Amortized Molecular Optimization via GRPO

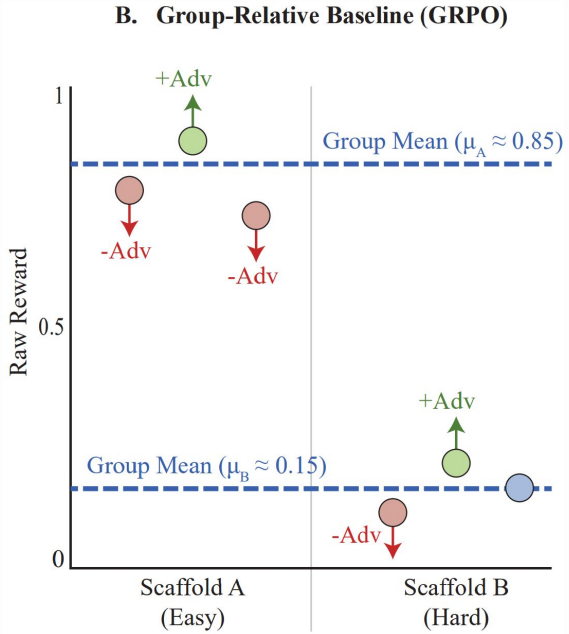
Joint work by: Muhammad bin Javaid, Hasham Hussain, Ashima Khanna, Berke Kişin, Jonathan Pirnay, Alexander Mitsos, Dominik G. Grimm, Martin Grohe

Why Amortized Molecular Optimization?

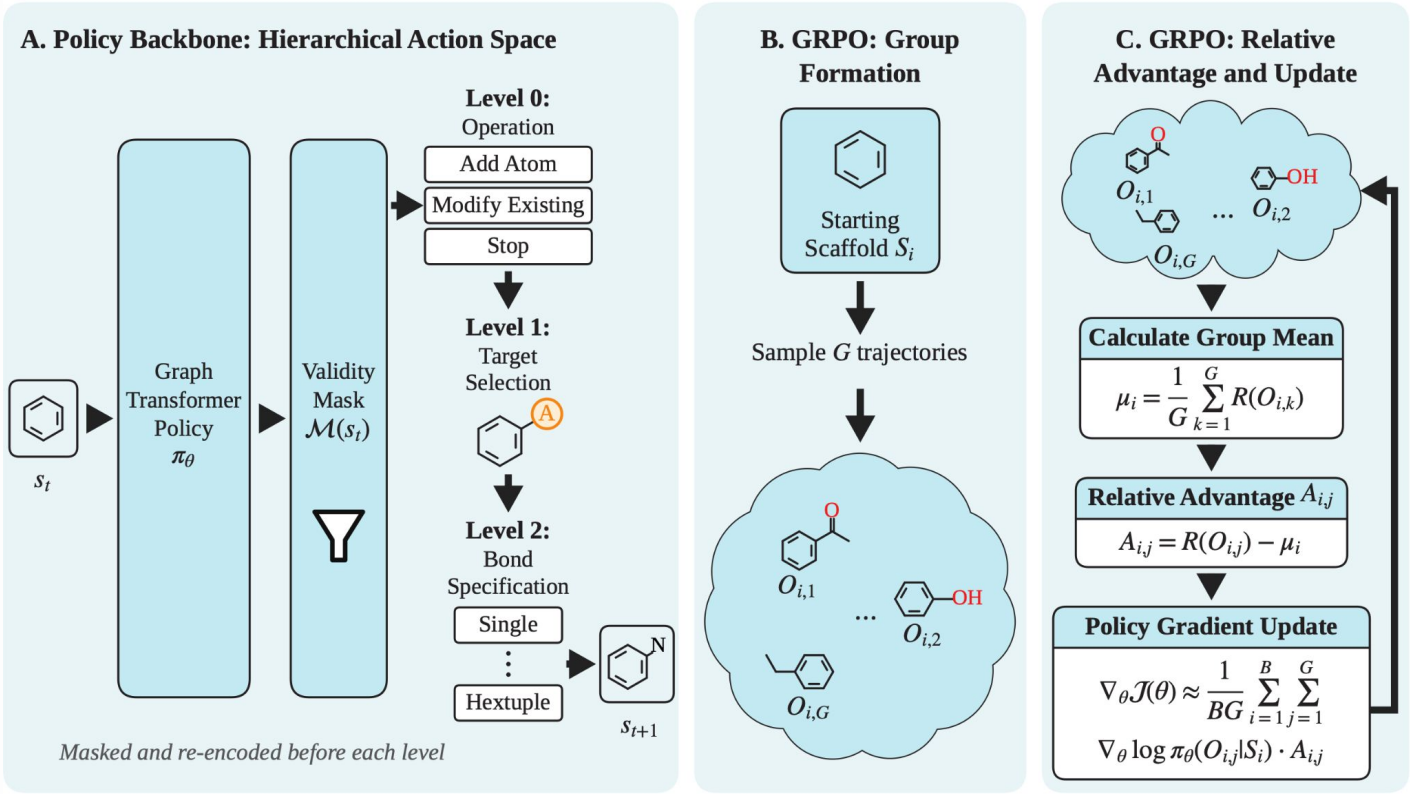
- **Molecular design:** propose candidates, score them with a costly oracle, test the best in the lab.
- **State of the art is instance optimization:** every new input molecule triggers a fresh search and hundreds of oracle calls.
- **Our goal:** one reusable model that improves any input molecule, paying the search cost once during training.
- **Why it matters:** inference needs no oracle calls, enabling high-throughput library screening and interactive design tools.

Key Idea: Group-Relative Rewards

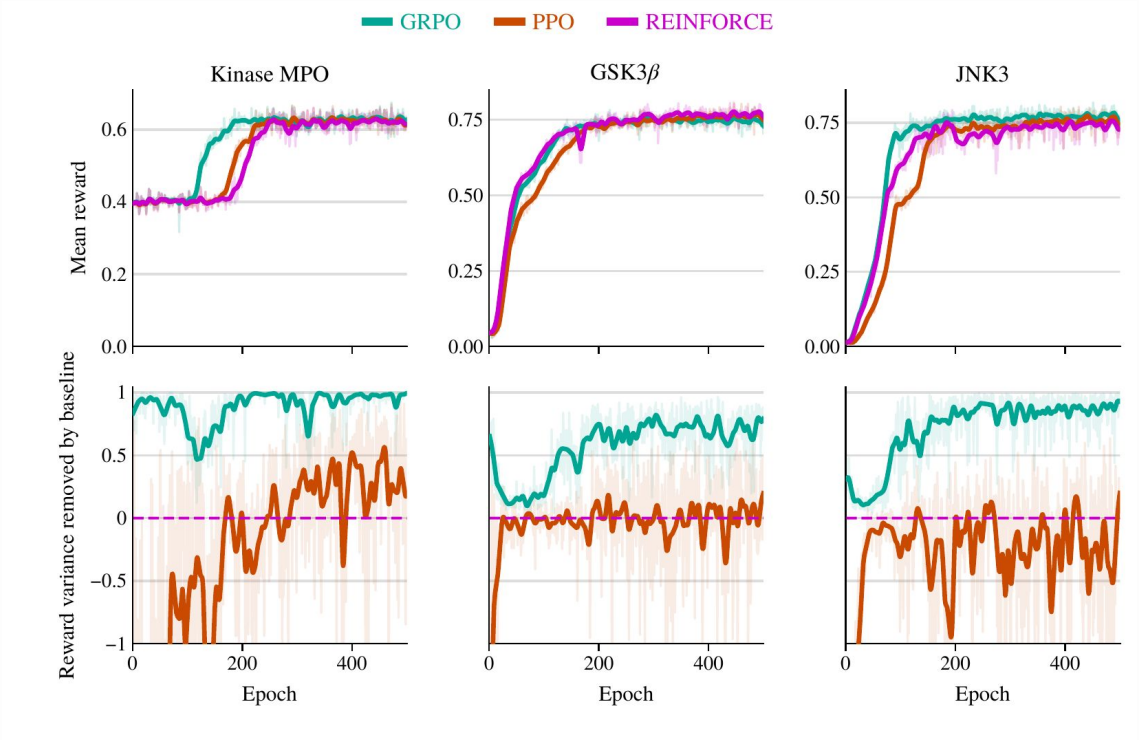
- Input molecules differ widely in how much reward is achievable.
- Sample a group of completions per input; **center rewards on the group mean.**
- Rewards the **relatively best** attempts on each molecule, so the model learns from easy and hard inputs alike.



Our Approach: AMORTIX



GRPO vs. PPO vs. REINFORCE



- **Top:** all three reach similar training reward, but GRPO gets there first on Kinase MPO and JNK3.
- **Bottom:** GRPO's group baseline removes the reward variance caused by differences between input molecules; PPO's critic removes little and often adds variance.

Results: Scaffold Decoration

Method	GSK3β	JNK3	Kinase MPO
Amortized Policies			
LibINVENT	0.414 ± 0.122	0.261 ± 0.065	0.453 ± 0.007
GraphXForm	0.110 ± 0.001	0.051 ± 0.002	0.409 ± 0.003
DrugEx v3	0.194 ± 0.004	0.356 ± 0.003	0.354 ± 0.003
AMORTIX-DeNovo	0.572 ± 0.015	0.381 ± 0.029	0.397 ± 0.005
AMORTIX-REINFORCE	0.757 ± 0.084	0.754 ± 0.092	0.475 ± 0.120
AMORTIX-PPO	0.725 ± 0.017	0.728 ± 0.008	0.552 ± 0.015
AMORTIX	$\mathbf{0.775 \pm 0.005}$	$\mathbf{0.796 \pm 0.004}$	$\mathbf{0.619 \pm 0.004}$
Instance Optimizers			
Mol GA	0.735 ± 0.014	0.659 ± 0.011	0.461 ± 0.013
GenMol	0.700 ± 0.012	0.470 ± 0.015	0.479 ± 0.012

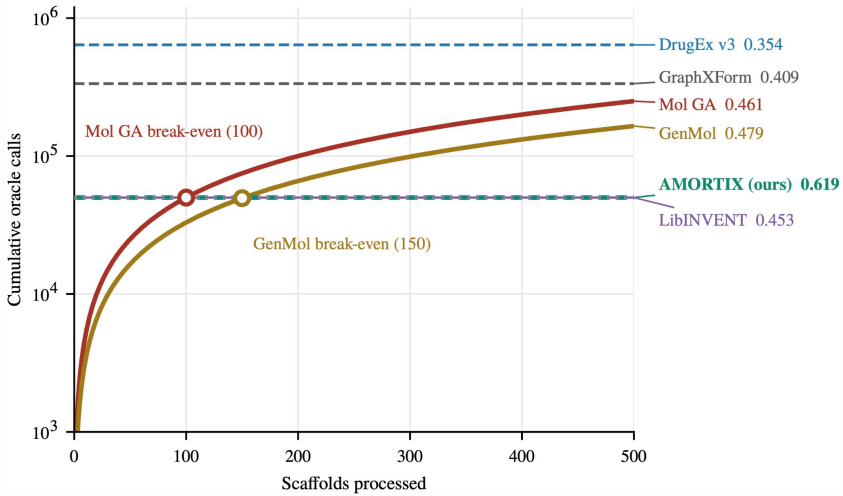
Across expanded suite of 23 objectives:

Method	Mean score	Best on	Split SD	GRPO higher on
GRPO	0.570	18/23	0.004	—
PPO	0.527	5/23	0.013	18/23
REINFORCE	0.512	0/23	0.072	22/23
LibINVENT	0.320	0/23	0.021	23/23

- **Best on all three kinase tasks,** ahead of the instance optimizers (Mol GA, GenMol).
- **Zero oracle calls** at test time.
- **Across 23 objectives, GRPO wins on 18** and varies least across splits.

Cost vs. Performance (Kinase MPO)

- AMORTIX: a budget of 50k training oracle calls, zero at test time.
- Breaks even with instance optimizers after ~100 scaffolds (Mol GA) and ~150 (GenMol), with a higher score.



Takeaways

- Group-relative rewards handle the wide spread in input difficulty and outperform PPO and REINFORCE on held-out scaffolds.
- Search cost shifts from inference to training: one model, no test-time oracle calls, scalable to library screening.
- **Limitations:** cheap proxy oracles only (no docking or 3D scoring); training is oracle-intensive, so amortization pays off over many scaffolds.
- **Next:** atom/bond deletion and substitution; pair the amortized policy with an instance optimizer for refinement.

Thank You!