Extended Abstract

Motivation Most molecular reinforcement learning methods maximize a scalar reward or search across a Pareto frontier. In practical design settings, however, a chemist often wants molecules that match a specific target profile rather than simply score well under a weighted objective. We therefore study molecular generation as a goal-conditioned RL problem, surfacing three distinct challenges that are absent from scalar-optimization settings: natural reward sparsity (properties are only defined on complete molecules), a variable state-dependent valid-action set governed by BRICS chemical compatibility rules, and a generalization requirement to target vectors held out from training.

Method We model molecular design as a fragment-assembly Markov decision process in which the state consists of a partial molecule and a target property vector. The agent sequentially attaches BRICS-compatible fragments or terminates. The objective is to minimize Euclidean distance between the final molecule's normalized properties and the target vector. The main task uses `sLogP`, `QED`, and `TPSA`, with training and evaluation goals kept disjoint. We additionally study one-, five-, and seven-property variants, and fragment-library sizes from 50 to 6400, to characterize scaling behavior.

Architectures We compare actor-critic policies over Morgan fingerprints and graph neural networks. Beyond these baselines, we study exact site-aware action parameterizations, which expose precise attachment-site information but enlarge the action space; fragment-site embeddings, which cache GNN representations of fragment dummy atoms; and Counterfactual Action Scoring (CAS), where the policy scores features of the predicted next partial molecule rather than only the candidate fragment identity. We also train a successor-style graph model that predicts goal-space outcomes directly, scoring actions by negative distance between the predicted property vector and the target: $Q(s, a; \mathbf{g}) = -\|\psi(s, a; \mathbf{g}) - \mathbf{g}\|_2$.

Training Because chemically meaningful properties are defined only on complete molecules, the environment is naturally sparse-reward. We compare sparse terminal reward against distance-delta, potential-based, and property-surrogate shaping, all combined with Hindsight Experience Replay (HER). We also ablate reward scale, HER relabeling frequency k, value normalization, and PPO-style updates to isolate optimization stability from generalization. To make large-scale counterfactual evaluation practical, we use a calibrated additive property surrogate for environment-side property estimation during CAS.

Results Learned policies consistently outperform random valid-action assembly (6.7% success, 0.416 mean distance) on held-out targets. A2C with fingerprints is the strongest standard baseline at 11.9% success and 0.277 distance; sparse terminal reward outperforms all denser shaping variants. Exact site-aware factorization alone hurts performance (4.7%, 0.386), while CAS recovers the loss (9.4%, 0.299), confirming that richer action spaces require richer action evaluation. The strongest overall model is the successor-style site-aware graph policy at 34.2% success and 0.175 distance, demonstrating that representing action outcomes directly in goal space is substantially more effective than standard policy-gradient optimization.

Scaling experiments show that joint success drops rapidly with target dimensionality while per-property error remains stable, and performance peaks near 200 fragments before degrading as the library grows, confirming that exploration and branching factor are the central bottlenecks.

Conclusion Targeted molecular design is a compelling goal-conditioned RL problem in which representation and action evaluation matter more than reward density alone. Sparse rewards with HER are sufficient for nontrivial goal matching; the decisive factors are how action consequences are evaluated and how outcomes are represented in goal space. The largest gains come from CAS and successor-style value representation rather than from increasingly dense shaping terms, and the central RL bottleneck is generalization under combinatorial branching rather than reward sparsity per se.

Fragment Assembly as Goal-Reaching: An RL Approach to Targeted Molecular Design

Katie Liu
Department of Computer Science
Stanford University
katiel25@stanford.edu

Megan Santhumayor
Department of Computer Science
Stanford University
msanth@stanford.edu

Asmani Yamin
Department of Computer Science
Stanford University
ayamin@stanford.edu

Abstract

We study targeted molecular design as a goal-conditioned reinforcement learning problem in which an agent assembles molecules from BRICS fragments to match a specified target property vector. Unlike scalar-reward molecular optimization, this setting requires precise control over a point in property space, a variable valid-action set determined by chemical compatibility, and generalization to held-out goals. We compare actor-critic policies over fingerprint and graph encoders, exact site-aware action parameterizations, fragment-site embeddings, Counterfactual Action Scoring (CAS), and a successor-style graph model that predicts goal-space outcomes directly. Across experiments, learned policies outperform random assembly on held-out targets, but the strongest gains come from better action evaluation and value representation rather than denser shaping rewards. Sparse terminal reward remains the most reliable training objective among standard policy-gradient methods. CAS substantially improves over naive site-aware policies by letting the policy score the predicted next partial molecule, and a successor-style site-aware graph model achieves the strongest overall performance, reaching 34.2% held-out success and mean distance 0.175 on the main three-property task. Scaling experiments further show that difficulty grows sharply with both the number of target properties and the size of the fragment library, indicating that generalization and exploration are the central RL bottlenecks.

1 Introduction

Designing molecules with desired physicochemical properties is a central problem in drug discovery. Since chemical space is combinatorially large, exhaustive search is infeasible, making sequential decision-making methods attractive. Reinforcement learning has therefore become a natural framework for molecular generation, where an agent incrementally constructs molecules while optimizing a target objective.

Most molecular RL methods formulate the task as scalar optimization: maximize a weighted combination of properties, satisfy reward thresholds, or trade off multiple objectives along a Pareto frontier. These formulations are useful when the goal is simply to find high-scoring molecules. They are less appropriate when the design target is a specific property profile. In that setting, the problem is not reward maximization but goal reaching.

We formulate molecular generation as a goal-conditioned RL problem in which an agent assembles a molecule from BRICS fragments to match a target property vector. This formulation introduces three distinct RL challenges. First, reward is naturally sparse because chemically meaningful properties are only defined on complete molecules. Second, the action space is variable and state-dependent: the policy must choose among only the fragment attachments that are chemically valid for the current partial molecule. Third, the learned policy must generalize to target vectors that were not observed during training.

Within this setting, we study which RL design choices matter most. We compare sparse and shaped rewards, fingerprint and graph encoders, A2C and PPO, exact site-aware action parameterizations, Counterfactual Action Scoring (CAS), and a successor-style graph model that predicts goal-space outcomes directly. The resulting picture is clear: sparse terminal reward remains the strongest standard objective, richer action spaces require richer action evaluation, and the largest performance gains come from goal-conditioned outcome representation rather than denser shaping alone.

2 Related Work

Existing molecular RL methods typically optimize scalar objectives or explore multi-objective trade-offs. CPRL Wang and Zhu (2024) constructs a Pareto-style ranking through clustering, while HybridMolGen Amiri and Nasirinia (2026) combines Pareto optimization with curriculum learning and scalarization. These methods are effective for discovering good molecules under broad desirability criteria, but they do not directly solve the problem of matching a specific target vector.

Other approaches improve exploration rather than target matching. Mol-AIR Park et al. (2025), for example, combines policy gradients with intrinsic bonuses based on count-like signals and random network distillation. Such methods encourage broad exploration of chemical space, but the exploration signal is not tied to a specified target profile.

Our setting differs from both lines of work. Rather than maximizing a scalar reward or encouraging novelty, we train policies to minimize distance to a desired target in property space. This turns molecular generation into a goal-conditioned control problem and makes action representation, transfer, and generalization central RL questions.

3 Method

3.1 Goal-Conditioned Fragment Assembly

We define a goal-conditioned Markov decision process in which a molecule is assembled by sequentially attaching BRICS fragments. At time t, the state is the current partial molecule M_t together with a normalized target vector $\mathbf{g}$. In the main task,

$$\mathbf{g} = (\text{sLogP}, \text{QED}, \text{TPSA}).$$

The episode terminates when the policy selects a stop action or reaches a fixed horizon.

Let $\phi(M)$ denote the normalized molecular property vector of a completed molecule. The target-matching objective is

$$\min \|\phi(M_T) - \mathbf{g}\|_2.$$

We define success as achieving error below $\epsilon = 0.1$ in every target dimension.

3.2 Environment and Action Space

The environment uses a fragment vocabulary extracted from BRICS decompositions of drug-like molecules. At each step, the environment enumerates all chemically valid fragment attachments for the current partial molecule according to BRICS compatibility rules, together with a terminate action. As a result, the policy always acts over a variable valid-action set rather than a fixed global vocabulary.

When site-awareness is disabled, an action identifies a candidate fragment together with compatible BRICS label types on the fragment and molecule. When site-awareness is enabled, an action

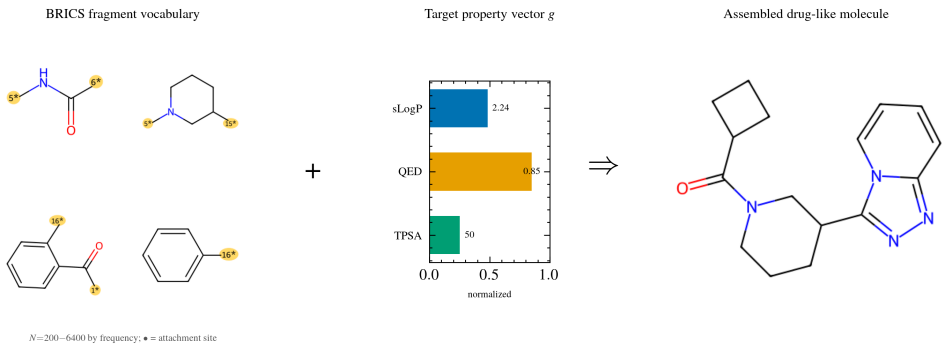

Figure 1: **Input data and RL problem setup.** The agent starts from a seed fragment and a target property vector derived from parent molecules, then constructs a molecule through BRICS-compatible fragment attachments until it terminates.

additionally specifies the exact open site on the current molecule and the exact dummy site on the fragment. This finer-grained action space is chemically richer but also harder to search.

Figure 1 summarizes the input distribution faced by the agent: a BRICS-derived fragment vocabulary, target vectors sampled from parent molecules, and a sequential assembly interface with a variable valid-action set. This makes the problem distinctly RL-shaped: the state is partially constructed, the action set changes after every attachment, and the goal is fixed throughout the episode.

3.3 Policy Families

We evaluate two standard goal-conditioned actor-critic families.

Fingerprint actor-critic. The first uses a 512-bit Morgan fingerprint of the partial molecule concatenated with the goal vector. A multilayer perceptron produces a state embedding, and each valid action is scored using the state embedding together with fragment and BRICS-label features.

Graph actor-critic. The second uses a three-layer message-passing neural network over the molecular graph. The resulting graph embedding is concatenated with the goal vector and used to score each valid action.

Because the valid action set changes from state to state, both policy families normalize only over the currently available actions:

$$\pi_\theta(a_i \mid s) = \frac{\exp(f_\theta(s, a_i))}{\sum_j \exp(f_\theta(s, a_j))}.$$

3.4 Site-Aware Actions, Fragment-Site Embeddings, and CAS

We study three extensions to the basic actor-critic setting.

Site-aware actions. The GNN policy can be given the exact selected molecule site and fragment site rather than only BRICS label types. This exposes more structural information to the policy.

Fragment-site embeddings. For site-aware graph policies, we optionally cache GNN embeddings for fragment dummy atoms and feed those into the action scorer. This allows the policy to distinguish different attachment sites on the same fragment.

Counterfactual Action Scoring (CAS). CAS augments each action with features of the predicted next partial molecule. For every candidate action, the environment simulates the merge and computes a counterfactual feature vector including the next fingerprint, approximate next properties, property deltas, open-site counts, and done indicators. The policy therefore scores the consequence of an action rather than only the action identity. This is especially useful once the action space becomes more structured.

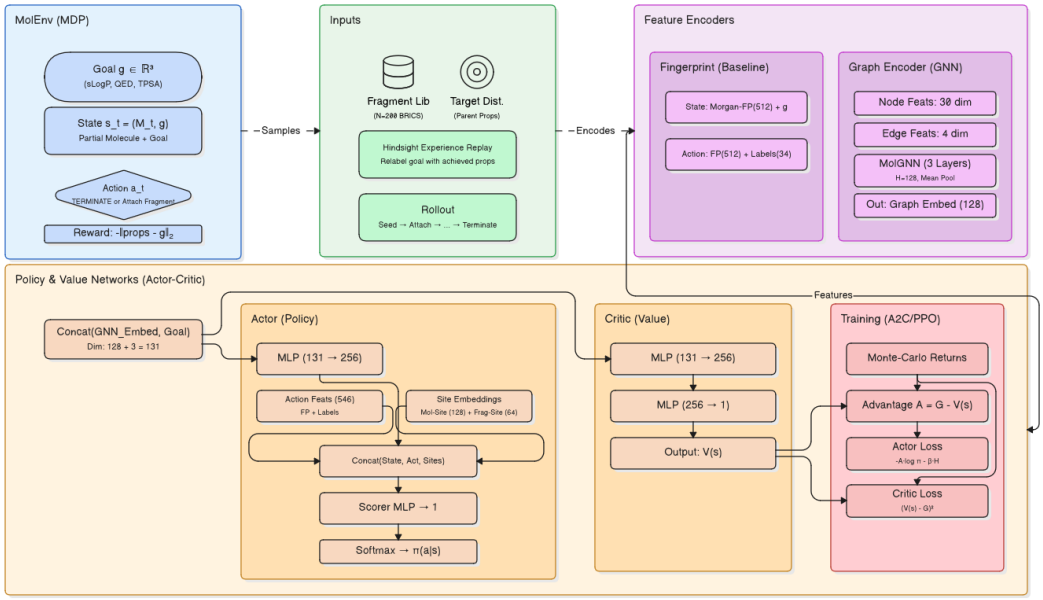

Figure 2: **Goal-conditioned RL system overview.** A fragment library and target sampler instantiate a molecular assembly environment with a variable valid-action set. The agent observes the partial molecule and goal, acts over valid BRICS-compatible attachments, and is trained with sparse reward plus HER relabeling.

3.5 Successor-Style Goal Representation

We also train a successor-style graph model that predicts, for each action, a goal-dimension vector representing the expected achieved property outcome. Actions are then scored by negative distance between this predicted vector and the reward goal:

$$Q(s, a; \mathbf{g}) = -\|\psi(s, a; \mathbf{g}) - \mathbf{g}\|_2.$$

This model uses the same molecular environment and graph encoder as the actor-critic families, but its inductive bias is different: it learns to represent action outcomes directly in goal space.

3.6 Rewards and Training

Our baseline reward is sparse terminal distance:

$$r_T = -\|\phi(M_T) - \mathbf{g}\|_2 - 0.05\, n_{\text{open}},$$

where n_{open} is the number of unfilled attachment sites at termination.

We compare this against three shaping variants: distance-delta reward, potential-based shaping, and property-surrogate reward on partial molecules. To improve sample efficiency, we use Hindsight Experience Replay (HER), relabeling trajectories with their achieved terminal property vectors. We use the same achieved-goal principle when training the successor-style model.

Finally, to make large-scale counterfactual evaluation practical, we use a calibrated additive property surrogate for environment-side property estimation. This speeds up candidate-action scoring without changing the held-out target-matching metric.

Figure 2 summarizes the full RL loop. The fragment library and target sampler define a goal-conditioned environment with a variable valid-action set; the agent scores only those valid actions, and HER relabeling feeds achieved-goal trajectories back into learning. This system view is helpful for interpreting the ablations later in the paper, since different method families intervene on different parts of the same loop: reward shaping changes the environment signal, CAS changes the action representation, and the successor-style model changes the goal-conditioned value representation itself.

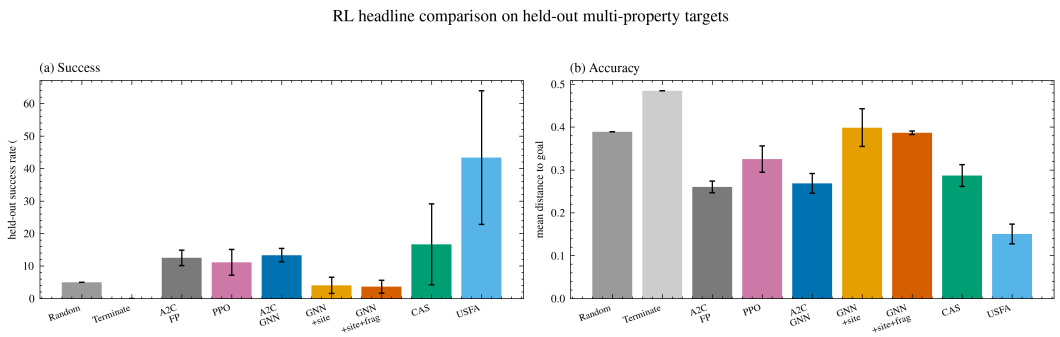

Figure 3: **Held-out three-property comparison among the main policy-gradient families.** A2C with fingerprints and A2C with a plain GNN are the strongest standard baselines. CAS remains competitive with the plain GNN baseline and clearly outperforms the naive site-aware variants.

4 Experimental Setup

The main environment uses the 200 most frequent BRICS fragments extracted from the M3-20M dataset. Target vectors are derived from approximately 300 parent molecules and normalized to $[0, 1]$. Training and evaluation goals are disjoint.

Unless otherwise specified, policies are trained for 6,000 episodes with a maximum horizon of six fragment attachments. We evaluate greedy rollouts on held-out goals and report mean distance and success rate. For the main three-property policy families, results are averaged over 8–20 seeds with 120 evaluation episodes per seed. The successor-style site-aware model is evaluated over 2 seeds in the main summary. We also compare against simple non-learning baselines such as random valid-action assembly to separate genuine policy learning from the intrinsic difficulty of the goal set.

In addition to the main three-property task, we evaluate one-, five-, and seven-property variants, as well as fragment-library sizes from 50 to 6400, to study scaling in both goal dimension and action-space size.

5 Results

5.1 Main Three-Property Performance

Figure 3 summarizes the main held-out three-property comparison among the standard policy-gradient families. Random valid-action assembly achieves 6.7% success with mean distance 0.416. A2C with fingerprints improves this to 11.9% success and 0.277 distance, while A2C with a plain GNN reaches 10.8% and 0.283. PPO with fingerprint states is competitive but weaker in this evaluation setting, reaching 8.1% success and 0.320 distance.

The site-aware actor-critic variants are weaker than the plain baselines. Exact site-aware GNN policies reach 4.7% success and 0.386 distance, while adding fragment-site embeddings further drops performance to 3.0% success and 0.418 distance. This indicates that finer-grained chemistry alone does not solve the control problem; it also enlarges the action space.

The strongest overall model is the successor-style site-aware graph policy, which reaches 34.2% success and mean distance 0.175. Its gain over the actor-critic baselines suggests that representing action outcomes directly in goal space is more effective than simply improving optimization over the standard policy-gradient objective.

5.2 Qualitative Goal Reaching

Figure 4 shows a best single greedy held-out rollout from the strongest successor-style site-aware checkpoint. The policy starts from a simple seed fragment, expands ring structure to raise lipophilicity, and then fine-tunes polarity to move TPSA and QED back toward the target. The resulting rollout

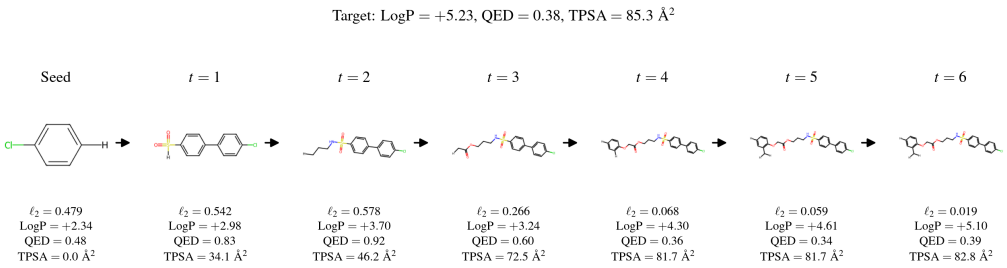

Figure 4: **Best single held-out trajectory from the strongest RL policy.** A successor-style site-aware graph policy incrementally adjusts molecular structure over several fragment attachments, steadily reducing goal-space error until termination near the target profile.

reaches a final goal-space error of 0.039, illustrating that the best policy is not only successful on aggregate metrics but also performs coherent multi-step goal-directed control within an episode.

5.3 Reward Design

Figure 6 shows the reward ablation for the fingerprint and GNN policy families. Sparse terminal reward yields the most reliable convergence in both cases. This pattern is also reflected in held-out evaluation. For fingerprints, sparse reward reaches 11.9% success and 0.277 distance, compared with 8.1% and 0.314 for property-surrogate reward and 8.1% and 0.378 for distance-delta reward. For graph policies, sparse reward reaches 10.8% and 0.283, compared with 10.0% and 0.328 for property-surrogate reward and 5.1% and 0.387 for distance-delta reward.

These results show that denser rewards are not the main solution to this task. Partial-molecule shaping can produce useful short-horizon training signals, but those signals do not align as well with held-out goal matching as the sparse terminal objective does.

5.4 Optimization Stability and Generalization

The earlier version of the project emphasized a second question beyond reward design: how much of the sparse-goal difficulty is optimization, and how much is genuine generalization to unseen targets? Figure 5 helps answer that question. Value normalization materially stabilizes early training, but aggressive reward-scale tuning and heavier HER relabeling do not reliably improve held-out performance. In particular, larger HER k values fail to help once the task is already goal-conditioned, and the GNN policy becomes less stable as relabeling increases. The algorithm panel tells a similar story: PPO and PPO+GAE can reduce training distance, but those gains are modest and do not overturn the broader held-out ranking from Figure 3. Together, these ablations reinforce the paper's main conclusion that the bottleneck is not simply optimizing a sparse objective, but learning a representation that transfers to new goals.

5.5 Counterfactual Action Scoring

Figure 6 also reports the CAS ablation. The plain GNN baseline achieves 10.8% success and 0.283 distance. Adding exact site actions alone reduces performance to 4.7% and 0.386. CAS largely recovers that loss: the plain CAS variant reaches 9.4% success and 0.299 distance.

This is the clearest policy-side architectural result in the paper. Once the action space is made more expressive, the policy needs information about what the action *does*, not only which fragment it selects. CAS provides that by scoring predicted next-state consequences directly.

5.6 Scaling in Target Dimension and Fragment Vocabulary

Figure 7 shows that difficulty increases sharply with both target dimensionality and fragment-library size.

As the number of target properties increases, per-property error remains comparatively stable while joint success drops rapidly. For example, the fingerprint policy achieves mean distance 0.195 on

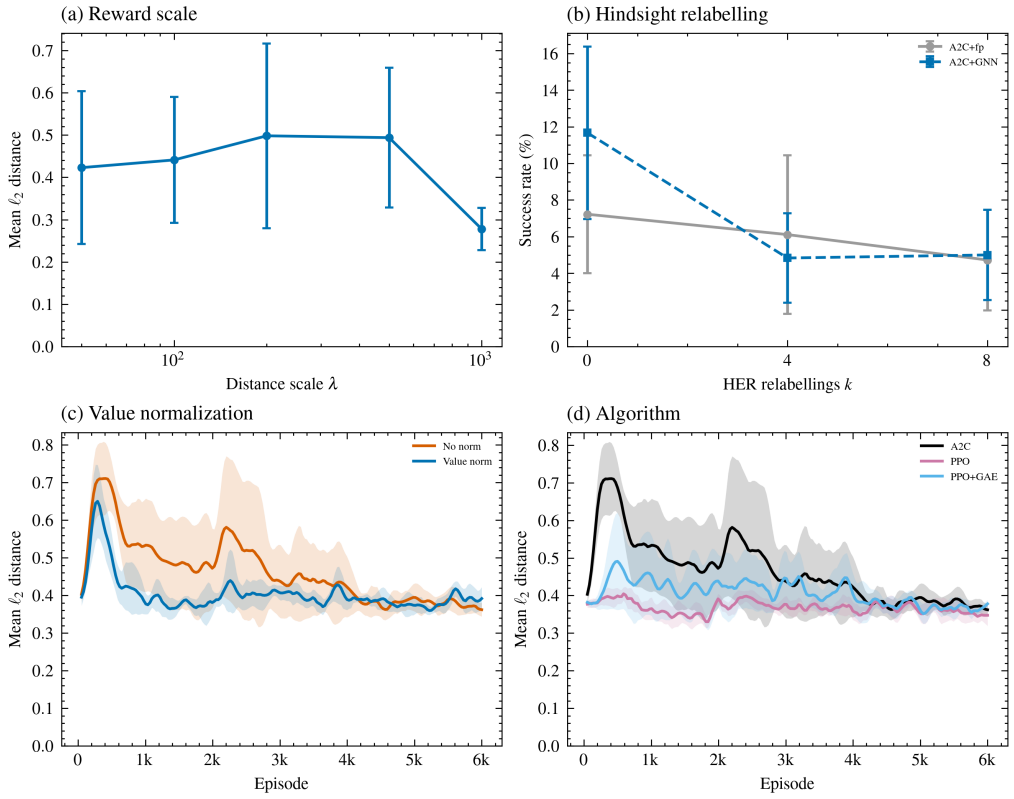

Figure 5: **Training stability and sensitivity ablations.** Reward scale, HER relabeling frequency, value normalization, and PPO-style updates affect optimization stability, but none changes the central generalization story: stable sparse-reward training matters more than increasingly aggressive shaping or optimization complexity.

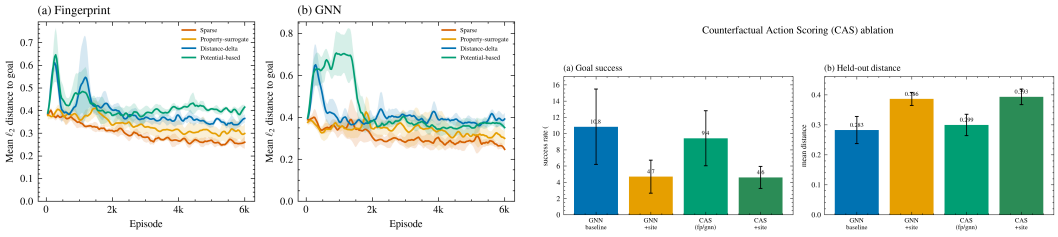

Figure 6: **Reward and counterfactual-action ablations.** Left: sparse terminal reward yields the strongest overall training behavior and best held-out performance among the standard reward families. Right: CAS substantially improves over naive site-aware graph policies by scoring predicted next partial molecules.

the one-property `QED` task, 0.391 on the five-property task, and 0.468 on the seven-property task; the corresponding graph distances are 0.230, 0.444, and 0.578. Since the left panel reports per-property distance, the curves remain far flatter than raw joint success would suggest. This means the policies learn useful directional control over individual properties, but satisfying multiple constraints simultaneously becomes increasingly difficult.

The fragment-vocabulary sweep shows a similar combinatorial effect. Success is highest around 200 fragments, reaching 8.2%, and then declines as the library grows, falling to 3.5% at 6400 fragments. The training curves broaden noticeably for the larger vocabularies, indicating higher variance and a harder exploration problem. Larger libraries add expressive power, but they increase branching factor even faster.

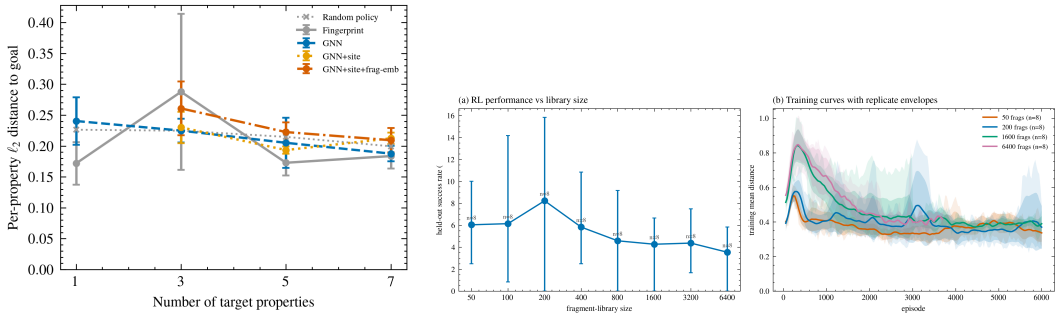

Figure 7: **Scaling in goal dimension and action-space size.** Left: per-property distance remains relatively stable as the number of controlled properties increases, indicating that the main challenge is joint constraint satisfaction rather than single-property steering. Right: performance peaks near 200 fragments and degrades as the library expands, showing that exploration and branching factor dominate the benefit of a larger action space.

6 Discussion

Two conclusions follow directly from these results. First, sparse terminal reward is not the main obstacle it appears to be. With HER and an appropriate state-action representation, standard RL already learns nontrivial target matching. The more important question is how the policy represents action consequences under a large valid-action set.

Second, richer chemistry-aware action spaces help only when paired with richer evaluation of action outcomes. Exact site-aware factorization alone increases difficulty. CAS partially resolves that by making the policy locally model-based over predicted next partial molecules. The successor-style graph model goes further by representing action outcomes directly in goal space, which explains its much stronger held-out performance.

Overall, the central RL bottleneck is generalization under combinatorial branching, not merely reward sparsity. The strongest gains come from structured goal-conditioned value representations and better evaluation of candidate actions.

7 Conclusion

We presented a goal-conditioned reinforcement learning framework for targeted molecular design through BRICS fragment assembly. Across reward ablations, architecture comparisons, and scaling studies, three findings are consistent. Sparse terminal reward remains the strongest standard objective. Counterfactual action evaluation is more effective than naive increases in action-space granularity. Successor-style goal-space representations provide the largest gains in held-out target matching.

These results position targeted molecular design as a challenging RL problem defined by variable valid-action sets, sparse terminal objectives, and strong generalization demands. Future progress is therefore likely to come from tighter integration of planning, transfer, and goal-conditioned representation learning rather than denser reward shaping alone.

8 Team Contributions

- **Katie Liu:** Project framing, literature review, and manuscript preparation, GNN + PPO experiments, and reward shaping analysis.
- **Megan Santhumayor:** Reward-design experiments, architecture ablations, and generalization analysis.
- **Asmani Yamin:** Fragment environment construction, dataset processing, large-scale evaluation infrastructure, and implementation of the successor-style and counterfactual experiments.

Changes from Proposal The initial hypothesis was that denser intermediate reward would be the dominant ingredient for solving sparse-goal molecular assembly. The final results instead show that the decisive factors are action evaluation and goal-conditioned outcome representation: sparse reward remains strong, CAS improves structured action spaces, and successor-style value representations produce the best overall performance.

AI Disclosure During the development of this project, we utilized AI tools including ChatGPT/Claude/Gemini and GitHub Copilot to assist with development and formatting tasks. Specifically, GitHub Copilot was used as an autocomplete aid to generate boilerplate PyTorch code, structure data loaders for the M3-20M dataset, and streamline standard RDKit molecule manipulations. Large language models were utilized to help debug infrastructure and containerization issues (e.g., interpreting tracebacks and configuring Modal volumes), as well as to assist with LaTeX formatting and refining the academic tone of our poster and final report.

References

Masoud Amiri and Zahra Nasirinia. 2026. HybridMolGen: A Unified Framework for Goal-Directed Molecular Generation via Multi-Objective Reinforcement Learning. *Bioinformatics* (2026). doi:10.1093/bioinformatics/btag170 btag170.

Jinyeong Park, Jaegyoon Ahn, Jonghwan Choi, and Jibum Kim. 2025. Mol-AIR: Molecular Reinforcement Learning with Adaptive Intrinsic Rewards for Goal-Directed Molecular Generation. *Journal of Chemical Information and Modeling* 65, 5 (2025), 2283–2296. doi:10.1021/acs.jcim.4c01669

Jing Wang and Fei Zhu. 2024. Multi-objective molecular generation via clustered Pareto-based reinforcement learning. *Neural Networks* 179 (Nov. 2024), 106596. doi:10.1016/j.neunet.2024.106596