## A TOY EXAMPLE IN SEC 3.1

We compare the error on $\nabla_{\pi_\theta(x_t,t)} Q_\phi(x_t, \pi_\theta(x_t,t),t)$. Let $\phi'$ denote the critic trained by Eq. 4 on $n$ trajectories and $\phi''$ denote the trained by Eq. 9 on $D_0$ with $|D_0| = n$. We evaluate $|\nabla_{\pi_\theta(x_t,t)} Q_{\phi'}(x_t, \pi_\theta(x_t,t),t) - \nabla_{\pi_\theta(x_t,t)} Q_{\phi'^*}(x_t, \pi_\theta(x_t,t),t)|$ and $|\nabla_{\pi_\theta(x_t,t)} Q_{\phi''}(x_t, \pi_\theta(x_t,t),t) - \nabla_{\pi_\theta(x_t,t)} Q_{\phi''^*}(x_t, \pi_\theta(x_t,t),t)|$ where $\phi'^*$ and $\phi''^*$ are trained on $10^5$ trajectories.

The architecture of the critic network is 3 layered-MLP with hidden dimemsion 256. We train each network for $10^5$ iteration with batchsize 256. And A For reward function, we use Rastrigin function which is a toy function, with many local minimas, designed for testing zero order optimization algorithm.

## B IMPLEMENTATION DETAILS

In this section, we will describe the implementation of experiments in detail.

Guan et al. (2023) employed an SE(3)-equivariant diffusion model, named TargetDiff, for structure-based drug design. Given a protein binding site, TargetDiff generates the atom coordinates in 3D Euclidean space and atom types by iteratively denoising from a prior distribution. After the reverse (generative) process of the diffusion model, the chemical bonds of the generated ligand molecules are defined as post-processing by OpenBabel (O'Boyle et al., 2011) according to the distances and types of atom pairs. We use TargetDiff as the actor and strictly follows the setting in Guan et al. (2023), such as noise schedules, model architecture, training objectives, etc.

We first pretrain TargetDiff on the training set. After that, we use the pretrained TargetDiff to first sample 100 ligand molecules for each pocket in the test set and evaluate their binding affinity by oracle. We pretrain the critic, which predicts the binding affinity based on the perturbed samples, on the $10,000$ generated pocket-ligand pair data. The model architecture of the critic is almost the same with TargetDiff. The only difference is that the critic has an aggregation layer at last to output a scalar based on global features. We finetune the pretrained TargetDiff (Guan et al., 2023) for 30 iterations for each pocket in the test set, respectively. In each iteration, we sample 34 ligand molecules induced by the diffusion model (i.e., the actor), evaluate the binding affinity by oracle, and then online update the diffusion model (i.e., the actor) and train the policy and critic following Alg. 2. We keep all sampled molecules in $D_0$ which falls into the class of off-policy policy gradient.

We use Adam (Kingma & Ba, 2014) with `init_learning_rate=0.001, betas=(0.95, 0.999), batch_size=8 and clip_gradient_norm=8.0` to pretrain TargetDiff (i.e., the actor) and the critic. We use Adam with `init_learning_rate=0.0003` for online updating the actor and critic. As for regularization in Eq. 14, we set $\eta_2 = 0.05$ for atom types and $\eta_2 = 0.00025$ for atom positions.

## C OPTIMIZATION CURVES OF 100 PROTEIN POCKETS

We plot the optimization curves of DiffAC for the pocket protein in the test separately in Fig. 5, 6, and 7. Given a pocket protein, at each iteration, the average Vina Score of the sampled ligand molecules in this iteration is plotted as a point in the figure. The curves show how the binding affinity change with the number of optimization iterations. Generally, in most cases, DiffAC performs better than the baselines.

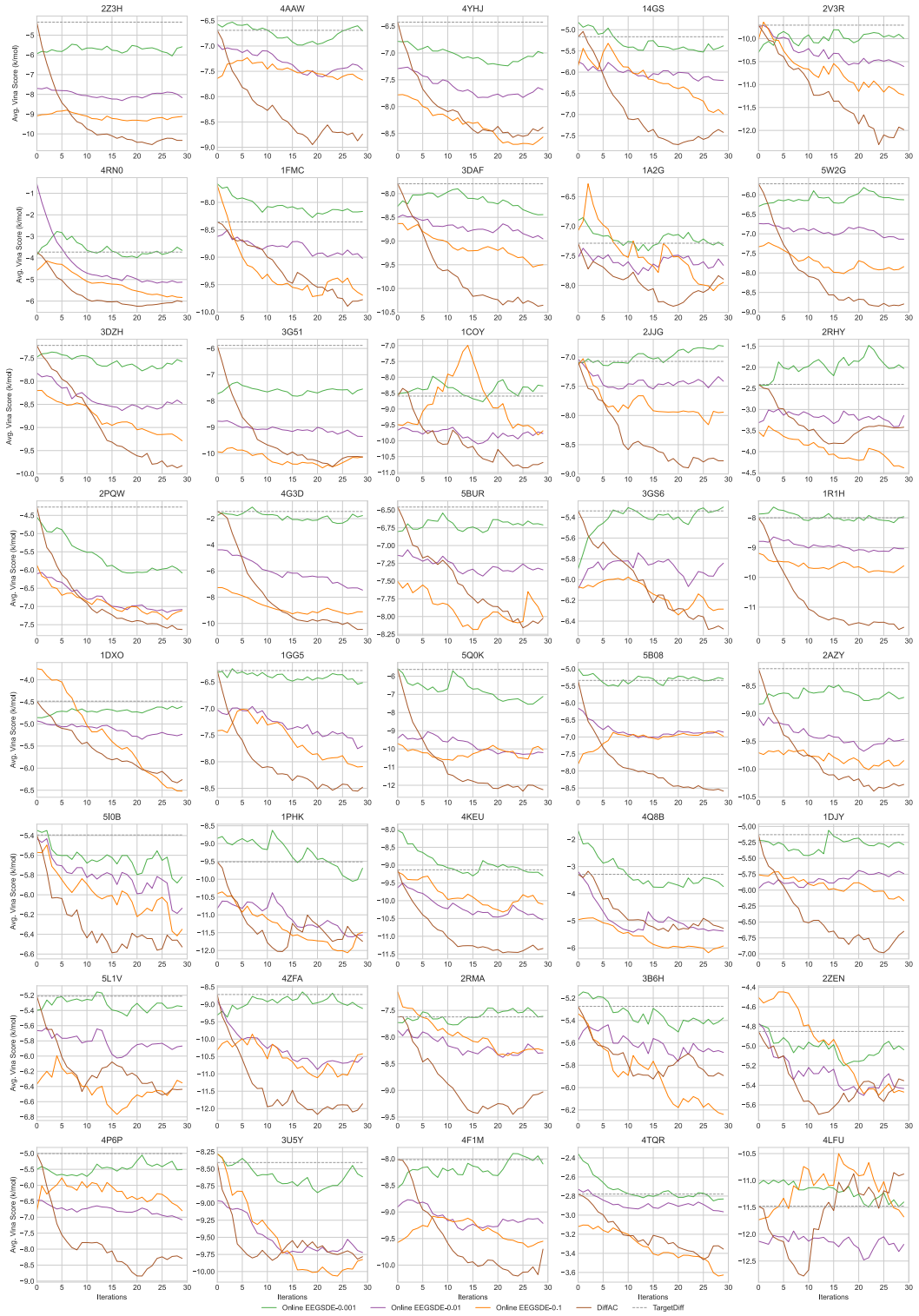


Figure 5: Optimization curves of the 1st to 40th protein pockets in the test set.

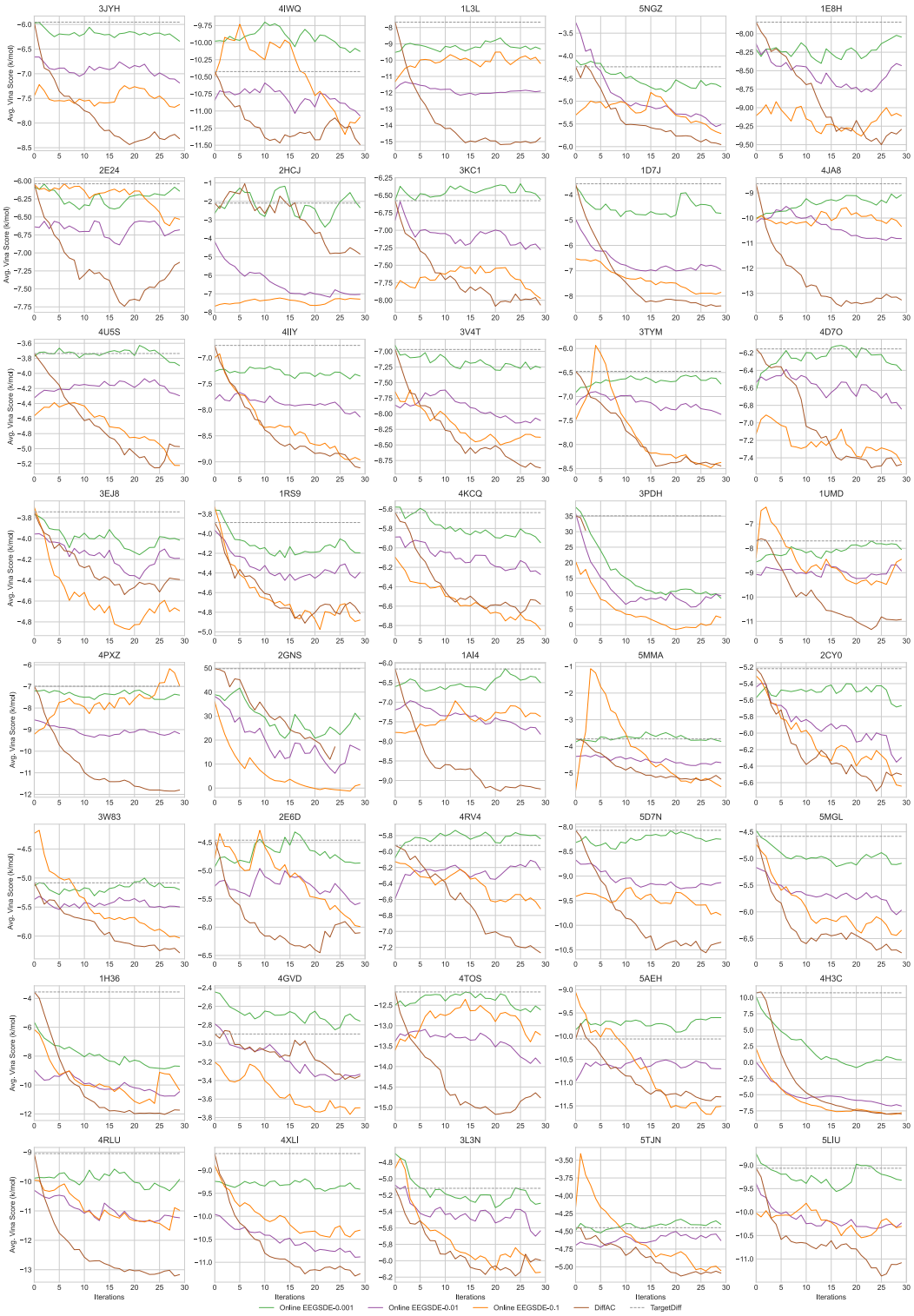


Figure 6: Optimization curves of the 41st to 80th protein pockets in the test set.

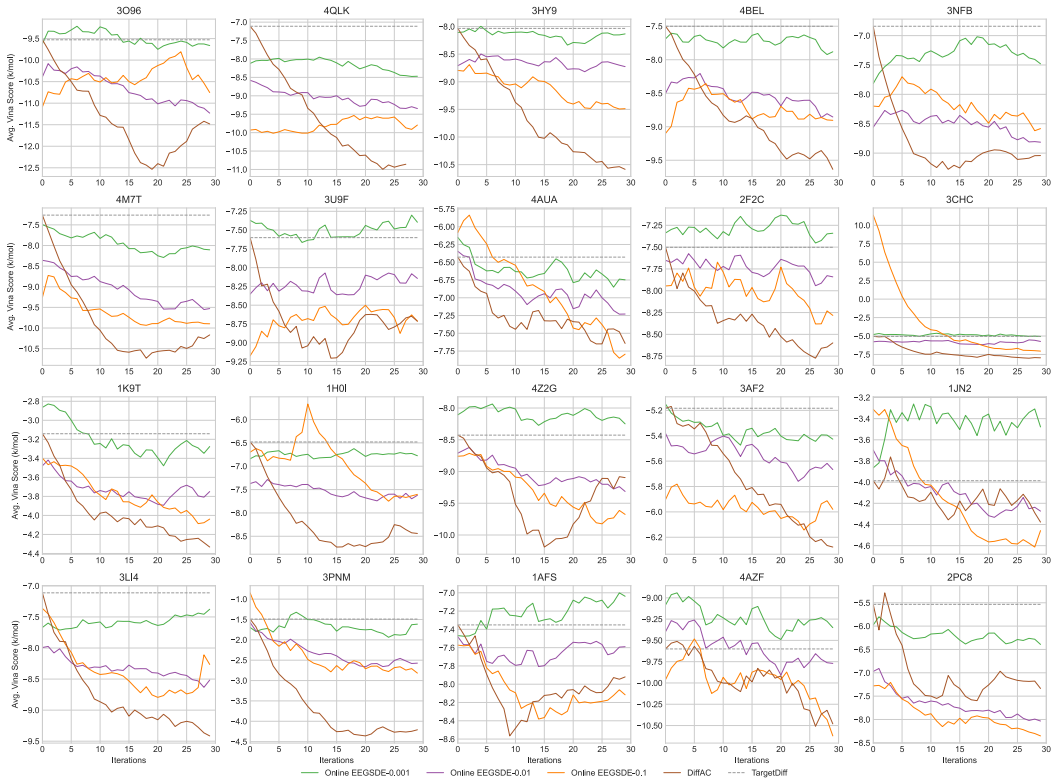


Figure 7: Optimization curves of the 81st to 100th protein pockets in the test set.

# D EXPERIMENTS ON TEXT-TO-IMAGE GENERATION

To demonstrate the generalizability of our method beyond the SBDD task, we also apply our method to text-to-image generation. In this experiment, we use DiffAC to fine-tune text-to-image generative models to better align with human preferences.

## D.1 EXPERIMENTAL SETUP

We use Stable Diffusion v1.5 (Rombach et al., 2022) as the baseline, which has been pre-trained on large image-text datasets (Schuhmann et al., 2021; 2022). For compute-efficient fine-tuning, we use Low-Rank Adaption (LoRA) (Hu et al., 2022), which freezes the parameters of the pre-trained model and introduces low-rank trainable weights. We apply LoRA to the UNet (Ronneberger et al., 2015) module and only update the added weights. For the reward model, we use ImageReward (Xu et al., 2023) which is trained on a large dataset comprised of human assessments of images. Compared to other scoring functions such as CLIP (Radford et al., 2021) or BLIP (Li et al., 2022), ImageReward has a better correlation with human judgments, making it the preferred choice for fine-tuning our baseline diffusion model. In practice, we use DiffAC (the REINFORCE version, i.e., Eq. 11) to fine-tune Stable Diffusion.

We also compare our method with DPOK (Fan et al., 2023). DPOK is a strong baseline that updates the pre-trained text-to-image diffusion models using policy gradient with KL regularization to maximize the reward. Notably, the difference between DPOK and our method is that DPOK estimates policy gradient with real trajectories sampled by backward process while our method estimates policy gradient with efficient forward process. And this difference is the key factor for stabler policy gradient.

We adopt a straightforward setup that uses one text prompt "A green colored rabbit" during fine-tuning and compares ImageReward scores of all methods. For both DPOK and our method, we perform 5 gradient steps per sampling step. The sampling batch size is 10 and the training batch size is 32.

## D.2 EXPERIMENTAL RESULTS

We plot the optimization curves of all methods as shown in Fig. 8. As the results indicates, our method can efficiently improve ImageReward scores and outperform baselines by a large margin.

We provide image examples as shown in Fig. 9. Stable Diffusion tends to generate images with obvious mistakes like generating a rabbit with a green background given the prompt "A green colored rabbit", while our method generates much more satisfying images that are well aligned with the given text prompt. The experiments on text-to-image generation along with structure-based drug design demonstrate the generalizability of our method and reveal its great potential on many real-world applications.

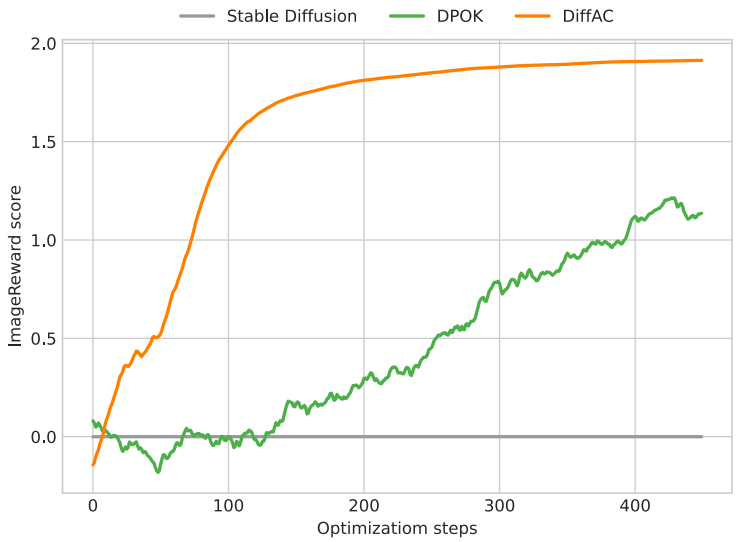


Figure 8: Optimization curves of ImageReward scores.


Figure 9: Example images generated by Stable Diffusion (Rombach et al., 2022) (top row) and our method (bottom row) givne text prompt "A green colored rabbit".