This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

IMAGINE: Image Synthesis by Image-Guided Model Inversion

Pei Wang*
UC, San Diego
pew062@ucsd.edu

Yijun Li, Krishna Kumar Singh, Jingwan Lu
Adobe Research
{yijli, krishsin, jlu}@adobe.com

Nuno Vasconcelos
UC, San Diego
nuno@ucsd.edu

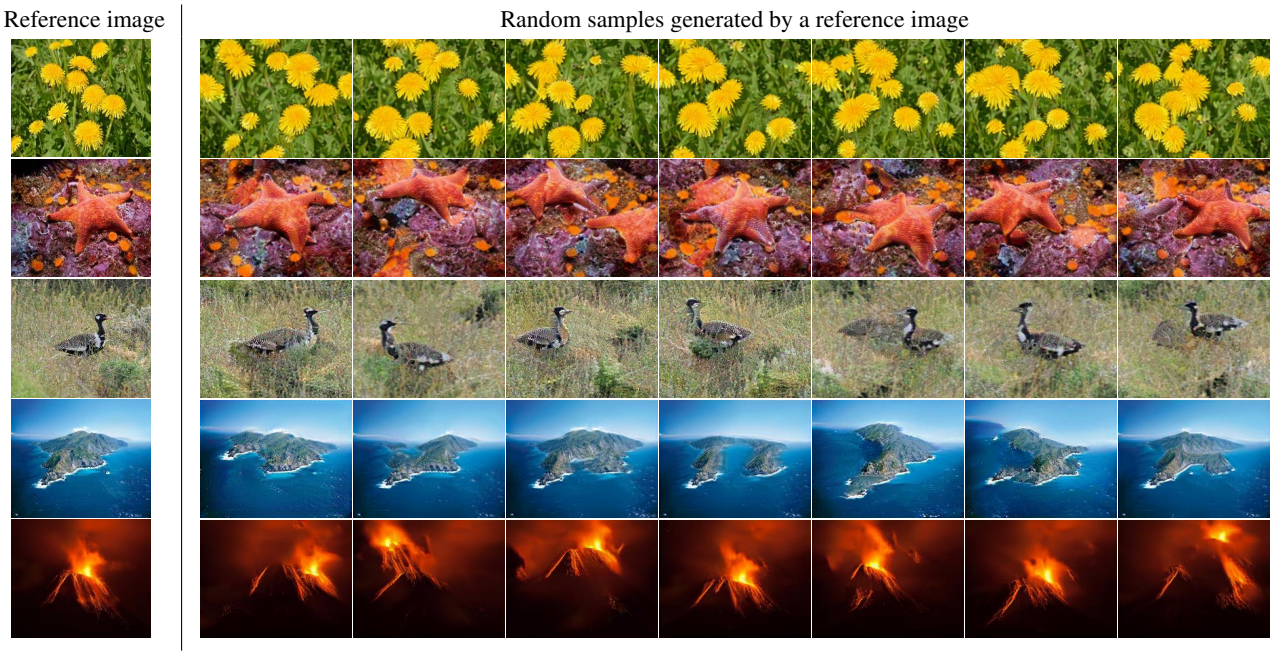


Figure 1: Results from our proposed IMAGINE - the first attempt to use classifier inversion for reference-guided image generation.

Abstract

We introduce an inversion based method, denoted as IMAge-Guided model INvErsion (IMAGINE), to generate high-quality and diverse images from only a single training sample. We leverage the knowledge of image semantics from a pre-trained classifier to achieve plausible generations via matching multi-level feature representations in the classifier, associated with adversarial training with an external discriminator. IMAGINE enables the synthesis procedure to simultaneously 1) enforce semantic specificity constraints during the synthesis, 2) produce realistic images without generator training, and 3) give users intuitive control over the generation process. With extensive experimental results, we demonstrate qualitatively and quantitatively that IMAGINE performs favorably against state-of-the-art GAN-based and inversion-based methods, across three different image domains (i.e., objects, scenes, and textures).

1. Introduction

Consider a reference image from the left-most column of Fig. 1, we humans could easily *imagine* its variants in other formats, e.g., patterns with different spatial arrangements, objects in different positions or viewpoints. Can we give machines such abilities to automatically synthesize semantically meaningful variations of a reference image? Such a system can generate interesting variations of assets for design ideations and can also be utilized as a data augmentation technique to benefit downstream data-hungry tasks.

With the popularity of generative adversarial networks

*Work done during internship at Adobe Research

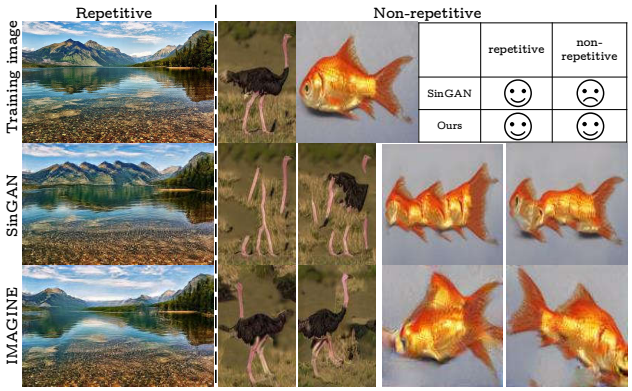


Figure 2: Image generation by SinGAN [29] and our IMAGINE. SinGAN only works well on repetitive images that contain repetitive structures such as mountains, but fails on non-repetitive images containing large semantic structures such as the ostrich and the goldfish. IMAGINE can generate higher quality and more diverse images on both types.

(GANs) [12, 3, 21, 24, 6], two categories of GAN-based approaches have been proposed for editing a specific image and/or synthesizing its variations. The first one is based on GAN projection (or inversion) [28, 32, 5, 16], which consists of projecting the target image into the latent space of the generator, jittering the projected latent code, and synthesizing a new image. However, there are several limitations to this type of approach. First, the projection step is far from accurate often resulting in unrealistic images due to the projected latent code falling outside of the learning distribution [32]. Second, the projection step can only handle the type of images that the generator is trained on. Since existing GANs are trained on specific domains, such as faces, or datasets containing a few classes [21, 32, 27, 47, 4, 16], the projection-based approaches cannot generalize. The second category is to learn individual GANs for individual target images, e.g., SinGAN [29]. SinGAN leverages image patches from the target image to learn image internal statistics. While enforcing patch consistency across resolutions enables the synthesis of multi-scale structures, SinGAN is best suited for the synthesis of repetitive images, like the mountain in Fig. 2. However, as the patch-based learning is less effective in capturing high-level semantics, it struggles to synthesize images of non-repetitive objects. As shown in the second row of Fig. 2, properties like object identity, shape, or consistency among parts are lost in their results.

This problem can be solved by model inversion of a pre-trained deep classifier which can capture multi-scale abstract features. Model inversion seeks to determine the image responsible for a particular classifier prediction [23, 43, 26, 42][1]. Given the prediction, an optimization is performed to synthesize the image that best explains it. Although several works have shown that introducing strong regularizers [23, 26, 42] can improve the quality of synthesized images, the quality is still not comparable to GANs. More importantly, all these methods suffer from *lack of specificity*. They simply generate random images with the appearance characteristics of the target class, but they cannot synthesize variations of a specific image.

In this work, to overcome the limitations of these previous methods, we introduce the concept of *IMAge-Guided model INvErsion* (IMAGINE). IMAGINE leverages the knowledge of image semantics from a pre-trained *classifier* (e.g., ResNet-50 pre-trained on ImageNet [8]) to generate semantically meaningful variations of a target. Specifically, we feed the target image into the classifier and optimize for a new image that best explains its class prediction, under the constraint that the features of the synthesized image must match those of the target at various network layers. This is then complemented by the adversarial training to further improve the realism of the synthesized images. However, unlike GANs, our adversarial training is conducted by alternating the optimization of the synthesized image and that of the discriminator weights.

The resulting algorithm enables the synthesis procedure to simultaneously 1) enforce semantic constraints during the synthesis, 2) produce realistic images, without the need for a separate generator training, and 3) allow multifaceted control over the image to be synthesized. When compared to the SinGAN, it captures much more abstract high-level image semantics, such as object shape or identity. This enables the successful synthesis of non-repetitive images, such as the objects of Fig. 1 and 2. Because the feature matching constraints are based on the matching of distributions, rather than minimizing geometric distances between feature maps [28, 34], IMAGINE leads to more visual variations in the synthesized results (Fig. 1). We show that, by manipulating the target feature statistics at the different levels of the network, it is possible to recreate objects with modified semantics, such as different shapes (Fig. 7). Furthermore, through the use of attribution functions [31, 37], it is even possible to manipulate the location of the target objects in the synthesized image (Fig. 6). Several other applications of IMAGINE are also discussed.

Our paper makes four main contributions. (1) To the best of our knowledge, this is the first attempt to utilize model inversion for image-guided synthesis problem. (2) We introduce adversarial training to improve the image quality of optimization-based model inversion. (3) We perform extensive quantitative and qualitative evaluations to demonstrate the superiority of IMAGINE over GAN counterparts and its generalizability across different image domains including single objects in front of simple backgrounds, complex scenes, and non-stationary textures. (4) We show multiple multifaceted image control applications including a newly

[1] In the remainder of the paper, we use "model inversion" to refer to "classifier inversion".

proposed object position control task.

2. Related Work

Model inversion. The goal of model inversion in vision tasks is usually to synthesize an image that maximizes the likelihood of either a network prediction or the response of some intermediate network unit. This includes the generation for adversarial attacks [10, 13], where a real image is used to initialize the optimization and the latter only seeks a prediction that is imperceptible under some constraints [38, 36]. Another application would be explainable AI, where the image is initialized with noise and the goal is to determine what image structures elicit different network predictions [35]. Several works in the literature have shown that the addition of different regularizers enables the synthesis of realistic images [23, 26, 25]. Recently, DeepInversion [42] used a regularizer based on feature matching to improve synthesis quality and showed its benefits for data-free knowledge distillation and other high-level vision tasks. These methods either produce images indistinguishable from the target image (attacks) or suffer from lack of specificity, i.e. synthesized images do not look similar to a target image. Our approach allows us to generate different variants of target image which are non-identical to target image and look real at the same time.

Image-guided image generation. There are several types of image-guided synthesis techniques. The most popular is GAN-inversion that uses a target image to guide the sampling of a trained GAN, attempting to sample similar images by inverting a pre-trained GAN generator [1, 28, 32, 5, 16]. They first project the target image into the latent space of the generator by minimizing certain reconstruction losses, then jitter or edit this projection so as to synthesize a new image. However, generator inversion is a difficult problem. It is hard to guarantee that the latent space projection of the target falls within the training distribution and find a semantic jittering direction. Although some improvements have been proposed with better reconstruction quality including finetuning the generator during projection [28], using multiple latent codes [16], these literature only work on specific domains such as faces or datasets of a few classes [1, 32, 47], or produce inadequate results [28], since capturing the distribution of highly diverse datasets with multiple object classes is still a challenging problem. Finetuning the generator is also not space-efficient because it needs a specific GAN trained for each target image in order to work well. A more promising alternative is to train a generator specifically for image-guided synthesis. For example, [34] proposes to feed the target image to a pre-trained classifier and introduces the resulting features into the layers of a generator, trained to simultaneously replicate the classifier features and produce realistic images. While improving on GAN inversion, these methods still have significant limitations. Training a GAN to support the synthesis of images from many classes is still a difficult problem [6], in particular when those are relatively fine grained and require representations with multiple semantic levels, such as ImageNet [8].

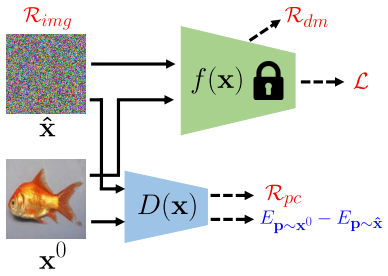


Figure 3: The framework of IMAGINE. The red characters represent losses for the input noise whereas blue for the discriminator. Bold arrows mean the data flow and dotted arrows loss computation.

Recently, some work [29, 33] propose to bypass these difficulties by leveraging internal statistics of patches within the image to train a patch-based GAN from scratch. These methods can be image conditional, i.e. take an image as input, in which case they are best suited for image processing operations, such as upsampling or retargeting [33], or unconditional, i.e. take noise as input, in which case they can perform image-guided synthesis [29]. Nevertheless, due to the patch-level learning, they have limited ability to capture high-level semantics, such as object parts and consistency among them, e.g., that 'an ostrich has two legs and a neck that are connected by a torso.' As a result, they have limited ability to synthesize images that lack self-similar repetitive structure, such as the objects of Fig. 2. Pyramid model of [29] fixes the overall structure of images but loses variations. By leveraging a deep representation with many levels of semantic abstraction not just patch-level, our image-guided model inversion offers a natural solution to this problem.

3. Proposed Method

In this section, we first describe existing model inversion work for image synthesis and their limitations. After that, we introduce our image-guided model inversion work for image synthesis which overcomes the limitation of previous approaches.

3.1. Image Synthesis by Model Inversion

A classifier $f(\mathbf{x})$ maps images $\mathbf{x} \in \mathcal{X}$ into classes $y \in \mathcal{Y} = \{1, ..., C\}$, according to $y = \arg\max_{\hat{y}} f_{\hat{y}}(\mathbf{x})$. In this work, we consider classifiers implemented with convolutional neural networks (CNNs). Model inversion aims to synthesize an image $\mathbf{x}$ that elicits a class response y^* from

the classifier. This is formulated as

$$\mathbf{x}^* = \arg\min_{\hat{\mathbf{x}}} \mathcal{L}(f(\hat{\mathbf{x}}), y^*) + \mathcal{R}(\hat{\mathbf{x}}), \quad (1)$$

where $\mathcal{L}$ is the cross-entropy loss, $\mathcal{R}$ a regularization term, and $\hat{\mathbf{x}}$ is initialized randomly.

Model inversion has long been used in the explainable AI literature, to visualize the patterns most representative of particular class predictions [23, 35, 43, 26]. These works have shown that the addition of the regularizer $\mathcal{R}(\hat{\mathbf{x}})$ is important to avoid highly non-realistic images. A popular choice is to combine the total variance (TV) loss [14] and the L_2 norm of $\mathbf{x}$ to steer $\mathbf{x}$ away from unrealistic images with no discernible visual information [42], i.e. use

$$\mathcal{R}_{img}(\hat{\mathbf{x}}) = \alpha R_{TV}(\hat{\mathbf{x}}) + \beta ||\hat{\mathbf{x}}||^2, \quad (2)$$

where α and β are scaling factors.

Recently, [42] has shown that realism can be substantially improved by penalizing the distance between the network activations in response to $\hat{\mathbf{x}}$ and the average activations over the data used to train the network. This is denoted as a feature distribution regularizer

$$\mathcal{R}_{feat}(\hat{\mathbf{x}}) = \sum_l ||\mu_l(\hat{\mathbf{x}}) - <\mu_l>_{\mathcal{D}}||^2 + \sum_l ||\sigma_l^2(\hat{\mathbf{x}}) - <\sigma_l^2>_{\mathcal{D}}||^2, \quad (3)$$

where $\mu_l(\hat{\mathbf{x}})$ and $\sigma_l^2(\hat{\mathbf{x}})$ are the vectors of feature means and variances at network layer l, for image $\hat{\mathbf{x}}$, and $<\mu_l>_{\mathcal{D}}$ and $<\sigma_l^2>_{\mathcal{D}}$ the corresponding averages over the dataset $\mathcal{D}$ used to train the network.

While improving realism, these methods suffer from *lack of specificity*. Although it can synthesize images of a particular class, it is impossible to synthesize image variations of a particular image. This problem has been considered in the GAN literature by inverting a pre-trained generator G. Given a target image $\mathbf{x}^0$, this consists of solving

$$\mathbf{z}^* = \arg\min_{\mathbf{z} \in \mathcal{R}^d} \mathcal{M}(\mathbf{x}^0, G(\mathbf{z})), \quad (4)$$

where $\mathcal{M}$ is a distance metric and $\mathbf{z}^*$ is the projection of $\mathbf{x}^0$ on the latent space. New images can then be generated with $\mathbf{x}^* = G(\mathbf{z}^* + \delta)$, where δ is a noise term. Although it overcomes lack of specificity, it has two problems. First, the solution of (4) is not trivial. Simple metrics $\mathcal{M}$, such as the L_2 or perceptual loss [20], favour similarity in terms of low-level statistics, such as color and texture, over more abstract features, such as shape or object identity. A second problem is that GANs are still too difficult to train for large scale datasets involving many classes, such as ImageNet, where mode collapse and class leakage become exponentially more difficult [6]. Hence most models in the literature are trained for specific image domains, such as faces or datasets of a few classes [1, 32], and do not support the synthesis beyond these domains. [28] proposed to solve this problem by finetuning the generator G to the target image, but this fails to guarantee realistic images because the optimization process is difficult and non-trivial (see Fig. 4).

Single image based GANs [33, 29] offer an alternative to this problem. In the image-guided synthesis context, the SinGAN [29] uses the target image $\mathbf{x}^0$ to learn a patch generator from scratch. To guarantee image consistency, both generator and discriminator operate on patches of multiple resolutions, using a pyramid structure where each stage improves on the synthesis by a stage of lower resolution. While the patch-based formulation bypasses the problem of training data scarcity, it has a few drawbacks. First, SinGAN needs to train a specific generator for each target image, which is not space-efficient. This is also true for GAN finetuning method [28]. Second, while the multi-resolution structure constrains the organization of patches at different scales, these constraints are of low-level: the network mostly learns to tile and replicate patches at multiple scales to resemble the target image. As a result, it does not necessarily learn the high-level semantic features needed to represent object shape or long-range correlations of such features, e.g. that "an ostrich has two legs and a neck connected by a torso" or that a "fish has an head and a tail at opposite ends of the body." As shown in Fig. 2 and 4, while the SinGAN is very effective for repetitive images (characterized by highly self-similar regions of different resolutions), it cannot synthesize visual concepts like object that require learning high-level semantic features capturing shape and relationships between object parts. We next introduce a new model inversion technique that aims to solve these problems.

3.2. Image Guided Model Inversion

To avoid the difficulties discussed above for GANs, we consider image synthesis by inversion of an image classifier. This immediately guarantees large models that encompass many classes (1k for ImageNet) and representations with multiple levels of abstraction, learned from very large datasets. However, as we saw earlier that image synthesis by inversion of an image classifier would have issues with specificity and realism. Hence, we do image guided inversion to improve these two aspects of the synthesized image. In practice, we complement $\mathcal{R}$ in (2) with two *image-guided regularizers*.

The first is a regularizer inspired by (3). Like (3), we aim to preserve similarity between target ($\mathbf{x}^0$) and synthesized image at all levels of semantics, from class identity, to shape, contours, texture, and color. This is implemented by matching feature statistics at multiple network layers. However, to avoid lack of specificity, we seek a regularizer based on the target image alone. The goal is met by the feature

distribution matching regularizer

$$\mathcal{R}_{dm}(\hat{\mathbf{x}};\mathbf{x}^0,\Phi)=\sum_{l\in\Phi}||\mu_l(\hat{\mathbf{x}})-\mu_l(\mathbf{x}^0)||_2+||\sigma_l(\hat{\mathbf{x}})-\sigma_l(\mathbf{x}^0)||_2, \tag{5}$$

where $\mu_l(\mathbf{x})$ and $\sigma_l(\mathbf{x})$ are the channel-wise mean and standard deviation of feature maps at the lth network layer, Φ the sets of layers included in the summation.

While this regularizer enforces semantic consistency, enabling the synthesis of object-like structures, it is not sufficient to synthesize realistic images. The fine levels of image detail are usually not reproduced with enough quality to create a realistic image. To address this problem, inspired by SinGAN, we train a patch-based discriminator D [22, 19] adversarially during image synthesis. Given synthesized image $\hat{\mathbf{x}}$, a patch classifier d^* is trained to discriminate between $\hat{\mathbf{x}}$ and the target $\mathbf{x}^0$, using a Wasserstein loss

$$d^*=\arg\max_d E_{\mathbf{p}\sim\mathbf{x}^0}[d(\mathbf{p})]-E_{\mathbf{p}\sim\hat{\mathbf{x}}}[d(\mathbf{p})], \tag{6}$$

where $\mathbf{p}$ denotes an image patch. The image discriminator is then implemented as

$$D(\hat{\mathbf{x}})=E_{\mathbf{p}\sim\hat{\mathbf{x}}}[d^*(\mathbf{p})]. \tag{7}$$

For improving training stability we use a WGAN-GP loss [17] instead of (6), but omit the details for brevity. A patch consistency regularizer

$$\mathcal{R}_{pc}(\hat{\mathbf{x}})=-D(\hat{\mathbf{x}}) \tag{8}$$

is then introduced in the optimization of (1). This encourages images that cannot be discriminated from $\mathbf{x}^0$ in terms of the visual consistency of their patches.

Finally, IMAGINE solves the optimization of (1) with regularizer

$$\mathcal{R}(\hat{\mathbf{x}};\mathbf{x}^0,\Phi)=\mathcal{R}_{img}(\hat{\mathbf{x}})+\lambda\mathcal{R}_{dm}(\hat{\mathbf{x}};\mathbf{x}^0,\Phi)+\gamma\mathcal{R}_{pc}(\hat{\mathbf{x}}). \tag{9}$$

where λ and γ are scaling factors.

3.3. Implementation Details

The overall framework is illustrated in Fig. 3. In all cases, the loss of (5) was computed on four layers: $\{conv1_1, conv2_3, conv3_4, conv4_6\}$ for ResNet-50 models, $y^*=\arg\max_{\hat{y}} f_{\hat{y}}(\mathbf{x}^0)$ in (1), by default unless otherwise noted. The detailed structure of the discriminator D is discussed in the supplementary material. Adam was used for all optimizations (learning rate 0.2 for that of $\hat{\mathbf{x}}$ and $5e-4$ for the discriminator). Except where otherwise stated, images were synthesized with 224×224 pixels.

For all scaling weights involved in (9), α and β of $\mathcal{R}_{img}$ in (1) are set to $1e-4$ and $1e-5$, respectively, following [42]. For λ, we have found that the overall performance is stable as long as $\lambda>1$. All results reported in this work were obtained with $\lambda=5$. The bulk of the complexity of this optimization is in the training of the discriminator used by $\mathcal{R}_{pc}$. We have found that a much more efficient solution is to use a two-stage optimization. In the first stage, γ is set to zero, i.e. only the feature distribution matching regularizer is used. The discriminator $D(\mathbf{x})$ is then introduced in the second stage ($\gamma=10$), to refine the synthesized image by adversarial training. The first stage can be seen as a warm-up stage that provides a good initialization, which is refined by the second. Under this approach, training time is still dominated by the optimization of the patch-based adversarial loss in the second stage. However, the availability of a good initialization significantly reduces the convergence time of this optimization and, consequently, of the overall image synthesis. In experiments, 2000 iterations are assigned for each stage. We have also tried a single stage optimization with non-zero λ and γ but this did not have a noticeable effect on the quality of the synthesized images and takes much more time.

4. Experiments and Applications

We extensively evaluate the proposed IMAGINE on different types of images against existing approaches and discuss some novel and interesting applications of our method.

4.1. Image Synthesis

IMAGINE is a new paradigm for image synthesis, which fills a void in the landscape of synthesis techniques. To the best of our knowledge, it is the first method to support image-guided synthesis by classifier inversion. Fig. 4 compare its image synthesis performance to those of two GAN methods, DGP [28] and SinGAN [29], and an image-guided version of DeepInversion [42], implemented by complementing (5) to (3) with weight factor of 5. Just like our approach, DeepInversion method uses ResNet-50 pre-trained on ImageNet [8] and Places365 [46] for object and scene images respectively as a classifier.

The results of SinGAN, DGP, and IMAGINE are all clearly superior to those of DeepInversion. This shows the benefits of adversarial training. However, SinGAN tends to synthesize images by copying and shifting local patterns of the training image. This is effective for scenes images, e.g., "mountain path", containing fractal-like structures, like mountains, whose realism is not affected by this type of synthesis. However, for non-repetitive structures, such as objects like "brambling," "ostrich" or "fox squirrel" (shown in Fig. 4), it generates non-realistic images. In several cases, e.g., "European fire salamander" or "red-breasted merganser", the object is eliminated or "absorbed" into the background. DGP fails on most of the object images, perhaps because inverting a generator is always more difficult than a classifier, which causes searching a bad embedding in the latent space.

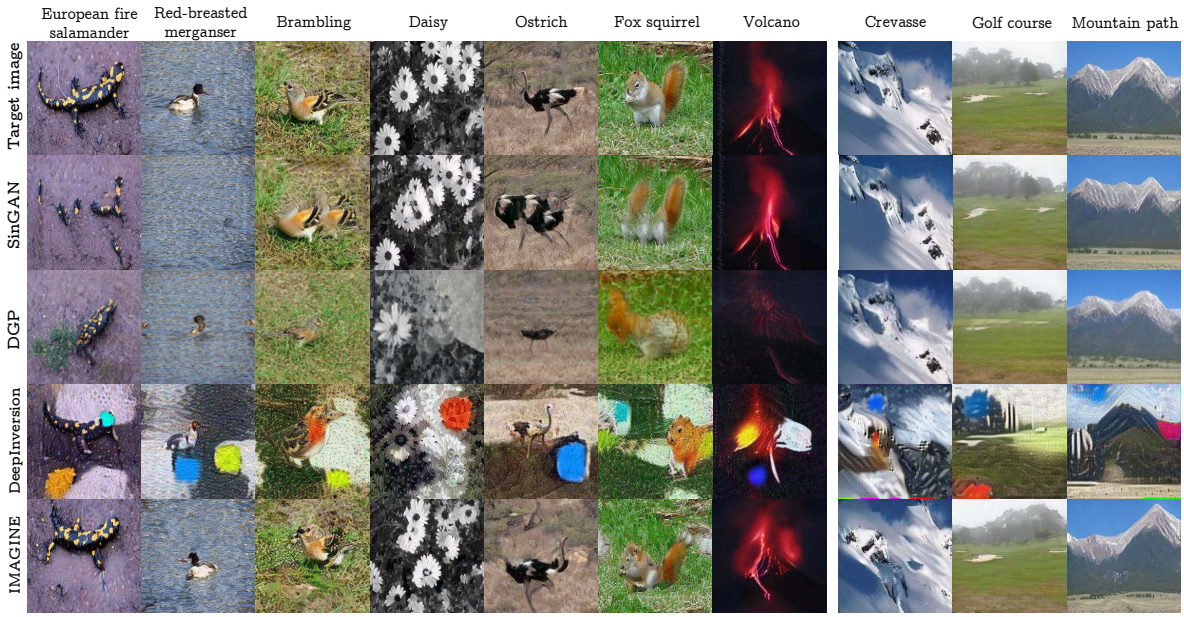


Figure 4: Image generation comparison of different methods (object images on left and scene images on right).

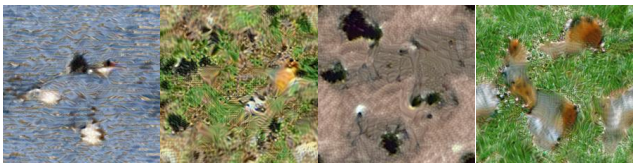


Figure 5: Results when we suppress the contribution of deeper layer in (5). From left to right corresponding to "red-breasted merganser", "brambling", "ostrich" and "fox squirrel" in Fig. 4.

IMAGINE is not affected by these problems, because the regularizer of (5) encourages the synthesis of images with activation statistics close to those of the training image *throughout* the network. This includes the activations of deeper layers, which reflect the image content in terms of higher level semantics, such as object shape and identity. This is illustrated by Fig. 5, where the regularizer of (5) was restricted only to the three earliest network layers of the four used in Fig. 4. Similarly to the SinGAN, the objects disintegrate and are absorbed by the background.

When the SinGAN successfully reconstructs an object or scene, e.g., "volcano", "golf course", it tends to reconstruct near replicas of the training image. This is partly due to its multi-scale structure, which enforces tight constraints on both global and local features. As a result, the network has almost no ability to synthesize diverse images whereas our approach can generate diverse images as it does not has multi-scale constraint and adversarial loss is applied to the final image. Beyond synthesizing realistic objects, IMAGINE is also able to modify the object pose ("bustard", "starfish"), image location ("volcano"), object contour ("island"), or even the image layout ("daisy") as illustrated in Fig. 1.

Table 1: Quantitative comparison of different methods on both object and scene images. Results are based on images with resolution of 224 and presented as mean(std).

	Object			Scene		
method	IS↑	FID↓	LPIPS↑	IS↑	FID↓	LPIPS↑
DeepInversion [42]	68.5(5.0)	65.9(2.7)	0.52(0.10)	12.3(0.5)	53.7(2.3)	0.56(0.10)
SinGAN [29]	-	-	0.24(0.11)	-	-	0.27(0.08)
DGP [28]	46.3(2.4)	46.0(1.5)	0.30(0.09)	12.7(0.6)	51.3(1.6)	0.27(0.10)
IMAGINE	117.1(6.2)	38.3(1.1)	0.46(0.09)	21.8(0.4)	47.3(0.9)	0.43(0.13)

Table 2: Comparison of user preference (%), $\text{mean}_{\text{std, confidence interval}}$ (conf. interval at 95% conf. level).

Object			
	DeepInversion/Ours	SinGAN/Ours	DGP/Ours
realism	$1.0/\mathbf{99.0}_{2.0,1.4}$	$29.5/\mathbf{70.5}_{10.3,6.0}$	$26.0/\mathbf{74.0}_{8.3,6.1}$
diversity	$26.5/\mathbf{73.5}_{17.5,6.7}$	$17.5/\mathbf{82.5}_{6.8,5.3}$	$28.5/\mathbf{71.5}_{7.1,6.3}$
Scene			
	DeepInversion/Ours	SinGAN/Ours	DGP/Ours
realism	$2.0/\mathbf{98.0}_{2.5,2.0}$	$35.5/\mathbf{64.5}_{10.3,6.0}$	$34.0/\mathbf{66.0}_{7.0,6.6}$
diversity	$18.0/\mathbf{82.0}_{8.7,5.3}$	$19.5/\mathbf{80.5}_{10.1,6.0}$	$23.5/\mathbf{76.5}_{10.5,5.9}$

The quantitative evaluation is conducted on three different datasets, ImageNet [8], Places365 [46] and DTD [7]. Tab. 1 presents the comparison across different methods. IMAGINE substantially outperforms all other methods, in terms of both IS [30] and FID [18]. LPIPS [44] is calculated between all the target and synthesized image pairs to measure the patch-wised distance between them. Higher value means synthesized image is more different and diverse from target image, and vice versa. DeepInversion obtained the best scores of LPIPS but this is meaningless as the differences between target and synthesized image pairs are due to artifacts that compromise image realism. This is evident from the low values of IS and high FID values. On the other hand, the LPIPS metric confirms that IMAGINE

produces a much more diverse set of synthesized images than the SinGAN and DGP. As IS and FID have to be computed based on a large number of samples and classes, it is not affordable for SinGAN due to its large computational time. Meanwhile, for target domains that only have one example available, FID is not the best metric for measuring the generation quality. Therefore, more convincing user studies are conducted to evaluate how realistic and diverse our generated results are compared with different alternatives via Amazon Mechanical Turk.

In user studies, turkers were asked to select the preferred image from a pair synthesized by two algorithms, IMAGINE vs. a competitor listed in Tab. 2. For each comparison, 50 turkers were recruited and each of them was assigned 20 same forced-choice tasks. Two answers have to be given for each comparison with regard to two criteria: realism (select the more realistic image); diversity (select the image more different from the given target image). The study results are shown in Tab. 2. A few conclusions can be drawn from the table. First, the human preferences are consistent with our qualitative observations and the quantitative results of Tab. 1. For all comparisons, IMAGINE is preferred to the competing method more than 50% of the time. Second, under the realism criterion, the IMAGINE has more advantage for objects compared to scenes which has more repetitive-like images. Third, unlike the LPIPS criterion, turkers found IMAGINE to outperform DeepInversion in terms of diversity. This is because humans consider real diversity, disregarding diversity due to artifacts that make images unrealistic.

4.2. Position Control

The combination of both a strong image prior, in the form of a deep classifier pre-trained over a large set of semantics, and a training image for image synthesis also offers new possibilities for image synthesis. One example is to leverage the attribution algorithms [37, 31, 2], which produce a saliency map highlighting the image regions responsible for a class prediction. When the classifier is used for image synthesis, the attribution map can influence the synthesis itself. In particular, (1) can encourage images whose attribution maps meet a specification for a object location. This entails introducing in (9) an additional regularizer

$$\mathcal{R}_{loc}(\hat{\mathbf{x}}; \mathbf{a}^0, y^*) = ||\mathbf{m}(f(\hat{\mathbf{x}}), y^*) - \mathbf{a}^0||_2, \qquad (10)$$

that encourages a user-specified saliency map $\mathbf{a}^0$ (in our experiments a Gaussian blob at a certain image location), where $\mathbf{m}(f(\mathbf{x}), y)$ is the attribution map for the prediction of the object class y in image x. We refer to (10) as the location regularizer. In this way, rather than inferring the focus of attention of an existing image [31, 45], image guided model inversion enables explicit *control* of this focus of attention while generating the image.

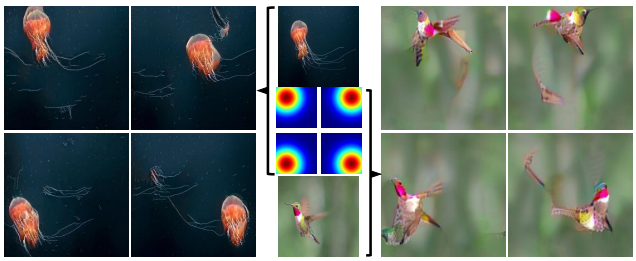


Figure 6: Object position control. Four different target positions (center) are used to supervise the position of objects, jellyfish (top-center) or hummingbird (bottom-center) on the generated images.



Figure 7: Shape control: Our approach can transfer shape of a clipart to the target image.

Many algorithms [31, 37, 45] can be used to compute the attribution map $\mathbf{m}(f(\mathbf{x}), y)$. Since most of these boil down to back-propagating $f_y(\mathbf{x})$ to some intermediate network layer and combining with the layer activations, the attribution map can be easily computed as a side product of the optimization that is already performed to synthesize the image. This leads to an essentially cost-free mechanism for controlling object position in the synthesized image. In our experiments we have used the popular Grad-CAM [31] procedure and a weight $\nu = 10$ for (10) while adding it in (9). Fig. 6 shows four images synthesized, for each of two training images ("jellyfish" shown in top-center and "hummingbird" in bottom-center of the Figure), with four different target attribution maps $\mathbf{a}^0$ (shown in the center of the Figure). Note how, by specifying the blob $\mathbf{a}^0$, the user can control the position of either "jellyfish" or "hummingbird" object in the synthesized image. The process is effective even when the object is moved to the corner opposite to where it appears in the training image.

4.3. Shape Control

Beyond position control, IMAGINE also enables manipulation of the semantics of the synthesized images as it leverages an image prior trained to discriminate semantics at various levels by using image representation at various levels of abstraction. Since the high and low level semantics of $\hat{\mathbf{x}}$ are controlled by the distribution matching of (5) at the higher and lower network layers respectively, it is possible to use different target images for low and high level semantics.

Fig. 7 illustrates this idea, with an application that complements the target image $\mathbf{x}^0$ with a synthetic image, e.g. a hand-drawn or clipart image $\mathbf{x}^c$. The latter specifies a desired semantic property for the synthetic images, namely an object shape different from that depicted in the target

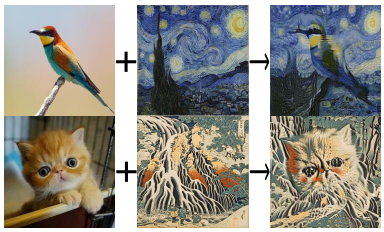


Figure 8: Style control: our approach can transfer style of an image to target image.

$\mathbf{x}^0$. This exploits the fact that clipart images tend to have less texture but strong silhouette information, and are thus a good representation for shape. Given a clipart image $\mathbf{x}^c$, the feature distribution matching regularizer of (5) in (9) is modified to

$$\mathcal{R}_{dm}(\hat{\mathbf{x}}; \mathbf{x}^c, \Phi^c) + \mathcal{R}_{dm}(\hat{\mathbf{x}}; \mathbf{x}^0, \Phi^r). \tag{11}$$

By including deeper and shallower layers in Φ^c and Φ^r respectively, it is possible to guide IMAGINE to synthesize images that combine the low-level semantics of the target and the high-level semantics (shape) of the clipart image. In all these examples, we use a ResNet-50 model and set $\Phi^c = \{conv4_6\}$ and $\Phi^r = \{conv1_1, conv2_3, conv3_4\}$.

4.4. Style Control

Inspired by the fact that style patterns usually can be encoded by the low-level layers of a classifier, similar to semantic control, IMAGINE also enables the style control by leveraging a style image to manipulate the image style of a target image [11, 20, 39]. Given a style image $\mathbf{x}^0$, this is achieved by replacing the distribution matching regularizer of (5) in (9) with

$$\mathcal{R}_{dm}(\hat{\mathbf{x}}; \mathbf{x}^0, \Phi^s), \tag{12}$$

where $\Phi^s = \{conv1_1, conv2_3, conv3_4\}$. It should be noted that here the discriminator in (6) is fed a style image $\mathbf{x}^0$ not the real target. The optimization of $\hat{\mathbf{x}}$ *starts with the target image* to be stylized. The target image here is not included in the regularizer because [9] found that it is redundant when starting from the target. The synthesized images are shown in Fig. 8.

4.5. Counterfactual visual explanations

For a query image ($\mathbf{x}^q$), counterfactual explanations are to answer the question with the form of "why the prediction is class A but not class B" [15, 40]. For example, in Fig. 9 when the prediction of a classifier to the top-center image is "cardinal", if people ask why not a "summer tanager" (called counterfactual class), the system should feedback a response with the form of "if some regions are replaced with other regions, the image would be summer tanager". Existing literature solve this problem by exhaustively searching

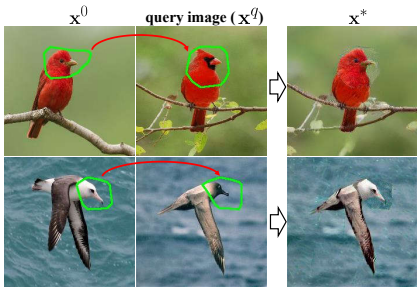


Figure 9: Counterfactual explanations of two examples. Upper: the query image is "Cardinal" and counterfactual class is "Summer Tanager"; lower: the query image is "Sooty Albatross" and counterfactual class is "Laysan Albatross".

a pair of class-discriminant regions on the query image and an image $\mathbf{x}^0$ of the counterfactual class [15]. However, it is time-consuming. [40] adopts an optimization-free method by jointly analyzing different attribution maps to localize two regions but no region replacement is introduced and changes are not done in the image space, which makes the explanation hard to understand. IMAGINE can solve this problem by semantically translating the counterfactual region from $\mathbf{x}^0$ to $\mathbf{x}^q$ by complementing below regularizer to (9)

$$\begin{aligned}\mathcal{R}_{cou}(\hat{\mathbf{x}}; \mathbf{x}^0, \mathbf{x}^q) &= ||\mathbf{r}^q \odot (\mathbf{x}^q - \hat{\mathbf{x}})||_1 \\ &+ \mathcal{R}_{dm}((1-\mathbf{r}^q) \odot \hat{\mathbf{x}}; (1-\mathbf{r}^0) \odot \mathbf{x}^0, \Phi^q)\end{aligned} \tag{13}$$

where $\odot$ denotes element-wise multiplications and $\Phi^q = \{conv1_1, conv2_3, conv3_4\, conv4_6\}$. $\mathbf{r}^q$ and $\mathbf{r}^0$ are the discriminant regions with the form of 0-1 mask, produced by [40], rounded by green circles in Fig. 9, which are also side products of the classifier. The values out of the regions are 1. Our explanation is illustrated in Fig. 9 with two counterfactual explanations results. These results are based on a ResNet-50 pre-trained on CUB200 [41]. So finally, taking the upper case as an example, for the center-top image, if someone asks "why this not a summer tanager?", our explanations would be "if it is a summer tanager, the circled regions should look like the right image."

5. Conclusion

In this work, we have proposed the IMAGINE for image synthesis from one single training image. Instead of training a GAN model, IMAGINE has synthesized images by matching various levels of semantic features of a pre-trained classifier.

Acknowledgement This work was partially funded by NSF awards IIS-1924937, IIS-2041009, a gift from Amazon, a gift from Qualcomm, and NVIDIA GPU donations. We also acknowledge and thank the use of the Nautilus platform for some of the experiments discussed above.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019.

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *ICLR*, 2018.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. Semantic photo manipulation with a generative image prior. *arXiv preprint arXiv:2005.07727*, 2020.

[5] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4502–4511, 2019.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Len Du. How much deep learning does neural style transfer really need? an ablation study. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 3150–3159, 2020.

[10] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[14] Sigal Gottlieb and Chi-Wang Shu. Total variation diminishing runge-kutta schemes. *Mathematics of computation*, 67(221):73–85, 1998.

[15] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. *ICML*, 2019.

[16] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2020.

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[22] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016.

[23] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4467–4477, 2017.

[26] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in neural information processing systems*, pages 3387–3395, 2016.

[27] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651, 2017.

[28] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *European Conference on Computer Vision (ECCV)*, 2020.

[29] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Computer Vision (ICCV), IEEE International Conference on*, 2019.

[30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[31] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[32] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[33] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[34] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T Freeman, and Tali Dekel. Semantic pyramid for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7457–7466, 2020.

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[36] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[37] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *ICML*, 2017.

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[39] Pei Wang, Yijun Li, and Nuno Vasconcelos. Rethinking and improving the robustness of image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[40] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8981–8990, 2020.

[41] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[42] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[43] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[45] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[46] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[47] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.