BIGRoC: Boosting Image Generation via a Robust Classifier

Roy Ganz [1] **Michael Elad** [2]

Abstract

The interest of the machine learning community in image synthesis has grown significantly in recent years, with the introduction of a wide range of deep generative models and means for training them. In this work, we propose a general model-agnostic technique for improving the image quality and the distribution fidelity of generated images, obtained by any generative model. Our method, termed BIGRoC (Boosting Image Generation via a Robust Classifier), is based on a post-processing procedure via the guidance of a given robust classifier and without a need for additional training of the generative model. Given a synthesized image, we propose to update it through projected gradient steps over the robust classifier, in an attempt to refine its recognition. We demonstrate this post-processing algorithm on various image synthesis methods and show a significant improvement, both quantitatively and qualitatively, on CIFAR-10 and ImageNet. Specifically, BIGRoC improves the best performing diffusion model on ImageNet 128×128 by 14.81%, attaining an FID score of 2.53 and on 256×256 by 7.87%, achieving an FID of 3.63.

1. Introduction

Deep generative models are a class of deep neural networks trained to model complicated high-dimensional data (Bond-Taylor et al., 2021). Such models receive a large number of samples that follow a certain data distribution, $x \sim P_D(x)$, and aim to produce new ones from the same statistics. One of the most fascinating generative tasks is image synthesis, which is notoriously hard due to the complexity of the natural images' manifold. Nevertheless, deep generative models for image synthesis have gained tremendous popularity in recent years, revolutionized the field, and became state-of-the-art in various tasks (Isola et al., 2017; Zhu et al., 2017; Karras et al., 2018; 2019; Brock et al., 2019; Karras et al., 2020). Energy-based models, variational autoencoders (VAEs), generative adversarial networks (GANs), autoregressive likelihood models, normalization flows, diffusion-based algorithms, and more, all aim to synthesize natural images, ranging from relatively simple to extremely complicated generators, often containing millions of parameters (Kingma & Welling, 2014; Goodfellow et al., 2014; Rezende & Mohamed, 2015; Oord et al., 2016; Ho et al., 2020).

When operating on a multiclass labeled dataset, as considered in this paper, image synthesis can be either conditional or unconditional. In the unconditional setup, the generative model aims to produce samples from the target data distribution without receiving any information regarding the target class of the synthesized images, i.e., a sample from $P_D(x)$. In contrast, in the conditional setup, the generator goal is to synthesize images from a designated class, i.e., a sample from $P_D(x|y)$ where y is the label. As such, conditional generative models receive class-related information.

Most of the work in the deep generative models' field has been focusing on improving the quality and the variety of the images produced by such models, tackled by seeking novel architectures and training procedures. In this work, while still aiming to improve the performance of trained generative models, we place a different emphasis than in most of these studies and propose a method for boosting generative models without any re-training or fine-tuning. More specifically, our method improves the perceptual quality of the images synthesized by any given model via an iterative post-processing procedure driven by a *robust classifier*.

With the introduction of learning-based machines into "real-world" applications, the interest in the robustness of such models has become a central concern. While there are abundant definitions for robustness, the most common and studied is the adversarial one. This definition upholds if a classifier is robust to a small perturbation of its input, made by an adversary to fool it. Previous work (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017) has demonstrated that deep neural networks are not robust at all and can be easily fooled by an adversary. In light of this observation, many robustification methods were proposed, but the most popular among these is adversarial training

[1]Department of EE, Technion - Israel Institute of Technology, Haifa, Israel [2]Department of CS, Technion - Israel Institute of Technology, Haifa, Israel. Correspondence to: Roy Ganz <ganz.campus@technion.ac.il>.

(Goodfellow et al., 2015; Madry et al., 2018). According to this method, in order to train a robust classifier, one should generate adversarial examples and incorporate them into the training process. While examining the properties of such classifiers, researchers have revealed a fascinating phenomenon called *perceptually aligned gradients* (Tsipras et al., 2019). This trait implies that modifying an image to sharpen such a classifier's decision yields visual features perceptually aligned with the target class. In other words, drifting an image to be better classified yields visually pleasing changes that are faithful to natural image content.

In this work, we harness and utilize the above-described phenomenon – we propose to iteratively modify the images created by a trained generative model so as to maximize the conditional probability of a target class approximated by a given robust classifier. This modification can potentially improve the quality of the synthesized images since it emphasizes visual features aligned with the target class. Due to the fundamental differences between classification and generation, we hypothesize that the robust classifier could capture different semantic features than the generative model. Thus, incorporating the two for image refinement can improve sample quality. We term this method "BIGRoC" – **B**oosting **I**mage **G**eneration via a **Ro**bust **C**lassifier.

The method presented in this article is general and model-agnostic that can be applied to any image generator, both conditional or unconditional, without requiring access to its weights, given an adversarially trained classifier. The marked performance improvement achieved by our proposed method is demonstrated in a series of experiments on a wide range of image generators on CIFAR-10 and ImageNet datasets. We show that this approach enables us to significantly improve the quality of images synthesized by relatively simple models, boosting them to a level of more sophisticated and complex ones. Furthermore, we show the ability of our method to enhance the performance of generative architectures of the highest quality, both qualitatively and quantitatively. Specifically, applying BIGRoC on the outputs of guided diffusion, (Dhariwal & Nichol, 2021) significantly improves its performance on ImageNet 128×128 and 256×256 – achieving FIDs of 2.53 and 3.63, an improvement of 14.81% and 7.87%, respectively. As such, our work exposes the striking generative capabilities of adversarially trained classifiers, that was only partially discovered by (Santurkar et al., 2019), as explained in Appendix B.

2. Background

2.1. Adversarial Examples

Adversarial examples are instances intentionally designed by an attacker to cause a false prediction by a machine learning-based classifier (Szegedy et al., 2014; Goodfellow et al., 2015; Kurakin et al., 2017). The generation procedure of such examples relies on applying modifications to given training examples while restricting the allowed perturbations Δ. Ideally, the "threat model" Δ should include all the possible unnoticeable perturbations to a human observer. As it is impossible to rigorously define such a set, in practice a simple subset of the ideal threat model is used, where the most common choices are the ℓ_2 and the ℓ_∞ balls: $\Delta = \{\delta \ : \ \|\delta\|_{2/\infty} \leq \epsilon\}$. Given Δ, the attacker receives an instance x and generates $\hat{x} = x + \delta \ s.t. \ \delta \in \Delta$, while aiming to fool the classifier. Adversarial attacks can be both untargeted or targeted: An untargeted attack perturbs the input in a way that minimizes $p(y|\hat{x})$ with respect to δ. In contrast, a targeted attack receives in addition the target class $\hat{y}$, and perturbs x to maximize $p_(\hat{y}|\hat{x})$. There are diverse techniques for generating adversarial examples, yet, in this work, we focus on targeted attacks using the Projected Gradient Descent (PGD) (Madry et al., 2018)– an iterative method for creating adversarial examples that operates as shown in Algorithm 1, where Π_ϵ is the projection operator onto Δ, and $\ell(\cdot)$ is the classification loss.

Algorithm 1 Targeted Projected Gradient Descent (PGD)

Input: classifier f_θ, input x, target class $\hat{y}$, ϵ, step size α, number of iterations T
$\delta_0 \leftarrow 0$
for *t from 0 to T* **do**
  $\delta_{t+1} = \Pi_\epsilon(\delta_t - \alpha \nabla_\delta \ell(f_\theta(x + \delta_t), \hat{y}))$;
end
$x_{adv} = x + \delta_T$
Output: x_{adv}

2.2. Adversarial Robustness

Adversarial robustness is a property of classifiers, according to which applying a small perturbation on a classifier's input in order to fool it does not affect its prediction (Goodfellow et al., 2015). To attain such classifiers, one should solve the following optimization problem:

$$\min_\theta \sum_{x,y \in D} \max_{\delta \in \Delta} \ell(f_\theta(x + \delta), y) \tag{1}$$

Namely, train the classifier to accurately predict the class labels of the "toughest" perturbed images, allowed by the threat model Δ. In practice, solving this optimization problem is challenging, and there are several ways to attain an approximated solution. The most simple yet effective method is based on approximating the solution of the inner-maximization via adversarial attacks, such as PGD (Madry et al., 2018). According to this strategy, the above optimization is performed iteratively, fixing the classifier's parameters θ and optimizing the perturbation δ for each example, and then fixing these and updating θ. Repeating these steps results in a robust classifier, as we use in this work.

2.3. Perceptually Aligned Gradients

Perceptually aligned gradients (PAG) is a trait of adversarially trained models, best demonstrated when modifying an image to maximize the probability assigned to a target class. (Tsipras et al., 2019) show that performing the above process on such models yields meaningful features aligned with the target class. It is important to note that this phenomenon does not occur in non-robust models. The perceptually aligned gradients property indicates that the features learned by robust models are more aligned with human perception. We present a visual demonstration of this fascinating phenomenon in Figure 2 in Appendix A.

3. Boosting Image Generation via a Robust Classifier

We propose a method for improving the quality of images synthesized by trained generative models, named BIGRoC: Boosting Image Generation via a Robust Classifier. Our method is model agnostic and does not require additional training or fine-tuning of the generative model that can be viewed as a post-processing step performed on the synthesized images. Thus, BIGRoC can be easily applied to any generative model, both conditional or unconditional. This mechanism harnesses the perceptually aligned gradients phenomenon to modify the generated images to improve their visual quality. To do so, we perform an iterative process of modifying the generated image x to maximize the posterior probability of a given target class $\hat{y}$, $p_\theta(\hat{y}|x)$, where p_θ is modeled by an adversarially trained classifier. This can be achieved by performing a PGD-like process that instead of adversarially changing an image x of class y to a different class $\hat{y} \neq y$, modifies it in a way that maximizes the probability that x belongs to class y. Therefore, our method requires a trained robust classifier that operates on the same data source as the generated images we aim to improve.

In the conditional generation process, the generator G receives the class label y, from which it suppose to draw samples. Hence, in this setup, we have information regarding the class affiliation of the image, and we can maximize the corresponding conditional probability. In the unconditional generation process, the generator does not receive class labels at all, and its goal is to draw samples from $p(x)$. Thus, in this case, we cannot directly maximize the desired posterior probability, as our method suggests. To bridge this gap, we propose to estimate the most likely class via our robust classifier f_θ and afterward modify the image via the suggested method to maximize its probability. The proposed image generation boosting is described in Algorithm 2 for both of the setups.

While the above-described approach for unconditional sampling works well, it could be further improved. We have noticed that in this case, estimating the target classes Y of X_{gen} via f_θ might lead to unbalanced labels. For example, in the CIFAR-10, when generating 50,000 samples, we expect approximately 5,000 images per each of the ten classes, and yet the labels' estimation does not distribute uniformly at all. This imbalance stems from the incompetence of the generative model to capture the data distribution, which leads to a bias in the target labels estimation of the boosting algorithm, affecting the visual content of X_{boost}. Since quantitative evaluation is impaired by such class imbalances, this bias limits the quantitative improvement attained by BIGRoC. We emphasize that this issue is manifested only in the quantitative metrics, and when qualitatively evaluating the boosted images, the improvement is significant, as can be seen in Appendix G.

To further enhance the quantitative results of our algorithm in the unconditional case, we propose to de-bias the target class estimation of X_{gen} and attain close to uniform class estimations. A naive solution to this can be achieved by generating more samples and extracting a subset of these images with a labels-balance. This approach is computationally heavy and does not use the generated images as-is, raising questions regarding the fairness of the quantitative comparison. Thus, we propose a different debiasing technique – we modify the classifier's class estimation to become more balanced by calibrating its logits. More specifically, we shift the classifier's logits by adding a per-class pre-calculated value, d_{c_i}, that induces equality of the mean logits value across all classes. We define $\mathbf{d_c}$ as a vector containing all d_{c_i} values: $\mathbf{d_c} = [d_{c_0}, \dots d_{c_{N-1}}]$ where N is the number of classes. For simplicity, we denote $logit_{c_i}$ as the logit of class c_i corresponding to a generated sample x_{gen}. We approximate $\mathbb{E}_{x_{gen}}[logit_{c_i}]$ for each class c_i, using a validation set of generated images, and calculate a per-class debiasing factor: $d_{c_i} = a - \hat{\mathbb{E}}_{x_{gen}}[logit_{c_i}]$ (WLOG, $a = 1$), where $\hat{\mathbb{E}}_{x_{gen}}[logit_{c_i}]$ is a mean estimator. After calculating d_{c_i}, given a generated image x_{gen}, we calculate its logits and add d_{c_i} to it to obtain debiased logits ($\hat{logit}_{c_i}$), from which we derive the unbiased class estimation via softmax. The following equation shows that, given a correct estimation of the per-class logits' mean, the per-class means of the debiased logits are equal:

$$\mathbb{E}_{x_{gen}}[\hat{logit}_{c_i}] = \mathbb{E}_{x_{gen}}[d_{c_i} + logit_{c_i}] =$$
$$\mathbb{E}_{x_{gen}}[a - \hat{\mathbb{E}}_{x_{gen}}[logit_{c_i}] + logit_{c_i}] =$$
$$a - \hat{\mathbb{E}}_{x_{gen}}[logit_{c_i}] + \mathbb{E}_{x_{gen}}[logit_{c_i}] \approx a$$

Figure 11 in Appendix G presents an empirical demonstration that verifies the validity of this method and shows its qualitative effects on unconditional generation boosting.

As can be seen in Algorithm 2, it receives as input the generated images and their designated labels (if exist) and returns

Algorithm 2 BIGRoC: Boosting Image Generation via a Robust Classifier

Input: Robust classifier f_θ, x_{gen}, y_{gen}, ϵ, step size α, number of iterations T, $\mathbf{d_c}$
if y_{gen} *is None* **then**
  $y_{gen} = argmax(f_\theta(x_{gen}) + \mathbf{d_c})$
end
$x_{boost} = Targeted\ PGD(f_\theta, x_{gen}, y_{gen}, \epsilon, \alpha, T)$
Output: x_{boost}

an improved version of them. As such, this method can be applied at the inference phase of generative models to enhance their performance in a complete separation from their training. Furthermore, BIGRoC does not require access to the generative models at all, and thus can be applied on standalone images, regardless of their origin. As can be seen from the algorithm's description, it has several hyperparameters that determine the modification process of the image: ϵ sets the maximal size of the perturbation allowed by the threat model Δ, α controls the step size at each update step, and T is the number of updates. Another choice is the norm used to define the threat model Δ.

The hyperparameter ϵ is central in our scheme - when ϵ is too large, the method overrides the input and modifies the original content in an unrecognizable way, as can be seen in Figure 2. On the other hand, when ϵ is too small, the boosted images remain very similar to the input ones, leading to a minor enhancement. As our goal is to obtain a significant enhancement to the synthesized images, a careful choice of ϵ should be practiced, which restricts the allowed perturbations in the threat model.

Another important choice is the threat model Δ itself. Two of the most common choices of Δ for adversarial attacks are the ℓ_∞ and the ℓ_2 balls. Due to the desired behavior of our method, using the ℓ_∞ ball is less preferable: it allows a change of $\pm\epsilon$ to every pixel, and as such, it will not focus on meaningful specific locations, and might not preserve the existing structure of the synthesized input image. Thus, we choose the ℓ_2 ball as our threat model, with relatively small ϵ. Such a choice restricts the allowed perturbations and leads to changes that may concentrate in specific locations while preserving most of the existing content in the generated images. A visual demonstration of these considerations is given in Figure 3 in Appendix D.1.

To better understand BIGRoC's contribution, we compare our method with other work addressing sample refinement as we do. While the leading competitive methods rely on the guidance of GANs discriminator and are thus suited for GANs only, we propose a general and model agnostic approach. Moreover, such methods require access to the weights of the entire GAN and the latent code of the generated images. However, we propose a method with a much simple setting that does not require access to the generative models and can operate on standalone images. In Appendix B we provide a complete overview of the related and competitive work and compare it to ours.

4. Experimental Results

In this section, we present experiments that demonstrate the effectiveness of our method on the most common datasets for image synthesis – CIFAR-10 (Krizhevsky, 2012) and ImageNet (Deng et al., 2009). Given a generative model, we use it to synthesize a set of images X_{gen} and apply our method to generate X_{boost}, according to Algorithm 2. We utilize the model-agnostic property and the simplicity of BIGRoC to examine its effects over a wide variety of image generators of different qualities: from relatively simple to sophisticated and complex ones. Moreover, we test our method on both conditional and unconditional generative models to validate that the proposed scheme can enhance different synthesis procedures. We term an application of our approach without ground truth labels as BIGRoC$_{\text{PL}}$, as it generates pseudo labels (PL), and with such as BIGRoC$_{\text{GT}}$.

In all the experiments, for each dataset, we use the same adversarial robust classifier to refine all the generated images of different sources. The only needed adjustment is at tuning ϵ, which defines the allowed size of the visual modifications done by BIGRoC. The striking fact that a single robust classifier improves both low and high-quality images strongly demonstrates the versatility of our approach and the surprising refinement capabilities possessed by such a model. We analyze BIGRoC performance both quantitatively and qualitatively, using Fréchet Inception Distance (FID, (Heusel et al., 2017), lower is better), and Inception Score (IS, (Salimans et al., 2016), higher is better). We provide additional explanations regarding our evaluation strategy in Appendix C. In addition, we compare our approach with other image refinement SOTA methods (see a complete overview of these in Appendix B).

4.1. CIFAR-10

In this section, we evaluate the performance of the proposed BIGRoC on the CIFAR-10 dataset, using a single publicly-available adversarially trained ResNet-50 on CIFAR-10 as the robust classifier (Engstrom et al., 2019).

Tested Architectures In the conditional image generation setup, we experiment with cGAN (Mirza & Osindero, 2014), cGAN-PD (Miyato & Koyama, 2018), BigGAN (Brock et al., 2019) and Diff BigGAN (Zhao et al., 2020). In the unconditional case, we experiment with Variational AutoEncoder (VAE) (Kingma & Welling, 2014), DCGAN (Radford et al., 2015), WGAN-GP (Salimans et al., 2016), SNGAN

Table 1. BIGRoC$_{GT}$ quantitative results on CIFAR-10.

ARCHITECTURE	FID	IS
CGAN	29.26 ± 0.10	6.95 ± 0.03
W/ BIGROC	8.89 ± 0.05	8.57 ± 0.05
CGAN-PD	11.10 ± 0.07	8.54 ± 0.03
W/ BIGROC	8.33 ± 0.06	8.76 ± 0.05
BIGGAN	7.45 ± 0.08	9.38 ± 0.05
W/ BIGROC	6.79 ± 0.02	9.47 ± 0.02
DIFF BIGGAN	4.37 ± 0.03	9.48 ± 0.03
W/ BIGROC	3.95 ± 0.02	9.61 ± 0.03

Table 2. BIGRoC$_{PL}$ quantitative results on CIFAR-10.

ARCHITECTURE	FID	IS
VAE	152.04 ± 0.19	3.05 ± 0.01
W/ BIGROC	88.68 ± 0.37	6.27 ± 0.04
DCGAN	38.34 ± 0.11	6.10 ± 0.01
W/ BIGROC	29.93 ± 0.05	7.28 ± 0.04
WGAN-GP	22.62 ± 0.09	7.49 ± 0.03
W/ BIGROC	16.28 ± 0.08	8.15 ± 0.03
SNGAN	17.19 ± 0.07	8.04 ± 0.02
W/ BIGROC	13.25 ± 0.10	8.61 ± 0.04
INFOMAXGAN	15.41 ± 0.12	8.09 ± 0.05
W/ BIGROC	11.27 ± 0.11	8.48 ± 0.03
SSGAN	15.05 ± 0.06	8.20 ± 0.02
W/ BIGROC	10.77 ± 0.06	8.61 ± 0.03

Table 3. Quantitative results on ImageNet 128×128.

ARCHITECTURE	FID	IS
SNGAN	62.28	13.05
W/ BIGROC$_{PL}$	40.40	71.67
SSGAN	63.60	13.75
W/ BIGROC$_{PL}$	38.93	73.94
INFOMAXGAN	60.61	13.79
W/ BIGROC$_{PL}$	37.70	75.49
BIGGAN-DEEP	6.02	145.83
W/ BIGROC$_{PL}$	5.69	176.42
W/ BIGROC$_{GT}$	5.71	226.17
GUIDED DIFFUSION	2.97	141.37
W/ BIGROC$_{PL}$	2.77	150.43
W/ BIGROC$_{GT}$	2.53	169.73

Table 4. Quantitative results on ImageNet 256×256.

ARCHITECTURE	FID	IS
BIGGAN-DEEP	7.03	202.65
W/ BIGROC$_{PL}$	6.93	221.78
W/ BIGROC$_{GT}$	6.84	228.23
GUIDED DIFFUSION	3.94	215.84
W/ BIGROC$_{PL}$	3.69	249.91
W/ BIGROC$_{GT}$	3.63	260.02

(Miyato et al., 2018), InfoMaxGAN (Lee et al., 2021) and SSGAN (Chen et al., 2019).

Quantitative Results Table 1 contains our quantitative results for conditional image synthesis and Table 2 for the unconditional case. These results indicate that BIGRoC achieves a substantial improvement across a wide range of tested generator architectures, both conditional and unconditional, demonstrating the method's versatility and validity.

Qualitative Results To strengthen the quantitative results, we show in Figure 1(a) and in Appendix F qualitative results that verify that the "boosted" results indeed look better to human observers.

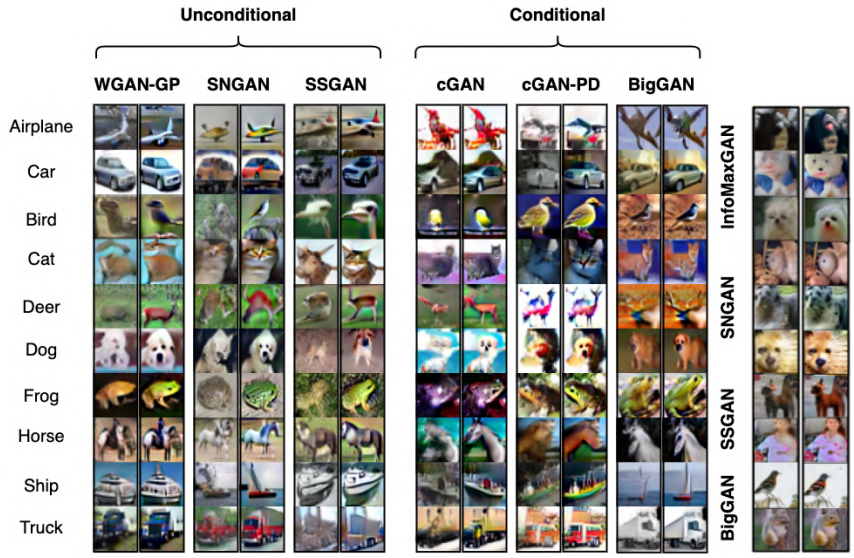
4.2. ImageNet

We turn to evaluate the performance of the proposed BIGRoC on the ImageNet 128×128 and 256×256 datasets, using a single publicly available adversarially trained ResNet-50 on ImageNet as the robust classifier (Engstrom et al., 2019) for both of the resolutions.

Tested Architectures For ImageNet 128×128, we experiment with SNGAN (Miyato et al., 2018), SSGAN (Chen et al., 2019), InfoMaxGAN (Lee et al., 2021), BigGAN-deep (Brock et al., 2019) and Guided Diffusion (ADM-G) (Dhariwal & Nichol, 2021). For ImageNet 256×256, we use BigGAN-deep and Guided Diffusion (ADM-G,ADM-U).

Quantitative Results Tables 3 and 4 summarize our quantitative results on ImageNet 128×128 and 256×256, respectively. These results strongly indicate that BIGRoC is also highly beneficial for higher-resolution images from richer datasets. Specifically, BIGRoC improves the best-performing diffusion model (Dhariwal & Nichol, 2021) on ImageNet 128×128 by 14.81%, leading to FID of 2.53, and on 256×256 by 7.87%, leading to FID of 3.63. The obtained results show a similarly marked improvement in IS. In addition, our approach outperforms this model even when the already available ground truth labels are disregarded (BIGRoC$_{PL}$).

Qualitative Results To verify that our method indeed leads to better perceptual quality, we show visual results on ImageNet 128×128 in Figure 1(b). Moreover, we present Figure 6 in in Appendix F a qualitative comparison between images generated by a guided-diffusion trained on ImageNet 256×256 and the outputs of BIGRoC applied to them. In addition, we show the images' differences (after contrast stretching) to better grasp the perceptual modifications applied by our method. As can be seen, BIGRoC focuses on the edges and textures and leads to sharper and refined images, which are more pleasing to a human observer.



(a) CIFAR-10 (b) ImageNet

Figure 1. BIGRoC's qualitative results: We present pairs of images – the left columns contain generated images, and the right ones contain the boosted results. Figure 1(a): We show six pairs of sets of images where each contains ten images, one per CIFAR-10 class. Each pair contains the generated images (left) by different CFAR-10 GANs, and their "boosted version" (right), attained by applying BIGRoC. Figure 1(b): We show pairs of images generated by different GANs, trained on ImageNet 128×128 and their "boosted" version.

4.3. Comparison with other methods

Previous works, such as DOT (Tanaka, 2019), DDLS (Che et al., 2021) and DG-$f$low (Ansari et al., 2021), address image generation refinement, as we do. As stated in Section 3 and further explained in Appendix B, we propose a much simpler approach than our competitors and require much less information since BIGRoC can operate without access to the image generator, the discriminator, and without knowing the latent codes corresponding with the generated images. In this section, we aim to demonstrate that although our method utilizes less information than other methods and can operate in setups in which the other approaches can not, BIGRoC performs on par and even better. To this end, we compare our method with current SOTA image refinement methods on CIFAR-10 and ImageNet 128×128. For both these setups, we adopt the same publicly available pre-trained models of SN-ResNet-GAN as used in these work and apply our algorithm to the generated images and evaluate its performance quantitatively (see Appendix E.3 for additional implementation details). In Table 5 we compare the quantitative results of DOT, DDLS, and DG-$f$low with BIGRoC using IS, as this is the common metric reported in these papers.

Table 5. Quantitative comparison between BIGRoC and competitive methods on CIFAR-10 and ImageNet 128×128, using IS.

ARCHITECTURE	INCEPTION SCORE	
	CIFAR-10	IMAGENET
SN-RESNET-GAN	8.38	36.8
W/ DOT	–	37.29
W/ DDLS	9.09	40.2
W/ DG-fLOW	**9.35**	–
W/ BIGROC	9.33	**44.68**

5. Discussion and Conclusions

In this work, we propose a novel method that leverages the perceptually aligned gradients phenomenon for enhancing the visual quality of synthesized images. Due to the core ability of such a robust classifier to better capture significant visual features, it is capable of effectively and efficiently improving the output of generative models. Our approach does not require additional training of the generative model, and it is completely model agnostic, unlike other methods. Thus, BIGRoC can be applied during the inference phase of any generator. Moreover, it can operate on sets of images without requiring access to the generative model themselves and the latent code, a setup where competing methods can not. In a line of experiments, we show that our method is highly effective and capable of substantially enhancing qualitative and quantitative results of various generative models over multiple datasets. Moreover, the fact that the same robust classifier can improve generators of different qualities exposes its surprising generative capabilities.

6. Acknowledgements

This research was partially supported by the Israel Science Foundation (ISF) under Grant 335/18.

References

Ansari, A. F., Ang, M. L., and Soh, H. Refining deep generative models via discriminator gradient flow, 2021.

Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., and Odena, A. Discriminator rejection sampling, 2019.

Bond-Taylor, D., Leach, A., Long, Y., and Willcocks, C. G. Deep generative modelling: A comparative review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models, 2021.

Brock, A., Donahue, J., and Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *7th International Conference on Learning Representations, ICLR*, 2019.

Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling, 2021.

Chen, T., Zhai, X., Ritter, M., Lucic, M., and Houlsby, N. Self-Supervised GANs via Auxiliary Rotation Loss. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 248–255, 2009.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems, NeurIPS*, pp. 2672–2680, 2014.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations, ICLR*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 770–778, 2016.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems, NeurIPS*, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations, ICLR*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In *Conference on Computer Vision and Pattern Recognition, CVPR*, 2020.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations, ICLR*, 2014.

Krizhevsky, A. Learning multiple layers of features from tiny images. *University of Toronto Technical Report*, 2012.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world, 2017.

Lee, K. S., Tran, N. T., and Cheung, N. M. Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations, ICLR*, 2018.

Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. *CoRR*, 2014.

Miyato, T. and Koyama, M. cGANs with Projection Discriminator. In *International Conference on Learning Representations, ICLR*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations, ICLR*, 2018.

Oord, A. V., Kalchbrenner, N., and Kavukcuoglu, K. Pixel Recurrent Neural Networks. In *International Conference on Machine Learning, ICML*, pp. 1747—1756, 2016.

Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations, ICLR*, 2015.

Rezende, D. and Mohamed, S. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning, ICML*, pp. 1530—-1538, 2015.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems, NeurIPS*, 2016.

Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Image Synthesis with a Single (Robust) Classifier. In *Advances in Neural Information Processing Systems, NeurIPS*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations, ICLR*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision, 2015.

Tanaka, A. Discriminator optimal transport. In *Advances in Neural Information Processing Systems, NeurIPS*, volume 32, 2019.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations, ICLR*, 2019.

Turner, R., Hung, J., Frank, E., Saatchi, Y., and Yosinski, J. Metropolis-Hastings generative adversarial networks. In *International Conference on Machine Learning, ICML*, 2019.

Zhao, S., Liu, Z., Lin, J., Zhu, J., and Han, S. Differentiable augmentation for data-efficient GAN training. In *Advances in Neural Information Processing System, NeurIPS*, 2020.

Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *International Conference on Computer Vision, ICCV*, 2017.

A. Perceptually Aligned Gradient

As explained in Section 2, a common way to gauge whether a model has perceptually aligned gradients is via large-ϵ targeted adversarial attacks. If a model upholds PAG, the above process should yield perceptually meaningful modifications. As can be seen in Figure 2, while adversarially trained models have such gradients, non-robust ones do not.

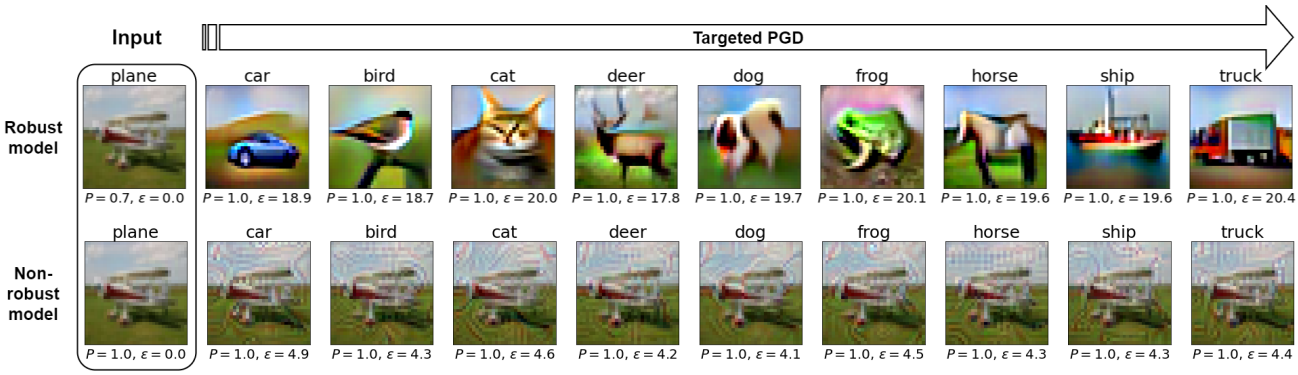


Figure 2. Visual demonstration of large l_2-based adversarial examples on robust (top) and non robust (bottom) ResNet50 classifiers (He et al., 2016) trained on CIFAR-10. The classifier's certainty and the effective ℓ_2 perturbation norm are denoted as P and ϵ, respectively. As can be seen, robust models with PAG guides the attack towards semantically meaningful features, whereas non-robust ones do not.

B. Related Work

B.1. Improving Image Generation

There are two main lines of work that aim to improve the quality of generated images. One is based on rejection sampling – improving the generation quality of GANs by discarding low-quality images, identified by the GAN's discriminator (Turner et al., 2019; Azadi et al., 2019). In contrary to such work that does not enhance the generated images but rather acts as a selector, BIGRoC does not discard any of the synthesized images and improves their quality by modifying them.

Another line of work (Tanaka, 2019; Che et al., 2021; Ansari et al., 2021), which is closely related to ours, addresses the task of sample refinement – modifying the generated images to attain improved perceptual quality. These papers propose methods for improving synthesized images using the guidance of the GAN's discriminator. More precisely, given a latent code of a generated image, their strategy is to modify it to maximize the score given by the GAN's discriminator. Therefore, to enhance the perceptual quality of a set of generated images, these approaches require access to the generator and the discriminator weights, and the corresponding latent code of the generated images. Since image refinement is an applicative task, this constraint prevents such methods from operating directly on images without additional information, making their configuration less realistic. In contrast to these, our work offers a much simpler and different way of boosting generated images by an external pretrained robust classifier in a completely model-agnostic way. Our method can operate without requiring access to the latent vectors generating the images or the weights of the generative model that produces them. Thus, BIGRoC can be applied to standalone images – a realistic setup where none of the existing methods can operate. Moreover, we show that a single robust classifier on a given dataset is capable of improving a wide variety of sample qualities, making our configuration even more realistically appealing, as it requires only one model per dataset. In addition, while (Tanaka, 2019; Che et al., 2021) are limited to GANs only, our method is model-agnostic and capable of improving generated images of any source, e.g., diffusion models. In Section 4.3 we empirically demonstrate that although the existing methods have much stricter requirements than ours, it leads to improved performance. Indeed, our approach is the first to be successfully applied to ImageNet 256×256, proving its scalability.

B.2. Perceptually Aligned Gradients (PAG) in Computer Vision

PAG phenomenon was previously utilized for solving various computer vision tasks, such as inpainting, image translation, super-resolution, and image generation (Santurkar et al., 2019). In image synthesis, they reach a 7.5 Inception Score (IS) on the CIFAR-10 dataset, far from state-of-the-art (SOTA) performance. It raises the question of whether this performance

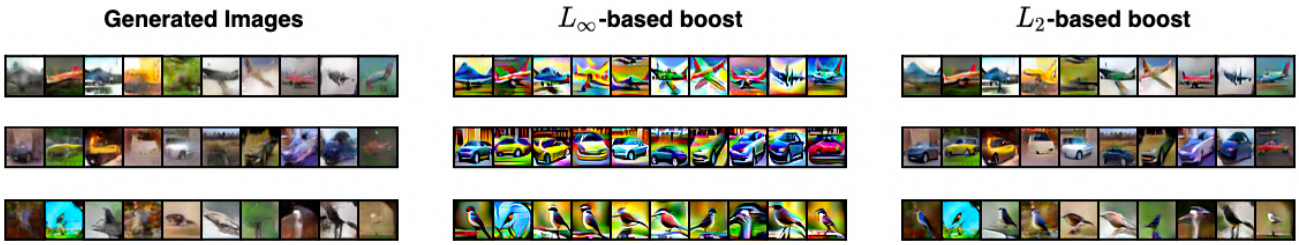


Figure 3. Effect of the threat model's choice: generated images via a conditional GAN (left) perturbed via a targeted PGD attack to maximize the probability of their target classes (planes, cars and birds) using either ℓ_∞ or ℓ_2 threat models. While the boosted outputs attained by the ℓ_∞ entirely change the structure of the images and lead to unnatural results, the ℓ_2 threat model leads to better results.

limit is due to the capabilities of robust image classifiers or is it algorithmic. In our work, we provide a definite answer to this question by harnessing PAG to a completely different task – image refinement. To this end, we build upon any existing generative model, including high-performing ones, and empirically show that a robust classifier can boost the performance of image generators well beyond 7.5 IS. As such, our work exposes a much stronger force that does exist in adversarially robust classifiers in capturing high perceptual quality features.

C. Proposed Evaluations

Other methods (Tanaka, 2019; Che et al., 2021; Ansari et al., 2021; Dhariwal & Nichol, 2021), just like ours, utilize class-conditional models (be it discriminators or classifiers) to perceptually improve image quality. This might raise a concern regarding model-based quantitative evaluation, as these are evaluated via a classifier as well. One might suspect that such methods guide the images to be "easier to classify" and thus deceptively improve the quantitative metrics without performing any solid perceptual improvement. We answer this concern in three separate ways:

- While the IS measure is more susceptible to this sensitivity, as it relies on classification, the FID metric is far more detached from the conditional model since it stems from feature distribution. Our quantitative results show that BIGRoC significantly improves both, thus relieving such concerns.
- To better validate that our improvement does not stem from the similarity of the guiding classifier with the Inception (Szegedy et al., 2015) evaluation one, we apply our algorithm using the evaluation model itself. If BIGRoC's success stems from this similarity, this experiment should yield better quantitative results than presented in the paper. We experiment with WGAN-GP, trained on CIFAR-10, which obtains an FID of 22.62. While guiding using the Inception slightly improves it to 21.78, using a robust model, as we suggest, yields an FID of 16.28. In addition, such guidance does not lead to perceptually meaningful modifications, as harnessing PAG does.
- Throughout this paper we qualitatively examine the modifications obtained by our approach and verify that they indeed lead to a perceptual improvement.

D. Ablation Study

In this section we conduct a series of experiments to further understand the performance improvements obtained, and analyze the effects of the central hyperparameters in our algorithm.

D.1. The effect of BIGRoC's threat model

In this section, we demonstrate the effect of the threat model used to perform BIGRoC's algorithm and visualize the results of this choice in Figure 3. As discussed in Section 3, L_∞ leads to unrealistic results, while L_2 leads to more pleasing images.

D.2. The effect of the robust classifier's threat model

In all of our experiments, we use an adversarially trained (AT) robust classifier with a threat model Δ based on ℓ_2 norm with a predefined ϵ value. In this section, we study the effect of ϵ in the training of the AT robust classifier on BIGRoC's

Table 6. The effects of the robust classifier's threat model on BIGRoC's performance, measured in FID, using WGAN-GP trained on CIFAR-10 dataset.

w/o BIGRoC	w/ BIGRoC			
	$\epsilon_2 = 0$	$\epsilon_2 = 0.25$	$\epsilon_2 = 0.5$	$\epsilon_2 = 1$
22.32	19.71	18.79	16.19	15.87

Table 7. The effect of the number of steps in BIGRoC's algorithm, measured in FID, using WGAN-GP trained on CIFAR-10 dataset.

w/o BIGRoC	w/ BIGRoC			
	1 step	7 steps	20 steps	30 steps
22.32	17.48	16.19	16.15	16.27

performance. In Table 6 we compare the influence of using a non-adversarial classifier (i.e. $\epsilon = 0$) and adversarial classifiers trained with different threat models' sizes on our proposed method, while the rest of the hyperparameters are fixed. As can be seen, using AT classifiers in BIGRoC enhances the results significantly.

D.3. The effect of BIGRoC's ϵ size

As stated in Section 3, the hyperparameter ϵ has a significant effect on our proposed method, since it defines the allowed perturbation. In Figure 4, we demonstrate the effect of ϵ when applying BIGRoC$_{\text{GT}}$ over images generated by guided diffusion, trained on ImageNet 256×256. As can be seen, ϵ affects the trade-off between diversity and fidelity, demonstrated by IS versus FID values. Our method attains lower FID scores in a range of tested values of ϵ, while achieving better IS, leading to much better trade-offs.

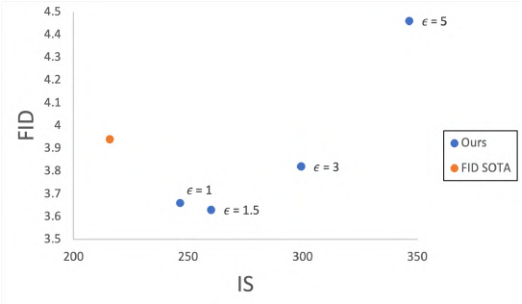


Figure 4. Trade-off effects of BIGRoC's ϵ on ImageNet 256×256.

D.4. Effect of BIGRoC's number of steps

When ϵ is fixed, increasing the number of steps leads to smaller steps. Fine-grained steps can lead to better performance, but with a computational cost. In Table 7, we summarize the effect of the number of steps w.r.t a fixed ϵ. As can be concluded, 7 is a plausible value since it is a good trade-off between refinement performance and computational cost.

D.5. Visual demonstration of BIGRoC's iterations

BIGRoC is an iterative boosting algorithm, and as such, it performs several update steps. In Figure 5, we visualize the optimization algorithm performed by our method. As can be seen, the perceptual quality of the images obtained by BIGRoC gradually improves during its application.

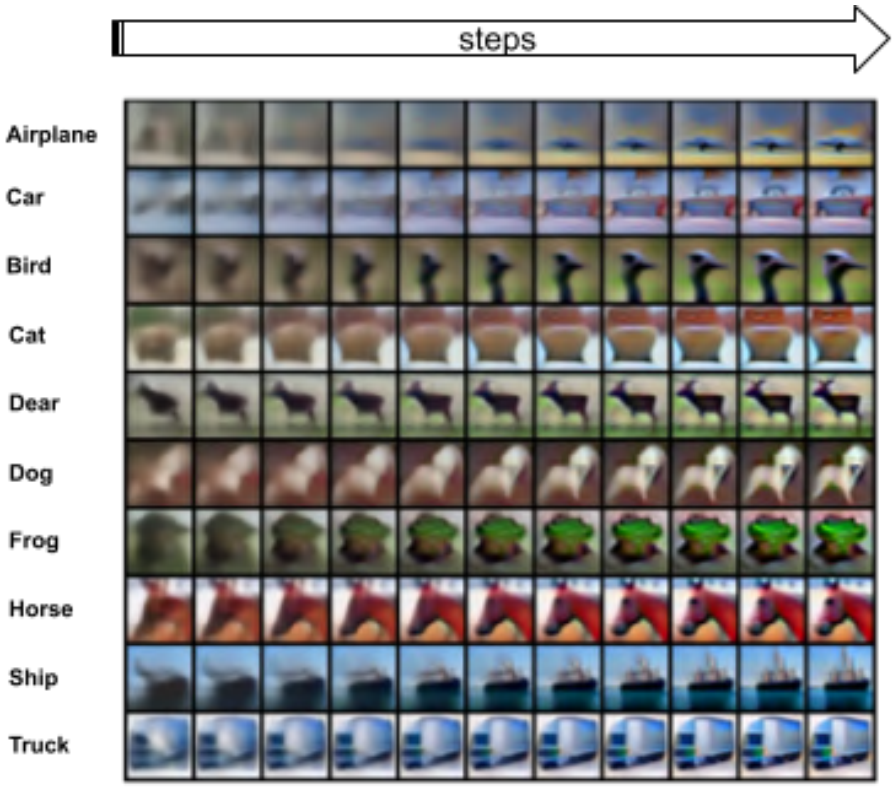


Figure 5. Demonstration of BIGRoC's steps, demonstrated using VAE trained on CIFAR-10: On the left column we show the generated images of the model. The righmost column corresponds with the final output of BIGRoC. The middle 9 columns are the results obtained after each intermediate step of our algorithm.

E. Implementation Details

In this work, except for three basic generative models (see description in E.1), we did not train models and used only available pretrained ones from verified sources. For quantitative evaluation, we use common implementations of IS and FID metrics. In all of our experiments, we use the same robust classifier to boost all the generative models that operate on the same dataset. We use the pretrained generators to synthesize sets of images and BIGRoC to refine them. The specific details regarding the experimental results are listed in the sections below.

E.1. CIFAR-10

Adversarial Robust Classifier We use a pretrained robust ResNet-50 on CIFAR-10, provided in (Engstrom et al., 2019). This model is adversarially trained with a threat model $\Delta = \{\delta \ : \ \|\delta\|_2 \leq 0.5\}$ with step size of 0.1.
Image Generators Besides VAE, DCGAN, and cGAN that we trained from scratch, using the relevant available codebases, we did not train any other generator and used only publicly available ones. For cGAN-PD, WGAN-GP, SNGAN, InfoMax-GAN, and SSGAN, we use the ones from mimicry repository[1]. We use the pretrained versions of BigGAN and Differential Augmentation CR BigGAN (Diff BigGAN) from data-efficient GANs repository[2].

BIGRoC hyperparameters As stated above, we use the same robust classifier in all our experiments. Thus, the remaining hyperparameters to tune are ϵ, step size, and the number of steps. We empirically find out that the number of steps is relatively marginal (see Section D.4), and thus we opt to use seven steps. In all of the experiments, we fix the step size to be $1.5 * \frac{\epsilon}{num\ steps}$. Since we test various image generators of different qualities, each requires a particular amount of visual enhancement, defined by the value of ϵ. Low-quality generators require substantial improvement and therefore benefit from

[1] https://github.com/kwotsin/mimicry
[2] https://github.com/mit-han-lab/data-efficient-gans/tree/master/DiffAugment-biggan-cifar

high ϵ values, while better ones benefit from smaller values. In the below table, we summarize the value of ϵ for each tested architecture.

Table 8. BIGRoC's ϵ values in CIFAR-10 experiments

ARCHITECTURE	ϵ	NORMALIZATION
VAE	25	✓
DCGAN	5	✓
WGAN-GP	5	✓
SNGAN	3	✓
SSGAN	3	✓
CGAN	5	✓
CGAN-PD	2	✓
BIGGAN	1	✓
DIFF BIGGAN	1	✓

Where normalization is referred to images in [-1, 1]. To better interpret the meaning of ϵ in terms of pixels modification, the average change of a pixel value is expressed by $\frac{\epsilon}{\sqrt{32\times32\times3}}$. For example, in DCGAN, $\epsilon = 5$ is equivalent to an average change of ≈ 0.1. We note that in this example, the pixels are in a range between -1 to 1 and therefore, the mean change is $\approx 5\%$

E.2. ImageNet

Adversarial Robust Classifier We use a pretrained robust ResNet-50 on ImageNet, provided in (Engstrom et al., 2019), on both 128×128 and 256×256. This model is adversarially trained with a threat model $\Delta = \{\delta : \|\delta\|_2 \leq 3\}$ with step size of 0.5.

Image Generators We did not train any generator and utilize the publicly available ones. For SNGAN, SSGAN, and InfoMaxGAN, we use the ones from the mimicry repository. As for BigGAN-deep (truncation$= 1.0$) and guided diffusion, we utilize the fact that BIGRoC can operate on standalone images and utilize the sets of generated images, published in guided diffusion's repository[3], and we apply our method upon these. We test our method using the aforementioned sets of generated images using two setups – with and without the ground truth labels. When operating without the labels, we produce pseudo labels using our robust classifier and then apply BIGRoC.

BIGRoC hyperparameters As in CIFAR-10, we only tune the value of ϵ. In the below table, we report the used values for our tested architectures.

Table 9. BIGRoC's ϵ values in ImageNet experiments

RESOLUTION	ARCHITECTURE	ϵ	NORMALIZATION
128	SNGAN	40	✓
	SSGAN	40	✓
	INFOMAXGAN	40	✓
	BIGGAN-DEEP	5	✗
	GUIDED DIFFUSION	1.5	✗
256	BIGGAN-DEEP	1	✗
	GUIDED DIFFUSION	1.5	✗

E.3. Comparison with other methods

Adversarial Robust Classifier For CIFAR-10 we use the same robust classifier as described in Appendix E.1 and for ImageNet we use the same model as in Appendix E.2.

Image Generators To fairly compare between BIGRoC and the competitive methods, we use BIGRoC to refine the outputs

[3] https://github.com/openai/guided-diffusion

of the same pretrained model as in DOT, DDLS, and DG-*f*low – SN-ResNet-GAN[4]. We experiment in both CIFAR-10 and ImageNet and compare our results with the reported ones of the other methods. The missing values in Table 5 stem from the fact that DOT did not report its results on SN-ResNet-GAN on CIFAR-10, and DG-*f*low was not tested on ImageNet or any other high-resolution dataset.

BIGRoC hyperparameters We report in the table below the value of epsilon used to attain the results in Table 5.

Table 10. BIGRoC's ϵ values in Table 5 experiments

DATASET	ϵ	NORMALIZATION
CIFAR-10	1.8	✗
IMAGENET	15	✗

F. Qualitative Results

In this section, we show additional qualitative results to further demonstrate the qualitative enhancement attained by our method. In Figure 6 we visualize the modifications done by BIGRoC on the outputs of a guided diffusion on ImageNet 256×256, which strongly demonstrates that our approach focuses on boosting meaningull perceptual features.

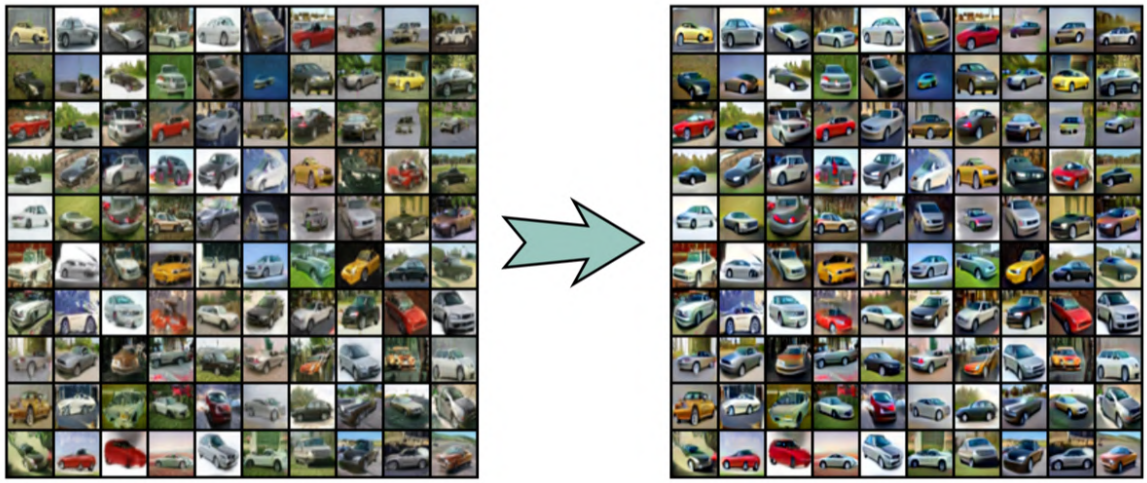
Figure 6. Qualitative comparison on ImageNet 256×256. left: Images generated by guided diffusion. Middle: Images boosted by BIGRoC$_{\text{PL}}$. Right: The difference after contrast stretching.

In addition, we provide additional qualitative visualization of the effects of BIGRoC. We use image generators of different qualities, both conditional and unconditional, and show the generated images and the boosted ones in Figures 7, 8, 9 and 10. The images below are simply the 100 first synthesized images from each class.

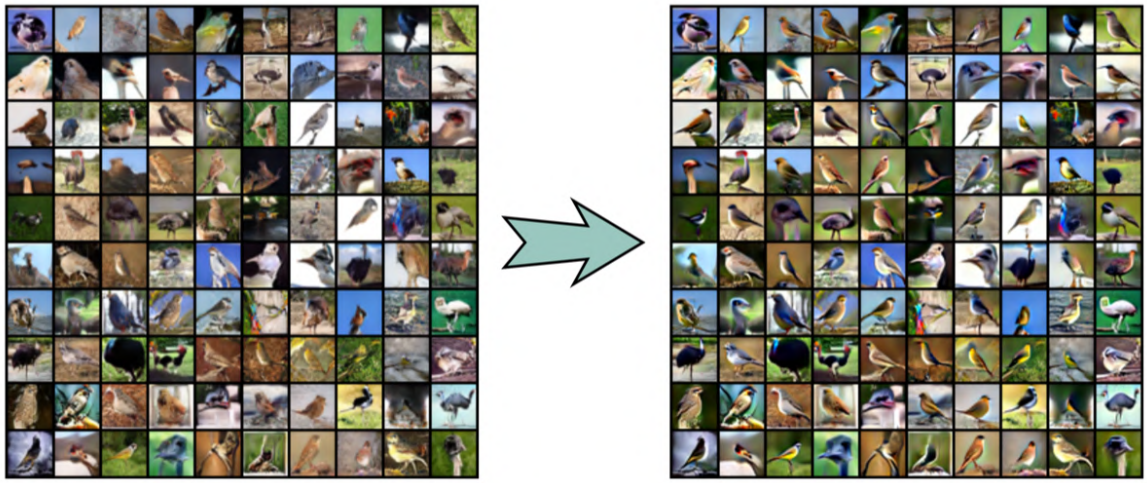
[4] https://github.com/pfnet-research/sngan_projection

G. Debiasing

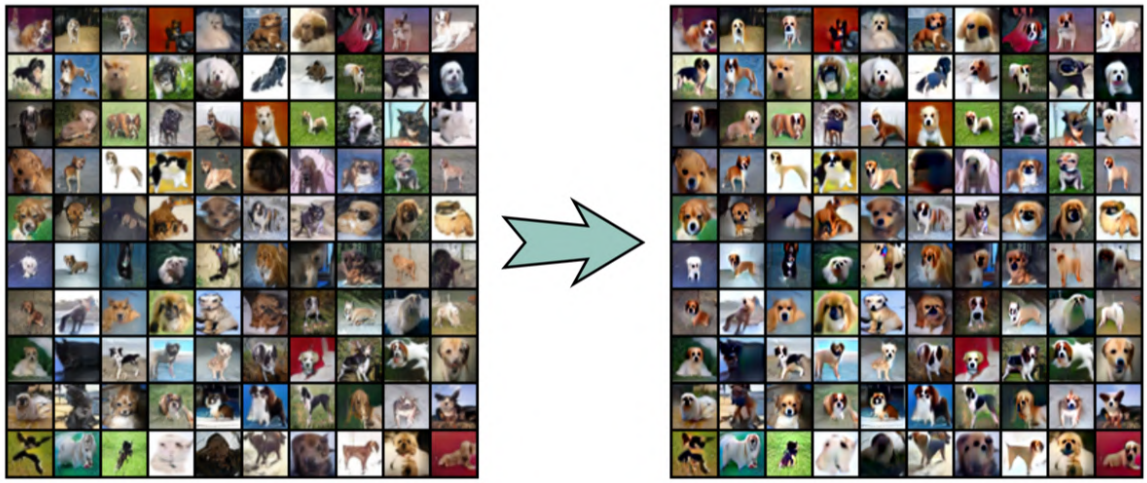
In Section 3 we describe our debiasing algorithm, which aims to induce uniform class distribution over the outputs of BIGRoC. The rationale behind this procedure is expressed in Equation 2. To better understand the outcome of our debiasing, we present in Figure 11 its effect visually. We compare the results of applying BIGRoC in the unconditional case, with and without debiasing. One can clearly see that it reduces the amount of the majority class and leads to a more uniform class distribution, by modifying the images accordingly.



(a) A comparison between BigGAN generated images of class automobile and the corresponding boosted ones.



(b) A comparison between BigGAN generated images of class bird and the corresponding boosted ones.



(c) A comparison between BigGAN generated images of class dog and the corresponding boosted ones.

Figure 7. A qualitative comparison between BigGAN generated images of CIFAR-10 samples and the proposed BIGRoC algorithm.

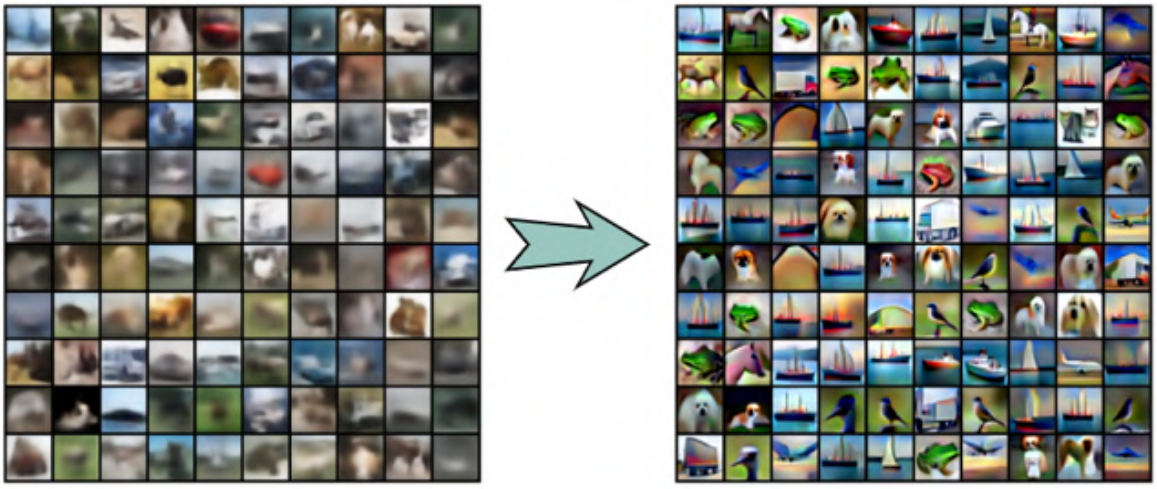


Figure 8. A qualitative comparison between an unconditional VAE generated images of CIFAR-10 samples and the proposed BIGRoC algorithm.

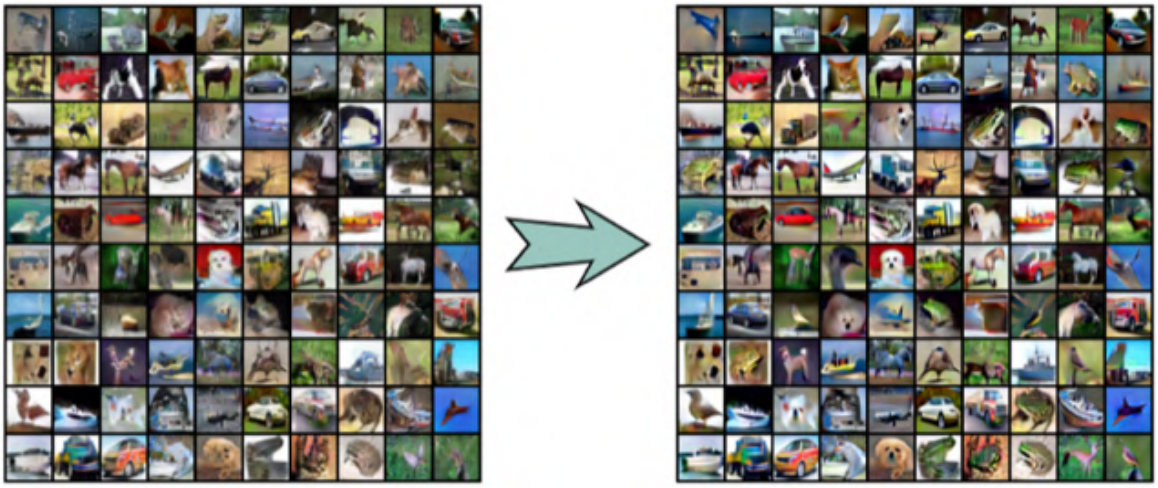


Figure 9. A qualitative comparison between an unconditional WGAN-GP generated images of CIFAR-10 samples and the proposed BIGRoC algorithm.

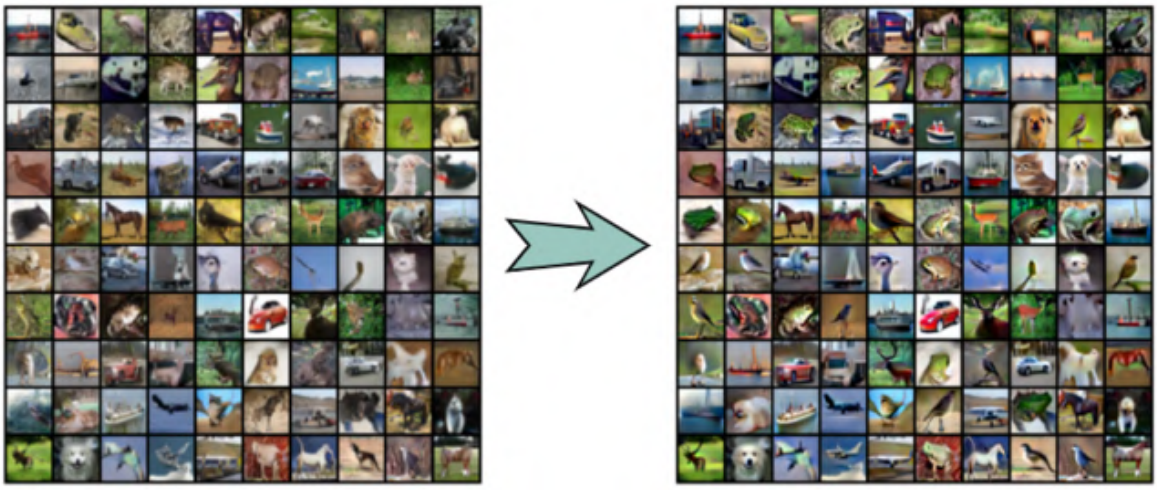


Figure 10. A qualitative comparison between an unconditional SSGAN generated images of CIFAR-10 samples and the proposed BIGRoC algorithm.

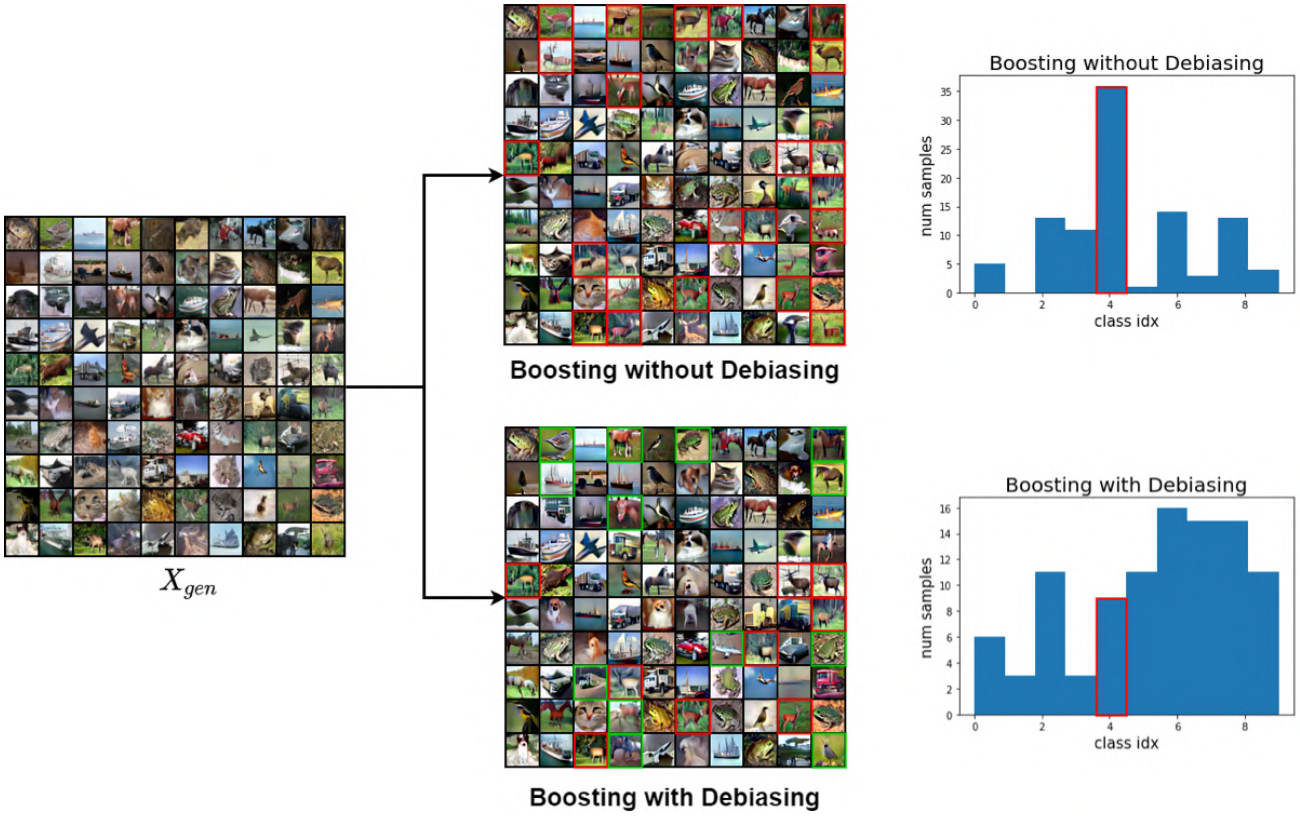


Figure 11. Demonstration of the debiasing technique: We show 100 generated images by an unconditional SNGAN and the results of the BIGRoC algorithm, with and without the proposed debiasing. As can be seen, the outputs of the boosting algorithms are perceptually superior, while the histograms expose the fact that the suggested debiasing algorithm induces a more uniform labels distribution. In the "Boosting without Debiasing" experiment, 36 out of 100 images are classified as deers, and only 3 are horses. The most prominent deer images are marked in red. However, when applying the debiased boosting, the number of deers is reduced to 9, and the number of horses is increased to 15. We mark the boosted images that remain deer in red, and images that are modified to other minority classes in green. As can be seen, many of the deers were changed to be horses, a perceptually similar class.