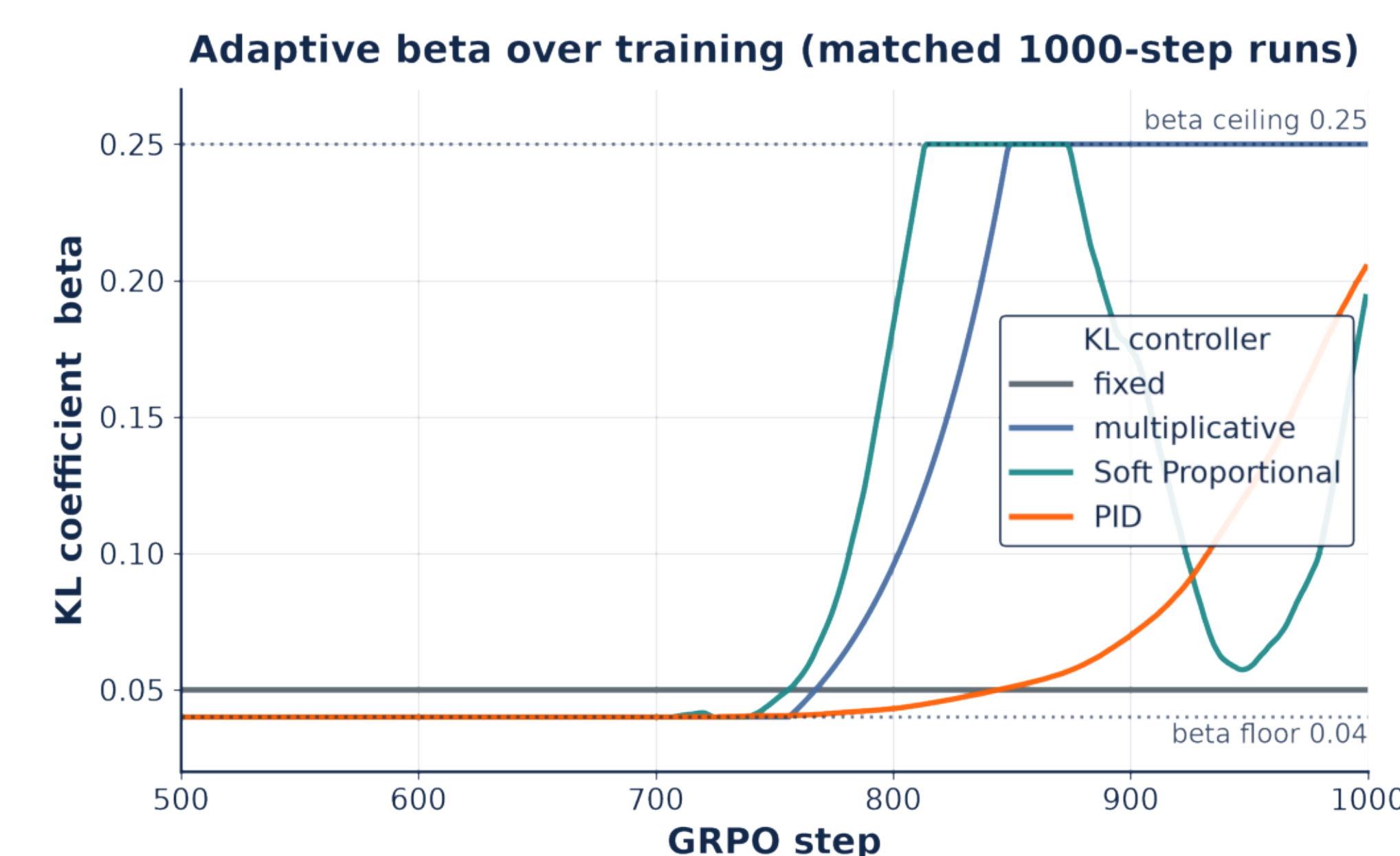
Rewards are assigned based on step. Computationally cheap rewards that don't require assembly run every step. Expensive rewards are only applied on validation steps (1 in 10 steps).



5. Contribution 1: Adaptive KL

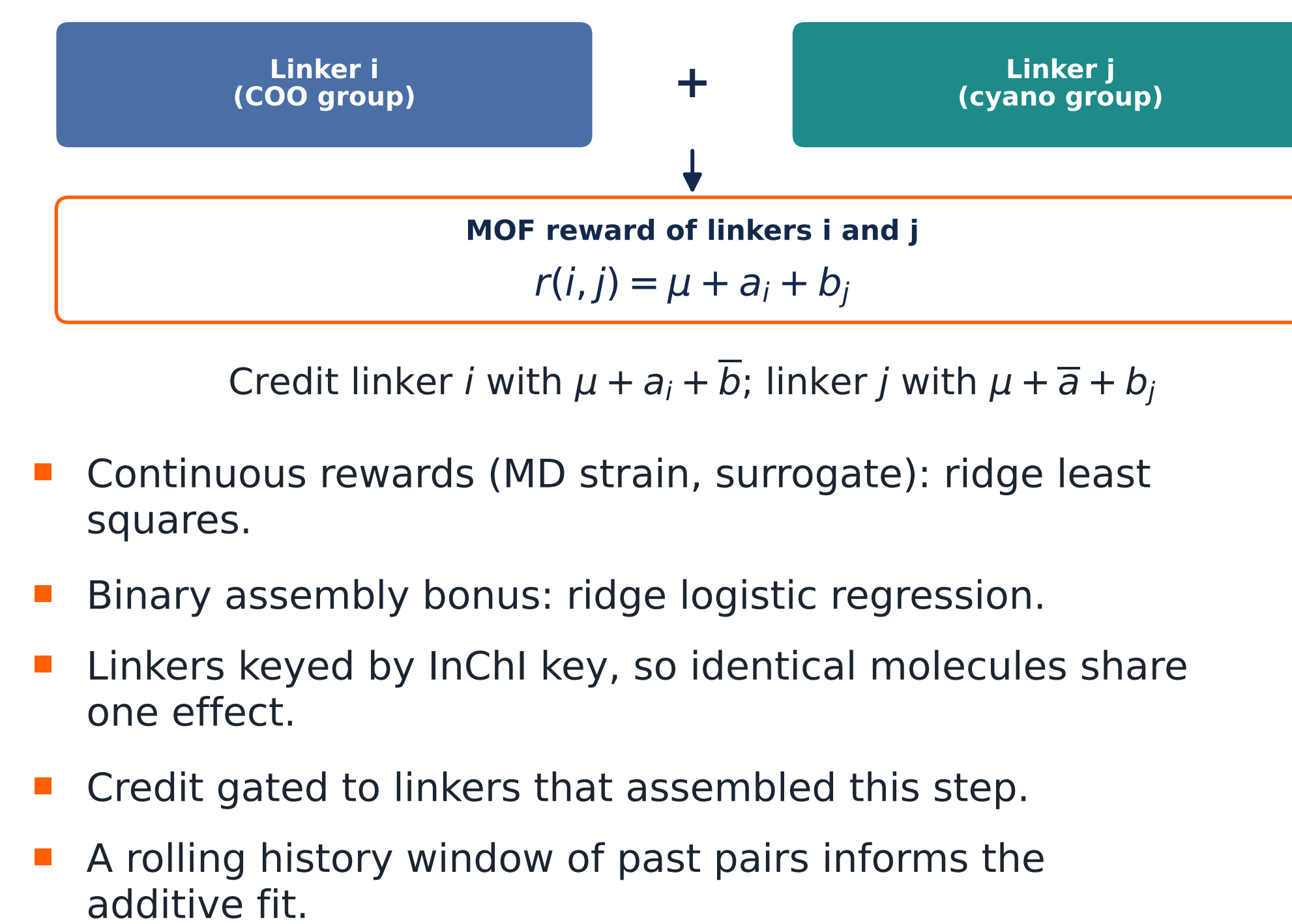
GRPO (Shao et al., 2024) fine-tunes the policy with a reward signal but also penalizes how far the updated model drifts from a frozen reference copy, measured by Kullback-Leibler (KL) divergence. That penalty, scaled by a coefficient beta, keeps generation from collapsing into invalid or low-diversity linkers. A fixed beta either over-constrains learning or lets KL spike, so I built a controller that adapts beta mid-run from a rolling-mean KL. I tested four strategies:

- **Fixed:** constant beta (baseline)
- **Multiplicative:** multiplies beta by a constant factor to adjust
- **Soft Proportional:** a multiplicative controller whose update is scaled by the current proportional error, introduced in Ziegler et al. (2019).
- **PID:** Rate-limited PID based on KL error



6. Contribution 2: Credit assignment

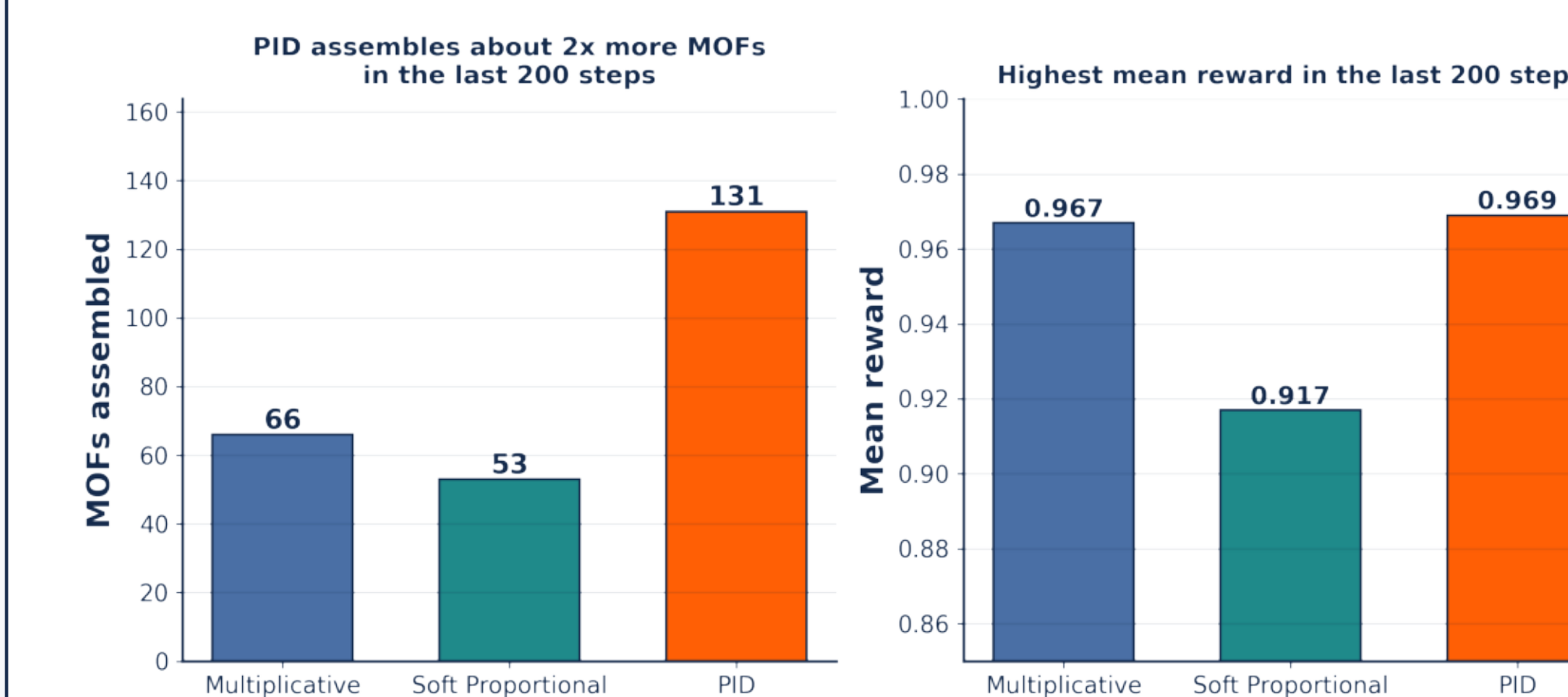
We are tuning two molecular building blocks of a MOF (cyano and carboxylate linkers) that both contribute to the fitness of the finished product. To allocate credit from MOF rewards I fit a two-way additive model over pooled pairs, crediting each linker with its own marginal effect versus an average partner.



7. Results

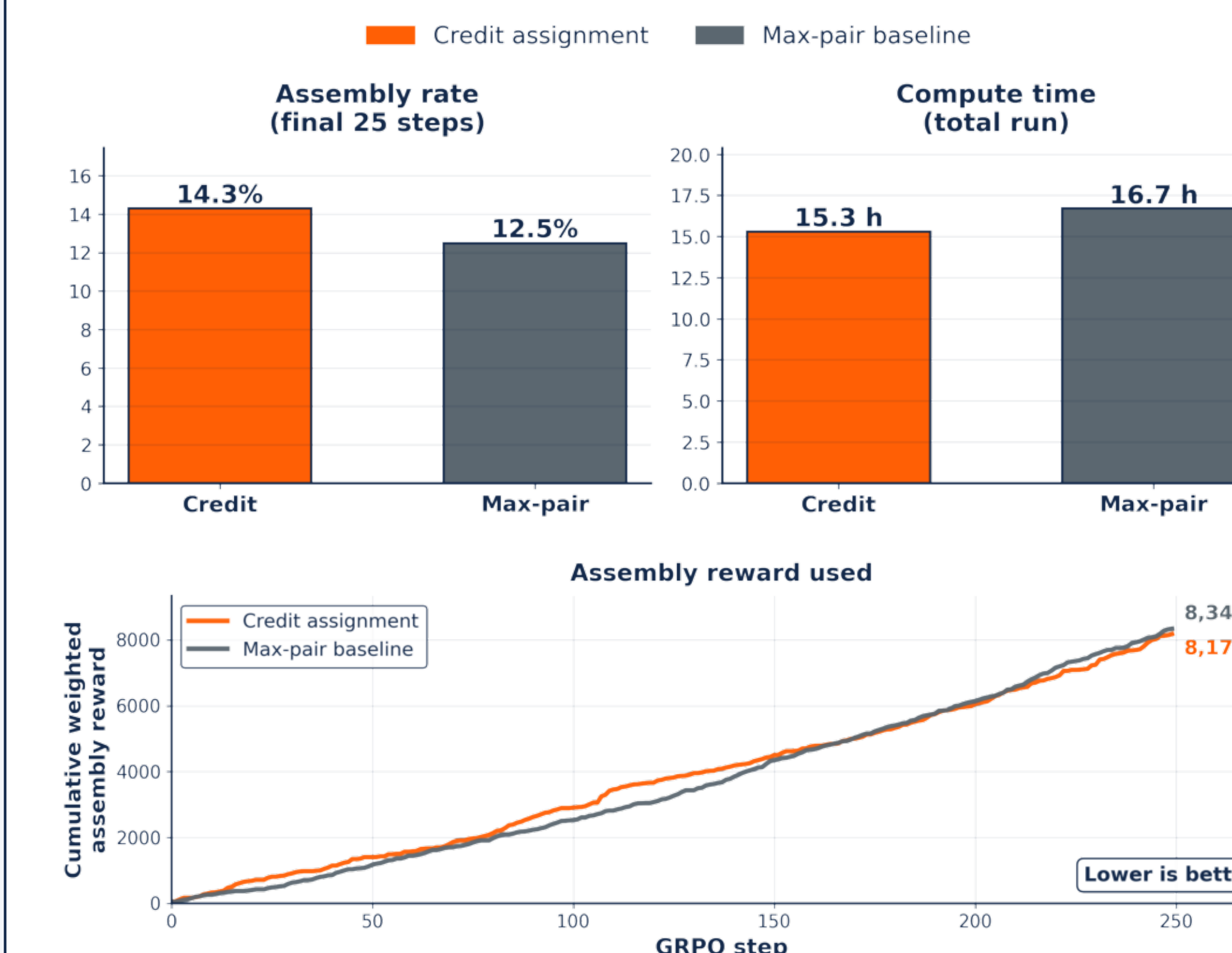
Adaptive KL controller (Contribution 1)

A rate-limited PID controller assembles 131 MOFs, about twice the multiplicative (66) and soft proportional (53) controllers, at the highest mean reward (0.969). These results suggest that PID may have smoother beta control that improves learning.



Credit assignment (Contribution 2)

Over a matched 250-step run, credit assignment reaches a higher final-phase assembly rate (14.3% vs 12.5%) and finishes faster, all while spending slightly less assembly reward. This shows that credit assignment amplifies the effect of MOF rewards by distributing a stronger, more informative learning signal across linkers.



8. Conclusions and future work

A rate-limited PID KL controller assembles MOFs most effectively while holding overall reward, and linker-level credit assignment raises MOF assembly rates over the max-pair baseline. Next steps:

- Combine adaptive KL with credit assignment to compound both gains.
- Integrate NeMo RL library to scale GRPO training across multiple GPUs and nodes.
- Add an LLM assembly predictor for denser reward.



Scan for references & further reading

tinyurl.com/2d5wm3am