Ligand Iterative Sampling for Affinity Refinement and Drug Discovery (LISARDD)

Valentin Badea [1] **Shyam Chandra** [1] **John Lin** [1]

Abstract

De novo drug generation is a challenging task that aims to generate novel molecules with specific properties from scratch. Deep learning can accelerate this process by efficiently exploring the drug-like chemical space. Here, we introduce LISARDD, a Reinforcement Learning framework to optimize sampling in the latent space of a pretrained target-agnostic generative model. We demonstrate that our approach can generate candidate molecules that simultaneously optimize multiple drug properties, including target-specific binding affinity, drug-likeness, and synthetic accessibility. This fully modular framework can leverage any molecular generative model, binding affinity scoring model, or optimization algorithm to identify novel drug candidates for future experimental validation.

1. Introduction

Drug development is a time-consuming and expensive process. On average, developing a new drug takes 8.3 years and $1.3 billion to reach market approval (Wouters et al., 2020). Deep learning can accelerate this process at various steps, including *de novo* drug generation, or the process of generating novel molecules that bind to a specific target protein. Several conditional generative models have therefore been developed to identify candidate molecules (Ragoza et al., 2022; Peng et al., 2022). However, these approaches typically involve computationally expensive training to infer target-specific conditional distributions (Zhang et al., 2024). Alternatively, *de novo* drug generation can be framed as a Reinforcement Learning (RL) problem, in which an agent learns how to navigate the latent space of a generative model and identify molecules that satisfy certain properties, such as drug-target binding affinity. Here, target specificity is adapted at training time without retraining the generative model. Recently, Haddad et al. introduced MOLRL, an RL framework that uses Proximal Policy Optimization (PPO) to sample a generative model's latent space (2025). MOLRL was able to identify molecules with biological activity for two target proteins (GSKβ3 and JNK3), where binding affinity was estimated using target-specific scoring models.

Here, we extend the work of Haddad et al. and introduce Ligand Iterative Sampling for Affinity Refinement and Drug Discovery (LISARDD)[1], an RL framework that optimizes sampling in the latent space of a pretrained target-agnostic generative model. Our approach applies a target-agnostic scoring model to predict drug-target binding affinity and guide sampling. We demonstrate that LISARDD can generate candidate molecules that simultaneously optimize multiple drug properties. By incorporating multiple properties into the reward function, we can reduce failure rates in subsequent stages of drug development. We emphasize the modularity of our approach, which can easily implement any molecular generator, target objective, or optimization algorithm. Thus, any improvement to an individual module can be integrated into the overall framework.

2. Methodology

An overview of the LISARDD framework is presented in Figure 1. Briefly, the generative model is a pretrained, target-agnostic encoder-decoder, where the decoder converts a latent vector z to a molecule. A reward function then scores the generated molecule. Here, a scoring model can be applied to assess binding affinity between a candidate ligand and a target protein, without retraining the generative model. Then, the agent perturbs the latent vector z through an action a and learns how to navigate the generative model's latent space to optimize the reward function.

We evaluated two RL algorithms: REINFORCE and PPO (Williams, 1992; Schulman et al., 2017). REINFORCE is a simple and efficient on-policy algorithm, whereas PPO is more state-of-the-art.

[1]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. Correspondence to: Valentin Badea <valentin_badea@hms.harvard.edu>.

Proceedings of the ICML 2025 Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences, Vancouver, Canada. 2025.

[1]Code and data are available at https://github.com/valbad/LISARDD

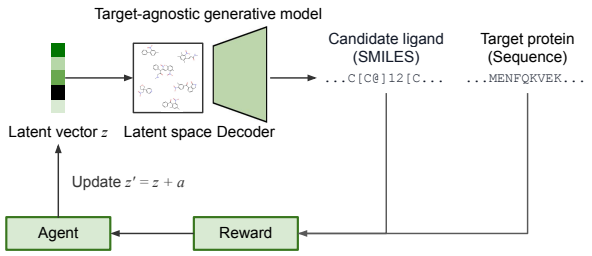

Figure 1. LISARDD framework.

2.1. Proximal Policy Optimization

Drawing on the work of Haddad et al. (2025), we reverse-engineered the PPO approach described in their paper and applied it to our targeted generation task. Our PPO leverages an actor-critic architecture. Both networks include a 3-layer multilayer perceptron (MLP) entry with ReLU activations and batch normalization. The actor head outputs a deterministic shift $\mu_\theta(z)$ and a standard deviation vector $\sigma_\theta(z)$, while the critic head outputs a scalar reward estimate $V_\phi(z)$. The RL agent updates the policy as:

$$z' = z + \mu_\theta(z) + \sigma_\theta(z) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

where a latent vector z is perturbed through an action $a \sim \mathcal{N}\left(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2\right)$. Here, the standard deviation vector that controls the exploration-exploitation trade-off is learned over time.

At every iteration t, our PPO algorithm aims to minimize a total loss term, defined as:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}_t \left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right]$$

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[(\text{Reward} - V_\phi(z))^2]$$

$$\mathcal{L}_{\text{entropy}} = -\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\text{Entropy}(\mathcal{N}(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2\right]$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{actor}} + 0.5\mathcal{L}_{\text{critic}} + 0.01\mathcal{L}_{\text{entropy}} \tag{2}$$

See Appendix A for more details regarding our implementation of PPO.

2.2. REINFORCE

REINFORCE aims to minimize the following loss:

$$\mathcal{L}_{\text{REINFORCE}} = -\mathbb{E}_{\pi_\theta}[\text{Reward}] \tag{3}$$

where π_θ represents a parametric Gaussian policy over the same continuous action space. Our actor network (defined similarly as in the previous section) outputs a shift $\mu_\theta(z)$ and standard deviation vector $\sigma_\theta(z)$. At every training step, the action a the agent takes is:

$$a = \mu_\theta(z) + \sigma_\theta(z) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{4}$$

Then, we approximate the policy gradient using the REINFORCE trick as:

$$\nabla_\theta \mathcal{L}_{\text{REINFORCE}} \approx -\mathbb{E}_{z\sim\pi_\theta}[\text{Reward} \times \nabla_\theta \log \pi_\theta(z)] \tag{5}$$

See Appendix B for more details regarding our implementation of REINFORCE.

2.3. Reward Functions

To guide the RL agent toward generating biologically relevant molecules, we implemented a variety of reward functions. First, we started with a function that rewards drug-likeness. Here, we used quantitative estimation of drug-likeness (QED), whose values range from 0 to 1, with 1 being the most drug-like (Bickerton et al., 2012).

$$R_{\text{QED}} = \text{QED}$$

Second, we defined a function that rewards synthetic accessibility (SA). SA values range from 1 to 10, where molecules with higher values are harder to synthesize (Ertl & Schuffenhauer, 2009). Here, we reverse-normalized SA values to the range $[0, 1]$:

$$R_{\text{SA}} = 1 - 0.1 \cdot \text{SA}$$

Next, we focused on drug-target binding affinity. Given a candidate ligand and a target protein, the scoring model predicts a dissociation constant $\text{p}K_\text{d}$ (log scale), where higher values indicate greater binding affinity.

$$R_{\text{aff}} = \text{p}K_\text{d}$$

Finally, we constructed a multi-objective reward function that integrates SA, QED, and drug-target binding affinity. We binarized binding affinity values to the range [0, 1] by rewarding high $\text{p}K_\text{d}$ values within a biologically plausible range [7, 14] (thus discouraging reward hacking and candidates with impossible affinity).

$$R_{\text{aff}}^{\text{bin}} = \begin{cases} 1 & \text{if } 7 \le \text{p}K_\text{d} \le 14 \\ 0 & \text{otherwise} \end{cases}$$

In fact, we derived a differentiable version of this score:

$$R_{\text{aff}}^{\text{bin-diff}} = \frac{1}{1 + e^{-s(\text{p}K_\text{d}-7)}} \times \frac{1}{1 + e^{s(\text{p}K_\text{d}-14)}} \tag{6}$$

where s is a hyperparameter controlling the steepness of the transitions between 0 and 1. Then, we defined the multi-objective reward function as:

$$R_{\text{MO}} = w_1 R_{\text{SA}} + w_2 R_{\text{QED}} + (1 - w_1 - w_2) R_{\text{aff}}^{\text{bin-diff}}$$

where w_1 and w_2 are weights that can be adjusted to prioritize different objectives. By default, w_1 and w_2 were set to 0.1 to ensure that the binding affinity reward drives molecular optimization.

2.4. Generative Model

To generate molecules, we used HierVAE, a hierarchical variational autoencoder that autoregressively aggregates motifs to emerging molecules (Jin et al., 2020). We chose this model as it generates molecules with high validity, uniqueness, and diversity. Here, the HierVAE model was pretrained on the ChEMBL dataset (Zdrazil et al., 2024), and its latent space represents each molecule with three 32-dimensional vector embeddings. We hypothesized that HierVAE can access more advanced latent representations of small molecule ligands compared to the generative models evaluated in Haddad et al.

2.5. Scoring Model

To assess drug-target binding affinity between any given pair of target protein and small molecule ligand, we used MGraphDTA, a deep multiscale graph neural network (Yang et al., 2022). Across multiple benchmark datasets, MGraphDTA outperformed other machine learning models. In our implementation, we trained the MGraphDTA model on the Davis dataset (2011). Predicted drug-target binding affinities from the MGraphDTA model were validated with the industry standard AutoDock Vina (Trott & Olson, 2010). To the best of our knowledge, no past work has made use of RL-driven latent molecular optimization using an affinity reward as comprehensive as MGraphDTA.

2.6. Target Proteins

We evaluated our framework on two target proteins: human c-Jun N-terminal kinase 3 (JNK3) and *Escherichia coli* DNA gyrase subunit A (gyrA). JNK3 is an enzyme that regulates apoptosis in neurons and is a thereapeutic target for Alzheimer's disease (Solas et al., 2023). On the other hand, DNA gyrase is an enzyme that modulates DNA supercoiling, and gyrA mutations are associated with antibiotic resistance (Weigel et al., 1998). Structural data for JNK3 (PDB ID: 3FI2) is available on RCSB PDB (Habel, 2008; Kamenecka et al., 2009).

2.7. Framework Evaluation

To assess whether PPO and REINFORCE can learn to effectively sample the latent space of a generative model, we reported average batch-level reward trajectories across epochs. In addition, we recorded the top 100 highest reward molecules that were generated during RL optimization and calculated the following metrics: QED, SA, drug-target binding affinity, and Tanimoto similarity (TS), where TS reflects the structural novelty of the generated molecules (Bajusz et al., 2015). Lastly, we sampled 100 shared latent vectors and applied each trained policy (PPO or REINFORCE) to independently perturb them. The resulting latent vectors were decoded into molecules, and their predicted binding affinities were evaluated using a shared scoring function. Then, we performed a paired *t*-test to determine statistical significance.

3. Results

3.1. Single-Objective Reward Optimization

We first applied PPO and REINFORCE to optimize single-objective reward functions that reflect physicochemical properties. We demonstrate that both PPO and REINFORCE were able to generate molecules with strong QED or SA profiles (Figure 2).

Next, we evaluated whether PPO and REINFORCE can learn to identify candidate molecules with high biological activity for a given target protein. Here, REINFORCE outperformed PPO in optimizing binding affinity to JNK3 (Figure 3). However, the molecules generated by both PPO and REINFORCE exhibited poor QED and SA profiles (Figure 4, Table 1). For gyrA, PPO and REINFORCE achieved similar performance in binding affinity optimization.

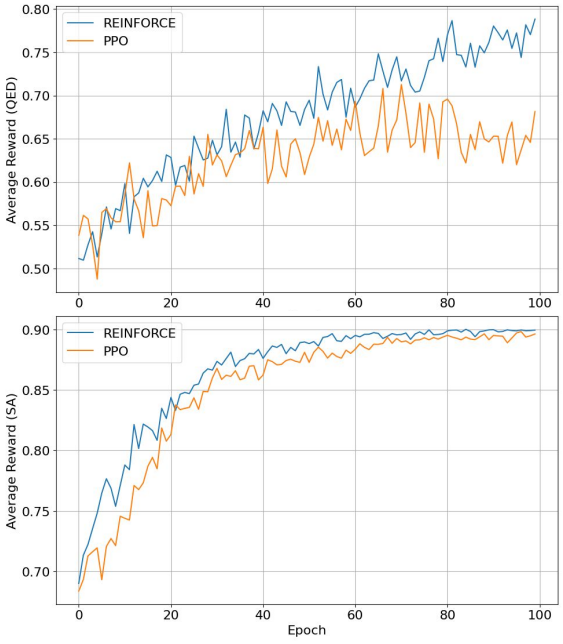

Figure 2. Comparison of reward trajectories across epochs between REINFORCE and PPO. Here, the reward function is QED (top) or SA (bottom).

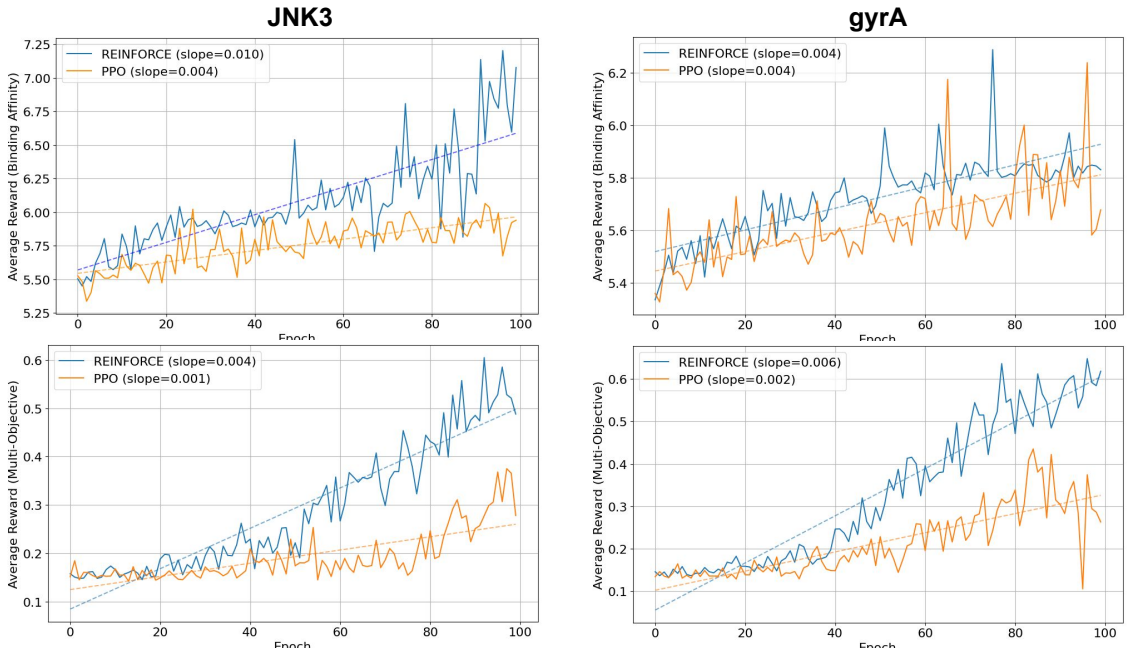

Figure 3. Comparison of reward trajectories across epochs between REINFORCE and PPO. The single-objective function (top left) rewards binding affinity to JNK3 ($p = 4.00 \times 10^{-5}$), whereas the multi-objective function (bottom left) rewards binding affinity to JNK3, drug-likeness, and synthetic accessibility ($p = 5.74 \times 10^{-5}$). The single-objective function (top right) rewards binding affinity to gyrA ($p = 0.65$), whereas the multi-objective function (bottom right) rewards binding affinity to gyrA, drug-likeness, and synthetic accessibility ($p = 3.04 \times 10^{-8}$).

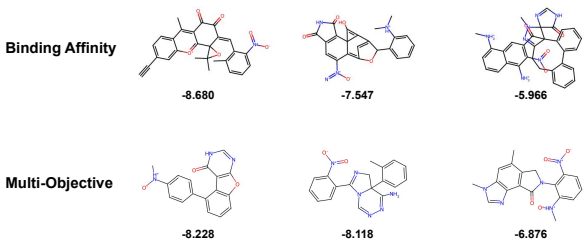

Figure 4. Lead molecules generated by REINFORCE for different reward functions given target protein JNK3. AutoDock Vina scores (kcal/mol) are shown for each molecule.

3.2. Multi-Objective Reward Optimization

Finally, we assessed whether PPO and REINFORCE can simultaneously optimize a multi-objective reward function that integrates drug-target binding affinity, QED, and SA. Again, REINFORCE outperformed PPO (Figure 3). Both methods successfully identified molecules with high binding affinity to JNK3 while simultaneously improving QED and SA (Figure 4, Table 1). Similar results were observed for gyrA.

4. Discussion

Our work demonstrates that RL frameworks can effectively explore the latent space of a molecular generator to optimize physicochemical properties, and more specifically, reward schemes based on binding affinity estimation. We observed a broad upward trend in all reward trajectories for both PPO and REINFORCE, which suggests that training for more epochs could lead to greater policy improvements. To the best of our knowledge, no previous work has incorporated a generalizable deep learning model for binding affinity prediction as a reward function within reinforcement learning.

From a drug discovery perspective, the most promising reward strategy is the multi-objective function that jointly optimizes drug-target binding affinity, QED, and SA. This approach generated lead candidates that were not only strong predicted binders but also structurally and synthetically reasonable, even when the weights associated with these terms were relatively small (respectively 10% each). External validation with AutoDock Vina further supported the biological plausibility of these candidates, offering a complementary docking-based signal aligned with predicted pK_d (Figure 4). These results suggest that reward composition and constraint tuning are critical in generative molecular optimization, especially when the scoring model is approximate. Additionally, the relatively low pairwise similarity among

Table 1. Summary statistics of the top 100 molecules generated by REINFORCE or PPO during RL optimization for different reward functions. Mean values are shown for drug-likeness (QED), synthetic accessibility (SA), binding affinity to JNK3 or gyrA (pK_D), and Tanimoto similarity (TS).

REWARD	ALGORITHM	TARGET	QED ↑	SA ↓	pK_D ↑	TS ↓
BINDING AFFINITY	REINFORCE	JNK3	0.15	5.47	13.91	0.13
MULTI-OBJECTIVE	REINFORCE	JNK3	0.59	3.29	9.16	0.21
BINDING AFFINITY	PPO	JNK3	0.34	4.05	9.84	0.15
MULTI-OBJECTIVE	PPO	JNK3	0.52	3.69	9.29	0.19
BINDING AFFINITY	REINFORCE	GYRA	0.29	4.36	7.91	0.10
MULTI-OBJECTIVE	REINFORCE	GYRA	0.60	2.95	9.23	0.27
BINDING AFFINITY	PPO	GYRA	0.23	4.01	9.72	0.17
MULTI-OBJECTIVE	PPO	GYRA	0.56	4.59	9.35	0.16

the top 100 molecules suggests that both REINFORCE and PPO can identify structurally diverse high-performing candidates, supporting their potential for discovering novel small molecules.

We were surprised to observe that REINFORCE often outperformed PPO, a more advanced RL algorithm. This result contrasts with prior findings, such as those reported by Haddad et al. Our results show that both methods can hold very similar optimization trajectories, specifically when optimizing single rewards (Figure 2). On the other hand, complex multi-objective targets seem to be more difficult for PPO to optimize (Figures 3). A plausible explanation to this phenomenon could be that our generator's latent space is efficiently organized, allowing for rapid REINFORCE optimization. In the meantime, the cautious, clipped PPO updates may impose an upper boundary on the speed at which the model improves.

Extending our work to other molecular generators may allow us to explore this last hypothesis. More broadly, each component of our pipeline - including the learning algorithm, the molecular generator, and the reward function - can be replaced in a modular way, allowing for maximum flexibility. Our results demonstrate that the proposed approach can generate realistic small-molecule ligands targeting arbitrary protein sequences, while supporting a variety of reinforcement learning algorithms in non-differentiable optimization scenarios. Finally, our current validation pipeline can be further improved through experimental assays or ADMET profiling.

5. Conclusion

In this work, we introduce LISARDD, a novel approach for *de novo* drug generation that applies RL algorithms to efficiently navigate the latent space of a target-agnostic generative model. We demonstrate that our framework can identify small molecules with enhanced drug-target binding affinity, drug-likeness, and synthetic accessibility. Importantly, our approach is fully modular and can easily implement other generative models, scoring models, and optimization algorithms. Potential future directions therefore include exploring alternative generative or scoring models that may utilize different representations for proteins and small molecule ligands. Overall, our study provides a flexible framework for targeted molecular generation with multi-objective optimization to support drug development.

Acknowledgments

We thank Marinka Zitnik, Debora Marks, Courtney Shearer, Pascal Notin, and Artem Gazizov for helpful discussions and insightful feedback.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bajusz, D., Rácz, A., and Héberger, K. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7:20, 2015.

Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K., and Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29(11):1046–1051, 2011.

Ertl, P. and Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular

complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8, 2009.

Habel, J. E. Crystal structure of jnk3 with amino-pyrazole inhibitor, sr-3451, 2008. URL https://doi.org/10.2210/pdb3fi2/pdb.

Haddad, R., Litsa, E. E., Liu, Z., Yu, X., Burkhardt, D., and Bhisetti, G. Targeted molecular generation with latent reinforcement learning. *Scientific Reports*, 15(1):15202, 2025.

Jin, W., Barzilay, R., and Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Kamenecka, T., Habel, J., Duckett, D., Chen, W., Ling, Y. Y., Frackowiak, B., Jiang, R., Shin, Y., Song, X., and LoGrasso, P. Structure-activity relationships and x-ray structures describing the selectivity of aminopyrazole inhibitors for c-jun n-terminal kinase 3 (jnk3) over p38. *The Journal of biological chemistry*, 284(19):12853–12861, 2009.

Peng, X., Luo, S., Guan, J., Xie, Q., Peng, J., and Ma, J. Pocket2mol: Efficient molecular sampling based on 3d protein pockets. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Ragoza, M., Masuda, T., and Koes, D. R. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical Science*, 13(9):2701–2713, 2022.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017.

Solas, M., Vela, S., Smerdou, C., Martisova, E., Martínez-Valbuena, I., Luquin, M.-R., and Ramírez, M. J. Jnk activation in alzheimer's disease is driven by amyloid β and is associated with tau pathology. *ACS chemical neuroscience*, 14(8):1524–1534, 2023.

Trott, O. and Olson, A. J. 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.

Weigel, L. M., Steward, C. D., and Tenover, F. C. gyra mutations associated with fluoroquinolone resistance in eight species of enterobacteriaceae. *Antimicrobial agents and chemotherapy*, 42(10):2661–2667, 1998.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Wouters, O. J., McKee, M., and Luyten, J. Estimated research and development investment needed to bring a new medicine to market and 2009-2018. *JAMA*, 323(9): 844–853, 2020.

Yang, Z., Zhong, W., Zhao, L., and Chen, C. Y.-C. Graphdta: deep multiscale graph neural network for explainable drug-target binding affinity prediction. *Chemical Science*, 13(3):816–833, 2022.

Zdrazil, B., Felix, E., Hunter, F., Manners, E. J., Blackshaw, J., Corbett, S., de Veij, M., Ioannidis, H., Lopez, D. M., Mosquera, J. F., Magarinos, M. P., Bosc, N., Arcila, R., Kizilören, T., Gaulton, A., Bento, A. P., Adasme, M. F., Monecke, P., Landrum, G. A., and Leach, A. R. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. *Nucleic Acids Research*, 52(D1):D1180–D1192, 2024.

Zhang, Z., Yan, J., Huang, Y., Liu, Q., Chen, E., Wang, M., and Zitnik, M. Geometric deep learning for structure-based drug design: A survey, 2024.

A. Proximal Policy Optimization

Here, we provide more details on the PPO training process, drawing on previous work (Haddad et al., 2025):

1. **Sample Latent Vectors:** Initialize latent vectors $z \sim \mathcal{N}(0, I)$.
2. **Sample Actions:** Compute a deterministic shift $\mu_\theta(z)$ and standard deviation vector $\sigma_\theta(z)$, and sample an action $a \sim \mathcal{N}\left(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2\right)$.
3. **Decode Molecules:** Update latent vectors $z' = z + a$ and convert them into SMILES using the generative model's decoder. Invalid decoded SMILES are penalized with a constant penalty (set to -1 in our implementation).
4. **Compute Rewards:** Evaluate decoded SMILES using a reward function and compute the estimated reward $V_\phi(z)$.
5. **Policy Update:** The policy is updated with the clipped surrogate and critic losses over 1 time step and 6 update steps. We use a Generalized Advantage Estimation scheme (more precisely GAE$(\gamma, 0)$) to estimate the advantage A_t, with a discount factor $\gamma = 0.95$ (Schulman et al., 2015):

$$A_t = \text{Reward}(z) + \gamma V_\phi(z') - V_\phi(z) \tag{7}$$

To stabilize training, rewards are normalized to zero mean and unit variance prior to policy updates. Then, we compute the clipped surrogate loss:

$$\mathcal{L}_{\text{actor}} = -\mathbb{E}[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)]$$

where $r_t(\theta)$ is the probability ratio of the action between the old and updated policies and ϵ is the clipping hyperparameter. Next, we compute the critic loss:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{z\sim\mathcal{N}(0,I)}[(\text{Reward} - V_\phi(z))^2]$$

Since the standard deviation vector $\sigma_\theta(z)$ is set learnable, it is common place to consider the following entropy loss:

$$\mathcal{L}_{\text{entropy}} = -\mathbb{E}_{z\sim\mathcal{N}(0,I)}\left[\text{Entropy}\left(\mathcal{N}(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2\right)\right]$$

Then the final loss is defined as:

$$\mathcal{L}_{\text{PPO}} = \mathcal{L}_{\text{actor}} + \frac{1}{2}\mathcal{L}_{\text{critic}} + 0.01\mathcal{L}_{\text{entropy}} \tag{8}$$

Our model discovers new high-scoring ligands through 2 distinct mechanisms:

- While training the model, we store the top-100 best hits ever encountered. This first approach produces a set of high-quality candidates (however at the cost of diversity).
- Once the model is trained, we can also sample random latent vectors in the generator latent space, and improve them using the trained actor (generating diverse candidates, with usually lower scores than the previous method).

Algorithm 1 PPO for latent molecular optimization

Input: Actor parameters θ, Critic parameters ϕ, Reward scheme, n_{epochs}, n_{PPO}, $n_{\text{batch size}}$, $T = 1$, $\epsilon = 0.2$, $\gamma = 0.95$
for $t = 1$ **to** n_{epochs} **do**
  Sample $n_{\text{batch size}}$ random latent vectors $z \sim \mathcal{N}(0, I)$
  Compute policy parameters $(\mu_\theta(z_t), \sigma_\theta(z_t))$
  Sample actions $a \sim \mathcal{N}(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2)$
  Update latent vectors $z' = z + a$
  Compute associated rewards and critic estimates
  Compute Advantages A_t
  for $j = 1$ **to** n_{epochs} **do**
    Take new set of actions a'
    Compute corresponding new states z''
    Compute likelihood ratios $r_t(\theta)$
    Compute average $\mathcal{L}_{\text{actor}}$
    Compute new critic estimates $V_\phi(z)$
    Compute average $\mathcal{L}_{\text{critic}}$
    Compute average $\mathcal{L}_{\text{entropy}}$
    Compute $\mathcal{L}_{\text{total}}$ and update θ and ϕ parameters.
  end for
end for

B. REINFORCE Optimization

Here, we provide some more details on our implementation of the REINFORCE algorithm (see Algorithm 2).

Algorithm 2 REINFORCE for latent molecular optimization

Input: Actor parameters θ, Reward scheme, n_{epochs}, $n_{\text{batch size}}$
for $t = 1$ **to** n_{epochs} **do**
  Sample $n_{\text{batch size}}$ random latent vectors $z \sim \mathcal{N}(0, I)$
  Take actions $a \sim \pi_\theta = \mathcal{N}\left(\mu_\theta(z), \text{Diag}(\sigma_\theta(z))^2\right)$
  Compute the corresponding rewards
  Compute average loss $-\mathbb{E}_{z\sim\pi_\theta}(\text{Reward} \cdot \log(\pi_\theta(z))$
  Update actor parameters θ.
end for