

---

# Noise Search for Inference Time Reward Optimization with Diffusion Models

---

Adam Klein  
aklein4@stanford.edu

## Abstract

This paper presents a method to steer pre-trained diffusion models into optimizing an objective function, without any tuning of the model’s weights. Rather than looking at the model itself, this method treats the model as a black-box and instead optimizes one of the model’s inputs: the initial noisy image state. By applying stochastic search algorithms to the initial noise, we show that arbitrary objectives can be optimized while maintaining or improving image quality. We demonstrate the capabilities of this method by applying it to three different tasks: human preferences, prompt alignment, and image compressability. Furthermore, we show that objective scores tend to vary smoothly with initial noise interpolation, illustrating the potential more advanced local optimization algorithms operating on such noise. <https://github.com/aklein4/NoiseSearch>

## 1. Introduction

In many down-stream tasks, the goal of diffusion models is not to strictly sample from the data distribution. Instead, we want them to optimize some objective, such as image aesthetics or, in biology domains, protein energy (Delaunay et al., 2022) or drug effectiveness. Typically, models are trained to maximize objectives using reinforcement learning (RL). However, the intractability of diffusion models’ likelihoods makes this difficult (Black et al., 2023).

Some work has seen success by treating the diffusion rollout as a Markov Decision Process (MDP) and applying a policy gradient algorithm, such as Proximal Policy Optimization (PPO) (?). However, this method is slow and requires many samples to be taken from the model.

Other work has looked at back-propagating directly from the objective function into the model, or using classifier-guidance with the objective function (Xu et al., 2023). However, these methods require a differentiable objective function, which can be difficult to obtain.

The method introduced in this paper seeks to remove the need to retrain the model with costly RL, and remove the necessity for the objective function to be differentiable.

## 2. Background

### 2.1. Diffusion Models

Diffusion Probabilistic Models (Ho et al., 2020) have seen great success for the task of image generation, and have become the standard for open-source foundation models in the area. In their basic form, these models parameterize the data distribution  $x_0 \sim q(x_0)$  as a markov process, where each latent representation has the same dimensional as the data:

$$p_\theta(x_0) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad (1)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (2)$$

The initial noisy state, often referred to in this paper as the ‘initial noise’, ‘initial state’ or simple the ‘state’, is represented by a simple Gaussian distribution:

$$p(x_T) = \mathcal{N}(x_{t-T}; 0, \mathbf{I}) \quad (3)$$

To reduce the number of steps required for generation, some extensions modify the sampling procedure to be non-markovian. For example, Denoising Diffusion Implicit Models (DDIM) predict  $x_0$  and move towards it (Song et al., 2022):

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left( \frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t) + \sigma_t \epsilon_t \quad (4)$$

Methods like DDIM have two properties that are important for our application: they are deterministic with respect to  $x_T$ , and by spherically interpolating between different initial noise vectors one can reasonably interpolate between output images (Song et al., 2022).

### 2.2. Fast Diffusion Models

The method used in this paper would not be practical without the recent development of distilled diffusion models or consistency models.

Distilled models use pre-trained diffusion models to teach new diffusion models to skip generation steps. Distillation methods can reduce the number of timesteps required for high-quality images from tens to as few as one or two (Luo et al., 2023).

Consistency models are a new class of diffusion models that are designed to create outputs in a single step. This is done by enforcing consistency between time-steps, meaning that  $p_\theta(x_0|x_a)$  and  $p_\theta(x_0|x_b)$  are trained to give the same output when  $x_a$  and  $x_b$  come from the same markov chain (Song et al., 2023).

By leveraging these models, we can search over many different initial noise states in the same time it would take a classical diffusion model to perform a single generation.

### 2.3. Constrained Generation

The method used in this work is inspired by inference-time algorithms that exist for auto-regressive text generation. For example, in Plug and Play Language - Monte Carlo Tree Search (PPL-MCTS) (Chaffin et al., 2022), the authors apply a tree search algorithm to optimize a discriminator-based objective. Their method does not require retraining the model, and instead focuses on sampling outputs from the model that give higher discriminator scores.

The work in this paper uses a similar approach. However, unlike text generators, image generators operate in a continuous domain. Therefore, rather than combinatorial optimization methods, we must rely on continuous optimization algorithms.

## 3. Objective Functions

In this paper, we evaluate our methods on three separate image-based objective functions to capture a variety of use cases.

### 3.1. Human Preference

Human Feedback Reinforcement Learning HFRL has been very important to the recent success of text generation models. HFRL seeks to generate outputs that are more aligned with human preferences, essentially creating images and text that people like better (Xu et al., 2023). However, research is still ongoing to apply HFRL to image generation. Given the importance and outstanding need for HFRL in image generation, we chose our first objective to be a pre-trained HFRL image evaluation model.

For our HFRL objective, we are using the model introduced in ImageReward (Xu et al., 2023). This model was trained on over a hundred thousand ranked images to assign images a score based on the probability that users will prefer them.

Their results show that ImageReward is highly correlated to user satisfaction.

### 3.2. Prompt Alignment

Prompt alignment evaluates how well the contents match the prompt that the user provided, and is a key metric measured in many image generation benchmarks.

For our prompt alignment objective, we are using the CLIP score (Radford et al., 2021). This metric measures the similarity between the prompt and the image by embedding them each into vectors and taking their normalized dot product.

### 3.3. Image Compressibility

Images that are shared over the internet are often compressed to reduce their file size. Images that have fewer details and less variety can be compressed to smaller sizes, and are therefore more efficient to distribute. This objective is particularly interesting because it is not differentiable, and therefore would not be optimizable using common gradient-based approaches.

We measure image compressibility using the standard JPEG format. Our score is measured by compressing the output image into a JPEG file, and taking the ratio between the uncompressed file size ( $3 * \text{height} * \text{width}$ ) of the image and its compressed size.

## 4. Methods

The key insight of this work is that given a pre-trained diffusion model with a deterministic scheduler, we can treat the diffusion model as part of the objective function while optimizing its inputs.

Consider a pre-trained diffusion model  $f_\theta(x_T, c) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_c} \rightarrow \mathbb{R}^{d_x}$  that takes in an initial noise state  $x_T \in \mathbb{R}^{d_x}$  and a condition  $c \in \mathbb{R}^{d_c}$  and produces an image  $x_o \in \mathbb{R}^{d_x}$ . Furthermore, we have an objective function  $R(x_o, c) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_c} \rightarrow \mathbb{R}$  that takes in an image  $x_o \in \mathbb{R}^{d_x}$  and a condition  $c \in \mathbb{R}^{d_c}$  and produces a reward  $r \in \mathbb{R}$ . We can take the composite of our diffusion model and objective function to get a single optimization function with two inputs, the initial noise state and the condition:  $R_\theta(x_o, c) : \mathbb{R}^{d_x} \times \mathbb{R}^{d_c} \rightarrow \mathbb{R} = R(f_\theta(x_T, c), c)$ .

Now, given that the condition and reward function are selected by the user, and our model is fixed, our goal is to find the initial noise state that maximizes the composite objective function.

$$x_T^* = \arg \max_{x_T} R_\theta(x_T, c) \quad (5)$$

This is a classic continuous optimization problem. There-

fore, to solve it, we can apply common hill climbing-based continuous optimization algorithms such as Simulated Annealing (SA) and Stochastic Hill Climbing (SHC).

#### 4.1. Neighbor Selection

A challenge of applying hill climbing algorithms to this problem is that they require a neighbor selection function. This means that given a current state  $x_T$ , we must be able to sample  $x'_T$  that is "close to"  $x_T$ . Ideally, the image produced by  $x'_T$  will be similar to that of  $x_T$ , and they will have similar scores associated with them (S.H. Jacobson, 2004). If the score of  $x'_T$  is independent of  $x_T$ 's score, then the local optimization of hill climbing algorithms provides no benefit over random sampling.

For our neighborhood function, we leverage the interpolation properties shown by DDIM. By spherically interpolating (Shoemaker, 1985) between  $x_T$  and a randomly sampled  $\hat{x}_T$ , we can generate images that are more or less like  $x_T$  depending on the interpolation strength. Furthermore, image quality remains high at each step of the interpolation, since each intermediate  $x'_T$  lies within the prior  $p(x_T) \sim \mathcal{N}(0, \mathbf{I})$  distribution. See section 6.1 for further analysis.

#### 4.2. Random Sampling

The most simple search algorithm to optimize the objective is to sample  $n$  random independent  $x_T^{(i)}$ , then take the one that gives the highest score.

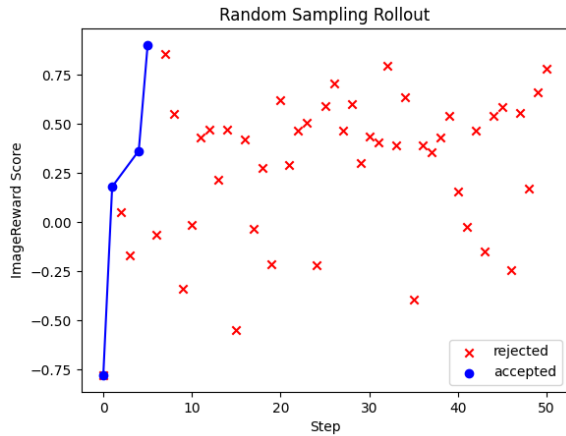


Figure 1. Example of a Random Sampling search rollout for the prompt "a modern, contemporary studio apartment interior". We see that each state is sampled independently. Unlike the other search algorithms higher scoring states do not become more frequent.

As seen in Figure 1, this method does not use local optimization or take advantage of our neighborhood function.

#### Algorithm 1 Random Sampling Algorithm

---

**Input:** model  $f_\theta$ , objective  $R$ , condition  $c$   
 $\mathcal{X} = \{\}$   
**for**  $i = 1$  **to**  $n$  **do**  
 $x_T^{(i)} = \mathcal{N}(\mathbf{0}, \mathbf{I})$   
 $\mathcal{X} \leftarrow f_\theta(x_T, c)$   
**end for**  
**return**  $\arg \max_{x_0 \in \mathcal{X}} r(x_0, c)$

---

#### 4.3. Stochastic Hill Climbing (SHC)

Stochastic Hill Climbing (SHC) takes advantage of the neighborhood function by sampling a neighbor at each step, then greedily accepting the change if and only if the new state leads to a higher score (Juels & Wattenberg, 1995). This means that SHC iteratively improves the state, and samples higher scoring states more frequently as it progresses, as shown in Figure 2.

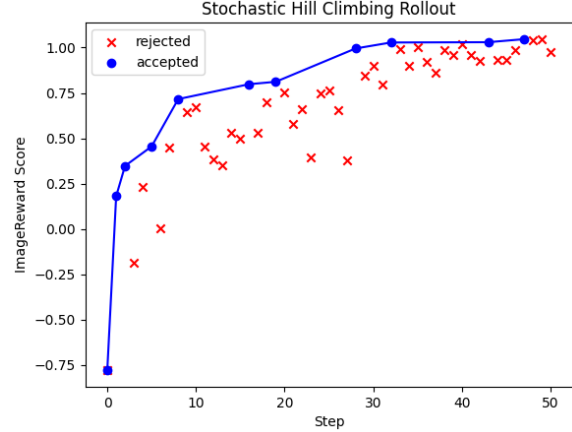


Figure 2. Example of a Stochastic Hill Climbing search rollout for the prompt "a modern, contemporary studio apartment interior". We see that sampled states get closer to the current state as  $m$  is annealed, and that the score iteratively improves.

The interpolation value  $m$  in our neighborhood function is a hyper-parameter of this algorithm. Too large, and the samples are too independent for iterative improvement. Too small, and progress is very slow. To balance exploration and exploitation, we anneal  $m$  from a high value at the beginning of the search to a low value at the end.

#### 4.4. Simulated Annealing (SA)

Simulated Annealing (SA) makes the acceptance criteria of SHC stochastic, to increase exploration. Since SHC ascends the score function greedily, it often gets stuck in local maxima. SA avoids this by sometimes taking sub-optimal moves in order to escape the maxima (S.H. Jacobson, 2004), as demonstrated in Figure 3.

---

**Algorithm 2** Stochastic Hill Climbing Algorithm

---

**Input:** model  $f_\theta$ , objective  $R$ , condition  $c$ , starting interpolation value  $m_0$ , interpolation decay value  $\gamma_m$

$x_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$   
 $x_0 = f_\theta(x_T, c)$   
 $s = R(x_0, c)$   
 $m = m_0$

**for**  $i = 1$  **to**  $n$  **do**  
   $\hat{x}_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$   
   $x'_T = \text{slerp}(x_T, \hat{x}_T, m)$   
   $x'_0 = f_\theta(x'_T, c)$   
   $s' = R(x'_0, c)$   
  **if**  $s' > s$  **then**  
     $x_T, x_0, s \leftarrow x'_T, x'_0, s'$   
  **end if**  
   $m \leftarrow m * \gamma_m$   
**end for**  
**return**  $x_0$

---

SA is parameterized by a temperature  $\tau$  that controls how likely a sub-optimal move is to be accepted. Initially  $\tau$  is high, so moves are likely to be accepted regardless of score. As the search progresses,  $\tau$  lowers until only improvements are accepted, mimicking SHC.

---

**Algorithm 3** Simulated Annealing

---

**Input:** model  $f_\theta$ , objective  $R$ , condition  $c$ , starting interpolation value  $m_0$ , interpolation decay value  $\gamma_m$ , starting temperature  $\tau_0$ , temperature decay value  $\gamma_\tau$

$x_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$   
 $x_0 = f_\theta(x_T, c)$   
 $s = R(x_0, c)$   
 $m, \tau = m_0, \tau_0$

**for**  $i = 1$  **to**  $n$  **do**  
   $\hat{x}_T = \mathcal{N}(\mathbf{0}, \mathbf{I})$   
   $x'_T = \text{slerp}(x_T, \hat{x}_T, m)$   
   $x'_0 = f_\theta(x'_T, c)$   
   $s' = R(x'_0, c)$   
  **if**  $(s' > s) \vee (\mathbf{y} \sim \text{Uniform}(0, 1) < \exp[\frac{s'-s}{\tau}])$  **then**  
     $x_T, x_0, s \leftarrow x'_T, x'_0, s'$   
  **end if**  
   $m, \tau \leftarrow \gamma_m m, \gamma_\tau \tau$   
**end for**  
**return**  $x_0$

---

## 5. Evaluation

We evaluate each search method against the single-shot performance of the model, as well as against other popular image generation models when metrics are available.

The model that we used for evaluation is the Stable Diffusion Turbo model (Luo et al., 2023). This is a distilled

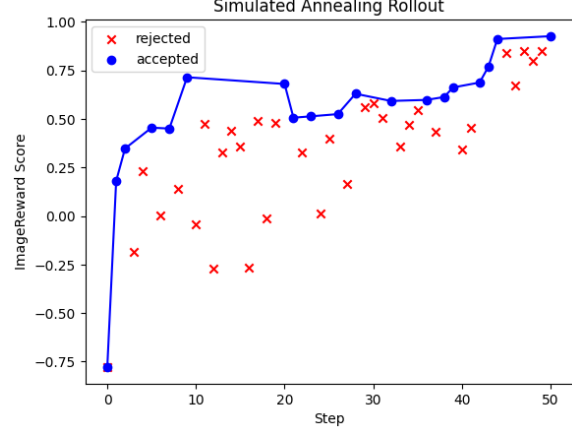


Figure 3. Example of a Simulated Annealing search rollout for the prompt “a modern, contemporary studio apartment interior”. We see that sampled states get closer to the current state as  $m$  is annealed, and that acceptance becomes more greedy as  $\tau$  is annealed.

latent diffusion model that can generate 512 by 512 pixel images in 1-4 steps. We chose this model due to its speed, as well as its high quality. Furthermore, we noticed that this model gives more diverse images for a given prompt than the larger version Stable Diffusion Turbo XL, which makes our search method more powerful. We leave the task of benchmarking the performance of our search algorithms on other models to future work.

See section 9.1 in the appendix for hyper-parameters used during evaluation. The number of search steps and inference steps per generation were selected such that a complete single search with Stable Diffusion Turbo takes a comparable amount of time to a single sample from a larger image generator.

### 5.1. Benchmarks

Model	ImageReward	CLIP Score	JPEG Ratio
Openjourney	0.2614	0.2726	N/A
Stable Diffusion 2.1-base	0.2458	0.2683	N/A
DALL-E 2	0.2114	0.2684	N/A
Stable Diffusion 1.4	0.1344	0.2763	N/A
Versatile Diffusion	-0.2470	0.2606	N/A
CogView 2	-1.2376	0.2044	N/A
Turbo One-Shot	0.4330	0.2651	13.3470
Turbo Random Sampling	1.2638	0.3025	23.5013
Turbo SHC	<b>1.4033</b>	<b>0.3126</b>	24.4096
Turbo SA	1.3808	0.3074	<b>24.5515</b>

Table 1. Comparison of Stable Diffusion Turbo with various search algorithms against the one-shot performance of other open-source models.

To benchmark the performance of our model, we used the benchmark set of 100 prompts provided by ImageReward

(Xu et al., 2023). In Table 1, we see the performance of our search algorithms compared to the one-shot performance of other models. The metrics for other models come from ImageReward. Note that the original paper averaged performance over 1000 images for each prompt, while we only used 1 image per prompt.

Firstly, we see that the ImageReward score of Stable Diffusion Turbo with a single shot is the best out of the one-shot models. Looking at the examples in the appendix, this fits with the Turbo model seemingly giving higher quality images than Stable Diffusion 2.1 base (Rombach et al., 2022).

More importantly, we see that the search algorithms beat the one-shot models across the board. This is to be expected given that we are specifically selecting for the benchmarked metrics. The important thing is that the local search algorithm (SHC and SA) outperform Random Sampling across all of the objectives. This demonstrates that our more advanced methods actually work, and that local search algorithms are applicable to this problem. These results show that image-based objective functions can be optimized by applying continuous optimization methods to the initial noisy state of a pre-trained diffusion model.

## 5.2. Qualitative Assessment

See section 9.2 in the appendix for examples of images generated by different algorithms and objective functions. We also compare results to those generated by Stable Diffusion 2.1 base (Rombach et al., 2022). However, note that the results from that model seem unusually low quality, possibly due to an error.

For the ImageReward objective, the search algorithms seem to give higher quality images that are better aligned with the given prompt. For example, with the "...old sea captain..." prompt, the search algorithms gave the character a hat, while one-shot generation did not. Others, such as the car and planet examples, showed clear quality improvement. There were not any examples where search made the quality worse. For some examples, the search images look similar to that of the single-shot, even though they have higher scores. This could show a shortcoming of the ImageReward model.

Results for the CLIP score objective are less clear. Some examples, such as the captain and car examples mentioned above, seem to show better alignment. However, image quality is generally the same between the one-shot images and the search images. This result may show the limitation of the CLIP score once images are generally aligned with the prompt.

The JPEG compression objective shows very interesting results. For many prompts, the search generated images can be compressed to half the size of their one-shot coun-

terparts. However, those images maintain the same, if not higher, quality than one-shot images. Since the JPEG compression objective is non-differentiable, this could not be achieved by doing most fine-tuning methods.

For the CLIP Score and JPEG compression objectives, there is little difference between the image quality of the Random Sampling algorithm and the local optimization algorithms. For the ImageReward objective, SHC gives marginally better results than the other algorithms in the eyes of the author. It may be the case that a bigger difference would be seen with a worse pre-trained model or more difficult objective functions, since Random Sampling can already do pretty well for most of the examples.

## 6. Analysis

### 6.1. Objective Function Smoothness

As stated in the methods section, hill climbing algorithms like those used in this paper make the assumption that nearby states have similar scores. In Figure 4, we illustrate how the score changes as you spherically interpolate the initial state in two orthogonal directions. In Figure 5, we show selected images from the interpolation. The method used for two dimensional interpolation is the same as that used for grid interpolation in DDIM (Song et al., 2022).

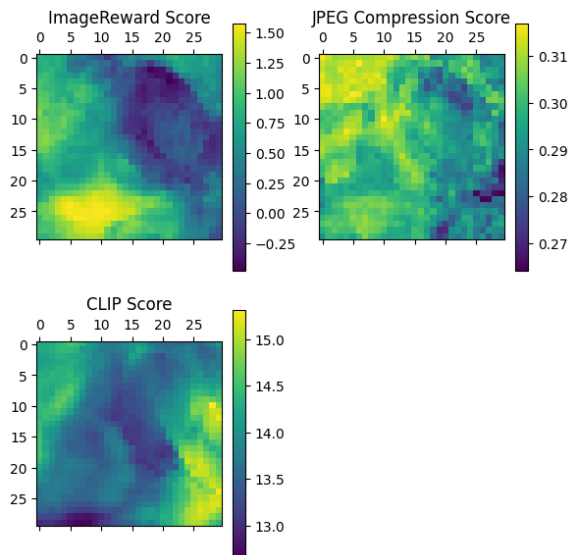


Figure 4. Scores throughout 2 dimensional interpolation of  $x_T$  with the prompt "a concept art of a vehicle, cyberpunk" for each of the studied scoring functions. **Note that the CLIP and JPEG labels should be switched.**

Figure 4 demonstrates that the objective functions are generally smooth with respect to interpolation. Since our neighborhood function is represented by a small random interpolation, this means that neighbors tend to have similar



scores. This corroborates with our benchmark evaluations to show that local optimization works better than random sampling for our setup. Surprisingly, the CLIP score seems to be the most rugged, while it showed the biggest improvement in using local optimization versus random sampling. One possibility is that local search is able to exploit the small peaks of the score function with very small changes.



Figure 5. Two dimensional grid interpolation of  $x_T$  for the prompt "a concept art of a vehicle, cyberpunk"

Figure 5 shows that image interpolation is meaningful for Stable Diffusion Turbo, and that closer images are more similar.

## 6.2. Search Method Comparison

In Figure 6, we see the scores at each step of the search process for each algorithm and for each objective function. While the values are typically similar early in the search, the Random Sampling method tends to plateau before the others. This is especially clear for the CLIP objective, where SHC continues almost linearly while the improvement of the others slows down. While improvement between the local search methods and Random Sampling is not huge, they do present an improvement across every metric, which is notable.

The staircase nature of the random sampling method is interesting: there seemed to have been a coincidence where many of the benchmark prompts found a new best state around the same time. This could be related to the fact that each rollout was initialized with the same random seed. If that is the case, then some objectives like compressability might be almost prompt independent, meaning that the best initial states are almost always the same, re-

gardless of prompt. Further analysis here is left to potential future work.

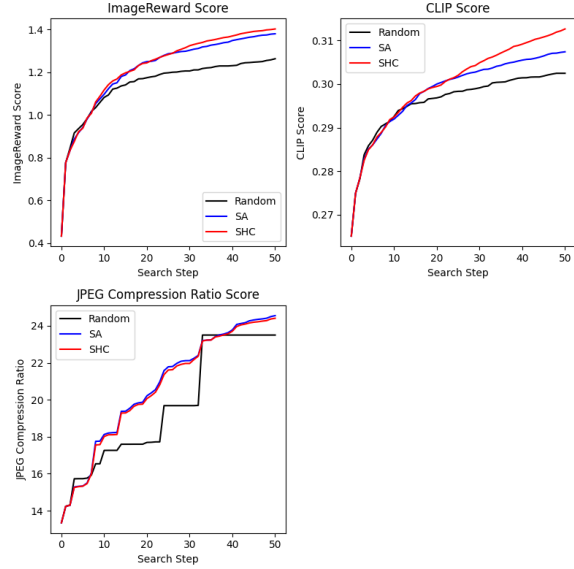


Figure 6. Score throughout search steps for different algorithms and objective functions. Each line represents the mean score at that step from across each the benchmark set.

## 6.3. Limitations

There are several potential shortcomings of this method. First, if an objective function requires samples from outside of the model's distribution, the model likely won't be able to create a good image regardless of  $x_T$ .

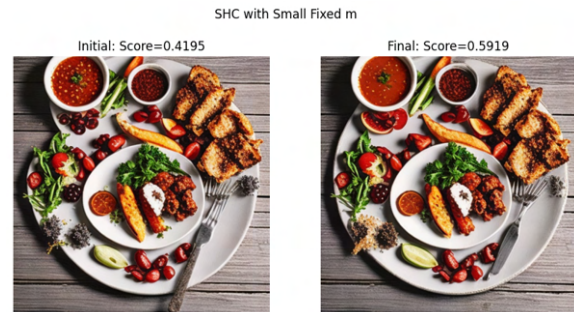


Figure 7. Images generated for the prompt "delicious plate of food". SHC search was performed with a fixed  $m = 0.025$  value.

A shortcoming that I will illustrate here is the potential to maximize the objective function in undesirable ways. Figure 7 shows images before and after SHC was used to optimize the ImageReward objective. In this example a small fixed  $m = 0.025$  neighbor interpolation value was used. We see that while the score increased considerably, the image barely changed. This is because with small  $m$ , SHC acts as a gradient attack on the score function. This means that

the score function can be exploited for high values without actually improving the image.

## 7. Future Work

Since the idea of optimizing  $x_T$  seems to be largely unexplored in the literature, there are many possible avenues for continued work on this concept.

First is the application of reinforcement learning, where an agent could be trained to manipulate  $x_T$  given the current image. This has interesting implications to reasoning in image generation, since such an agent would get the opportunity to iteratively improve its output given previous trials and feedback.

This idea could also apply to reinforcement learning in a different way: using imitation learning (Ross et al., 2011). Since our method can generate images with such high rewards, another one-shot model could be fine-tuned on search-generated images to improve its one-shot capabilities. It’s also possible that repeated iterations of using search to generate high-reward examples and fine-tuning with same model might lead to a model with very good one-shot capabilities.

This method might also be useful for image-to-image applications, where the noise added to the starting image is optimized. Since image-to-image diffusion rollouts tend to use fewer steps, this method would scale better for image-to-image than for text-to-image.

Finally, image generation is likely not the best application of this method. Some structural biology applications use non-differentiable energy scores to evaluate the predictions of generative models (Delaunay et al., 2022). This method could be very useful in those applications, especially when the models are small and many optimization steps can be performed.

## 8. Conclusion

In this work, we introduced a method to steer pre-trained diffusion models into optimizing an objective function, without tuning any of the model’s weights. The method presented uses continuous optimization over the initial noisy state of the diffusion rollout. To the knowledge of the author, this is this idea had not been explored before.

Algorithms demonstrated in this paper included Random Sampling, Stochastic Hill Climbing, and Simulated Annealing. We showed that local optimization is effective for optimizing the initial noisy state. Furthermore, we showed that spherical interpolation can be used to generate neighboring states for stochastic sampling.

The results of this paper show a new state-of-the art (to

the knowledge of the author) for the ImageReward HFRL benchmark. Furthermore, we demonstrated affective optimization of a non-differentiable JPEG image compression objective.

Potential future work relating to this paper includes applications to reinforcement learning, and studying the method in more fitting domains such as biology.

## References

- Black, Kevin, Janner, Michael, Du, Yilun, Kostrikov, Ilya, and Levine, Sergey. Training diffusion models with reinforcement learning, 2023.
- Chaffin, Antoine, Claveau, Vincent, and Kijak, Ewa. Ppl-mcts: Constrained textual generation through discriminator-guided mcts decoding, 2022.
- Delaunay, Antoine P., Fu, Yunguan, Bégué, Alberto, McHardy, Robert, Djermani, Bachir A., Rooney, Michael, Tovchigrechko, Andrey, Copoiu, Liviu, Skwark, Marcin J., Carranza, Nicolas Lopez, Lang, Maren, Beguir, Karim, and Şahin, Uğur. Peptide-mhc structure prediction with mixed residue and atom graph neural network. 2022. doi: 10.1101/2022.11.23.517618.
- Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. Denoising diffusion probabilistic models, 2020.
- Juels, Ari and Wattenberg, Martin. Stochastic hill-climbing as a baseline method for evaluating genetic algorithms. In Touretzky, D., Mozer, M.C., and Hasselmo, M. (eds.), *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL [https://proceedings.neurips.cc/paper\\_files/paper/1995/file/36a1694bce9815b7e38a9dad05ad42e0-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1995/file/36a1694bce9815b7e38a9dad05ad42e0-Paper.pdf).
- Luo, Simian, Tan, Yiqin, Patil, Suraj, Gu, Daniel, von Platen, Patrick, Passos, Apolinário, Huang, Longbo, Li, Jian, and Zhao, Hang. Lcm-lora: A universal stable-diffusion acceleration module, 2023.
- Radford, Alec, Kim, Jong Wook, Hallacy, Chris, Ramesh, Aditya, Goh, Gabriel, Agarwal, Sandhini, Sastry, Girish, Askell, Amanda, Mishkin, Pamela, Clark, Jack, Krueger, Gretchen, and Sutskever, Ilya. Learning transferable visual models from natural language supervision, 2021.
- Rombach, Robin, Blattmann, Andreas, Lorenz, Dominik, Esser, Patrick, and Ommer, Björn. High-resolution image synthesis with latent diffusion models, 2022.
- Ross, Stephane, Gordon, Geoffrey J., and Bagnell, J. Andrew. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.

---

S.H. Jacobson, E. Yucesan. Analyzing the performance of generalized hill climbing algorithms, 2004.

Shoemake, Ken. Animating rotation with quaternion curves. 19(3), 1985. ISSN 0097-8930. doi: 10.1145/325165.325242. URL <https://doi.org/10.1145/325165.325242>.

Song, Jiaming, Meng, Chenlin, and Ermon, Stefano. Denoising diffusion implicit models, 2022.

Song, Yang, Dhariwal, Prafulla, Chen, Mark, and Sutskever, Ilya. Consistency models, 2023.

Xu, Jiazheng, Liu, Xiao, Wu, Yuchen, Tong, Yuxuan, Li, Qinkai, Ding, Ming, Tang, Jie, and Dong, Yuxiao. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

## 9. Appendix

### 9.1. HyperParameters

Algorithm	Search Steps	Inference Steps	$m_0$	$\gamma_m$
Rand Sampling	50	2	N/A	N/A
SHC	50	2	1.0	0.92
SA	50	2	1.0	0.92

Table 2. Hyper-parameters for different search algorithms.

Objective	$\tau_0$	$\gamma_\tau$
ImageReward	0.25	0.92
CLIP	0.025	0.92
JPEG	1.0	0.92

Table 3. Temperature parameters for Simulated Annealing.



## 9.2. Example Images

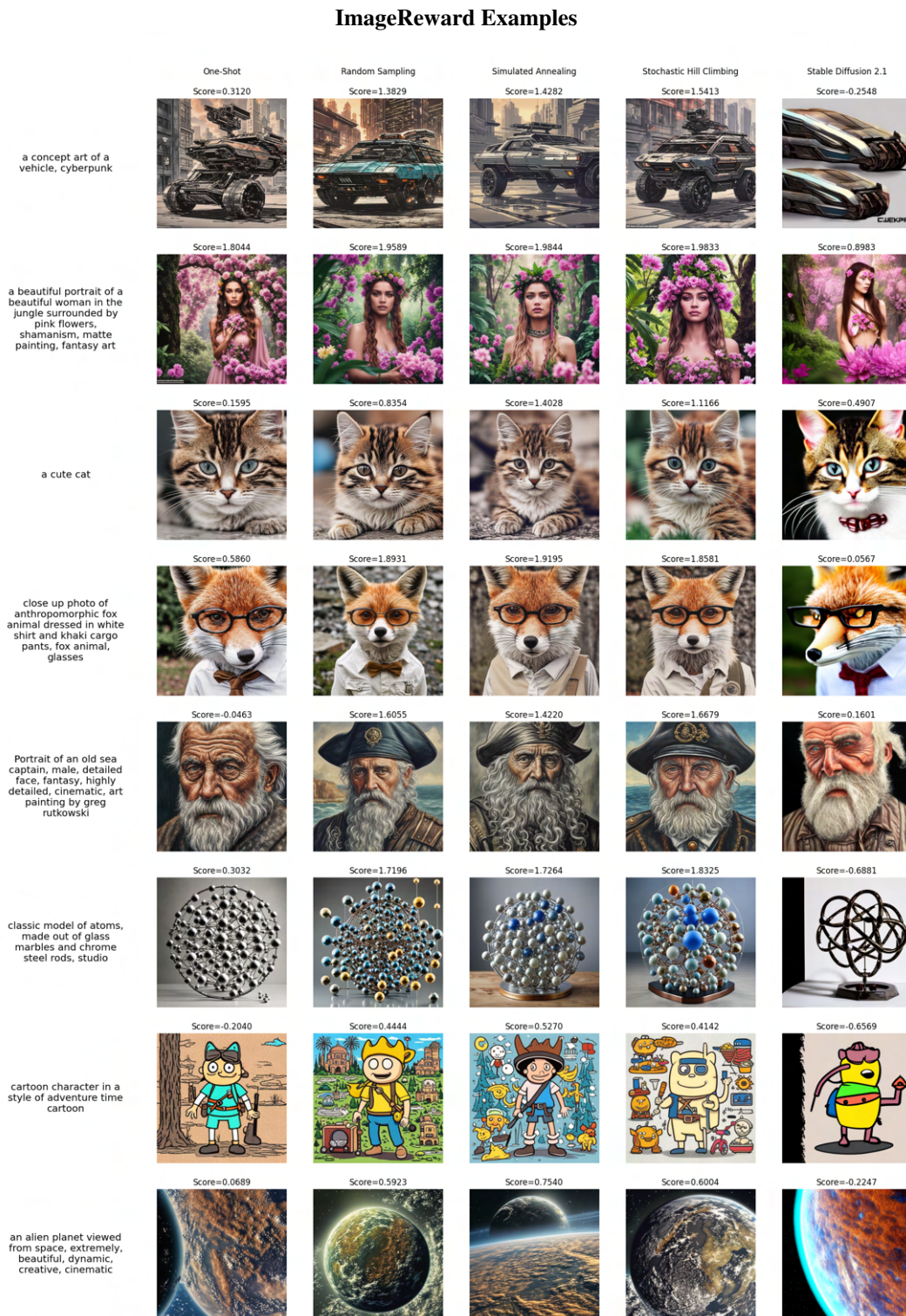


Figure 8. Examples of images generated using the Stable Diffusion Model Turbo model with various search methods and Stable Diffusion 2.1 base. Prompts taken from the ImageReward benchmark set. The objective function for the search rollouts was the ImageReward Score.



## CLIP Score Examples





















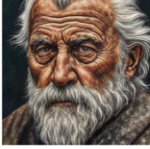



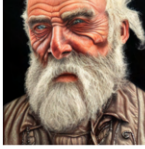

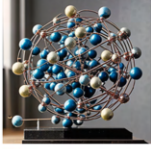

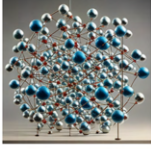









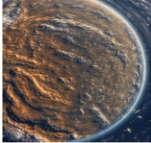
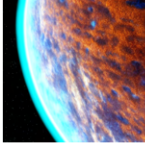
	One-Shot Score=0.3094	Random Sampling Score=0.3129	Simulated Annealing Score=0.3230	Stochastic Hill Climbing Score=0.3323	Stable Diffusion 2.1 Score=0.2797
a concept art of a vehicle, cyberpunk					
a beautiful portrait of a beautiful woman in the jungle surrounded by pink flowers, shamanism, matte painting, fantasy art					
a cute cat					
close up photo of anthropomorphic fox animal dressed in white shirt and khaki cargo pants, fox animal, glasses					
Portrait of an old sea captain, male, detailed face, fantasy, highly detailed, cinematic, art painting by greg rutkowski					
classic model of atoms, made out of glass marbles and chrome steel rods, studio					
cartoon character in a style of adventure time cartoon					
an alien planet viewed from space, extremely beautiful, dynamic, creative, cinematic					

Figure 9. Examples of images generated using the Stable Diffusion Model Turbo model with various search methods and Stable Diffusion 2.1 base. Prompts taken from the ImageReward benchmark set. The objective function for the search rollouts was the CLIP Score.

## JPEG Compression Ratio Examples






















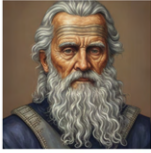


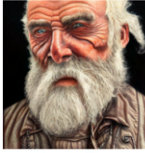














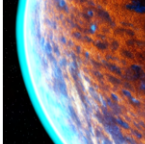
	One-Shot Score=12.4127	Random Sampling Score=19.7432	Simulated Annealing Score=19.5523	Stochastic Hill Climbing Score=19.3046	Stable Diffusion 2.1 Score=15.7649
a concept art of a vehicle, cyberpunk					
a beautiful portrait of a beautiful woman in the jungle surrounded by pink flowers, shamanism, matte painting, fantasy art					
a cute cat					
close up photo of anthropomorphic fox animal dressed in white shirt and khaki cargo pants, fox animal, glasses					
Portrait of an old sea captain, male, detailed face, fantasy, highly detailed, cinematic, art painting by greg rutkowski					
classic model of atoms, made out of glass marbles and chrome steel rods, studio					
cartoon character in a style of adventure time cartoon					
an alien planet viewed from space, extremely, beautiful, dynamic, creative, cinematic					

Figure 10. Examples of images generated using the Stable Diffusion Model Turbo model with various search methods and Stable Diffusion 2.1 base. Prompts taken from the ImageReward benchmark set. The objective function for the search rollouts was the JPEG compression ratio.