Targeted Molecular Generation with Latent Reinforcement Learning

Ragy Haddad[1] *, Eleni E. Litsa[1], Zhen Liu[1†], Xin Yu[2], Daniel Burkhardt[1], and Govinda Bhisetti[1]

[1]Cellarity, Inc
[†]Carnegie Mellon University
[2]NVIDIA
[*]Corresponding author: rhaddad@cellarity.com

SUPPLEMENTARY INFORMATION

S 1: VAE

S1.1 VAE Training

To mitigate posterior collapse, we adopted the following modifications to the standard VAE: β-VAE : The β-VAE has a modified loss function with respect to the original VAE model. A multiplier (beta term) is introduced in the part of the loss function that corresponds to the KL divergence. The beta term contributes to achieving a trade-off between the reconstruction loss and the latent space representation which in turn affects the validity rate.

- Cyclical annealing schedule of the beta term [36]: This is an improvement over the Beta-VAE where the beta term alternates between phases of gradual increase and phases of gradual decrease. The gradual periodic increase of the beta term prevents the vanishing of the KL divergence term.

- Max beta term [18] In this variation of the cyclical annealing beta-VAE, the beta term is limited to a maximum value of 0.6 to further assist finding a trade-off between the reconstruction loss and the validity rate.

An additional training modification we adopted is a word dropout mechanism where randomly selected tokens are masked with a random probability during training time [15]. This serves as a form of regularization during the VAE training and helps with preventing the decoder from overfitting as well as introducing diversity and novelty in the generated sequences.

S1.2 VAE Architecture

Encoder: Gated Recurrent Unit (GRU) Decoder: GRU Trained with word dropout and cyclical annealing.

Hyper-parameter	**Value**
General Hyper-parameters	
Learning rate	0.0001
Latent space size	64
Word dropout	0.3
Encoder - GRU	
Number of layers	2
Bidirectional	True
Dropout probability	0.3
Decoder - GRU	
Number of layers	4
Bidirectional	False
Dropout probability	0.3

Table S1. Hyper-parameters and their values for the model.

S1.3 Evaluating cyclical annealing on VAE pretraining

Model	**Loss**	**Encoder**	**Decoder**	**Annealing**	**Validity**	**Reconstruction**
CNN2RNN	VAE	CNN	RNN	None	0.168	0.860
RNN2RNN	VAE	RNN	RNN	None	0.94	0.00007
RNN2RNN	VAE	RNN	RNN	Logistic	0.97	0.0001
RNN2RNN	VAE	RNN	RNN	Cyclical	0.94	0.763
MolMIM	MIM	Transformer	Transformer	-	0.97	0.870

Table S2. Comparison of different models with their respective configurations and performance metrics.

S 2: Policy Network Architecture

The shared network is a 3-layer fully connected network ReLU activation and batch normalization at each layer, it encodes the action dimension into a hidden dimension that is later fed forward into both the actor network and critic network, in the input dimension corresponds to the dimension of the latent vector input and the output dimension corresponds to the action dimension. The actor network is a single layer fully connected network with a Tanh activation, the critic network is a 2-layer fully connected network with ReLU activation.

We have experimented with multiple architectures including not applying a shared network but found more consistent performance with this architecture across tasks.

S 3: Ablation Study: PPO VS REINFORCE

In this section, we compare the PPO algorithm with the traditional REINFORCE algorithm on the task of generating molecules that contain at least one sulphur atom. For this task, we draw inspiration from the work introduced by Olivecrona et al.[1] where the authors presented an analysis on the task of generating molecules with no sulphur atoms. Herein, we inverse the original problem to generate molecules that do contain sulphur atoms which we found to be a more challenging problem. We use this task as an informative baseline for the following: 1) To evaluate the efficiency of PPO over standard REINFORCE, 2) to validate the initial ability of RL for molecular optimization in structured latent space of two independent models on a simple task.

Figure S1 shows the evolution of the reward as training progresses when using the PPO algorithm and when using the REINFORCE algorithm for the two latent spaces, VAE and MolMIM. The reward is computed as the proportion of generated molecules containing at least one sulphur atom. What is recorded is the average reward of the top-K molecules with the highest reward as a function of the number of oracle calls for up to 500 calls. In the MolMIM space, both PPO and REINFORCE converge to the same reward value, however, PPO reaches convergence faster. In the VAE space, the REINFORCE algorithm falls behind the PPO in terms of convergence reaching a lower reward value within the limit of 500 calls.

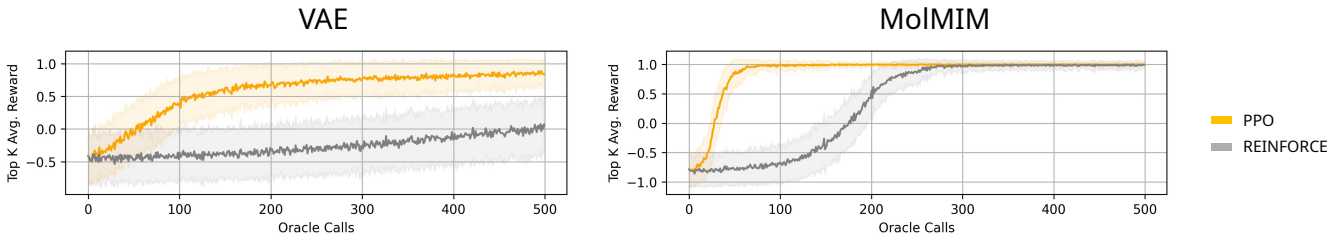

Figure S1. Top-K average reward as a function of the number of oracle calls during training when optimizing using the PPO and when using the REINFORCE algorithm in the two latent spaces.

To further evaluate the efficiency of the two methods, we computed the area under the curve (AUC) of the mean reward of generated molecules over the number of oracle calls at n = 128 intervals , a metric introduced by Gao et al. for evaluating efficiency of molecular optimization methods[2]. This analysis offers a quantifiable metric for comparing PPO with REINFORCE in terms of data efficiency in the context of molecular optimization. We run the RL training 5 times with a fixed starting seed vector and the same standard deviation σ and decay factor γ in both spaces for a fair comparison of the methods.

Method	**Model**	**Top-K AUC (n=128)**
REINFORCE	VAE	0.32 ± 0.06
REINFORCE	MolMIM	0.65 ± 0.04
PPO	VAE	0.77 ± 0.07
PPO	MolMIM	0.93 ± 0.01

Table S3. Top-K AUC (k=128) results for different methods and models.

Table S3 shows the average value and standard deviation over the 5 runs when using REINFORCE and PPO in the two spaces. The PPO algorithm achieved a significantly higher AUC value in the latent space of both, VAE and MolMIM and therefore is more efficient for the task of generating molecules with sulphur atoms.

S 4: Number of carbons correlated to pLogP

To examine the correlation between the number of carbon atoms in a molecule and the pLogP value, we calculated Pearson's R for a random sample of 1000 molecules from the ZINC library which was found to be equal to 0.49 (p-value=0.0).

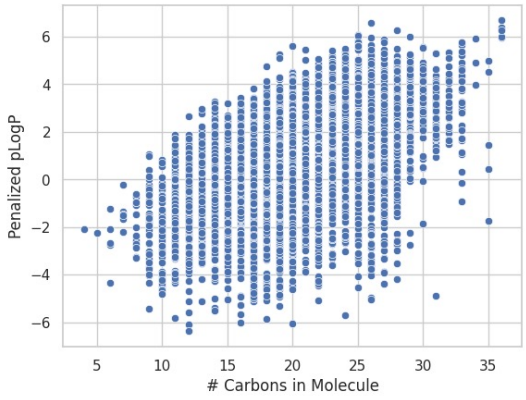

Figure S2. PLogP value and number of carbon atoms for 1K molecules from ZINC

S 5: Optimization for GSK3β and JNK3

Here we show the performance of PPO for optimizing molecules for biological activity for GSK3β and JNK3 individually as well as for dual binding towards the two targets. The reward functions for the single binding experiments correspond to the predicted activity for each target respectively and for the dual binding, the reward function is the summation of the activities towards the individual targets.

The following table shows the performance of the PPO method in the VAE space and in the MolMIM space for generating molecules for the desired targets as well as the performance of state-of-the-art molecule optimization methods as reported in the literature[3].

Method	**GSK3β**			**JNK3**			**GSK3β + JNK3**		
	Success	**Novelty**	**Diversity**	**Success**	**Novelty**	**Diversity**	**Success**	**Novelty**	**Diversity**
JT-VAE	32.2%	11.8%	0.901	23.5%	2.9%	0.882	3.3%	7.9%	0.883
GCPN	42.4%	11.6%	0.904	32.3%	4.4%	0.884	3.5%	8.0%	0.874
REINVENT	99.3%	61.0%	0.733	57.7%	62.6%	0.729	97.4%	39.7%	0.595
RationaleRL	100%	53.4%	0.888	98.5%	31.6%	0.862	100%	97.3%	0.824
PPO (VAE-CYC)	96.7%	71.64%	0.825	82.3%	75.1%	0.765	41.4%	76.3%	0.755
PPO (MolMIM)	100%	57.58%	0.745	100%	57.58%	0.572	100%	65.3%	0.397

Table S4. Comparison of different methods for bio-activity objectives. Results from PPO are reported as the best 5000 molecules during optimization

S 6: Effect of Clipping Epsilon on Optimization

For scaffold-guided optimization we employed a benchmark addressing the effects of variable clipping thresholds of the PPO loss and observed the training dynamics of the agent.

We observe more clipping corresponds to faster convergence of the PPO objective

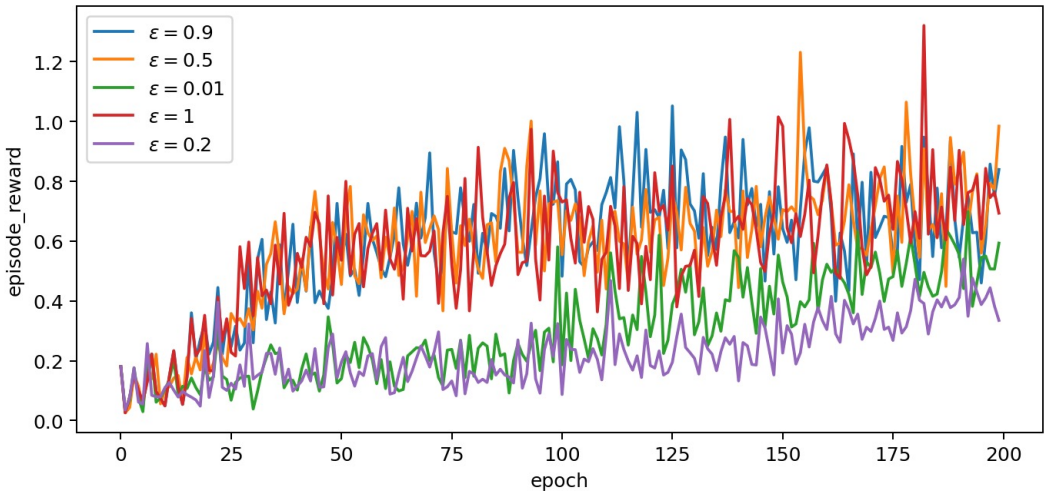

Figure S3. Effetcs of varrying clipping epsilon on training

Supplementary References

1. M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen, "Molecular de-novo design through deep reinforcement learning," *Journal of Cheminformatics*, vol. 9, p. 48, Sept. 2017.
2. W. Gao, T. Fu, J. Sun, and C. W. Coley, "Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization," Oct. 2022. arXiv:2206.12411.
3. W. Jin, R. Barzilay, and T. Jaakkola, "Multi-Objective Molecule Generation using Interpretable Substructures," July 2020. arXiv:2002.03244 [cs, stat].