Pushing Biomolecular Utility-Diversity Frontiers with Supergroup Relative Policy Optimization

Xinwu Ye[1,2,3*] **He Cao**[3*] **Hao Li**[4] **Bin Feng**[2]
Zijing Liu[2] **Xiangru Tang**[5] **Yu Li**[2†] **Shenghua Gao**[1†]

[1]The University of Hong Kong
[2]International Digital Economy Academy
[3]AbIntelli Bio
[4]Peking University
[5]Yale University

Abstract

Biomolecular generators are often adapted with reward feedback to improve task-specific utility, but pushing utility alone can concentrate generation on a narrow family of candidates. Maintaining diversity is difficult because sample diversity is a set-level property. We introduce Supergroup Relative Policy Optimization (SGRPO), a flexible GRPO-style framework that directly constructs rewards from set-level diversity. For each condition, SGRPO samples a supergroup of candidate sets, compares their diversity under the same condition, and redistributes the group diversity reward to individual rollouts through leave-one-out diversity contributions before combining it with rollout-level utility. This design decouples SGRPO from a particular generator, utility reward, or diversity metric, and allows instantiation with different GRPO-style approaches. We evaluate SGRPO on three main settings: *de novo* small-molecule design, pocket-based small-molecule design, and *de novo* protein design, instantiating it with both GRPO and Coupled-GRPO across autoregressive and discrete diffusion generators. Across decoding sweeps, SGRPO expands the utility-diversity Pareto frontier and achieves the best frontier-level metrics relative to baselines. Our analyses further show that direct set-level diversity rewards remain effective with small groups and help preserve broader generation-distribution coverage during post-training. The code is available at `https://github.com/IDEA-XL/SGRPO`.

1 Introduction

Biomolecular generation aims to produce candidates that satisfy chemical or biological design objectives, and reinforcement learning (RL) provides a natural framework for post-training pretrained generators from reward feedback toward desired properties, structures, or functions [39]. In practice, however, generation quality is not determined by utility alone. A model that maximizes a property score, docking proxy, or protein-level objective may concentrate probability mass on a narrow family of candidates, while a highly diverse generator may fail to deliver enough high-utility samples. This creates a utility-diversity trade-off: different downstream settings may prefer different operating points, often modulated by decoding choices such as temperature, so the relevant objective is not a single best reward value but an improved Pareto frontier of attainable utility-diversity pairs. While many successful RL approaches are closely tailored to specific model classes, molecular or protein

*Equal contribution. Email: `xinwuye43@connect.hku.hk`
†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

representations, and design settings [39, 59, 13, 54], our goal is a broadly applicable post-training principle. We evaluate it across different generator families, conditioning settings, utility functions, and diversity metrics, while leaving broader task-specific instantiations to future work.

A more broadly applicable class of diversity-aware RL methods encourages diversity through memory- or history-dependent novelty penalties [7, 36]. Such methods down-weight candidates that are too similar to previously sampled molecules, scaffolds, clusters, or neighborhoods, and can be effective in practice. However, novelty relative to past samples is only an indirect surrogate for the diversity of the current candidate set produced under a given condition. As a result, these methods may over-penalize useful high-density modes or induce distributional drift during post-training. More fundamentally, the target quantity itself is set-level: diversity is defined over collections of samples, whereas policy optimization updates individual rollouts. This raises the central question of this paper: *can we optimize sample-set diversity directly, as a first-class objective, while still assigning useful credit to individual generated candidates?*

We address this with Supergroup Relative Policy Optimization (SGRPO), a simple framework for directly rewarding sample-set diversity together with rollout-level utility. For each condition, SGRPO samples multiple candidate sets from the current policy, scores each set using a user-specified diversity metric, and compares sets only against other sets generated under the same condition. To make this set-level signal actionable for policy learning, SGRPO redistributes each set's diversity reward to its members through leave-one-out diversity contributions, so candidates that genuinely support set diversity receive stronger credit. The resulting *supergroup*-relative advantage can be instantiated with different GRPO-style optimizers.

We instantiate SGRPO with two GRPO-style optimizers and evaluate it on three main biomolecular generation settings: unconditional *de novo* small-molecule design with GenMol [30], pocket-based small-molecule design with GenMol-P, our pocket-conditioned variant of GenMol, and unconditional *de novo* protein design with ProGen2 [37]. Across decoding sweeps, SGRPO consistently improves the attainable utility-diversity Pareto frontier over baselines. It remains effective even with small group sizes and better preserves generation-distribution coverage during post-training, showing that directly rewarding set-level diversity can yield robust gains across both molecule and protein generation.

2 Related Work

2.1 Objective Optimization in Biomolecular Generation

Objective optimization in biomolecular generation is commonly approached either by conditioning or guiding generators toward desired properties, structures, or functions [32, 29, 3, 28, 12, 45, 56, 50], or by improving candidates from oracle feedback through methods such as latent-space Bayesian or evolutionary optimization, iterative retraining, preference optimization, and reinforcement learning [19, 21, 9, 8, 52, 58, 11, 55, 31]. We focus on the RL branch, which provides a general feedback-driven post-training formulation in which generated candidates are scored by objectives such as molecular properties, stability, or multi-objective reward functions, and the generator is updated to increase the likelihood of high-reward samples. RL-based biomolecular optimization has been instantiated across diverse representations, including SMILES sequence models such as REINVENT, ReLeaSE, and ChemRLformer [39, 43, 18], graph- or fragment-based molecular generators such as GCPN, MolDQN, RationaleRL, LibINVENT, and DrugEx v3 [59, 60, 27, 16, 35], and protein sequence or structure-conditioned generators such as model-based RL for biological sequence design, RL-DIF, and ProteinZero [1, 13, 54]. This broad applicability of RL motivates our focus on diversity-aware reward design at the post-training level.

2.2 Diversity-Promoting RL for Biomolecular Generation

Diversity-promoting RL has been explored in both molecular and protein generation to mitigate mode collapse and sample redundancy. One line of work builds diversity objectives around the structure of a specific generator or design task, for example by jointly generating multiple SMILES strings in a single sequence [26], exploiting augmented SMILES and score reuse [6], incorporating diversity into fragment-based molecular construction [57], pairing exploitation and exploration policies during generation [34], or adding task-specific regularization in protein inverse folding and

sequence design [14, 54, 41]. A more broadly applicable family instead promotes diversity through indirect reward shaping, such as diverse mini-batch selection [49], memory- or scaffold-based penalties and filters [7, 42, 36, 51, 22, 61], distance-to-memory or novelty rewards [25, 48, 40, 10], entropy regularization [47], or count-based visitation bonuses [2]. These approaches have shown empirical benefits, but they are either tightly coupled to particular generator interfaces or optimize indirect proxies such as novelty, entropy, or history-relative exploration rather than the diversity of the current generated sample set itself. Our work focuses on this latter gap.

3 Problem Setup: Utility–Diversity Frontier in Biomolecular Generation

3.1 Setup

We consider a conditional biomolecular generator $\pi_\theta(x \mid \mathcal{C})$, where x is a generated candidate and $\mathcal{C}$ is the conditioning input. Depending on the task, $\mathcal{C}$ may be empty, a task or property specification, or a target environment such as a protein binding pocket. The formulation is model-agnostic and applies to pretrained *de novo* molecular generators, pocket-conditioned molecular generators, and protein language models. Each candidate receives an individual utility score $r(x, \mathcal{C})$. The exact form of r depends on the domain. For small molecules, it may combine drug-likeness and synthesizability, and in pocket-conditioned generation, it may additionally include target-specific terms such as docking. For proteins, utility may reflect sequence plausibility, stability, foldability, or developability.

We also care about diversity among the generated outputs. For a set of K candidates generated under the same condition, denoted by $G = \{x_1, \ldots, x_K\}$, let $D(G)$ be a set-level diversity score. This score may measure internal diversity, scaffold diversity, sequence diversity, or cluster coverage. The key point is that diversity is *not* a per-sample reward: in general, it depends on the relationships among samples in the set and cannot be reduced to independently scoring each candidate. Optimizing diversity, therefore, requires reasoning over groups of outputs rather than isolated generations.

3.2 Utility–diversity frontier

Let $p(\mathcal{C})$ denote the distribution over generation conditions. At inference time, the trained generator is paired with a decoding strategy $a \in \mathcal{A}$, such as a sampling temperature or related decoding hyperparameters. Together, (θ, a) determine two expected quantities: the expected individual utility, denoted by $U(\theta, a)$, and the expected set-level diversity, denoted by $V(\theta, a)$. Here $U(\theta, a)$ is computed from single generated samples, while $V(\theta, a)$ is computed from sets of K samples drawn under the same condition.

Varying the decoding strategy induces a set of attainable utility–diversity trade-offs for the generator, which we denote by $\mathcal{P}(\theta) = \{(U(\theta, a), V(\theta, a)) : a \in \mathcal{A}\}$. A point on this set is Pareto-optimal if no other decoding strategy achieves both higher utility and higher diversity at the same time.

Our goal is to improve this frontier itself. Rather than optimizing only utility or only diversity, we seek post-training methods that push $\mathcal{P}(\theta)$ outward, so that the same generator can achieve better utility at a fixed diversity level, better diversity at a fixed utility level, or both.

4 Supergroup Relative Policy Optimization

SGRPO is a post-training reinforcement learning method for improving the utility–diversity frontier of a pretrained biomolecular generator. Its central idea is simple: since diversity is a set-level property, training should compare *sets* of candidates generated under the same condition, rather than scoring each candidate in isolation. For each condition $\mathcal{C}$, SGRPO samples several candidate groups, scores each group by diversity, redistributes the group-level signal back to individual candidates according to their within-group contribution, and then applies a PPO-style update using a same-condition relative advantage. Figure 1 illustrates the overall pipeline, and detailed pseudocode is provided in Appendix A.

4.1 Same-condition supergroups

For a condition $\mathcal{C}$, we sample M groups from the old policy, each containing K independently generated rollouts. Denote the resulting collection by $\mathcal{S}(\mathcal{C}) = \{G_1, \ldots, G_M\}$, where $G_m =$

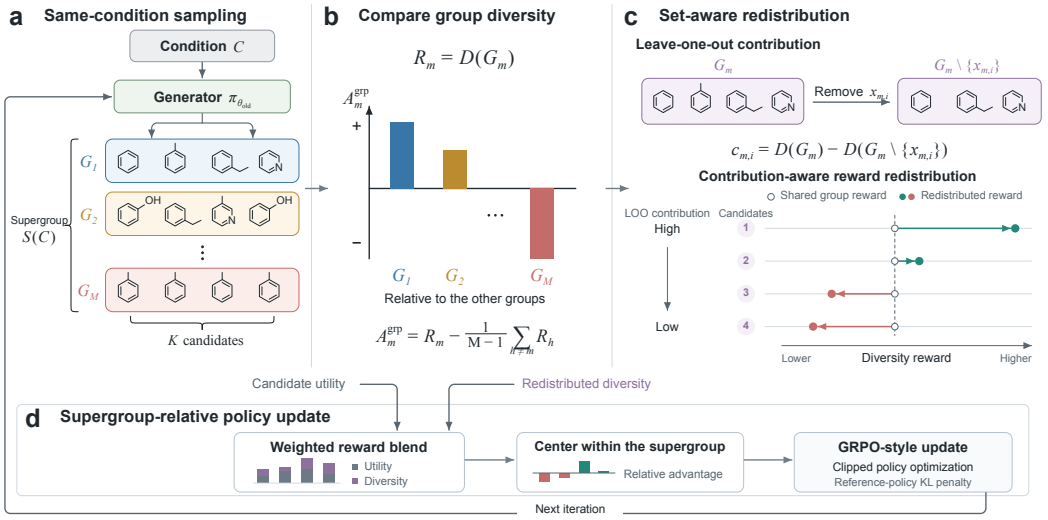

Figure 1: Overview of SGRPO. For each condition, SGRPO samples a same-condition supergroup, computes rollout-level utility and group-level diversity, compares groups by leave-one-out group-relative diversity, redistributes the diversity signal within each group according to leave-one-out set contributions, centers the composed rewards over the supergroup, and updates the policy with a PPO-style objective and KL regularization.

$\{x_{m,1}, \ldots, x_{m,K}\}$ and each $x_{m,i} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathcal{C})$. We refer to $\mathcal{S}(\mathcal{C})$ as a *supergroup*. It contains $N = MK$ candidates generated under the same condition, with M controlling how many alternative groups are compared and K controlling the size of each group.

Restricting comparisons to a single supergroup is important. In conditional biomolecular generation, different conditions can have very different intrinsic difficulty, so comparing samples across conditions would confound policy quality with condition difficulty. SGRPO instead performs only *local* comparisons: under the same $\mathcal{C}$, which groups are more diverse, and which rollouts are more useful?

4.2 Utility and group-level diversity

Each rollout $x_{m,i}$ receives an individual utility reward $r_{m,i} = r(x_{m,i}, \mathcal{C})$, while each group G_m receives a diversity score $R_m = D(G_m)$. Thus, utility is defined at the candidate level, whereas diversity is defined over the whole group.

To compare groups generated under the same condition, we center group diversity within the supergroup. Let $\bar{R} = \frac{1}{M}\sum_{h=1}^{M} R_h$. We define the group-relative diversity signal as

$$A_m^{\text{grp}} = R_m - \frac{1}{M-1}\sum_{h \neq m} R_h = \frac{M}{M-1}(R_m - \bar{R}). \tag{1}$$

This is simply a leave-one-out comparison among same-condition groups: $A_m^{\text{grp}} > 0$ means that G_m is more diverse than its alternatives, and $A_m^{\text{grp}} < 0$ means the opposite. For the normalized pairwise diversity used in our experiments, average diversity over groups of size K is an unbiased proxy for the diversity of a larger same-condition sample, so optimizing group diversity is aligned with improving diversity at the supergroup level. Formal statements are given in Appendix B.

4.3 Set-aware redistribution

The diversity score R_m evaluates an entire group, but policy optimization ultimately acts on individual rollouts. SGRPO bridges this gap by assigning more of the diversity signal to candidates that matter more for the diversity of their own group.

For each rollout $x_{m,i} \in G_m$, we first compute its leave-one-out contribution $c_{m,i} = D(G_m) - D(G_m \setminus \{x_{m,i}\})$, and standardize these contributions within the group as $z_{m,i} = (c_{m,i} -$

$\bar{c}_m)/(\sigma(c_{m,\cdot}) + \zeta)$. We then form two sign-aware softmax weight vectors:

$$w^{\pm}_{m,i} = K \cdot \frac{\exp(\pm z_{m,i}/\tau_c)}{\sum_{j=1}^{K} \exp(\pm z_{m,j}/\tau_c)}. \tag{2}$$

By construction, $\sum_i w^{+}_{m,i} = \sum_i w^{-}_{m,i} = K$. Here $w^{+}_{m,i}$ emphasizes candidates with larger diversity contributions, while $w^{-}_{m,i}$ emphasizes candidates with smaller ones.

We then define the redistributed diversity reward

$$\widetilde{R}_{m,i} = R_m + [A^{\mathrm{grp}}_m]_+(w^{+}_{m,i} - 1) - [-A^{\mathrm{grp}}_m]_+(w^{-}_{m,i} - 1), \tag{3}$$

where $[a]_+ = \max(a, 0)$. Proposition C.1 in Appendix C proves that this redistribution preserves the mean group reward. The redistribution is sign-aware. If $A^{\mathrm{grp}}_m > 0$, then G_m is more diverse than its same-condition alternatives, and the extra positive signal is concentrated on candidates that contributed more to that diversity. If $A^{\mathrm{grp}}_m < 0$, then G_m is less diverse, and the negative signal is concentrated on candidates that contributed less. By construction, redistribution preserves the original group reward on average, i.e., $\frac{1}{K}\sum_{i=1}^{K} \widetilde{R}_{m,i} = R_m$.

4.4 Supergroup-relative policy update

We combine candidate-level utility and redistributed diversity into a single reward, $\widehat{r}_{m,i} = (1 - \lambda)r_{m,i} + \lambda\widetilde{R}_{m,i}$, where $\lambda \in [0, 1]$ controls the utility–diversity trade-off. Let $\bar{r}_{\mathcal{S}} = \frac{1}{MK}\sum_{m=1}^{M}\sum_{i=1}^{K} \widehat{r}_{m,i}$ denote the average composed reward within the supergroup. The final supergroup-relative advantage is

$$A_{m,i} = \frac{MK}{MK - 1}\big(\widehat{r}_{m,i} - \bar{r}_{\mathcal{S}}\big). \tag{4}$$

Equivalently, this is a leave-one-out baseline over all rollouts in the same supergroup. Since all rollouts in the supergroup share the same condition, $A_{m,i}$ measures whether a rollout is better or worse than its local same-condition alternatives after utility and diversity have been combined.

We then update the policy with a clipped PPO objective and a KL penalty to a reference policy π_{ref}. Let $\rho_{m,i} = \pi_\theta(x_{m,i} \mid \mathcal{C})/\pi_{\theta_{\mathrm{old}}}(x_{m,i} \mid \mathcal{C})$. The objective is

$$\mathcal{L}_{\mathrm{SGRPO}}(\theta) = -\mathbb{E}\Big[\min\Big(\rho_{m,i}A_{m,i}, \mathrm{clip}(\rho_{m,i}, 1 - \epsilon, 1 + \epsilon)A_{m,i}\Big)\Big] + \beta\,\mathbb{E}\Big[\mathrm{KL}\Big(\pi_\theta(\cdot \mid \mathcal{C}) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid \mathcal{C})\Big)\Big] \tag{5}$$

The expectation is over conditions, sampled supergroups, and rollouts. In practice, SGRPO alternates between sampling same-condition supergroups, computing group-level diversity and rollout-level redistributed rewards, and updating the policy with the objective above. Full pseudocode is provided in Appendix A.

5 Experiments

We evaluate whether SGRPO expands the utility-diversity Pareto frontier across three biomolecular generation settings: unconditional *de novo* small-molecule design, pocket-based small-molecule design, and *de novo* protein design. In each setting, we decode each model under a sweep of operating points and summarize every operating point by its utility and set-level diversity. We compare the resulting frontiers against the pretrained generator, GRPO, memory-assisted GRPO when applicable, Diverse Mini-Batch GRPO (DMB-GRPO), and Entropy-Regularized GRPO (ER-GRPO), using the same Pareto-level metrics across tasks. This evaluation tests whether supergroup-relative diversity pressure improves the trade-off frontier itself, rather than merely shifting generation toward higher utility or higher randomness.

5.1 Evaluation Protocol

Each experiment evaluates a generator under a range of task-specific decoding settings, treating each setting as one utility–diversity operating point. For a given model, this yields a set of points $A = \{(U_i, V_i)\}_{i=1}^{n}$, where U_i and V_i denote the utility and diversity of the i-th setting. Both metrics

are scaled to $[0, 1]$, with higher values indicating better performance. We summarize performance by the non-dominated subset of A, denoted by $ND(A)$. A point belongs to $ND(A)$ if no other decoding setting achieves at least as much utility and at least as much diversity, with one of them being strictly better. In other words, $ND(A)$ is the Pareto frontier of the model under the evaluated decoding settings.

Hypervolume. For each experiment, we use a common reference point $r_{\text{exp}} = (r_U, r_V)$, where r_U and r_V are the minimum utility and diversity observed across all operating points from all compared methods. In two dimensions, the hypervolume of a model is the area of the staircase-shaped region dominated by its non-dominated operating points and bounded below by r_{exp}. Equivalently, $HV(A; r_{\text{exp}}) = \text{Area}\big(\cup_{(U,V)\in ND(A)}[r_U, U] \times [r_V, V]\big)$. Thus, HV is the union area of axis-aligned rectangles induced by the non-dominated set, rather than the area of a single rectangle. Larger HV indicates that the frontier extends further toward high utility and high diversity and/or spans a broader utility–diversity range. Because the reference point is experiment-specific, HV is intended for within-experiment comparison rather than direct comparison across tasks.

Distance to Ideal Point. Let $z^\star = (U^\star, V^\star)$ denote the ideal point, whose coordinates are the best attainable values of the two objectives. For an operating-point set A, we report $\text{DIP}(A, z^\star) = \min_{(U,V)\in A} \sqrt{(U^\star - U)^2 + (V^\star - V)^2}$. Since utility and diversity are scaled to $[0, 1]$, we set $z^\star = (1, 1)$. Lower distance is better.

R2 Indicator. R2 evaluates an operating-point set under multiple utility-diversity preference weights. For a weight $\lambda = (\lambda_U, \lambda_V)$, we define the best weighted Tchebycheff shortfall as $g(A \mid \lambda, z^\star) = \min_{(U,V)\in A} \max\{\lambda_U(z^\star_U - U), \lambda_V(z^\star_V - V)\}$, and compute $R2(A, \Lambda, z^\star) = \frac{1}{|\Lambda|}\sum_{\lambda\in\Lambda} g(A \mid \lambda, z^\star)$. In our implementation, A is the full model-specific sweep set and $\Lambda = \{\lambda^{(\ell)} = (\ell/100, 1 - \ell/100)\}_{\ell=0}^{100}$. Lower R2 means a smaller average weighted worst-case shortfall to $z^\star$.

Table 1: Frontier-level metrics for the three tasks. Each cell reports mean ± 95% confidence interval over five independent sweep runs. HV is higher-is-better, while distance to ideal point (DIP) and R2 are lower-is-better. All small-molecule RL methods use coupled-GRPO, while protein methods use GRPO.

De novo Small-Molecule Design						
Metric	Original	GRPO	Mem-GRPO	DMB-GRPO	ER-GRPO	SGRPO
HV ↑	$0.0686_{\pm 0.0030}$	$0.0703_{\pm 0.0029}$	$0.0657_{\pm 0.0025}$	$\mathbf{0.0774_{\pm 0.0030}}$	$0.0694_{\pm 0.0027}$	$0.0758_{\pm 0.0029}$
DIP ↓	$0.2719_{\pm 0.0015}$	$0.2677_{\pm 0.0013}$	$0.2696_{\pm 0.0025}$	$0.2581_{\pm 0.0022}$	$0.2677_{\pm 0.0019}$	$\mathbf{0.2551_{\pm 0.0020}}$
R2 ↓	$0.1001_{\pm 0.0004}$	$0.1017_{\pm 0.0005}$	$0.1055_{\pm 0.0002}$	$0.0962_{\pm 0.0006}$	$0.0988_{\pm 0.0004}$	$\mathbf{0.0955_{\pm 0.0004}}$
Pocket-Based Small-Molecule Design						
Metric	Original	GRPO	Mem-GRPO	DMB-GRPO	ER-GRPO	SGRPO
HV ↑	$0.0266_{\pm 0.0031}$	$0.0098_{\pm 0.0016}$	–	$0.0219_{\pm 0.0023}$	$0.0250_{\pm 0.0017}$	$\mathbf{0.0630_{\pm 0.0033}}$
DIP ↓	$0.4643_{\pm 0.0044}$	$0.7459_{\pm 0.0153}$	–	$0.6757_{\pm 0.0119}$	$0.6701_{\pm 0.0120}$	$\mathbf{0.3797_{\pm 0.0022}}$
R2 ↓	$0.2381_{\pm 0.0022}$	$0.3836_{\pm 0.0078}$	–	$0.3497_{\pm 0.0066}$	$0.3457_{\pm 0.0062}$	$\mathbf{0.1807_{\pm 0.0007}}$
De novo Protein Design						
Metric	Original	GRPO	Mem-GRPO	DMB-GRPO	ER-GRPO	SGRPO
HV ↑	$0.2752_{\pm 0.0076}$	$0.2140_{\pm 0.0054}$	$0.0306_{\pm 0.0009}$	$0.0022_{\pm 0.0000}$	$0.3128_{\pm 0.0052}$	$\mathbf{0.3687_{\pm 0.0086}}$
DIP ↓	$0.4279_{\pm 0.0076}$	$0.6519_{\pm 0.0058}$	$1.0128_{\pm 0.0011}$	$1.0195_{\pm 0.0000}$	$0.5561_{\pm 0.0099}$	$\mathbf{0.3538_{\pm 0.0048}}$
R2 ↓	$0.2201_{\pm 0.0027}$	$0.3023_{\pm 0.0034}$	$0.4840_{\pm 0.0024}$	$0.5145_{\pm 0.0001}$	$0.1832_{\pm 0.0022}$	$\mathbf{0.1693_{\pm 0.0036}}$

5.2 *De novo* Small-Molecule Design

Setup. We study unconditional *de novo* small-molecule design with GenMol [30], a discrete diffusion language model that generates molecules as SAFE strings [38]. Unlike autoregressive LMs, GenMol generates samples through iterative denoising, so applying GRPO requires a diffusion-compatible training objective. We therefore instantiate SGRPO on top of coupled-GRPO [20], which adapts GRPO-style relative policy optimization to discrete diffusion models via coupled denoising

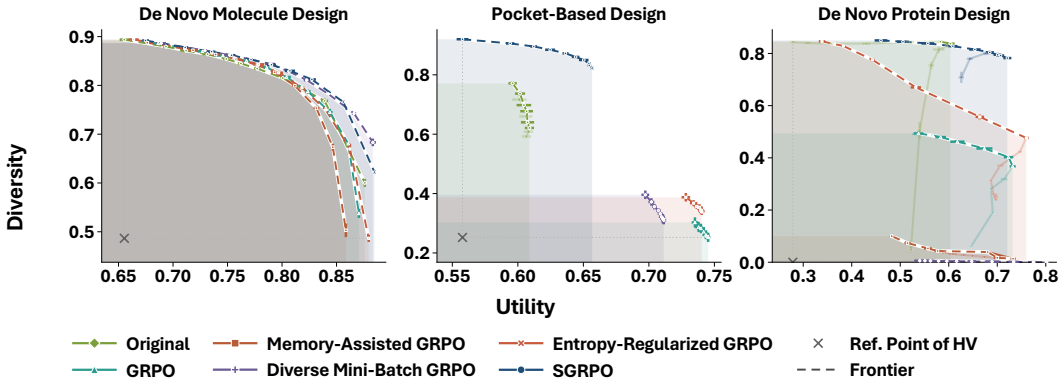

Figure 2: Utility–diversity operating points for de novo small-molecule design, pocket-based small-molecule design, and de novo protein design. Each marker corresponds to one decoding setting and reports the mean utility and diversity over five independent generation sweeps. Error bars show 95% confidence intervals on both axes. Dashed lines trace the method-specific non-dominated subsets.

samples. We use $K = 64$ rollouts per group and $M = 8$ groups per supergroup, matching the 512 rollouts per condition used by GRPO. The molecule-level utility in this experiment is defined by drug-likeness and synthetic accessibility. We use QED [5] as a normalized drug-likeness score in $[0, 1]$, and denote the raw synthetic accessibility score [15] by $\mathrm{SA}(x)$, where lower values indicate easier synthesis. We convert it into a high-is-better score $s_{\mathrm{SA}}(x) = \mathrm{clip}((6 - \mathrm{SA}(x))/5, 0, 1)$, and define the rollout utility as $u(x) = \alpha_{\mathrm{QED}}\mathrm{QED}(x) + \alpha_{\mathrm{SA}} s_{\mathrm{SA}}(x)$. Unless otherwise noted, we use $\alpha_{\mathrm{QED}} = 0.6$ and $\alpha_{\mathrm{SA}} = 0.4$, slightly prioritizing drug-likeness while retaining synthetic accessibility as a feasibility-oriented component. We keep these weights fixed across methods to compare their frontiers under the same utility definition. Sample diversity is measured by internal diversity over valid generated molecules using Morgan-fingerprint Tanimoto distances [4]. Specifically, $V(S) = 1 - \frac{2}{|S|(|S|-1)} \sum_{i<j} s_{ij}$, where $s_{ij} = \mathrm{Tan}(\phi(x_i), \phi(x_j))$, $\phi(x)$ is the Morgan fingerprint of molecule x, and Tan denotes Tanimoto similarity. We compare SGRPO (denoted as coupled-SGRPO) against the pretrained GenMol model, coupled-GRPO, Memory-assisted RL-based coupled-GRPO [7], DMB-GRPO, and ER-GRPO. For Pareto evaluation, each model is decoded under the same sweep of GenMol randomness ρ and temperature τ, using ten paired settings: $(0.1, 0.5)$, $(0.2, 0.65)$, $(0.3, 0.8)$, $(0.4, 0.95)$, $(0.5, 1.1)$, $(0.6, 1.25)$, $(0.7, 1.4)$, $(0.8, 1.55)$, $(0.9, 1.7)$, and $(1.0, 2.0)$. At each sweep point, we generate 1000 molecules per model and compute both utility metrics and internal diversity over the valid molecules generated at that point.

Result. In Figure 2, SGRPO retains more diversity in the high-utility region than pretrained GenMol, GRPO, Memory-Assisted GRPO, and ER-GRPO. DMB-GRPO yields a similarly strong frontier and retains more diversity near the highest-utility endpoint of the sweep. In Table 1, SGRPO achieves the lowest mean DIP (0.2551) and R2 (0.0955). DMB-GRPO has a slightly higher mean HV, 0.0774 ± 0.0030 versus 0.0758 ± 0.0029, with overlapping 95% confidence intervals. SGRPO is therefore competitive with DMB-GRPO on this task.

5.3 Pocket-Based Small-Molecule Design

Setup. We train GenMol-P for pocket-based small-molecule design. GenMol-P initializes from the pretrained GenMol and adds pocket-prefix conditioning: a frozen ESM-IF1 [24] pocket encoder embeds pocket, and a two-layer MLP projector maps these embeddings into the GenMol hidden space before molecular denoising. We supervised-tune GenMol-P on the CrossDocked2020 [17] training set. Following Section 5.2, we instantiate SGRPO as coupled-SGRPO for this discrete diffusion generator. Coupled-GRPO supplies the diffusion-compatible policy-gradient estimator and serves as the utility-only baseline, playing the role of standard GRPO for an autoregressive generator. We use $K = 24$, $M = 8$, and thus 192 rollouts per pocket for both GRPO and SGRPO. The rollout utility augments the QED–SA utility with a target-dependent docking term: $u(x, \mathcal{C}) = \alpha_{\mathrm{QED}}\mathrm{QED}(x) + \alpha_{\mathrm{SA}} s_{\mathrm{SA}}(x) + \alpha_{\mathrm{dock}} s_{\mathrm{dock}}(x, \mathcal{C})$, where $s_{\mathrm{dock}}(x, \mathcal{C}) = \mathrm{clip}(-a_{\mathrm{Vina}}(x, \mathcal{C})/10, 0, 1)$ maps the raw AutoDock Vina [53] score $a_{\mathrm{Vina}}(x, \mathcal{C})$ to a high-is-better score in $[0, 1]$. Unless otherwise noted,

we set $\alpha_{\mathrm{QED}} = 0.3$, $\alpha_{\mathrm{SA}} = 0.2$, and $\alpha_{\mathrm{dock}} = 0.5$, giving docking the largest weight because pocket compatibility is the primary task objective, while QED and SA act as chemistry-oriented regularizers. We do not retune these coefficients per method, ensuring a shared utility definition across comparisons. We compare coupled-SGRPO against the original GenMol-P model, coupled-GRPO, DMB-GRPO, and ER-GRPO under the same ten paired (ρ, τ) settings used in Section 5.2, which span conservative to exploratory decoding regimes, using pockets from the CrossDocked2020 test set. [‡] At each sweep point, each model generates 16 ligands for each of 100 held-out pockets, for 1600 ligand samples in total. This protocol provides stable estimates at a manageable docking cost. Utility metrics are averaged over the valid generated ligands at that point, while diversity is computed within each pocket's 16 ligands and then averaged over pockets. This measures target-conditional diversity rather than conflating diversity with variation across different pockets. Specifically, for the valid ligand collection $S_{\mathcal{C}}$ of pocket $\mathcal{C}$, $V(S_{\mathcal{C}}) = 1 - \frac{2}{|S_{\mathcal{C}}|(|S_{\mathcal{C}}|-1)} \sum_{i<j} \mathrm{Tan}(\phi(x_i), \phi(x_j))$, where ϕ is the Morgan fingerprint.

Result. Pocket-based design is the setting where SGRPO helps most: because optimizing docking to a fixed pocket tends to collapse generation onto a few high-scoring chemotypes, coupled-GRPO, DMB-GRPO, and ER-GRPO improve utility only by sacrificing within-pocket diversity, whereas coupled-SGRPO shifts the GenMol-P operating-point trajectory outward relative to baselines and retains markedly higher diversity at comparable utility, especially in the high-utility regime where collapse is strongest (Figure 2). This matches the conditional nature of the task: multiple chemically distinct ligands can be similarly plausible for the same pocket, but utility-only relative updates over-amplify small score differences and reinforce redundancy, while SGRPO explicitly rewards high-utility samples that also contribute marginal diversity. Accordingly, coupled-SGRPO achieves the best frontier-level performance in Table 1, with the largest HV and the smallest DIP and R2, and its advantage is more pronounced than in *de novo* molecule generation. Figure 2 shows all evaluated decoding settings for completeness, so some adjacent baseline points may improve in both utility and diversity before the true trade-off boundary is reached.

5.4 *De novo* Protein Design

Setup. We evaluate unconditional *de novo* protein design with ProGen2 [37], an autoregressive amino-acid language model, and apply SGRPO via GRPO [46]. The sequence-level utility combines naturalness, foldability, stability, and developability: $u(y) = \alpha_{\mathrm{nat}} r_{\mathrm{nat}}(y) + \alpha_{\mathrm{fold}} r_{\mathrm{fold}}(y) + \alpha_{\mathrm{stab}} r_{\mathrm{stab}}(y) + \alpha_{\mathrm{dev}} r_{\mathrm{dev}}(y)$. Here, the four terms are computed from ESM2, ESMFold [33], TemBERTure [44], and ProteinSol-based scorers [23], respectively, with weights $\alpha_{\mathrm{nat}} = 0.25$, $\alpha_{\mathrm{fold}} = 0.30$, $\alpha_{\mathrm{stab}} = 0.20$, and $\alpha_{\mathrm{dev}} = 0.25$. Diversity is measured at the set level using normalized Levenshtein similarity over valid sequences: $V(S) = 1 - \frac{2}{|S|(|S|-1)} \sum_{i<j} s_{ij}^{\mathrm{edit}}$. SGRPO uses $K = 12$ and $M = 8$, matching the 96 rollouts per condition in GRPO. We compare SGRPO against the original ProGen2 model, GRPO, Memory-assisted RL-based GRPO, DMB-GRPO, and ER-GRPO under the same temperature sweep $\tau \in \{0.1, 0.2, \ldots, 1.0, 1.1, 1.2\}$. At each temperature, we sample 512 sequences per model and evaluate both utility and diversity over valid outputs.

Result. SGRPO achieves the best utility–diversity trade-off in *de novo* protein design. In Figure 2, GRPO loses substantial diversity, Memory-Assisted GRPO loses still more, and DMB-GRPO exhibits near-zero diversity across the sweep. ER-GRPO preserves more diversity than GRPO but remains below SGRPO in the high-utility region. SGRPO reaches high utility while retaining diversity much closer to the pretrained model. This pattern is reflected consistently in Table 1, where SGRPO attains the best HV, DIP, and R2.

6 Analysis

6.1 Ablation Study

We isolate two components of SGRPO on the GenMol-based *de novo* small-molecule design task: the supergroup diversity reward and the leave-one-out diversity contribution mechanism.

[‡]We exclude Memory-assisted coupled-GRPO here: global memory mixes unrelated pockets, while pocket-specific memories require separate optimization and extra compute.

As shown in Figure 3, removing the diversity reward gives the innermost utility-diversity curve. Adding the supergroup diversity reward without leave-one-out credit assignment moves the curve outward, indicating that group-level diversity pressure itself improves the trade-off. Full SGRPO further dominates this variant, showing that leave-one-out credit assignment is important for redistributing the group diversity reward to the rollouts that actually contribute to set-level diversity.

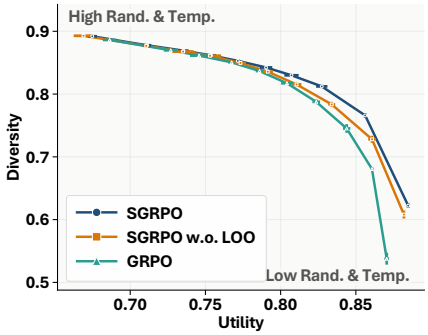

Figure 3: Ablation study on GenMol-based *de novo* small-molecule generation. Each point reports the mean over five independent sweeps, and error bars indicate 95% confidence intervals for utility and diversity. Removing the diversity term yields coupled-GRPO, while removing leave-one-out group credit weakens set-aware credit assignment.

6.2 Training Dynamics of Generated Distributions

To understand why the three training algorithms lead to different final utility-diversity trade-offs, we use *de novo* protein design as a diagnostic setting and visualize how their generated sequence distributions move during training. For GRPO, Memory-assisted GRPO, and SGRPO, we sample 512 sequences from the shared original ProGen2 model, the checkpoint after 20 optimization steps, and the final checkpoint after 100 steps. We then pool all sequences and compute a shared two-dimensional UMAP embedding using distances derived from normalized Levenshtein similarity, so that movements are comparable across methods and checkpoints.

Figure 4 suggests two distinct training dynamics. After 20 steps, GRPO and Memory-assisted GRPO each move into a relatively concentrated region of the embedding, whereas SGRPO spreads across multiple clusters, including the regions explored by the other two methods. This early behavior indicates that supergroup-relative diversity pressure promotes broader exploration. From 20 to 100 steps, GRPO contracts into an even smaller region. Memory-assisted GRPO instead drifts toward a distant, narrow region: its memory penalty discourages revisiting high-density regions, which is not equivalent to directly optimizing set diversity and can drive distributional drift. SGRPO refines within the explored regions while retaining multiple clusters, explaining why it reaches the high-utility regime while preserving substantially more sequence diversity.

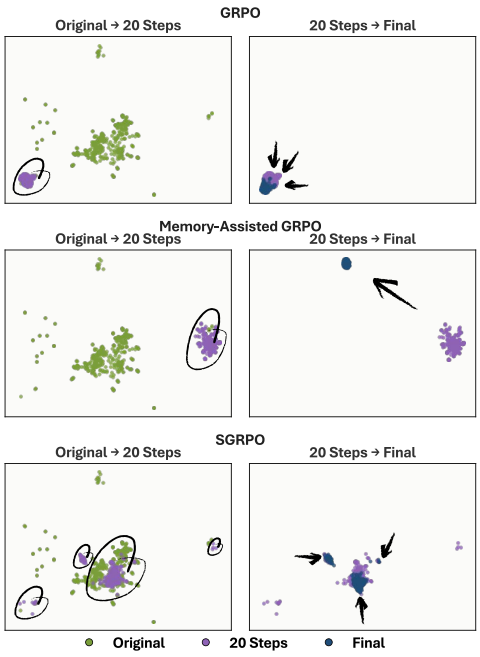

Figure 4: Distribution dynamics during ProGen2 post-training. SGRPO explores multiple clusters early and preserves them, whereas GRPO contracts and Memory-assisted GRPO drifts toward a narrow distant region.

6.3 Robustness to Diversity-Estimator Efficiency and Reward Weighting

We study two practical sensitivities of SGRPO on the GenMol-based *de novo* small-molecule design task: the efficiency of the group-diversity estimator and the choice of group-reward weight λ. In both cases, we train one model per setting, evaluate each model using the same decoding sweep as in Section 5.2, and measure the HV of the resulting utility-diversity frontier using a common lower reference point computed from all operating points in this experiment. To test tolerance to diversity-estimator inefficiency, we hold the total supergroup size fixed while varying the group size $K \in \{1, 4, 16, 32, 64\}$, where $K = 1$ recovers GRPO. As shown in the left panel of Figure 5, SGRPO already outperforms GRPO in the near-minimal setting $K = 4$, indicating that useful group-diversity estimates do not require a large multiplicative increase in rollouts. Larger groups generally lead to better frontier-level performance, suggesting that more efficient diversity estimates help but are not necessary for SGRPO to be effective.

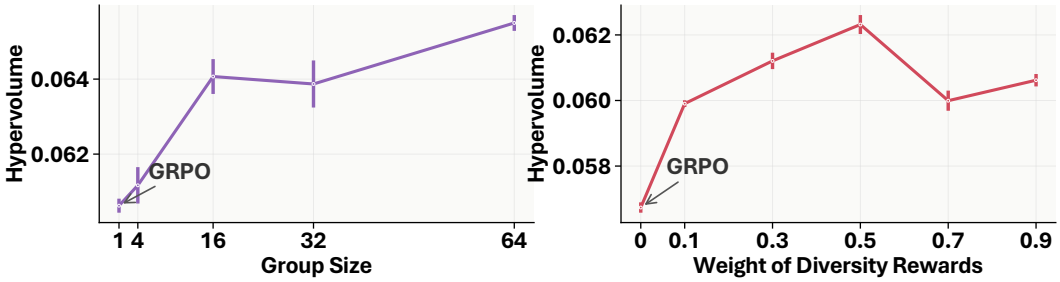

Figure 5: Analysis of diversity-estimator efficiency and group-reward weighting on the GenMol-based *de novo* small-molecule design task. Each point reports the mean HV over five independent sweeps, and error bars indicate 95% confidence intervals. Left: under a fixed total supergroup size, SGRPO already outperforms GRPO with small groups, indicating that useful group-diversity estimates do not require a large rollout multiplier. Right: SGRPO robustly improves HV over GRPO across all tested nonzero λ, with the strongest frontier expansion at $\lambda = 0.5$.

We next vary the group-reward coefficient $\lambda \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$, which controls the trade-off between rollout-level utility and redistributed group-level diversity in the composed reward. The right panel of Figure 5 shows that every tested nonzero value of λ improves HV over the GRPO baseline $\lambda = 0$, demonstrating that the benefit of supergroup-relative diversity pressure is robust to reward weighting. Performance peaks at $\lambda = 0.5$, suggesting that the strongest frontier expansion is achieved by balancing molecule-level utility with set-level diversity. Overall, Figure 5 shows that SGRPO is robust to both imperfect diversity estimation and moderate variation in reward composition.

7 Conclusion

We presented SGRPO, a GRPO-style framework for directly combining rollout-level utility with set-level sample diversity in biomolecular post-training. The central idea is simple: sample multiple candidate sets under the same condition, score their diversity, compare them within the supergroup, and redistribute the resulting set reward to individual rollouts through leave-one-out diversity contributions. Across *de novo* small-molecule design, pocket-based small-molecule design, and *de novo* protein design, SGRPO improves the utility-diversity Pareto frontier over pretrained generators and RL baselines. This benefit comes with additional computational overhead, which we mitigate by reusing pairwise similarities for diversity rewards and leave-one-out credit assignment (Appendix G). These results suggest that directly rewarding diverse generated sets is a practical way to expand the operating points available to biomolecular generation models.

Acknowledgments and Disclosure of Funding

This work is partially supported by funding and computing resource provided by HKU-Shanghai Intelligent Computing Research Center. This work was also supported by the General Research Fund of the Research Grants Council of Hong Kong under Grant No. 17200725 and the JC STEM Lab funded by The Hong Kong Jockey Club Charities Trust.

Competing interests. The authors declare no competing interests.

References

[1] Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

[2] Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

[3] Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. Molgpt: molecular generation using a transformer-decoder model. *Journal of chemical information and modeling*, 62(9): 2064–2076, 2021.

[4] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):20, 2015.

[5] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.

[6] Esben Jannik Bjerrum, Christian Margreitter, Thomas Blaschke, Simona Kolarova, and Raquel López-Ríos de Castro. Faster and more diverse de novo molecular optimization with double-loop reinforcement learning using augmented smiles. *Journal of Computer-Aided Molecular Design*, 37(8):373–394, 2023.

[7] Thomas Blaschke, Ola Engkvist, Jürgen Bajorath, and Hongming Chen. Memory-assisted reinforcement learning for diverse molecular de novo design. *Journal of cheminformatics*, 12 (1):68, 2020.

[8] David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

[9] Egbert Castro, Abhinav Godavarthi, Julian Rubinfien, Kevin Givechian, Dhananjay Bhaskar, and Smita Krishnaswamy. Transformer-based protein generation with regularized latent space optimization. *Nature Machine Intelligence*, 4(10):840–851, 2022.

[10] Mohamed-Amine Chadi, Hajar Mousannif, and Ahmed Aamouche. Curiosity as a self-supervised method to improve exploration in de novo drug design. In *2023 International Conference on Information Technology Research and Innovation (ICITRI)*, pages 151–156. IEEE, 2023.

[11] Xiwei Cheng, Xiangxin Zhou, Yuwei Yang, Yu Bao, and Quanquan Gu. Decomposed direct preference optimization for structure-based drug design. *arXiv preprint arXiv:2407.13981*, 2024.

[12] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[13] Yasha Ektefaie, Olivia Viessmann, Siddharth Narayanan, Drew Dresser, J Mark Kim, and Armen Mkrtchyan. Reinforcement learning on structure-conditioned categorical diffusion for protein inverse folding. *arXiv preprint arXiv:2410.17173*, 2024.

[14] Yasha Ektefaie, Olivia Viessmann, Siddharth Narayanan, Drew Dresser, J Mark Kim, and Armen Mkrtchyan. Reinforcement learning on structure-conditioned categorical diffusion for protein inverse folding. *arXiv preprint arXiv:2410.17173*, 2024.

[15] Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8, 2009.

[16] Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. Libinvent: reaction-based generative scaffold decoration for in silico library design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 2021.

[17] Paul G Francoeur, Tomohide Masuda, Jocelyn Sunseri, Andrew Jia, Richard B Iovanisci, Ian Snyder, and David R Koes. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *Journal of chemical information and modeling*, 60(9):4200–4215, 2020.

[18] Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. Searching for high-value molecules using reinforcement learning and transformers. *arXiv preprint arXiv:2310.02902*, 2023.

[19] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[20] Shansan Gong, Ruixiang Zhang, Huangjie Zheng, Jiatao Gu, Navdeep Jaitly, Lingpeng Kong, and Yizhe Zhang. Diffucoder: Understanding and improving masked diffusion models for code generation. *arXiv preprint arXiv:2506.20639*, 2025.

[21] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical science*, 11(2):577–586, 2020.

[22] Hampus Gummesson Svensson, Christian Tyrchan, Ola Engkvist, and Morteza Haghir Chehreghani. Utilizing reinforcement learning for de novo drug design. *Machine Learning*, 113(7):4811–4843, 2024.

[23] Max Hebditch, M Alejandro Carballo-Amador, Spyros Charonis, Robin Curtis, and Jim Warwicker. Protein–sol: a web tool for predicting protein solubility from sequence. *Bioinformatics*, 33(19):3098–3100, 2017.

[24] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[25] Xiuyuan Hu, Guoqing Liu, Quanming Yao, Yang Zhao, and Hao Zhang. Hamiltonian diversity: effectively measuring molecular diversity by shortest hamiltonian circuits. *Journal of Cheminformatics*, 16(1):94, 2024.

[26] Hyosoon Jang, Yunhui Jang, Jaehyung Kim, and Sungsoo Ahn. Can llms generate diverse molecules? towards alignment with structural diversity. *arXiv preprint arXiv:2410.03138*, 2024.

[27] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Multi-objective molecule generation using interpretable substructures. In *International conference on machine learning*, pages 4849–4859. PMLR, 2020.

[28] Alexia Jolicoeur-Martineau, Aristide Baratin, Kisoo Kwon, Boris Knyazev, and Yan Zhang. Any-property-conditional molecule generation with self-criticism using spanning trees. *arXiv preprint arXiv:2407.09357*, 2024.

[29] Panagiotis-Christos Kotsias, Josep Arús-Pous, Hongming Chen, Ola Engkvist, Christian Tyrchan, and Esben Jannik Bjerrum. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence*, 2(5):254–265, 2020.

[30] Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Paliwal, Weili Nie, and Arash Vahdat. Genmol: A drug discovery generalist with discrete diffusion. *arXiv preprint arXiv:2501.06158*, 2025.

[31] Yanting Li, Zhuoyang Jiang, Enyan Dai, Lei Wang, Wen-Cai Ye, and Li Liu. Cagenmol: Condition-aware diffusion language model for goal-directed molecular generation. *arXiv preprint arXiv:2604.11483*, 2026.

[32] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 10(1):31, 2018.

[33] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[34] Xuhan Liu, Kai Ye, Herman WT Van Vlijmen, Adriaan P IJzerman, and Gerard JP Van Westen. An exploration strategy improves the diversity of de novo ligands using deep reinforcement learning: a case for the adenosine a2a receptor. *Journal of cheminformatics*, 11(1):35, 2019.

[35] Xuhan Liu, Kai Ye, Herman WT van Vlijmen, Adriaan P IJzerman, and Gerard JP van Westen. Drugex v3: scaffold-constrained drug design with graph transformer-based reinforcement learning. *Journal of Cheminformatics*, 15(1):24, 2023.

[36] Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, 2024.

[37] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

[38] Emmanuel Noutahi, Cristian Gabellini, Michael Craig, Jonathan SC Lim, and Prudencio Tossou. Gotta be safe: a new framework for molecular design. *Digital Discovery*, 3(4):796–804, 2024.

[39] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9(1):48, 2017.

[40] Jinyeong Park, Jaegyoon Ahn, Jonghwan Choi, and Jibum Kim. Mol-air: Molecular reinforcement learning with adaptive intrinsic rewards for goal-directed molecular generation. *Journal of Chemical Information and Modeling*, 65(5):2283–2296, 2025.

[41] Ryan Park, Darren J Hsu, C Brian Roland, Maria Korshunova, Chen Tessler, Shie Mannor, Olivia Viessmann, and Bruno Trentini. Improving inverse folding for peptide design with diversity-regularized direct preference optimization. *arXiv preprint arXiv:2410.19471*, 2024.

[42] Tiago Pereira, Maryam Abbasi, Bernardete Ribeiro, and Joel P Arrais. Diversity oriented deep reinforcement learning for targeted molecule generation. *Journal of cheminformatics*, 13(1):21, 2021.

[43] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

[44] Chiara Rodella, Symela Lazaridi, and Thomas Lemmin. Temberture: advancing protein thermostability prediction with deep learning and attention mechanisms. *Bioinformatics Advances*, 4(1):vbae103, 2024.

[45] Nicholas T Runcie and Antonia SJS Mey. Silvr: guided diffusion for molecule generation. *Journal of chemical information and modeling*, 63(19):5996–6005, 2023.

[46] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[47] Tristan SW Stevens, Oisín Nolan, Jean-Luc Robert, and Ruud JG Van Sloun. Sequential posterior sampling with diffusion models. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[48] Hampus Gummesson Svensson, Christian Tyrchan, Ola Engkvist, and Morteza Haghir Chehreghani. Diversity-aware reinforcement learning for de novo drug design. *arXiv preprint arXiv:2410.10431*, 2024.

[49] Hampus Gummesson Svensson, Ola Engkvist, Jon Paul Janet, Christian Tyrchan, and Morteza Haghir Chehreghani. Diverse mini-batch selection in reinforcement learning for efficient chemical exploration in de novo drug design. *arXiv preprint arXiv:2506.21158*, 2025.

[50] Xiangru Tang, Xinwu Ye, Fang Wu, Yimeng Liu, Anna Su, Antonia Panescu, Guanlue Li, Daniel Shao, Dong Xu, and Mark Gerstein. Bc-design: A biochemistry-aware framework for inverse protein design. *bioRxiv*, 2025. doi: 10.1101/2024.10.28.620755. URL `https://www.biorxiv.org/content/early/2025/11/24/2024.10.28.620755`.

[51] Morgan Thomas, Noel M O'Boyle, Andreas Bender, and Chris De Graaf. Augmented hill-climb increases reinforcement learning efficiency for language-based de novo molecule generation. *Journal of cheminformatics*, 14(1):68, 2022.

[52] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272, 2020.

[53] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.

[54] Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. Proteinzero: Self-improving protein generation via online reinforcement learning. *arXiv preprint arXiv:2506.07459*, 2025.

[55] Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via direct preference optimization. *bioRxiv*, pages 2024–05, 2024.

[56] Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M Oltrogge, David F Savage, and Jennifer Listgarten. Proteinguide: On-the-fly property guidance for protein sequence generative models. *arXiv preprint arXiv:2505.04823*, 2025.

[57] Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation. *Advances in Neural Information Processing Systems*, 34:7924–7936, 2021.

[58] Xinwu Ye, Yicheng Mao, Jia Zhang, Yimeng Liu, Li Hao, Fang Wu, Zhiwei Li, Yuxuan Liao, Zehong Wang, Yingcheng Wu, et al. Latentchem: From textual cot to latent thinking in chemical reasoning. *arXiv preprint arXiv:2602.07075*, 2026.

[59] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *Advances in neural information processing systems*, 31, 2018.

[60] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):10752, 2019.

[61] Xingzheng Zhu, Zhihong Zhao, and Fei Zhu. Scaffold-driven molecular generation via reinforced rnn with centroid distance evaluation. *Expert Systems with Applications*, 292:128606, 2025.

A Full Training Procedure of SGRPO

This appendix provides the full training procedure of Supergroup Relative Policy Optimization (SGRPO), corresponding to Section 4 in the main text. For each condition, SGRPO samples a same-condition supergroup, computes rollout-level utility and group-level diversity, redistributes the diversity signal to individual rollouts in a set-aware manner, forms supergroup-relative advantages, and updates the policy with a PPO-style objective regularized toward a reference policy.

Algorithm 1: Supergroup Relative Policy Optimization (SGRPO)

Input: initial policy π_θ, reference policy π_{ref}, condition batch $\{\mathcal{C}_b\}_{b=1}^{B}$, groups per condition M, rollouts per group K, diversity weight λ, PPO clip parameter ϵ, KL coefficient β, contribution temperature τ_c
for $t = 1$ to T **do**
  Set $\theta_{\mathrm{old}} \leftarrow \theta$
  for $b = 1$ to B **do**
    Sample a same-condition supergroup $\mathcal{S}_b = \{G_{b,1}, \ldots, G_{b,M}\}$, where each group is $G_{b,m} = \{x_{b,m,1}, \ldots, x_{b,m,K}\}$ with $x_{b,m,i} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid \mathcal{C}_b)$
    Compute rollout utilities $r_{b,m,i} = r(x_{b,m,i}, \mathcal{C}_b)$ for all m, i
    Compute group diversities $R_{b,m} = D(G_{b,m})$ for all m
    Compute the supergroup mean diversity $\bar{R}_b = \frac{1}{M}\sum_{h=1}^{M} R_{b,h}$
    Compute group-relative diversity signals $A^{\mathrm{grp}}_{b,m} = \frac{M}{M-1}(R_{b,m} - \bar{R}_b)$ for all m
    for $m = 1$ to M **do**
      For each rollout $x_{b,m,i}$, compute leave-one-out contribution $c_{b,m,i} = D(G_{b,m}) - D(G_{b,m} \setminus \{x_{b,m,i}\})$
      Compute within-group standardized contributions $z_{b,m,i} = \dfrac{c_{b,m,i} - \bar{c}_{b,m}}{\sigma(c_{b,m,\cdot}) + \zeta}$ for all i
      Form sign-aware redistribution weights $w^{\pm}_{b,m,i} = K\,\dfrac{\exp(\pm z_{b,m,i}/\tau_c)}{\sum_{j=1}^{K}\exp(\pm z_{b,m,j}/\tau_c)}$ for all i
      Assign redistributed diversity rewards $\widetilde{R}_{b,m,i} = R_{b,m} + [A^{\mathrm{grp}}_{b,m}]_+(w^{+}_{b,m,i} - 1) - [-A^{\mathrm{grp}}_{b,m}]_+(w^{-}_{b,m,i} - 1)$ for all i
    end for
    Form combined rollout rewards $\widehat{r}_{b,m,i} = (1-\lambda) r_{b,m,i} + \lambda \widetilde{R}_{b,m,i}$ for all m, i
    Compute the supergroup mean reward $\bar{r}_{\mathcal{S}_b} = \frac{1}{MK}\sum_{m=1}^{M}\sum_{i=1}^{K}\widehat{r}_{b,m,i}$
    Compute supergroup-relative advantages $A_{b,m,i} = \frac{MK}{MK-1}\left(\widehat{r}_{b,m,i} - \bar{r}_{\mathcal{S}_b}\right)$ for all m, i
  end for
  Update θ by minimizing the PPO-style objective with KL regularization to π_{ref} using all collected tuples $(\mathcal{C}_b, x_{b,m,i}, A_{b,m,i})$
end for
Return: trained policy π_θ

Implementation notes. All relative comparisons in SGRPO are performed within the same-condition supergroup. In particular, group-relative diversity signals are centered only across the M groups sampled for the same condition, and supergroup-relative advantages are centered only across the corresponding MK rollouts. This avoids confounding policy quality with variation in condition difficulty. In practice, the diversity metric $D(\cdot)$ can be instantiated according to the domain, and the utility reward $r(x, \mathcal{C})$ can be any task-specific scalar oracle.

B Population Diversity and Finite-Group Estimation

We first establish that finite-group diversity is an unbiased estimator of policy-level expected diversity for normalized fixed-order U-statistics, with partition consistency as a corresponding finite-sample property. We then study concentration for bounded pairwise diversity. These are properties of the diversity estimator. The effect of candidate-wise reward redistribution on policy optimization is treated separately in Appendix C.

Setup. Samples below are indexed collections, retaining repeated generated candidates. Partitions act on sample indices. We assume $K \geq 2$ for pairwise diversity, $M \geq 1$ for the estimation results, and a symmetric similarity function. The SGRPO group-relative baseline additionally requires $M \geq 2$. Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be a same-condition sample of size $N = MK$, where M is the number of

groups and K is the group size. A *balanced partition* Π divides $\mathcal{X}$ into M disjoint groups of size K:

$$\Pi = \{G_1, \ldots, G_M\}, \qquad |G_m| = K, \qquad \bigsqcup_{m=1}^{M} G_m = \mathcal{X}.$$

For a set $A = \{a_1, \ldots, a_L\}$, define the normalized pairwise diversity

$$D_L(A) = \frac{2}{L(L-1)} \sum_{1 \le i < j \le L} d(a_i, a_j), \qquad d(x, x') := 1 - s(x, x'), \tag{6}$$

where $s(x, x') \in [0, 1]$ is a biomolecular similarity function. Equivalently, $D_L(A)$ is the average pairwise dissimilarity under $d \in [0, 1]$.

Given a balanced partition $\Pi = \{G_1, \ldots, G_M\}$, define the average small-group diversity

$$\overline{D}_{M,K}(\mathcal{X}, \Pi) = \frac{1}{M} \sum_{m=1}^{M} D_K(G_m). \tag{7}$$

B.1 Population Unbiasedness and Partition Consistency

Proposition B.1 (Population unbiasedness for fixed-order diversity)**.** *Let h be a symmetric, integrable kernel of fixed order $q \le K$. For an indexed sample of size $L \ge q$, define*

$$D_L^{(q)}(X_{1:L}) = \binom{L}{q}^{-1} \sum_{\substack{I \subseteq \{1,\ldots,L\} \\ |I| = q}} h(X_I), \qquad \mu_q(\theta, \mathcal{C}) = \mathbb{E}_{X_1, \ldots, X_q \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot|\mathcal{C})}[h(X_1, \ldots, X_q)]. \tag{8}$$

For $N = MK$ i.i.d. samples under a fixed condition $\mathcal{C}$, partitioned independently and uniformly into M groups of size K, every group satisfies

$$\mathbb{E}[D_K^{(q)}(G_m) \mid \mathcal{C}] = \mathbb{E}[D_N^{(q)}(X_{1:N}) \mid \mathcal{C}] = \mu_q(\theta, \mathcal{C}). \tag{9}$$

For any fixed realization $x_{1:N}$, the corresponding partition identities are

$$\mathbb{E}_\Pi[D_K^{(q)}(G_m) \mid x_{1:N}] = \mathbb{E}_\Pi\left[\frac{1}{M} \sum_{h=1}^{M} D_K^{(q)}(G_h) \,\middle|\, x_{1:N}\right] = D_N^{(q)}(x_{1:N}). \tag{10}$$

Proof. Each q-tuple of distinct sample indices has the same expected kernel value $\mu_q(\theta, \mathcal{C})$, which proves population unbiasedness by linearity of expectation. For a fixed realization, a given q-subset lies in a particular group with probability $\binom{K}{q}/\binom{N}{q}$, and in any one of the M groups with probability $M\binom{K}{q}/\binom{N}{q}$. These probabilities cancel the within-group normalization and, for the aggregate identity, the factor $1/M$. Equivalently, exchangeability of the groups gives the per-group identity from the aggregate one. □

For pairwise diversity, $q = 2$ and $h(x, x') = 1 - s(x, x')$. Thus the group reward is unbiased for the expected diversity of independent policy samples, not merely for a particular sampled supergroup. Sampling a supergroup and partitioning it independently is distributionally equivalent to sampling the groups directly. The original pairwise partition calculation is given below as a special case.

Proposition B.2 (Partition consistency of normalized pairwise diversity)**.** *Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be any fixed set of size $N = MK$. If Π is drawn uniformly from all balanced partitions of $\mathcal{X}$ into M groups of size K, then*

$$\mathbb{E}_\Pi\big[\overline{D}_{M,K}(\mathcal{X}, \Pi)\big] = D_N(\mathcal{X}). \tag{11}$$

Proof. For brevity, let $d_{ij} := d(x_i, x_j)$. For a random balanced partition Π, define the indicator

$$I_{ij}(\Pi) = \mathbf{1}\{x_i \text{ and } x_j \text{ belong to the same group under } \Pi\}.$$

Then

$$\overline{D}_{M,K}(\mathcal{X},\Pi) = \frac{1}{M}\sum_{m=1}^{M}\frac{2}{K(K-1)}\sum_{\substack{i<j\\ x_i,x_j\in G_m}} d_{ij}$$
$$= \frac{1}{M}\cdot\frac{2}{K(K-1)}\sum_{1\le i<j\le N} I_{ij}(\Pi)\, d_{ij}. \tag{12}$$

Taking expectation over Π gives

$$\mathbb{E}_\Pi\big[\overline{D}_{M,K}(\mathcal{X},\Pi)\big] = \frac{1}{M}\cdot\frac{2}{K(K-1)}\sum_{1\le i<j\le N}\mathbb{E}_\Pi[I_{ij}(\Pi)]\, d_{ij}. \tag{13}$$

By symmetry of the uniform balanced partition, for any fixed pair (i,j),

$$\mathbb{P}_\Pi(I_{ij}(\Pi)=1) = \frac{K-1}{N-1}, \tag{14}$$

because once x_i is assigned to a group, exactly $K-1$ of the remaining $N-1$ positions lie in that same group. Substituting Eq. (14) into Eq. (13),

$$\mathbb{E}_\Pi\big[\overline{D}_{M,K}(\mathcal{X},\Pi)\big] = \frac{1}{M}\cdot\frac{2}{K(K-1)}\cdot\frac{K-1}{N-1}\sum_{1\le i<j\le N} d_{ij}$$
$$= \frac{2}{MK(N-1)}\sum_{1\le i<j\le N} d_{ij}$$
$$= \frac{2}{N(N-1)}\sum_{1\le i<j\le N} d_{ij} = D_N(\mathcal{X}), \tag{15}$$

where we used $N = MK$ in the last step. □

Combining the finite-sample identity with policy sampling gives $\mathbb{E}_{\mathcal{X},\Pi}[D_K(G_m) \mid \mathcal{C}] = \mathbb{E}_{\mathcal{X}}[D_N(\mathcal{X}) \mid \mathcal{C}] = \mu_2(\theta,\mathcal{C})$. This identifies the population quantity estimated by each group reward.

B.2 Concentration Around Full-Sample Diversity

We now show that the average small-group diversity not only matches the full-sample diversity in expectation, but also concentrates around it when the groups are sufficiently large.

Proposition B.3 (Concentration of average small-group diversity)**.** *Fix a condition c, and let $X_1,\dots,X_N \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid c)$ with $N = MK$. Let $\mathcal{X} = \{X_1,\dots,X_N\}$, and let Π be a uniformly random balanced partition of $\mathcal{X}$ into M groups of size K, independent of the samples. Define*

$$\mu(c) := \mathbb{E}_{X,X'\sim\pi_\theta(\cdot|c)}\big[d(X,X')\big]. \tag{16}$$

Then, for any $\varepsilon > 0$,

$$\mathbb{P}\big(\big|\overline{D}_{M,K}(\mathcal{X},\Pi) - D_N(\mathcal{X})\big| \ge \varepsilon\big) \le 4\exp\left(-\frac{1}{2}M\left\lfloor\frac{K}{2}\right\rfloor\varepsilon^2\right). \tag{17}$$

Consequently, for any $\delta \in (0,1)$,

$$\big|\overline{D}_{M,K}(\mathcal{X},\Pi) - D_N(\mathcal{X})\big| \le \varepsilon \tag{18}$$

with probability at least $1-\delta$ whenever

$$M\left\lfloor\frac{K}{2}\right\rfloor \ge \frac{2\log(4/\delta)}{\varepsilon^2}. \tag{19}$$

Equivalently, a sufficient lower bound on the group size is

$$K \ge 2\left\lceil\frac{2\log(4/\delta)}{M\varepsilon^2}\right\rceil. \tag{20}$$

Proof. Both $D_N(\mathcal{X})$ and $\overline{D}_{M,K}(\mathcal{X}, \Pi)$ estimate the same population quantity $\mu(c)$ defined in Eq. (16).

First, $D_N(\mathcal{X})$ is a bounded U-statistic of order two with kernel $d(\cdot, \cdot) \in [0, 1]$. By Hoeffding's concentration inequality for bounded U-statistics,

$$\mathbb{P}(|D_N(\mathcal{X}) - \mu(c)| \geq t) \leq 2 \exp\left(-2 \left\lfloor \frac{N}{2} \right\rfloor t^2\right). \tag{21}$$

Next, consider $\overline{D}_{M,K}(\mathcal{X}, \Pi)$. Because the samples are i.i.d. and the partition is independent of the samples, its distribution is the same as that obtained by first drawing MK i.i.d. samples and then forming M disjoint blocks of size K. Hence

$$\overline{D}_{M,K}(\mathcal{X}, \Pi) = \frac{1}{M} \sum_{m=1}^{M} U_m,$$

where $U_1, \ldots, U_M$ are independent copies of a bounded order-two U-statistic based on K i.i.d. samples with mean $\mu(c)$. For each U_m, Hoeffding's inequality yields

$$\mathbb{P}(|U_m - \mu(c)| \geq t) \leq 2 \exp\left(-2 \left\lfloor \frac{K}{2} \right\rfloor t^2\right).$$

To retain the stated constants when averaging groups, let $n_K = \lfloor K/2 \rfloor$. A pairwise U-statistic is the average, over permutations of its sample indices, of averages of n_K disjoint pair terms. Each such average consists of independent variables in $[0, 1]$. Jensen's inequality and Hoeffding's lemma therefore give $\mathbb{E} \exp\{a(U_m - \mu(c))\} \leq \exp\{a^2/(8n_K)\}$. Independence across groups implies $\mathbb{E} \exp\{a(\frac{1}{M} \sum_m U_m - \mu(c))\} \leq \exp\{a^2/(8Mn_K)\}$. Applying the Chernoff bound to both tails yields

$$\mathbb{P}\big(\big|\overline{D}_{M,K}(\mathcal{X}, \Pi) - \mu(c)\big| \geq t\big) \leq 2 \exp\left(-2M \left\lfloor \frac{K}{2} \right\rfloor t^2\right). \tag{22}$$

Applying the triangle inequality,

$$\big|\overline{D}_{M,K}(\mathcal{X}, \Pi) - D_N(\mathcal{X})\big| \leq \big|\overline{D}_{M,K}(\mathcal{X}, \Pi) - \mu(c)\big| + |D_N(\mathcal{X}) - \mu(c)|. \tag{23}$$

Setting $t = \varepsilon/2$ in Eqs. (21) and (22), and then using a union bound, we obtain

$$\begin{aligned} &\mathbb{P}\big(\big|\overline{D}_{M,K}(\mathcal{X}, \Pi) - D_N(\mathcal{X})\big| \geq \varepsilon\big) \\ &\leq 2 \exp\left(-\frac{1}{2} M \left\lfloor \frac{K}{2} \right\rfloor \varepsilon^2\right) + 2 \exp\left(-\frac{1}{2} \left\lfloor \frac{N}{2} \right\rfloor \varepsilon^2\right). \end{aligned} \tag{24}$$

Since $N = MK$ and $\lfloor N/2 \rfloor \geq M \lfloor K/2 \rfloor$, the second term is no larger than the first, which gives Eq. (17). Finally, solving

$$4 \exp\left(-\frac{1}{2} M \left\lfloor \frac{K}{2} \right\rfloor \varepsilon^2\right) \leq \delta$$

yields the sufficient condition in Eq. (19), and Eq. (20) follows immediately. □

The concentration statement concerns the average group estimator and probability over both policy samples and the independent partition. At fixed M, its bound improves with K. At fixed total rollout count $N = MK$, $M \lfloor K/2 \rfloor$ is approximately $N/2$, so this bound alone does not explain the group-size effect in the fixed-budget ablation. A single group's diversity has the K-dependent bound above.

Scope beyond pairwise diversity. The fixed-order result applies to normalized U-statistics, but empirical entropy, coverage, and relative scaffold diversity are generally outside this class. Their expectations can depend on group size, so exact alignment with full-sample diversity is not implied. SGRPO can nevertheless use a computable set score whenever both the group and its leave-one-out subsets can be scored. This algorithmic compatibility is distinct from the metric-specific unbiasedness guarantee. Appendix F.5 evaluates a non-pairwise example.

C Reward Conservation and the Optimization Objective

C.1 Conservation Under Redistribution

Proposition C.1 (Conservation of the mean group reward)**.** *For the weights in Eq.* (2) *and redistributed rewards in Eq.* (3)*,*

$$\frac{1}{K}\sum_{i=1}^{K}\widetilde{R}_{m,i} = R_m = D(G_m), \qquad \sum_{i=1}^{K}\widetilde{R}_{m,i} = KR_m. \tag{25}$$

Proof. Both sign-aware weight vectors sum to K. Averaging Eq. (3) within a group gives

$$\frac{1}{K}\sum_{i}\widetilde{R}_{m,i} = R_m + [A_m^{\mathrm{grp}}]_+\left(\frac{1}{K}\sum_{i}w_{m,i}^{+} - 1\right) - [-A_m^{\mathrm{grp}}]_+\left(\frac{1}{K}\sum_{i}w_{m,i}^{-} - 1\right) = R_m. \tag{26}$$

Thus redistribution preserves the mean group reward; summing the rollout rewards yields K times that reward. □

C.2 Population Target and Credit-Shaped Update

Let the expected K-sample diversity under the policy be

$$V_K(\theta) = \mathbb{E}_{\mathcal{C}\sim p(\mathcal{C}),\, G\sim\pi_\theta^K(\cdot|\mathcal{C})}[D(G)], \qquad s_{m,i} = \nabla_\theta \log \pi_\theta(x_{m,i} \mid \mathcal{C}). \tag{27}$$

Here D has no explicit parameter dependence. We assume independent policy samples conditional on $\mathcal{C}$, $M \geq 2$, and the regularity needed to exchange differentiation and expectation, including parameter-independent support and integrable score-weighted rewards. These statements concern exact likelihood scores; the diffusion implementation uses the coupled likelihood estimator described in the experimental implementation details.

Conditional on $\mathcal{C}$, rewards from other groups are independent of the candidates in G_m, and $\mathbb{E}[s_{m,i} \mid \mathcal{C}] = 0$. The other-group mean is therefore a valid baseline, giving

$$\mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M}A_m^{\mathrm{grp}}\sum_{i=1}^{K}s_{m,i}\right] = \nabla_\theta V_K(\theta). \tag{28}$$

To see this, the score identity gives $\mathbb{E}[R_m \sum_i s_{m,i}] = \nabla_\theta V_K(\theta)$, while the cross-group baseline term has expectation zero.

To characterize the final update, write

$$\delta_{m,i} = A_m^{\mathrm{grp}}(w_{m,i}^{\sigma_m} - 1), \qquad \widetilde{R}_{m,i} = R_m + \delta_{m,i}, \qquad w_{m,i}^{\sigma_m} = \begin{cases} w_{m,i}^{+}, & A_m^{\mathrm{grp}} \geq 0, \\ w_{m,i}^{-}, & A_m^{\mathrm{grp}} < 0. \end{cases} \tag{29}$$

Conservation implies $\sum_i \delta_{m,i} = 0$, so the mean redistributed diversity over the supergroup is $\bar{R}$. The diversity-only part of Eq. (4), averaged over rollouts and multiplied by their scores, is

$$\widehat{g}_{\mathrm{div}} = \frac{1}{MK}\sum_{m,i}\frac{MK}{MK-1}(\widetilde{R}_{m,i} - \bar{R})s_{m,i} \tag{30}$$

$$= \frac{1}{MK-1}\sum_{m,i}(R_m - \bar{R})s_{m,i} + \frac{1}{MK-1}\sum_{m,i}\delta_{m,i}s_{m,i}. \tag{31}$$

Using $R_m - \bar{R} = (M-1)A_m^{\mathrm{grp}}/M$ and Eq. (28) yields

$$\mathbb{E}[\widehat{g}_{\mathrm{div}}] = \frac{M-1}{MK-1}\nabla_\theta V_K(\theta) + B_{\tau_c}(\theta), \qquad B_{\tau_c}(\theta) = \frac{1}{MK-1}\mathbb{E}\left[\sum_{m,i}A_m^{\mathrm{grp}}(w_{m,i}^{\sigma_m} - 1)s_{m,i}\right]. \tag{32}$$

The first term is a positive multiple of the exact expected-set-diversity gradient. The second is the candidate-level credit-shaping term: it emphasizes high-contribution candidates in above-average-diversity groups and low-contribution candidates in below-average-diversity groups with the appropriate sign. This term need not vanish, so the complete redistributed update is not generally unbiased for $\nabla_\theta V_K(\theta)$. With uniform redistribution weights, it vanishes and only the scaled exact gradient component remains.

C.3 The Local Policy-Optimization Surrogate

At each iteration, fix the behavior policy $\mu = \pi_{\theta_{\text{old}}}$, sample the supergroup $\mathcal{S} \sim \mu^{MK}(\cdot \mid \mathcal{C})$, and compute the rewards and advantages using Eqs. (3)–(4). These quantities are held fixed during the policy update. The maximization form of Eq. (5) is

$$\mathcal{J}_{\text{local}}(\theta;\mu) = \mathbb{E}_{\mathcal{C},\mathcal{S}}\left[\frac{1}{MK}\sum_{m,i}\min\{\rho_{m,i}A_{m,i}, \text{clip}(\rho_{m,i}, 1-\epsilon, 1+\epsilon)A_{m,i}\}\right] - \beta\,\mathbb{E}_{\mathcal{C}}[\text{KL}(\pi_\theta(\cdot \mid \mathcal{C})\|\pi_{\text{ref}}(\cdot \mid \mathcal{C}))]\,, \qquad \rho_{m,i} = \frac{\pi_\theta(x_{m,i} \mid \mathcal{C})}{\mu(x_{m,i} \mid \mathcal{C})}. \tag{33}$$

At $\theta = \theta_{\text{old}}$, the diversity component of the surrogate gradient is λ times Eq. (32). Away from the behavior policy, optimization follows this clipped local surrogate. Thus the population diversity target, the credit-shaped local objective, and the estimator used by a particular GRPO-family implementation are distinct.

D GenMol-P Implementation

D.1 Method

GenMol-P extends GenMol [30] from unconditional molecular generation to pocket-conditioned molecular generation by adding a continuous pocket prefix to the discrete diffusion language model. Let $x = (x_1, \ldots, x_T)$ denote the SAFE-token sequence of a ligand and let $\mathcal{C}$ denote a protein pocket. In GenMol-P, $\mathcal{C}$ is represented by the residue sequence together with backbone coordinates (N, C_α, C) for each pocket residue. A frozen ESM-IF1 encoder [24] maps these pocket backbone coordinates to residue-level embeddings $h_1, \ldots, h_L$, where L is the number of pocket residues. A trainable two-layer projector P_ψ then maps each residue embedding into the GenMol hidden space, producing prefix vectors $p_\ell = P_\psi(h_\ell) \in \mathbb{R}^d$, with $d = 768$. The projector consists of a linear layer, GELU nonlinearity, a second linear layer, and layer normalization.

The projected pocket vectors are inserted as a prefix before the molecular token embeddings. Given corrupted molecule tokens $\tilde{x}_t$ at diffusion time t, GenMol-P forms the transformer input

$$[p_1, \ldots, p_L, e(\tilde{x}_{t,1}), \ldots, e(\tilde{x}_{t,T})],$$

where $e(\cdot)$ is the molecular token embedding. The transformer attends jointly over the pocket prefix and molecular positions, but logits are read only from the molecular positions. Thus, the pocket prefix conditions molecular denoising without being treated as tokens to be generated. The total prefix-plus-molecule length is capped at 256 positions, matching the GenMol positional budget. Training examples that exceed this budget are excluded during preprocessing. This construction keeps the molecular generation mechanism unchanged while allowing the denoising network to condition on the target pocket through continuous structural context.

D.2 Training

GenMol-P is supervised-tuned on CrossDocked2020 [17] pocket-ligand pairs. For each complex, the ligand SMILES string is converted to a SAFE string and tokenized with the GenMol tokenizer. The pocket is converted into a residue-level structural prefix by extracting, or deterministically reconstructing, the N, C_α, and C backbone coordinates for each pocket residue. We discard examples without an assigned train/validation/test split, examples whose ligand cannot be converted to a nonempty SAFE string, malformed pockets with missing backbone information, and examples whose pocket-prefix plus ligand-token length exceeds 256.

The supervised objective is the same masked discrete diffusion objective as GenMol, with the pocket prefix supplied as additional context. For a training pair $(x, \mathcal{C})$, we sample a diffusion time $t \in [\varepsilon_{\text{time}}, 1]$ with $\varepsilon_{\text{time}} = 10^{-3}$, corrupt the ligand token sequence through the masked discrete diffusion forward process, and train the model to recover the original SAFE tokens from $(\tilde{x}_t, \mathcal{C})$. We use antithetic time sampling within minibatches and optimize the globally averaged molecular-token loss. The ESM-IF1 pocket encoder is frozen throughout training. The GenMol backbone and the pocket projector are trainable. The GenMol backbone is initialized from the pretrained GenMol checkpoint, so supervised tuning learns pocket conditioning while retaining the molecular prior learned by the unconditional generator.

The reported GenMol-P checkpoint is trained on 8 H200 GPUs with bf16 precision, per-device batch size 384, and no gradient accumulation, giving a global batch size of 3072. We use AdamW with learning rate 3×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, optimizer $\epsilon = 10^{-8}$, and zero weight decay. The learning-rate schedule is constant after 2500 warmup steps, gradients are clipped at norm 1.0, and an exponential moving average with decay 0.9999 is maintained over the trainable parameters. The original GenMol-P model used in the pocket-conditioned experiments is the verified checkpoint at 5,500 supervised optimization steps, and the subsequent coupled-GRPO and coupled-SGRPO runs are initialized from this checkpoint.

E Experimental Implementation Details

E.1 *De novo* Small-Molecule Design

Model. The *de novo* small-molecule design experiment uses GenMol [30] as the base generator. GenMol is a masked discrete diffusion language model over SAFE molecular strings [38]. SAFE represents a molecule as an unordered sequence of fragment blocks: a molecule is decomposed into BRICS fragments, each fragment is written as a contiguous string with its attachment points preserved, and fragments are concatenated with separator tokens. This representation is well matched to non-autoregressive denoising because the molecular identity is insensitive to the order in which fragment blocks are listed. As a result, GenMol can model the whole molecular string bidirectionally and fill multiple masked positions in parallel, instead of committing to a left-to-right token order.

GenMol follows the masked discrete diffusion formulation of masked diffusion language models. For a clean SAFE sequence $x = (x_1, \ldots, x_L)$, the forward process independently corrupts each token by interpolating between the clean token and a mask token m:

$$q(z_t^l \mid x_l) = \text{Cat}\left(z_t^l;\, \alpha_t x_l + (1 - \alpha_t)m\right),$$

where z_t^l is the noisy token at diffusion time t, and α_t decreases from $\alpha_0 = 1$ to $\alpha_1 = 0$. A BERT-style denoiser $x_\theta(z_t, t)$ predicts the clean token distribution at each position from the partially masked sequence. It is trained with the masked-diffusion negative-ELBO objective, which can be viewed as a time-weighted masked language modeling loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x,t,z_t}\left[-\sum_{l:z_t^l=m} \log x_\theta^l(x_l \mid z_t, t)\right].$$

During generation, GenMol starts from a masked SAFE sequence and simulates the reverse process. Unmasked tokens are kept fixed, while each masked position is predicted from $x_\theta(z_t, t)$ and sampled with temperature τ. GenMol then confirms only the most confident predictions and leaves the remaining positions masked for later denoising steps. With randomness r, the confidence score for a sampled token $\hat{x}_l$ is

$$c_t^l = \log p_\theta^l(\hat{x}_l \mid z_t, t) + rt\epsilon_l, \qquad \epsilon_l \sim \text{Gumbel}(0, 1).$$

This confidence-based parallel unmasking gives GenMol a native diffusion sampler rather than an autoregressive action sequence. All post-training methods in this experiment, therefore, use the coupled-GRPO instantiation described in the main text, which evaluates completed molecules under paired diffusion masks while preserving GenMol's native denoising sampler. The original-model baseline is the pretrained GenMol checkpoint without any RL post-training.

Reward definition. The rollout-level utility is the same QED–SA reward used in Section 5.2. For a generated molecule x, we compute QED [5] and the raw synthetic accessibility score $\mathrm{SA}(x)$ [15]. QED is a normalized drug-likeness score that combines several medicinal-chemistry descriptors, including molecular weight, lipophilicity, hydrogen-bond donors and acceptors, polar surface area, rotatable bonds, aromatic rings, and structural alerts, into a single high-is-better value in $[0, 1]$. The raw SA score is a heuristic estimate of synthetic difficulty: it rewards molecules composed of common chemical fragments and penalizes structural complexity, such as large rings, stereochemical complexity, and unusual molecular topology. Since lower raw SA indicates easier synthesis, we use the high-is-better transformation $s_{\mathrm{SA}}(x) = \mathrm{clip}((6 - \mathrm{SA}(x))/5, 0, 1)$. The rollout utility is $u(x) = 0.6\,\mathrm{QED}(x) + 0.4\,s_{\mathrm{SA}}(x)$. For SGRPO, the group-level diversity reward is internal diversity over valid molecules, computed as one minus the mean pairwise Tanimoto similarity between Morgan fingerprints. Morgan fingerprints encode circular atom neighborhoods as binary substructure features, and Tanimoto similarity measures the overlap between two such feature sets. Thus, SGRPO optimizes the same molecule-level quality reward as GRPO, but adds an explicit finite-group estimate of sample diversity.

Baselines. We compare SGRPO against five baselines: pretrained GenMol (Original), coupled-GRPO (GRPO), Memory-Assisted GRPO, Diverse Mini-Batch GRPO (DMB-GRPO), and Entropy-Regularized GRPO (ER-GRPO). Original measures the utility-diversity frontier before RL post-training. Coupled-GRPO [20] uses the rollout utility reward but has no explicit set-level diversity reward. Directly applying GRPO to a masked diffusion generator is nontrivial because a completed sample does not come with the autoregressive factorization normally used to compute token-level policy ratios. Treating the whole completion as fully masked gives a weak likelihood proxy, while independently resampling masks gives high-variance estimates. Coupled-GRPO keeps the GRPO reward and advantage structure but changes how the diffusion policy ratio is estimated.

For a condition c, coupled-GRPO samples a group of completed molecules $\{x_i\}_{i=1}^{G}$ and computes a group-relative utility advantage

$$A_i = u(x_i) - \frac{1}{G}\sum_{j=1}^{G} u(x_j).$$

The update then uses a PPO-style clipped objective, but the log-probability proxy is computed through coupled diffusion masks. For each completed molecule x_i, coupled-GRPO samples timestep pairs $(t, \hat{t})$ with $t + \hat{t} = T$, and constructs complementary completion masks M_t and $\widehat{M}_{\hat{t}}$ such that $M_t^l + \widehat{M}_{\hat{t}}^l = 1$ for every completion token l. A compact way to write the corresponding log-probability proxy is

$$\ell_\theta(x_i; c) = \sum_{(t,\hat{t})} \left[\sum_{l: M_t^l = 1} \log p_\theta(x_i^l \mid c, x_i^{\neg M_t}, t) + \sum_{l: \widehat{M}_{\hat{t}}^l = 1} \log p_\theta(x_i^l \mid c, x_i^{\neg \widehat{M}_{\hat{t}}}, \hat{t}) \right],$$

where $x_i^{\neg M}$ denotes the visible, unmasked part of the completion under mask M. Thus, each token contributes exactly once across the coupled pair, but is evaluated under a realistic, partially observed context rather than an all-mask context. The policy ratio in the clipped GRPO objective is then formed from this coupled log-probability proxy. Coupled-GRPO therefore preserves the native diffusion denoising interface and reduces the variance of the policy-gradient estimate relative to independently sampled masks. In our baseline, it optimizes only rollout-level utility, without any set-level diversity reward.

Memory-assisted coupled-GRPO, denoted Memory-Assisted GRPO, augments coupled-GRPO with the memory unit of memory-assisted RL [7]. We use the compound-similarity version of this memory: it is organized as index–bucket pairs, where each index is a molecule that represents a similarity neighborhood, and each bucket stores the high-scoring generated molecules assigned to that neighborhood. For a newly generated molecule x, the memory is queried only if its utility score exceeds a threshold η. The method then compares x with all indexed molecules using Morgan-fingerprint Tanimoto similarity. If no index has similarity at least γ, a new index–bucket pair is created with x, and x is stored in the new bucket. If x matches an existing index and the corresponding bucket has not reached its capacity C, x is added to that bucket and the reward is left unchanged. If the matched bucket is already full, the memory returns zero, and the reward is suppressed before

GRPO advantage computation. Thus, Memory-Assisted GRPO encourages diversity indirectly by suppressing reward in historically saturated similarity neighborhoods, rather than by optimizing the internal diversity of the current generated set.

DMB-GRPO adapts diverse mini-batch selection [49] to coupled-GRPO. It generates a candidate pool twice the optimized sample count and selects a diverse subset for the update using a fixed-cardinality determinantal point process (k-DPP), which favors subsets with a large determinant of their similarity kernel. This promotes diversity through training-sample selection while preserving the utility-based optimization objective.

ER-GRPO adds a policy-entropy bonus following Mnih et al. (2016). For a categorical policy distribution p_θ over vocabulary $\mathcal{V}$, we use normalized entropy $H(p_\theta)/\log|\mathcal{V}|$, where $H(p_\theta) = -\sum_{v\in\mathcal{V}} p_\theta(v)\log p_\theta(v)$, with coefficient $\lambda_{\text{ent}} = 0.01$. For GenMol, p_θ is the denoiser distribution evaluated under the coupled masking estimator. ER-GRPO reuses policy logits and retains the GRPO rollout count.

Training details. All RL-trained *de novo* small-molecule models start from the same GenMol checkpoint and are trained on 8 H200 GPUs with bf16 precision and seed 42. Training rollouts are sampled with random masking enabled, GenMol randomness 0.3, sampling temperature 1.0, minimum added length 60, generation batch size 1024, and one training iteration per rollout batch. GRPO uses 512 rollouts per prompt, a per-device training batch size of 1024, learning rate 5×10^{-5}, Adam $(\beta_1, \beta_2) = (0.9, 0.99)$, optimizer $\epsilon = 10^{-8}$, weight decay 0.1, maximum gradient norm 0.2, KL coefficient $\beta = 0.005$, clipping parameter $\epsilon_{\text{clip}} = 0.5$, cosine learning-rate decay with minimum learning-rate ratio 0.1, and warmup ratio 10^{-4}. The reference model is synchronized every 64 steps with a mixup coefficient 0.6. The reported GRPO model is the 2,000-step checkpoint. Memory-Assisted GRPO uses the same GRPO settings and additionally enables the history-based memory with bucket size 25, score threshold 0.9, and similarity cutoff 0.4. The reported model is also the 2,000-step checkpoint. SGRPO uses the same optimizer, KL, clipping, scheduler, and training-sampling settings, but keeps the total rollout budget matched by forming 8 same-condition groups with 64 rollouts per group. We set the group-reward weight to $\lambda = 0.5$ and use leave-one-out group credit at temperature 1.0. The reported SGRPO model is the 2,000-step checkpoint.

DMB-GRPO and ER-GRPO use the same initialization, utility reward, optimizer, learning rate, 2,000 training steps, KL coefficient, and other training settings as GRPO. DMB-GRPO matches the number of optimized samples while generating twice as many candidates; ER-GRPO uses the same rollout count as GRPO. All methods use the same evaluation sample counts, decoding settings, and five seeds.

Evaluation protocol. For the main Pareto curve, we evaluate all six methods at the ten paired GenMol randomness–temperature settings in Eq. (34). Each model generates 1,000 molecules at each sweep point. QED, transformed SA, and utility are averaged over valid generated molecules at that sweep point, and diversity is computed over the same valid set using Morgan-fingerprint Tanimoto distance. The frontier metrics in Table 1 are computed from these ten operating points for each method.

E.2 Pocket-Based Small-Molecule Design

Model. The pocket-based small-molecule experiment uses GenMol-P, the pocket-conditioned extension of GenMol described in Appendix D. GenMol-P conditions molecular denoising on a protein pocket by prepending a continuous pocket prefix, obtained from frozen ESM-IF1 pocket embeddings and a trainable projector, before the SAFE-token embeddings. The original GenMol-P model used in this experiment is the supervised CrossDocked2020 checkpoint at 5,500 optimization steps. All RL post-training methods start from this same checkpoint and use the coupled-GRPO family of objectives because the generator remains a discrete diffusion model.

Reward definition. The pocket-conditioned utility extends the *de novo* QED–SA reward with a target-dependent docking term. The QED and transformed SA components retain the same interpretation as in the unconditional small-molecule task: QED favors drug-like physicochemical profiles, while s_{SA} favors molecules estimated to be easier to synthesize. For ligand x and pocket $\mathcal{C}$, we define $u(x, \mathcal{C}) = 0.3\,\text{QED}(x) + 0.2\,s_{\text{SA}}(x) + 0.5\,s_{\text{dock}}(x, \mathcal{C})$, where $s_{\text{SA}}(x) = \text{clip}((6 -$

$\mathrm{SA}(x))/5, 0, 1)$. The docking term is derived from AutoDock Vina [53], which searches ligand poses in the pocket and scores them with an empirical approximation to binding affinity. If $a_{\mathrm{Vina}}(x, \mathcal{C})$ is the raw Vina affinity, lower and more negative values indicate stronger predicted binding, so we use the high-is-better transformation $s_{\mathrm{dock}}(x, \mathcal{C}) = \mathrm{clip}(-a_{\mathrm{Vina}}(x, \mathcal{C})/10, 0, 1)$. Diversity is condition-specific: for each pocket, we compute internal diversity among the ligands generated for that same pocket using the Morgan-fingerprint Tanimoto distance, and then average this value over pockets. This avoids rewarding diversity across unrelated targets, where chemical dissimilarity may simply reflect different pocket requirements rather than useful within-pocket diversity.

Baselines. We compare SGRPO against the original supervised GenMol-P model (Original), coupled-GRPO (GRPO), DMB-GRPO, and ER-GRPO. All RL methods use the same QED–SA–Vina rollout utility. DMB-GRPO uses the molecular k-DPP selection described in Appendix E.1. ER-GRPO uses the same normalized entropy bonus with $\lambda_{\mathrm{ent}} = 0.01$, evaluated on GenMol-P denoiser logits under the coupled masking estimator. We do not include a memory-assisted GRPO baseline in the main pocket-based comparison. The memory mechanism stores historically generated candidates across optimization steps, but the scientific diversity objective in this task is pocket-conditional: diversity should be encouraged among ligands for the same pocket, not across ligands generated for unrelated pockets. Making a memory-assisted baseline operate at the pocket level would require repeated optimization steps on the same pocket so that the memory becomes meaningful, which would change the data exposure and make the comparison no longer matched.

Training details. All pocket-conditioned RL methods are trained on 8 H200 GPUs with bf16 precision from the same 5,500-step GenMol-P checkpoint and seed 42. Training rollouts are sampled with random masking enabled, GenMol randomness 0.3, sampling temperature 1.0, minimum added length 60, generation batch size 384, and one training iteration per rollout batch. Coupled-GRPO uses 192 rollouts per pocket condition, per-device training batch size 384, learning rate 5×10^{-5}, Adam $(\beta_1, \beta_2) = (0.9, 0.99)$, optimizer $\epsilon = 10^{-8}$, weight decay 0.1, maximum gradient norm 0.2, KL coefficient $\beta = 0.005$, clipping parameter $\epsilon_{\mathrm{clip}} = 0.5$, cosine learning-rate decay with minimum learning-rate ratio 0.1, and warmup ratio 10^{-4}. The reference model is synchronized every 64 steps with a mixup coefficient 0.6. AutoDock Vina rewards are computed with fast search mode, one output pose, docking batch size 384, and a timeout of 1800 seconds. The reported GRPO checkpoint is trained for 1,000 steps. Coupled-SGRPO keeps the total rollout budget matched by forming 8 same-pocket groups with 24 rollouts per group. It uses the same optimizer, KL, clipping, scheduler, docking, and training-sampling settings as GRPO, sets the group-reward weight to $\lambda = 0.9$, and uses leave-one-out group credit at temperature 1.0. The reported SGRPO checkpoint is also trained for 1,000 steps.

DMB-GRPO and ER-GRPO share the GRPO initialization, utility reward, optimizer, learning rate, 1,000 training steps, KL coefficient, and other training settings. DMB-GRPO matches the optimized sample count using twice as many candidate rollouts, while ER-GRPO retains the GRPO rollout count. All methods use the same evaluation sample counts, decoding settings, and five seeds.

Evaluation protocol. The main pocket-based Pareto curve uses pockets from the CrossDocked2020 test set. We evaluate each model under the same paired (ρ, τ) sweep as the settings in Eq. (34). At each sweep point, each model generates 16 ligands for each of 100 test pockets, giving 1,600 ligand samples per method and sweep point. We scored generated ligands with QED, transformed SA, and AutoDock Vina. Utility metrics are averaged over valid generated ligands, while diversity is computed within each pocket's 16 ligands and then averaged over the 100 pockets. The frontier metrics in Table 1 are computed from the resulting ten utility-diversity operating points.

E.3 *De novo* Protein Design

Model. The *de novo* protein design experiment uses ProGen2-small [37], an autoregressive protein language model, as the base generator. ProGen2 models raw protein sequences with a decoder-only Transformer trained for next-token prediction. For an amino-acid sequence $y = (a_1, \ldots, a_L)$, the model defines a left-to-right sequence distribution

$$p_\theta(y) = \prod_{t=1}^{L+1} p_\theta(a_t \mid a_{<t}),$$

where a_{L+1} denotes the terminal token. Pretraining minimizes the corresponding negative log-likelihood over large collections of unaligned protein sequences, so generation is performed by repeatedly sampling the next amino-acid token from $p_\theta(\cdot \mid a_{<t})$ until the terminal token or a maximum length is reached.

Architecturally, ProGen2 is a causal Transformer decoder with rotary positional encodings and a parallelized residual block in which self-attention and the feed-forward network are applied to the same normalized hidden state.

$$h^{(m+1)} = h^{(m)} + \mathrm{Attn}(\mathrm{LN}(h^{(m)})) + \mathrm{MLP}(\mathrm{LN}(h^{(m)})).$$

The ProGen2 family scales this architecture across model sizes. We use the official ProGen2-small checkpoint, which has 151M parameters, 12 layers, 16 attention heads, head dimension 64, and context length 1024. This checkpoint is pretrained on the standard ProGen2 mixture of UniRef90 and BFD30 sequences, giving it broad coverage of natural protein sequence statistics. The original baseline is this checkpoint evaluated without RL post-training. Since ProGen2 provides an explicit token-level likelihood for every generated sequence, SGRPO can be instantiated with standard GRPO rather than the coupled-GRPO estimator required for diffusion generators. All post-training methods generate unconditional amino-acid sequences from the same prompt set and are evaluated at the 100-step checkpoint used in the main comparison.

Reward definition. The sequence-level utility combines four normalized protein-design scores, each targeting a different failure mode of unconstrained protein generation. Naturalness is computed from the average per-token log-likelihood under ESM2, so sequences that look more plausible under a large protein language model receive higher scores. Foldability is the ESMFold mean pLDDT score divided by 100. pLDDT is a per-residue confidence estimate, so this term favors sequences whose predicted structures are internally confident. Stability is based on TemBERTure-predicted melting temperature, which serves as a sequence-based proxy for thermal robustness. Developability combines Protein-Sol solubility with simple liability filters, so it penalizes sequences that may be hard to express or handle experimentally. Naturalness and stability are quantile-normalized using calibration sequences, while foldability and developability are already on a $[0, 1]$-compatible scale. The rollout utility is $u(y) = 0.25\, r_{\mathrm{nat}}(y) + 0.30\, r_{\mathrm{fold}}(y) + 0.20\, r_{\mathrm{stab}}(y) + 0.25\, r_{\mathrm{dev}}(y)$. For developability, the underlying score is $r_{\mathrm{dev}}(y) = 0.8\, r_{\mathrm{sol}}(y) + 0.2\, r_{\mathrm{liability}}(y)$, where r_{sol} is the Protein-Sol score clipped to $[0, 1]$ and $r_{\mathrm{liability}}$ penalizes transmembrane-like hydrophobicity, low sequence complexity, long hydrophobic runs, and cysteine outlier frequency. Sequence diversity is measured as one minus the average normalized Levenshtein similarity over valid generated sequences. The normalized Levenshtein similarity compares two sequences by their edit distance after accounting for sequence length, so lower similarity corresponds to larger sequence-level variation.

Baselines. We compare SGRPO against the original ProGen2-small generator (Original), GRPO, Memory-Assisted GRPO, DMB-GRPO, and ER-GRPO. GRPO optimizes only the rollout-level protein utility. Memory-Assisted GRPO uses the same rollout utility but adds the history-based diversity memory over generated amino-acid sequences. Sequence similarity in this memory is normalized Levenshtein similarity. The low diversity of Memory-Assisted GRPO in the protein experiment should be interpreted as a consequence of its history-relative novelty mechanism rather than a difference in training budget: all RL methods start from the same ProGen2 checkpoint, use the same rollout utility and matched GRPO training settings, and are evaluated under the same temperature sweep.

DMB-GRPO adapts diverse mini-batch selection [49] to autoregressive GRPO. It generates twice as many candidate sequences as are optimized and uses a task-matched MaxMin selection rule that favors sequences distant from those already selected. ER-GRPO applies the normalized entropy bonus defined in Appendix E.1 to the next-token policy, with $\lambda_{\mathrm{ent}} = 0.01$. It reuses policy logits and retains the GRPO rollout count.

Training details. All ProGen2 post-training runs are trained on 8 H200 GPUs and use the official ProGen2-small checkpoint as both the initial policy and the initial reference model. GRPO uses group size 96, per-device training batch size 192, gradient accumulation 1, maximum generation length 256, top-$p = 0.95$, training sampling temperature 0.8, learning rate 5×10^{-5}, Adam $(\beta_1, \beta_2) = (0.9, 0.999)$, optimizer $\epsilon = 10^{-8}$, zero weight decay, maximum gradient norm 1.0, KL coefficient $\beta = 0.01$, and clipping parameter $\epsilon_{\mathrm{clip}} = 0.2$. Memory-Assisted GRPO uses the same training

settings and enables the history-based diversity memory with bucket size 25, score threshold 0.6, and similarity cutoff 0.6. In each supergroup, SGRPO keeps the rollout budget matched by forming 8 groups with 12 sequences per group. It uses the same optimizer, generation, KL, and clipping settings as GRPO, sets the group-reward weight to $\lambda = 0.8$, and uses leave-one-out group credit at temperature 1.0. For GRPO, Memory-Assisted GRPO, and SGRPO, reward calibration uses 1,024 generated sequences. Naturalness, stability, and developability rewards are computed at every step, while the more expensive foldability reward is computed every 4 steps. Each model is trained 100 steps.

DMB-GRPO and ER-GRPO use the same initialization, rollout utility, optimizer, learning rate, 100 training steps, KL coefficient, and other task-specific settings as GRPO. DMB-GRPO matches the optimized sample count using twice as many candidate rollouts, while ER-GRPO uses the GRPO rollout count. All methods use the same evaluation sample counts, temperatures, and five seeds.

Evaluation protocol. The protein Pareto curve is evaluated with a temperature sweep $\tau \in \{0.1, 0.2, \ldots, 1.0, 1.1, 1.2\}$. At each temperature, each model generates 512 sequences. Naturalness and stability are calibrated once using a fixed set of 256 calibration sequences. The resulting normalization parameters are then held fixed for all models and all temperatures in the sweep. The plotted utility is the weighted utility defined above, and the frontier metrics in Table 1 are computed from the 12 temperature-specific operating points for each method.

F Extended Evaluation

F.1 Evaluation Protocol and Metric Interpretation

Both small-molecule tasks use ten paired randomness–temperature settings:

$$\begin{aligned}(\rho, \tau) \in \{&(0.1, 0.5), (0.2, 0.65), (0.3, 0.8), (0.4, 0.95), (0.5, 1.1),\\ &(0.6, 1.25), (0.7, 1.4), (0.8, 1.55), (0.9, 1.7), (1.0, 2.0)\}.\end{aligned} \tag{34}$$

The protein task retains its twelve temperatures. Each comparison uses five independent generation sweeps of the trained checkpoints, with matched generation budgets and seeds across methods; these are evaluation runs, not five independently trained models. Every frontier metric is computed separately for each run before reporting its mean and 95% confidence interval. For HV, the reference coordinates are the minimum utility and diversity across all methods and operating points in the corresponding comparison panel for that run. Thus HV values are comparable within a panel, but can change when the methods included in the panel change. Separate ablation and scaffold-diversity tables use their own stated comparison panels.

The three indicators summarize different aspects of a frontier. HV measures geometric coverage, rewarding both outward movement and a broad range of trade-offs. DIP measures only the closest operating point to the ideal and ignores the rest of the frontier. R2 averages the best weighted shortfall over preference directions, assessing whether different preferences have suitable operating points. A single strong compromise can yield a good DIP without broad coverage; a wide frontier can have a high HV without the best preference-averaged shortfall. In Table 1, for example, DMB-GRPO has the highest mean small-molecule HV, whereas SGRPO has the lowest mean DIP and R2.

We describe numerical rankings separately from statistical evidence. Overlapping 95% confidence intervals do not establish a significant difference, and we describe such results as competitive.

F.2 Larger Within-Pocket Evaluation Sets

To test sensitivity to the finite evaluation set, we increase the generation budget from 16 to 100 ligands for each of 100 held-out CrossDocked2020 pockets. At every one of the ten decoding settings, each model therefore generates 10,000 ligands per run. Utility is aggregated over valid ligands, while Morgan-fingerprint internal diversity is computed within each pocket and then averaged across pockets. Table 2 uses a common per-run HV reference point determined by its three models. SGRPO retains the best mean HV, DIP, and R2.

Table 2: Pocket-based evaluation with 100 ligands per pocket, ten decoding settings, and five independent sweeps. Entries are mean ± 95% CI; the HV reference is shared only by the three methods in this table.

Method	HV ↑	DIP ↓	R2 ↓
Original	$0.0261_{\pm 0.0012}$	$0.4646_{\pm 0.0023}$	$0.2395_{\pm 0.0007}$
GRPO	$0.0094_{\pm 0.0005}$	$0.7421_{\pm 0.0069}$	$0.3820_{\pm 0.0036}$
SGRPO	$\mathbf{0.0632_{\pm 0.0023}}$	$\mathbf{0.3805_{\pm 0.0007}}$	$\mathbf{0.1810_{\pm 0.0004}}$

F.3 Raw Diversity Contributions and Alternative Centering

Raw-LOO diversity reward. We compare SGRPO with a baseline that directly uses the raw leave-one-out contribution $c_{m,i} = D(G_m) - D(G_m \setminus \{x_{m,i}\})$ as its diversity reward, without the supergroup comparison and sign-aware redistribution. For normalized pairwise diversity with $K > 2$, let $q_i = (K-1)^{-1} \sum_{j \neq i} d(x_i, x_j)$. Direct calculation gives $c_i = 2(q_i - D_K(G))/(K-2)$, making its relation to mean distance from the other candidates explicit. The experiment uses the same ten-point, five-run *de novo* small-molecule protocol, with 1,000 molecules per setting. Table 3 shows lower DIP and R2 and higher HV for SGRPO, supporting the empirical value of the group comparison and redistribution beyond using raw contributions alone.

Table 3: Raw-LOO comparison on *de novo* small-molecule design. Mean ± 95% CI over five sweeps; the HV reference is determined jointly from these two methods for each run.

Method	HV ↑	DIP ↓	R2 ↓
Raw-LOO	$0.0372_{\pm 0.0013}$	$0.2680_{\pm 0.0021}$	$0.1064_{\pm 0.0008}$
SGRPO	$\mathbf{0.0410_{\pm 0.0015}}$	$\mathbf{0.2551_{\pm 0.0020}}$	$\mathbf{0.0955_{\pm 0.0004}}$

Relative-baseline choice. Leave-one-out mean centering without subsequent standard-deviation normalization follows the RLOO formulation of Ahmadian et al. (2024) and is a design choice rather than a core contribution. It is distinct from computing a candidate's leave-one-out diversity contribution: the former centers rewards against other samples, whereas the latter measures how removing a candidate changes a set score. For L rewards $a_1, \ldots, a_L$, leave-one-out centering satisfies $a_i - \frac{1}{L-1} \sum_{j \neq i} a_j = \frac{L}{L-1}(a_i - \bar{a})$, so it differs from mean centering by a fixed positive scale before further transformations.

We compare the default SGRPO centering with mean subtraction and mean subtraction followed by standard-deviation normalization. The three variants use the same ten-point, five-run *de novo* small-molecule evaluation. Table 4 shows broadly comparable performance. Mean centering is slightly better numerically across the three metrics, while Mean + Std has similar HV and R2 to SGRPO but a larger DIP. These results support mean centering as a competitive alternative rather than establishing an advantage for the default centering rule.

Table 4: Centering ablation on *de novo* small-molecule design. Mean ± 95% CI over five sweeps, with the HV reference determined jointly from these three variants per run.

Centering	HV ↑	DIP ↓	R2 ↓
SGRPO (leave-one-out)	$0.0407_{\pm 0.0019}$	$0.2551_{\pm 0.0020}$	$0.0955_{\pm 0.0004}$
Mean	$\mathbf{0.0415_{\pm 0.0018}}$	$\mathbf{0.2534_{\pm 0.0037}}$	$\mathbf{0.0942_{\pm 0.0005}}$
Mean + Std	$0.0411_{\pm 0.0019}$	$0.2666_{\pm 0.0024}$	$0.0954_{\pm 0.0012}$

F.4 Transfer to Scaffold Diversity

We evaluate whether diversity gains transfer to a metric not used for training. Define relative Bemis–Murcko scaffold diversity as

$$D_{\mathrm{scaf}}(S) = \begin{cases} \dfrac{|\{\mathrm{BM}(x) : x \in S_{\mathrm{valid}}\}|}{|S_{\mathrm{valid}}|}, & |S_{\mathrm{valid}}| > 0, \\ 0, & |S_{\mathrm{valid}}| = 0, \end{cases} \tag{35}$$

where S_{valid} retains sample multiplicity and $\mathrm{BM}(x)$ is the canonical Bemis–Murcko scaffold of x. The numerator counts distinct scaffolds; the denominator counts all valid generated molecules.

For pocket-based design, we reuse the saved generations and utility scores from the ten-point evaluation of Original, GRPO, and Morgan-diversity-trained SGRPO. Each run contains 16 ligands for each of 100 pockets per setting. Scaffold diversity is computed within each pocket and then averaged over pockets. No model is retrained for this comparison. Table 5 shows that SGRPO retains the best frontier metrics under this alternative evaluation metric, providing evidence of transfer beyond the pairwise training score.

Table 5: Utility–scaffold-diversity evaluation for pocket-based design, without scaffold-diversity training. Mean ± 95% CI over five sweeps; each run uses a reference point shared by these three methods.

Method	HV ↑	DIP ↓	R2 ↓
Original	$0.0252_{\pm 0.0031}$	$0.4988_{\pm 0.0052}$	$0.2586_{\pm 0.0033}$
GRPO	$0.0060_{\pm 0.0013}$	$0.7976_{\pm 0.0111}$	$0.4093_{\pm 0.0056}$
SGRPO	$\mathbf{0.0522}_{\pm \mathbf{0.0031}}$	$\mathbf{0.4393}_{\pm \mathbf{0.0054}}$	$\mathbf{0.2269}_{\pm \mathbf{0.0015}}$

F.5 Training with a Non-Pairwise Diversity Reward

To test compatibility with a different diversity definition while holding the task and generator fixed, we train SGRPO on *de novo* small-molecule design using D_{scaf} in Eq. (35). The optimization procedure is unchanged: group scores and leave-one-out contributions are obtained from this set function. Evaluation uses the utility–scaffold-diversity frontier, ten decoding settings, 1,000 molecules per model and setting, and five independent generation runs.

We compare with Original, GRPO, and ER-GRPO. The implemented Memory-Assisted GRPO and DMB-GRPO baselines rely on pairwise similarities for memory matching and mini-batch selection, respectively. Relative scaffold diversity cannot replace those pairwise quantities directly without redesigning the baselines, so they are omitted from this training comparison. This does not preclude other scaffold-aware adaptations of either method.

Scaffold-trained SGRPO has the best mean values on all three metrics in Table 6; some confidence intervals overlap. The result supports practical compatibility with a non-pairwise coverage statistic. The fixed-order U-statistic guarantee in Appendix B does not apply to this score.

Table 6: *De novo* small-molecule design with scaffold diversity as the SGRPO training reward and evaluation metric. Mean ± 95% CI over five sweeps, using a reference point shared by the four methods in each run.

Method	HV ↑	DIP ↓	R2 ↓
Original	$0.1133_{\pm 0.0026}$	$0.2818_{\pm 0.0027}$	$0.1125_{\pm 0.0027}$
GRPO	$0.1287_{\pm 0.0032}$	$0.2767_{\pm 0.0050}$	$0.1098_{\pm 0.0020}$
ER-GRPO	$0.1292_{\pm 0.0039}$	$0.2667_{\pm 0.0024}$	$0.1059_{\pm 0.0022}$
SGRPO (scaffold)	$\mathbf{0.1353}_{\pm \mathbf{0.0042}}$	$\mathbf{0.2511}_{\pm \mathbf{0.0050}}$	$\mathbf{0.0999}_{\pm \mathbf{0.0017}}$

F.6 Motif-Conditioned Molecular Generation

Setup. GenMol motif extension provides a second conditional generation task. The condition is a discrete molecular motif with a single attachment point, and the generated molecule must retain that

motif. Training uses 5,543 single-attachment motifs derived from CReM ChEMBL22, excluding the ten held-out motifs in the official GenMol evaluation benchmark. All RL methods start from the same initialization and use 1,000 training steps with matched optimized sample counts. We compare Original, GRPO, DMB-GRPO, ER-GRPO, and SGRPO with $\lambda = 0.5$; the RL methods use coupled-GRPO because the generator is a masked diffusion model.

Each method generates 100 molecules per test motif at each of the ten settings in Eq. (34), with five independent sweeps. Utility assesses full-molecule quality subject to molecular validity and retention of the conditioning motif. Morgan internal diversity is evaluated within each motif condition and then averaged across the ten motifs. This tests a discrete conditioning interface in addition to the three-dimensional pocket interface.

Results. SGRPO has the highest mean HV, while ER-GRPO has the lowest mean DIP and R2 (Table 7). SGRPO is competitive with GRPO and ER-GRPO, with overlapping confidence intervals across all three indicators. These results support its applicability to motif-conditioned generation without establishing a significant advantage over those two baselines.

Table 7: Motif-extension frontier metrics. Mean ± 95% CI over five sweeps, with 100 generations per motif and setting. The five methods share a per-run HV reference point.

Method	HV ↑	DIP ↓	R2 ↓
Original	$0.0036_{\pm 0.0013}$	$0.8324_{\pm 0.0084}$	$0.4369_{\pm 0.0051}$
GRPO	$0.0086_{\pm 0.0014}$	$0.7795_{\pm 0.0073}$	$0.4093_{\pm 0.0025}$
DMB-GRPO	$0.0054_{\pm 0.0007}$	$0.8135_{\pm 0.0082}$	$0.4268_{\pm 0.0064}$
ER-GRPO	$0.0093_{\pm 0.0016}$	$\mathbf{0.7756_{\pm 0.0104}}$	$\mathbf{0.4051_{\pm 0.0065}}$
SGRPO	$\mathbf{0.0096_{\pm 0.0018}}$	$0.7760_{\pm 0.0099}$	$0.4066_{\pm 0.0042}$

G Computational Details

G.1 Computational Cost Under Matched Rollouts

For B conditions and M groups of K candidates per condition, write $N = MK$. The M groups partition the N rollouts rather than multiplying an existing budget by M. Candidate generation, utility evaluation, and policy optimization process the same BN candidates as GRPO. The additional work is diversity scoring and credit assignment. Let $T_D(s)$ denote the cost of evaluating the diversity function on s candidates. Without metric-specific reuse, a black-box set score requires one evaluation on each complete group and K leave-one-out evaluations:

$$O(BM[T_D(K) + KT_D(K-1)]). \tag{36}$$

Group centering, contribution standardization, sign-aware weighting, and reward composition require $O(BN)$ additional arithmetic.

For pairwise diversity, let $T_m = \sum_{i<j} d(x_{m,i}, x_{m,j})$ and $t_{m,i} = \sum_{j \neq i} d(x_{m,i}, x_{m,j})$. Accumulating the total and all row sums in one pass gives

$$D(G_m) = \frac{2T_m}{K(K-1)}, \qquad D(G_m \setminus \{x_{m,i}\}) = \frac{2(T_m - t_{m,i})}{(K-1)(K-2)} \quad (K > 2). \tag{37}$$

Thus all K leave-one-out contributions follow without additional similarity evaluations. The pair count is $BM\binom{K}{2} = BN(K-1)/2$. Writing C_{sim} for the cost of one similarity evaluation, the additional similarity cost is $O(BNKC_{\text{sim}})$, followed by linear post-processing. At $K = 2$, a convention for singleton diversity is required. The smallest nontrivial SGRPO group size in our reported ablation is $K = 4$, while $K = 1$ denotes the separate utility-only GRPO setting.

Constructing a metric representation for each candidate is an additional per-candidate cost. Once fixed-length Morgan fingerprints are available, a Tanimoto comparison operates on fixed-size bit vectors. For two protein sequences of length at most L, Levenshtein dynamic programming has $O(L^2)$ worst-case cost. The pairwise reward calculation can therefore be more expensive for longer proteins. Reusing row sums removes repeated scoring of leave-one-out subsets but does not remove the metric's underlying pair cost.

G.2 End-to-End Training Efficiency

We profile *de novo* small-molecule training on eight H200 GPUs with 8,192 optimized samples per step. Table 8 reports mean step time and throughput over the full 2,000-step runs. SGRPO takes 18.96 s per step versus 15.10 s for GRPO, a 25.6% increase.

The profiling in Table 8 measures complete training steps, not an isolated diversity-scoring kernel. It shows the practical overhead under the reported implementation and hardware, but it does not identify how much time is attributable to each component. In particular, DMB-GRPO matches optimized sample counts but draws twice as many candidates, so its timing comparison should not be interpreted as matched candidate-generation cost.

Table 8: End-to-end efficiency on *de novo* small-molecule design, averaged over 2,000 training steps on eight H200 GPUs with 8,192 optimized samples per step. DMB-GRPO generates twice as many candidates to select the same number for optimization.

Method	Mean step time (s)	Optimized samples/s
GRPO	15.10	542.50
Memory-Assisted GRPO	20.12	407.22
DMB-GRPO	27.27	300.45
ER-GRPO	17.81	459.85
SGRPO	18.96	432.07

H Limitations and Future Work

SGRPO trades an additional rollout structure for a direct set-level diversity signal. Unlike diversity-agnostic RL, SGRPO must sample groups of candidates under the same conditions in order to estimate and compare group diversity. Our analysis shows that SGRPO remains effective with a near-minimal group size, but the method can still require more rollouts than objectives that score each candidate independently. Improving the efficiency of finite-group diversity estimators, or reusing rollouts across compatible diversity computations, is therefore an important direction for making SGRPO cheaper at larger scales.

SGRPO also adds the cost of computing the diversity reward and the leave-one-out diversity contribution. For the pairwise-similarity diversity objectives used in our experiments, this cost is dominated by constructing the within-group similarity matrix. Let a supergroup contain M same-condition groups, each with K rollouts, and let C_{sim} denote the cost of one similarity evaluation. Computing all within-group pairs requires $O(MK^2C_{\mathrm{sim}})$ work, or $MK(K-1)/2$ pair evaluations up to constants. The same similarity matrix is then reused to compute both the group diversity reward and all leave-one-out contributions, so leave-one-out credit assignment does not require a second pass over the pairwise similarities. Our implementation follows this reuse pattern to avoid redundant similarity computation.

This overhead should be interpreted relative to other diversity-aware training mechanisms. Under the matched-rollout protocol used in our experiments, Memory-Assisted GRPO evaluates the same MK generated candidates but compares high-scoring candidates against a memory with B index–bucket pairs. Its memory lookup therefore scales as $O(MKBC_{\mathrm{sim}})$ in the number of stored similarity neighborhoods. Once B is group-scale or larger, this lookup can exceed the pairwise group-diversity computation. In the *de novo* small-molecule run, for example, SGRPO uses $K = 64$ and $M = 8$, while the memory already contains more than 300 index–bucket pairs after the first training step.

The diversity-computation overhead is nevertheless real. For pairwise-similarity diversity, beyond the similarity-matrix reuse used here, future implementations could cache pairwise similarities across repeated candidates or cache reusable metric-specific computations such as fingerprints, embeddings, or nearest-neighbor structures. For diversity notions that are not naturally pairwise-similarity based, the appropriate efficiency strategy may be different: GPU-batched diversity computation, differentiable or learned proxy metrics, or approximate set summaries may be needed to keep set-level rewards practical at larger rollout scales.

Our experiments evaluate the utility-diversity trade-off with respect to the scalar utility specified for each task. These utilities aggregate several design-relevant components, such as drug-likeness,

synthesizability, docking affinity, foldability, stability, and developability, using task-specific weights. This scalarization is the shared interface through which a design preference is provided to reward-based post-training: the method is given a utility function and a diversity metric, and the evaluation asks whether the attainable trade-off between them improves. Questions about how to choose, calibrate, or audit the internal utility components are important reward-design questions shared by all reward-based post-training methods. They are complementary to the contribution studied here: given a user-specified utility and diversity metric, SGRPO directly improves the attainable trade-off between them. Although the scope of this work treats utility and diversity as user-specified evaluation axes, a natural extension is to expose the utility components themselves as additional controllable axes, allowing future SGRPO variants to study diversity jointly with more fine-grained within-utility trade-offs.

SGRPO is intentionally decoupled from a specific generator, task, utility objective, or diversity metric. This makes it easy to instantiate through different GRPO-style optimizers, but it also leaves room for more specialized designs. Future work could use the same supergroup-relative framework with task-specific diversity notions, condition-aware chemical series constraints, structure-aware protein diversity metrics, or adaptive group construction rules that better match the scientific objective of a particular design campaign.

Diversity definitions and guarantees. Replacing the set-level diversity function changes the property encouraged by the reward, even when the optimization procedure remains unchanged. Exact population and partition alignment is established here for normalized fixed-order U-statistics. The scaffold experiment supplies empirical evidence for a non-pairwise metric. The candidate-wise redistribution also introduces the credit-shaping term in Eq. (32). However, reward conservation alone does not remove this term or provide a general convergence guarantee.

Computational proxies and biological validation. Our evaluations measure the attainable trade-off between specified computational utility and diversity scores. Whether a utility proxy predicts biological activity or functional relevance is a separate validation question shared by reward-based design methods. Improvements in these computational metrics do not establish experimental activity, efficacy, or functional diversity. Experimental validation of selected candidates, and assessment of the correspondence between each utility proxy and measured outcomes, are important future work alongside more task-specific SGRPO instantiations.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are supported by the proposed framework in the method and the experimental results on representative datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: Include limitations in the appendix.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Include proof in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the datasets, preprocessing protocol, evaluation splits, baselines, metrics, and implementation details needed to reproduce the main experiment results.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and data for reproducing the main experimental results are publicly available at `https://anonymous.4open.science/r/SGRPO/README.md`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the datasets, train-test splits, evaluation metrics, baselines, model settings, and key hyperparameters used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports error bars suitably and correctly defined about the statistical significance of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details of training in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics and uses publicly available biological datasets and curated knowledge resources without involving private or sensitive human-subject data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive impact of the efficient protein design framework in the introduction motivation part.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release high-risk models, scraped datasets, or other assets with substantial misuse risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the sources of the datasets in the experimental setups part.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new public dataset.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human-subject studies and therefore does not require IRB approval.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLM is used for writing code and polishing the paper.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.