ImageBreeder: Guiding Diffusion Models with Evolutionary Computation

Dominik Sobania
Johannes Gutenberg University
Mainz, Germany
dsobania@uni-mainz.de

Martin Briesch
Johannes Gutenberg University
Mainz, Germany
briesch@uni-mainz.de

Franz Rothlauf
Johannes Gutenberg University
Mainz, Germany
rothlauf@uni-mainz.de

Abstract

With the recent advancements of diffusion models, it is quite easy to generate high-quality images. However, many attempts and manual changes are often necessary to achieve this high quality. Leveraging evolutionary algorithms to automate this process therefore presents a promising approach. Consequently, we introduce ImageBreeder as a framework to improve image generation at inference time driven by evolutionary algorithms. Additionally, we study the effectiveness of 10 different variation operators ranging from pixel-based blending techniques to modifications directly on the latent representation of the images. The results show that using evolutionary algorithms can significantly increase an image's quality as well as the alignment with a given prompt. Furthermore, all tested guided search methods are human competitive, as a random trial and error approach is outperformed on over 75% of the benchmark problems. In addition to the empirical results, we also critically discuss the benefits and challenges of our framework uncovering future research directions. We recommend researchers to focus on optimising latent image representations for this purpose as this may improve the ability of evolutionary algorithms to transfer promising features from one image to another, unlocking even better image generation.

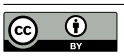
CCS Concepts

• **Computing methodologies** → **Machine learning**; • **Theory of computation** → **Evolutionary algorithms**.

Keywords

Diffusion Model, Evolutionary Algorithm, Image Generation, Variation, Crossover

ACM Reference Format:
Dominik Sobania, Martin Briesch, and Franz Rothlauf. 2025. ImageBreeder: Guiding Diffusion Models with Evolutionary Computation. In *Genetic and Evolutionary Computation Conference (GECCO '25), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3712256.3726439



GECCO '25, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1465-8/2025/07
https://doi.org/10.1145/3712256.3726439

1 Introduction

Diffusion models enable the automated generation of images that meet the specifications of a given prompt [22]. Many of these models are also freely available to the public and can be obtained from platforms such as Hugging Face.[1] Thanks to user-friendly tools such as Automatic1111[2], ComfyUI[3], or Pinokio[4], it is no longer the exclusive domain of programmers and scientists to use diffusion models, but any user interested in new technologies can now explore the details of automatic image generation. Even entire online communities and marketplaces have emerged where settings and prompts for tweaking diffusion models to the best possible output are shared or sold, respectively [16].

However, images are often generated using a demanding trial and error approach. Once the prompt has been defined and the settings are made, images with different seeds are successively generated until one is found that fully meets the user's expectations and is visually appealing. Actually promising image features and styles of the discarded images are often ignored. In addition, this process could also take a considerable amount of time, depending on the model used. Consequently, it is reasonable to replace this manual trial and error approach with a guided search using an evolutionary algorithm (EA). In addition to saving manual work, features of images found during the algorithm's search can possibly be included, which could lead to even better images than possible with the trial and error approach. With recent models like ImageReward [29], which has been trained to automatically evaluate images based on how visually appealing an image is and how well it matches the given prompt, it is also straightforward to define an objective function for an EA. The time-consuming and complicated integration of human participants, which was often required in previous work [2, 10, 23], is no longer necessary. However, how suitable variation operators should be designed in the context of diffusion models and EAs has not yet been sufficiently studied.

To analyse the effectiveness of different variation operators, we introduce ImageBreeder, a lightweight framework for studying image generation driven by an EA. Thanks to the ImageReward model, which we use as objective function, ImageBreeder works fully automatic without the need of human intervention. Overall, in this work we study 10 different variation operators for the EA-driven image generation with diffusion models. These operators range from different pixel-based operators to operators that work directly on the latent representation of an image. For the evaluation, we use a representative sample of prompts from the

[1] https://huggingface.co/
[2] https://github.com/AUTOMATIC1111/stable-diffusion-webui
[3] https://github.com/comfyanonymous/ComfyUI
[4] https://pinokio.computer/

ImagenHub benchmark set [13], which provides complex tasks for image generation. Furthermore, we discuss current challenges in EA-driven image generation with diffusion models and identify future research directions.

We find that the guided search approaches significantly outperform the random trial and error approach on over 75% of the benchmark prompts. While the effectiveness of a variation operator depends on the problem at hand, we find that recombining the channel dimension of the latent tensor representation of two images, achieves the best mean rank on the studied benchmark prompts. For future work, we recommend researchers to explore the potential of different latent image representations as this may improve the ability of evolutionary algorithms to transfer promising features across images in ways that could lead to an even more effective image generation.

To sum up, our main contributions are:

(1) The ImageBreeder framework, which allows to study different aspects of EA-based image generation for diffusion models, such as the variation operators, the selection, or the objective function.
(2) An in-depth analysis of a wide range of different variation operators.
(3) The identification of challenges and future research directions in the area of EA-based image generation.

The remainder of this paper is structured as follows. Section 2 presents related work on EAs as well as deep learning based methods for image generation and evaluation. In Sect. 3, we describe the EA used by the ImageBreeder framework and list the studied variation operators. Section 4 presents the experimental setting and describes the achieved results. In Sect. 5, we discuss the results and identify future research directions before concluding the paper in Sect. 6.

2 Related Work

While evolutionary methods can be directly used for image generation [7, 15, 23], recently, methods from traditional machine learning are mostly used in the domain of text-to-image generation like generative adversarial networks [9, 18, 26], autoregressive methods [4, 17], and diffusion models [3, 12, 22]. Stable diffusion [19] is probably the best known approach to generate an image given a textual prompt that indicates the user's intent.

While these models offer impressive performance, they can also suffer from issues like insufficient quality, undesired or unsafe outputs, and misalignment with the user's intent [29].

Multiple approaches have been developed to improve these text-to-image generation models. One approach is to fine-tune the model parameters during training to be more aligned with human preferences with regards to some learned reward score [5, 14, 28, 29].

Other approaches try to optimise the model output during inference by adjusting the user prompt [11, 16, 27] or optimising latent vectors directly [2, 10, 21]. Both methods can be done either automatically using a reward model or interactively with humans in the loop.

Lastly, the image generation workflow as a whole can also be optimised [1, 25] or generated automatically from scratch [8].

Algorithm 1 ImageBreeder

Require: Image Description $Prompt$, Population Size P, Generations g, Tournament Size t, Strength Range $[s_{\min}, s_{\max}]$, Crossover Range $[c_{\min}, c_{\max}]$
Ensure: $BestImage$

Initialize $BestImage \leftarrow \emptyset$, $BestScore \leftarrow -\infty$
Initialize Population $Pop \leftarrow P * [\emptyset]$
for $i = 0$ to P **do** ▷ Initial Generation
  $Pop[i] \leftarrow Diffusion(Prompt)$
  $Score \leftarrow RewardModel(Pop[i].Image, Prompt)$
  if $Score > BestScore$ **then**
    $BestImage \leftarrow Pop[i].Image$
    $BestScore \leftarrow Score$
  end if
end for
for $k = 1$ to g **do** ▷ Generational Loop
  $Pop \leftarrow Tournament(Pop, t)$
  for $i = 0$ to $P - 1$ **do**
    $SplitPoint \leftarrow Random(c_{\min}, c_{\max})$
    $Pop[i] \leftarrow Variation(Pop[i], Pop[i+1], SplitPoint)$
    $Strength \leftarrow Random(s_{\min}, s_{\max})$
    $Pop[i] \leftarrow Diffusion(Prompt, Pop[i].Image, Strength)$
    $Score \leftarrow RewardModel(Pop[i].Image, Prompt)$
  end for
  $Pop[-1] \leftarrow Individual(BestImage, BestScore)$ ▷ Elitism
  for $i = 0$ to P **do**
    $Score \leftarrow Pop[i].Score$
    if $Score > BestScore$ **then**
      $BestImage \leftarrow Pop[i].Image$
      $BestScore \leftarrow Score$
    end if
  end for
end for
return $BestImage$

To the best of out knowledge, we are the first to explore applying different variation operator variants to populations of images generated by diffusion models to automatically evolve better solutions using the ImageReward score [29] as objective function.

3 ImageBreeder

We developed ImageBreeder to study the different aspects of EA-driven image generation with diffusion models. Albeit, we focus in this work on variation operators, ImageBreeder can easily be adapted to analyse different selection methods or objective functions. To enable further research with ImageBreeder, we made the code for the framework available online.[5]

In the following, we present the EA used in the experiments as well as the suggested variation operators.

3.1 Evolutionary Algorithm

ImageBreeder uses a simple EA strategy to successively improve the output of a diffusion model for image generation for a given prompt. Algorithm 1 shows the pseudo-code for this EA.

[5] https://github.com/domsob/ImageBreeder

To generate the initial population, we call the diffusion model P times and generate images for the given *Prompt* (see line 4). After generating each individual image for the population, we use a reward model to assign a *Score* to it (line 5). In our experiments we use the ImageReward model [29] for this purpose as this model is trained to evaluate both the image's visual appeal as well as the alignment with the prompt as human reviewers would do. It has already been confirmed several times in the literature that the scores given correspond well with the rating of human reviewers [6, 14, 25, 29]. During the generation of the initial population we make sure to save the best image as well as its corresponding score found so far (lines 6-9).

Next, we enter the generational loop which is repeated n times (line 11). Here we first perform a tournament selection to find the offspring for the next generation (line 12). Each image then undergoes the variation (for details see Sect. 3.2) and is also given as an additional conditioning to the diffusion model, which introduces additional minor changes (lines 14-17). To vary how strong these additional changes are, we randomly set the diffusion models *Strength* parameter to an adequate value (lines 16 and 17). After that, we evaluate the altered image with the reward model (line 18). Additionally, we introduce elitism to make sure that the best image found so far stays in the population (line 20). After the offspring population has been successfully generated, varied, and evaluated, we again update the best image found so far and its score (lines 21-27). In the end, the image with the highest score is returned (line 29).

3.2 Variation Operators

In each EA run, we use a single distinct variation operator. The crossover variants supported in the current version of ImageBreeder expect two images in pixel representation as well as a random split point (crossover point) as input and return one image (see Algorithm 1, line 15). The crossover variants in our study are as follows:

- **`pixel-blend`**: Blends two images in the pixel space using the `blend()` method from the Pillow package[6] for Python which employs the following equation for blending:

$$\text{image}_{\text{child}} = (1 - \alpha) * \text{image}_1 + \alpha * \text{image}_2,$$

  where image_1 and image_2 are the input images, $\text{image}_{\text{child}}$ is the blended output image, and α is the blending factor. As blending factor α, we use the provided random split point. The α value determines the weighting of the input images in the resulting blend.
- **`pixel-blend-with-score`**: Works like `pixel-blend` but instead of using the given split point, the α value is used to give the image with the better score more weight.
- **`latent-blend`**: Similar strategy as with `pixel-blend`, but works directly on the latent representation of the given images. For blending the latents of the input images, latent_1 and latent_2, we use the following equation:

$$\text{latent}_{\text{child}} = (1 - \alpha) * \text{latent}_1 + \alpha * \text{latent}_2,$$

  where $\text{latent}_{\text{child}}$ is the resulting blended latent and, as before, α is the blending factor determined by the given random split point.
- **`latent-blend-with-score`**: Uses the same weighting strategy as `pixel-blend-with-score` but employs the crossover method `latent-blend` to perform the blending directly on the latent representation of the input images.
- **`channel-cut`**: Cuts the latent tensors on the channel dimension at the given split point and recombines them. Randomly selects one of the two resulting outcomes and returns it as offspring.[7]
- **`horizontal-cut`**: Like `channel-cut` but slices the latent tensors of the images horizontally.
- **`vertical-cut`**: Like `channel-cut` and `horizontal-cut` but slices vertically.
- **`mixed-cut`**: Combines `channel-cut`, `horizontal-cut`, and `vertical-cut` by randomly using one of these methods each time it is called.
- **`mixed-all`**: Randomly selects and applies one of the previously mentioned crossover methods when called.

After the crossover has been applied, the resulting image is used to condition the diffusion model in image-to-image mode with a strength < 1.0. In our experiments, we also investigate how the EA behaves without using a crossover operator. We refer to these runs as **`mutation-only`**.

4 Experiments and Results

In this section, we describe the experimental setup for the EA as well as for the used diffusion model and present the achieved results.

4.1 Experimental Setting

For ImageBreeder's EA, we choose a population size P of 30 and set the number of generations $g = 15$ (in addition to the initial generation). We use tournament selection with a tournament size $t = 5$ and employ elitism by keeping the best image found so far during a run in the population.

As diffusion model, we choose ReV Animated 1.2.2[8] as this model generates qualitative images at a low inference time. We set the number of inference steps for the model to 30. For the inference in image-to-image mode (see Algorithm 1, line 17), we set the interval for randomly choosing the *Strength* parameter to $[0.6, 0.95]$.

For comparison, we generate the same number of images as the EA throughout its entire search using a random sampling approach with the same diffusion model and selecting the best generated image according to its ImageReward score. This approach corresponds to the approach of users who keep sampling new images until they find one they find visually appealing, serving as a strong baseline. In the following, we refer to this approach as **`random-sampling`**.

To evaluate the proposed EA with all its variation operators as well as the `random-sampling` approach, we randomly sampled 9 prompts from the ImagenHub benchmark set [13]. The sampled prompts we use in the experiments are shown in Table 1. The prompts range from geometric figures with textures to highly complex descriptions of persons or fictional characters.

[6] https://pillow.readthedocs.io/en/stable/reference/Image.html

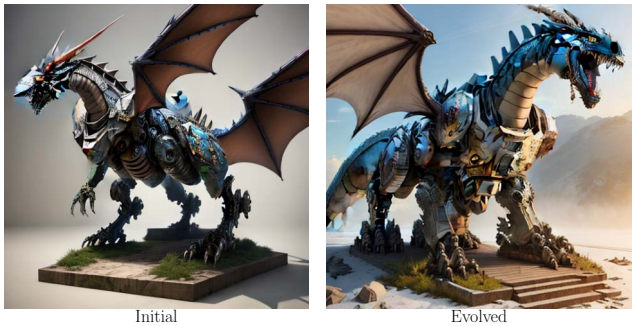
[7] For `channel-cut`, we biased the operator towards placing the cut in the middle.

[8] https://huggingface.co/danbrown/RevAnimated-v1-2-2

Figure 1: An example demonstrating the evolution of an image generated with the prompt: *"a cube made of brick. a cube with the texture of brick"*. For each image, we show the generation in which the image was found.

(a) Prompt: "emperor palpatine in the desert of tatooine, film still, wide shot, heat, desert, sci fi, epic, dramatic light"

(b) Prompt: "hyperrealism photo of a man with an orange hood covering his face"



(c) Prompt: "photorealistic, hyper detailed sculpture of a large mech dragon"

Figure 2: Example images evolved by ImageBreeder.

Table 1: Prompts used in the experiments. Sampled from ImagenHub [13].

#	Prompt
1	a cube made of brick. a cube with the texture of brick
2	a vehicle composed of two wheels held in a frame one behind the other, propelled by pedals and steered with handlebars attached to the front wheel
3	a yellow colored giraffe
4	beautiful witch, hyper detailed, flowing psychadelic background intricate and detailed, 8 k, octane render
5	big budget scifi movie set in postapocalyptic world with alien brainsuckers female cyborg in yellow rubber and gas mask in ruins of church
6	emperor palpatine in the desert of tatooine, film still, wide shot, heat, desert, sci fi, epic, dramatic light
7	hyperrealism photo of a man with an orange hood covering his face
8	photorealistic, hyper detailed sculpture of a large mech dragon
9	supreme court justices play a baseball game with the fbi. the fbi is at bat, the justices are on the field

Overall, we perform 8 runs for each prompt-method combination resulting in 792 runs. In these 792 runs, we generate and evaluate a total of 368.280 images.

4.2 Results

Figure 1 shows example images generated over the course of an evolutionary run for the prompt *"a cube made of brick. a cube with the texture of brick"*. The first image has the highest ImageReward score in the initial population sampled from a diffusion model. While the image is aligned with the prompt already, the quality is lacking: e.g., the image has very uneven rows of bricks and oversaturated lighting. After the first generation these flaws are mostly eliminated and, consequently, the ImageReward score increases. In the following generations a background is added and the textures of the block become more realistic and pleasant to look at with the best image found in generation 12.

Figure 2 provides further examples of images improved by our ImageBreeder framework. For each sub-figure the left image is the

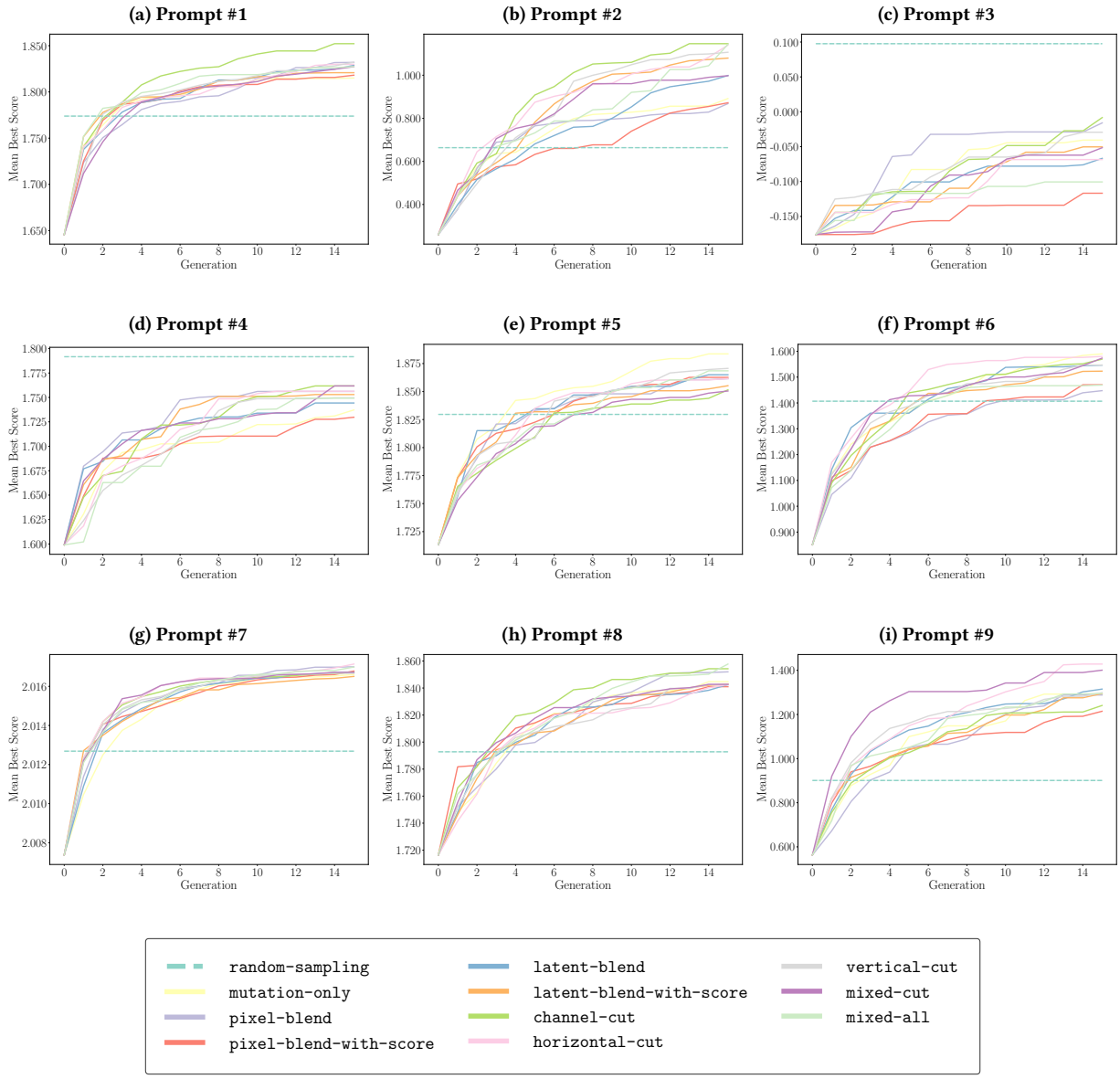


Figure 3: Mean best ImageReward score over generations for each studied method and prompt (higher values are better). The dashed line displays the score of the best image found by the random-sampling baseline.

best image of the initial population and the right image is the best image found by evolution (according to the ImageReward score). Figure 2a corresponds to the prompt *"emperor palpatine in the desert of tatooine, film still, wide shot, heat, desert, sci fi, epic, dramatic light"*. The initial image depicts a low detailed person in a desert. We can observe a Darth Vader-like helmet with several artifacts making the image only partially aligned with the prompt. In the evolved counterpart, we see a highly detailed person with a face that strongly resembles Emperor Palpatine. While the hood of the person still has elements of Darth Vader's helmet, the overall image is better aligned with the prompt. In Fig. 2b the initial image is already of high quality and mostly aligned with the prompt *"hyperrealism photo of a man with an orange hood covering his face"*. However, the face of the man is covered by a separate face mask instead of the orange hood itself. ImageBreeder is able to adjust this and further improves the alignment with the prompt without compromising visual fidelity. Lastly, Fig. 2c displays the initial and evolved image for the prompt *"photorealistic, hyper detailed sculpture of a large mech dragon"*. While the initial image is well aligned to the prompt, it suffers from quality issues like three legs and one arm, a blurry dragon head, a messy body, and a misplaced wing. The evolved

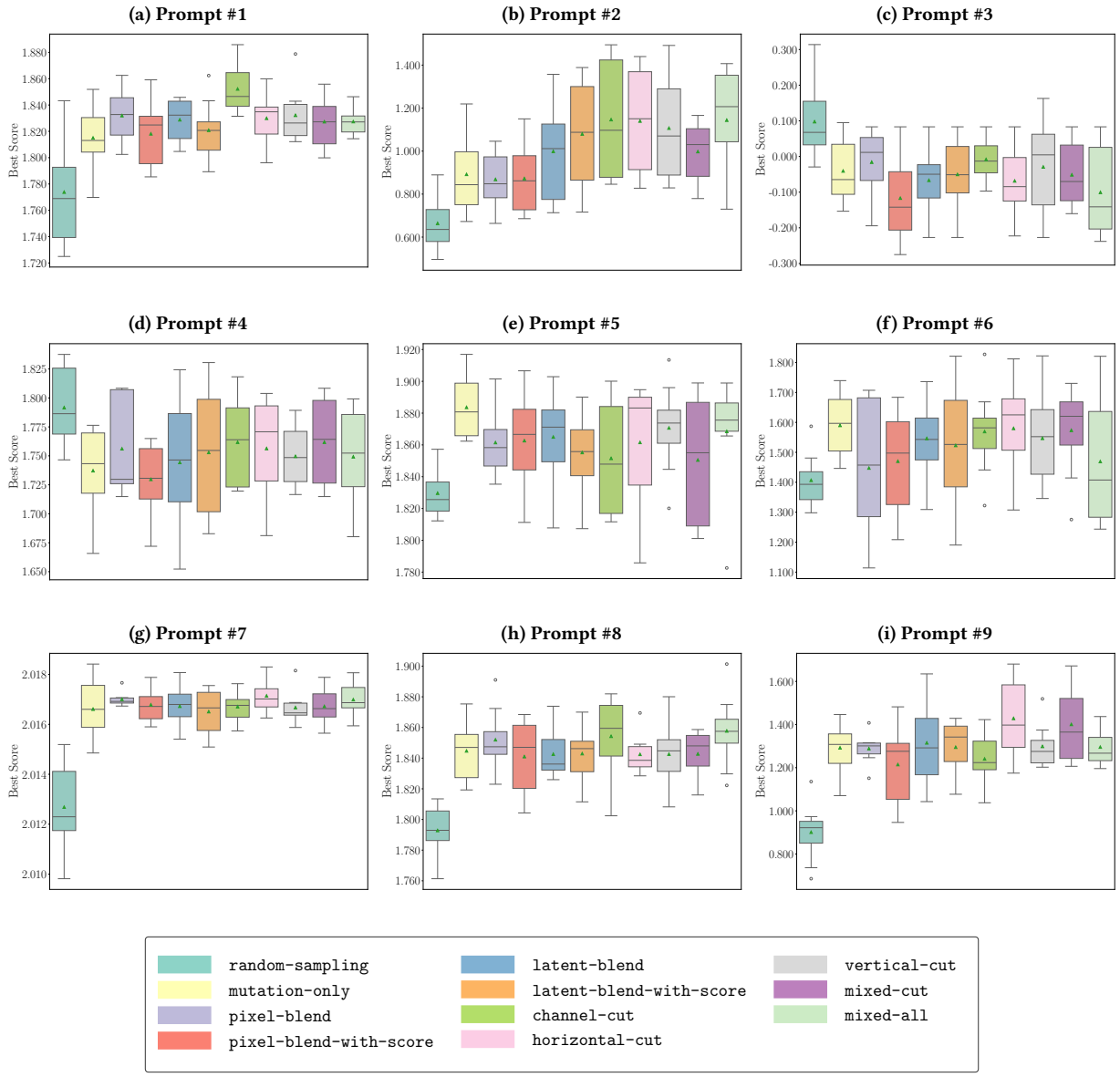


Figure 4: Box-plots for the ImageReward score for the best individual for each studied method and prompt (higher values are better). Each box-plot displays the results for all performed runs.

image has most of those quality issues resolved, only missing a second wing.

This exemplary observations are also reflected across different prompts and runs. Figure 3 displays the mean best ImageReward score over generations of all studied variation methods for all 9 prompts. The dashed line indicates the performance of the `random-sampling` baseline. We observe that the best score of the initial population is improved by ImageBreeder for all variation operators (including `mutation-only`) across all prompts. In 7 out of 9 prompts ImageBreeder is able to beat the `random-sampling` baseline approach. This happens already in early generations in which our method only generated a fraction of the images that the baseline has access to. ImageBreeder further improves performance over the course of 15 generations with all graphs still pointing upwards. This indicates that the evolutionary search could potentially further improve the images given additional computational resources. ImageBreeder is especially successful in improving the scores for scenarios where the initial population has a score that is acceptable but still low (ImageReward score above 0 but below 1) as it is the case for prompts #2, #6, and #9. These prompts are also rather complex descriptions in which the evolutionary search not only improves quality but also alignment with the user's intention.

Table 2: Mean rank and standard deviation (SD) for all studied methods across all prompts and runs. Best values are printed in bold font. A letter in superscript indicates significantly better performance compared to the corresponding method using a Wilcoxon signed-ranked test ($\alpha = 0.05$) and Bonferroni-Holm correction.

Variation	Mean	SD
`random-sampling`$_{a}$	8.25	3.77
`mutation-only`$_{b}$	5.96^{a}	2.97
`pixel-blend`$_{c}$	5.79^{a}	2.98
`pixel-blend-with-score`$_{d}$	6.86	2.95
`latent-blend`$_{e}$	5.83^{a}	3.01
`latent-blend-with-score`$_{f}$	6.14^{a}	3.16
`channel-cut`$_{g}$	**4.89**^{a}	3.04
`horizontal-cut`$_{h}$	5.07^{a}	2.93
`vertical-cut`$_{i}$	5.75^{a}	2.77
`mixed-cut`$_{j}$	5.86^{a}	2.99
`mixed-all`$_{k}$	5.60^{a}	3.02

Notably, prompt #3 seems to be challenging for the studied diffusion model resulting in an initial population with bad initial scores (ImageReward score below 0) which seems to prevent ImageBreeder from meaningfully improving the ImageReward score. We further discuss this in Sect. 5.

Figure 4 shows box-plots including means for all prompts representing the ImageReward score of the best individuals generated by the `random-sampling` baseline, as well as the evolutionary search variants. As expected, we observe similar results as before: ImageBreeder is able to outperform `random-sampling` in 7 out of 9 prompts with all variation operator variants. Which evolutionary variant performs best depends on the problem. E.g., `mutation-only` performs competitively in most problem settings, even achieving the best performance in prompt #5. Similar, the `channel-cut` crossover variant performs best for prompt #1 and also performs well for the other prompts. The only crossover operator that seems to perform notably worse than the other variation operators is `pixel-blend-with-score`.

To better compare the different methods we rank them by their achieved best scores for all runs and prompts. The best score for each run and prompt is assigned the rank 1 while the worst score is assigned the rank 11. We report the mean ranks and standard deviations of all methods in Table 2. The best mean rank is achieved by the `channel-cut` crossover variant with a mean rank of 4.89 while the worst mean rank of 8.25 is achieved by `random-sampling`. The evolutionary search variants with the lowest mean rank are `pixel-blend-with-score` and `latent-blend-with-score` with a mean rank of 6.86 and 6.14, respectively. We also perform a pairwise statistical test between all methods using a Wilcoxon signed-ranked test ($\alpha = 0.05$) and correct for multiple comparisons using a Bonferroni-Holm correction. We observe a statistical significant difference between the `random-sampling` baseline and all evolutionary search methods except the `pixel-blend-with-score` crossover. We do not observe any statistical difference between the evolutionary search variations themselves. This confirms that our ImageBreeder framework is indeed successful in improving diffusion model outputs at inference time. The best evolutionary variation operator, however, is problem depended.

Figure 5: Images generated for the prompt: "a vehicle composed of two wheels held in a frame one behind the other, propelled by pedals and steered with handlebars attached to the front wheel". This example illustrates how ImageBreeder can suffer from reward hacking by over-aligning the image to the "vehicle" part of the prompt.

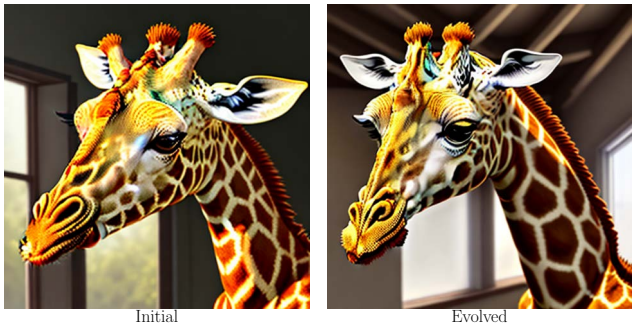


Figure 6: Images generated for the prompt: "a yellow colored giraffe". This example illustrates how ImageBreeder only does minimal changes with small improvements to the score if the guidance from the ImageReward model is insufficient.

5 Discussion and Future Research Directions

The results achieved so far show that with ImageBreeder it is possible to produce high quality images that are both visually appealing and aligned with the prompt. However, during our experiments we gained insights that the aggregated results shown so far cannot reflect. Therefore, in the following we discuss challenges that arise when combining EAs and diffusion models and also identify future research directions.

Even though the ImageReward model works very well in the vast majority of cases, the strong search performance of evolutionary methods can also be counterproductive resulting in an over-adjustment of the image content to the given prompt. For example, Fig. 5 shows the best image from the initial generation (left) and the image evolved by ImageBreeder (right) for an overly complex description of a bicycle given as prompt. The initially generated image shows a bicycle with some artifacts and quality issues.

However, a bicycle can be clearly recognized. The evolved image, however, over-adjusts to the word "vehicle" given in the prompt. It integrates a driver's cabin and combines it with a bicycle-like structure. The evolutionary search was able to discover a clever trick to increase the ImageReward score by interpreting the prompt more literally and thus increasing the attributed alignment which is included in the ImageReward score. In traditional machine learning, this phenomenon is also known as reward hacking [24].

However, this behaviour often has its advantages, which can also be seen in many of the examples shown so far. The biases of the ImageReward score also promote that images often get more detailed and have a more vibrant background as a result of the optimisation (see Figs. 1, 2b, and 2c).

Further, for the prompt *"a yellow colored giraffe"* we observed only small improvements of the ImageReward score in our experiments. We hypothesize that for this specific case the guidance of the ImageReward model is insufficient as the ImageReward model assigns very similar low scores to all generated images. This makes it hard to perform a guided search. For example in Fig. 6 the initial image correctly displays a giraffe with sufficient image fidelity but the ImageReward score is only −0.5. This is similar for the other generated images for this prompt resulting in also a very low score for the `random-sampling` baseline. Consequently, without proper guidance the evolved image has only minimal changes to the giraffe and the background, resulting in a similarly low score of −0.123.

During our experiments, we also noticed that the evolutionary search could benefit from smaller and more meaningful latent representations. With the model and the settings used in our experiments, the latent tensor has a shape of 4x64x64 which captures very small details of the images and is sometimes even close to the corresponding pixels in the decoded image. With a smaller latent representation, it would not only reduce the size of the search space but also force the representation to better capture the underlying latent variables. This would in turn make it easier for evolutionary search to adjust those latent variables (like the colour or number of objects in an image) and traverse the search space in a more meaningful way. From literature we know that good representations are crucial for the success of evolutionary methods [20]. Therefore, for future work we encourage researchers to explore this further.

Finally, we will take a closer look at how the images evolved with ImageBreeder can be further improved. A straightforward solution would be to simply use a larger and better diffusion model to evolve the images. However, this is not a suitable solution in practice, as larger models usually come with a significantly higher inference time. What is possible, however, is to use a more sophisticated model once in a post-hoc refinement procedure. Figure 7 shows an image where we have tested this post-hoc refinement exemplarily. The left image shows the image generated by ImageBreeder with the diffusion model ReV Animated 1.2.2 for the prompt *"emperor palpatine in the desert of tatooine, film still, wide shot, heat, desert, sci fi, epic, dramatic light"*. The right one shows an image generated with the Flux Schnell FP8[9] model conditioned with the evolved image (image-to-image generation). We can see that the facial contours have been softened and further details have been worked out. In addition, the blurry structure in the background is

Evolved / ReV Animated 1.2.2 Refined / Flux Schnell FP8

Figure 7: Images generated for the prompt: "emperor palpatine in the desert of tatooine, film still, wide shot, heat, desert, sci fi, epic, dramatic light". This example demonstrates that a post-hoc refinement with a computationally more expensive model can improve image quality even further.

now more detailed and sharper. It is also worth mentioning that the refinement of the evolved image in our test also lead to a better median ImageReward score than the direct generation of images with Flux Schnell FP8.[10] We assume that this is because important image details such as poses and the clothing of the persons depicted have already been worked out through evolution. We therefore recommend further pursuing this research direction and investigating image generation pipelines of this type.

6 Conclusion

In this paper we introduced ImageBreeder as a framework to leverage evolutionary search operators to automatically improve the image quality of diffusion models during inference time. We studied a wide range of variation operators across a diverse set of prompts. We found that evolutionary search significantly outperforms the baseline approach regardless of variation operators used, providing more aligned images with better visual fidelity.

We also discussed challenges for EA-based image generation with diffusion models and identified potential future research directions, like improving the latent representation to further improve search performance. Therefore, we encourage researchers to continue exploring EA-based approaches to enhance image generation with diffusion models at inference time.

References

[1] Harel Berger, Aidan Dakhama, Zishuo Ding, Karine Even-Mendoza, David Kelly, Hector Menendez, Rebecca Moussa, and Federica Sarro. 2023. Stableyolo: Optimizing image generation for large language models. In *International Symposium on Search Based Software Engineering*. Springer, 133–139.

[2] Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi. 2018. Deep interactive evolution. In *Computational Intelligence in Music, Sound, Art and Design: 7th International Conference, EvoMUSART 2018, Parma, Italy, April 4-6, 2018, Proceedings*. Springer, 267–282.

[3] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.

[4] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. 2021. Cogview: Mastering text-to-image generation via transformers. *Advances in neural information processing systems* 34 (2021), 19822–19835.

[9] https://huggingface.co/Comfy-Org/flux1-schnell

[10] 1.622 vs. 1.268, respectively (median, $n = 7$).

[5] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767* (2023).

[6] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. 2024. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems* 36 (2024).

[7] Erik M Fredericks, Jared M Moore, and Abigail C Diller. 2024. GenerativeGI: creating generative art with genetic improvement. *Automated Software Engineering* 31, 1 (2024), 23.

[8] Rinon Gal, Adi Haviv, Yuval Alaluf, Amit H Bermano, Daniel Cohen-Or, and Gal Chechik. 2024. ComfyGen: Prompt-Adaptive Workflows for Text-to-Image Generation. *arXiv preprint arXiv:2410.01731* (2024).

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[10] Ole Hall and Anil Yaman. 2024. Collaborative Interactive Evolution of Art in the Latent Space of Deep Generative Models. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 194–210.

[11] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. 2024. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems* 36 (2024).

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[13] Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. 2024. ImagenHub: Standardizing the evaluation of conditional image generation models. In *The Twelfth International Conference on Learning Representations*.

[14] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. 2023. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192* (2023).

[15] Matthew Lewis. 2008. Evolutionary visual art and design. In *The art of artificial evolution: A handbook on evolutionary art and music*. Springer, 3–37.

[16] Tiago Martins, João M Cunha, João Correia, and Penousal Machado. 2023. Towards the evolution of prompts with metaprompter. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 180–195.

[17] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*. Pmlr, 8821–8831.

[18] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *International conference on machine learning*. PMLR, 1060–1069.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[20] Franz Rothlauf. 2006. *Representations for genetic and evolutionary algorithms*. Springer.

[21] Baptiste Roziere, Fabien Teytaud, Vlad Hosu, Hanhe Lin, Jeremy Rapin, Mariia Zameshina, and Olivier Teytaud. 2020. Evolgan: Evolutionary generative adversarial networks. In *Proceedings of the Asian Conference on Computer Vision*.

[22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.

[23] Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. 2011. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary computation* 19, 3 (2011), 373–403.

[24] Joar Skalse, Nikolaus HR Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 9460–9471.

[25] Dominik Sobania, Martin Briesch, and Franz Rothlauf. 2024. ComfyGI: Automatic Improvement of Image Generation Workflows. *arXiv preprint arXiv:2411.14193* (2024).

[26] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. 2022. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16515–16525.

[27] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. 2024. On Discrete Prompt Optimization for Diffusion Models. *arXiv preprint arXiv:2407.01606* (2024).

[28] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. 2023. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2096–2105.

[29] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. 2024. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems* 36 (2024).