Hacking Generative Models with Differentiable Network Bending

Giacomo Aldegheri[*] University of Amsterdam

Alina Rogalska[†] Independent Researcher

Ahmed Youssef[†] University of Cincinnati

Eugenia Iofinova[†] IST Austria

Abstract

In this work, we propose a method to 'hack' generative models, pushing their outputs away from the original training distribution towards a new objective. We inject a small-scale trainable module between the intermediate layers of the model and train it for a low number of iterations, keeping the rest of the network frozen. The resulting output images display an uncanny quality, given by the tension between the original and new objectives that can be exploited for artistic purposes. Project website: https://galdegheri.github.io/diffbending/.

Figure 1: Example outputs, using a variety of loss functions and bending modules. See Appendix B for the corresponding prompts and more examples.

1 Introduction

Systems that fail to perform their intended task, or that are repurposed for new tasks, have long been recognized for their subversiveness and creative potential. Glitch artists [1] explore the aesthetic properties of media malfunctions, while hackers [2] take pleasure in the challenge of pushing existing hardware and software beyond its intended function [3]. In this work, we hack existing generative models to generate images they were not originally trained for. We adapt *network bending* [4], a technique consisting of injecting a transformation between intermediate layers of a generator, by making the transformation (the *bending module*, BM) differentiable. The images generated by these

[*]Correspondence to giacomo.aldegheri@gmail.com

[†]These authors contributed equally.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

hacked models are a blend of the objects that the original model was trained to generate (butterflies) and new visual features introduced by the BMs. We find that they exhibit an uncanny quality that can be exploited for creative purposes, similar to glitch art's use of unintended media artifacts.

Our method, thanks to its low computational cost, is accessible to a wide variety of artists and practitioners. Unlike current state of the art text-to-image pipelines, which aim to generate perfect-looking images, it provides a tool to explore strange and unexpected variations of existing models.

2 Method

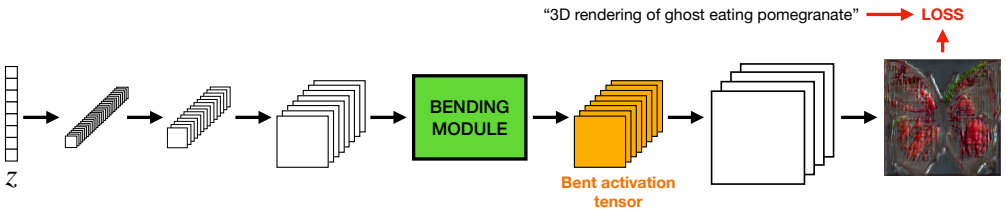

Figure 2: Overview of our proposed method. The BM takes the activation map of any chosen layer as input, and outputs a 'bent' activation map (shown in orange) which is fed as an input to the subsequent layers.

Architecture. The differentiable BM can be injected after any layer of a generator network. Here we use a lightweight GAN based on the architecture in [5] and trained on a dataset of butterfly images[3]. We chose a model trained on a narrow domain of images in order to be able to clearly distinguish the effect of the BM from the model's original structure. Specifically, in our outputs the outline of the butterfly remains visible to varying degrees depending on the specific BM used. The BM takes as input the activation map of the chosen layer, and outputs a tensor of the same dimensionality. We use a variety of small-scale network architectures for our BMs (see **Appendix** for architectural details): **(1) Convolutional:** a plain convolutional neural network (CNN), using either $ReLU(x)$ or $\sin(x)$ as an activation function. **(2) Convolutional + Coordinates:** a CNN with spatial coordinates (x, y and optionally r, the distance from the center) concatenated with the input features. This allows the BM to generate spatially-varying structures. **(3) Convolutional + Sorting:** a differentiable sorting network [6], operating across the width and/or height of the input feature map, followed by a CNN. This allows the BM to rearrange the spatial structure of the input activation map.

Loss function. While the differentiable BM can be trained with any objective, here we experiment with two loss functions. The first one is the squared great circle distance between the CLIP [7] embeddings of the output images and of a user-provided prompt, to generate semantically evocative images. The second one is a loss function that minimizes the distance between images and prompt in CLIP space, while maximizing the distance among different images in the batch, to increase output diversity in a semantically meaningful space. Following the approach of [8], we use the InfoNCE [9] contrastive objective to maximize the mutual information between image and caption embeddings, while minimizing that among images in the batch, as expressed by the following equation:

$$\mathcal{L}_{NCE} = \log \frac{e^{(Q \cdot K^+/\tau)}}{e^{(Q \cdot K^+/\tau)} + \sum_{K^-} e^{(Q \cdot K^-/\tau)}}$$

Where Q is the image embedding, K^+ the prompt embedding, K^- are the embeddings of the other images in the batch, and τ is a temperature hyperparameter.

3 Ethical Implications

The work was done entirely on open-source or publicly available models and data. In particular, we avoided, as much as possible, the use of tools and data that use the uncompensated work of

[3]This model is available at https://huggingface.co/ceyda/butterfly_cropped_uniq1K_512.

(traditional) artists. The only possible exception is the use of the pretrained CLIP model, for which we were not aware of alternatives.

Further, since our method provides only a very coarse control of the output, we do not believe that it aids the creation of works that break copyright or produce obscene or incendiary images, over what is already available.

References

[1] Rosa Menkman. Glitch studies manifesto. *Video vortex reader II: Moving images beyond YouTube*, pages 336–347, 2011.

[2] Jon Erickson. *Hacking: the art of exploitation*. No starch press, 2008.

[3] Zack Kotzer. A catalogue of all the devices that can somehow run 'Doom'. *VICE*, May 2016 (Accessed 20 September 2023). URL https://www.vice.com/en/article/qkjv9x/a-catalogue-of-all-the-devices-that-can-somehow-run-doom.

[4] Terence Broad, Frederic Fol Leymarie, and Mick Grierson. Network bending: Expressive manipulation of deep generative models. In *Artificial Intelligence in Music, Sound, Art and Design: 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 10*, pages 20–36. Springer, 2021.

[5] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2020.

[6] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Monotonic differentiable sorting networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=IcUWSh ptD7d.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[8] Yingchen Yu, Fangneng Zhan, Rongliang Wu, Jiahui Zhang, Shijian Lu, Miaomiao Cui, Xuansong Xie, Xian-Sheng Hua, and Chunyan Miao. Towards counterfactual image manipulation via CLIP. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3637–3645, 2022.

[9] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.