MAGIC: Mask-Guided Image Synthesis by Inverting a Quasi-robust Classifier

Mozhdeh Rouhsedaghat[1], **Masoud Monajatipoor**[2], **C.-C. Jay Kuo**[1], **Iacopo Masi**[3]

[1] University of Southern California (USC)
[2] University of California, Los Angeles (UCLA)
[3] Sapienza, University of Rome
rouhseda@usc.edu, monajati@ucla.edu, cckuo@sipi.usc.edu, masi@di.uniroma1.it

Abstract

We offer a method for one-shot mask-guided image synthesis that allows controlling manipulations of a single image by inverting a *quasi-robust* classifier equipped with strong regularizers. Our proposed method, entitled MAGIC, leverages structured gradients from a pre-trained *quasi-robust* classifier to better preserve the input semantics while preserving its classification accuracy, thereby guaranteeing credibility in the synthesis. Unlike current methods that use complex primitives to supervise the process or use attention maps as a weak supervisory signal, MAGIC aggregates gradients over the input, driven by a guide binary mask that enforces a strong, spatial prior. MAGIC implements a series of manipulations with a single framework achieving shape and location control, intense non-rigid shape deformations, and copy/move operations in the presence of repeating objects and gives users firm control over the synthesis by requiring to simply specify binary guide masks. Our study and findings are supported by various qualitative comparisons with the state-of-the-art on the *same* images sampled from ImageNet and quantitative analysis using machine perception along with a user survey of 100+ participants that endorse our synthesis quality.

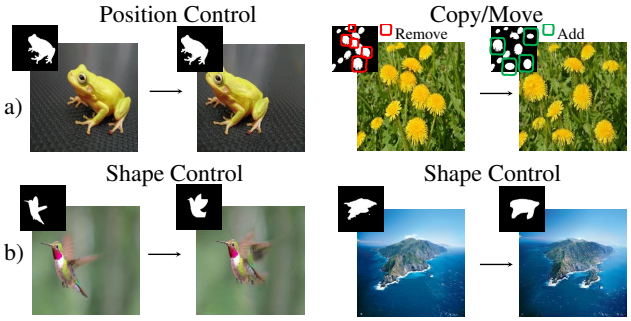


Figure 1: MAGIC allows a diverse set of image synthesis tasks following the semantic of objects and scenes requiring a single image, its binary segmentation source mask, and a target mask. In each pair, the left image is the input, and the right one is the manipulated image, guided by the mask shown on top. *a)* position control and copy/move manipulation; *b)* non-rigid shape control on object (non repetitive) and scene (repetitive) images.

1 Introduction

"A picture is worth a thousand words": a famous English language adage that is even more relevant nowadays, where the influence of multimedia data is making an impact in our daily lives through social media, web pages, and TV shows. Thus, image synthesis, a widely studied task in computer vision which enables editing an input image and/or generating new variations out of it, is even more critical today.

With advances of deep learning techniques and the availability of large annotated datasets, image synthesis methods could achieve promising results. A game changer technique was an implicit density model that learns the data density with no explicit likelihood by an adversarial game between a generator, and a discriminator—Generative Adversarial Networks (GANs) (Goodfellow et al. 2014). While powerful supervised models learning a mapping from one domain to the other have been introduced in Wang et al. (2018), they tend to perform poorly when synthesizing and manipulating rare or "long tail" images which their data distribution is not effectively learned. One-shot image synthesis is a relatively new task which focuses on using a single image as the training data for the image synthesis task which not only addresses the mentioned challenge but also obviates the need for large annotated datasets.

SINGAN (Shaham, Dekel, and Michaeli 2019) is the first model proposed for unconditional one-shot image synthesis. SINGAN uses multi-scale image generator and patch discriminator to synthesize multi-scale images and is suitable for synthesizing images with repetitive structures like landscapes of desired size but fails to synthesize realistic non-repetitive images. IMAGINE (Wang et al. 2021) is proposed to address this issue and handle both repetitive and non-repetitive images by leveraging the knowledge of a classification model. Similar to SINGAN, IMAGINE utilizes a patch discriminator for maintaining patch consistency between the training and synthesized images, while also benefits from model inversion for preserving the high-level semantics while synthesizing a new image. Although IMAGINE succeeds to synthesize more realistic results for non-repetitive images compared with SINGAN, they tend to be similar to the original image. In fact, when synthesizing new images using IMAGINE (and also SINGAN) there is no

[*] **Code:** https://github.com/mozhdehrouhsedaghat/magic

control for enforcing extreme deformations in the synthesized image. IMAGINE proposes a slightly different model for shape control, however, it requires a detailed painting from the target image with the same colors as the training image which is difficult and costly to obtain in many cases, e.g., scene images. Furthermore, the proposed method can not be used to enforce extreme deformation in the synthesized image. DEEPSIM (Vinker et al. 2021) is introduced as a mask-guided one-shot image synthesis model which allows the user to control the deformation in the synthesized image through a mask in the form of edge map, segmentation map, or their combination to enforce the desired outcome. DEEPSIM is a Pix2PixHD (Wang et al. 2018) model which is trained by augmenting the single training image and its corresponding mask using the Thin Plate Spline (TPS) method (Donato and Belongie 2002). Then, the corresponding output image is synthesized by providing a target mask. Although this mask-guided method enables synthesizing an image with the desired deformations, it requires detailed source and target masks which are not easy to obtain.

In this work, we propose a mask-guided one-shot image synthesis model to address this challenge. We entitle our method as MAGIC following "**MA**sk-**G**uided **I**mage synthesis by inverting a quasi-robust **C**lassifier". MAGIC can synthesize new real-looking high-quality variation of an image by just using binary masks as a loose supervision.

Addressing the limitations of the prior art, we make the following contributions: (1) Our proposed model, MAGIC, addresses the problem of mask-guided one-shot image synthesis using binary masks as a loose supervision and handles both repetitive and non-repetitive images. (2) We perform extensive quantitative and qualitative evaluations including a subjective evaluation with more than 100 survey participants to demonstrate the superiority of MAGIC compared with the existing work in synthesizing more realistic and higher quality images. (3) To the best of our knowledge, this is the first work that demonstrates the importance of quasi-robust model inversion for image synthesis compared with non-robust and strongly-robust model inversion.

2 Prior Work

Image Synthesis by Model Inversion. Model inversion is the process of using back-propagation of errors to maximize the likelihood of a model prediction while keeping the model weights frozen and optimizing the input. Inversion implies optimizing a pre-image subject to regularizations to resemble a natural image: this process enables producing mesmerizing pictures with Google's "DeepDream". Despite recent progress, generating high-fidelity natural images by classifier inversion *while controlling attributes such as the position of the objects and their shape* remains a challenge. The main limitation is that NNs do not provide any explicit mechanism to control these attributes. Recent methods working towards the aforementioned objective are "Dream to Distill" (Yin et al. 2020) and "IMAGINE" (Wang et al. 2021). The work in Yin et al. (2020) takes inspiration from "DeepDream" and uses image synthesis as a data generation process for a teacher-student framework. Yin *et al.* impose additional regularization on the pre-image and constraints between the statistics of the feature maps of the pre-image and those internally stored in the batch normalization (BN) statistics. IMAGINE produces variations of a guide image yet changes the feature map constraint of Yin et al. (2020) to take into account *specificity*.

Synthesis with a Robust Classifier. Santurkar et al. (2019) are the first to use a *robust* classifier for synthesis. Robust indicates a classifier optimized with adversarial training (AT) resilient to a threat model. The threat is described by bounding the magnitude of the perturbation with a ℓ_p norm (Madry et al. 2018). Robust models retain input gradients more aligned with human perception (Aggarwal et al. 2020; Kaur, Cohen, and Lipton 2019) and better capture the global shape of objects (Zhang and Zhu 2019). The invertibility property of robust models has been recently employed by (Rojas-Gomez et al. 2021) for solving inverse problems such as image denoising, example-based style transfer, or anomaly detection. Contrastingly, we use a *"quasi-robust"* model: i.e., a low max-perturbation bound quasi-robust model which retains a high classification accuracy, enabling *simultaneous* classification *and* synthesis. Another characteristic trait is that we focus on location and shape control which are applications that Rojas-Gomez et al. (2021) does not cover.

Constraining Patch-Level Statistics with GANs. The first to apply GAN at the patch level is Li and Wand (2016) with the term "neural patch", followed by Shrivastava et al. (2017) referring to as "local adversarial loss". The usage of GAN to constrain patch statistics has been used in pix2pix (Isola et al. 2017) under the name of Markovian discriminator. The work *par excellence* exploiting GAN at the patch level is SINGAN (Shaham, Dekel, and Michaeli 2019) employing a multi-scale hierarchy of GANs.

Conditioning the Synthesis on Masks. Several relevant works used segmentation masks to condition the synthesis; notable papers are Gu et al. (2019); Tang et al. (2020); Zhu et al. (2020); Tan et al. (2021): they can be categorized as mask-guided synthesis yet, unlike ours, they are not one-shot and need a training set for training their model. While Vinker et al. (2021) is a one-shot method, it uses a detailed multi-class segmentation mask whereas ours uses binary ones.

3 Method

Preliminaries and Objective. We are given an image $\mathbf{x}$ along with an aligned source binary mask $\mathbf{y}$, where this latter supervises the pixels of the object or scene that we seek to manipulate and takes values $\in \{0, 1\}$. Referring to the diagram in Fig. 2 (a), we aim at synthesizing $\mathbf{x}'$ by simply providing a binary target mask $\mathbf{y}'$, that functions as a prior for a variety of tasks such as position control, non-rigid shape control, and copy/move. For instance, for each given pair in Fig. 1, $\mathbf{y}$ and $\mathbf{y}'$ are shown in the upper left part of the input and synthesized images, respectively. In the following sections, we explain how we implement the mapping $\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{x}'$ contingent to the constraint $\mathbf{y}' \rightarrow \mathbf{x}'$, while aligning the patch distributions of $\mathbf{x}$—$\mathbf{x}'$.

Overview of the Method. As shown in Fig. 2 b), we propose inverting two main models, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{\text{ED}}$, while leveraging a PatchGAN, $\boldsymbol{\theta}_{\text{d}}$, to achieve image synthesis, preserving the semantics of objects and scenes while satisfying the target

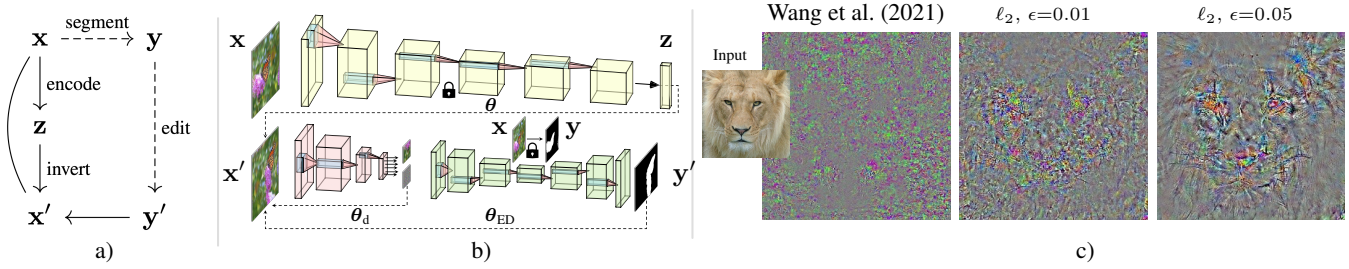


Figure 2: *a)* The binary mask $\mathbf{y}'$ is used as a guide; $\mathbf{x}'$ is inverted from $\mathbf{x}$ latent code $\mathbf{z}$, constrained with $\mathbf{y}'$. *b)* $\mathbf{x}'$ receives structured gradients from $\boldsymbol{\theta}$ to preserve the semantics of $\mathbf{z}$; it receives gradients from a discriminator to match $\mathbf{x}$'s patch distribution. An encoder-decoder (ED) is pre-trained to map $\mathbf{x}$ to $\mathbf{y}$, we then introduce gradients from ED to guide $\mathbf{x}'$ shape/location constrained with $\mathbf{y}'$. *c)* Gradients from ResNet-50—also used in Wang et al. (2021)—exhibit a sparse structure with activations around the borders; compared with a non-robust model, a quasi-robust model yields structured gradients as ϵ increases.

mask. The first inversion implements $\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{x}'$ by getting gradients from a frozen quasi-robust classifier $\boldsymbol{\theta}$. This part ensures that the reconstruction contains structured gradients to preserve object semantics. We also invert a patch-based encoder-decoder (ED) $\boldsymbol{\theta}_{\text{ED}}$ for manipulation control. Offline, we train $\boldsymbol{\theta}$ with a variant of adversarial training (AT) that perturbs the data with a very small ϵ-ball around the training samples under ℓ_2 norm, which is different than what is usually done in robust machine learning, where ϵ is set to be high to make the model resilient to attacks. Before synthesis, we also train $\boldsymbol{\theta}_{\text{ED}}$ to encode the mapping from $\mathbf{x}$ to $\mathbf{y}$. Conversely, at synthesis time, we freeze both $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{\text{ED}}$ to get gradients from them: in particular, with $\boldsymbol{\theta}_{\text{ED}}$, we replace $\mathbf{y}$ with $\mathbf{y}'$ to force the foreground object to be deformed guided by the mask $\mathbf{y}'$. Following Wang et al. (2021), we require the patch distribution of $\mathbf{x}'$ to be aligned with the patch data density of $\mathbf{x}$ with a PatchGAN $\boldsymbol{\theta}_{\text{d}}$, though, in MAGIC, the receptive field of the PatchGAN is much larger than the one in Wang et al. (2021) which helps with improving the shape consistency in the synthesized images.

Quasi-Robust Model as a Prior for Synthesis

Model Inversion. The mapping $\mathbf{x} \rightarrow \mathbf{z} \rightarrow \mathbf{x}'$ defined in Fig. 2 a) is formalized as inverting the latent embedding $\mathbf{z}$ of a deep classifier $\boldsymbol{\theta}$. A classifier $\boldsymbol{\theta} : \mathbb{Z}_{0,255}^{H \times W \times 3} \mapsto \mathbb{R}^C$ maps high-dimensional data $\mathbf{x}$ to an embedding $\mathbf{z}$ where C is the number of classes—for ImageNet (Deng et al. 2009) is $C = 1,000$. Inverting a classifier implies solving:

$$\begin{aligned} \mathbf{x}' &= \arg\min_{\mathbf{x}'} \mathcal{L}\big(\mathbf{x}', \mathbf{x}; \boldsymbol{\theta}\big) \quad \text{where} \\ \mathcal{L}\big(\mathbf{x}', \mathbf{x}; \boldsymbol{\theta}\big) &= \ell\big(\boldsymbol{\theta}(\mathbf{x}'), \mathbf{z}\big) + \rho(\mathbf{x}'), \end{aligned} \tag{1}$$

where $\mathbf{z} \doteq \boldsymbol{\theta}(\mathbf{x})$ is the latent code given the source image $\mathbf{x}$, which corresponds to the reference distribution over classes and $\boldsymbol{\theta}$ is frozen. This is an ill-posed problem since the learned function $\boldsymbol{\theta}$ is non-injective per the requirement of building invariance in the input space with respect to the same class. Hence, given a latent code $\mathbf{z}$, multiple pre-images could be generated from this code. This issue motivates the need for strong regularization ρ on the optimized pre-image $\mathbf{x}'$. After transforming the two logit terms, $\boldsymbol{\theta}(\mathbf{x}')$ and $\mathbf{z}$, into probabilities using softmax, the loss $\ell(\cdot, \cdot)$ in Eq. 1 can be Kullback–Leibler (KL) divergence. Alternatively, we can also follow a greedy approach that assigns $c = \arg\max_c \boldsymbol{\theta}_c(\mathbf{x})$ as the most likely class given $\mathbf{x}$. In this case, we can solve:

$$\mathbf{x}' = \arg\min_{\mathbf{x}'} \ell\big(\boldsymbol{\theta}(\mathbf{x}'), c\big) + \rho(\mathbf{x}') + \rho_{\boldsymbol{\theta}}(\mathbf{x}', \mathbf{x}), \tag{2}$$

where KL divergence ℓ transforms to the cross-entropy loss and c selects the index of the most likely class, following $\boldsymbol{\theta}$'s prediction. Note that for Eq. 1, Eq. 2, and in general for model inversion to work, the classifier has to retain a good accuracy on natural images, otherwise $\mathbf{x}'$ may be optimized for an incorrect class distribution. Importantly, we highlight that the accurate prediction needed by Eq. 2 *is not a property of a robust classifier*, given that exhibits low accuracy on natural images (Tsipras et al. 2019), thereby we cannot naively replace $\boldsymbol{\theta}$ with a robust model for structured gradients (Kaur, Cohen, and Lipton 2019; Aggarwal et al. 2020).

Basic Regularization. Following prior work of Mahendran and Vedaldi (2015), we used a basic regularization in the image space by bounding its squared Euclidean norm and imposing a total variation (TV) loss thus penalizing the sum of the norm of the pre-image gradient $\rho(\mathbf{x}') = \alpha\rho_{\text{TV}}(\mathbf{x}') + \beta||\mathbf{x}'||^2$ where α and β are tunable hyperparameters. We also ask $\mathbf{x}'$ to match the first and second-order statistics of the feature maps of the source image as suggested in Wang et al. (2021) to enforce a *mild* semantic consistency with the source image $\mathbf{x}$: $\rho_{\boldsymbol{\theta}}(\mathbf{x}', \mathbf{x}) = \sum_{j \in \boldsymbol{\theta}} ||\boldsymbol{\mu}_j(\mathbf{x}') - \boldsymbol{\mu}_j(\mathbf{x})||_2 + \sum_{j \in \boldsymbol{\theta}} ||\boldsymbol{\sigma}_j(\mathbf{x}') - \boldsymbol{\sigma}_j(\mathbf{x})||_2$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation of the feature maps across the spatial dimension and j indicates the layer at which the map is taken in $\boldsymbol{\theta}$. Note that this formulation per se does not fully take into account the semantic of the objects as shown in Fig. 7, second row. It is thus essential to introduce a better prior that can induce structured gradients when solving Eq. 2 for $\mathbf{x}'$.

Quasi-Robust Model for Synthesis. In order to synthesize a new image, we initialize the pre-image with normal random noise, i.e., $\mathbf{x}'_{t=0} \sim \mathcal{N}(0, 1)$. We then proceed iteratively updating the pre-image following the direction provided by the gradient of the loss in Eq. 1 with respect to the pre-image

as $\mathbf{x}'_t = \mathbf{x}'_{t-1} - \lambda \nabla_{\mathbf{x}'} \mathcal{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$ where t indicates the iteration of gradient descent and λ is the learning rate of the synthesis. The more structured is $\nabla_{\mathbf{x}'} \mathcal{L}(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$, the better and faster will be the optimization for image synthesis. As mentioned in Sec. 2, we could train $\boldsymbol{\theta}$ offline with AT as:

$$\begin{aligned} \boldsymbol{\theta}^\star &= \arg\min_{\boldsymbol{\theta}} \ell\big(\boldsymbol{\theta}(\mathbf{x} + \boldsymbol{\delta}^\star), y\big) \quad \text{where} \\ \boldsymbol{\delta}^\star &= \arg\max_{||\boldsymbol{\delta}||_p < \epsilon} \ell\big(\boldsymbol{\theta}\left(\mathbf{x} + \boldsymbol{\delta}\right), y\big) \end{aligned} \tag{3}$$

so that at synthesis time, we can obtain more structured gradients from the model $\boldsymbol{\theta}$. Eq. 3 alternates between finding an additive perturbation $\boldsymbol{\delta}$ with bounded ℓ_p norm using Projected Gradient Descent (PGD) (Madry et al. 2018) and updating the weights $\boldsymbol{\theta}$ to lower the cost on the perturbed points. However, for pre-training $\boldsymbol{\theta}$, instead of using Eq. 3 with a large perturbation ball around the data point ϵ, i.e. a strongly-robust classifier, we propose using a *very small* ϵ value so that we can retain the same accuracy of a standard classifier while getting the benefit of structured gradients of a robust one. Furthermore, we demonstrate that using a strongly-robust classifier makes the image synthesis prone to neglecting fine edges and details of images. Thereby, we replace $\boldsymbol{\theta}$ with a quasi-robust model trained on ImageNet with Eq. 3 with a ℓ_2 perturbation ball centered on the input with a very small $\epsilon = 0.05$. We refer to this model as "quasi-robust" since it is a good trade-off between clean accuracy and structured gradients, pointing out that the model is robust within our small ϵ yet is not robust from an adversarial machine learning perspective. Quasi-robust model gradients are visualized in Fig. 2 c) compared to those of Wang et al. (2021) that exhibit activations not in salient parts of the objects.

Shape Preservation and Manipulation Control

Larger Receptive Field in the Discriminator Better Preserves Shape. Similar to Wang et al. (2021) MAGIC uses a PatchGAN—patch-based discriminator $\boldsymbol{\theta}_{\text{d}}$— to ensure patch consistency between $\mathbf{x}$ and $\mathbf{x}'$. In this PatchGAN, the generator is the pre-image $\mathbf{x}'$ itself, and the discriminator plays an adversarial game to classify patches of $\mathbf{x}$ and $\mathbf{x}'$ using the Wasserstein loss with gradient penalty of Gulrajani et al. (2017). The architecture of $\boldsymbol{\theta}_{\text{d}}$ includes a series of 2D convolution followed by Batch Normalization and LeakyReLu and is shown in Fig. 2 b). Compared to the PatchGAN used in Wang et al. (2021) which has a receptive field of 9×9, $\boldsymbol{\theta}_{\text{d}}$ has a much larger receptive field of 21×21, for a 224×224 pre-image which significantly improves the synthesis results. The enhancements can be appreciated in Fig. 3 in which we replaced the PatchGAN in Wang et al. (2021) with our $\boldsymbol{\theta}_{\text{d}}$ while employing the *same* location control mechanism as in Wang et al. (2021) based on attention maps for a fair comparison. Note that because of not including the rest of our contributions, in Fig. 3, in some cases, ours incorrectly hallucinates two hummingbirds. Besides, the shape of object tends to be very similar to the training image which is not desired in image synthesis. In the next section, we explain how to resolve these issues and present our final contribution in shape control.

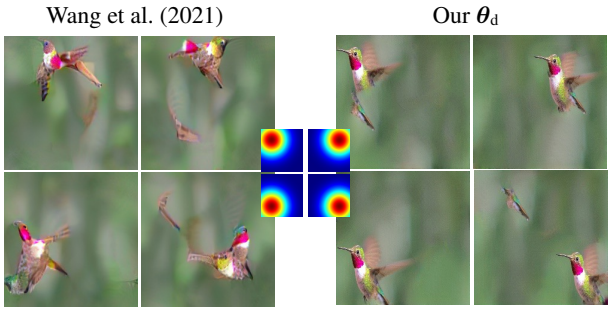


Figure 3: Shape is better preserved with ours (right) compared to Wang et al. (2021) (left).

Manipulation Control via Mask-Guided Encoder-Decoder Inversion. Unlike DEEPSIM of Vinker et al. (2021) that maps primitives to images, we work in the reverse direction by learning a mapping from the image to the binary mask specifying the object or scene of interest. In DEEPSIM, for training the Pix2PixHD model, it is required to apply strong deformations employing TPS to generate a large training set from the single image-mask pair that heavily bias the model towards producing "curved" objects and scenes. In contrast, our method's last building block consists of obtaining gradients from a patch-based encoder-decoder (ED) trained offline for binary pixel-wise segmentation supervised by $\mathbf{y}$. By doing so, we create a bottleneck through $\boldsymbol{\theta}_{\text{ED}}$ that incorporates spatial knowledge of the region of interest along with its shape. Unlike Vinker et al. (2021), our $\boldsymbol{\theta}_{\text{ED}}$ computes the expectation of the loss with respect to a set of patches by means of fully convolutional layers (Long, Shelhamer, and Darrell 2015), thereby regularizing the training. Doing so, we avoid complex data augmentation procedures such as using non-linear deformations of the input to generate new samples. At synthesis time, we can invert $\boldsymbol{\theta}_{\text{ED}}$ obtaining gradients on $\mathbf{x}'$ by replacing $\mathbf{y}$ with the target mask $\mathbf{y}'$ specified as input to the algorithm. These new gradients will guide $\mathbf{x}'$ to deform its shape according to $\mathbf{y}'$.

Final Formulation. Our final 'MAGIC' formulation preserves object and scenes semantics using gradients from a quasi-robust model, aligns patch distributions *without fragmenting objects*, and finally achieves manipulation control as described above. Our inversion with the main regularizers is provided below:

$$\mathbf{x}' = \arg\min_{\mathbf{x}'} \underbrace{\ell\big(\boldsymbol{\theta}(\mathbf{x}'), c\big)}_{\substack{\text{semantics via} \\ \text{quasi robust} \\ \text{inversion}}} + \eta \underbrace{\rho_{\boldsymbol{\theta}_{\text{d}}}(\mathbf{x}', \mathbf{x})}_{\substack{\text{align large} \\ \text{patch distribution}}} + \gamma \underbrace{\rho_{\boldsymbol{\theta}_{\text{ED}}}(\mathbf{x}', \mathbf{y}')}_{\substack{\text{manipulation} \\ \text{control}}} \tag{4}$$

where $\ell(\cdot, \cdot)$ indicates quasi-robust model inversion, $\rho_{\boldsymbol{\theta}_{\text{d}}}(\mathbf{x}', \mathbf{x})$ is patchGAN discriminator and $\rho_{\boldsymbol{\theta}_{\text{ED}}}(\mathbf{x}', \mathbf{y}')$ inverts the binary cross-entropy averaged across all the pixels of the mask $\mathbf{y}'$. We also have standard regularizer $\kappa\rho(\mathbf{x}')$ in the image space from Mahendran and Vedaldi (2015) and $\nu\rho_{\boldsymbol{\theta}}(\mathbf{x}', \mathbf{x})$, that matches feature maps distributions between

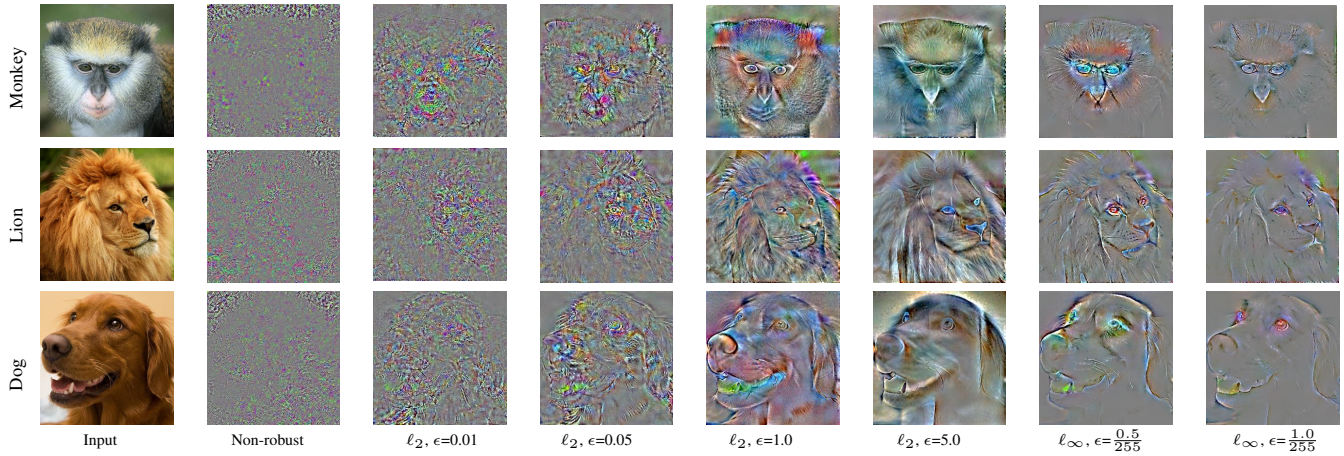


Figure 4: Visualization of the gradient of the loss with respect to the input for ResNet-50 (He et al. 2016). Input gradients seem noisy for the non-robust model used in IMAGINE but for the ℓ_2 quasi-robust models, they start to be aligned with edges as soon as ϵ slightly departs from zero. For larger ϵ, e.g., $\epsilon = 5.0$, the model becomes more robust yet gradients are more aligned with course edges. The same holds for ℓ_∞-quasi-robust models.

the two images, following Wang et al. (2021).

4 Experimental Evaluation

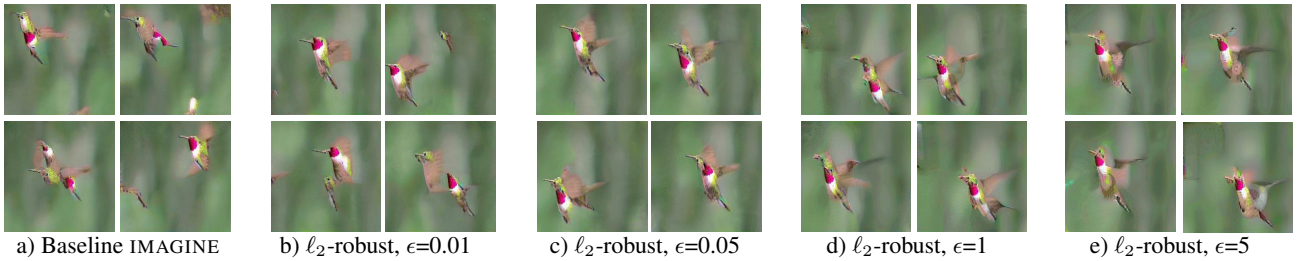
In this section, we investigate MAGIC's capabilities and the effect of the proposed components on synthesized images. We offer an ablation study illustrating the effect of the contributions on our baseline IMAGINE and analyze the improvements. We further compare MAGIC with state-of-the-art by performing qualitative and quantitative evaluations.

Implementation Details. In our experiments, the image size is H=W=224. To obtain $\mathbf{y}'$ for an image, we either manipulate its corresponding $\mathbf{y}$ or manually draw a binary mask from scratch. We use an ℓ_2-quasi-robust ResNet-50 with ϵ=0.05 as the classifier. The discriminator $\boldsymbol{\theta}_\text{d}$ is trained using the Wasserstein loss similar to what is described in Wang et al. (2021) yet by increasing the number of iterations; $\boldsymbol{\theta}_\text{d}$ weights are the only parameters optimized along with synthesizing $\mathbf{x}'$, the rest of networks are held frozen and we simply get gradients from them. For the quasi-robust model, we used the implementation publicly available in Salman et al. (2020). $\boldsymbol{\theta}_\text{d}$ consists of 5 convolutional layers with a kernel size of 4×4 in the first three layers and 3×3 in the last two layers with a stride of 1 for all layers except the second and third layers which have a stride of 2. The number of filters is set as 128 in all layers except the first, which has 64 filters. $\boldsymbol{\theta}_\text{ED}$ consists of 3 convolutional layers (encoder) followed by 3 transposed convolutional layers (decoder); each layer consists of 64 filters of size 3×3 with a stride of 1, followed by BatchNorm and Leaky ReLU with a slope of 0.2. For optimizing $\mathbf{x}'$, initially the hyper-parameters $\mathbf{h}$ in Eq. 4 are set as follows: $\eta = 0.0$, $\gamma = 30.0$, $\kappa = 1.0$, $\nu = 5.0$ while the parameters in $\rho(\mathbf{x}')$ are $\alpha = 1\text{e}{-4}$ and $\beta = 1\text{e}{-5}$. After 5,000 iterations, we start training $\boldsymbol{\theta}_\text{d}$ with $\eta = 0.05$. This technique improves the alignment of the generated image with $\mathbf{y}'$ and makes the training process more stable. We use the Adam optimizer with learning rate λ of $5\text{e}{-4}$. For other unmentioned parameters, we employ the values from IMAGINE (Wang et al. 2021).

Ablation Study

The Impact of the Quasi-Robust Model. To give insights of the effect of the quasi-robust model in Eq. 4, we visualize the input gradients for several images from ImageNet in Fig. 4. In particular, we study the influence of ℓ_2 and ℓ_∞ norm in Eq. 3 with different ϵ values on the input gradients that we get from $\boldsymbol{\theta}$. For visualizing the gradients, we follow Tsipras et al. (2019) by first clipping the gradient intensity to stay within ±3 standard deviation with respect to their mean and then rescaling it to lie $\in [0, 1]$ for each example. As illustrated in Fig. 4, as soon as ϵ slightly increases from zero, the quasi-robust models trained with the ℓ_2 norm start to pay more attention to edges in the input image which makes the gradients more aligned with human perception (Santurkar et al. 2019) and thus more suitable to be used for synthesis. However this yields a trade-off: if the model is trained with stronger attacks, e.g., $\epsilon = 5.0$, equivalent to increasing the ℓ_2-ball around the data point, then it learns to rely mostly on coarse edges as compared to fine edges so we suppose that image synthesis using strongly robust models is prone to neglect fine edges and details of the object. Our ablation in Fig. 5 offering the impact of ϵ in Eq. 3 on synthesized images confirms this hypothesis. According to this evidence, we always use a quasi-robust model with ϵ=0.05 and ℓ_2 without optimizing ϵ further. Besides, per the requirement of Eq. 2, keeping a high classification accuracy is mandatory, supporting this choice even further. According to Fig. 4, input gradients of the model trained with the ℓ_∞ norm, are more aligned with coarse edges which has the same disadvantage mentioned before.

The Interplay of the Quasi-Robust Model with Our Discriminator. MAGIC uses a PatchGAN with a receptive field of 21×21 while IMAGINE uses one with 9×9. Though hav-



a) Baseline IMAGINE b) ℓ_2-robust, ϵ=0.01 c) ℓ_2-robust, ϵ=0.05 d) ℓ_2-robust, ϵ=1 e) ℓ_2-robust, ϵ=5

Figure 5: Synthesized images by IMAGINE using models with different amounts of adversarial robustness. *a)* Using a non-robust classifier for model inversion, IMAGINE synthesizes fragmented objects. *b)* By leveraging a quasi-robust model, synthesized images look less fragmented. *c)* By increasing the robustness a bit more, the generated objects become non-fragmented and unbroken. *d-e)* Using strongly-robust models makes generated objects blurry and some of the object details disappear.

Figure 6: Synergy between the quasi-robust classifier and our discriminator.

Method	Objects		Scenes	
	FID(↓)	SIFID(↓)	FID(↓)	SIFID(↓)
BL (IMAGINE)	75.90	0.082	79.14	0.086
BL + quasi-robust	60.99	0.073	82.11	0.084
BL + quasi-robust + $\boldsymbol{\theta}_{\text{d}}$	58.36	0.045	51.86	0.042
DEEPSIM	72.71	0.271	136.87	0.578
MAGIC	**30.79**	**0.032**	**41.36**	**0.029**

Table 1: Quantitative analysis. Ablation study and comparison with the state-of-the-art using machine perception following FID an Single Image FID (SIFID) metrics.

ing a smaller receptive field is required in IMAGINE for injecting variations in the synthesized images compared with the output, it is also more prone to produce artifacts and non-realistic outputs. Fig. 6 shows our final results after incorporating the quasi-robust model along with our discriminator $\boldsymbol{\theta}_{\text{d}}$. We can appreciate how artifacts still clearly visible in Fig. 3 are removed when *these two contributions* are employed together.

The Effect of Manipulation Control. By using $\boldsymbol{\theta}_{\text{d}}$, the resulting model tends to generate images similar to the input image, yet the contribution of inverting the mask-guided $\boldsymbol{\theta}_{\text{ED}}$ is the *key* in controlling the manipulation in MAGIC: we offer qualitative results all along the paper in Fig. 1, Fig. 7, and Fig. 8. These are evidence of how the method enforces object and scene deformations albeit preserving realism.

Comparison with State-of-the-Art

We evaluate MAGIC by conducting extensive experiments on images either randomly selected from the ImageNet validation set or collected from the web, or the *same* images that previous methods used. We compare the results against DEEPSIM Vinker et al. (2021) which, to the best of our knowledge, is the state-of-the-art model for one-shot mask-guided image synthesis. For a fair comparison, we re-trained DEEPSIM with every pair used in our experiments and then fed the provided target mask. We also perform a qualitative analysis against IMAGINE Wang et al. (2021). Note that IMAGINE requires a detailed and color segmentation map for shape control and does not work with binary masks. We have already demonstrated the strengths of MAGIC compared with IMAGINE in Section 4 - Ablation Study.

Methods	Objects	Scenes
DEEPSIM (Vinker et al. 2021)	44.58%	13.19%
MAGIC (Ours)	**55.42%**	**86.81%**

Table 2: Quantitative comparison using human perception. Average preference by the users drawn from the user survey for DEEPSIM vs MAGIC for object and scene images.

Qualitative Evaluation. In Fig. 1, we present MAGIC's results for shape control and copy/move manipulation on scenes and objects. Fig. 7 shows that compared with DEEPSIM, MAGIC can better handle scene deformation (scene shape control) and also extreme object deformations. Though MAGIC is guided by $\mathbf{y}'$, even if we fix $\mathbf{y}'$, MAGIC can sample different variations and synthesize diverse images, which is highly beneficial, especially for scenes. This capability is illustrated in Fig. 8: we repeat the synthesis from three different starting points $\mathbf{x}'_{t=0} \sim \mathcal{N}(0,1)$ subject to the *same* $\mathbf{y}'$. Interestingly, diversity is usually injected in the background for object images, whereas scenes vary more.

Quantitative Evaluation. We use machine perception as a proxy for measuring the quality by employing Frechet Inception Distance (FID) by Heusel et al. (2017) and Single Image FID by Shaham, Dekel, and Michaeli (2019).

As shown in Tab. 1, MAGIC significantly outperformed DEEPSIM on both object and scene synthesis.

To further evaluate our method, we used human perception by conducting subjective evaluation of the image quality for images synthesized by MAGIC compared to DEEPSIM. For subjective evaluation, we prepared a survey containing 20 questions, each of which offers a pair of synthesized images, one by DEEPSIM and the other by MAGIC, along with the corresponding input image. The survey asks to select the

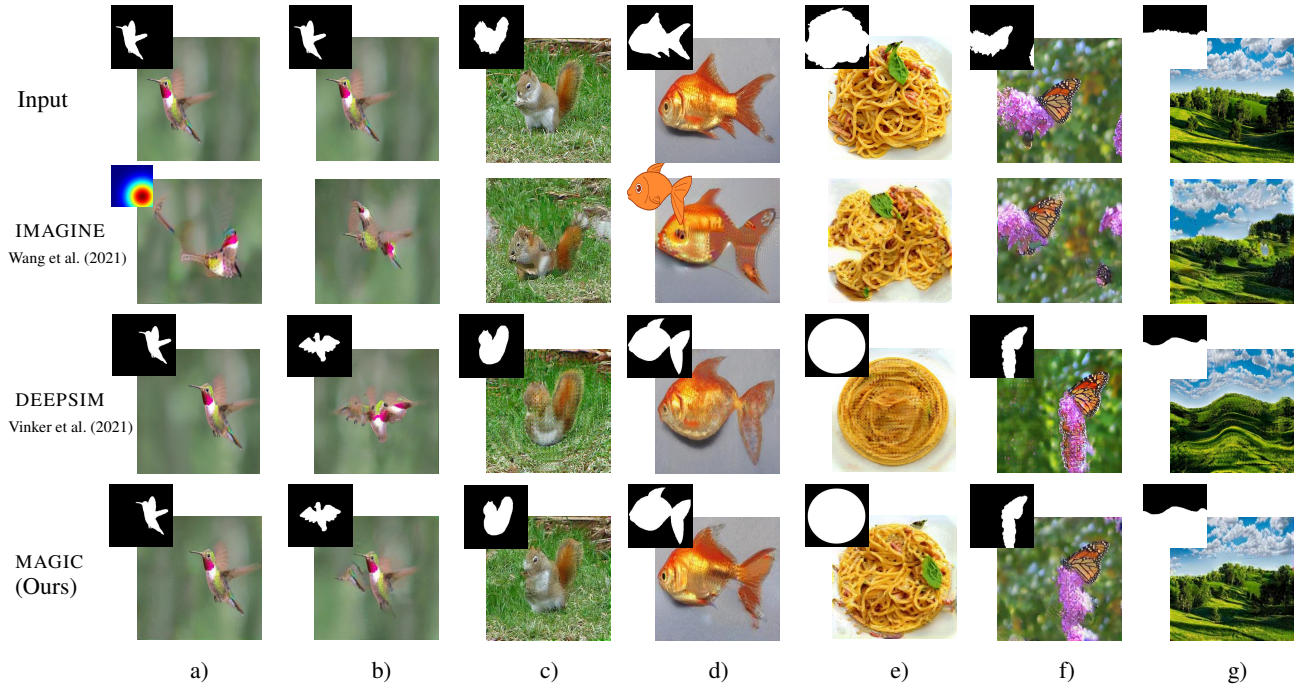


Figure 7: Qualitative comparison. DEEPSIM and MAGIC use the same guide masks $\mathbf{y}'$. *a)* IMAGINE fails to perform position control and generates fragmented results. *b) & c) & e)* DEEPSIM cannot synthesize realistic objects when $\mathbf{y}'$ is extremely different from $\mathbf{y}$ whereas MAGIC succeeds. *d)* The synthesized object by IMAGINE has an unrealistic texture while requiring more supervision for performing shape control, i.e., a color painting of the target image. *f)* IMAGINE generates samples similar to the input with no supervision while MAGIC enforces large variation using the guide masks. *g)* For shape control on scenes, DEEPSIM synthesizes blurry and 'curved' images. The results of IMAGINE in a), b), c), and d) are taken from (Wang et al. 2021).

Figure 8: For each input, we fix the mask and start the synthesis from three different starting points $\mathbf{x}'_{t=0} \sim \mathcal{N}(0, 1)$. While observing the boundaries specified by the target mask $\mathbf{y}'$ and generating realistic images, MAGIC keeps specificity and generates diverse results.

image with higher quality. In every question, each synthesized image was randomly placed in the lower left or lower right of the input image to prevent bias. Severe failure cases of DEEPSIM, e.g., Fig. 7 b), c), e), and g) were not included in the survey to further avoid biasing the evaluation. The survey was taken by 120 subjects not involved with the project.

According to the survey results shown in Tab. 2, although we removed severe failure cases of DEEPSIM, MAGIC was generally preferred more compared to DEEPSIM on objects, whereas on scenes was preferred with a very high margin.

Model Size Comparison. MAGIC with 26.253M parameters is slightly larger than IMAGINE (26.102M parameters) but much smaller than DEEPSIM (183M parameters).

5 Conclusions and Future Work

We proposed MAGIC, an effective method for one-shot mask-guided images synthesis that can find ample applications in advanced image manipulation programs. MAGIC can perform a diverse set of image synthesis tasks including shape and location control and intense non-rigid shape deformation using a single training image, its binary segmentation source mask, and a target mask. MAGIC synthesis capabilities have been judged as competing or superior to the state-of-the-art by a pool of more than one hundred surveyees. To the best of our knowledge, this is the first work that demonstrates the advantage of a quasi-robust model inversion for image synthesis. As future work, we plan to theoretically investigate the relationships between a quasi-robust model and sampling from a score matching generative model (Hyvärinen and Dayan 2005). Furthermore, we would like to extend MAGIC to handle other image synthesis tasks, e.g., image inpainting.

References

Aggarwal, G.; Sinha, A.; Kumari, N.; and Singh, M. 2020. On the benefits of models with perceptually-aligned gradients. In *ICLR*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Donato, G.; and Belongie, S. J. 2002. Approximate Thin Plate Spline Mappings. In *ECCV*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NeurIPS*.

Gu, S.; Bao, J.; Yang, H.; Chen, D.; Wen, F.; and Yuan, L. 2019. Mask-guided portrait editing with conditional gans. In *CVPR*, 3436–3445.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, volume 30.

Hyvärinen, A.; and Dayan, P. 2005. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*, 1125–1134.

Kaur, S.; Cohen, J.; and Lipton, Z. C. 2019. Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*.

Li, C.; and Wand, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 702–716. Springer.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*, 3431–3440.

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

Mahendran, A.; and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *CVPR*, 5188–5196.

Rojas-Gomez, R. A.; Yeh, R. A.; Do, M. N.; and Nguyen, A. 2021. Inverting Adversarially Robust Networks for Image Synthesis. *arXiv preprint arXiv:2106.06927*.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do adversarially robust imagenet models transfer better? In *NeurIPS*.

Santurkar, S.; Tsipras, D.; Tran, B.; Ilyas, A.; Engstrom, L.; and Madry, A. 2019. Image Synthesis with a Single (Robust) Classifier. In *NeurIPS*.

Shaham, T. R.; Dekel, T.; and Michaeli, T. 2019. Singan: Learning a generative model from a single natural image. In *ICCV*, 4570–4580.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2017. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2107–2116.

Tan, Z.; Chai, M.; Chen, D.; Liao, J.; Chu, Q.; Liu, B.; Hua, G.; and Yu, N. 2021. Diverse semantic image synthesis via probability distribution modeling. In *CVPR*, 7962–7971.

Tang, H.; Xu, D.; Yan, Y.; Torr, P. H.; and Sebe, N. 2020. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *CVPR*, 7870–7879.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness may be at odds with accuracy. In *ICLR*.

Vinker, Y.; Horwitz, E.; Zabari, N.; and Hoshen, Y. 2021. Image Shape Manipulation from a Single Augmented Training Sample. In *ICCV*, 13769–13778.

Wang, P.; Li, Y.; Singh, K. K.; Lu, J.; and Vasconcelos, N. 2021. IMAGINE: Image Synthesis by Image-Guided Model Inversion. In *CVPR*, 3681–3690.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 8798–8807.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *CVPR*, 8715–8724.

Zhang, T.; and Zhu, Z. 2019. Interpreting adversarially trained convolutional neural networks. In *ICML*, 7502–7511. PMLR.

Zhu, P.; Abdal, R.; Qin, Y.; and Wonka, P. 2020. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 5104–5113.