DySCo: A Lightweight Data Curation Framework for Reinforcement Learning in SMILES-based Molecular Design

Anonymous authors
Paper under double-blind review

Abstract

Molecular design aims to generate chemicals with desired properties, but the vastness of chemical space makes exhaustive search impractical. Reinforcement learning (RL)-based generative models have shown promise by enabling self-guided exploration. However, standard RL fine-tuning faces a trade-off: aggressive reward maximization often leads to mode collapse with loss of structural diversity, while maintaining diversity typically comes at the cost of slow convergence and high computational overhead due to extensive sampling. In practice, this cost is amplified because molecular design campaigns typically repeat the optimization loop across multiple targets and objective configurations. To address this challenge, we introduce DySCo (***Dy****namic* ***S****tate-adaptive* ***Co****reset*), a lightweight and task-agnostic framework for adaptive data curation during RL training. Unlike static replay buffers, DySCo treats the pre-trained prior policy as a continual source of structurally diverse molecules. It dynamically monitors the contribution of the current policy to high-reward elites as a simultaneous signal of diversity loss, and adaptively constructs compact corrective coresets that balance structural novelty and reward optimization. Empirical results on challenging multi-objective molecular generation tasks demonstrate that DySCo reduces per-epoch computational cost by 20%-45% while achieving competitive or superior reward scores, significantly higher structural diversity, and improved sample efficiency compared to existing baselines. As a data-level method, DySCo is complementary to reward-level and policy-level diversity approaches and compatible with different RL optimizers.

1 Introduction

Molecular design is a cornerstone of modern chemistry, focused on engineering molecules with desired properties for applications ranging from therapeutics to advanced materials. The field has shifted from traditional empirical discovery to systematic, data-driven engineering, as demonstrated by breakthroughs such as structure-based enzyme inhibitors and targeted functional polymers (Corey & Cheng, 1995; Cushman & Ondetti, 1999). Today, it plays a central role across diverse domains, including small-molecule therapeutics (Jorgensen, 2004; Özçelik et al., 2025), targeted protein degraders (Kombo & LaMarche, 2025), and advanced functional materials (Merchant et al., 2023).

However, designing novel molecules is challenged by the astronomical scale of the chemical universe. In the context of drug discovery alone, the candidate pool is estimated to exceed 10^{60} compounds (Bohacek et al., 1996), a magnitude that renders exhaustive experimental screening intractable. This scale has motivated the transition from empirical heuristics to computational approaches, such as virtual screening (Lipinski et al., 2001) and computer-aided drug design (Cherkasov et al., 2014). Nevertheless, these methods often rely on explicitly enumerating and ranking static libraries. Because existing datasets (typically $< 10^9$) represent only a tiny fraction of the total chemical space, these workflows are limited to searching known compounds rather than discovering entirely new structures (Reymond, 2015). This reliance on explicit enumeration imposes an inherent boundary on exploration, restricting the discovery of novel molecules to the finite diversity of pre-defined libraries.

To overcome this limitation, the community has undergone a paradigm shift toward AI-driven generation, where models autonomously navigate the chemical space to construct novel molecular structures

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

rather than querying pre-defined libraries (Brown et al., 2019; Tang et al., 2024). By representing molecules as 1D strings (i.e., *SMILES*, formally introduced in Section 2) (Weininger, 1988), sequence-based methods recast molecular design as a sequence modeling task. Utilizing the models such as Recurrent Neural Networks (RNNs) or Variational Autoencoders (VAEs), these methods can navigate chemical space via token-wise prediction with high efficiency (Segler et al., 2018). In parallel, geometry-based frameworks such as graph architectures (De Cao & Kipf, 2018) and diffusion models (e.g., EDM(Hoogeboom et al., 2022), 3DMolFormer(Hu et al., 2025)) utilize explicit 3D coordinates to capture fine-grained spatial information. Different molecular representations therefore offer different trade-offs between structural fidelity and computational efficiency (Loeffler et al., 2024; Wigh et al., 2022). In this work, we focus on SMILES because it offers a practical balance between chemical expressiveness and computational efficiency. Although 3D representations can encode fine-grained geometric information, they often require additional conformer generation or spatial modeling, which increases the total computational cost. SMILES, by contrast, provides a compact string representation that is directly compatible with mature sequence modeling architectures, making it well suited for the repeated sampling, scoring, and updating cycles required in reinforcement-learning-based molecular design (Loeffler et al., 2024; Blaschke et al., 2020a).

Beyond the vastness of the search space, molecular generation is typically a **multi-objective task** requiring the simultaneous optimization of conflicting properties such as *potency*, *selectivity*, and *synthesizability* (Olivecrona et al., 2017; Ertl & Schuffenhauer, 2009). Reinforcement Learning (RL) has emerged as a predominant framework to navigate these trade-offs by transforming molecular generation into a dynamic, "self-evolving" process (Blaschke et al., 2020a). Please see Figure 1 for an illustration. Unlike static models, an RL agent refines its strategy through iterative trial-and-error, leveraging feedback from its own generated samples to autonomously discover novel molecules. This data-driven policy refinement enables the exploration of uncharted chemical regions even when pre-labeled training data is scarce, ensuring the discovery of entities that satisfy complex functional requirements.

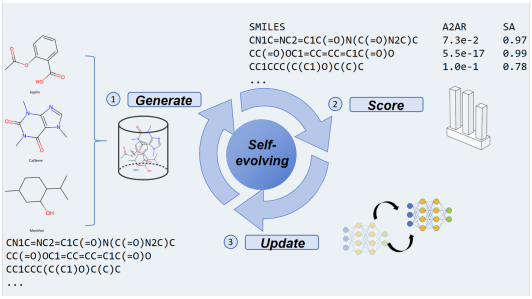


Figure 1: The "**self-evolving**" reinforcement learning loop for molecular design. The framework functions as an autonomous closed system driven by a continuous three-step cycle: (1) the generative policy generates a pool of SMILES sequences; (2) these molecules are encoded and evaluated by a reward function to yield scalar scores; and (3) the policy parameters are updated via gradient ascent to maximize expected returns. This iterative feedback drives policy refinement but relies entirely on self-generated data.

Despite these advantages, the autonomous loops of RL often lack diagnostic insight into their own search dynamics, leading to a precarious balance between exploration and exploitation (Zhou et al., 2019; Renz et al., 2019). This frequently results in **mode collapse**, where agents prematurely converge on repetitive scaffolds. Traditional stabilization techniques, such as experience replay or diversity filters, mitigate this entropy loss but impose heavy memory and computational penalties (Blaschke et al., 2020b; Gao et al., 2022). Since computational molecular design is an iterative upstream stage of the drug discovery pipeline that is typically repeated across multiple protein targets and reward configurations (Blaschke et al., 2020a; Loeffler et al., 2024), reducing per-iteration training cost without sacrificing generation quality is of practical importance. To alleviate such training costs, data curation strategies, particularly the idea of *coreset* (a compact and representative subset of the training data), have been widely adopted in machine learning (Feldman, 2019; Bachem et al., 2017; Mirzasoleiman et al., 2020; Qin et al., 2024; Tiwari et al., 2022; Tong et al., 2025; Zhan et al., 2025; Bachem et al., 2018). However, directly applying classic coreset methods to RL-based molecular generation presents a unique difficulty. In RL training, the generated molecules (expressed as SMILES strings) continuously change as the policy evolves, often resulting in batches that contain low-reward molecules or structurally similar scaffolds. In contrast, traditional coreset algorithms are designed to approximate a fixed, static data distribution. Compressing such evolving batches tends to preserve redundant low-reward and structurally similar molecules, thereby limiting further exploration of the chemical space. Consequently, we consider the following question:

*Is it possible to develop an adaptive data curation framework that simultaneously preserves **generation quality** (averts mode collapse), and ensures **computational efficiency** within self-evolving reinforcement learning for molecular design?*

Our main contributions. Resolving the above question requires overcoming two challenges. First, it is difficult to determine exactly *when* and *how much* structural intervention is required; static or blind injection of diverse data can easily dilute the learning signal and hinder reward convergence. Second, deciding *which* molecules to select remains non-trivial, as the algorithm must efficiently navigate the trade-off between exploring novel chemical scaffolds and exploiting high-reward functional modes. To tackle these challenges, we propose **DySCo** (***Dy**namic **S**tate-adaptive **Co**reset*), a lightweight, model-independent framework for state-adaptive data curation in molecular reinforcement learning. **First, to ensure an appropriate data source**, DySCo maintains a composite candidate pool at each training step by integrating molecules generated from both the task-adapted active policy and the unoptimized prior policy. By monitoring the generator contribution ratio among the top-rewarded molecules as a lightweight diagnostic signal, DySCo adaptively detects the immediate risk of structural homogenization and dynamically determines the appropriate number of corrective samples needed from the prior. **Second, for the coreset selection strategy**, DySCo takes account of the balance between exploitation and exploration. It employs three complementary strategies: "stochastic replay", "diversity induced greedy selection", and a novel "reward-aware semi-relaxed transport scoring" method. In particular, the latter leverages a **KL-regularized semi-relaxed optimal transport** formulation (Peyré et al., 2019; Villani et al., 2008; Vincent-Cuaz et al., 2022) to jointly consider molecular structural similarity and the relative reward importance of elite samples. By solving a closed-form row-wise coupling, this formulation avoids expensive iterative Sinkhorn updates while retaining reward-aware structural diversity, all at low computational overhead.

Our experiments across multiple benchmark datasets demonstrate that DySCo achieves higher computational efficiency and superior structural diversity compared to existing stabilization methods. Specifically, the framework reduces the training overhead by 20%-45% while maintaining robust optimization quality across varying multi-objective scales. Extended validations on DrugEx v3 (Liu et al., 2023), REINVENT 4 (Loeffler et al., 2024), and the standardized PMO benchmark (Gao et al., 2022) support broad applicability (Appendices D–I). Since DySCo operates at the data curation level, it is complementary to reward-level and policy-level diversity methods; additional analyses on sample efficiency, long-horizon stability, and compatibility with PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), multi-agent (Hu et al., 2023), and reward-level diversity approaches (Svensson et al., 2025) are in Appendix J.

2 PRELIMINARIES

Due to the space limit, detailed related work is provided in Appendix B, while this section introduces the preliminaries for SMILES-based molecular generation. ***SMILES*** (*Simplified Molecular-Input Line-Entry System*) represents molecular structures as one-dimensional strings (Weininger, 1988). Its compact sequential format makes it naturally compatible with sequence modeling architectures, which is particularly useful for iterative molecular generation. SMILES-based molecular generation can be formulated as an autoregressive sequence modeling problem. Let $x = (x_1, \ldots, x_L)$ denote a SMILES string of length L, where each token x_i is drawn from a fixed chemical vocabulary $\mathcal{V}$. A generative policy π_θ, typically parameterized by an RNN or Transformer architecture (Liu et al., 2021; Loeffler et al., 2024), defines the conditional probability of generating x as

$$P_\theta(x \mid c) = \prod_{i=1}^{L} \pi_\theta(x_i \mid x_{1:i-1}, c). \tag{1}$$

Here, "c" denotes an optional task-specific context. When $c = \emptyset$, the formulation corresponds to unconstrained de novo generation; otherwise, c may encode structural constraints such as a core scaffold for scaffold decoration (Fialková et al., 2021) or molecular fragments for linker generation (Guo et al., 2023).

A generated SMILES sequence must first be syntactically valid and correspond to a chemically valid molecular graph. Within this feasible space, molecular design seeks to improve task-specific molecular properties while maintaining sufficient structural diversity for continued exploration. The optimization objective is typically formulated as maximizing the expected reward under the generative policy:

$$\mathcal{J}(\theta) = \mathbb{E}_{x \sim P_\theta(\cdot|c)} \left[R(x) + \alpha\, \Omega(\theta, x) \right]. \tag{2}$$

Here, $R(x)$ denotes the task-specific reward of molecule x, often defined as a weighted combination of molecular property scores, i.e., $R(x) = \sum_{k=1}^{K} w_k r_k(x)$, where $r_k(x)$ is the score of the k-th property, such as binding affinity, synthetic accessibility (Ertl & Schuffenhauer, 2009), or drug-likeness (Olivecrona et al., 2017). The term $\Omega(\theta, x)$ represents a framework-specific regularization component, with its strength controlled by $\alpha > 0$; additional details are provided in Section 3.3.

We also need to emphasize that maximizing $\mathcal{J}(\theta)$ alone does not guarantee a useful molecular population. Because the reward may provide limited structural guidance, policy updates could repeatedly reinforce a small number of high-scoring scaffolds, leading to mode collapse and reduced structural diversity (Renz et al., 2019; Polykovskiy et al., 2020; Brown et al., 2019). Therefore, in addition to functional reward, molecular design also requires preserving structural diversity so that the generated candidates can cover multiple chemically distinct regions for downstream validation and further optimization (Svensson et al., 2025; Renz et al., 2024; Medina et al., 2026; Özçelik & Grisoni, 2025). Formally, this property is typically quantified using Internal Diversity (IntDiv), which is defined as the average pairwise structural dissimilarity within a generated molecular set; in this work, the structural dissimilarity is computed using Extended-Connectivity Fingerprints (ECFPs), with the exact formulation provided in Eq. 15 of Appendix A.3.

3 METHODOLOGY

As discussed in Section 1, the generated molecules in self-evolving are not only evaluated as outcomes, but also reused as training signals for subsequent policy updates. This closed-loop design enables adaptive exploration, but reward-driven updates can also make the policy progressively concentrate on a limited set of high-scoring structural motifs. Although we can maintain diversity through large replay buffers, external datasets, or complex filtering rules, they may introduce significant extra memory and computational costs. Thus, the key question is *how to construct a compact update set that preserves reward-driven exploitation while preventing excessive self-reinforcement.*

Our high level ideas. Our proposed **DySCo** addresses this question through state-adaptive corrective curation. At each epoch, it maintains two complementary sources of candidate molecules: samples from the *active policy*, i.e., the current policy being optimized during reinforcement learning, and samples from the *prior policy*, i.e., the pre-trained generator before task-specific optimization. We refer to the molecules generated by these two policies as *active-policy samples* and *prior samples*, respectively. Intuitively, the active-policy samples reflect the current task-adapted search direction, whereas the prior samples provide chemically valid candidates from a broader, task-unoptimized distribution. Rather than combining these two sources with a fixed mixing ratio, DySCo uses the fraction of active-policy samples in the high-reward subset to decide whether prior samples should be added. When this subset is overly dominated by active-policy samples, DySCo introduces corrective prior samples; otherwise, it keeps the update focused on reward-driven exploitation.

This design decomposes DySCo into three steps. **First**, DySCo constructs a two-source candidate pool and selects a high-reward subset, denoted by $\mathcal{S}_{\text{top}}$ and referred to as the *elite set*. It then estimates active-policy dominance within this elite set to determine the required number of additional prior samples, which we refer to as the *correction size*. **Second**, given this correction size, DySCo selects a subset of prior samples, denoted by $\mathcal{B}$, to complement the elite molecules. We refer to these selected prior samples as *corrective samples*. Together, the elite set and corrective samples form the ***curated training coreset***:

$$\mathcal{D}_{\text{train}} = \mathcal{S}_{\text{top}} \cup \mathcal{B}. \tag{3}$$

Finally, the policy is updated only on $\mathcal{D}_{\text{train}}$, reducing the cost of full generated-batch updates while preserving structural diversity. Section 3.1 formalizes the construction of the elite set $\mathcal{S}_{\text{top}}$ and the determination of the correction size; Section 3.2 presents how to select the corrective samples $\mathcal{B}$; and Section 3.3 describes the final policy update.

3.1 ELITE SET CONSTRUCTION AND CORRECTION SIZE DETERMINATION

At each training epoch t, DySCo first constructs the elite set $\mathcal{S}_{\text{top}}$ from a two-source candidate pool. Specifically, DySCo builds the **candidate pool** as

$$\mathcal{P}_t = \mathcal{S}_{\text{curr}} \cup \mathcal{S}_{\text{prior}}, \tag{4}$$

where $\mathcal{S}_{\text{curr}}$ contains molecules generated from the current active policy π_θ, and $\mathcal{S}_{\text{prior}}$ contains molecules generated from the unoptimized prior policy π_{prior}. The active-policy candidates reflect

the current task-adapted search direction, whereas the prior candidates provide chemically valid candidates from a broader, task-unoptimized distribution.

DySCo then selects the top-K molecules from $\mathcal{P}_t$ according to the reward function $R(x)$ and denotes the resulting elite set as $\mathcal{S}_{\text{top}}$, where $R(x)$ is previously defined in Section 2. *This elite set has two roles in the curation process: it provides the main exploitation signal through high-reward molecules, and it also reveals how strongly the current high-reward region is dominated by the active policy.* If most molecules in $\mathcal{S}_{\text{top}}$ come from $\mathcal{S}_{\text{curr}}$, the elite set is largely drawn from the active policy's current search distribution, indicating a higher risk of structural homogenization. If prior-generated molecules also appear among the elites, the high-reward set is not solely dominated by the active policy, suggesting that additional prior molecules may be less necessary. To quantify this effect, we define the "*generator contribution ratio*" at training epoch t:

$$\gamma_t = \frac{|\mathcal{S}_{\text{top}} \cap \mathcal{S}_{\text{curr}}|}{|\mathcal{S}_{\text{top}}|}. \tag{5}$$

DySCo uses γ_t to determine how many prior molecules should be added. Rather than adding prior molecules with a fixed proportion, DySCo adds only the minimum number of prior molecules required to keep the fraction of active-policy elites in the final training batch below a predefined balance threshold $\beta \in (0, 1]$. The advantage of such an adaptive strategy is that *it avoids adding unnecessary prior samples when the elite set is already balanced, while still introducing prior samples to broaden structural coverage when active-policy dominance becomes excessive*. Formally, after adding N_{corr} corrective samples, DySCo requires

$$\frac{|\mathcal{S}_{\text{top}} \cap \mathcal{S}_{\text{curr}}|}{|\mathcal{S}_{\text{top}}| + N_{\text{corr}}} \leq \beta. \tag{6}$$

Solving for the smallest non-negative integer satisfying this condition gives

$$N_{\text{corr}} = \max\left(0, \left\lceil |\mathcal{S}_{\text{top}}| \left(\frac{\gamma_t}{\beta} - 1\right) \right\rceil\right). \tag{7}$$

Eq. 7 specifies how many corrective prior samples should be added after observing the active-policy contribution in the elite set. If $\gamma_t \leq \beta$, the active-policy molecules do not exceed the allowed fraction, so no additional prior molecules are introduced and the elite set is kept as is. If $\gamma_t > \beta$, the elite set contains too large a fraction of active-policy molecules, and DySCo adds the smallest number of molecules from the remaining prior pool needed to bring this fraction back below β. Section 3.2 next describes how these prior molecules are selected.

3.2 Corrective Sample Selection Strategies

Given the correction size N_{corr}, DySCo next selects the corrective subset $\mathcal{B}$ of prior samples introduced in Eq. 3. This step does not replace reward-driven exploitation: $\mathcal{S}_{\text{top}}$ remains the high-reward component of the batch. Instead, $\mathcal{B}$ introduces additional prior-sampled molecules to improve structural diversity when the elite set is overly dominated by active-policy samples. We select $\mathcal{B}$ from the remaining prior pool:

$$\mathcal{S}_{\text{prior}}^{\text{rem}} = \mathcal{S}_{\text{prior}} \setminus \mathcal{S}_{\text{top}}, \tag{8}$$

where $\mathcal{S}_{\text{prior}}^{\text{rem}}$ contains prior samples that have not already been included in the elite set. The selected subset satisfies $|\mathcal{B}| = N_{\text{corr}}$.

The remaining question is how to choose $\mathcal{B}$ from $\mathcal{S}_{\text{prior}}^{\text{rem}}$. We consider three selection rules from different perspectives: stochastic replay uses only the prior source, diversity induced greedy selection additionally uses structural distance to the elite set, and reward-aware transport scoring further incorporates the relative importance of elite molecules.

I. Stochastic Replay (SR). The simplest strategy constructs $\mathcal{B}$ by uniformly sampling N_{corr} molecules from $\mathcal{S}_{\text{prior}}^{\text{rem}}$. This introduces additional molecules from the prior distribution with almost no extra computation. However, because the selection is random, it does not consider the structures already present in the elite set and may fail to target the specific regions where diversity is lacking.

II. Diversity Induced Greedy Selection (DGS). To make the selection structure-aware, this strategy favors prior samples that are structurally distant from the current elite set. We use the Tanimoto

distance $\mathcal{D}_T$ between ECFP fingerprints as the structural distance between molecules (Rogers & Hahn, 2010; Bajusz et al., 2015) (detailed definitions are provided in Appendix A.3). For each candidate $x_i \in \mathcal{S}_{\text{prior}}^{\text{rem}}$, we define its repulsion score as the average structural distance to the elite molecules:

$$s_{\text{rep}}(x_i) = \frac{1}{|\mathcal{S}_{\text{top}}|} \sum_{x_j \in \mathcal{S}_{\text{top}}} \mathcal{D}_T(x_i, x_j). \tag{9}$$

The corrective subset $\mathcal{B}$ is formed by selecting the top-N_{corr} candidates from $\mathcal{S}_{\text{prior}}^{\text{rem}}$ ranked by $s_{\text{rep}}(x_i)$, thereby favoring prior samples that are structurally distinct from the elite set. Alternatively, the normalized Hamming distance between fingerprint bit-vectors can be used when diversity in the binary representation is preferred (Appendix A.3).

III. Reward-Aware Transport Scoring (RTS). While Strategy II promotes structural diversity through geometric repulsion, it treats all elite molecules equally and ignores differences in reward. As a result, the corrective subset $\mathcal{B}$ may not prioritize candidate molecules that fill in gaps around the highest-reward molecules, leaving some important high-reward regions under-sampled. To address this limitation, we adopt a reward-aware selection mechanism based on *Optimal Transport* (OT) (Villani et al., 2008; Peyré et al., 2019). In general, OT compares two weighted sets by finding a non-negative coupling matrix Π that transports mass from one set to the other with minimum total cost. Given source weights $a = (a_1, \ldots, a_M)$, target weights $b = (b_1, \ldots, b_N)$, and a ground-cost matrix $C \in \mathbb{R}_+^{M \times N}$, the classical formulation minimizes $\langle \Pi, C \rangle$ under the marginal constraints $\Pi \mathbf{1}_N = a$ and $\Pi^\top \mathbf{1}_M = b$.

The key intuition behind using OT is that each prior candidate should be compared more strongly to elite molecules that are both high-reward and close in structural space, rather than being averaged over all elites uniformly. *Using OT, we select corrective molecules that supplement high-reward elite molecules with structurally diverse candidates, increasing structural diversity where it is most needed while maintaining high reward* . Classical OT, however, enforces strict marginal constraints and requires iterative solvers, which are computationally expensive for high-throughput RL. To address this, we adopt a carefully crafted **KL-regularized semi-relaxed OT** formulation, inspired by recent advances in semi-relaxed transport (Vincent-Cuaz et al., 2022; Shi et al., 2023): the elite-side marginal is relaxed and KL regularization induces smooth associations. This variant admits a closed-form row-wise solution, allowing efficient computation while preserving reward-aware structural diversity. We elaborate the details below.

We treat the remaining prior samples $\mathcal{S}_{\text{prior}}^{\text{rem}}$ as the candidate set with uniform source weights $a_i = 1/|\mathcal{S}_{\text{prior}}^{\text{rem}}|$, following the same unbiased scheme as in Strategy I. For the elite set $\mathcal{S}_{\text{top}}$, we define reward-dependent reference weights

$$b_j = \frac{\exp(R(x_j)/\tau)}{\sum_{x_k \in \mathcal{S}_{\text{top}}} \exp(R(x_k)/\tau)},$$

where τ controls the sharpness of reward preference. The ground cost is the Tanimoto distance $C_{ij} = \mathcal{D}_T(x_i, x_j)$, consistent with that used in Strategy II. Instead of using standard entropy regularization, we solve a KL-regularized semi-relaxed transport problem with the reward-weighted product measure $ab^\top$ as reference:

$$\Pi^\star = \arg\min_{\Pi \geq 0,\ \Pi \mathbf{1}_N = a} \left\{ \langle \Pi, C \rangle + \varepsilon \mathrm{KL}\big(\Pi \,\|\, ab^\top\big) \right\}. \tag{10}$$

We use the row-normalized coupling $q_{ij}^\star = \Pi_{ij}^\star / a_i$ as the reward-aware soft association between a prior candidate x_i and an elite molecule x_j. Higher-reward elites and structurally closer pairs receive greater association weight. We then define the transport score for each candidate as the expected structural cost under this conditional association:

$$q_{ij}^\star = \frac{\Pi_{ij}^\star}{a_i} = \frac{b_j \exp(-C_{ij}/\varepsilon)}{\sum_{x_k \in \mathcal{S}_{\text{top}}} b_k \exp(-C_{ik}/\varepsilon)}. \tag{11}$$

$$s_{\text{OT}}(x_i) = \sum_{x_j \in \mathcal{S}_{\text{top}}} q_{ij}^\star C_{ij}. \tag{12}$$

Compared with the reward-weighted average distance $\sum_j b_j C_{ij}$, this transport score introduces a local soft assignment through the kernel $\exp(-C_{ij}/\varepsilon)$. Thus, each prior candidate is compared

mainly with nearby high-reward elites, rather than with all elites uniformly. A large $s_{\mathrm{OT}}(x_i)$ indicates that the candidate remains structurally distinct under this reward-aware local comparison, and DySCo selects the top N_{corr} candidates to form $\mathcal{B}$.

3.3 Optimization and Model Training

Once the curated training coreset $\mathcal{D}_{\mathrm{train}}$ is constructed, the generative policy π_θ is updated using this compact subset. DySCo does not introduce a new reinforcement learning objective; instead, it changes which molecules are used to estimate the update. Following the general optimization objective defined in Eq. (2), a generic reward-regularized update over $\mathcal{D}_{\mathrm{train}}$ can be written as

$$\nabla_\theta \mathcal{J}(\theta) = \frac{1}{|\mathcal{D}_{\mathrm{train}}|} \sum_{x \in \mathcal{D}_{\mathrm{train}}} \left[R(x) \nabla_\theta \log P_\theta(x \mid c) + \alpha \nabla_\theta \Omega(\theta, x) \right], \tag{13}$$

The form of $\Omega(\theta, x)$ depends on the underlying molecular RL framework. For example, it may correspond to an augmented-likelihood loss, a prior-likelihood penalty, or other stabilization terms used to constrain policy updates. We provide the framework-specific implementations in Appendix D.1. Because DySCo operates at the data curation level, it is compatible with these regularization mechanisms without requiring changes to the model architecture or loss design.

In summary, DySCo affects training through the construction of $\mathcal{D}_{\mathrm{train}}$ rather than through changes to the policy-update rule. The elite set $\mathcal{S}_{\mathrm{top}}$ provides the reward-driven exploitation signal, while the selected prior subset $\mathcal{B}$ introduces additional structural diversity when needed. Since $\mathcal{D}_{\mathrm{train}}$ is only a compact subset of the candidate pool $\mathcal{P}_t$ (as defined in Eq. 4), this selective update can reduce per-epoch computation compared with updates based on the full candidate pool. The complete state-adaptive curation and optimization process is summarized in Algorithm 1.

Algorithm 1 DySCo: State-Adaptive Optimization Loop

Require: Pre-trained Prior π_{prior}, Total Steps T, Balance Threshold β, Top-K ratio $\kappa \in (0, 1]$, Learning Rate η
Ensure: Optimized parameters θ
Initialization:
$\pi_\theta \leftarrow \pi_{\mathrm{prior}}$
$\mathcal{S}_{\mathrm{prior}} \sim \pi_{\mathrm{prior}}$, $\mathcal{R}_{\mathrm{prior}} \leftarrow \mathrm{EvaluateRewards}(\mathcal{S}_{\mathrm{prior}})$ // Offline prior evaluation
for $t = 1$ **to** T **do**
  $\mathcal{S}_{\mathrm{curr}} \leftarrow \mathrm{Sample}(\pi_\theta)$, $\mathcal{R}_{\mathrm{curr}} \leftarrow \mathrm{EvaluateRewards}(\mathcal{S}_{\mathrm{curr}})$ // Online sampling
  $\mathcal{P}_t \leftarrow \mathcal{S}_{\mathrm{curr}} \cup \mathcal{S}_{\mathrm{prior}}$, $\mathcal{R}_t \leftarrow \mathcal{R}_{\mathrm{curr}} \cup \mathcal{R}_{\mathrm{prior}}$ // Composite pool
  $\mathcal{S}_{\mathrm{top}} \leftarrow \mathrm{SelectTopK}(\mathcal{P}_t, \mathcal{R}_t, \lfloor \kappa \cdot |\mathcal{P}_t| \rfloor)$
  $\gamma_t \leftarrow |\mathcal{S}_{\mathrm{top}} \cap \mathcal{S}_{\mathrm{curr}}| / |\mathcal{S}_{\mathrm{top}}|$ // Diagnosis
  $N_{\mathrm{corr}} \leftarrow \max(0, \lceil |\mathcal{S}_{\mathrm{top}}| \cdot (\gamma_t / \beta - 1) \rceil)$ // Corrective size (Eq. 7)
  if $N_{\mathrm{corr}} > 0$ **then**
    $\mathcal{S}^{\mathrm{rem}}_{\mathrm{prior}} \leftarrow \mathcal{S}_{\mathrm{prior}} \setminus \mathcal{S}_{\mathrm{top}}$
    $\mathcal{B} \leftarrow \mathrm{TopologicalBalancing}(\mathcal{S}^{\mathrm{rem}}_{\mathrm{prior}}, \mathcal{S}_{\mathrm{top}}, N_{\mathrm{corr}})$ // Adaptive curation
    $\mathcal{D}_{\mathrm{train}} \leftarrow \mathcal{S}_{\mathrm{top}} \cup \mathcal{B}$
  else
    $\mathcal{D}_{\mathrm{train}} \leftarrow \mathcal{S}_{\mathrm{top}}$ // Safe regime
  end if
  $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{J}(\theta)$ // Update via Eq. 13
end for

4 Experiments

We conduct experiments on multi-objective molecular generation tasks. Our primary evaluation uses the DrugEx v2 platform (Liu et al., 2021) in a dual-objective setting that jointly optimizes *A2AR binding affinity* and *Synthetic Accessibility* (SA), with both scores normalized to $[0, 1]$. To assess different trade-offs between bioactivity and synthesizability, we sweep nine preference weights with $w_{\mathrm{A2AR}} \in \{0.1, 0.2, \ldots, 0.9\}$ and $w_{\mathrm{SA}} = 1 - w_{\mathrm{A2AR}}$. To further examine generality, we conduct additional robustness studies on DrugEx v3 (Liu et al., 2023) for graph-based scaffold-constrained generation and REINVENT 4 (Loeffler et al., 2024) across diverse multi-objective generation settings. Due to the space limit, some details are omitted here, and **the full experimental details are provided in Appendix D**. DySCo introduces two algorithm-specific hyperparameters: the top-K ratio κ and the balance threshold β. The default ($\kappa = 0.1$, $\beta = 0.5$) was selected from a 5×6 grid (Appendix E.4); the grid shows a gradual reward–diversity trade-off across configurations, from

which we chose the balanced operating point. The OT regularization strength ε in Eq. (10) is set to 5 following standard practice in entropy-regularized transport (Peyré et al., 2019); an ablation over $\varepsilon \in \{0.5, 1, 2, 5, 10, 20\}$ confirms that performance is insensitive to this choice (Appendix J.5). Additional experiments with PPO, GRPO, multi-agent, reward-level diversity methods, and the standardized PMO benchmark (Gao et al., 2022) are in Appendix J.

Our methods and baselines. For our DySCo, we use the three selection rules described in Section 3.2, denoted as **DySCo (SR)**, **DySCo (DGS)**, and **DySCo (RTS)**. For DGS and RTS, which depend on the structural distance $\mathcal{D}_T$ as defined in Eq. 9 and Eq. 10, we specify the metric used, e.g., DySCo (DGS-Hamming) or DySCo (RTS-Tanimoto). Formal definitions of Tanimoto and Hamming distances are in Appendix A.3. We consider three groups of baselines. **First**, Vanilla RL updates the policy using the full self-generated batch and serves as the standard reference without data curation. **Second**, we adapt several representative coreset selection methods to the online molecular RL setting, including loss-based selection (InfoBatch (Qin et al., 2024) and Sorted Loss, the latter of which greedily selects the top-K samples by absolute policy loss at each epoch), gradient/bilevel matching (GCR (Tiwari et al., 2022), CSReL (Tong et al., 2025)), meta-learning-based selection (META (Zhan et al., 2025)), and clustering-based selection (KMeans (Bachem et al., 2018)). **Third**, we compare with common molecular RL stabilization methods, including experience replay and diversity filters (Gummesson Svensson et al., 2024) (denoted as "ER" and "Filter" respectively). Replay methods reuse historical or current generated molecules, while diversity filters penalize structurally redundant scaffolds.

Metrics. The experimental performance is evaluated on these criteria: *Optimization Quality* (average reward of unique molecules), *Structural Diversity* (Internal Diversity, IntDiv, computed on ECFP fingerprints via Eq. 15, as outlined in Section 2 and Appendix A.3), *Computational Efficiency* (training runtime), and *Sample Efficiency* (SE; mean reward of the cumulative top-1,000 molecules after 10,000 molecular evaluations, following the PMO protocol (Gao et al., 2022); detailed metric definitions in Appendix A.3).

4.1 Main Results

We report the averaged performance across all nine reward-weight combinations in Table 1, and visualize the reward–diversity trade-off in Fig. 2. Detailed results and extended robustness studies on DrugEx v3 and REINVENT 4 are provided in Appendices D–H.

As shown in Table 1 and Fig. 2, existing baselines exhibit different limitations. The reward-greedy methods, such as **ER(TC)** (relying exclusively on **T**op elites from the **C**urrent epoch) and **ER(TH)** (replaying **T**op **H**istorical elites), achieve very high rewards but occupy high-reward, low-diversity regions with concentrated structural coverage—the scaffold count drops from 882 to 468 and the top-10 scaffold share rises from 8.4% to 28.7% (Appendix J.7). More balanced replay and diversity-filter methods better preserve structural diversity but often incur higher runtime. Generic coreset methods provide useful compression baselines but are limited in the reward–diversity trade-off.

Compared with these baselines, DySCo variants achieve a more favorable practical trade-off. Rather than updating on the full candidate pool or relying on fixed replay rules, DySCo constructs the compact coreset $\mathcal{D}_{\text{train}} = \mathcal{S}_{\text{top}} \cup \mathcal{B}$, preserving the high-reward elite component while adaptively adding prior samples for structural coverage. Among the variants, **DySCo (SR)** achieves the fastest runtime within DySCo, reducing per-epoch time by 44.4% compared with Vanilla RL while maintaining com-

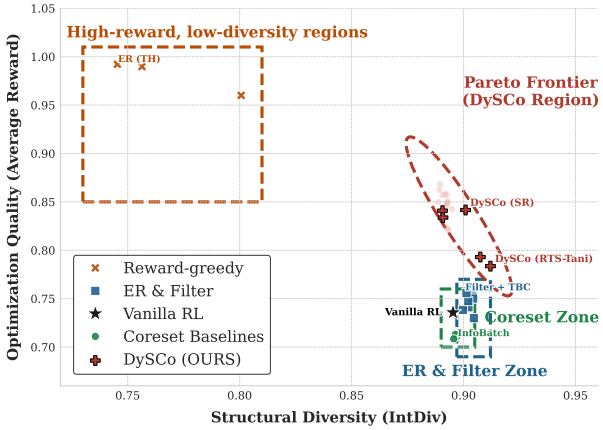


Figure 2: Optimization quality vs. structural diversity across methods. DySCo variants achieve high diversity while maintaining competitive reward. High-reward, low-diversity regions and conservative trade-off regions of coreset/replay baselines are indicated. Markers: Vanilla RL (star), coreset (circle), ER (square), DySCo (cross).

Table 1: Results over five seeds averaged across nine weight combinations. Details in Appendices E and J.3.

Method	Reward↑	IntDiv↑	Time (s)↓	SE(@10K)↑
Vanilla RL	0.7368 ± 0.0071	0.8950 ± 0.0011	1102	0.9439
InfoBatch	0.7112 ± 0.0062	0.8971 ± 0.0015	253	0.9497
Sorted Loss	0.7132 ± 0.0073	0.8964 ± 0.0002	196	0.9461
CSReL	0.7085 ± 0.0078	0.8956 ± 0.0033	638	0.9666
KMeans (Score)	0.8937 ± 0.0047	0.8320 ± 0.0017	1307	0.9846
ER(TC)	0.9956 ± 0.0013	*0.7484 ± 0.0028*	764	0.9873
ER(TH)	0.9921 ± 0.0011	*0.7454 ± 0.0011*	936	0.9839
ER(TBC)	0.7616 ± 0.0011	0.9039 ± 0.0024	848	0.9665
ER(TBH)	0.7562 ± 0.0055	0.9040 ± 0.0028	1142	0.9692
Filter + TBC	0.7558 ± 0.0048	0.9013 ± 0.0025	1342	0.9702
DySCo (SR)	**0.8416 ± 0.0023**	0.9009 ± 0.0007	**612**	0.9880
DySCo (DGS-Hamming)	0.8409 ± 0.0021	0.8906 ± 0.0009	827	0.9873
DySCo (RTS-Hamming)	0.8338 ± 0.0049	0.8908 ± 0.0011	887	0.9872
DySCo (DGS-Tanimoto)	0.7930 ± 0.0064	0.9075 ± 0.0023	845	0.9880
DySCo (RTS-Tanimoto)	0.7836 ± 0.0072	**0.9120 ± 0.0014**	885	**0.9881**

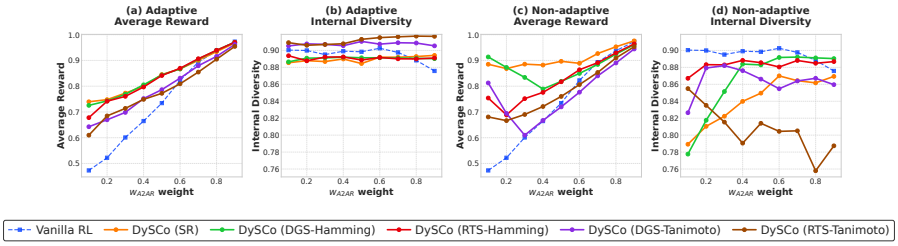


Figure 3: **Effect of Adaptive Curation.** (a,b) DySCo with adaptive N_{corr} control (top 10%, $\beta = 0.5$). (c,d) Non-adaptive baseline with a fixed corrective budget (top 10%, 10% extra prior samples).

petitive reward and diversity. Meanwhile, **DySCo (RTS-Tanimoto)** achieves the highest structural diversity and sample efficiency (SE column in Table 1), suggesting that reward-aware topological selection better preserves structural coverage. More detailed results are presented in Appendix E. We note that the 100-epoch protocol is a strict setting for DySCo: extended experiments show that its reward advantage grows with longer training while the diversity gap narrows (Appendix J.3). When evaluated under a sample-efficiency protocol across all Table 1 baselines, DySCo variants accumulate high-reward candidates faster: at 3,000 molecular evaluations, DySCo achieves a cumulative Top-1,000 mean reward of 0.945 compared with 0.841 for Vanilla RL (Table 15 in Appendix J.3).

4.2 Further Analysis

Adaptive Curation. To evaluate the impact of adaptive corrective curation, we compare DySCo under adaptive and non-adaptive settings using the same top-K selection (top 10% of candidates). In the adaptive setting, we dynamically compute the number of corrective prior samples N_{corr} using a balance threshold $\beta = 0.5$ (Eq. (7)). In contrast, the non-adaptive baseline applies a fixed corrective budget (equivalent to adding 10% extra prior samples after selecting the top 10% elites). Note that under our setting ($\beta = 0.5$), since $\gamma_t \in [0, 1]$, we always have $N_{\text{corr}} \leq |\mathcal{S}_{\text{top}}| \approx 10\%|\mathcal{P}_t|$. Thus, the total number of training samples per step remains no larger than that of the fixed-budget baseline.

Fig. 3(a,b) shows that adaptive control enables optimization trajectories to closely follow the intrinsic property trends of the chemical space while preserving a stable diversity profile. In contrast, the non-adaptive baseline (Fig. 3(c,d)) exhibits larger fluctuations in both reward and diversity, deviating noticeably from the expected trends. These results demonstrate that adaptive control of N_{corr} is critical for stable multi-objective optimization in generative molecular RL. We note that γ_t serves as a diagnostic signal of source imbalance, not a standalone predictor of IntDiv. Across 16 independent runs, higher γ_t was consistently associated with lower diversity across multiple metrics including IntDiv and scaffold-level indicators (Appendix J.1).

Utilization of the prior policy pool. To evaluate the importance of external diversity, we consider four protocols for constructing the composite candidate pool $\mathcal{P}_t$ (Eq. (4)): (1) **Zero-Shot**: The prior pool is entirely ignored ($\mathcal{P}_t = \mathcal{S}_{\text{curr}}$). The corrective subset $\mathcal{B}$ is sampled from the remaining current batch ($\mathcal{S}_{\text{curr}}^{\text{rem}}$) rather than the prior ($\mathcal{S}_{\text{prior}}^{\text{rem}}$, Eq. (8)). (2) **Single-Shot**: A fixed, small subset of prior molecules is sampled once before training and repeatedly reused at every epoch. (3) **Full-Shot**: The entire offline prior pool is integrated into $\mathcal{P}_t$ at every training step. (4) **Dynamic-Shot**: A new, distinct subset of prior molecules is dynamically sampled from the offline pool at each epoch (the default setting in DySCo and Buffer-based methods).

Table 2: Update-data sourcing protocols; full results are reported in Table 4.

Base Method	Protocol	Reward↑	IntDiv↑
ER(TBC)	Zero-Shot	0.7313	0.8994
	Single-Shot	0.7388	0.8975
	Full-Shot	0.7333	0.8977
	Dynamic-Shot	**0.7616**	**0.9039**
DySCo (SR)	Zero-Shot	0.8868	0.8370
	Single-Shot	0.8045	0.8947
	Full-Shot	0.7701	0.9034
	Dynamic-Shot	**0.8416**	**0.9009**

As shown in Table 2, the Dynamic-Shot protocol achieves the most balanced trade-off between reward optimization, structural diversity, and computational efficiency. Alternative approaches exhibit distinct limitations. Completely bypassing the corrective phase, as in the reward-greedy methods, produces concentrated structural coverage because no external regularization is introduced (Appendix J.7). Even with correction enabled, the Zero-Shot protocol fails to preserve diversity, since the active policy alone cannot inject novel scaffolds. Furthermore, static prior injection (Single-Shot) yields a suboptimal reward–diversity balance, whereas the Full-Shot protocol incurs higher computational overhead and risks diluting the exploitation signal with excessive prior molecules.

5 Conclusion

We presented DySCo, a state-adaptive curation framework for molecular RL that dynamically adjusts corrective prior samples to preserve high-reward molecules while enhancing structural diversity. Experiments across multiple platforms show a favorable reward–diversity trade-off with reduced per-epoch cost. DySCo builds on common molecular RL foundations—a pre-trained prior and standard fingerprints—focusing the current design on cheminformatics; future work will explore domain-adaptive diversity measures for materials design and protein engineering (Appendix L).

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing and to support the retrieval and discovery of related work. The authors reviewed all AI-assisted text and verified the retrieved sources. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work proposes a data curation method for computational molecular design. It does not involve human subjects, personal data, or dual-use concerns. Broader societal impacts are discussed in Appendix M.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide the following resources. The complete DySCo source code, including the core selection logic (`dysco/selectors.py`, `dysco/metrics.py`) and integration examples for the DrugEx framework, is included in the supplementary material as an anonymized code package. All experimental configurations (hyperparameters, reward functions, weight combinations, random seeds) are detailed in Appendix D. The main experiments use five independent seeds, and all results report mean $\pm$ standard deviation. The adaptation of coreset baselines to generative RL is described in Appendix D.6, and the experience replay configurations are in Appendix D.7. The closed-form solution of the KL-regularized semi-relaxed OT formulation is derived in Appendix C. All molecular evaluation metrics (ECFP fingerprints, Tanimoto distance, Internal Diversity) are defined in Appendix A.3.

REFERENCES

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In *The Twelfth International Conference on Learning Representations*, 2023.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 214–223. JMLR.org, 2017.

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k -means clustering via lightweight coresets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pp. 1119–1127, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219973. URL https://doi.org/10.1145/3219819.3219973.

Thomas Back and H-P Schwefel. Evolutionary computation: An overview. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 20–29. IEEE, 1996.

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.

Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

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.

Thomas Blaschke, Josep Arús-Pous, Hongming Chen, Christian Margreitter, Christian Tyrchan, Ola Engkvist, Kostas Papadopoulos, and Atanas Patronov. Reinvent 2.0: an ai tool for de novo drug design. *Journal of chemical information and modeling*, 60(12):5918–5922, 2020a.

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, 2020b.

Regine S. Bohacek, Colin McMartin, and Wayne C. Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal Research Reviews*, 16(1):3–50, 1996. doi: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6.

Albert Bou, Morgan Thomas, Sebastian Dittert, Carles Navarro, Maciej Majewski, Ye Wang, Shivam Patel, Gary Tresadern, Mazen Ahmad, Vincent Moens, et al. Acegen: Reinforcement learning of generative chemical agents for drug discovery. *Journal of Chemical Information and Modeling*, 2024.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

Artem Cherkasov, Eugene N Muratov, Denis Fourches, Alexandre Varnek, Igor I Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C Martin, Roberto Todeschini, et al. Qsar modeling: where have you been? where are you going to? *Journal of medicinal chemistry*, 57(12): 4977–5010, 2014.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Carlos A Coello Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

Elias James Corey and Xue-Min Cheng. *The Logic of Chemical Synthesis*. John Wiley & Sons, New York, 1995. ISBN 0-471-11594-0.

David W. Cushman and Miguel A. Ondetti. Design of angiotensin converting enzyme inhibitors. *Nature Medicine*, 5(10):1110–1112, 1999. doi: 10.1038/13423.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pp. 3–34. Springer, 2011.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

Peter Ertl and Ansgar Schuffenhauer. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1:1–11, 2009.

Dan Feldman. Core-sets: Updated survey. In *Sampling techniques for supervised or unsupervised tasks*, pp. 23–44. Springer, 2019.

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.

James A Foster. Evolutionary computation. *Nature Reviews Genetics*, 2(6):428–436, 2001.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor Coley. Sample efficiency matters: a benchmark for practical molecular optimization. *Advances in neural information processing systems*, 35: 21342–21357, 2022.

Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

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.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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.

Jeff Guo, Franziska Knuth, Christian Margreitter, Jon Paul Janet, Kostas Papadopoulos, Ola Engkvist, and Atanas Patronov. Link-invent: generative linker design with reinforcement learning. *Digital Discovery*, 2(2):392–408, 2023.

Xin Guo, Zhiheng Xi, Yiwen Ding, Yitao Zhai, Xiaowei Shi, Xunliang Cai, Tao Gui, Qi Zhang, and Xuanjing Huang. Counteracting matthew effect in self-improvement of lvlms through head-tail re-balancing. *arXiv preprint arXiv:2510.26474*, 2025.

Thomas A Halgren. Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.

James B Hendrickson. Concepts and applications of molecular similarity. *Science*, 252(5009): 1189–1190, 1991.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Xiuyuan Hu, Guoqing Liu, Yang Zhao, and Hao Zhang. De novo drug design using reinforcement learning with multiple gpt agents. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiuyuan Hu, Guoqing Liu, Can Chen, Yang Zhao, Hao Zhang, and Xue Liu. 3DMolformer: A dual-channel framework for structure-based drug discovery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=RgE1qiO2ek.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 1051–1068, 2023.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7): 1757–1768, 2012.

Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.

William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. URL https://api.semanticscholar.org/CorpusID:117819162.

William L. Jorgensen. The many roles of computation in drug discovery. *Science*, 303(5665): 1813–1818, 2004. doi: 10.1126/science.1096361. URL https://www.science.org/doi/abs/10.1126/science.1096361.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30: 595–608, 2016.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395, 2021.

David C Kombo and Matthew J LaMarche. The logic of chemical optimization. *Journal of Medicinal Chemistry*, 2025.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

Pedro Larrañaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2001.

Christopher A. Lipinski, Franco Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced Drug Delivery Reviews*, 46(1-3):3–26, 2001. doi: 10.1016/s0169-409x(00)00129-0.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eYHoC5FX.

Xuhan Liu, Kai Ye, Herman WT van Vlijmen, Michael TM Emmerich, Adriaan P IJzerman, and Gerard JP van Westen. Drugex v2: de novo design of drug molecules by pareto-based multi-objective reinforcement learning in polypharmacology. *Journal of cheminformatics*, 13(1):85, 2021.

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.

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.

Yvonne C Martin, James L Kofron, and Linda M Traphagen. Do structurally similar molecules have similar biological activity? *Journal of medicinal chemistry*, 45(19):4350–4358, 2002.

Borja Medina, Alessandro Tibo, Jiazhen He, Jon Paul Janet, and Nicklas Österbacka. Diversity beats size scaling for chemical language models. *npj Drug Discovery*, 3(1):26, 2026.

Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. Neural architecture search without training. In *International conference on machine learning*, pp. 7588–7598. PMLR, 2021.

Amil Merchant, Simon Batzner, Samuel S. Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023. doi: 10.1038/s41586-023-06735-9.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016. URL http://arxiv.org/abs/1611.02163.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics*, 9:1–14, 2017.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rıza Özçelik and Francesca Grisoni. How evaluation choices distort the outcome of generative drug discovery. *Journal of Cheminformatics*, 17(1):169, 2025.

Rıza Özçelik, Helena Brinkmann, Emanuele Criscuolo, and Francesca Grisoni. Generative deep learning for de novo drug design—a chemical space odyssey. *Journal of Chemical Information and Modeling*, 65(14):7352–7372, 2025.

Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, 1993. doi: 10.1109/ACSSC.1993.342465.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

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

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *ICLR*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4780–4789, 2019.

Philipp Renz, Dries Van Rompaey, Jörg Kurt Wegner, Sepp Hochreiter, and Günter Klambauer. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, 32:55–63, 2019.

Philipp Renz, Sohvi Luukkonen, and Günter Klambauer. Diverse hits in de novo molecule design: Diversity-based comparison of goal-directed generators. *Journal of Chemical Information and Modeling*, 64(15):5756–5761, 2024.

Jean-Louis Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3):722–730, 2015.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12):2562–2574, 2015.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Marwin HS Segler, Thierry Kogej, Christian Tyrchan, and Mark P Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science*, 4(1): 120–131, 2018.

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.

Liangliang Shi, Gu Zhang, Haoyu Zhen, Jintao Fan, and Junchi Yan. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *International conference on machine learning*, pp. 31408–31421. PMLR, 2023.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross J. Anderson. The curse of recursion: Training on generated data makes models forget. *CoRR*, abs/2305.17493, 2023. URL https://doi.org/10.48550/arXiv.2305.17493.

Giovanni Squillero and Alberto Tonda. Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329: 782–799, 2016. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2015.09.056. URL https://www.sciencedirect.com/science/article/pii/S002002551500729X. Special issue on Discovery Science.

Hampus Gummesson Svensson, Christian Tyrchan, Ola Engkvist, and Morteza Haghir Chehreghani. Diversity-aware reinforcement learning for de novo drug design. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/1022. URL https://doi.org/10.24963/ijcai.2025/1022.

Xiangru Tang, Howard Dai, Elizabeth Knight, Fang Wu, Yunyang Li, Tianxiao Li, and Mark Gerstein. A survey of generative AI for de novo drug design: new frontiers in molecule and protein generation. *Briefings in Bioinformatics*, 25(4):bbae338, 2024.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 99–108, June 2022.

Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 13964–13999, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/24a8d40f6656e542f3fd43bac678e71b-Paper-Conference.pdf.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. URL https://doi.org/10.48550/arXiv.2302.13971.

Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. doi: 10.1109/TIT.2007.909108.

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.

Daniel F Veber, Stephen R Johnson, Hung-Yuan Cheng, Brian R Smith, Keith W Ward, and Kenneth D Kopple. Molecular properties that influence the oral bioavailability of drug candidates. *Journal of medicinal chemistry*, 45(12):2615–2623, 2002.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Cédric Vincent-Cuaz, Rémi Flamary, Marco Corneli, Titouan Vayer, and Nicolas Courty. Semi-relaxed gromov wasserstein divergence with applications on graphs. In *ICLR 2022-10th International Conference on Learning Representations*, pp. 1–28, 2022.

Sheng Wang, Yuzhi Guo, Yuhong Wang, Hongmao Sun, and Junzhou Huang. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 429–436, 2019.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Daniel S Wigh, Jonathan M Goodman, and Alexei A Lapkin. A review of molecular representation in the age of machine learning. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1603, 2022.

Weiwei Xiao, Yongyong Chen, Qiben Shan, Yaowei Wang, and Jingyong Su. Feature distribution matching by optimal transport for effective and robust coreset selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9196–9204, 2024.

Xueming Yan and Yaochu Jin. Emodm: A diffusion model for evolutionary multi-objective optimization. *arXiv preprint arXiv:2401.15931*, 2024.

Yuejiang Yu, Chun Cai, Jiayue Wang, Zonghua Bo, Zhengdan Zhu, and Hang Zheng. Uni-dock: Gpu-accelerated docking enables ultralarge virtual screening. *Journal of chemical theory and computation*, 19(11):3336–3345, 2023.

Donglin Zhan, Leonardo F. Toso, and James Anderson. Coreset-based task selection for sample-efficient meta-reinforcement learning. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pp. 4167–4173, 2025. doi: 10.1109/CDC57313.2025.11312186.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

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.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

A SMILES-BASED MOLECULAR REPRESENTATION: THE MULTI-MODAL GATEWAY

In this section, we elucidate the rationale behind selecting the *Simplified Molecular-Input Line-Entry System* (SMILES) (Weininger, 1988) as the foundational representation for our framework. Far from being a mere legacy format, SMILES functions as a versatile interlingua in computational chemistry. Its unique grammatical structure allows for deterministic transformation into diverse representation modalities which ranges from 1D numerical vectors to 3D geometric conformers. This enables our framework to "generate in language" while "evaluating in simulation."

A.1 THE GRAMMAR OF SMILES

SMILES serializes a molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ into a linear sequence of ASCII tokens via a depth-first traversal. This encoding is governed by a rigorous set of syntactic rules that implicitly capture full chemical information:

- **Atoms & Aromaticity:** Atoms are represented by standard abbreviations (e.g., `C`, `N`). Lowercase letters (e.g., `c`, `n`) explicitly denote aromaticity, encoding delocalized electron systems without requiring explicit bond definitions.
- **Topology via Branching:** Side chains are enclosed in parentheses "`()`". This recursive grammar allows a linear string to describe complex dendritic and branched structures losslessly.
- **Cyclicity via Ring Closures:** Rings are broken at a single bond, and the connection points are marked with numeric labels (e.g., `C1...C1`). This enables the representation of complex polycyclic and fused systems within a 1D sequence.

A.2 THE ECOSYSTEM ADVANTAGE: DATA ABUNDANCE FOR PRE-TRAINING

A decisive strategic advantage of adopting the SMILES representation is the unparalleled maturity of its data ecosystem. Unlike 3D conformational data, which requires computationally expensive quantum mechanical calculations or scarce crystallographic experiments (e.g., PDB), SMILES strings have been systematically curated for decades. This abundance serves as the fuel for *Self-Supervised Learning*, allowing sequence models to learn a robust "chemical grammar" on billion-scale datasets before engaging in task-specific optimization.

Our framework benefits directly from three tiers of SMILES repositories:

- **Synthesizable Space (ZINC):** The ZINC database (Irwin et al., 2012) contains over 230 million commercially available compounds. Training on ZINC ensures that the generative model implicitly learns the rules of *synthetic accessibility*, biasing the agent towards molecules that can actually be made in a wet lab.
- **Chemical Knowledge (PubChem):** As the world's largest collection of chemical information, PubChem (Kim et al., 2021) ($\sim$111 million entries) provides a diverse coverage of general chemical space, enabling the model to generalize across distinct structural classes.
- **Bioactivity Data (ChEMBL):** For drug discovery specifically, ChEMBL (Gaulton et al., 2012) offers manually curated bioactivity data for over 2 million compounds against $\sim 13,000$ targets. Fine-tuning on this subset aligns the model's prior with "drug-like" physicochemical properties.

This data-rich environment stands in stark contrast to graph-based or 3D-based approaches, which often suffer from data sparsity and must rely on smaller, synthetic benchmarks. By anchoring our framework in the SMILES ecosystem, we leverage the "Unreasonable Effectiveness of Data" to achieve superior generalization.

A.3 SMILES TO 1D: VECTORIZATION AND FINGERPRINTS

The most direct transformation of SMILES is into fixed-length numerical vectors, which facilitate rapid similarity calculation and machine learning integration.

Molecular Fingerprints (Explicit Features). Fingerprinting algorithms map the SMILES string into a bit-vector space $\{0,1\}^d$, encoding the presence or absence of specific substructures.

- **Extended-Connectivity Fingerprints (ECFP):** Also known as Morgan fingerprints, ECFP represents a widely accepted standard for encoding molecular topology. The algorithm operates through an iterative process:
  1. *Initialization:* Each atom in the SMILES-derived molecular graph is assigned an initial integer identifier based on its physicochemical properties (e.g., element, valence, and connectivity).
  2. *Iterative Update:* The algorithm collects circular neighborhoods of increasing radius R around each atom. In each iteration, an atom's identifier is updated by hashing it together with the identifiers of its immediate neighbors.
  3. *Duplicate Removal and Hashing:* After a specified number of iterations (e.g., radius 3 for ECFP6, the set of unique structural identifiers is mapped onto a fixed-length bit-vector (typically 1024 or 2048 bits) using a hash function.

  In our DySCo framework, ECFP6 serves as the rigorous basis for calculating *Tanimoto Similarity*, ensuring that our diversity metrics are grounded in chemical substructure overlaps rather than mere string edit distances.
- **MACCS Keys & RDKit Fingerprints:** These represent predefined dictionaries of structural keys (e.g., "Is there a carbonyl group?"). They provide a human-interpretable vectorization often used for coarse-grained filtering.

Tanimoto Distance. To quantify the structural divergence between two molecules x_i and x_j, we compute the Tanimoto distance based on their ECFP bit-vectors $\text{FP}(x)$:

$$\mathcal{D}_T(x_i, x_j) = 1 - \frac{|\text{FP}(x_i) \cap \text{FP}(x_j)|}{|\text{FP}(x_i) \cup \text{FP}(x_j)|}, \tag{14}$$

where the intersection $\cap$ and union $\cup$ operators correspond to bitwise AND and OR operations on the vectors, respectively. This metric provides an objective measure of topological dissimilarity: a distance near 1 indicates no shared substructures, while a distance near 0 suggests high structural similarity.

Internal Diversity (IntDiv). Building upon the pairwise Tanimoto distance, we quantify the overall structural diversity of a generated molecular set $\mathcal{S}$ using Internal Diversity (IntDiv). It is defined as the average Tanimoto distance between all unique pairs of molecules within the set:

$$\text{IntDiv}(\mathcal{S}) = \frac{2}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{\substack{x_i, x_j \in \mathcal{S} \\ i<j}} \mathcal{D}_T(x_i, x_j), \tag{15}$$

where $\mathcal{D}_T(x_i, x_j)$ is computed via Eq. 14. A higher IntDiv score indicates that the generative policy is successfully exploring a broad, structurally heterogeneous chemical space rather than collapsing onto a redundant set of scaffolds.

Normalized Hamming Distance. As an alternative to Tanimoto distance, the normalized Hamming distance measures the fraction of differing bits between two fingerprint vectors:

$$\mathcal{D}_H(x_i, x_j) = \frac{1}{d} \sum_{k=1}^{d} \mathbf{1}[\text{FP}(x_i)_k \neq \text{FP}(x_j)_k], \tag{16}$$

where d is the fingerprint dimensionality. While Tanimoto distance focuses on the overlap of active ("on") bits and is invariant to shared inactive bits, Hamming distance treats all bit positions equally. In DySCo, we provide both options: Tanimoto-based variants prioritize topological scaffold diversity, while Hamming-based variants operate directly on the binary representation.

Learned Embeddings (Latent Features). Beyond hard-coded fingerprints, SMILES strings can be processed by Neural Language Models (e.g., Transformers, LSTMs) to produce continuous dense vectors $\mathbf{z} \in \mathbb{R}^k$.

- **Mechanism:** Models like SMILES-BERT (Wang et al., 2019) or ChemBERTa (Chithrananda et al., 2020) treat SMILES tokens as words. The resulting embeddings capture latent physicochemical properties (e.g., solubility, acidity) implicitly learned from massive unsupervised corpora (e.g., PubChem) (Kim et al., 2021).

A.4 SMILES TO 2D: TOPOLOGICAL GRAPHS AND DESCRIPTORS

SMILES strings contain the complete connectivity table of a molecule, allowing for immediate and lossless parsing into 2D Molecular Graphs.

Molecular Graphs. A SMILES string can be parsed into an Adjacency Matrix $\mathbf{A}$ and a Node Feature Matrix $\mathbf{X}$.

- **Conversion:** Cheminformatics libraries (e.g., RDKit (Landrum et al., 2013)) parse the string to reconstruct the bond orders and atom types. Implicit hydrogens are added based on valence rules.
- **Application:** This graph representation is the prerequisite for Graph Neural Networks (GNNs) (Kearnes et al., 2016) and for calculating topological invariants such as **Graph Diameter** or **Scaffold Analysis** (e.g., Bemis-Murcko scaffolds (Bemis & Murcko, 1996)), which are essential for categorizing the structural diversity of our generated molecules.

Physicochemical Descriptors. Once the 2D topology is reconstructed, a plethora of property calculators can be applied.

- **QED & SA Score:** Metrics like the *Quantitative Estimate of Drug-likeness* (QED) (Bickerton et al., 2012) and *Synthetic Accessibility* (SA) score (Ertl & Schuffenhauer, 2009) are derived directly from the 2D graph structure. These serve as the primary reward signals in our RL loop.

A.5 SMILES TO 3D: GEOMETRIC CONFORMERS

Perhaps the most critical capability is the projection of 1D SMILES into 3D conformational space. Although SMILES does not explicitly store coordinates, it contains the connectivity constraints necessary to solve the Distance Geometry Problem.

Conformer Generation. The transformation from SMILES to a 3D structure (Conformer) involves optimizing spatial coordinates (x, y, z) for each atom to satisfy bond length and angle constraints.

- **ETKDG (Experimental-Torsion Knowledge Distance Geometry):** This state-of-the-art algorithm combines distance geometry with experimental torsion angle preferences derived from the Cambridge Structural Database (CSD) (Riniker & Landrum, 2015). It rapidly generates realistic 3D starting structures from SMILES inputs.
- **Force Field Minimization:** The raw conformers are further relaxed using force fields (e.g., MMFF94 (Halgren, 1996) or UFF (Rappé et al., 1992)) to find local energy minima.

Structure-Based Evaluation (Docking). This 1D-to-3D pipeline allows our text-based agent to interface with structure-based drug design (SBDD) workflows.

- **Application:** In our framework, generated SMILES are converted to 3D poses and docked into protein pockets utilizing tools like *AutoDock Vina* (Trott & Olson, 2010) or *Uni-Dock* (Yu et al., 2023). This ensures that while our optimization happens in the efficient 1D language manifold, the final selection is validated in the rigorous 3D physical manifold.

B Related Work

The Self-Evolving Optimization Paradigm. In practical molecular design, candidate molecules that perfectly meet a desired multi-objective profile are rarely available a priori. This lack of pre-labeled data motivates a *self-evolving* optimization paradigm, where a generative model iteratively samples candidate molecules, evaluates them via task-specific scoring functions (or property predictors), and updates its policy using the resulting feedback (Liu et al., 2021; 2023; Loeffler et al., 2024; Popova et al., 2018; Blaschke et al., 2020a; Gómez-Bombarelli et al., 2018).

In modern implementations of this sample-score-update loop, memory-assisted mechanisms are frequently integrated to improve training stability and sample efficiency. A prominent technique is *Experience Replay* (ER). As systematically evaluated by Svensson et al. (Gummesson Svensson et al., 2024), ER involves maintaining a buffer of previously generated molecules alongside their scores. By reusing both high-scoring and occasionally low-scoring historical samples during the policy update, ER helps off-policy and on-policy algorithms learn a more robust decision boundary (specific ER taxonomy and configurations are detailed in Appendix D.7).

Alongside ER, *Diversity Filters* are commonly employed to actively manage structural variety (Blaschke et al., 2020a). A diversity filter acts as a dynamic registry that tracks generated molecular scaffolds. Once a specific scaffold exceeds a predefined capacity within the memory bucket, any newly generated molecule sharing this scaffold is heavily penalized (typically assigned a reward of zero). This forces the agent to redirect its focus toward uncharted chemical regions, structurally diversifying the output. The configuration of this self-evolving loop depends on the specific generative task. We detail the various generative contexts and their corresponding task definitions in Appendix D.5.

Optimal Transport. Optimal Transport (OT) provides a rigorous mathematical framework for computing the optimal matching between two sets of points, conceptually treated as "supply" and "demand" (Villani et al., 2008; Peyré et al., 2019). Suppose the numbers of supply and demand nodes are m_1 and m_2, respectively. For the supply nodes, we define an m_1-dimensional vector $p = [p_1, p_2, \ldots, p_{m_1}]$ with each entry $p_i > 0$ representing the amount of resource available; similarly, for the demand nodes, we define an m_2-dimensional vector $q = [q_1, q_2, \ldots, q_{m_2}]$, where $q_j > 0$ represents the required resource amount. The classical balanced formulation requires strict mass conservation on both sides, i.e., $\sum_{i=1}^{m_1} p_i = \sum_{j=1}^{m_2} q_j$. Furthermore, given an $m_1 \times m_2$ cost matrix C, where each entry $C_{i,j}$ indicates the unit transportation cost, the OT objective is to find a non-negative transportation plan Π that minimizes the total cost $\sum_{i,j} \Pi_{i,j} C_{i,j}$ subject to these dual marginal constraints.

However, enforcing strict marginal constraints on both sides can be overly rigid for practical machine learning tasks, such as subset selection, where exactly reproducing a target distribution with redundant nodes is unnecessary. Moreover, computing exact OT is computationally expensive. To resolve this, recent advances utilize *KL-regularized semi-relaxed transport* (Cuturi, 2013; Vincent-Cuaz et al., 2022; Shi et al., 2023). Conceptually, the *semi-relaxed* formulation drops the strict mass conservation requirement on one of the marginals (typically the target side). This allows the supply nodes to flexibly cover the most relevant demand nodes without being forced into rigid, redundant allocations. Simultaneously, *KL-regularization* (entropic smoothing) transforms the deterministic matching into a probabilistic "soft" assignment. Together, this combination not only provides a more flexible matching structure but also allows the optimal coupling to bypass iterative solvers and admit a highly efficient closed-form solution.

Beyond its classical applications in economics and computer vision, OT has been increasingly adopted in machine learning for tasks such as distribution matching, domain adaptation, and representative subset selection (Xiao et al., 2024). In the context of our work, OT offers a principled way to evaluate the complementary structural value of generated candidates. The detailed theoretical application of OT to our molecular curation task is provided in Appendix C. Based on this formulation, we introduce our topological balancing strategies in Section 3.2.

C Theoretical Foundations: Geometric Data Curation via Optimal Transport

In this section, we establish the theoretical foundations underlying our data curation strategy from a geometric optimal transport perspective. Rather than relying on distributionally robust optimization, our approach leverages KL-regularized transport with a reward-weighted reference measure to characterize structural alignment between candidate molecules and high-reward anchors.

C.1 Discrete Optimal Transport Formulation

Let $\mathcal{X} = \{x_1, \ldots, x_M\}$ denote the available corrective candidate pool, instantiated in DySCo as the remaining prior-sampled molecules, and let $\mathcal{Y} = \{y_1, \ldots, y_N\}$ denote a set of high-reward anchor molecules obtained via reward-based selection. We associate $\mathcal{X}$ and $\mathcal{Y}$ with discrete probability measures $\mu = \sum_{i=1}^{M} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{N} b_j \delta_{y_j}$, where $a_i = 1/M$ represents a uniform prior over candidates, and $b_j \propto \exp(R(y_j)/\tau)$ is a reward-weighted distribution.

Given a ground cost matrix $C \in \mathbb{R}_+^{M \times N}$, where $C_{ij} = d(x_i, y_j)$, the KL-regularized transport problem is defined as:

$$W_\varepsilon^{\mathrm{KL}}(\mu, \nu) = \min_{\Pi \in \mathcal{U}} \left\{ \langle \Pi, C \rangle + \varepsilon \mathrm{KL}\big(\Pi \,\big\|\, ab^\top\big) \right\}, \tag{17}$$

where $\mathrm{KL}\big(\Pi \,\big\|\, ab^\top\big) = \sum_{i,j} \Pi_{ij} \log \frac{\Pi_{ij}}{a_i b_j}$ up to constants independent of Π. The structure of the optimal coupling $\Pi^\star$ depends critically on the constraint set $\mathcal{U}$. We analyze three regimes relevant to our curation strategy.

C.2 Marginal Relaxations and Closed-Form Solutions

In the canonical OT formulation, strict mass conservation is enforced on both distributions, which typically requires iterative numerical solvers. However, as identified in recent distribution matching tasks (Shi et al., 2023; Vincent-Cuaz et al., 2022), relaxing certain marginal constraints allows the iterative procedures to be avoided while better handling structural redundancies.

Fully Constrained OT (Sinkhorn). When both marginals are strictly enforced, $\mathcal{U} = \{\Pi \in \mathbb{R}_+^{M \times N} \mid \Pi \mathbf{1}_N = a,\ \Pi^\top \mathbf{1}_M = b\}$. In this standard formulation, no closed-form solution exists. The unique solution takes the form:

$$\Pi_{ij}^\star = u_i \exp(-C_{ij}/\varepsilon) v_j, \tag{18}$$

where u and v are scaling vectors found via the iterative **Sinkhorn algorithm** (Cuturi, 2013):

$$u^{(k+1)} = a \oslash (K v^{(k)}), \quad v^{(k+1)} = b \oslash (K^\top u^{(k+1)}), \tag{19}$$

with $K_{ij} = \exp(-C_{ij}/\varepsilon)$. This method is used when precise matching to the target distribution b is required.

Semi-Relaxed OT (Source Constraint Only). In our **Strategy III**, we relax the target constraint to allow flexibility in how anchors are covered, enforcing only the source constraint (mass conservation of candidates) (Vincent-Cuaz et al., 2022). The constraint set becomes $\mathcal{U} = \{\Pi \in \mathbb{R}_+^{M \times N} \mid \Pi \mathbf{1}_N = a\}$. This formulation admits a **closed-form solution** based on the Gibbs kernel modulated by the target mass (Shi et al., 2023):

$$\Pi_{ij}^\star = a_i \frac{b_j \exp(-C_{ij}/\varepsilon)}{\sum_{k=1}^{N} b_k \exp(-C_{ik}/\varepsilon)}. \tag{20}$$

Proof Sketch. Under the source constraint $\sum_j \Pi_{ij} = a_i$, the Lagrangian is

$$\mathcal{L} = \sum_{i,j} \Pi_{ij} C_{ij} + \varepsilon \sum_{i,j} \Pi_{ij} \log \frac{\Pi_{ij}}{a_i b_j} + \sum_i \lambda_i \left(\sum_j \Pi_{ij} - a_i \right).$$

The first-order condition gives

$$C_{ij} + \varepsilon \log \frac{\Pi_{ij}}{a_i b_j} + \lambda_i = 0.$$

Thus

$$\Pi_{ij} = a_i b_j \exp(-C_{ij}/\varepsilon) \exp(-\lambda_i/\varepsilon).$$

Normalizing each row by $\sum_j \Pi_{ij} = a_i$ yields Eq. 20.

Fully Relaxed OT. If we relax both marginal constraints, requiring only total mass conservation ($\sum_{i,j} \Pi_{ij} = 1$), the solution becomes a global Gibbs distribution:

$$\Pi^\star_{ij} = \frac{a_i b_j \exp(-C_{ij}/\varepsilon)}{\sum_{p,q} a_p b_q \exp(-C_{pq}/\varepsilon)}. \tag{21}$$

This represents unconstrained matching based on geometric proximity modulated by the reward-weighted reference measure.

C.3 Marginal Transport Cost as a Geometric Curation Criterion

For our curation strategy, we use the row-normalized association $q^\star_{ij} = \Pi^\star_{ij}/a_i$ from Eq. 20 to define the *marginal transport score* $\mathcal{K}(i)$ for each candidate x_i:

$$\mathcal{K}(i) = \sum_{j=1}^{N} q^\star_{ij} C_{ij}. \tag{22}$$

Geometrically, $\mathcal{K}(i)$ measures the expected topological distance from candidate x_i to the elite set under a reward-weighted local comparison. Maximizing this score selects candidates that remain structurally distant under the reward-aware association induced by the KL-regularized transport plan.

D Supplementary Experimental Details

In this section, we provide the complete parameter configurations, task definitions, and scoring functions to ensure the reproducibility of our results. To comprehensively evaluate DySCo, we designed distinct optimization landscapes for each framework, ranging from target-specific binding to general physicochemical property constraints.

D.1 Implementation and Computing Environment

All experiments were conducted on a high-performance workstation equipped with a single **NVIDIA GeForce RTX 3090 GPU** (24GB VRAM) and a 63-core CPU. To ensure fair comparisons across different architectures, we utilized standard pre-trained priors and specific regularization schemes as follows:

Regularization Formulations (Ω). The regularization term in Eq. 13 varies across frameworks to balance exploration and exploitation:

- **REINVENT 4:** Employs an augmented likelihood squared error, $\Omega_{\text{REINVENT}}(x) = (\log P_\theta(x) - [\log P_{\text{prior}}(x) + \sigma R(x)])^2$, where σ modulates the reward's influence on the likelihood landscape.
- **DrugEx (v2 & v3):** Typically utilizes a prior likelihood penalty to constrain the agent within the valid chemical space, formulated as $\Omega_{\text{DrugEx}}(x) = \log P_\theta(x) - \log P_{\text{prior}}(x)$.

Framework-Specific Configurations.

- **DrugEx v2 Framework:** We utilized the RNN-based generator (LSTM) initially pre-trained on the **ChEMBL** database (version 26, $\sim$1.7 million molecules) (Liu et al., 2021). Before reinforcement learning, the generator was **fine-tuned** on the **LIGAND** set (bioactive ligands for the targets of interest) to serve as the initial policy.

- **DrugEx v3 Framework:** For graph-based generation, we employed the **Graph Transformer** (Liu et al., 2023), which constructs molecules via iterative editing on a specialized `VocGraph` vocabulary. The model was pre-trained on the **Papyrus** dataset (v05.5) and subsequently **fine-tuned** on the **A2AR** ligand set.
- **REINVENT 4 Framework:** For all tasks (De Novo, LinkInvent, LibInvent, Mol2Mol), we used the standard priors provided by the framework pre-trained on **ChEMBL** (Gaulton et al., 2012), providing a robust chemical syntax foundation.

D.1.1 Empirical Analysis: The Bottleneck of Classic Coresets in RL

As illustrated in Table 3, these advanced coreset algorithms successfully formulated a robust baseline group. They effectively approximated the generative quality and structural diversity (IntDiv $\approx 0.87 - 0.89$) of the full-batch Vanilla RL, proving the soundness of their respective sampling mechanisms. However, they failed to transcend the Pareto frontier and exhibited severe computational bottlenecks. We attribute this to two fundamental RL-specific limitations:

- **The Transient Non-stationarity Limit:** Unlike continual learning datasets, the generated batches in online RL are "disposable" slices of a constantly moving policy distribution. Even if CSReL or META perfectly compresses the current samples, this compressed subset remains bounded by the biased exploration horizon of the current fragile policy. Perfectly summarizing a suboptimal batch does not intrinsically expand the global exploration boundary.
- **The Variance Dilemma in PG Matching:** REINFORCE-based policy gradients exhibit notoriously high variance. Attempting to execute precise mathematical operations—such as OMP matching in GCR or Facility Location in META—within a noisy, stochastic gradient space inevitably loses fine-grained structural optimization signals. Furthermore, calculating per-sample gradients and running these complex optimizations incurs massive computational overhead (e.g., 9861s for GCR and 7398s for META, as detailed in Table 3), rendering them practically prohibitive for fast-paced generative RL loops.

In contrast, DySCo bypasses these mathematical overheads entirely by anchoring curation to a static, pre-scored prior and directly operating in the topological state space (via Optimal Transport), thereby achieving superior Pareto efficiency in both structural diversity and runtime (e.g., 612s).

D.2 DrugEx Optimization Landscapes

We implemented two scenarios using the DrugEx native scoring engine, focusing on Lead Optimization and "Rule-of-5" compliance.

D.2.1 Scenario A: A2AR Polypharmacology (2 Objectives)

This task simulates a specific lead optimization campaign targeting the Adenosine A_{2A} Receptor (A2AR). The goal is to maximize biological affinity while strictly constraining synthetic feasibility.

- **A2AR Affinity (S_{act}):** *Rationale:* Drives the generator toward bioactive regions specific to the target protein. *Implementation:* We utilized a Random Forest QSAR model ($R^2 > 0.8$) trained on bioactive data. The output probability is processed via a `ClippedScore` ($L = 0.2, H = 0.8$) to filter low-confidence noise.
- **Synthetic Accessibility (S_{SA}):** *Rationale:* Prevents the generation of "fantasy molecules" that are theoretically active but practically unsynthesizable. *Implementation:* Calculated via RDKit based on fragment complexity. We utilize a `SmoothClippedScore` with bounds $[5.0, 3.0]$, rewarding molecules with SA score < 3 (easy to synthesize) and penalizing those > 5.

Evaluation Protocol: Evaluated using the **Linear Sweep Protocol** (9 vectors).

D.2.2 SCENARIO B: GENERAL DRUG-LIKENESS (3 OBJECTIVES)

This scenario acts as a broad "chemical hygiene" filter. It enforces physicochemical properties aligned with **Lipinski's Rule of 5** (Lipinski et al., 2001), ensuring generated candidates possess favorable oral bioavailability profiles.

- **Molecular Weight (MW):** *Rationale:* Limits size to ensure bioavailability. Heavy molecules (> 500 Da) often suffer from poor absorption. *Implementation:* `SmoothClippedScore` ($L = 500, H = 100$). Linearly penalizes MW > 500.
- **LogP (Lipophilicity):** *Rationale:* Maintains the balance between water solubility and membrane permeability. *Implementation:* `SmoothHump` ($L = 2.0, H = 5.0, \sigma = 0.5$). Unlike simple thresholds, this Gaussian-like function targets a "sweet spot," penalizing both excessive hydrophobicity (LogP > 5) and hydrophilicity (LogP < 2).
- **H-Bond Acceptors (HBA):** *Rationale:* Excessive polar centers hinder passive diffusion across cell membranes. *Implementation:* `SmoothClippedScore` ($L = 10, H = 5$). Penalizes HBA counts exceeding 5, with a hard cutoff at 10.

Evaluation Protocol: Evaluated using the **Simplex Sweep Protocol** (13 vectors).

D.3 REINVENT OPTIMIZATION LANDSCAPES

For the REINVENT framework, we implemented standard medicinal chemistry filters to test the agent's ability to navigate smooth, composite component landscapes.

D.3.1 SCENARIO C: ORAL BIOAVAILABILITY (2 OBJECTIVES)

This task screens for orally active candidates using high-level composite metrics.

- **QED:** *Quantitative Estimate of Drug-likeness*. A holistic metric aggregating solubility, polarity, and lipophilicity. We maximize this directly.
- **SlogP:** *Lipophilicity Control*. *Rationale:* Excessive lipophilicity correlates with toxicity and promiscuity. *Implementation:* **Reverse Sigmoid** ($L = 5, H = 0, k = 0.5$). Creates a "soft cap" that heavily penalizes SlogP > 5.

Evaluation Protocol: Evaluated using the **Linear Sweep Protocol** (9 vectors).

D.3.2 SCENARIO D: LEAD-LIKENESS (3 OBJECTIVES)

This scenario enforces stricter geometric constraints based on **Veber's Rules** (Veber et al., 2002), aiming for compact, rigid, and permeable scaffolds suitable for subsequent lead optimization.

- **Molecular Weight (MW):** *Rationale:* Confines molecules to a "sweet spot" ($200 - 500$ Da) for lead-like specificity. *Implementation:* **Double Sigmoid** ($L = 200, H = 500$). Scores decay rapidly outside this window.
- **TPSA:** *Topological Polar Surface Area*. *Rationale:* A proxy for intestinal absorption. *Implementation:* **Double Sigmoid** ($L = 0, H = 140$). Limits polar surface area to < 140 Å^2.
- **Graph Length:** *Rationale:* Penalizes long, flexible aliphatic chains which are entropically unfavorable for binding and metabolically unstable. *Implementation:* **Reverse Sigmoid** ($L = 20, H = 50, k = 0.1$). Encourages structural compactness by penalizing paths exceeding 20 bonds.

Evaluation Protocol: Evaluated using the **Simplex Sweep Protocol** (13 vectors).

D.4 SYSTEMATIC PREFERENCE EVALUATION PROTOCOLS

To verify that DySCo is agnostic to the reward topology, we systematically swept the preference vectors $\mathbf{w}$ for both 2-objective and 3-objective tasks.

1. Linear Sweep Protocol (for 2-Objective Tasks). For Scenarios A and C, we evaluated 9 linear combinations $\mathbf{w} = (w_1, w_2)$ where $w_1 \in \{0.1, 0.2, \ldots, 0.9\}$ and $w_2 = 1 - w_1$. This validates the model's ability to navigate the Pareto front between two competing objectives (e.g., Potency vs. Synthesizability).

2. Simplex Sweep Protocol (for 3-Objective Tasks). For Scenarios B and D, we evaluated 13 distinct vectors $\mathbf{w} = (w_1, w_2, w_3)$ chosen to systematically cover the decision simplex. As defined in Eq. 23, this set includes **Corner Cases** (single-objective dominance), **Hierarchical Cases** (asymmetric prioritization), and **Center Cases** (balanced optimization).

$$W_{set} = \left\{ \begin{array}{l} (0.1, 0.1, 0.8), (0.1, 0.8, 0.1), (0.8, 0.1, 0.1), \\ (0.1, 0.3, 0.6), (0.3, 0.6, 0.1), (0.6, 0.3, 0.1), \\ (0.2, 0.5, 0.3), (0.25, 0.3, 0.45), (0.35, 0.5, 0.15), \\ (0.45, 0.1, 0.45), \\ (0.3, 0.3, 0.4), (0.3, 0.4, 0.3), (0.4, 0.3, 0.3) \end{array} \right\} \tag{23}$$

D.5 Generative Contexts and Task Definitions

Building upon the unified formulation presented in Section 2, where molecular design is modeled as a conditional generation task $p_\theta(x \mid c)$, we define the specific nature of the context variable c for each REINVENT generation mode. This mapping demonstrates how DySCo's state-adaptive mechanism generalizes across different structural constraints, from unconstrained exploration to strict substructure preservation.

1. De Novo Design (Unconstrained Exploration)(Blaschke et al., 2020a; Liu et al., 2021). *Context Mapping:* $c = \emptyset$. In this foundational setting, the context is an empty set. The generative policy π_θ is initialized with a generic prior (e.g., learned from ChEMBL) and must generate molecules from scratch, token by token. The agent operates without explicit structural anchors, driven solely by the reward landscape (e.g., the 2-Objective Bioavailability task). This tests the algorithm's ability to navigate the vast, unconstrained chemical space without succumbing to mode collapse.

2. LinkInvent (Fragment Connectivity)(Guo et al., 2023). *Context Mapping:* $c = \{\mathcal{G}_1, \mathcal{G}_2\}$. Here, the context comprises a set of disjoint pharmacophores or "warheads" annotated with attachment points (*). The model functions as a linker generator, tasked with bridging these segments into a single cohesive entity.

- **Example Input (c):** `Oc1cncc(*)c1|*c1ccoc1`

Rather than free generation, the agent must satisfy strict semantic constraints—maintaining valency and ensuring valid ring closures between the specified anchor points—while simultaneously optimizing the "Lead-Likeness" objectives.

3. LibInvent (Substructure Preservation)(Fialková et al., 2021). *Context Mapping:* $c = \mathcal{G}_{\text{core}}$. The context represents a mandatory bioactive scaffold containing specific growth vectors. This formulation strictly limits the search space to the decoration of the core, enforcing "Substructure Preservation."

- **Example Input (c):** `[*:1]Cc2ccc1cncc(C[*:2])c1c2`

The core structure remains immutable throughout the generation process. The agent's role is to explore the chemical space of valid R-groups (substituents) at the defined positions, optimizing properties such as solubility and permeability while retaining the scaffold's binding geometry.

4. Mol2Mol (Analog Generation)(Olivecrona et al., 2017). *Context Mapping:* $c = x_{\text{ref}}$. In this mode, c denotes a complete reference compound (seed molecule) that anchors the generative trajectory. This setup targets Lead Optimization, where the goal is local property refinement.

- **Example Input (c):** `O=S(=O)(c1cccc2cnccc12)N1CCCNCC1` (CHEMBL38380)

The agent must balance *exploitation*—staying close to the seed's high-performing features—with *exploration* to modify the structure sufficiently for score improvement. DySCo's diversity injection is particularly critical here to prevent the policy from collapsing back to the seed molecule itself or getting trapped in a narrow neighborhood of trivial analogues.

5. DrugEx v3 (Graph-Based Scaffold Design)(Liu et al., 2023). *Context Mapping:* $c = \mathcal{G}_{\text{frag}}$. Distinct from the sequence-based paradigms above, this task evaluates DySCo on a **Graph Transformer** architecture. Here, the context c is defined as a set of molecular graph fragments (scaffolds) loaded via the `GraphFragDataSet`.

- **Example Input (c):** A set of disjoint graph fragments initialized from SMILES, e.g., `{"c1ccncc1", "C1CC1"}` (representing Pyridine and Cyclopropane rings).

Instead of predicting next-tokens, the agent iteratively expands the graph by predicting atom and bond additions to the input fragments. This setting is pivotal for our analysis: it verifies that DySCo's "State-Adaptive Curation" mechanism is topologically agnostic, effectively mitigating mode collapse even when the underlying generative process shifts from 1D sequential autoregression to 2D graph editing.

D.6 Adaptation of Advanced Coreset Baselines to Generative RL

In traditional supervised and continual learning, coreset algorithms typically rely on static ground-truth labels and stationary loss landscapes (e.g., cross-entropy). However, in *de novo* molecular design via Reinforcement Learning, the data distribution is inherently non-stationary (as the policy continuously evolves), and explicit ground-truth labels are absent. To construct rigorous, fair, and competitive baselines, we mathematically reformulated several state-of-the-art subset selection methods originally designed for other domains, adapting them specifically to our online generative RL framework.

1. InfoBatch and Sorted Loss. For training-signal-driven methods, we directly substituted the standard supervised loss with the surrogate policy objective. **InfoBatch** (Qin et al., 2024) probabilistically prunes samples with low policy loss, dynamically rescaling the gradients of retained samples to maintain an unbiased gradient expectation. **Sorted Loss** acts as a deterministic counterpart, greedily preserving the top-K samples with the highest absolute policy loss in the current epoch.

2. Gradient Coreset Replay (GCR). GCR (Tiwari et al., 2022) was originally proposed to mitigate catastrophic forgetting in continual learning by matching the mean gradient of a replay buffer. We adapted this cross-entropy gradient matching into **Policy Gradient Matching**. To circumvent the dimensionality curse associated with large language models, we extracted gradients strictly from the final linear layers of the policy network. We then applied Random Projection (via the Johnson-Lindenstrauss Lemma (Johnson & Lindenstrauss, 1984; Pati et al., 1993; Tropp & Gilbert, 2007)) to reduce the gradient dimensionality to 1024 while preserving angular relationships. Finally, Orthogonal Matching Pursuit (OMP) (Tiwari et al., 2022) was employed to select a weighted subset of molecules whose combined policy gradient optimally approximates the mean policy gradient of the entire generated batch.

3. Coreset Selection via Reducible Loss (CSReL). CSReL (Tong et al., 2025) posits that selecting samples with the highest loss often leads to a "noise trap," as ambiguous samples are inherently unlearnable. In molecular RL, a generated SMILES sequence with a massive surrogate loss or extreme advantage is frequently an invalid or nonsensical structure. We elegantly adapted this concept by utilizing the **pre-trained prior network as an irreducible loss proxy**. The reducible loss (ReL) for a sequence s was redefined as $\text{ReL}(s) = \log P_{\text{prior}}(s) - \log P_{\text{agent}}(s)$. We then scaled this spatial margin by the positive advantage $\max(A(s), 0)$ to compute a priority score. This adaptation enabled highly efficient, gradient-free, value-driven selection that naturally filters out chemical noise.

4. Gradient-Space Diversity Selection (META). Originally designed to select representative tasks in Meta-RL (Zhan et al., 2025), we adapted META for **sample-level subset selection**. Recognizing that simply matching mean gradients (like GCR) might cause conflicting gradient directions to cancel out to zero, this adaptation prioritizes *Gradient-Space Diversity*. Using the projected individual policy gradients (similar to our GCR adaptation), we employed greedy Facility Location Submodular Maximization to select K molecules that maximize spatial coverage across all evolutionary directions

in the gradient space. Unselected molecules dynamically transferred their voting weights to the nearest selected "representative" in the gradient space (Voronoi partitioning).

5. KMeans (Score). We implemented a geometry- and value-aware clustering baseline. KMeans was applied to the joint space of normalized ECFP6 fingerprints and scalar reward scores to ensure that the selected subset forms a topologically and functionally representative skeleton of the current generative batch.

D.7 Experience Replay and Diversity Mechanisms

We benchmark DySCo against a structured taxonomy of strategies, distinguishing between memoryless (Current) and memory-augmented (History) approaches.

1. Experience Replay Strategies. Following the notation by Svensson et al. (Gummesson Svensson et al., 2024), we evaluate combinations of source (History 'H' vs. Current 'C') and sampling logic:

- **AC (All-Current):** Vanilla RL baseline; uses the entire current batch.
- **BH (Binned-History):** Samples uniformly from a history buffer stratified by reward bins (0.1 increments).
- **TBH (Top-Bottom-History):** A robust baseline that replays a mix of top-performing elites and low-scoring failures from history to clarify the decision boundary.
- **TH (Top-History):** Replays only the highest-scoring historical samples (Risk of feedback loops).
- **TBC/BC/TC:** Analogous strategies applied only to the current batch without a persistent history buffer.

2. Diversity Filters. We enable the `IdenticalMurckoScaffold` filter, a standard mechanism in the REINVENT framework designed to actively prevent mode collapse during generative optimization.

- **Mechanism:** In medicinal chemistry, a Bemis-Murcko scaffold represents the core functional framework of a molecule—comprising its macrocyclic ring systems and connecting linker bonds—stripped of its terminal side chains. The filter maintains a dynamic, first-in-first-out (FIFO) memory buffer of recently generated scaffolds with a capacity of `bucket_size=25`. If a newly generated molecule shares its core scaffold with any entry currently residing in this memory buffer, it is flagged as structurally redundant. This effectively penalizes the agent for repeatedly exploiting a single, easily optimized chemical series (i.e., making trivial side-chain decorations on the same backbone).
- **Thresholds and Penalties:** To rigidly enforce exploration, the filter utilizes strict gating criteria: `Minimum score` $= 0.4$ and `Minimum similarity` $= 0.4$. In practice, if a candidate molecule violates the novelty constraint (e.g., possessing a high structural similarity to a penalized scaffold or failing to meet the minimum baseline score of 0.4), its final composite reward is truncated to zero. This represents a "rigid" penalty-based approach to diversity, forcing the reinforcement learning policy to continuously escape local optima and discover topologically distinct chemotypes.

D.8 Metrics and Ablation Protocols

Evaluation Metrics.

- **Optimization Quality:** Average reward of valid unique molecules.
- **Internal Diversity (IntDiv):** Calculated as the average Tanimoto distance between all unique pairs of molecules in the set $\mathcal{S}$ (see Eq. 15 for the exact definition). Similarities are computed using **ECFP6 fingerprints** (Morgan radius=3, 2048 bits) via RDKit.

Ablation Settings. To isolate the contributions of DySCo, we defined the following comparison groups:

1. **Injection Protocols:**
   - `Dynamic-Shot` (Proposed): Injects prior samples at every epoch ($T = 1 \dots 100$).
   - `Single-Shot`: Injects prior samples only at initialization ($T = 0$).
   - `Static-All`: Mixes the entire static pre-training dataset into the buffer.
   - `Zero-Shot`: No prior injection.
2. **Balancing Strategies:** Comparing `Stochastic Replay (SR)`, `Diversity Induced Greedy Selection (DGS)`, and `Reward-Aware Transport Scoring (RTS)`.
3. **Hyperparameters:** Grid search over Balance Threshold $\beta \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ and Top-K Ratio $K \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.

E Supplementary Results on DrugEx

E.1 Comprehensive Baseline Comparison

Table 3 presents the exhaustive performance metrics for all evaluated methods. We explicitly detail the complete spectrum of baseline variants, including the "Greedy Top-K" progression, to characterize the concentrated structural coverage produced by reward-greedy selection and to benchmark the computational costs across all paradigms.

E.2 Pareto Frontier Analysis on Weight Preferences

To verify the robustness of DySCo across different multi-objective preferences, we evaluated all methods on 9 distinct weight combinations for the A2AR task. The preference vector takes the form $[w, 1 - w]$, where $w \in \{0.1, 0.2, \dots, 0.9\}$ controls the importance of the A2AR binding affinity relative to the SA score.

Figure 4 illustrates the performance trajectories of DySCo variants against baselines across this spectrum. Two key trends emerge:

1. Consistency with Chemical Priors. Across all methods, the average reward tends to increase as the weight for A2AR affinity (w) increases. This trend (visible as the upward trajectory in Figure 4a) suggests an intrinsic property of the chemical space defined by the pre-trained prior: the generative model finds it relatively easier to satisfy the binding affinity classifier than to optimize the SA constraint under strict penalties. Importantly, DySCo strategies follow this natural landscape gradient but consistently maintain a higher reward ceiling than baselines at every weight point, indicating superior optimization efficiency.

2. Pareto Dominance and Stability.

As shown in Figure 4, DySCo variants achieve a favorable Pareto front by maintaining strong performance in both reward optimization and structural diversity across a wide range of preference weights.

- **Optimization Quality:** DySCo strategies (solid lines) consistently achieve competitive or higher rewards compared to standard baselines (e.g., Vanilla RL, InfoBatch) and experience replay methods (dashed lines) across the full spectrum of w_{A2AR}. While the score-based K-Means (KM) baseline also reaches high reward levels, it does so at a notable cost to structural diversity.
- **Diversity Resilience:** As shown in Figure 4(b), although KM achieves high rewards, its internal diversity drops significantly below the Vanilla RL baseline, especially under high w_{A2AR}. Several other baselines also exhibit declining diversity as the task becomes more narrowly focused. In contrast, DySCo variants maintain a stable and robust diversity profile across all preference weights. This stability suggests that our continuous prior injection mechanism helps decouple diversity preservation from aggressive reward exploitation.

Table 3: Exhaustive performance benchmark. Comparison of DySCo against the full spectrum of Classic Coresets, Greedy strategies, Replay Buffers, and Diversity Filters. Data represents the average of 9 preference weights. Time is measured in seconds for 100 epochs. **Memory** is binarized: "Extra" means the method requires additional memory for large data structures (e.g., replay buffers, bilevel graphs), while "None" means it uses little to no extra memory.

Method Configuration	**Reward** ↑	**IntDiv** ↑	**Time (s)** ↓	**Memory**	**Mechanism Note**
1. Standard Baseline					
Vanilla RL (Linear)	0.7357	0.8953	1101.60	None	Foundational Reference
2. Classic Coreset Selection					
InfoBatch	0.7112	0.8971	253.19	None	Loss Pruning
Sorted Loss	0.7132	0.8964	195.91	None	Loss Pruning
CSReL	0.7085	0.8956	637.55	None	Reducible Loss Approx.
KMeans (Score-based)	0.8937	0.8320	1307.28	Extra	Feature Clustering
META	0.7476	0.8923	7397.99	Extra	Task-Selection
GCR	0.7312	0.8997	9860.57	Extra	Bilevel Match
3. Experience Replay & Diversity Filters					
Greedy Strategies:					
Greedy Top-K ($K = 0.50$)	<u>0.9956</u>	0.7484	764.38	None	Concentrated
Greedy Top-K ($K = 0.40$)	0.9791	0.7517	696.29	None	Concentrated
Greedy Top-K ($K = 0.30$)	0.9623	0.7593	673.08	None	Concentrated
Greedy Top-K ($K = 0.20$)	0.9401	0.7520	626.39	None	Concentrated
Greedy Top-K ($K = 0.10$)	0.9011	0.7565	617.16	None	Concentrated
Greedy Top-K ($K = 0.05$)	0.8781	0.7601	592.62	None	Concentrated
Experience Replay:					
ER(TH) [Top-History]	<u>0.9921</u>	0.7454	935.79	Extra	Feedback Loop
ER(TBH) [Top-Bot-Hist]	0.7562	0.9040	1141.60	Extra	Best Baseline (H)
ER(TBC) [Top-Bot-Curr]	0.7616	0.9039	847.55	None	Best Baseline (C)
ER(BH) [Bin-History]	0.7299	0.9046	930.24	Extra	Slow Convergence
ER(BC) [Bin-Current]	0.7501	0.9043	876.69	None	-
Diversity Filters:					
Filter + TH	0.9600	0.8007	1139.61	Extra	Filter Overwhelmed
Filter + TC	0.9898	0.7564	1056.16	Extra	Filter Failed
Filter + BH	0.7411	0.9022	1843.10	Extra	-
Filter + BC	0.7384	0.8997	1054.21	Extra	-
Filter + TBH	0.7475	0.9021	1859.36	Extra	Reward Penalty
Filter + TBC	0.7558	0.9013	1342.14	Extra	Reward Penalty
4. Ours (State-Adaptive Curation)					
DySCo (SR)	**0.8416**	0.9009	**612.12**	**None**	**Pareto Optimal**
DySCo (DGS-Hamming)	0.8409	0.8906	827.40	None	Sequence-based
DySCo (RTS-Hamming)	0.8338	0.8908	886.95	None	Distributional
DySCo (DGS-Tanimoto)	0.7930	0.9075	844.61	None	Geometric Repulsion
DySCo (RTS-Tanimoto)	0.7836	**0.9120**	885.39	**None**	**Metric Aligned**

Notably, the RTS-Tanimoto variant consistently achieves the highest diversity across the weight spectrum.

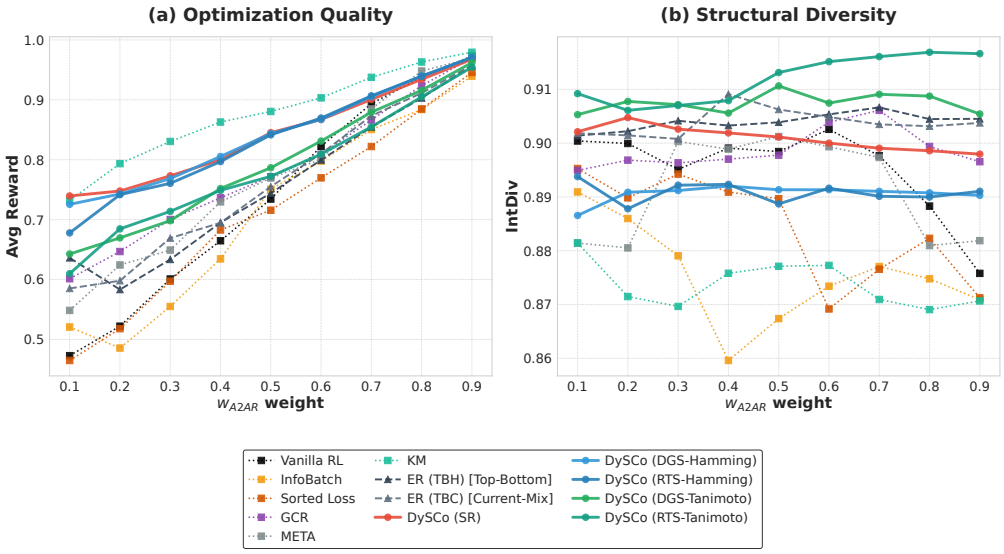


Figure 4: **Robustness across preference vectors.** Performance of DySCo strategies versus baselines on the A2AR multi-objective task with varying weights w_{A2AR}. **(a)** Average Reward: DySCo variants and KM achieve strong optimization scores. **(b)** Internal Diversity: KM suffers a steep decline in diversity (falling below Vanilla RL), while DySCo variants maintain robust diversity across the full range of weights, demonstrating improved trade-offs.

E.3 Detailed Analysis of Data Source

To rigorously validate the contribution of the "Dynamic-Shot Injection" mechanism, we performed an ablation study across the entire spectrum of Experience Replay strategies. Table 4 reports the performance of each backbone algorithm under four data sourcing protocols. Note that computation times are omitted here as they follow the general trends reported in the comprehensive benchmark (Table 3).

The dynamic advantage of our approach is further elucidated in Figure 5. By mapping the macro-trajectories of method categories, we observe that `Dynamic-Shot` injection acts as a powerful catalyst for DySCo, driving it toward the Pareto-optimal region. While other methods remain stagnant or show horizontal drift, DySCo exhibits a clear “up-and-right" vector, reinforcing the necessity of persistent prior intervention.

E.4 Detailed Hyperparameter Grid Search & Strategy Motivation

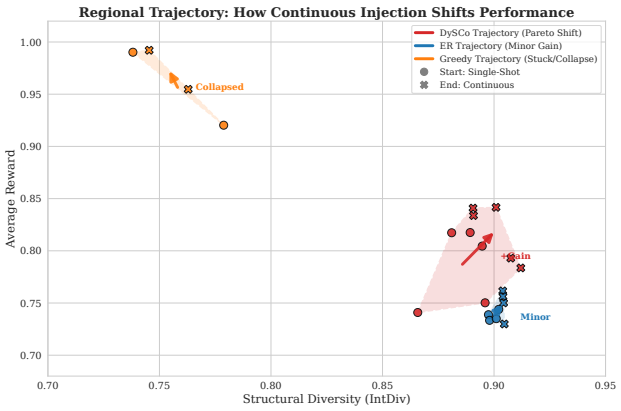
Here, we present the exhaustive results of the grid search (Table 5) and utilize these findings to empirically motivate the design of our advanced balancing strategies (Strategy II & III).

The Necessity of Balancing. First, we reference the “Greedy Dilemma" shown in Table 3. Without active intervention, purely greedy selection ($K \to 0.5$) produces pronounced structural concentration (IntDiv drops to ≈ 0.75), motivating a regularization mechanism. Appendix J.7 characterizes this benchmark-relative failure region without treating this IntDiv value as a universal threshold.

Behavioral Divergence: SR vs. Optimal Transport (OT). The hyperparameter grid search in Table 5 was conducted using **Strategy I (Stochastic Replay/SR)**. While this foundational strategy restores IntDiv to ≈ 0.90, we observe a distinct behavioral divergence when comparing its learning trajectory to the advanced mechanisms, particularly **Strategy III (Reward-Aware Transport Scoring (RTS))**.

Table 4: Impact of data source across all strategies. Comparison of `Single` (initialization only), `Full-Shot` (static full dataset), and `Dynamic-Shot` (dynamic per epoch) injection. **Key finding:** `Dynamic-Shot` injection consistently yields the highest reward improvement relative to `Single-Shot` for Balanced and DySCo strategies. For Greedy strategies, injection does not recover broad structural coverage.

Backbone & Protocol	Reward	IntDiv	△ vs Single	Mechanism Note
1. DySCo Framework (Ours)				
DySCo (SR) – Dynamic-Shot	**0.8416**	**0.9009**	**+4.6%**	**Optimal Trade-off**
DySCo (SR) – Single-Shot	0.8045	0.8947	Ref	Policy Drift
DySCo (SR) – Full-Shot	0.7701	0.9034	-4.3%	Signal Dilution
DySCo (SR) – Zero-Shot	0.8868	0.8370	-	(NoCore Balanced)
DySCo (DGS-Hamming) – Dynamic-Shot	0.8409	0.8906	+2.9%	Robust Sequence
DySCo (DGS-Hamming) – Single-Shot	0.8175	0.8893	Ref	
DySCo (DGS-Hamming) – Full-Shot	0.8226	0.8934	+0.6%	
DySCo (DGS-Hamming) – Zero-Shot	0.7962	0.8888	-	
DySCo (RTS-Hamming) – Dynamic-Shot	0.8338	0.8908	+4.4%	
DySCo (RTS-Hamming) – Single-Shot	0.8173	0.8810	Ref	
DySCo (RTS-Hamming) – Full-Shot	0.8475	0.8737	+3.7%	High Reward, Lower Div
DySCo (RTS-Hamming) – Zero-Shot	0.7983	0.8836	-	
DySCo (Tanimoto) – Dynamic-Shot	0.7930	0.9075	+7.8%	High Diversity
DySCo (Tanimoto) – Single-Shot	0.7502	0.8960	Ref	
DySCo (Tanimoto) – Full-Shot	0.7661	0.9037	+2.1%	
DySCo (Tanimoto) – Zero-Shot	0.7474	0.8429	-	
DySCo (RTS-Tanimoto) – Dynamic-Shot	0.7836	0.9120	+9.0%	Highest Diversity
DySCo (RTS-Tanimoto) – Single-Shot	0.7409	0.8658	Ref	
DySCo (RTS-Tanimoto) – Full-Shot	0.8569	0.8175	+15.6%	Div Drop (Overfitting)
DySCo (RTS-Tanimoto) – Zero-Shot	0.7026	0.7248	-	
2. Balanced Buffers				
ER(TBC) – **Dynamic-Shot**	0.7616	0.9039	+3.1%	Best Baseline Buffer
ER(TBC) – Single-Shot	0.7388	0.8975	Ref	Drift
ER(TBC) – Full-Shot	0.7333	0.8977	-0.7%	Dilution
ER(TBC) – Zero-Shot	0.7313	0.8994	-	(Vanilla Baseline)
ER(TBH) – **Dynamic-Shot**	0.7562	0.9040	+1.7%	History Stabilizes
ER(TBH) – Single-Shot	0.7439	0.9021	Ref	
ER(TBH) – Full-Shot	0.7438	0.9054	0.0%	
ER(TBH) – Zero-Shot	0.7505	0.8987	-	
3. Uniform Buffers				
ER(BC) – **Dynamic-Shot**	0.7501	0.9043	+2.3%	
ER(BC) – Single-Shot	0.7334	0.8980	Ref	
ER(BC) – Full-Shot	0.7228	0.8996	-1.4%	
ER(BH) – Dynamic-Shot	0.7299	0.9046	-0.7%	Buffer Dominates
ER(BH) – Single-Shot	0.7350	0.9010	Ref	
ER(BH) – Full-Shot	0.7416	0.9045	+0.9%	
4. Greedy Strategies				
ER(TC) – Dynamic-Shot	0.9547	0.7629	+3.7%	Concentrated
ER(TC) – Single-Shot	0.9203	0.7788	Ref	
ER(TC) – Full-Shot	0.9525	0.7905	+3.5%	
ER(TH) – Dynamic-Shot	0.9921	0.7454	+0.2%	Concentrated
ER(TH) – Single-Shot	0.9902	0.7381	Ref	
ER(TH) – Full-Shot	0.9905	0.7579	0.0%	
5. With Diversity Filter				
Filter + TBC – Dynamic-Shot	0.7558	0.9013	+1.9%	Filter Masks Gain
Filter + TBC – Single-Shot	0.7420	0.8952	Ref	
Filter + TBC – Full-Shot	0.7384	0.8972	-0.5%	



h

Figure 5: **Regional Trajectory Analysis of Data Source Protocols.** This plot illustrates the performance shift when transitioning from `Single-Shot` (circles) to `Dynamic-Shot` (crosses) data injection. The **DySCo Trajectory** (red arrow) shows a significant Pareto improvement, climbing in reward while sustaining optimal diversity. In contrast, the **Buffer Trajectory** (blue) shows marginal gains, and the **Greedy Trajectory** (orange) remains in the concentrated reward-greedy region despite reward increases. This confirms that continuous prior injection specifically synergizes with DySCo's state-adaptive selection.

The performance difference fundamentally arises from what each strategy emphasizes during the curation phase. The **SR** strategy imposes a looser structural constraint on the optimization trajectory. By injecting prior samples without explicitly enforcing topological alignment, it acts as a lightweight regularizer. This "unconstrained" nature allows the policy to pursue the multi-objective reward more aggressively, free from strict structural anchoring friction.

In contrast, the **OT** strategy is inherently more conservative. It performs *targeted correction* by selecting prior samples that are both topologically diverse and distributionally aligned with the current elite set. By rigorously penalizing chemical outliers and enforcing structural alignment, OT ensures that the chemical manifold is preserved more effectively.

The Optimization-Diversity Trade-off. The observation that the SR strategy performs better in terms of reward maximization on some multi-objective tasks is, therefore, consistent with a natural trade-off between aggressive functional optimization and rigorous structural preservation. This aligns perfectly with the empirical main results : SR tends to excel at pushing the reward ceiling due to its flexible gradient landscape, while OT-based variants are highly reliable when long-term structural diversity and topological coverage are the primary concerns. Rather than viewing one as strictly superior, they provide complementary tools depending on the specific constraint requirements of the generative task.

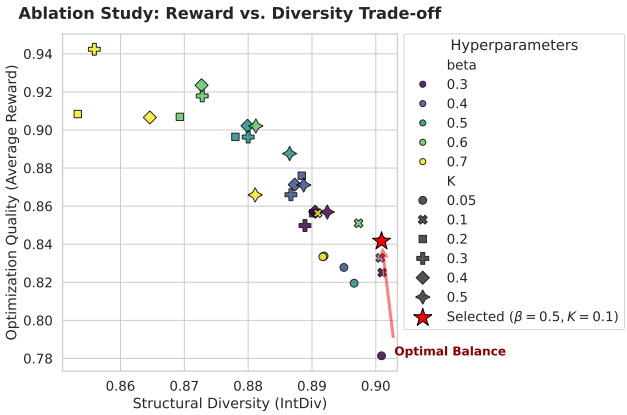


Figure 6: **Hyperparameter Sensitivity: Reward vs. Diversity Trade-off.** Each point represents a unique configuration (β, K). The **red star** highlights our selected configuration ($\beta = 0.5, K = 0.1$), which achieves an optimal balance compared to structurally concentrated regions (left) or low-reward regions (bottom).

Table 5: Exhaustive hyperparameter sensitivity analysis. Performance metrics across the full hyperparameter grid. The chosen configuration ($\beta = 0.5, K = 0.1$) is highlighted in bold.

Balance Threshold (β)	**Top-K Ratio (K)**	**Avg Reward ↑**	**IntDiv ↑**
0.3	0.05	0.7814	0.9009
0.3	0.10	0.8251	0.9010
0.3	0.20	0.8563	0.8902
0.3	0.30	0.8498	0.8889
0.3	0.40	0.8571	0.8905
0.3	0.50	0.8569	0.8924
0.4	0.05	0.8278	0.8950
0.4	0.10	0.8328	0.9007
0.4	0.20	0.8760	0.8884
0.4	0.30	0.8660	0.8867
0.4	0.40	0.8713	0.8873
0.4	0.50	0.8711	0.8887
0.5	0.05	0.8195	0.8966
0.5	**0.10**	**0.8416**	**0.9009**
0.5	0.20	0.8964	0.8780
0.5	0.30	0.8963	0.8800
0.5	0.40	0.9022	0.8799
0.5	0.50	0.8876	0.8865
0.6	0.05	0.8338	0.8919
0.6	0.10	0.8510	0.8973
0.6	0.20	0.9069	0.8693
0.6	0.30	0.9179	0.8728
0.6	0.40	0.9235	0.8727
0.6	0.50	0.9021	0.8812
0.7	0.05	0.8334	0.8917
0.7	0.10	0.8563	0.8909
0.7	0.20	0.9084	0.8533
0.7	0.30	0.9424	0.8559
0.7	0.40	0.9066	0.8646
0.7	0.50	0.8659	0.8811

F Robustness on 3-Objective Constraints (DrugEx v2)

To complete our evaluation on the DrugEx v2 (sequence-based) framework, we extended the benchmark to the **3-Objective Drug-Likeness Task** (Task 2 defined in Appendix D). Table 6 summarizes the results.

These results provide a compelling counterpoint to the Graph-based results (Appendix G) and strongly support our theoretical discussion on metric selection (Appendix K):

- **Exploitation Power (SR): DySCo (SR)** achieves the highest reward (**0.956**), significantly surpassing the baseline (0.812). This confirms that in simpler sequence-based tasks with smooth landscapes (Lipinski constraints), random buffering provides sufficient gradient variance for aggressive optimization.
- **The Diversity Champion (RTS-Tanimoto):** Most notably, **DySCo (RTS-Tanimoto)** achieves an exceptional Internal Diversity score of **0.951**, the highest recorded across all our experiments. While its reward (0.822) is lower than SR, it remains superior to the baseline. This empirically validates our argument in Appendix K: Tanimoto-based OT actively pushes the generative policy towards topologically distinct regions of the chemical space (maximizing semantic diversity), where as Hamming or SR strategies tend to cluster around high-scoring modes.

Table 6: Robustness on DrugEx v2 (3-objective task). Comparison of reward and diversity. While the SR strategy maximizes the reward in this sequence-based task, RTS-Tanimoto achieves near-perfect diversity (0.951), serving as strong empirical evidence for the topological superiority of Tanimoto similarity in coverage tasks.

Method Configuration	Reward ↑	IntDiv ↑
Baselines		
Vanilla RL	0.812	0.909
Filter + ER	0.816	0.907
DySCo Strategies (Ours)		
DySCo (SR)	**0.956**	0.891
DySCo (DGS-Hamming)	0.893	0.918
DySCo (RTS-Hamming)	0.860	0.916
DySCo (DGS-Tanimoto)	0.817	0.937
DySCo (RTS-Tanimoto)	0.822	**0.951**

G Robustness Analysis on Graph-Based Generation (DrugEx v3)

To verify the topological generalizability of DySCo, we extended our evaluation to **DrugEx v3** (Liu et al., 2023), which employs a **Graph Transformer** to construct molecules via iterative graph editing rather than sequence generation. Table 7 presents the performance on the 2-Objective (A2AR) and 3-Objective (Drug-Likeness) tasks.

Emergence of OT Dominance. Unlike sequence-based models where the `SR` strategy often sufficed, graph-based generation exhibits a strong preference for **Optimal Transport (OT)** driven strategies, revealing a distinct trade-off between metric spaces:

- **Optimization Quality: DySCo (RTS-Hamming)** consistently achieves the highest rewards across both tasks (**0.758** in 2-Obj and **0.775** in 3-Obj), significantly outperforming Vanilla RL and SR strategies. This suggests that aligning distributions based on discrete structural edits (Hamming) is most effective for guiding the graph editor toward high-scoring regions.
- **Structural Diversity: DySCo (RTS-Tanimoto)** excels in preserving diversity (reaching **0.939** in 3-Obj), as it explicitly maximizes transport cost in the fingerprint space.

Interpretation: The graph editing action space is significantly larger and more discontinuous than SMILES token generation. In this complex landscape, simple stochastic regularization (`SR`) fails to

provide sufficient guidance. In contrast, OT-driven curation—by explicitly solving for the optimal transport plan—provides a "chemically semantic" anchor, effectively mitigating mode collapse.

Table 7: Robustness benchmark on DrugEx v3 (graph-based). Comparison of reward and internal diversity (IntDiv). RTS-Hamming achieves the highest optimization quality across both tasks, while RTS-Tanimoto excels in structural diversity. This confirms that OT-driven distributional alignment is particularly effective in complex graph generation spaces.

Method Configuration	**Task 1: A2AR (2-Obj)**		**Task 2: Lipinski (3-Obj)**	
	Reward ↑	**IntDiv** ↑	**Reward** ↑	**IntDiv** ↑
Baselines				
Vanilla RL	0.654	0.901	0.709	0.913
Filter + ER	0.636	0.899	0.624	0.904
DySCo Strategies (Ours)				
DySCo (SR)	0.665	0.898	0.724	0.909
DySCo (DGS-Hamming)	0.689	0.900	0.640	0.937
DySCo (DGS-Tanimoto)	0.662	**0.901**	0.687	0.924
DySCo (RTS-Hamming)	**0.758**	0.892	**0.775**	0.928
DySCo (RTS-Tanimoto)	0.735	0.896	0.708	**0.939**

H Robustness Analysis on the REINVENT Framework

Table 8 provides the performance metrics across **four** distinct generation modes (**De Novo, LibInvent, LinkInvent, and Mol2Mol**), serving as a comprehensive robustness check for the DySCo framework.

The results demonstrate that DySCo's dynamic curation of compact, high-utility coresets effectively balances reward maximization with structural preservation. Notably, despite using a significantly reduced data footprint, **DySCo (SR)** achieves peak rewards in complex constrained tasks such as Lib/2 (**0.891**) and Link/2 (**0.914**). Simultaneously, Tanimoto-based strategies (e.g., **RTS-Tanimoto**) consistently yield the highest structural diversity (reaching ≈ 0.92 in RL/2). This empirical evidence confirms that our adaptive framework is highly robust, matching or exceeding the performance of full-batch baselines across diverse multi-objective landscapes while maintaining the essential exploration capacity required for pharmaceutical research.

Table 8: **Robustness Benchmark on REINVENT tasks (rounded to 3 decimal places).** Performance comparison across generation modes. DySCo strategies consistently maintain a superior Pareto frontier compared to Vanilla RL and Origin (Filter+ER) baselines, proving that adaptive coreset selection does not sacrifice optimization quality or diversity.

Method Configuration	**RL (2-Obj)**		**RL (3-Obj)**		**Lib (2-Obj)**		**Lib (3-Obj)**		**Link (2-Obj)**		**Link (3-Obj)**		**Mol (2-Obj)**		**Mol (3-Obj)**	
	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv	Reward	IntDiv
Baselines																
Vanilla RL	0.775	0.909	0.703	0.758	0.827	0.745	0.660	0.742	0.757	0.731	0.715	0.711	0.829	0.816	0.720	0.832
Origin (Filter+ER)	**0.779**	0.908	0.705	0.762	0.835	0.751	0.719	0.749	0.903	0.761	0.720	0.741	0.835	0.841	0.715	0.842
DySCo Strategies (Ours)																
DySCo (SR)	0.776	0.905	**0.710**	0.726	**0.891**	0.722	**0.734**	0.726	**0.914**	0.762	**0.748**	0.740	**0.881**	0.840	**0.748**	0.822
DySCo (DGS-Hamming)	0.768	0.908	0.683	0.740	0.849	0.734	0.703	0.740	0.893	0.716	0.693	0.762	0.827	0.847	0.714	0.840
DySCo (RTS-Hamming)	0.765	0.908	0.684	0.735	0.841	0.731	0.713	0.735	0.903	0.700	0.697	0.754	0.834	0.849	0.713	0.825
DySCo (Tanimoto)	0.771	0.918	0.699	0.781	0.866	0.772	0.709	0.778	0.824	**0.775**	0.698	0.772	0.834	0.861	0.713	**0.854**
DySCo (RTS-Tanimoto)	0.770	**0.923**	**0.707**	**0.785**	0.872	**0.775**	0.718	**0.781**	0.870	0.772	0.702	**0.776**	0.827	**0.863**	0.713	0.850

I Standardized Benchmark Evaluation (PMO)

To evaluate DySCo on the full set of standardized benchmarks, we follow the Practical Molecular Optimization (PMO) protocol (Gao et al., 2022) with a budget of 10,000 oracle calls per task, 3 seeds per condition. We compare DySCo (RTS-Tanimoto) against Vanilla RL across 8 PMO tasks spanning rediscovery, similarity, multi-property optimization, and isomer generation objectives. Table 9 reports the results.

DySCo achieves higher structural diversity (IntDiv of top-100 molecules) on 7 out of 8 tasks while maintaining competitive reward AUC, and consistently reduces wall-clock time by 20–30%. On

the Isomers task, DySCo also achieves higher AUC Top-10 and Top-10 reward. These results are consistent with the main experiments: DySCo's primary strength lies in producing a more structurally diverse set of high-quality candidates within a fixed oracle budget, which is particularly valuable for downstream hit-to-lead selection in drug discovery.

Table 9: PMO benchmark results (10,000 oracle calls, 3 seeds). Best values per task in **bold**. DySCo achieves higher IntDiv on 7/8 tasks and lower runtime on all 8 tasks. GSK3B and Valsartan_SMARTS are excluded (zero scores for both methods).

	Vanilla RL				**DySCo (RTS-Tanimoto)**			
Task	AUC_{10}	Top-10	IntDiv	Time(s)	AUC_{10}	Top-10	IntDiv	Time(s)
QED	**0.937**	**0.946**	0.887	50	0.936	0.945	**0.894**	**43**
DRD2	0.961	1.000	0.876	87	**0.963**	1.000	**0.878**	**78**
Albuterol Sim.	**0.519**	**0.597**	0.888	40	0.499	0.545	**0.899**	**31**
Aripiprazole Sim.	**0.747**	**0.818**	0.842	41	0.735	0.792	**0.854**	**31**
Celecoxib Redis.	**0.482**	**0.638**	0.842	39	0.469	0.563	**0.870**	**30**
Thiothixene Redis.	**0.358**	**0.391**	0.843	39	0.353	0.380	**0.849**	**30**
Isomers $C_7H_8N_2O_2$	0.600	0.746	**0.918**	58	**0.678**	**0.803**	0.916	**48**
Median 1	**0.189**	**0.227**	0.885	42	0.181	0.206	**0.895**	**33**

Table 10 extends the PMO evaluation to all data curation methods from Table 1, reporting the mean AUC Top-10 and IntDiv across the 8 PMO tasks.

Table 10: PMO benchmark: mean AUC Top-10 and IntDiv (top-100) across 8 tasks, 3 seeds per condition.

Method	**AUC_{10}↑**	**IntDiv↑**
Standard Baseline		
Vanilla RL	0.599	0.873
Classic Coreset Selection		
InfoBatch	0.596	0.877
Sorted Loss	0.604	0.873
Experience Replay & Filters		
Greedy Top-K	0.596	0.879
ER(TBC)	0.596	0.873
ER(TH)	0.605	0.876
ER(TBH)	0.602	0.872
Filter+TBC	0.600	0.872
DySCo (Ours)		
DySCo (SR)	0.604	0.874
DySCo (DGS-Hamming)	**0.608**	0.873
DySCo (RTS-Hamming)	0.605	0.873
DySCo (DGS-Tanimoto)	0.602	0.873
DySCo (RTS-Tanimoto)	0.602	**0.882**

J Additional Empirical Analyses

Unless stated otherwise, evaluation follows the final-policy protocol used in the main experiments: each trained policy generates 1,000 new molecules, and the reported metrics are computed according to their definitions in Section 2.

J.1 Diagnostic Evidence for γ_t

We recorded γ_t and active-batch IntDiv at every epoch in five runs of the balanced $(0.5/0.5)$ DrugEx v2 A2AR/SA setting. The epoch-wise association was negative in all five runs (Table 11). In a representative analysis, IntDiv was 0.7333 in an elite set dominated by active-policy molecules,

0.8854 after corrective curation, and 0.8980 for generations from the final policy. These observations are consistent with using γ_t as a lightweight signal of source imbalance rather than a standalone predictor of IntDiv.

Table 11: Epoch-wise Spearman correlation between γ_t and active-batch IntDiv in five runs.

Run	1	2	3	4	5	Mean
Spearman coefficient	-0.544	-0.465	-0.490	-0.587	-0.445	-0.506

Extended analysis across 16 independent runs and multiple diversity metrics. To further validate γ_t as a diagnostic signal, we extended the analysis to 16 independent runs spanning different training configurations and evaluated six diversity metrics beyond active-batch IntDiv. Table 12 reports the results. Higher γ_t is consistently associated with lower diversity across all metrics, with the strongest association observed for elite-set IntDiv ($\rho = -0.665$) and elite scaffold entropy ($\rho = -0.556$). The negative correlation also holds for scaffold-level indicators (scaffold entropy: $\rho = -0.350$; scaffold count: $\rho = -0.289$), confirming that γ_t captures structural homogenization beyond fingerprint-level similarity. Nearly all associations are statistically significant ($p < 0.05$).

Table 12: Spearman correlation between γ_t and six diversity metrics across 16 independent runs. "Sig." denotes the number of runs with $p < 0.05$.

Metric	**ρ (mean $\pm$ std)**	**Range**	**Sig.**
Elite IntDiv	-0.665 ± 0.150	$[-0.764, -0.127]$	16/16
Elite Scaffold Entropy	-0.556 ± 0.146	$[-0.677, -0.060]$	15/16
Active IntDiv	-0.473 ± 0.139	$[-0.587, +0.027]$	15/16
Train IntDiv	-0.391 ± 0.114	$[-0.506, -0.053]$	15/16
Scaffold Entropy	-0.350 ± 0.065	$[-0.399, -0.165]$	16/16
Scaffold Count	-0.289 ± 0.069	$[-0.357, -0.171]$	13/16

J.2 Long-Horizon Empirical Stability

We extended Vanilla RL and DySCo (RTS-Tanimoto) to 1,000 epochs in the balanced DrugEx v2 A2AR/SA setting. Figure 7 shows the active-batch trajectories, while Table 13 reports the final recorded metrics and linear slopes of the active-batch trajectories over the last 200 epochs. The late-stage slopes are close to zero, and the comparison remains consistent with the shorter runs. This constitutes evidence of empirical stability under the evaluated setting, not a general convergence theorem for a changing update distribution.

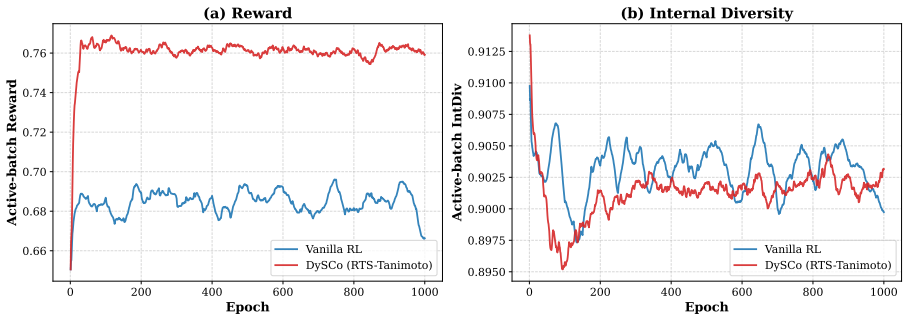


Figure 7: Active-batch reward and IntDiv over 1,000 epochs in the balanced A2AR/SA setting. Curves show 25-epoch moving averages.

J.3 Sample Efficiency

Following the molecular optimization protocol of Gao et al. (2022), we measure performance against the number of scored molecules. Across five runs of the balanced DrugEx v2 A2AR/SA setting,

Table 13: Long-horizon behavior over 1,000 epochs. Slopes are measured per epoch over the final 200 epochs.

Method	Final reward ↑	Final IntDiv ↑	Reward slope	IntDiv slope
Vanilla RL	0.7047	0.8960	-5.8×10^{-5}	-2.6×10^{-5}
DySCo (RTS-Tanimoto)	0.7282	0.9069	$+1.7 \times 10^{-5}$	-1.9×10^{-6}

the normalized Top-10 reward AUC through 10,000 molecular evaluations was 0.9865 ± 0.0009 for Vanilla RL and 0.9877 ± 0.0003 for DySCo (RTS-Tanimoto). Top-10 reward saturates early in this setting, so Fig. 8 and Table 14 also report cumulative Top-1,000 mean reward, which distinguishes performance over a broader set of candidates.

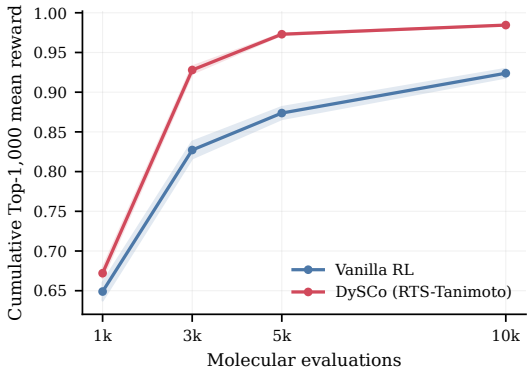


Figure 8: **Sample Efficiency.** Cumulative Top-1,000 mean reward versus molecular evaluations. Curves and bands show mean ± standard deviation over five runs.

Table 14: Cumulative Top-1,000 mean reward versus molecular evaluations (mean ± standard deviation over five runs).

Evaluations	Vanilla RL	DySCo (RTS-Tanimoto)
1,000	0.6490 ± 0.0113	0.6720 ± 0.0063
3,000	0.8272 ± 0.0105	0.9279 ± 0.0045
5,000	0.8737 ± 0.0075	0.9730 ± 0.0009
10,000	0.9239 ± 0.0054	0.9845 ± 0.0002

Together with the long-horizon results in Section J.2, these results provide additional context for the main experiments (Table 1), which use 100 training epochs following the standard DrugEx v2 protocol. At 100 epochs DySCo already achieves higher reward and diversity than Vanilla RL. As shown in Fig. 8, DySCo's reward advantage becomes more pronounced over longer horizons: by 10,000 evaluations the cumulative Top-1,000 gap widens to 0.9845 vs. 0.9239. Meanwhile, the diversity gap between the two methods narrows with additional training (IntDiv gap from 0.017 at 100 epochs to 0.011 at 1,000 epochs), indicating that Vanilla RL gradually recovers diversity given sufficient budget. This means the 100-epoch main results represent a conservative and strict evaluation setting for DySCo: the reward advantage only grows with more training, while the already-small diversity gap further diminishes.

Sample efficiency across all baselines. To contextualize the sample efficiency of DySCo relative to all baselines in Table 1, we report the cumulative Top-1,000 mean reward at oracle-call checkpoints for each method in Table 15. All methods are evaluated under the same balanced A2AR/SA setting with five seeds.

DySCo variants consistently achieve the highest cumulative Top-1,000 rewards across all checkpoints. At 3,000 evaluations, DySCo (RTS-Tanimoto) reaches 0.9451 ± 0.0024, compared with 0.8414 ± 0.0081 for Vanilla RL ($\Delta = +0.104$), 0.8513 ± 0.0053 for ER(TBC), and 0.9398 ± 0.0021 for

ER(TC). By 10,000 evaluations, the gap narrows but DySCo remains at the top (0.9881 vs. 0.9439 for Vanilla RL). These results confirm that DySCo's adaptive curation enables more efficient accumulation of a broad set of high-reward candidates.

Table 15: Sample efficiency on the A2AR/SA balanced setting (mean $\pm$ standard deviation over five seeds). Cumulative Top-1,000 mean reward at different oracle-call checkpoints.

Method	**@1,000**	**@3,000**	**@5,000**	**@10,000**
Standard Baseline				
Vanilla RL	0.6584 ± 0.0093	0.8414 ± 0.0081	0.8952 ± 0.0055	0.9439 ± 0.0054
Classic Coreset Selection				
InfoBatch	0.6658 ± 0.0040	0.8583 ± 0.0089	0.9024 ± 0.0112	0.9497 ± 0.0102
Sorted Loss	0.6661 ± 0.0014	0.8545 ± 0.0145	0.9025 ± 0.0237	0.9461 ± 0.0191
CSReL	0.6588 ± 0.0059	0.8510 ± 0.0083	0.9108 ± 0.0074	0.9666 ± 0.0048
KMeans (Score)	0.6662 ± 0.0078	0.8696 ± 0.0085	0.9470 ± 0.0098	0.9846 ± 0.0011
Experience Replay & Filters				
ER(TC)	0.6870 ± 0.0086	0.9398 ± 0.0021	0.9780 ± 0.0008	0.9873 ± 0.0007
ER(TH)	$\mathbf{0.6887 \pm 0.0061}$	0.9363 ± 0.0031	0.9752 ± 0.0005	0.9839 ± 0.0005
ER(TBC)	0.6556 ± 0.0077	0.8513 ± 0.0053	0.9116 ± 0.0048	0.9665 ± 0.0026
ER(TBH)	0.6586 ± 0.0062	0.8571 ± 0.0034	0.9163 ± 0.0043	0.9692 ± 0.0009
Filter + TBC	0.6564 ± 0.0079	0.8525 ± 0.0032	0.9158 ± 0.0058	0.9702 ± 0.0019
DySCo (Ours)				
DySCo (SR)	0.6839 ± 0.0070	0.9458 ± 0.0026	$\mathbf{0.9792 \pm 0.0011}$	0.9880 ± 0.0007
DySCo (DGS-Hamming)	0.6839 ± 0.0070	0.9469 ± 0.0018	0.9786 ± 0.0006	0.9873 ± 0.0004
DySCo (RTS-Hamming)	0.6839 ± 0.0070	$\mathbf{0.9469 \pm 0.0018}$	0.9786 ± 0.0007	0.9872 ± 0.0002
DySCo (DGS-Tanimoto)	0.6839 ± 0.0070	0.9448 ± 0.0025	0.9792 ± 0.0004	0.9880 ± 0.0003
DySCo (RTS-Tanimoto)	0.6839 ± 0.0070	0.9451 ± 0.0024	0.9792 ± 0.0003	$\mathbf{0.9881 \pm 0.0003}$

J.4 Robustness Across RL Configurations

DySCo curates update data without replacing the policy optimizer. We therefore evaluated it with two additional SMILES-based molecular RL optimizers. For PPO, we used the published molecular SMILES-RL implementation of Gummesson Svensson et al. (2024) on DRD2, with the same task, budget, and evaluation protocol for PPO with and without DySCo. The results are averaged over three seeds. For the GRPO check, we used the ACEGEN molecular-generation framework of Bou et al. (2024) on Albuterol similarity and implemented a critic-free group-relative policy update. The two GRPO conditions used the same training and evaluation settings, and results are reported over five seeds.

We also tested DySCo within a multi-agent configuration using ACEGEN on the Albuterol-similarity task. Four agents were trained independently, and DySCo (RTS-Tanimoto) curated the update data of each agent without changing the multi-agent algorithm. The two conditions used the same four-agent training and evaluation budgets: each agent processed 10,240 training samples, and the combined final policies were evaluated on 1,000 samples. This single completed supplementary run is a composability check, not a direct baseline comparison between single- and multi-agent RL.

Table 16: Checks across policy optimizers and multi-agent RL. Times are in seconds and are comparable within each paired setting. GRPO results are averaged over five seeds.

Setting	Method	Reward metric	Reward ↑	IntDiv ↑	Time ↓
PPO	Native	Top-100	0.9704	0.5746	96
PPO	+ DySCo	Top-100	**0.9826**	**0.7464**	**85**
GRPO	Native	Top-10	0.8023	0.5641	1684
GRPO	+ DySCo	Top-10	**0.8780**	**0.6983**	**521**
Multi-agent (4)	Native	Mean / Top-100	0.8831 / 0.9939	0.6771	197
Multi-agent (4)	+ per-agent DySCo	Mean / Top-100	0.8790 / **0.9988**	**0.7107**	**103**

Adding DySCo in the PPO and GRPO experiments gives higher reported reward and IntDiv together with lower runtime. In the four-agent configuration, mean reward remains comparable, while Top-100

reward increases and IntDiv rises from 0.6771 to 0.7107 (an absolute gain of 0.0336, or approximately 5.0%). These checks suggest that DySCo can remain effective with different policy optimizers and can be incorporated into each agent of a multi-agent method.

J.5 Sensitivity to the OT Regularization Parameter ε

To verify that the fixed $\varepsilon = 5$ choice does not critically affect performance, we varied ε over a wide range $\{0.5, 1, 2, 5, 10, 20\}$ on the balanced A2AR/SA setting with three seeds each. Table 17 reports the results. Both reward and IntDiv remain stable across all tested values, confirming that DySCo (RTS-Tanimoto) is robust to this parameter.

Table 17: Sensitivity to the OT regularization parameter ε on the balanced A2AR/SA setting (mean $\pm$ std over three seeds).

ε	Reward ↑	IntDiv ↑
0.5	0.8012 ± 0.0037	0.8741 ± 0.0048
1.0	0.7911 ± 0.0081	0.8815 ± 0.0027
2.0	0.7951 ± 0.0075	0.8792 ± 0.0037
5.0	0.8010 ± 0.0036	0.8757 ± 0.0008
10.0	0.7938 ± 0.0097	0.8790 ± 0.0035
20.0	0.7912 ± 0.0064	0.8785 ± 0.0010

J.6 Compatibility with Reward-Level Diversity Methods

RND and TanhRND modify the reward to encourage diversity, whereas DySCo curates the molecules used for policy updates. Using the published DRD2 configuration of Svensson et al. (2025), we evaluated DySCo alone and in combination with both reward-level methods under the same setting. In each combined condition, reward modification precedes DySCo curation; the purpose is to evaluate direct compatibility rather than tune a specialized coupling mechanism.

Table 18: Combination with reward-level diversity methods under the same DRD2 setting.

Method	Mean reward ↑	IntDiv ↑
Vanilla RL	0.961	0.643
DySCo (RTS-Tanimoto)	**0.984**	**0.802**
RND	0.743	**0.777**
RND + DySCo (RTS-Tanimoto)	**0.804**	0.769
TanhRND	0.741	**0.783**
TanhRND + DySCo (RTS-Tanimoto)	**0.808**	0.746

Under this protocol, DySCo alone gives the highest mean reward and IntDiv among the six conditions. Combining DySCo with RND or TanhRND increases mean reward and slightly decreases IntDiv relative to each reward-level method alone. Thus, DySCo can be combined with reward-level diversity methods, with the direct combination changing their reward–diversity trade-off.

J.7 Structural Characterization of Reward-Greedy Selection

There is no generally accepted absolute IntDiv threshold for mode collapse. We therefore use *high-reward, low-diversity region* to describe the benchmark-relative behavior in Fig. 2. Relative to Vanilla RL, the reward-greedy condition reduces IntDiv from 0.895 to 0.745, corresponding to an increase in mean pairwise similarity from 0.105 to 0.255 (approximately $2.4\times$). Table 19 provides additional structural statistics. Together, these statistics show narrower and more redundant structural coverage in this condition without defining a universal collapse threshold.

Table 19: Structural characterization of the reward-greedy condition relative to Vanilla RL.

Metric	Vanilla RL	Reward-greedy condition
IntDiv	0.895	0.745
Mean pairwise similarity	0.105	0.255
Scaffold count	882	468
Top-10 scaffold share	8.4%	28.7%
Median nearest-neighbor Tanimoto	0.278	0.720

K Theoretical Discussion: The Primacy of Topological Similarity over Syntactic Distance

A consistent observation across our benchmarks (Tables 3, 8, and 7) is that strategies leveraging **Tanimoto similarity** (specifically `DySCo-Tanimoto` and `DySCo-RTS-Tanimoto`) consistently yield the highest Internal Diversity scores. While Hamming-based methods offer optimization advantages by aligning with the token-based generation process (as noted in our experimental analysis), we argue that regarding *diversity preservation*, Tanimoto similarity is theoretically superior and chemically more meaningful. This section explores the cheminformatic rationale behind this preference.

K.1 Syntax vs. Semantics in Chemical Space

The fundamental distinction lies in the nature of the metric space:

- **Hamming Distance represents Syntactic Diversity.** Since SMILES is a 1D linearization of a molecular graph, Hamming distance measures the divergence in the *string representation*. This metric is brittle; two identical molecules can have large Hamming distances simply due to different atom traversal orders (canonicalization artifacts), while two chemically distinct molecules (e.g., changing a Carbon 'C' to a Nitrogen 'N') might have a negligible Hamming distance. Maximizing Hamming diversity risks generating "typographical variants" rather than true structural novelty.
- **Tanimoto Similarity represents Semantic Diversity.** Tanimoto similarity, calculated over Extended-Connectivity Fingerprints (ECFP) (Rogers & Hahn, 2010), operates on the **Molecular Graph**. It captures invariant topological features—circular substructures, bond environments, and pharmacophores—regardless of how the string is written. High diversity in Tanimoto space guarantees that the selected molecules possess distinct scaffolds and functional groups, which is the ultimate goal of drug discovery.

K.2 The "Similar Property Principle" and Scaffold Hopping

The preference for Tanimoto is deeply rooted in the **Similar Property Principle** (SPP) (Hendrickson, 1991; Martin et al., 2002), which postulates that structurally similar molecules are likely to exhibit similar biological properties.

- **Bioactive Neighborhoods:** In lead optimization, we aim to explore the "bioactive neighborhood" of a hit. Tanimoto similarity provides a continuous and smooth metric for this manifold. A Tanimoto distance of 0.4 implies a significant but related structural change (e.g., *Scaffold Hopping* from a phenyl to a pyridyl ring), whereas a Hamming distance of 4 tokens has no consistent chemical interpretation.
- **Avoiding the "Cliff" Illusion:** Sequence-based metrics often fail to detect *Activity Cliffs*—minor structural changes causing major potency shifts. By anchoring diversity curation to Tanimoto (as done in `RTS-Tanimoto`), DySCo explicitly navigates the chemical feature space, ensuring that the injected diversity contributes to exploring new areas of the Structure-Activity Relationship (SAR) landscape, rather than merely exploring new string permutations.

K.3 Conclusion on Metric Selection

Consequently, while Hamming distance acts as a useful proxy for *policy regularization*, our benchmarks reveal its **task-dependent instability**—while it excels in optimizing graph-based agents (Appendix G), it often fails to guarantee consistent structural diversity in sequence-based tasks. In sharp contrast, Tanimoto similarity remains the **robust** gold standard for *chemical space coverage*. The consistent superiority of Tanimoto-based variants in diversity metrics across all architectures is therefore not an artifact, but a validation that these strategies successfully and stably sample from the true topological diversity of the drug-like manifold.

L Future work discussion: The Self-Evolving Paradigm in Generative Intelligence

While our primary investigation focuses on molecular design, the methodologies discussed in this work resonate with a broader trend in Artificial Intelligence: the shift from static, supervised learning to dynamic, **Self-Evolving Systems**. In this appendix, we contextualize our approach within this emerging paradigm, unifying perspectives from Large Language Models (LLMs) (Wang et al., 2023; Huang et al., 2023; Tao et al., 2024), Evolutionary Computation (EC) (Foster, 2001; Yan & Jin, 2024), Adversarial Learning (Goodfellow et al., 2014; Arjovsky et al., 2017), and Neural Architecture Search (NAS) (Zoph & Le, 2016; Liu et al., 2019). We posit that the *Sample-Judge-Update* cycle constitutes a universal algorithmic skeleton for autonomous intelligence. However, this closed loop inherently carries the risk of **Iterative Entropy Decay**—a generalized pathology that manifests variously as *Mode Collapse* in GANs (Che et al., 2016; Metz et al., 2016), *Premature Convergence* in EC (Squillero & Tonda, 2016), or the *Matthew Effect* and *Model Collapse* in LLMs (Guo et al., 2025; Shumailov et al., 2023; Alemohammad et al., 2023).

L.1 A Unified Mathematical Framework

We abstract the self-evolving system as an iterative optimization process acting on a generative policy π_θ within a high-dimensional search space $\mathcal{X}$. The objective is to maximize a utility function $\mathcal{J}(\theta)$ (e.g., reward, likelihood, or fitness) in the absence of comprehensive ground-truth supervision. Formally, a single evolution step $t \to t+1$ can be decomposed into a composite function of three canonical operators:

$$\theta_{t+1} \leftarrow \mathcal{U}\Big(\Phi\big(\mathcal{S}(\pi_{\theta_t})\big)\Big) \tag{24}$$

where:

1. **Sample ($\mathcal{S}$):** The generator produces a candidate set $\mathcal{X}_t \sim \pi_{\theta_t}(\cdot)$. This step represents the exploration of the search space manifold.
2. **Judge (Φ):** A curation operator evaluates candidates against an Oracle $\mathcal{O}$ (or proxy) and filters or re-weights them to produce a training signal $\mathcal{D}_t = \Phi(\mathcal{X}_t, \mathcal{O})$. This is the critical control gate that determines the direction of evolution.
3. **Update ($\mathcal{U}$):** The policy parameters are updated to maximize the likelihood of the curated distribution $\mathcal{D}_t$, effectively "internalizing" the Judge's preferences.

Below, we demonstrate how this unified formulation captures the essence of distinct AI domains.

L.2 Domain-Specific Instantiations

While the *Sample-Judge-Update* loop serves as a unifying abstraction, its practical realization varies significantly across fields. These variations reflect the distinct constraints—such as discrete versus continuous search spaces or differentiable versus black-box oracles—inherent to each domain.

Large Language Models (Reasoning via Self-Improvement). Large Language Models (LLMs) represent a paradigm of foundation models trained on massive corpora to capture linguistic and

semantic patterns (Brown et al., 2020; Touvron et al., 2023). The primary research objective in this domain is to align these models with complex user instructions and enhance their multi-step reasoning capabilities (e.g., mathematical problem solving), a challenge where high-quality human supervision is increasingly scarce and expensive (Ouyang et al., 2022; Wei et al., 2022).

- **Sample ($\mathcal{S}$):** The model functions as a probabilistic explorer. To encourage diverse reasoning paths, sampling typically combines *Chain-of-Thought (CoT)* prompting with stochastic decoding strategies, such as high-temperature sampling or nucleus sampling (Huang et al., 2023; Wang et al., 2023). This step explicitly trades off local probability for semantic diversity to uncover novel solution trajectories.
- **Judge (Φ):** Lacking ground-truth labels for generated rationales, the curation operator often relies on *Self-Consistency* (majority voting on the final answer) or *LLM-as-a-Judge* (using a teacher model or the agent itself to score process validity) (Wang et al., 2023; Zheng et al., 2023). The filter is typically discrete, retaining only high-confidence trajectories while discarding potential hallucinations.
- **Update ($\mathcal{U}$):** The policy is optimized to internalize these "golden" trajectories. While standard Supervised Fine-Tuning (SFT) minimizes the negative log-likelihood of the selected data, recent approaches employ *Direct Preference Optimization (DPO)* to explicitly increase the margin between chosen and rejected samples, thereby aligning the model with the Judge's preferences (Rafailov et al., 2023).

Evolutionary Computation (Model-Based Optimization). Evolutionary Computation (EC) encompasses a family of population-based metaheuristics designed to solve black-box optimization problems where gradient information is unavailable or unreliable (Foster, 2001; Back & Schwefel, 1996). The central goal is to navigate complex, often multi-objective fitness landscapes to identify a set of optimal trade-off solutions (the Pareto Frontier) efficiently (Deb, 2011; Coello et al., 2007).

- **Sample ($\mathcal{S}$):** Instead of random mutation, modern Generative Evolutionary Algorithms (GEAs) utilize learned probabilistic models (e.g., VAEs or Diffusion Models) to propose offspring (Larrañaga & Lozano, 2001; Yan & Jin, 2024). Sampling occurs in the latent space of the generative model, often incorporating Gaussian noise to explore the vicinity of the current elite population.
- **Judge (Φ):** The Oracle is a fitness function $F(x)$. To mitigate high computational costs, Φ frequently employs *Surrogate Models* (e.g., Gaussian Processes or Neural Networks) for efficient pre-screening (Jin, 2011). Crucially, in multi-objective settings, the selection logic uses non-dominated sorting to maintain a Pareto Frontier rather than a single optimum (Deb, 2011).
- **Update ($\mathcal{U}$):** The update differs from gradient descent on a loss function. The generative model is retrained to minimize the distributional divergence (e.g., KL divergence) between its generated distribution and the superior *posterior* distribution of the selected elites, shifting the search focus toward high-fitness regions.

Adversarial Learning (Generative Adversarial Networks). Generative Adversarial Networks (GANs) constitute a framework for implicit density estimation, where a generative model learns to map latent noise to a target data distribution (e.g., realistic images) (Goodfellow et al., 2014). The research objective is to synthesize high-fidelity samples that are indistinguishable from real data by establishing a dynamic, zero-sum game between generation and discrimination (Arjovsky et al., 2017).

- **Sample ($\mathcal{S}$):** The Generator G acts as a deterministic mapper from a latent noise distribution p_z to the data space $\mathcal{X}$. Exploration is driven by the stochasticity of the input noise vector z and the evolving topology of the mapping function.
- **Judge (Φ):** The Discriminator D serves as a learnable, differentiable Oracle. Unlike the discrete filters in LLMs or EC, the Judge here provides a continuous, dense gradient signal $\nabla_x D(x)$ for every sample, indicating its likelihood of being "real" (Goodfellow et al., 2014).
- **Update ($\mathcal{U}$):** The system engages in a minimax game. The Generator's update does not maximize likelihood on a static set but performs gradient ascent to *maximize the Judge's*

error (fooling the discriminator). This dynamic creates a "moving target" for optimization, distinguishing it from other self-evolving loops.

Neural Architecture Search (NAS). Neural Architecture Search (NAS) is a subfield of Automated Machine Learning (AutoML) aimed at automating the design of neural network topologies (Elsken et al., 2019). The objective is to discover optimal architectures that maximize predictive performance and hardware efficiency (e.g., latency) within a discrete, combinatorial search space of computational graphs (Zoph & Le, 2016; Liu et al., 2019).

- **Sample ($\mathcal{S}$):** A Controller (typically an RNN or Transformer) autoregressively samples architecture descriptions (strings or computation graphs). Exploration is often enforced via entropy regularization in the policy to prevent collapsing into simple, local-optimum architectures (Real et al., 2019).
- **Judge (Φ):** Fully training every sampled architecture is prohibitive. Therefore, Φ employs *Zero-Cost Proxies* (e.g., measuring synaptic flow at initialization) or weight-sharing Supernets to estimate rank correlation (Mellor et al., 2021). The "Judge" here acts as an efficient but noisy approximation of the true validation accuracy.
- **Update ($\mathcal{U}$):** The Controller is updated via Reinforcement Learning (e.g., REINFORCE) to assign higher probability to the structural motifs found in the top-k candidates, effectively pruning the search space of invalid or low-performance topologies.

L.3 The Universal Pathology: Iterative Entropy Decay

Across all the aforementioned domains, the stability of the self-evolving loop hinges on the curation operator Φ. When Φ becomes overly "greedy"—systematically prioritizing the mode of the current distribution over its tails—it triggers a pervasive failure mode. We formalize this phenomenon as **Iterative Entropy Decay**, a dynamic process where the generative plasticity of the system degrades irreversibly over time.

General Theoretical Mechanism. Let $\pi_t(x)$ denote the probability density of the generative policy at step t, and $H(\pi_t) = -\mathbb{E}[\log \pi_t(x)]$ denote its differential entropy. In a closed self-consuming loop without external information injection, the update rule $\mathcal{U}$ typically acts as a contraction mapping on the distribution's support. Theoretical analysis suggests that if the curation operator Φ biases selection towards high-probability regions (exploitation), the sequence of distributions converges to a Dirac delta function centered at a local mode (Shumailov et al., 2023):

$$\lim_{t \to \infty} H(\pi_t) \to 0 \tag{25}$$

This occurs because the model is trained on a "head-heavy" subset of its own outputs ($\mathcal{D}_t \subset \mathcal{X}_t$). The "tails"—which often contain rare but innovative solutions—are statistically pruned, leading to a "Rich-Get-Richer" feedback loop that obliterates diversity.

Domain-Specific Manifestations. This mathematical inevitability manifests as distinct but isomorphic pathologies across fields:

- **In Large Language Models: The Matthew Effect.** *Manifestation:* The model's performance on "easy" tasks (high-frequency patterns) improves rapidly, while its ability to handle "hard" tasks (low-probability reasoning paths) deteriorates catastrophically. *Mechanism:* Self-generated fine-tuning data is dominated by the model's most confident outputs. Since the model is more confident in simple, common patterns, the update step disproportionately reinforces these "head" samples, causing the "tail" capabilities (e.g., complex logic or creativity) to vanish from the distribution (Guo et al., 2025; Alemohammad et al., 2023).
- **In Evolutionary Computation: Premature Convergence.** *Manifestation:* The population rapidly clusters around a local optimum, losing the genetic diversity required to cross fitness valleys. *Mechanism:* The "Judge" (survival selection) aggressively culls lower-fitness individuals. In Generative EAs, if the generative model overfits to the current elite population, it ceases to sample exploratory offspring in the broader search space, effectively halting evolutionary progress (Squillero & Tonda, 2016; Yan & Jin, 2024).

- **In Adversarial Learning: Mode Collapse.** *Manifestation:* The generator outputs a limited set of identical or near-identical samples (e.g., the same face with slight noise) regardless of the input latent vector. *Mechanism:* The Discriminator (Judge) seeks the single most distinguishable feature of "realness." The Generator (Update) exploits this by mapping the entire latent space to the single point that maximizes the Discriminator's error, collapsing the multi-modal data distribution into a single point mass (Che et al., 2016; Metz et al., 2016).
- **In Neural Architecture Search: Topology Convergence.** *Manifestation:* The controller converges prematurely to a single architectural motif (e.g., a network consisting entirely of identity mappings or "skip connections"), failing to discover complex, high-performing topologies. *Mechanism:* The policy update (Update) is driven by the rewards of the top-k sampled architectures. If a specific operation (e.g., a 3×3 convolution) yields a high reward early in the search due to randomness, the controller assigns it a probability near 1.0. This saturation prevents the sampling of alternative operations, effectively freezing the exploration of the architectural space (Zoph & Le, 2016; Real et al., 2019).
- **In Molecular Design: Structural Autophagy (Our Context).** *Manifestation:* The agent generates repetitive scaffolds (e.g., endlessly adding methyl groups to a known active core) that satisfy reward functions but lack structural novelty. *Mechanism:* High-reward molecules are selected for training. As the model learns to produce these specific motifs with high probability, the diversity of the sampled batch $\mathcal{X}_t$ drops. The Tanimoto similarity approaches 1.0, and the agent loses the "chemical intuition" required to hop to new, potentially superior scaffolds.

DySCo as an Entropic Regularizer.

Rather than directly modifying the policy update rule, DySCo acts as a dynamic curation operator on the **Judge** (Φ). By adaptively constructing a corrective coreset $\mathcal{B}$ from the prior policy distribution and injecting it into the training batch when structural homogenization is detected, DySCo prevents the iterative collapse of distributional entropy.

In essence, the initial prior policy π_0 serves as a distributional anchor. Through state-aware selection of N_{corr} corrective samples, DySCo enforces a soft lower bound on the entropy of the training distribution, thereby sustaining the exploratory capacity of the self-evolving loop over long horizons. This curation-based regularization complements conventional stabilization techniques (such as prior-likelihood penalties) while operating at the data level rather than the loss level.

M BROADER IMPACTS

Our work on DySCo provides a more efficient and diverse reinforcement learning framework for molecular design.

Positive Impacts: The primary positive impact is the potential to significantly accelerate the discovery of novel therapeutics and functional materials. By reducing computational overhead and maintaining structural diversity, DySCo helps researchers explore a wider chemical space at a lower cost, which could eventually lead to faster response times for emerging health crises and more affordable drug development.

Negative Impacts (Dual-use Risk): Like most AI-driven molecular generation tools, our framework faces the "dual-use" challenge. While designed for drug discovery, the same technology could theoretically be used to optimize harmful molecules, such as chemical toxins or environmental pollutants. To mitigate this risk, we emphasize that such tools should be used in conjunction with strict laboratory oversight and regulatory screening of generated structures.