
Appendix:

Through the looking glass: navigating in latent space to optimize over combinatorial synthesis libraries

Aryan Pedawi
Atomwise
aryan@atomwise.com

Saulo de Oliveira
Atomwise
saulo@atomwise.com

Henry van den Bedem
Atomwise & University of California, San Francisco
Dept. of Bioengineering & Therapeutic Sciences
vdbedem@atomwise.com

A Methodology

A.1 Policy parameterization

We seek a parameterization for the policy π_η that:

- Permits efficient sampling from $z \sim \pi_\eta(z|\tau)$;
- Permits efficient evaluation of log densities $\log \pi_\eta(z|\tau)$;
- Permits efficient evaluation of log density gradients $\nabla_\eta \log \pi_\eta(z|\tau)$.

A natural choice is to use normalizing flows [15, 17], a class of generative models in which noise is sampled from some base distribution $p(\epsilon)$ that admits straightforward sampling and log density evaluations (e.g., an isotropic standard Gaussian) and transformed through a series of neural network layers that are specially constructed so as to be invertible functions with efficient evaluation of the log absolute determinant of the Jacobian. Because the entire network is invertible, log densities for the output distribution can be computed via application of the change of variables formula. Let $g_\eta : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}^d$ be a conditional normalizing flow with parameters η , such that g_η satisfies $g_\eta(g_\eta^{-1}(z, \tau), \tau) = z$ for all τ by construction. Then the conditional log density for z under the policy can be evaluated via

$$\log \pi_\eta(z|\tau) = \log p(g_\eta^{-1}(z, \tau)) + \log \left| \det \frac{\partial g_\eta^{-1}(z, \tau)}{\partial z} \right|. \quad (1)$$

In our implementation, g_η is a neural spline flow [4] comprised of a stack of rational quadratic spline affine coupling layers interleaved with shuffling, with an affine transformation applied as the final operation.

A.2 Objective function

The Rényi α -divergence family generalizes the KL divergence and has been applied in variational inference via the variational Rényi bound (VRB) [11], which has the form

$$\mathcal{L}_\alpha(\pi_\eta, r_\tau) = \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right]. \quad (2)$$

In variational inference, the VRB is a lower bound on the marginal log likelihood for $\alpha < 1$, and it is related to the negative of an upper bound on the marginal log likelihood for $\alpha > 1$, which motivates maximization (cf. minimization) of the VRB for $\alpha < 1$ (cf. $\alpha > 1$). The VRB recovers the negative of the reverse KL divergence in the limit $\alpha \rightarrow 0$ and extrema of the log likelihood ratios in the left and right limits,

$$\lim_{\alpha \rightarrow 0} \mathcal{L}_\alpha(\pi_\eta, r_\tau) = \mathbb{E}_{\pi_\eta(z|\tau)} \left[\log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right] = -D_{\text{KL}}(\pi_\eta \| r_\tau); \quad (3)$$

$$\lim_{\alpha \rightarrow -\infty} \mathcal{L}_\alpha(\pi_\eta, r_\tau) = \min_{\pi_\eta(z|\tau)} \log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} = \max_{\pi_\eta(z|\tau)} \log \frac{\pi_\eta(z|\tau)}{r_\tau(z)}; \quad (4)$$

$$\lim_{\alpha \rightarrow +\infty} \mathcal{L}_\alpha(\pi_\eta, r_\tau) = \max_{\pi_\eta(z|\tau)} \log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} = \min_{\pi_\eta(z|\tau)} \log \frac{\pi_\eta(z|\tau)}{r_\tau(z)}. \quad (5)$$

It follows that optimizing the VRB for $\alpha \rightarrow -\infty$ requires that $r_\tau(z) = 0 \implies \pi_\eta(z|\tau) = 0$ (otherwise, the VRB attains a value of $-\infty$), which is exactly the mode-seeking behavior described earlier. Similarly, optimizing the VRB for $\alpha \rightarrow +\infty$ requires that $\pi_\eta(z|\tau) = 0 \implies r_\tau(z) = 0$, or equivalently $r_\tau(z) > 0 \implies \pi_\eta(z|\tau) > 0$ by contraposition, which is the mass-covering behavior.

It can be shown that the unknown normalizing constant contributes a level shift to the VRB:

$$\mathcal{L}_\alpha(\pi_\eta, r_\tau) = \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right] \quad (6)$$

$$= \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\exp \left(\alpha \log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right) \right] \quad (7)$$

$$= \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\exp \left(\alpha \log \frac{\Psi_\tau^{-1} R_\tau(z)}{\pi_\eta(z|\tau)} \right) \right] \quad (8)$$

$$= \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\exp \left(\alpha \log \frac{R_\tau(z)}{\pi_\eta(z|\tau)} - \alpha \log \Psi_\tau \right) \right] \quad (9)$$

$$= \frac{1}{\alpha} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\exp \left(\alpha \log \frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right) \right] - \log \Psi_\tau \quad (10)$$

$$= \mathcal{L}_\alpha(\pi_\eta, R_\tau) - \log \Psi_\tau. \quad (11)$$

Since Ψ_τ does not depend on η , we have $\nabla_\eta \mathcal{L}_\alpha(\pi_\eta, r_\tau) = \nabla_\eta \mathcal{L}_\alpha(\pi_\eta, R_\tau)$, and thus optimization of the VRB with respect to the policy parameters η does not require knowledge of the normalizing constant. However, for mass-covering $\alpha > 1$, optimization can be challenged by pronounced variance in the estimated gradient, which is unfortunate since this is the regime that is generally of interest.

The gradient of the VRB with respect to α is invariant to the unknown normalizing constant and has the following form:

$$\frac{\partial}{\partial \alpha} \mathcal{L}_\alpha(\pi_\eta, r_\tau) = \frac{\partial}{\partial \alpha} \mathcal{L}_\alpha(\pi_\eta, R_\tau) = \frac{\mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \log \frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right]}{\alpha \mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right]} - \frac{1}{\alpha} \mathcal{L}_\alpha(R_\tau, \pi_\eta). \quad (12)$$

It can be shown that (1) $\frac{\partial}{\partial \alpha} \mathcal{L}_\alpha(r_\tau, \pi_\eta) \geq 0$ for all $\alpha \in \mathbb{R}$, and (2) $\frac{\partial}{\partial \alpha} \mathcal{L}_\alpha(r_\tau, \pi_\eta) = 0$ if and only if $r_\tau = \pi_\eta$. Hence, the gradient of the VRB with respect to α , in fact, defines a valid (new) α -divergence family,

$$D_\alpha(\pi_\eta \| r_\tau) = \frac{\mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \log \frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right]}{\alpha \mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right]} - \frac{1}{\alpha^2} \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right]. \quad (13)$$

These α -divergences can be interpreted as the difference between a weighted arithmetic mean of the log likelihood ratios and the log of a weighted geometric mean of the likelihood ratios, all scaled by

α , and where the weights are proportional to the likelihood ratios exponentiated by α . Although we have not attempted to demonstrate this rigorously, we postulate that such a formulation acts like a control variate, thereby helping to reduce gradient variance. Our intuition is that the first term in (13) is effectively a self-normalized (exponentiated) importance weighted expression, which is a common trick for reducing variance in importance sampling at the expense of introducing bias, whereas the second term may act like a correction, perhaps mitigating bias similar to a control variate. Further, these α -divergences have the useful property of being invariant to the unknown normalizing constant of the target distribution.

Notably, it can be shown that $D_\alpha(\pi_\eta \| r_\tau)$ recovers the *forward* KL divergence for $\alpha = 1$,

$$D_1(\pi_\eta \| r_\tau) = \frac{\mathbb{E}_{\pi_\eta(z|\tau)} \left[\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right]}{\mathbb{E}_{\pi_\eta(z|\tau)} \left[\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right]} - \log \mathbb{E}_{\pi_\eta(z|\tau)} \left[\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right] \quad (14)$$

$$= \mathbb{E}_{\pi_\eta(z|\tau)} \left[\frac{r_\tau(z)}{\pi_\eta(z|\tau)} \log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right] \quad (15)$$

$$= \mathbb{E}_{r_\tau(z)} \left[\log \frac{r_\tau(z)}{\pi_\eta(z|\tau)} \right] \quad (16)$$

$$= D_{\text{KL}}(r_\tau \| \pi_\eta). \quad (17)$$

Interestingly, optimization of the VRB for $\alpha = 1$, by contrast, does not correspond to divergence minimization at all, since $\mathcal{L}_1(\pi_\eta, r_\tau) = \log \mathbb{E}_{\pi_\eta(z|\tau)} [r_\tau(z)/\pi_\eta(z|\tau)] = 0$ is constant.

For $\alpha \geq 1$, the derived α -divergence family has the mass-covering inductive bias. In our experiments, we therefore minimize the objective

$$\min_{\eta} \mathbb{E}_{p(\tau)} [D(\pi_\eta \| r_\tau)] \quad (18)$$

with the divergence defined in (13) for $\alpha \geq 1$, using Monte Carlo (MC) approximation and stochastic gradient optimization.

A.3 Policy gradient

We can use MC approximations for all expectations when minimizing the objective (18). For generality, we introduce a behavior policy $\beta(z|\tau)$ and re-write expectations using importance sampling, arriving at the following expression for the gradient of the temperature-marginalized α -divergence with respect to the policy parameters:

$$\begin{aligned} \nabla_{\eta} \mathbb{E}_{p(\tau)} [D_\alpha(\pi_\eta \| r_\tau)] &= \nabla_{\eta} \mathbb{E}_{p(\tau)} \left[\frac{\mathbb{E}_{\beta(z|\tau)} \left[\frac{\pi_\eta(z|\tau)}{\beta(z|\tau)} \left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \log \frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right]}{\alpha \mathbb{E}_{\beta(z|\tau)} \left[\frac{\pi_\eta(z|\tau)}{\beta(z|\tau)} \left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right]} \right. \\ &\quad \left. - \frac{1}{\alpha^2} \log \mathbb{E}_{\beta(z|\tau)} \left[\frac{\pi_\eta(z|\tau)}{\beta(z|\tau)} \left(\frac{R_\tau(z)}{\pi_\eta(z|\tau)} \right)^\alpha \right] \right]. \quad (19) \end{aligned}$$

Because expectations are with respect to $p(\tau)$ and $\beta(z|\tau)$, neither of which depend on η , we avoid the need to invoke a reparameterization trick in gradient estimation. This supports both on-policy (i.e., for $\beta(z|\tau) = \text{stop_grad}(\pi_\eta(z|\tau))$) and off-policy learning (e.g., using a replay buffer to store past episodes).

Given an episode $\{(z_i, \tau_i, \log \beta(z_i|\tau_i), \log R_{\tau_i}(z_i))\}_{i=1}^n$ of n evaluations, we express the MC objective as a numerically stable log-sum-exp, making iteration on (19) reliable in practice and avoiding under- or overflow.

Importantly, expression (19) does not depend on η ; we do not require that the reward $R_\tau(z)$ is differentiable with respect to z for policy gradient estimation. We can therefore incorporate numerous molecular properties as constraints without the need for complex modifications or approximations, such as surrogate models. However, in cases where evaluating molecular properties is costly, surrogate models can be employed.

Code Listing 1: A function for numerically stable MC estimation of $D_\alpha(\pi_\eta \| r_\tau)$ given $\log \pi_\eta(z|\tau)$, $\log \beta(z|\tau)$, and $\log R_\tau(z)$ evaluated over a minibatch of n samples drawn from $\beta(z|\tau)$.

```
def alpha_divergence(log_pi, log_reward, log_beta=None, alpha=2.0):
    # If behavior policy is unspecified, assume that it's on-policy
    log_beta = log_pi if log_beta is None else log_beta

    # Verify shapes, get batch size
    assert log_pi.shape == log_beta.shape == log_reward.shape
    n, = log_pi.shape

    # Get the log importance weights
    log_weights = log_pi - log_beta.detach()

    # Get the log (unnormalized) likelihood ratios
    log_ratios = log_reward.detach() - log_pi

    # Numerically stable estimator for the derived alpha-divergence
    logits = log_weights + alpha * log_ratios
    wavg = (logits.softmax(0) * log_ratios).sum(0)
    wlme = (logits - math.log(n)).logsumexp(0) / alpha
    return (wavg - wlme) / alpha
```

A.4 Diversity sampling

Suppose that we exactly know the constraint-satisfying set $X_{\mathcal{D}}^*$. Depending on the constraints, it may be highly likely that sampling k compounds uniformly at random from $X_{\mathcal{D}}^*$ results in far fewer than k distinct compound clusters, e.g., when using Butina clustering [2] with ECFP4 similarity. This can happen when the library includes clusters that contain a very large number of analogues that are constraint-satisfying. For the policy, this can manifest in probability density being more heavily concentrated in regions where many latent codes map to compounds that earn a large (expected) reward, such that sampling *i.i.d.* from the policy induces compound sets that are surprisingly lacking in diversity, even if the policy has correctly identified other relevant modes.

To address this, we can try to sample from an alternative distribution informed by the policy, but defined over a *set* of latent codes such that (i) they mutually attain a high log likelihood under the policy and (ii) they are sufficiently spread out to encourage coverage of distinct modes of the policy. Ideas similar to this have been proposed in prior work, which we take as inspiration [13, 22].

We can formalize these preferences with an energy function,

$$E(z_1, \dots, z_k) = - \sum_{i=1}^k \log \pi_\eta(z_i | \tau = 0) + \lambda \sum_{i=1}^{k-1} \sum_{j=i+1}^k \max(\delta - \|z_i - z_j\|_2^2, 0), \quad (20)$$

where $\delta > 0$ is a hyperparameter specifying the tolerated proximity between latent codes and $\lambda > 0$ is a hyperparameter that governs the penalty for violating the proximity constraint. Our interest is in the diversified policy, which satisfies $\tilde{\pi}_\eta(z_1, \dots, z_k) \propto \exp(-E(z_1, \dots, z_k))$.

To sample from $\tilde{\pi}_\eta$, we first randomly initialize k latent codes by sampling from the policy π_η and we then run stochastic gradient Langevin dynamics (SGLD) [21] on the latent codes for T iterations using $-E(z_1, \dots, z_k)$ as the target unnormalized log density. While the proximity term in the energy function requires $O(k^2)$ distance evaluations, we can form an MC approximation using a subset of $\ell \ll (k-1)k/2$ pairs sampled randomly each SGLD step to ensure tractability when k is large.

A.5 Discussion and limitations

Casting optimization in design space as optimization in the latent space of a trained auto-encoder is a well-known technique in machine learning [7, 8]. Nevertheless, our contributions, a policy that enriches for mass covering of the target distribution instead of mode seeking and which does not require knowledge of the unknown normalizing constant for the tempered reward function, coupled with a decoder constrained to vast catalogs addresses an important problem in drug discovery: multi-parameter design optimization that generates synthetically accessible compounds.

Our approach scales favorably as the size of the CSL grows. We hypothesize that the latent dimension d does not need to grow dramatically to accommodate larger library sizes $|X_{\mathcal{D}}|$.¹ Again, we have not formalized, for example, how large d must be to accommodate a library \mathcal{D} of a particular size $|X_{\mathcal{D}}|$, but it has been empirically demonstrated that modest choices are sufficient to attain suitably high reconstruction accuracy on ultra-large libraries.

Nonetheless, our approach also has limitations. We have not demonstrated that the proposed policy over the latent space Z adequately induces a uniform distribution over the constraint-satisfying subset of the library $X_{\mathcal{D}}^* \subset X_{\mathcal{D}}$. An exact characterization would require certain (stringent) assumptions to hold, such as regarding extent to which the decoder $p_{\theta}(x|z, \mathcal{D})$ approximately inverts the encoder $q_{\psi}(z|x)$ within the essential support of $q_{\psi}(z|\mathcal{D})$. Nevertheless, our approach provides a framework for casting what is otherwise an extensive discrete search problem over $X_{\mathcal{D}}$ as a tractable continuous search problem over \mathbb{R}^d and is sufficiently general to support a large class of virtual screening formulations applied to CSLs.

As a final note, to ensure that the policy $\pi_{\eta}(z|\tau)$ remains within the essential support of the aggregated variational posterior where the decoder approximately inverts the encoder, we augmented $R_{\tau}(z)$ by introducing an additional constraint that the log probability density under the aggregated variational posterior is sufficiently high, $\log q_{\psi}(z|\mathcal{D}) > \kappa$. In practice, we might instead choose to impose the constraint on the latent space prior, $\log p(z) > \kappa$, or on a density fit to a finite sample of latent codes drawn from $(x, z) \sim q_{\psi}(z|x)p(x|\mathcal{D})$, e.g., a Gaussian mixture model.

B Experiments

B.1 Synthetic example

We demonstrate our approach with a synthetic example. Given dimensionality d , let $W \in \mathbb{R}^{d \times d}$ be a random orthonormal matrix, $W \in \mathbb{R}^{d \times d}$ s.t. $W^{\top}W = WW^{\top} = I$. The latent space and design space are related through rotation by W , i.e., $x := Wz$. Our goal is to learn a policy $\pi_{\eta}(z|\tau = 0)$ that samples vertices of the d -dimensional signed hypercube after rotation by W . Specifically, define the binarized reward $R_0(x) = \mathbb{I}[x \in \{\pm 1\}^d]$. Thus,

$$R_0(z) = \begin{cases} 1, & \text{if } Wz =: x \in \{\pm 1\}^d; \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

We can define an associated tempered reward by

$$R_{\tau}(z) = \prod_{i=1}^d \sigma \left(-\frac{(x_i^2 - 1)^2}{\tau} \right). \quad (22)$$

Thus, in the zero temperature limit, the optimal policy $\pi_{\eta}(z|\tau = 0)$ should induce a uniform distribution over 2^d distinct modes of the design space. To assess the extent to which the policy iterates capture these various modes of the reward function, we track the log density under the worst-, average-, and best-case modes:

$$S_{\min}(\eta) = \min_{x \in \{\pm 1\}^d} \log \pi_{\eta}(z = W^{\top}x|\tau = 0); \quad (23)$$

$$S_{\text{avg}}(\eta) = \frac{1}{2^d} \sum_{x \in \{\pm 1\}^d} \log \pi_{\eta}(z = W^{\top}x|\tau = 0); \quad (24)$$

$$S_{\max}(\eta) = \max_{x \in \{\pm 1\}^d} \log \pi_{\eta}(z = W^{\top}x|\tau = 0). \quad (25)$$

The reward-induced density can be approximated by a mixture of isotropic Gaussians whose centers are given by $\{W^{\top}x : x \in \{\pm 1\}^d\}$. Letting $\sigma^2 > 0$ denote the component-wise variance, the approximate density follows:

$$r_{\sigma}(z) = \frac{1}{2^d} \sum_{x \in \{\pm 1\}^d} \text{Normal}(z - W^{\top}x|\mu = 0, \Sigma = \sigma^2 I). \quad (26)$$

¹Even a modest $d = 64$ dimensions has been shown to be sufficient to train a CSLVAE model with top-1 reconstruction accuracy in excess of 60% for libraries on the order 10^{10} molecules [16].

For $x \in \{\pm 1\}^d$ and for small $\sigma^2 > 0$, the log density for the induced $z := W^\top x$ is well approximated by $T_\sigma = d \log \text{Normal}(0|0, \sigma^2) - d \log(2)$. Hence, T_σ quantifies the log density that the mixture of Gaussians approximation to the ground truth discrete distribution assigns to each of the 2^d modes of the reward function. We can compare (23) - (25) to T_σ for different choices of α to assess how α affects worst-, average-, and best-case coverage of modes over the course of policy optimization.

For this example, we set $d = 10$ and parameterize π_η by a neural spline flow with eight rational quadratic spline flow coupling layers with eight knots each, with a fixed but randomly-initialized permutation applied after each coupling layer to mix units.

We examine a range of α -divergences by sweeping $\alpha = 2^j$ for $j = -2, \dots, 4$. This allows us to assess the performance for values of $\alpha \geq 1$ that induce the mass-covering behavior and demonstrates the optimization challenges encountered for values of $\alpha < 1$ that induce the mode-seeking behavior.

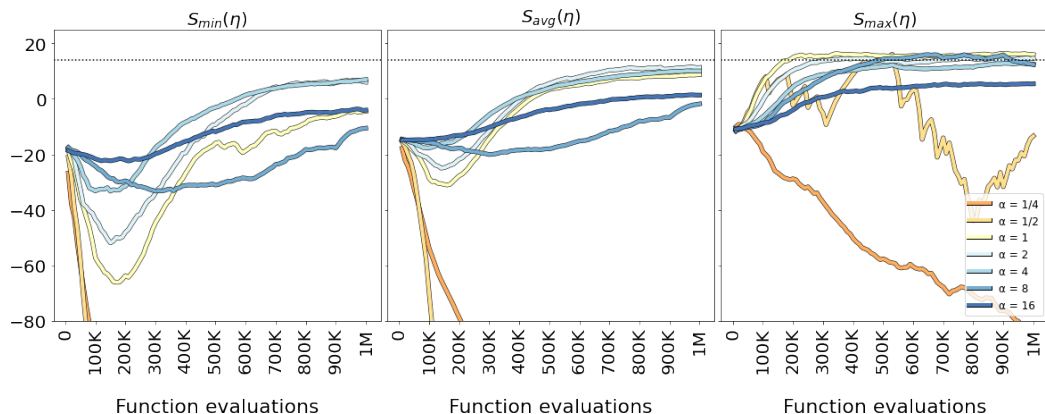
Training is performed over episodes, each of which is comprised of 1,000 pairs of the form $(z, \tau) \sim \beta(z|\tau)p(\tau)$ along with the corresponding tempered reward $R_\tau(z)$, where β is the behavior policy. We consider both the on-policy and off-policy regimes. In the off-policy setting, we maintain a replay buffer of the past 10 episodes and train the policy for 10 iterations on the replay buffer before sampling a new episode. In either setting, we train the policy until we reach a maximum of 1,000 episodes, which is equivalent to one million function evaluations. Before commencing policy optimization, we pre-train it for 500 iterations to match an isotropic standard Gaussian prior.

Supplementary Figures 1 (off-policy) and 2 (on-policy) shows how different choices of α affect coverage of the worst-, average-, and best-case modes during the course of policy optimization. We run all models for a total of 10M parameter updates, equating to 1M (cf. 10M) function evaluations for the off-policy (cf. on-policy) setting. We observed qualitative similarities in model performance as a function of α in both regimes, but observe slower convergence and greater variability in the on-policy setting as a result of the poorer sample efficiency from the lack of replay.

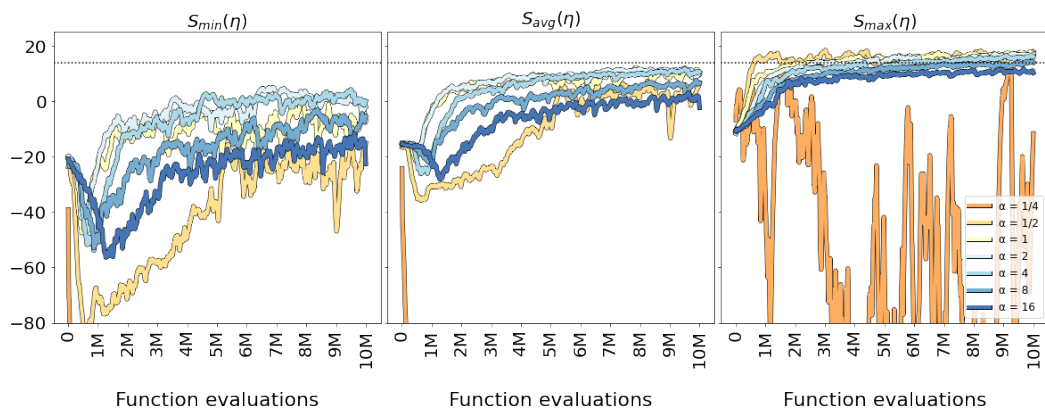
All of the mass-covering choices $\alpha \geq 1$ demonstrate progress in covering the various modes of the reward function relative to the initialized policy. As $\alpha \geq 1$ increases, these curves progressively flatten, suggesting that policy iterations at higher α values redistribute probability density more gradually. This aligns with the mass-covering inductive bias, which aims to avoid completely zero-ing out policy density in any region.

At lower values of $\alpha \geq 1$, coverage of worst- and average-case modes is sacrificed in the early stages of policy optimization at the expense of progress on the best-case modes. Moderate values like $\alpha = 2$ manage a good balance of this trade-off by ensuring the gap $S_{\max}(\eta) - S_{\min}(\eta)$ remains fairly small over the course of policy optimization. We note that the expression for the derived divergence when $\alpha = 2$ has, as one of its terms, the (scaled) Chi-squared divergence, which is relevant in the importance sampling literature due to its variance reduction characteristics [14], and may purportedly explain its more balanced behavior across the worst-, average-, and best-case modes relative to the forward KL at $\alpha = 1$, but we have not investigated this rigorously. For mode-seeking choices of $\alpha < 1$, the policy iterates show largely divergent behavior. In the case of $\alpha = 1/2$, we observe that on-policy training is able to quickly identify one of the modes but in comparison to its nearest mass-covering relative at $\alpha = 1$ dramatically sacrifices in its worst- and average-case coverage; in contrast, the policy iterates exhibit highly non-stationary behavior in the off-policy regime. These observations are consistent with prior work that has demonstrated some of the challenges in optimizing policies using mode-seeking divergences and of training instabilities introduced through importance sampling formulations of off-policy objectives in the absence of protections such as clipping [14, 19, 20].

Supplementary Figure 3 shows pairplots for samples drawn from $z \sim \pi_\eta(z|\tau = 0)$ and their induced $x := Wz$ for the model trained in the off-policy setting using the $\alpha = 2$ divergence. We note that the policy has successfully learned a correlated and multi-modal distribution over the latent space which induces an approximately uniform distribution (i.e., $S_{\max}(\eta) - S_{\min}(\eta) \approx 0$) via decoding with W to the constraint-satisfying region of the design space (namely, the vertices of the signed hypercube).



Supplementary Figure 1: Off-policy optimization in the synthetic example ($d = 10$). Policy iterations = 10, replay buffer size = 10. Dashed line corresponds to T_σ for $\sigma = 0.05$.

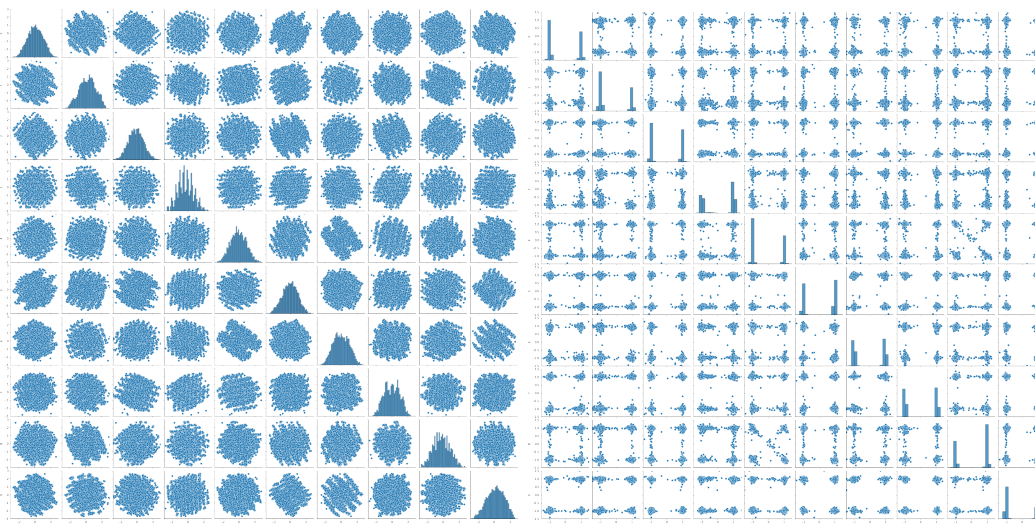


Supplementary Figure 2: On-policy optimization for the synthetic example ($d = 10$). Policy iterations = 1, replay buffer size = 1. Dashed line corresponds to T_σ for $\sigma = 0.05$.

B.2 Satisfying multiple molecular property constraints in an ultra-large CSL

We provide additional details concerning the section of the same name in the body of the paper. The hierarchical enumeration baseline was adapted from the V-SYNTHES method [18] but with some important differences. The V-SYNTHES method is effectively an approach for optimizing a docking score over large compound libraries using a beam search that exploits the library’s hierarchical structure. In our experiment, we apply such a beam search but rather than use a docking score as the optimization target, we instead define the optimization target to be the log of the tempered reward as defined in the paper, where we set the temperature sufficiently low, e.g., $\tau = 0.001$. This allows us to compare the performance of our proposed algorithm with a V-SYNTHES-like beam utilizing a similarly defined reward for guiding expansion.

For the library that we consider in this exercise, the minimal enumeration set is comprised of 260K compounds. This set represents "fragment-like compounds representing all possible scaffold-synthon combinations for all reactions" in the library [18] and serves as a basis for expansion in subsequent steps of the algorithm. The entirety of the minimal enumeration set is scored, representing a one-time cost of 260K function evaluations that can be re-used across different runs. Given a particular set of constraints, we compute the corresponding log (tempered) reward for the compounds in the minimal enumeration set and select the top k compounds by reward. Each of these compounds will have a number of *caps*, representing placeholders in need of synthon assignment. In the subsequent step of the algorithm, for the top k compounds, we sample m eligible synthons to use in place of each cap, and we evaluate the log reward on all of the resulting compounds. We again select the top



Supplementary Figure 3: Pairplot for the synthetic example ($d = 10$, $\alpha = 2$, off-policy). **Left:** Latent space samples generated by $z \sim \pi_\eta(z|\tau = 0)$. **Right:** Design space samples generated by transformation $x := Wz$ of latent space samples.

k compounds from that iteration and continue until we finally arrive at capless compounds (i.e., products with all synthons assigned). The choice of k and m affects the total number of function evaluations performed, and it is worth remembering that many function evaluations are performed in the presence of caps, indicating incomplete molecules; thus, in Table 1, we distinguish between the total number of function evaluations and the number without caps (i.e., of complete molecules). We set k to be sufficiently large so as to attain sufficient diversity (indeed, k can be considered a kind of upper bound on the diversity attainable by such a hierarchical approach) and then adjust m accordingly so that, across the experiments considered, we allow the hierarchical approach and our proposed policy optimization a similar number of function evaluations (roughly 100K).

We believe that the apparently poor performance of the hierarchical approach in this setting is likely due to the pre-mature optimization of the reward when building the molecule in a synthon-by-synthon fashion. Such a greedy selection procedure can lead to intermediate states that are constraint-satisfying but where all remaining cap replacements would lead to constraint violation (thereby earning a lower reward in the terminal iteration). In the absence of protocols for back-tracking, these situations can be very difficult to remedy. We made attempts to address this by initially tightening the constraints and then relaxing when there are fewer caps remaining. However, given that different filters will have different numbers of constraints, and the corresponding functions vary in units, among a host of other considerations, it was our experience that an obvious/general approach to codify iterative constraint relaxation proved elusive, and our best attempts to tune such a schedule did not reliably improve performance.

Supplementary Table 1 provides details on the constraints used by the filters in Table 1. We use the RDKit [9] implementation of all of the referenced properties.

Name	Constraints
Lipinski [12]	Molecular weight ≤ 500 CLogP ≤ 5 No. hydrogen bond donors ≤ 5 No. hydrogen bond acceptors ≤ 10
Ghose [6]	$160 \leq$ Molecular weight ≤ 480 $-0.4 \leq$ CLogP ≤ 5.6 $20 \leq$ No. atoms ≤ 70 $40 \leq$ Molar refractivity ≤ 130
Lee [10]	$300 \leq$ Molecular weight ≤ 400 $1.8 \leq$ CLogP ≤ 2.3
Rule-of-3 [3]	Molecular weight ≤ 300 CLogP ≤ 3 No. hydrogen bond donors ≤ 3 No. hydrogen bond acceptors ≤ 3 No. rotatable bonds ≤ 3
Macrocycles [5]	Molecular weight ≤ 982 $2.2 \leq$ CLogP No. hydrogen bond donors ≤ 7 No. rotatable bonds ≤ 11 $12 \leq$ Max ring size Topological PSA ≤ 292
QED [1]	$0.8 \leq$ QED

Supplementary Table 1: Filters used for experiments recorded in Table 1.

References

- [1] 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.
- [2] Darko Butina. Unsupervised data base clustering based on Daylight’s fingerprint and Tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, 39(4):747–750, 1999.
- [3] Miles Congreve, Robin Carr, Chris Murray, and Harren Jhoti. A “rule of three” for fragment-based lead discovery? *Drug Discovery Today*, 8(19):876–877, 2003.
- [4] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in Neural Information Processing Systems*, 32, 2019.
- [5] Diego Garcia Jimenez, Vasanthanathan Poongavanam, and Jan Kihlberg. Macrocycles in drug discovery: Learning from the past for the future. *Journal of Medicinal Chemistry*, 66(8):5377–5396, 2023.
- [6] Arup K Ghose, Vellarkad N Viswanadhan, and John J Wendoloski. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery: A qualitative and quantitative characterization of known drug databases. *Journal of Combinatorial Chemistry*, 1(1):55–68, 1999.
- [7] 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.
- [8] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [9] Greg Landrum et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. 2013.

- [10] Man-Ling Lee and Gisbert Schneider. Scaffold architecture and pharmacophoric properties of natural products and trade drugs: application in the design of natural product-based combinatorial libraries. *Journal of Combinatorial Chemistry*, 3(3):284–289, 2001.
- [11] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. *Advances in Neural Information Processing Systems*, 29, 2016.
- [12] Christopher A Lipinski. Lead-and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, 1(4):337–341, 2004.
- [13] Yecheng Jason Ma, Jeevana Priya Inala, Dinesh Jayaraman, and Osbert Bastani. Diverse sampling for normalizing flow based trajectory forecasting. *arXiv preprint arXiv:2011.15084*, 2020.
- [14] Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. *arXiv preprint arXiv:2208.01893*, 2022.
- [15] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.
- [16] Aryan Pedawi, Pawel Gniewek, Chaoyi Chang, Brandon Anderson, and Henry van den Bedem. An efficient graph generative model for navigating ultra-large combinatorial synthesis libraries. *Advances in Neural Information Processing Systems*, 35:8731–8745, 2022.
- [17] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, pages 1530–1538. PMLR, 2015.
- [18] Arman A Sadybekov, Anastasiia V Sadybekov, Yongfeng Liu, Christos Iliopoulos-Tsoutsouvas, Xi-Ping Huang, Julie Pickett, Blake Houser, Nilkanth Patel, Ngan K Tran, Fei Tong, et al. Synthon-based ligand discovery in virtual libraries of over 11 billion compounds. *Nature*, 601(7893):452–459, 2022.
- [19] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, pages 1889–1897. PMLR, 2015.
- [20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [21] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, pages 681–688. PMLR, 2011.
- [22] Ye Yuan and Kris Kitani. DLow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision (ECCV)*, pages 346–364. Springer, 2020.