Diffusion Illusions: Hiding Images in Plain Sight

RYAN BURGERT XIANG LI ABE LEITE KANCHANA RANASINGHE MICHAEL S. RYOO
STONY BROOK UNIVERSITY

RBURGERT@CS.STONYBROOK.EDU

We explore the problem of computationally generating special images that produce multi-arrangement optical illusions when physically arranged and viewed in a certain way, which we call 'prime' images. First, we propose a formal definition for this problem. Next, we introduce Diffusion Illusions, the first comprehensive pipeline designed to automatically generate a wide range of these multi-arrangement illusions. Specifically, we both adapt the existing 'score distillation loss' and propose a new 'dream target loss' to optimize a group of differentially parametrized prime images, using a frozen text-to-image diffusion model. We study three types of illusions, each where the prime images are arranged in different ways and optimized using the aforementioned losses such that images derived from them align with user-chosen text prompts or images. We conduct comprehensive experiments on these illusions and verify the effectiveness of our proposed method qualitatively and quantitatively. Additionally, we showcase the successful physical fabrication of our illusions — as they are all designed to work in the real world. Our code and examples are publicly available at our *interactive* project website: https://diffusionillusion.github.io/

ACM Reference Format:
Ryan Burgert Xiang Li Abe Leite Kanchana Ranasinghe Michael S. Ryoo , Stony Brook University , rburgert@cs.stonybrook.edu . 2024.

Diffusion Illusions: Hiding Images in Plain Sight. *ACM Trans. Graph.* 1, 1 (May 2024), 25 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

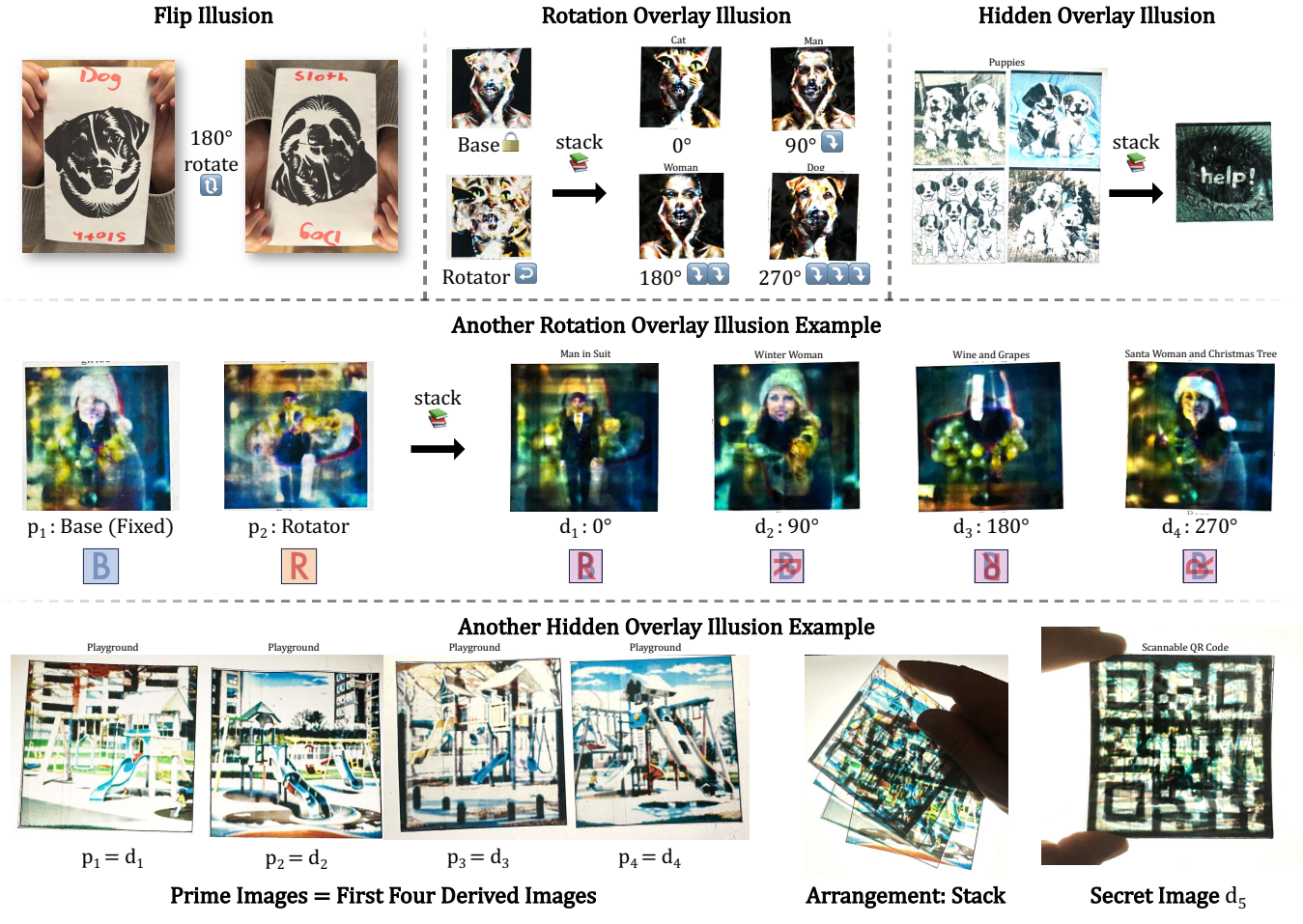


Fig. 1. Diffusion Illusions are a new class of automatically generated optical illusions. The images on top demonstrate the three major types of illusions we discuss in this paper: Flip Illusions, Rotation Overlay Illusions, and Hidden Overlay Illusions. (Terminology is formally defined in Section 2). The bottom showcases an example of Hidden Overlay Illusions: four images (prime images $p_{1...4}$) that when stacked on top of each other (arrangement) reveal a new fifth image (derived image d_5). *Please note that these illustrations are all photographs of the generated images physically fabricated in the real world.*

Stony Brook University
rburgert@cs.stonybrook.edu

1 Introduction

An image that is viewed right-side up appears to be an ordinary photo of a dog but viewed upside-down looks like a sloth. Four images, each showing an everyday playground, when superimposed form a QR code (see Fig. 1). These types of images that cause illusions have long required immense time and skill to create, but we have developed a general pipeline capable of generating appealing illusions automatically. More specifically, given a frozen text-to-image diffusion model, we adapt existing score distillation loss and propose a new dream target loss to optimize a group of prime images differentiably parametrized by fourier feature networks. Eventually, the images are optimized to comply with the textual and/or image prompts given by the user to trigger illusions in a certain arrangement.

Generating such images is not the sole domain of play. These multi-arrangement illusions – that is, visual stimuli whose interpretation depends on how they are arranged and viewed – have been created and studied for centuries. While they are an appealing sort of "visual puzzle", they also reveal much about how humans perceive the world and about the abstract structure of images. Even though illusions have been created and studied for centuries, and certain types have been generated by computers for decades, photorealistic illusions have remained largely out of reach until the very recent past, and until this point, there has been no general framework for understanding and generating such illusions.

1.1 Contributions

In this paper, we present the first formalized, generic framework for creating such illusions. We name our framework *Diffusion Illusions*. Our major contributions can be summarized as follows:

(1) We provide the a formal definition for the problem of generating these multi-arrangement illusions;
(2) We present Diffusion Illusions, a flexible tool for generating multiple types of illusions;
(3) We assess the quality of computer-generated illusions in multiple aspects and conduct computational comprehensive experiments to validate the effectiveness of our method;
(4) We successfully fabricate the generated images and their corresponding illusions in the real world.

1.2 Related Work: History of Illusions

1.2.1 Classical illusions Images whose interpretation depends on viewing angle or category bias, sometimes known as ambiguous images, have been designed for centuries. Such images have drawn the scholarly interest of psychologists [Boring 1930; Jastrow 1899] and philosophers [Wittgenstein 1953] since the 1800s. Ambiguous images have been used experimentally to understand how category bias during perception varies as people age [Nicholls et al. 2018], and families of ambiguous images, such as ambigrams [Hofstadter 1985], are often constructed as a way of better understanding the domains they belong to. We present some relevant examples of classical illusions in Figure 2.

1.2.2 Computationally-generated illusions A growing stream of research has focused on computationally generating specific types of illusions. One early example is hybrid images [Oliva et al. 2006]. Hybrid images are created from two images by combining the low-frequency features of one with the high-frequency features of the other. Viewers see the object from the low-frequency image when viewing the hybrid image from a distance, and see the object from the high-frequency image when viewing up-close. While this process may be automated, the authors note that for best results, the overall shapes of the low-frequency and high-frequency images should be manually aligned.

Fig. 2. A brief history of illusions. **Classical illusions**: **(A)** "Fruit Basket" (1500s) by Giuseppe Arcimboldo provides a very early example, depicting a face when viewed in one orientation and a fruit basket when viewed in the other. **(B)** When viewed directly, "Kaninchen und Ente" (1892) is ambiguous; 45° rotations make it appear as a rabbit or a duck [Jastrow 1899; Wittgenstein 1953]. **(C)** "My Wife and My Mother-in-Law" (1915) by William Ely Hill may be interpreted as showing either a young or an old woman depending on how it is grouped [Boring 1930; Nicholls et al. 2018]. **Computationally-generated illusions**: **(D)** A hybrid image which appears as a leopard when viewed close-up and an elephant when viewed from afar [Oliva et al. 2006]. **(E)** A wire sculpture depicting three 2010s American politicians when viewed from different angles [Hsiao et al. 2018]. **Diffusion-based illusions**: **(F)** A duck when viewed upright and a rabbit when rotated 90° ccw [Tancik 2023]. **(G, H, I)** ambigrams depicting 'Ohio cloud yeah python' [Samsudin 2023], 'G' and 'Q' [Loviscach 2010] and 'Beauty' [Zhao et al. 2023] respectively. **(J)** A giraffe when viewed upright and a penguin when viewed upside-down [Geng et al. 2023]. **(K)** A tiger camouflaged into a cliff [Chu et al. 2010] **(L)** The only overlay illusion example in Nakajima and Yamaguchi [2004], featuring grainy pixel-wise variation **(M)** A mask whose facial expression changes when lit differently [Chandra et al. 2022]

An related type of illusion is steganography, in which apparently normal objects may be viewed in a particular way to uncover a hidden meaning. In The Magic Lens [Papas et al. 2012], seemingly meaningless dots are generated such that, when viewed through an intricate refractive lens, they will comprise a specified image. [2010] camouflages one image into another by imitating its texture. [2013] makes a small change to an image that's hard to spot, but easily pointed out.

A number of researchers have created 3-dimensional objects that are interpreted as different objects when they are viewed from different angles. In multi-view wire art [Hsiao et al. 2018], a single 3D wire may be viewed or lit from multiple angles to obtain different clean line drawings; and in view-dependent surfaces [Perroni-Scharf and Rusinkiewicz 2023], a colored 3D-printed height field may be viewed from different angles to obtain different colored images. Chandra et al. [2022] creates models are perceived differently when relit, seen in 2.

Other types of illusion-generation have been also explored, such as auditory illusions [Chandra et al. 2021], still images that appear to move [Chi et al. 2008], the creation of impossible-geometry images from 3d models [Owada and Fujiki 2008]. Nakajima and Yamaguchi [2004] generates illusions that have a similar physical analogy to our Hidden Overlay illusions - involving stacked transparent sheets (see Fig. 2). Their paper did not showcase any photographs of their illusions working physically, and is limited to two transparencies.

1.2.3 Diffusion-based Image Generation Denoising Diffusion Probabilistic Models [Ho et al. 2020] are a class of generative models that resulted in rapid advances for image generation tasks, including text-to-image [Dhariwal and Nichol 2021; Nichol et al. 2022; Ramesh et al. 2022, 2021; Saharia et al. 2021a, 2022, 2021b; Yu et al. 2022] and robotics [Chi et al. 2023; Li et al. 2023; Zhu et al. 2023].

In this formulation, samples are generated via a reverse-diffusion process, where a denoising U-Net $\mathcal{F}_u$ iteratively denoises pure Gaussian noise to clean images. These models are often conditioned on text for controlability. We use latent diffusion models [Rombach et al. 2022][Podell et al. 2023], which denoise in latent space instead of image space, using an an auto-encoder $\mathcal{F}_e$ to decode the clean result into a high resolution image.

Recent works [Burgert et al. 2022a; Poole et al. 2022] sample pretrained diffusion models without re-training to generate outputs in novel domains. Score Distillation introduced in DreamFusion [Poole et al. 2022] is the underlying technique enabling optimization of samples in any arbitrary parameter space without backpropagation through the diffusion model. We utilize these techniques to construct a novel framework for illusion generation. These rapid advances have led to an exploration of suitable evaluation metrics, both quantitative and qualitative [Benny et al. 2020; Betzalel et al. 2022; Friedman and Dieng 2022; Lee et al. 2023b; Yeh et al. 2023], which we use to evaluate our proposed framework.

1.2.4 Contemporary Work Following recent image generation developments, a small but growing body of non-scholarly or unpublished work has approached the problem of generating multi-view 2D images [Tancik 2023] or textual ambigrams [Loviscach 2010; Samsudin 2023; Shirakawa and Uchida 2023; Zhao et al. 2023].

We presented a preliminary demo of our work at CVPR 2023 [Burgert et al. 2023]. Since then, Visual Anagrams [Geng et al. 2023] presented a formal framework for illusion generation. As explicitly stated in their paper, they operate on a *subset of Diffusion Illusions* (namely, those with a single "prime image" in our terminology). Their work is restricted to orthogonal transformations (i.e. flip illusions, breaking the prime into shuffled puzzle pieces, or image inverting) and cannot be used to generate the overlay illusions shown in this paper. Yet, their more restricted approach runs much faster than ours, requiring only a single diffusion pass.

Following recent image generation developments, a small but growing body of non-scholarly or unpublished work has approached the problem of generating multi-view 2D images [Tancik 2023] or textual ambigrams [Samsudin 2023][Zhao et al. 2023][Shirakawa and Uchida 2023][Loviscach 2010]. A preliminary version of our Diffusion Illusions project was shown at CVPR2023 as a demo [Burgert et al. 2023]. Since then, we have developed the formal, generic approach described in this paper. Also following and building on Burgert et al. [2023], Visual Anagrams [Geng et al. 2023] presents a formal framework for illusion generation. As explicitly stated in Geng et al. [2023], they operate *subset of Diffusion Illusions* (namely, those with a single "prime image" in our terminology). It is restricted to orthogonal transformations (i.e. flip illusions, breaking the prime into shuffled puzzle pieces, or image inverting) - and cannot be used to generate the overlay illusions shown in this paper. Geng et al. [2023] also runs much faster than our Diffusion Illusions - requiring only a single diffusion pass.

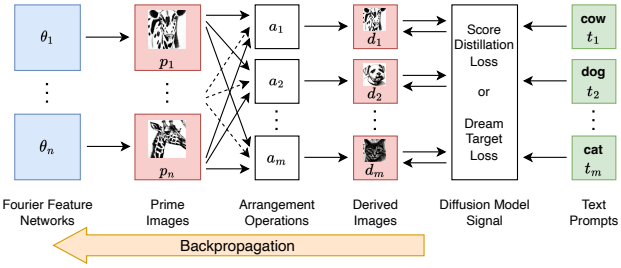


Fig. 3. Architecture overview. Trainable components are shown in blue, intermediate variables are in red, non-trainable functions are in white, and inputs are in green. A diffusion network provides two different loss signals pulling the derived images towards the text prompts. Only a single loss signal, either Score Distillation Loss or Dream Target Loss, is computed at each training step. Gradients on the derived images are backpropagated through the arrangement operations and prime images to the parameters of the Fourier Feature Networks. No backpropagation occurs through the diffusion network.

2 Problem Statement

A unifying pattern behind many types of ambiguous images or illusions is the situation where a single set of stimuli may be interpreted in multiple ways depending on how they are viewed. We leverage this pattern to define a quite general class of computational problems which we will use to represent the generation of illusions. We consider the situation that occurs when a set of physical images called *prime images* p are viewed or *arranged* in multiple ways, with each arrangement yielding a unique perceived image, referred to as a *derived image* d, that represents a specific object or scene.

Most existing illusions we discussed consist of a single 2D image or 3D object as a prime image, with the arrangements being simple translations and rotations of the prime image in 2D or 3D space. In the simplest case where a 2D drawing is rotated to yield different perceived objects, the arrangement operations may be modeled as simple rotations. The near and distant views composing the Hybrid Images illusion [Oliva et al. 2006], on the other hand, might be best modeled by high-pass and low-pass spatial frequency filters.

In an effort to find a general definition of our multi-arrangement illusions and leverage the new possibilities afforded by text-to-image models, we do not limit ourselves to a single prime image. We additionally consider situations where *multiple* composable prime images, for instance, stencils or light-filtering transparencies, may be arranged in different ways to yield different derived images. In the particular case of composing two light-filtering transparencies, the arrangement operation may be modeled as a rotation of each prime image followed by multiplication to model light-filtering.

Stony Brook University
rburgert@cs.stonybrook.edu

Formally, the illusion process is described as follows. Consider a prime image space $\mathcal{P}$ representing physically realizable visual stimuli, and a derived image space $\mathcal{D}$ representing a human view of a scene. (Practically, we use 2D RGB images to represent both spaces.) Then, an illusion consists of a tuple of n prime images $\{p_1, p_2, \ldots, p_n\}, p_i \in \mathcal{P}$ and a tuple of m arrangement operations $A = \{a_1, a_2, \ldots, a_m\}, a_j : \mathcal{P}^n \rightarrow \mathcal{D}$. Each a_j represents an arrangement of all of the prime images to obtain a single derived image d_j, such that the illusion yields a tuple of m derived images $\{d_1, d_2, \ldots, d_m\}, d_j \in \mathcal{D}$. (This articulation may be easily generalized to heterogeneous illusions, such as wireframes viewed through stencils; where each p_i belongs to its own prime image space $\mathcal{P}_i$.)

This framing is complementary to existing literature on "ambiguous images". The illusion process is not intended to cover images that have multiple interpretations when viewed in exactly the same way, though it may be possible to articulate a perceptual bias towards a certain category as a type of arrangement. However, the illusion process otherwise broadens the category to include situations involving multiple composed images. We propose multiple examples below that are to our knowledge wholly novel.

This definition allows one to separate the process of creating an illusion into two steps: first, selecting a prime image domain and defining and modeling the arrangement operation; and second, searching the prime image domain for images that yield the desired derived images when arranged in each way. While the first step requires creativity and experimentation, the second is concrete and may be practically automated, as discussed in Section 3.

3 Method

We introduce Diffusion Illusions, a flexible tool for generating multiple types of visual illusions that can be styled with unprecedented control (e.g. photorealistic images, artistic styles, or even arbitrary information such as QR codes). At a high level, the Diffusion Illusions pipeline consists of

- a set of prime images parameterized by θ ($\mathcal{P}$),
- a set of specific arrangement processes (A, that derive images from all primes),
- a frozen text-to-image diffusion model ($\mathcal{F}$)

We refer to the outputs of the arrangement processes as derived images (D). The diffusion model is used to provide a signal using one of two mechanisms (*Score Distillation Loss* or *Dream Target Loss*, which will be covered in Section 3.3) to suitably optimize the prime images, which in turn modifies the derived images. Our overall pipeline is illustrated in Fig. 3.

3.1 Prime Images

As described in Section 2, prime images are the physical images we eventually want to generate, that will trigger an illusion when viewed or arranged in multiple ways.

In our framework, prime images are represented as 512×512 dimensional RGB images, meaning that $\mathcal{P} \simeq \mathbb{R}^{(512,512,3)}$. Instead of direct pixel-space image representation, we use Fourier Features Networks (FFN) [Tancik et al. 2020] to represent prime images in parametric form. For each prime image, the learnable weights of a single MLP network act as its representation. The MLP network maps image-space coordinates to corresponding RGB values similar to [Burgert et al. 2022b], forming an implicit image representation.

3.2 Arrangement Processes

The purpose of arrangement processes, A, is to operate on a set of prime images (including single element sets) and produce unique outputs, the derived images. For a single arrangement process a_i,

$$d_i = a_i(P) \tag{1}$$

each unique sequence of prime images produces a distinct derived image, d_i. Each operation $a_i \in A$ should possess three properties: 1) For the same set of inputs the operation should always provide the same output (fixed operation). 2) a_i should also be differentiable, i.e., the possibility to explicitly calculate gradients propagation from output to input through the operation. 3) a_i should also be realizable in the real world: some series of physical actions on prime images (in physical form) should result in the same derived image. To summarize, an arrangement process must be fixed, differentiable, and realizable in the real world.

We select three illusion categories for further study:

- **Flip Illusion** is one of the most classical types of illusions. We define this illusion as consisting of a single 2D prime image, which is interpreted as some object when viewed upright (the first derived image d_1) and as another object when viewed upside-down (the second derived image d_2). Plesae see Fig. 7.
- **Rotation Overlay Illusion** is a type of illusion involving multiple prime images. This illusion is based on two square light-filtering 2D prime images, one base and one rotator. The rotator image is rotated by 0, 90, 180, and 270 degree angles and superimposed on the base image; each rotation yields a derived image interpreted as a different object (see Fig. 8).
- **Hidden Overlay Illusion** is introduced to push the boundaries of the prime-to-derived relationship, in which four light-filtering prime images, each of which is interpretable on its own, may be merged to obtain a fifth hidden image. Here the modeled view process for the first four derived images is simply the identity function; the view process for the fifth is the product of the four prime images (see Fig. 9).

We select these illusion styles to cover varying set cardinalities for prime images and arrangement processes. The arrangement process relevant to each illusion is presented in Table 1. We also present photographs of real-world fabrications for each illusion type in Fig. 1, Fig. 8 and Fig. 9.

3.3 Diffusion Illusion Optimization

Having selected three diverse illusion styles, we next discuss the process for learning optimal prime images. Given fully-differentiable operations (also realizable in the physical world) that arrange a set of prime images to produce a derived image, we leverage two types of losses in successive phases to provide suitable alignment signals to the derived images, which in turn would update the prime images. In the first phase, we use between 500 and 4000 steps of *Score Distillation Loss* [Poole et al. 2022], a high-fidelity but expensive algorithm that applies a conditional denoising model to the input at every image update step with a learning rate of 10^{-3}. In the second phase, we use between 8 and 20 steps of our *Dream Target Loss*, a faster technique that pulls the derived images towards periodically

Illusion	n	m	$\mathbf{a}$
Flip	1	2	$a_1(\mathbf{p}) = p_1$ $a_2(\mathbf{p}) = \mathrm{rot}(p_2, 180)$
Rotation Overlay	2	4	$a_j(\mathbf{p}) = p_1 * \mathrm{rot}(p_2, 90j)$
Hidden Overlay	4	5	$a_j(\mathbf{p}) = p_j, j \leq 4$ $a_5(\mathbf{p}) = p_1 * p_2 * p_3 * p_4$

Table 1. This table describes our mathematical models of the Flip, Rotation Overlay, and Hidden Overlay illusions, describing the number of prime images n, the number of derived images m, and the arrangement operator $\mathbf{a}$ mapping from prime image space $\mathcal{P}^n$ to derived image space $\mathcal{D}^m$. The arrangements in the Flip illusion are simply the identity and a 180 degree rotation. The arrangement operations in the Overlay illusions use a multiplication blend operation to model shining light through multiple transparencies; the result is multiplied by a constant and normalized using tanh to avoid losing dynamic range.

updated target images. The exact numbers of steps are variable and specified in Table 2.

Our algorithm is built on latent diffusion models, covered in Section 1.2.3. Given a frozen text-to-image latent diffusion model $\mathcal{F}$ [Rombach et al. 2022] which contains a text encoder $\mathcal{F}_t$, an image encoder $\mathcal{F}_e$ and the denoising network $\mathcal{F}_u$, we initialize a series of prime images p_i each represented by a Fourier Feature Network with random parameters θ_i. Derived images d_i then can be presented by the arrangement process as introduced in Section 3.2. For each derived image d_i, a target t_i that describes in natural language the expected visual appearance of its final form is given by the user.

3.3.1 Score Distillation Loss Score Distillation Loss ($\mathcal{L}^{\mathrm{SD}}$) is a widely-used technique to align images with external conditioning such as textual prompts. In essence, SDL ($\mathcal{L}^{\mathrm{SD}}$) randomly selects a timestep τ of the denoising process, adds noise η_τ proportionate to the timestep τ to a derived image d_i and applies the denoising process, which is conditioned on corresponding t_i, to $d_i+\eta_\tau$ to obtain an estimated noise $\hat{\eta_\tau}$. The difference, which we implement as a mean absolute error, between the estimated noise $\hat{\eta_\tau}$ and actual noise η_τ provides a signal for the discrepancy between the derived image d_i and the target description t_i for the derived image. This difference is normalized by τ and then provided as a gradient to the derived image and backpropagated through the arrangement process to the prime image. Importantly, this process does not require any backpropagation through the diffusion model.

As shown in Eq. (3), score distillation loss provides gradients to optimize the image parameterized by θ, such that iterative updates to the image converge its appearance towards the paired text t_i.

$$\hat{\eta_\tau} = \mathcal{F}_u(d_i + \eta_\tau, \tau, \mathcal{F}_t(t_i)) \tag{2}$$

$$\mathcal{L}_i^{\mathrm{SD}}(t_i, d_i) = \|\eta_\tau - \hat{\eta_\tau}\|_1 \tag{3}$$

3.3.2 Dream Target Loss is a novel alternative to Score Distillation Loss, producing higher fidelity illusions in less iterations. Please see Fig. 11 to how it reduces score distillation loss's artifacts.

Dream Target Loss ($\mathcal{L}^{\mathrm{DT}}$) first applies a conditional image-to-image process $z_i = \mathcal{G}(t_i, d_i)$ to obtain a target image z_i for each derived image d_i, conditioned on the textual prompt t_i. Then we gradually pull each derived image d_i towards its target image z_i using a combination of the structural image similarity loss ($\mathcal{L}_{SSIM}$) and a pixel-wise mean squared error loss ($\mathcal{L}_2$). Therein, we obtain a joint loss to similarly learn optimal prime images p_i resulting in derived images aligned to each of our target concepts.

$$z_i = \mathcal{G}(t_i, d_i) \tag{4}$$

$$\mathcal{L}_i^{\mathrm{DT}}(z_i, d_i) = \mathcal{L}_{\mathrm{SSIM}}(z_i, d_i) + \mathcal{L}_2(z_i, d_i) \tag{5}$$

Dream Target Loss produces less noise-like encoder artifacts than using score distillation loss alone (see Fig. 11). The total dream target loss is a weighted average across all per derived image loss terms.

$$\mathcal{L}^{\mathrm{DT}} = \sum w_i \mathcal{L}_i^{\mathrm{DT}} \tag{6}$$

where the loss terms are weighted by importance values $w_{1\ldots m}$. By default, all $w_i = 1$ except in the hidden overlay illusion where the hidden image is prioritized via $w_5 = 3$ to increase its fidelity.

In practice, for each target image, we optimize the prime image for multiple steps using the dream target loss with a learning rate of 10^{-2}. Then we repeat the process with the latest prime image so that the target image is updated towards the current derived image for faster convergence (Illustrated in Fig. 10). We implement $\mathcal{G}$ using SDEdit [Meng et al. 2022] where random noise is first added to the input image, and is then iteratively denoised conditioned on the text prompt using a frozen diffusion model to generate an output image. The strength of SDEdit is decreased according to a linear schedule in each dream-target-loss step, allowing it to converge faster. Please see our appendix for detailed pseudocode.

Note that in both Score Distillation Loss and Dream Target Loss, we propagate gradients to the prime images, updating their parametric representation (i.e. the weights of the MLP Fourier Feature Networks θ), and the diffusion model is kept frozen.

3.3.3 Visual Prompt Optionally, one or more t_i can be a specific target image instead of a text prompt — letting users hide targets such as QR codes or blocks of text. In that case, for both phases, the discrepancy between the derived image and the target image is measured using Eq. (5), providing gradients for the prime images.

3.4 Fabrication

The flip illusions are trivial to manufacture in real life and need only a printer. The hidden overlay and rotation overlay illusions are created by printing their prime images on overhead display sheets on a color laser printer, before being laminated to protect them from scratches. With a strong enough backlight, the hidden overlays and rotation overlay illusions can be performed on regular pieces of paper as well. Please refer to Appendix D for more details.

4 Experiments

In this section, we evaluate our framework presenting qualitative visualizations and quantitative metrics.

4.1 Qualitative Evaluation

We illustrate randomly selected example outputs of our Diffusion Illusions framework. Visualizations for our three selected illusion styles, Flip Illusion, Rotation Overlay Illusion, and Hidden Overlay Illusion are presented in Fig. 7, Fig. 15, and Fig. 14 respectively. For more interactive examples, please refer to the project website https://diffusionillusion.github.io/

4.2 Quantitative Evaluation

Next, we quantitatively benchmark the Hidden Overlay Illusion generated by the variants of Diffusion Illusion in multiple aspects and demonstrate the generalization ability and robustness of the

Stony Brook University
rburgert@cs.stonybrook.edu

proposed framework. Please check Appendix C as well for other illusions and more details.

Image Generation Protocol We design a pipeline that constructs diverse textual prompts randomly and automatically. The pipeline relies on two sets of textual prompts. The first set T^s is of sentences where each sentence describes a unique art style of an image and contains one *subject* token representing the potential subject of the sentence. The second set T^o is of different subjects like 'dog', 'cat', 'car', and so on. When generating images with a specific style $t^s \in T^s$, we uniformly sample five unique subjects t_i^o where $i \in \{1, \ldots, 5\}$ from T^o. Then we substitute the *subject* token in t^s with t_i^o to construct the textual prompt t_i. Finally, $t_1, \ldots, t_5$ is used to guide the generation of derived images.

For a full evaluation, the whole pipeline is repeated for N times per style t^s to generate N groups of illusion images. In practice, we set $|T^s| = 4$, T_o is the set of all object classes except 'person' in PASCAL VOC [Everingham et al. 2010] ($|T^o| = 19$), and $N = 64$. Please refer to the Appendix C for a list of all subjects and styles.

Evaluation Metrics Inspired by recent works on diffusion model evaluation [Lee et al. 2023a; Yeh et al. 2023], we measure the following properties of the derived images:

- *Controllability:* how well the generated images align with the textual prompts. For each generated image and its corresponding textual prompt, we measure the *average cosine similarity* between the image embedding and the text embedding, extracted from a pretrained CLIP [Radford et al. 2021] model.
- *Diversity:* the variety of generations given a single prompt. For images generated by the same textual prompt, we calculate two *Venti scores* [Friedman and Dieng 2022] independently based on two visual embeddings: the `[CLS]` embeddings of DINOv2 [Oquab et al. 2023] and CLIP visual embeddings (see Appendix).
- *Aesthetics:* the assessment of an image's visual appeal and artistic quality. For each image, we utilize AVA LAION-Aesthetics Predictor V2, which is pretrained on AVA [Murray et al. 2012] dataset, to estimate an aesthetics score range from 0 to 10.

In addition, we study a new property *Independence* specifically for the illusion scenario. Intuitively, each image is expected to stick to its corresponding textual prompt while not being distracted by other textual prompts in the same group. Such property is named as *Independence*, which is different from *Controllability* because independence is designed to reflect not only the similarity between an image and its corresponding textual prompt but also the *dissimilarity* between the image and the textual prompts for other images. In other words, this property focuses on how well the prime images can 'hide' the overlay image or how challenging it will be for people to infer the overlay image from a single prime image and vice versa.

- *Independence Score:* Therefore, we propose a new metric Independence Score to reflect such property. Consider a set of m derived images, denoted as $\{d_1, d_2, \ldots, d_m\}$, along with their corresponding textual prompts $\{t_1, t_2, \ldots, t_m\}$. Initially, we extract the visual embeddings $v_i = f_v(d_i)$ and text embeddings $e_j = f_t(t_j)$ using the visual encoder f_v and the text encoder f_t from a pretrained CLIP [Radford et al. 2021] model respectively. Subsequently, we compute the cosine similarity $k_{ij} = \text{CosineSimilarity}(v_i, e_j)$ between any visual and text embeddings v_i and e_j. The results are assembled into a matrix K, where k_{ij} is put in the i-th row and j-th column. The Independence Score S_{IS} is calculated by the following equations.

$$K_{d \in \{0,1\}} = \text{Softmax}(K/\tau, d) \tag{7}$$

$$S_{\text{IS}} := \min(\text{diag}(K_0) \cup \text{diag}(K_1)) \tag{8}$$

where $\tau = 0.05$ is a temperature constant, $\text{Softmax}(\cdot, l)$ stands for softmax operation along l-th dimension and $\text{diag}(\cdot)$ presents a set of the diagonal elements of $(\cdot)$. S_{IS} is designed to become higher when all images d_i align best with their corresponding textual prompts compared with other textual prompts.

Label	Model	w_5	SDL Steps	DTL Steps	Explanation
A	SDXL	3	500	8	Our Main Method
B	SDXL	1	500	8	Equal Weights
C	SD15	3	500	8	Stable Diffusion 1.5
D	SDXL	3	4000	1	Almost entirely SDL
Base	SDXL	3	0	1	Our Baseline

Table 2. A summary of our methods. SDL stands for Score Distillation loss, and DTL stands for Dream Target Loss. See our appendix for more ablations.

Methods

We compare several variants of our method against a baseline. The *Baseline* generates images with a single step of dream-target loss. Our main *Method A* optimizes the images using 500 steps of score distillation loss followed by 8 steps of dream target loss. It applies relative weights of $w = [1, 1, 1, 1, 3]$ to prioritize the quality of the hidden image over its constituent primes.

We also test three ablations: *Method B* uses equal weights $w = [1, 1, 1, 1, 1]$ for all derived images. *Method C* replaces SDXL with Stable Diffusion 1.5, keeping other settings the same as the main method. *Method D* uses 4000 steps of score distillation loss followed by a single dream target loss step for smoothing, to evaluate the effectiveness of score distillation loss alone. For a fair comparison, all methods were run for up to 15 minutes on a single NVIDIA A100 GPU. Please see the appendix for several more ablations.

Results For all metrics, we report the score distributions achieved by our default method and the baseline in Fig. 4.

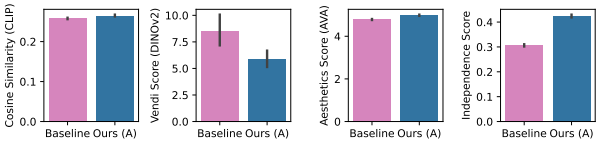


Fig. 4. Comparison of multiple score distributions. Refer to axes for metrics. Our framework clearly outperforms the baseline for all metrics except diversity (Vendi Score). We argue the additional constraints intrinsic to our task (of generating illusions) contributed to reduced diversity.

Our method significantly outperforms the baseline in all metrics except the Vendi Score, which is expected because, for our method, there are more constraints from the derived images applied during the generation process.

The score distributions of four variants of our method are presented in Fig. 5. Each row of Fig. 5 presents two metrics. The subfigures on the left-hand side show the overall performance of a specific method. In general, all methods perform similarly well in terms of Controllability (Cosine Similarity) and Diversity (Vendi Score) (the first two rows in Fig. 5). Method A shows significant advantages in Aesthetics (Aesthetics Score) and Methods A and D achieve relatively higher Independence Score.

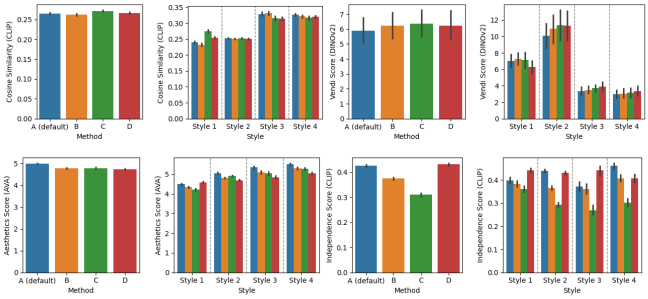


Fig. 5. Score distributions over methods (left) and styles (right). A, B, C, D (blue, orange, green, red respectively) stand for four variants of our method. Results indicate the significance of prompts for illusion generation.

A detailed look at different art styles is presented on the right-hand side of each row of Fig. 5, where different metrics respond diversely to different art styles. Controllability (Cosine Similarity) prefers Style 3 and Style 4 while the Diversity (Vendi Score) prefers Style 2. The Aesthetics Score and Independence Score are generally robust to the different styles. However, the Aesthetics Score prefers Style 4 slightly more than Style 1.

In conclusion, our quantitative evaluations show that the prompts used are more important than the chosen implementation, and there is no clear one-size-fits-all method. However, we observe that the optimal method depends on the art styles and subjects used. Therefore, one should carefully select a method when generating illusions in a specific art style. The main takeaways are: 1) no single method is universally optimal, and 2) the best method varies based on the art style and subject matter. A further study on subjects is available in the Appendix.

4.3 Discussions

Q1: Can we get better Diffusion Illusions by running for a longer time?

Yes. Fig. 6 shows the trend of Controllability (Cosine Similarity) and Aesthetics (Aesthetics Score) as the images from Section 4.2 are optimized. Relative time ranges from 0 (start) to 1 (end) of the optimization process. All four methods show a clear increase in scores as optimization progresses.

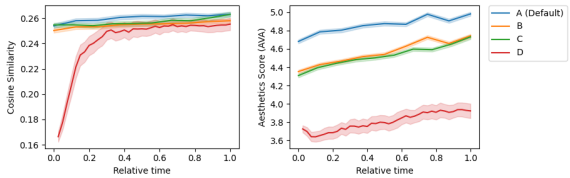


Fig. 6. CLIP Cosine Similarity (left) and Aesthetics Score (right) increase when optimizing for a longer time.

Q2: Is Independence Score a qualitatively valid metric?

Yes, as shown in Fig. 18. The figure shows how images with high independence scores align well with their textual prompts, while lower scores indicate less relevance between the overlay image and the subject. The first two examples have high scores, with each image aligning with its prompt. The third is not closely related to 'sofa', having a lower score. The last example has the lowest score, with the overlay visually biasing more towards cow' than 'bottle'.

Q3: Why do we use Fourier Feature Networks?

Earlier experiments optimizing prime images directly in pixel space resulted in information being encoded at very high frequencies and requiring pixel-perfect alignment to generate the intended derived images (see Fig. 13). While the result was pleasing when viewed digitally, it was impractical for real-world illusions. Motivated by previous arguments [Burgert et al. 2022a,b], we elect to use Fourier Features Network [Tancik et al. 2020] based parametric image representations. In the appendix we show parametrization ablations.

4.4 Failure Cases

Diffusion Illusions does not always manage to generate convincing illusions. Because of the difficult, over-constrained nature of this problem where we want to derive a greater number of derived images than we have primes, sometimes not all of them are well generated (see failure examples in Fig. 12).

In order to quantify this failure rate, we used a visual language model GPT4 [OpenAI 2023] to classify derived images from our hidden overlays illusion. We used 20 sets of prompts in Style 2, choosing subjects evenly from T^o, generating 20 illusions using method A for each set of prompts. We asked GPT4 to classify each one as a multiple choice question, allowing it to choose from 20 categories: T^o + 'person'.

GPT4 correctly classified 97% of the prime images, but only 56% of the hidden images. This indicates the derived hidden images were less clear than the prime images. In general, they are darker as well. It classified all 5 derived images for a given illusion correctly 53% of the time. Please see our appendix for more experimental details.

5 Limitations and Future Work

Diffusion Illusions is limited to generating optical illusions where prime images are arranged to yield various visual stimuli, as specified in our problem statement (Section 2). Other illusion types, like apparent motion, are out of scope. See the Twisting Squares Illusion in our appendix for an additional example within our domain.

The main drawbacks are slow inference time (over 10 minutes per illusion) and potential quality issues due to printing imperfections (see Fig. 17). Increasing the number of derived images from a small set of prime images also fundamentally limits output quality. For example, we could theoretically extend the flip illusion to four, or even eight viewing angles, but in doing so the quality of each derived image would decrease because of the limited amount of information contained in that single prime image. Other limitations include model biases (see ethics statement in supplementary).

Given the human-perceptual nature of illusions, we recognize the limitations of our automated metrics and plan to release user studies that measure ease of recognition (how well derived images match their targets), degree of concealment (how well the hidden overlays hide their target images), and preference ratings (which algorithm variants and prompting methods are most appealing) - using both physical and simulated illusions.

6 Conclusion

In this paper, we establish the formal definition of the problem of generating illusions and introduce Diffusion Illusions, a versatile pipeline designed for the generation of a diverse array of illusions. Complemented by comprehensive experiments conducted across multiple facets, we verify the effectiveness of our proposed method qualitatively and quantitatively. We also successfully fabricate the prime images in the real world. Other areas to explore include more types of illusion generation and creative ways to take advantage of diffusion models.

Stony Brook University
rburgert@cs.stonybrook.edu

References

Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. 2020. Evaluation Metrics for Conditional Image Generation. *International Journal of Computer Vision* 129 (2020), 1712 – 1731. https://api.semanticscholar.org/CorpusID:216553817

Eyal Betzalel, Coby Penso, Aviv Navon, and Ethan Fetaya. 2022. A Study on the Evaluation of Generative Models. *ArXiv* abs/2206.10935 (2022). https://api.semanticscholar.org/CorpusID:249926935

E. G. Boring. 1930. A new ambiguous figure. *The American Journal of Psychology* 42 (1930), 444–445. https://doi.org/10.2307/1415447 Place: US Publisher: Univ of Illinois Press.

Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael Ryoo. 2023. Diffusion Illusions: Hiding Images in Plain Sight. https://ryanndagreat.github.io/Diffusion-Illusions. Accessed: 2024-04-16.

Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S. Ryoo. 2022a. Peekaboo: Text to Image Diffusion Models are Zero-Shot Segmentors. *ArXiv* abs/2211.13224 (2022). https://api.semanticscholar.org/CorpusID:253801576

Ryan Burgert, Jinghuan Shang, Xiang Li, and Michael Ryoo. 2022b. Neural Neural Textures Make Sim2Real Consistent. In *Proceedings of the 6th Conference on Robot Learning*. https://tritonpaper.github.io

Kartik Chandra et al. 2022. Designing perceptual puzzles by differentiating probabilistic programs. In *ACM SIGGRAPH 2022 Conference Proceedings*.

Kartik Chandra, Chuma Kabaghe, and Gregory Valiant. 2021. Beyond Laurel/Yanny: An Autoencoder-Enabled Search for Polyperceivable Audio. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. *Robotics science and systems (RSS)* (2023).

Ming-Te Chi et al. 2008. Self-animating images: Illusory motion using repeated asymmetric patterns. *ACM Transactions on Graphics (TOG)* 27, 3 (2008), 1–8.

Hung-Kuo Chu et al. 2010. Camouflage images. *ACM Transactions on Graphics* 29, 4 (2010), 51.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. 2023. Vision Transformers Need Registers. *arXiv preprint arXiv:2309.16588* (2023).

Prafulla Dhariwal and Alex Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. *ArXiv* abs/2105.05233 (2021).

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88 (2010), 303–338.

Dan Friedman and Adji Bousso Dieng. 2022. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410* (2022).

Daniel Geng, Inbum Park, and Andrew Owens. 2023. Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models. *arXiv:2311.17919* (Novemeber 2023). https://arxiv.org/abs/2311.17919

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239 [cs.LG]

Douglas R Hofstadter. 1985. Metafont, Metamathematics, and Metaphysics: Comments on Donald Knuth's Article "The Concept of a Meta-Font". *Metamagical themas: Questing for the essence of mind and pattern* (1985), 274–278.

Kai-Wen Hsiao, Jia-Bin Huang, and Hung-Kuo Chu. 2018. Multi-view wire art. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 1–11. https://doi.org/10.1145/3272127.3275070

Joseph Jastrow. 1899. The Mind's Eye. *Popular Science Monthly* (1899), 299–312.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, et al. 2023a. Holistic Evaluation of Text-to-Image Models. *arXiv preprint arXiv:2311.04287* (2023).

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, Minguk Kang, Taesung Park, Jure Leskovec, Jun-Yan Zhu, Li Fei-Fei, Jiajun Wu, Stefano Ermon, and Percy Liang. 2023b. Holistic Evaluation of Text-To-Image Models. http://arxiv.org/abs/2311.04287 arXiv:2311.04287 [cs].

Xiang Li, Varun Belagali, Jinghuan Shang, and Michael S Ryoo. 2023. Crossway Diffusion: Improving Diffusion-based Visuomotor Policy via Self-supervised Learning. *arXiv preprint arXiv:2307.01849* (2023).

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744* (2023).

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *arXiv preprint arXiv:2304.08485* (2023).

Jörn Loviscach. 2010. Finding Approximate Ambigrams and Making them Exact. In *Eurographics (Short Papers)*.

Li-Qian Ma et al. 2013. Change blindness images. *IEEE Transactions on Visualization and Computer Graphics* 19, 11 (2013), 1808–1819.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. arXiv:2108.01073 [cs.CV]

Steve Mould. 2024. Self-assembling material pops into 3D. YouTube. Video available at https://www.youtube.com/watch?v=vrOjy-v5JgQ.

Naila Murray, Luca Marchesotti, and Florent Perronnin. 2012. AVA: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2408–2415.

Mizuho Nakajima and Yasushi Yamaguchi. 2004. Picture Illusion by Overlap. In *ACM SIGGRAPH 2004 Posters*. 56.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In *ICML*.

Michael E. R. Nicholls, Owen Churches, and Tobias Loetscher. 2018. Perception of an ambiguous figure is affected by own-age social biases. *Scientific Reports* 8 (Aug. 2018), 12661. https://doi.org/10.1038/s41598-018-31129-7

Aude Oliva, Antonio Torralba, and Philippe G Schyns. 2006. Hybrid images. *ACM Transactions on Graphics (TOG)* 25, 3 (2006), 527–532.

OpenAI. 2023. GPT-4. https://openai.com/. Accessed: November 16, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

Shigeru Owada and Jun Fujiki. 2008. Dynafusion: A modeling system for interactive impossible objects. In *Proceedings of the 6th International Symposium on Non-Photorealistic Animation and Rendering*.

Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz. 2012. The magic lens: refractive steganography. *ACM Transactions on Graphics* 31, 6 (Nov. 2012), 1–10. https://doi.org/10.1145/2366145.2366205

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. https://ar5iv.org/abs/1912.01703

Maxine Perroni-Scharf and Szymon Rusinkiewicz. 2023. Constructing Printable Surfaces with View-Dependent Appearance. In *ACM SIGGRAPH 2023 Conference Proceedings (SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3588432.3591526

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arXiv* (2023).

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *ArXiv* abs/2209.14988 (2022).

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. https://doi.org/10.48550/ARXIV.2204.06125

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. *ICML* (2021).

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

Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2021a. Palette: Image-to-Image Diffusion Models. https://doi.org/10.48550/ARXIV.2111.05826

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv:2205.11487* (2022).

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. 2021b. Image Super-Resolution via Iterative Refinement. https://doi.org/10.48550/ARXIV.2104.07636

Noufal Samsudin. 2023. Generating Ambigrams using Deep Learning: A Typography Approach. https://github.com/kvsnoufal/ambigramPytorch unpublished work.

Takahiro Shirakawa and Seiichi Uchida. 2023. Ambigram Generation by A Diffusion Model. arXiv:2306.12049 [cs.CV]

Matthew Tancik. 2023. Illusion Diffusion: optical illusions using stable diffusion. https://github.com/tancik/Illusion-Diffusion unpublished work.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. arXiv:2006.10739 [cs.CV]

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. 2022. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers.

Ludwig Wittgenstein. 1953. *Philosophical investigations.* Macmillan, Oxford, England. (Part 2, Section 11).

Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. 2023. Navigating Text-To-Image Customization: From LyCORIS Fine-Tuning to Model Evaluation. *arXiv preprint arXiv:2309.14859* (2023).

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. 2022. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *arXiv:2206.10789* (2022).

Boheng Zhao, Rana Hanocka, and Raymond A. Yeh. 2023. AmbiGen: Generating Ambigrams from Pre-trained Diffusion Model. arXiv:2312.02967 [cs.CV]

Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Yong Yu, and Weinan Zhang. 2023. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223* (2023).

Stony Brook University
rburgert@cs.stonybrook.edu

Fig. 7. **Flip Illusion Examples:** Please view these images upside-down as well as right-side-up to see two different subjects. Note: In this illusion, $d_1 = p_1$. *These images are high resolution - please zoom in!*

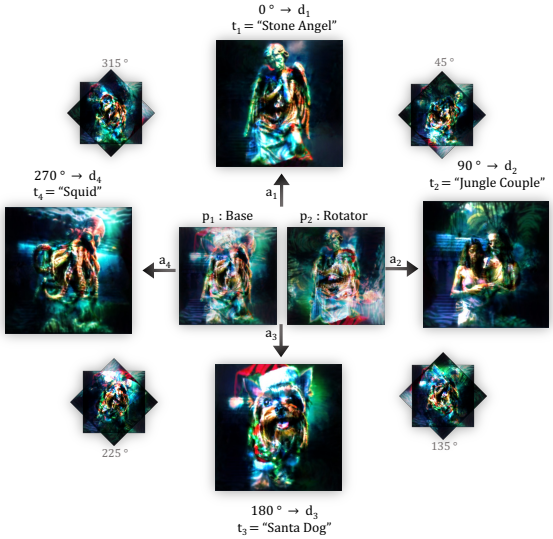


Fig. 8. We print our rotation overlay illusions onto two transparent sheets: the "rotator" image is placed on a "base" image over a backlight. Then, as the rotator spins, we derive four different images.

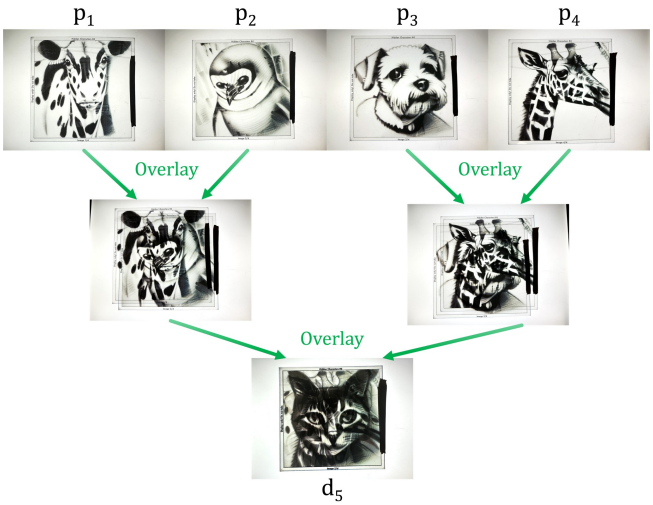


Fig. 9. We print our hidden overlay illusions onto four transparent sheets, and stack them on top of each other. These seemingly normal pictures of animals (cow, penguin, dog and giraffe) reveal a cat when overlaid and placed in front of a backlight. *Please note that these are all real photographs.*

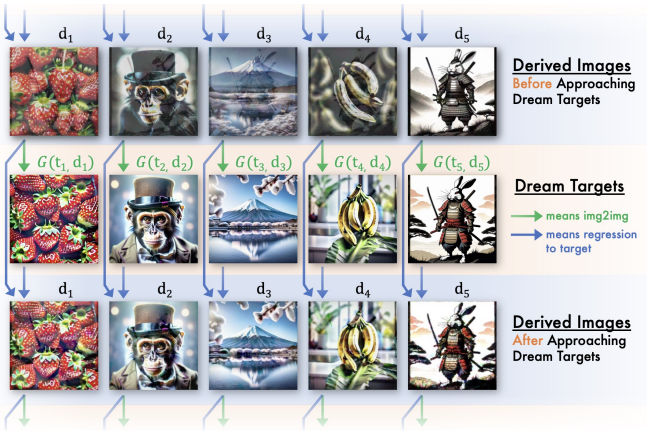


Fig. 10. We depict the dream-target loss above. It is an iterative process, refining derived images using SDEdit to create target images, which the derived images are then regressed to with gradient descent. The derived images look more like the targets after approaching them than before.

Fig. 11. An overlay illusion at different phases of the optimization process. Note how the artifacts and over-saturation caused by score distillation loss (phase 1) are fixed by dream target loss (phase 2).

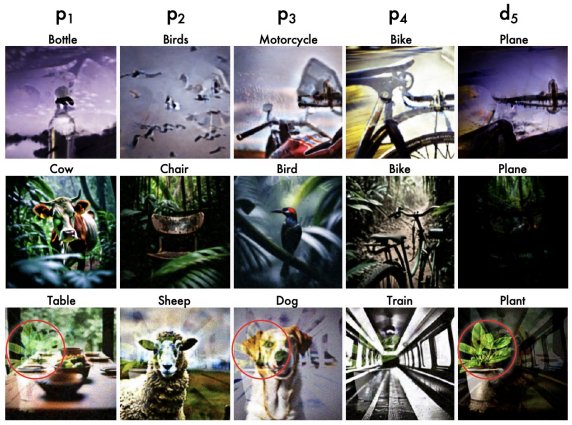


Fig. 12. Failure modes for Hidden Overlay illusions. Top row: not all subjects are easily recognizable. Middle row: the hidden image became too dark. Bottom row: the plant is not well hidden - it bleeds into the prime images.

Fig. 13. A Hidden Overlay image with prime images optimized directly in pixel space. While high-frequency encoding of the hidden image results in less perceivable interference in each individual image, it results in a brittle illusion that is disrupted without pixel-perfect printing and alignment.

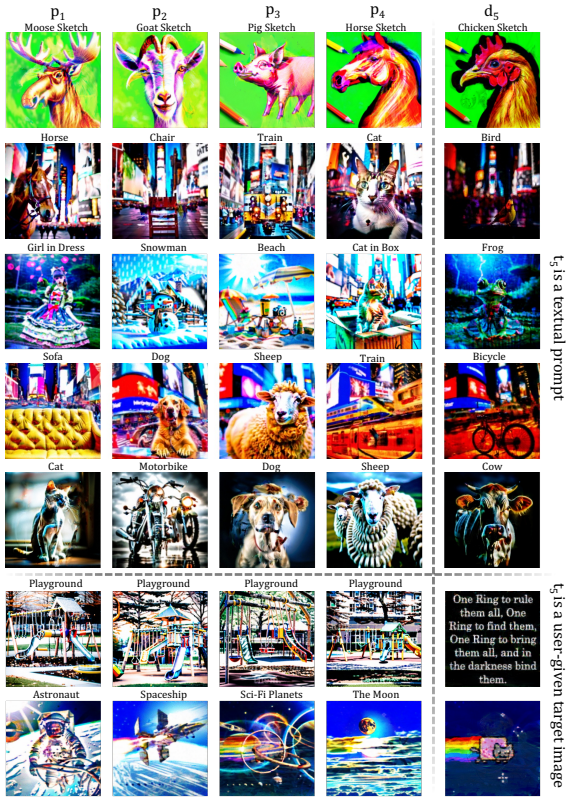


Fig. 14. **Hidden Overlay Examples:** On the left are the four prime images p_1, p_2, p_3, p_4 and on the right is the derived image $d_5 = p_1 \cdot p_2 \cdot p_3 \cdot p_4$, which simulates overlaying them over a backlight. Note: In this illusion, $d_i = p_i$ for $i \in 1 \ldots 4$. *These images are high resolution - please zoom in!*

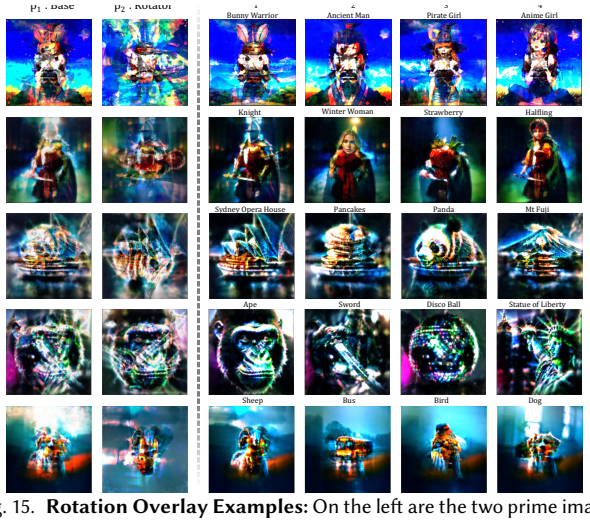


Fig. 15. **Rotation Overlay Examples:** On the left are the two prime images p_1, p_2, and on the right are the four derived images $d_{1\ldots4}$ that are obtained by taking the product of the primes, simulation of them overlaid on a backlight. *These images are high resolution - please zoom in!*

Fig. 16. **Overlay Illusion Fabrication**: Paper can be used instead of transparencies., in which case a stronger backlight may be needed. Printer color inaccuracy affects illusion quality, which can be caused by low toner levels.

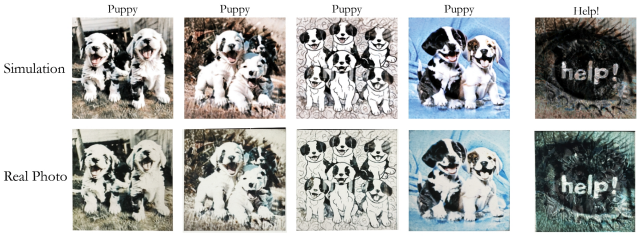


Fig. 17. The colors shift after printing out Hidden Overlay Illusion images. First row: digital copy of the images and the overlay simulation. Second row: real-world photos of the printed images.

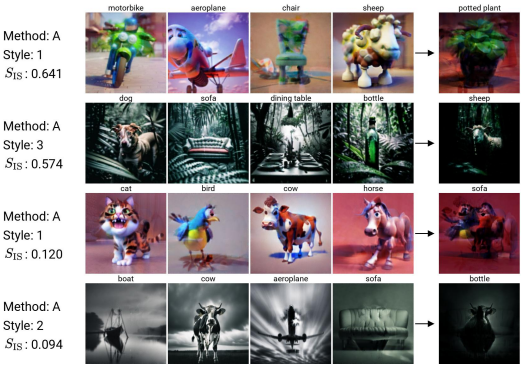


Fig. 18. Four illusions randomly selected with diverse independence scores. For each row, the subject is listed above, and the method, style, and independence scores are on the left. The four middle images are primes that derive the overlay on the right.

Stony Brook University
rburgert@cs.stonybrook.edu

```
#           _   ___  _  _       __ ___  _
#           | \  |  |_ |_  | | (_   |  / \ |\ |
#           |_/ _|_ |  |   |_| __) _|_ \_/ | \|
#           ___            __  ___  _        __
#            |  |  |  | | (_    |  / \ |\ | (_
#           _|_ |_ |_ |_| __)  _|_ \_/ | \| __)
#         _   __  _       _   _      _  _    _   _
#        |_) (_  |_ | |  | \ / \    /  / \ |  \ |_
#        |   __) |_ |_|  |_/ \_/    \_  \_/ |_/  |_

###### PART 1: Initialization
if ILLUSION_TYPE=='FLIP':
    n = 1 #Number of Prime images
    m = 2 #Number of Derived images
    A = [
        #A stands for Arrangements
        lambda P: P[0],
        lambda P: P[0].rot180(),
    ]
    W = [1, 1] # Importance of each derived image
    T = ['Dog', 'Sloth']

if ILLUSION_TYPE=='ROTATE':
    n = 2 #Two Prime Images: Base, Rotator
    m = 4 #Four Derived Images
    k = 2 #The backlight brightness constant
    A = [
        lambda P: k*P[0]*P[1],
        lambda P: k*P[0]*P[1].rot90(),
        lambda P: k*P[0]*P[1].rot180(),
        lambda P: k*P[0]*P[1].rot270(),
    ]
    W = [1, 1, 1, 1]
    T = ['Dog', 'Cat', 'Man', 'Woman']

if ILLUSION_TYPE=='HIDDEN':
    n = 4 #Two Prime Images: A, B, C, D
    m = 5 #Four Derived Images:
          # A, B, C, D, Hidden
    k = 3 #The backlight brightness constant
    A = [
        lambda P: P[0],
        lambda P: P[1],
        lambda P: P[2],
        lambda P: P[3],
        lambda P: k*P[0]*P[1]*P[2]*P[3],
    ]
    W = [1, 1, 1, 1, 3] # Prioritize the hidden image
    T = ['Dog', 'Penguin', 'Giraffe', 'Cow', 'Cat']
    ## OR, to use a QR code or another specific image...
    T = ['Dog', 'Penguin', 'Giraffe', 'Cow',
         load_image('qr_code.png') ]

assert len(T) == len(A) == len(W) == m

# Initialize all prime images
P = [RgbFourierFeatureNetwork(resolution=(512,512))
     for _ in range(n)]
# Initialize our latent diffusion model
F = StableDiffusion()
# We optimize the prime images via gradient descent.
optim = SGD(P.parameters())


###### PART 2: Helper Functions
def score_distill_loss(image, prompt):
    #Same loss proposed in DreamFusion -
    # but with a latent diffusion model
    image_latent = F.encode_image(image)
    timestep = random_int(0, F.max_timestep)
    noise = F.get_noise(timestep)
    noised_latent = F.add_noise(
        image_latent, noise, timestep
    )
    with torch.no_grad():
        text_embed = F.clip.embed(prompt)
        pred_noise = F.unet(
            noised_latent, text_embed, timestep
        )
    return abs(noise - pred_noise).sum()

def image_similarity(a, b):
    #Our image similarity metric
    return SSIM(a,b) - MSE(a,b)

def img2img(image, prompt, strength):
    #Based on SDEdit - simplified here
    #When strength=1, the entire image is replaced
    #When strength=0, nothing is changed
    image_latent = F.encode_image(image)
    timestep = int(strength * F.max_timestep)
    noise = F.get_noise(timestep)
    noised_latent = F.add_noise(
        image_latent, noise, timestep
    )

    #Perform diffusion as normal, but starting from
    #our noised_latent instead of pure noise
    diffused_latent = F.text_to_image(
        prompt,
        initial_latent=noised_latent,
        initial_timestep=timestep,
    )

    new_image = F.decode_image(diffused_latent)
    return new_image


###### PART 3: Optimization

#Phase 1: Score Distillation Loss
for iteration in range(2000):
    loss = 0
    for a,t,w in zip(A,T,W):
        # Derived image d
        # comes from an arrangement of prime images
        d = a(P)
        if isinstance(t, str):
            loss += w * score_distill_loss(d, t)
        elif is_image(t):
            # For hiding custom images such as QR codes
            loss -= w * image_similarity(d, t)
    optim.update(loss) # Take a gradient descent step

#Phase 2: Dream-Target Loss

#Start from strength = .90 instead of 1
# in order to use the results from Phase 1
schedule = [.90, .80, .70 ... .30, .20, .10]

for strength in schedule:
    # Define the image translation function
    G = lambda text,image: img2img(text,image,strength)

    # Step 1: Set our Dream-Targets
    Z = []
    for a, t in zip(A,T):
        if isinstance(t, str):
            # Tweak a derived image to get a new target
            d = a(P)
            z = G(t, d)
        elif is_image(t):
            #Use a predefined target (e.g. a QR code)
            z = t
        Z.append(z)

    # Step 2: Approach our Dream-Targets
    for iteration in range(1000):
        #Optimize P so that D approaches T

        loss = 0
        for a,z,w in zip(A,Z,W):
            d = a(P)
            loss -= w * image_similarity(d, z)

        # Take a gradient descent step
        optim.update(loss)


###### PART 4: Fabrication

#We're done! Return the primes -
# and print them out physically!
printed_P = send_to_laser_printer(P)

#Oh, and also, make sure someone uses them...
fun = have_human_arrange_the_illusions(printed_P)
```

A Twisting Squares Illusion

Diffusion Illusions is a very general framework and can generate more than just the three illusion types presented in the main paper. In Fig. 20, we show the Twisting Squares illusion - where we break up an image into a grid of tiles, and spin these tiles together as a mechanical linkage. This type of illusion was inspired by the mechanical linkage shown in Mould [2024].

It consists of one prime image and two derived images, where the

Like all other illusions printed in this paper, it can be physically fabricated - we have 3d printed these linkages and they are shown in Fig. 19.

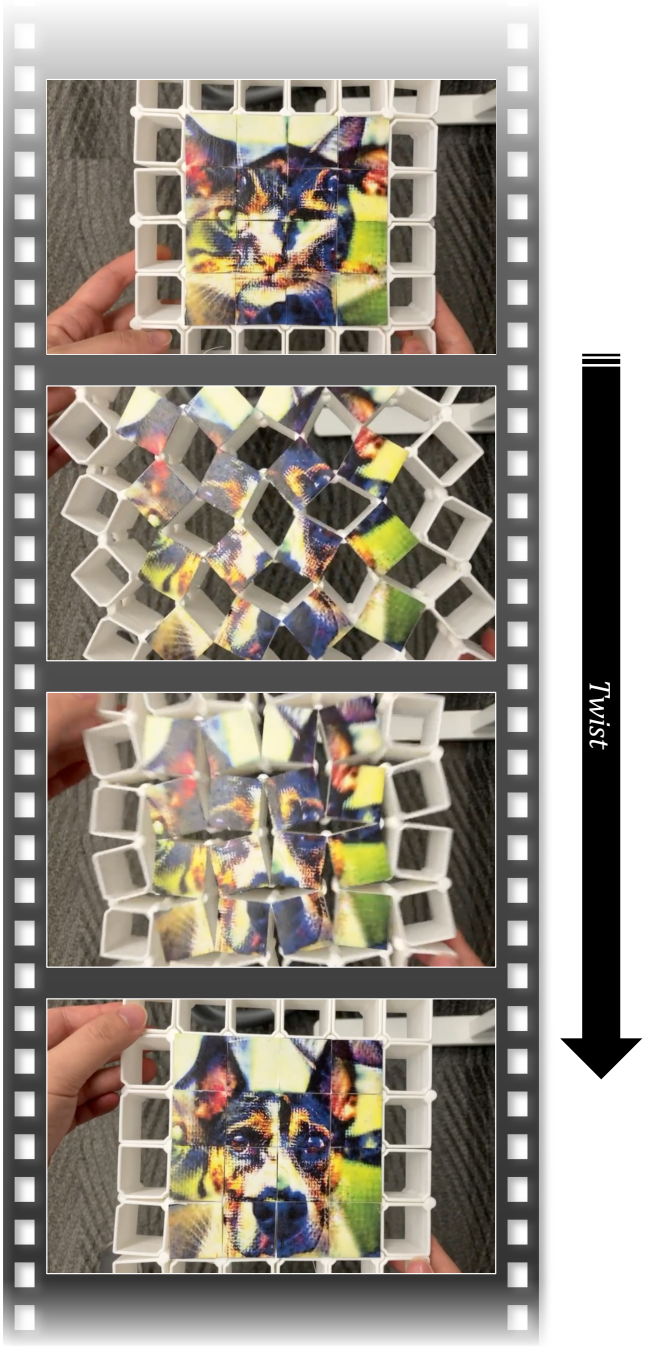


Fig. 19. The Twisting Squares illusion is made using a 3d printer, and the tiles are put on top. When one tile is twisted, all the others move together - where every odd tile spins 90°clockwise and every even tile spins 90°counterclockwise.

B Implementation Details

B.1 Brightness Constant

In the actual implementation, you'll see we multiply our derived overlay images by a scalar "brightness constant" k, that is chosen based on the type of illusion. This constant is visible in the given pseudocode — please see how it is used there. This is because in real life, when viewing the hidden overlay and rotating overlay illusions, the backlight can be arbitrarily bright. Without this term, the derived images obtained from overlaying other images would necessarily be darker than their prime images, because images have values between 0 and 1, and the product between any two numbers between 0 and 1 are guaranteed to be 1 or less.

Because the hidden character illusion deals with 4 overlays, it benefits from a higher brightness constant than the rotation overlay illusion ($k = 3$ vs $k = 2$). The brightness constant k is not applicable for the flip illusion, as it does not deal with overlay transparencies.

B.2 Static Targets

When creating an illusion, usually text prompts are used for all values of T. However, it is possible to specify a fixed image target by setting T as an image instead. This allows us to hide specific images such as QR codes, nyan cat, pentagrams, or even entire segments of text (see Fig. 14). Instead of applying score distillation loss for example, we regress torwards that given image. Please see the below pseudocode for an exact implementation.

B.3 Libraries

We use SDXL as our latent diffusion model [Podell et al. 2023]. Our SDEdit implementation of SDXL comes from von Platen et al. [2022], using PyTorch [Paszke et al. 2019]. Our implementation of Fourier Feature Networks is directly adapted from the TRITON [Burgert et al. 2022b], using the default parameters for their Neural Neural Textures. Our implementation of Score Distillation Loss comes from Peekaboo [Burgert et al. 2022a].

C Extended Quantitative Evaluation

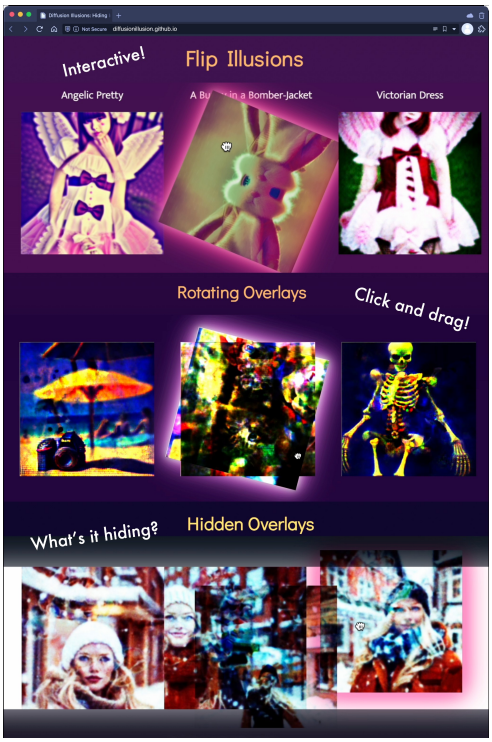
C.1 Additional Hidden Overlay Ablations

Extending our Hidden Overlay ablations in Section 3, we have added six extra ablations. Their details are summarized in Table 3. Extending our scoring results in Fig. 5 and Fig. 4, we compare all of our ablations in Fig. 30.

Ablations **I,J,K,L** compare different image parametrizations. Our default image parametrization is a Fourier Feature Network, discussed in Section 4.3. By comparison, we also try using a 512 × 512 resolution pixelwise representation, which we call the "Raster"

Stony Brook University
rburgert@cs.stonybrook.edu

Fig. 20. The twisting squares illusion breaks an image into a mechanical linkage of tiles, and twists them to create a new image. Every odd tile spins 90°clockwise and every even tile spins 90°counterclockwise.



https://diffusionillusion.github.io

Fig. 21. Please visit our project page — it contains fully interactive simulations of all illusions in this paper, as well as many more!

parametrization (visualized in Fig. 13). Additionally, to see if we can reduce grainy artifacts by using a lower raster resolution, we introduce ablations with the "Low-Res 64 × 64 Raster" parametrization. As it turned out, this does not help with the grainy artifacts.

Ablations **G,H** change the number of prime images - using two or three primes instead of the default four.

We have also included visualizations for new ablations in Fig. 22 and Fig. 23.

Abaltions **J,K** are visualized in Fig. 22. When using score distillation loss alone (without dream target loss), very poor results can be expected. Since the gradients pass through the latent diffusion model's encoder, instead of creating a visually pleasing image, it creates an adversarial image to satisfy the encoder. Decreasing the resolution of the image does appear to help a little bit, but the resulting image is still grainy and full of artifacts.

Dream target loss does not suffer from this problem as badly, and in Fig. 23 we showcase the effect of a single step of dream target loss, which defines the difference between methods **D** and **E**. Dream target loss helps to smooth out artifacts and increase detail.

C.2 Quantitative Evaluation Details

This section provides more details and additional experiments regarding benchmarking the derived images of Hidden Overlay Illusion and Rotation Overlay Illusion.

Textual Prompts The set of image styles T^s is listed as follow where <s> stands for the *subject* token:

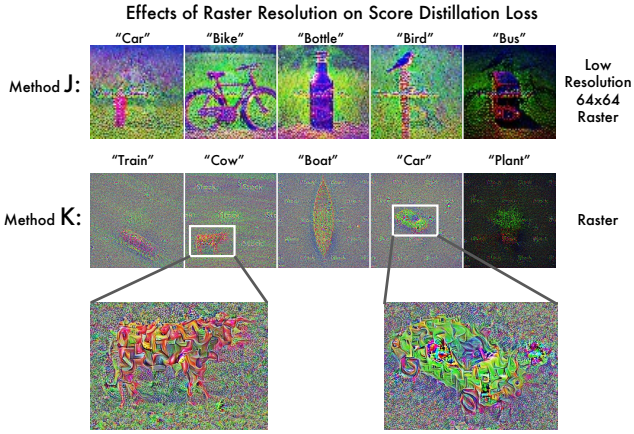


Fig. 22. When using only score distillation loss on a raster parametrization (without dream target loss and without using Fourier Feature Networks), very poor results can be expected. Note that adding dream target loss helps mitigate these artifacts (see Fig. 13 and Fig. 24), but does not entirely eliminate them.

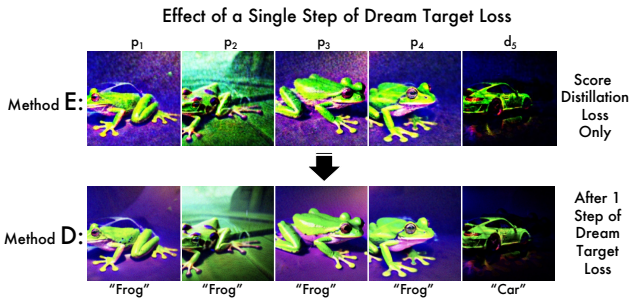


Fig. 23. The difference between method D and method E is a single dream target step. Note how artifacts are reduced.

Style 1: *3d pixar style render animation of a* <s>
Style 2: *an award winning photograph of a* <s>
Style 3: *an award winning photograph of a* <s> *in the deep jungle*
Style 4: *an award winning photograph of a* <s> *in times square*

The subject set T^o contains subjects from the PASCAL VOC dataset [Everingham et al. 2010]: *aeroplane*, *bicycle*, *bird*, *boat*, *bottle*, *bus*, *car*, *cat*, *chair*, *cow*, *dining table*, *dog*, *horse*, *motorbike*, *person*, *potted plant*, *sheep*, *sofa*, *train*, *tv/monitor*.

Additional Evaluation Metrics We further extend the evaluation introduced in the main paper by including more metrics in each aspect:

- *Controllability* We take advantage of a vision language model (VLM) LLaVA-1.5 [Liu et al. 2023a,b] to measure the similarity between the image and the textual prompt. The instruction sent to the VLM is

Stony Brook University
rburgert@cs.stonybrook.edu

Label	Model	w_5	SDL Steps	DTL Steps	Primes	Parametrization	Explanation
A	SDXL	3	500	8	4	Fourier Features	Our Main Method
B	SDXL	1	500	8	4	Fourier Features	Equal Weights
C	SD15	3	500	8	4	Fourier Features	Stable Diffusion 1.5
D	SDXL	3	4000	1	4	Fourier Features	Almost entirely SDL
E	SDXL	3	4000	0	4	Fourier Features	Pure Score-Distillation-Loss
F	SDXL	3	0	20	4	Fourier Features	Pure Dream-Target Loss
G	SDXL	3	500	8	2	Fourier Features	Only two prime images
H	SDXL	3	500	8	3	Fourier Features	Only three prime images
I	SDXL	3	500	8	4	Raster	Raster Parametrization
J	SDXL	3	4000	0	4	Raster	Raster Parametrization with Pure Score-Distillation-Loss
K	SDXL	3	4000	0	4	Low-Res 64×64 Raster	Low-Resolution Raster Parametrization with Pure Score-Distillation-Loss
L	SDXL	3	500	8	4	Low-Res 64×64 Raster	Low-Resolution Raster Parametrization
Base	SDXL	3	0	1	4	Fourier Features	Our Baseline

Table 3. A comprehensive summary of our ablation methods, continuing from Table 2. SDL stands for Score Distillation Loss and DTL for Dream Target Loss.

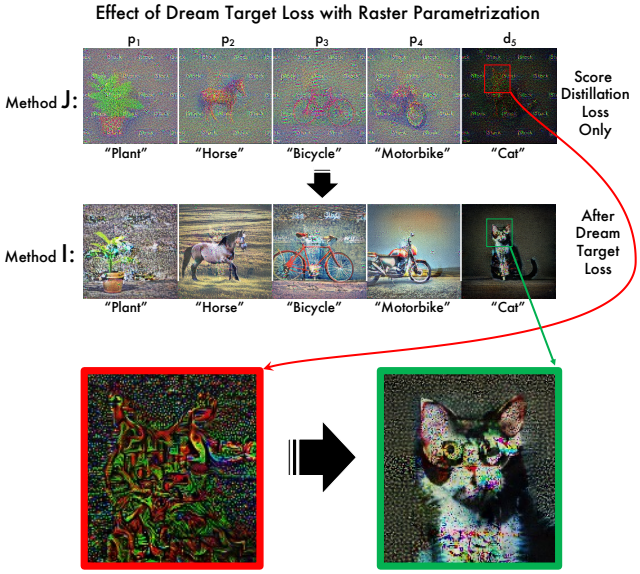


Fig. 24. Dream target loss helps to reduce artifacts when using the raster representation, but does not eliminate them. The high frequency pixel grain makes them hard to print and align in real life, as opposed to our fourier feature representation. Note the zoomed in cat face from the hidden images, and how its fidelity is improved.

> *Give a single score from 0 to 10 regarding how well the image looks like a* <s>. *A higher score means the image generally looks similar to a* <s>. *Only return the score.*

where <s> stands for the *subject* token and it will substituted by the actual subject for a specific image.

- *Diversity* Recent research [Darcet et al. 2023] suggests that the feature from the original DINOv2 might suffer from abnormal patches corresponding to the plain areas of the image. Therefore, we report the *Vendi Score* using the feature from DINOv2+reg [Darcet et al. 2023].
- *Aesthetics* Similar to Controllability, we collect an aesthetics score from LLaVA-1.5 using the following instruction:

  > *Give a single score from 0 to 10 regarding how well this image looks. A higher score means the image generally looks more natural and has fewer artifacts. Only return the score.*

In all metrics, the vision encoder of CLIP and the backbone of all DINO variants is a ViT-L/14 [Dosovitskiy et al. 2020]. The version of LLaVA-1.5 we utilized is fine-tuned from Vicuna-13B.

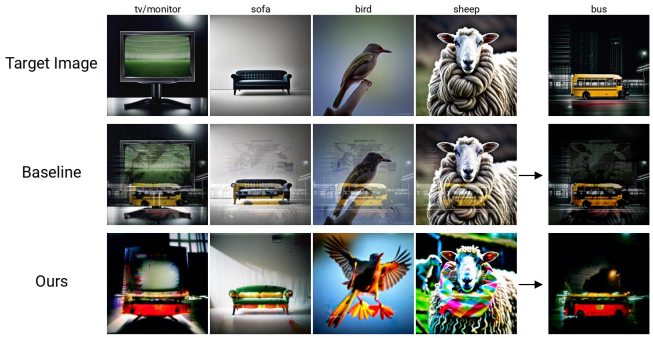


Fig. 25. Examples of our method and the baseline, starting from the same target image. Note how in the baseline, you can see the sheep in the bus image and the bus in the sheep image - which is why its independence score is lower.

C.3 Extended Results of Hidden Overlay Illusion

Fig. 25 presents comparative examples between the proposed method and the established baseline, starting from the same target image. The images from the baseline are heavily interfered with by others in the same group and the overlay image.

Fig. 26, Fig. 27, Fig. 28 and Fig. 29 show full evaluation results of the derived images from baseline and four variants of our method. The advantages of our method compared to the baseline are further supported by the new metrics introduced in this section, like better Controllability and Aesthetics Score from LLaVA (see Fig. 26). Meanwhile, LLaVA has relatively less bias on art styles and different subjects (Fig. 27 and Fig. 29)

C.4 Results of Rotation Overlay Illusion

We further benchmark the performance of Rotation Overlay Illusion. The evaluation follows the same protocol as the Hidden Overlay Illusion except that each group of Rotation Overlay Illusion images only has 4 derived images, which require 4 textual prompts at a time and we focus on one style:

> *a beautiful award-winning royalty-free full-frame stock photo of an isolated* <s>.

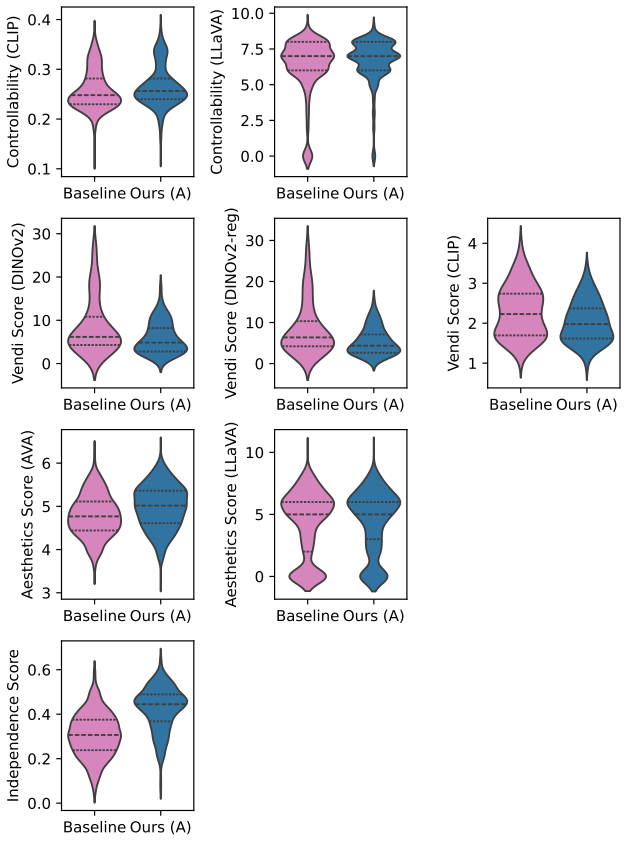


Fig. 26. Full evaluation on Hidden Overlay Illusion, each row is a group of thematically-aligned figures.

The result is presented in Fig. 31. Our method is significantly better than the baseline in terms of controllability (CLIP cosine similarity) and Aesthetics Score.

Ablation on the Number of Derived Images In this paper, by default we discuss a challenging rotation overlay illusion task where two prime images need to 'encode' four derived images. In this section, we conduct an ablation on the number of derived images. We perform an ablation study on the number of derived images, specifically focusing on cases with 2 to 4 derived images. Our hypothesis posits that reducing the number of derived images eases generation constraints, potentially enhancing image quality. This is corroborated by Fig. 32, which demonstrates improved image-text alignment and aesthetic scores in simpler tasks. Conversely, we observe a divergent trend in diversity, suggesting the interference between multiple derived images. Fig. 33 presents a qualitative comparison between problem formulations.

C.5 Failure-rate Experiment Details

Continuing the experiment outlined in the main text Section 4.4. Please see Table 4 for a complete list of prompts, and see Fig. 34 for a confusion matrix of hidden-image subjects. We would like to point out that this test does not capture how well the fifth is hidden (i.e. how hard it is to predict the hidden subject by looking at the primes without overlaying them). Since that is much more difficult to test automatically, we plan to do a user study (see Section 5). We created these illusions on a single NVIDIA RTX5000, with an average runtime of 11 minutes (see Fig. 35).

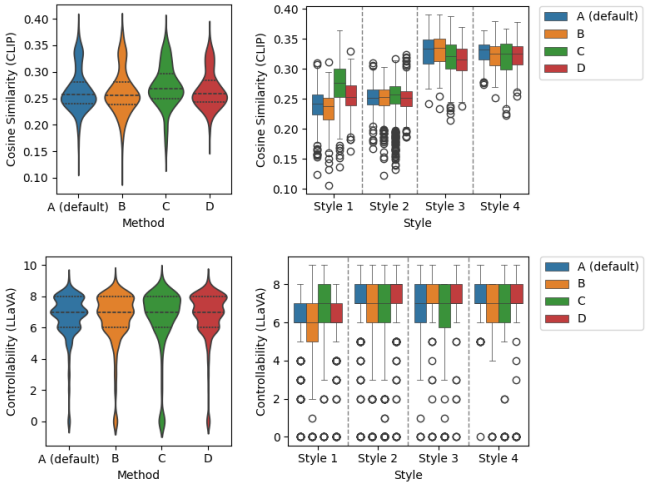


Fig. 27. Controllability score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

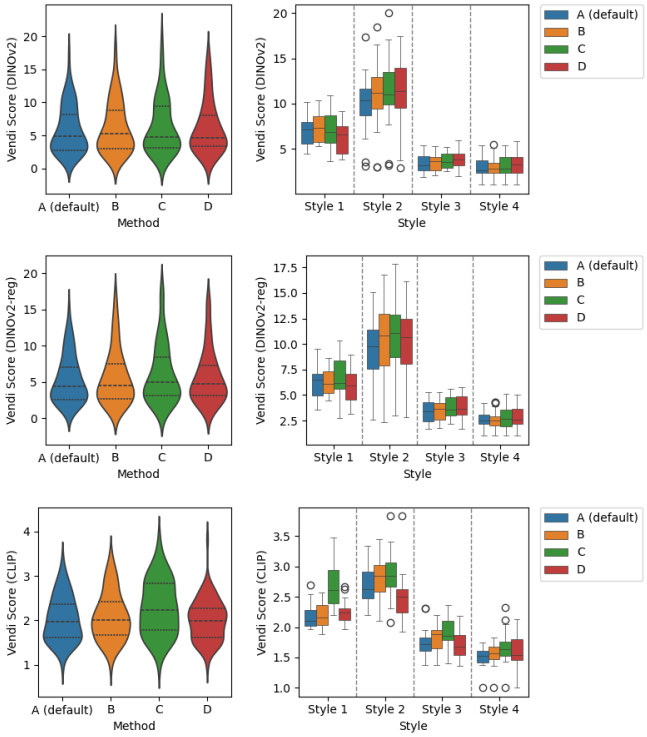


Fig. 28. Diversity score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

Stony Brook University
rburgert@cs.stonybrook.edu

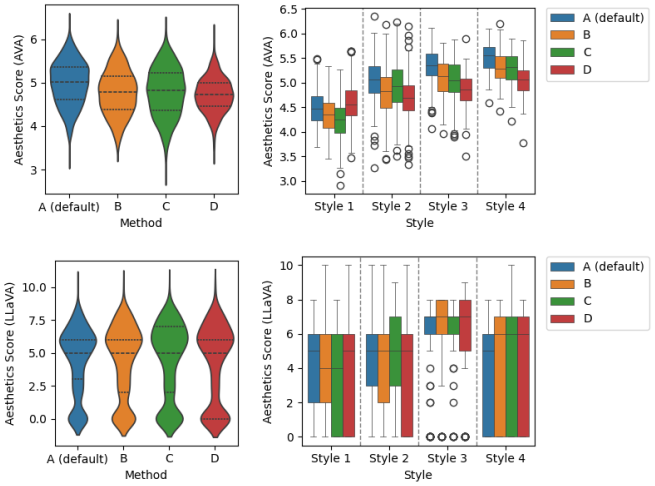


Fig. 29. Aesthetics score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

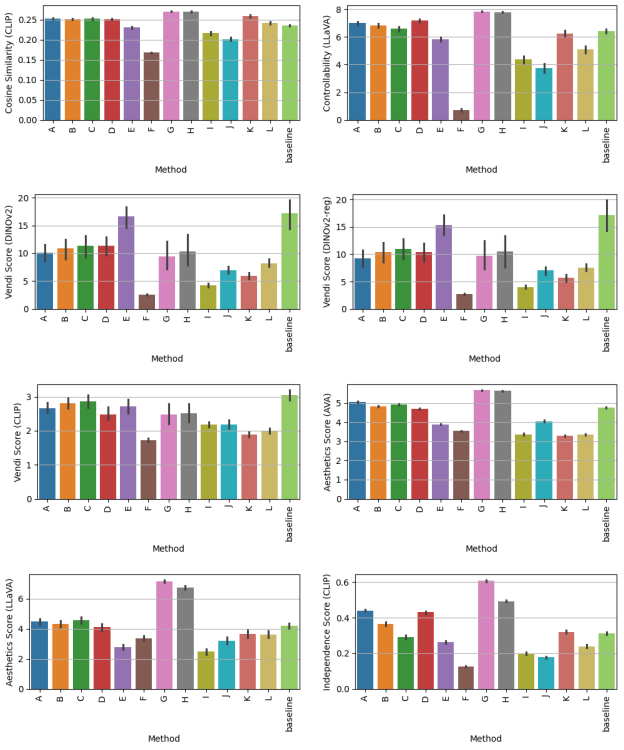


Fig. 30. Ablation results on Style 2. Method G and H yield overall better results due to fewer numbers of prime image limits.

The following textual prompt was given to GPT4 in addition to each derived image, all classified completely independent of one another

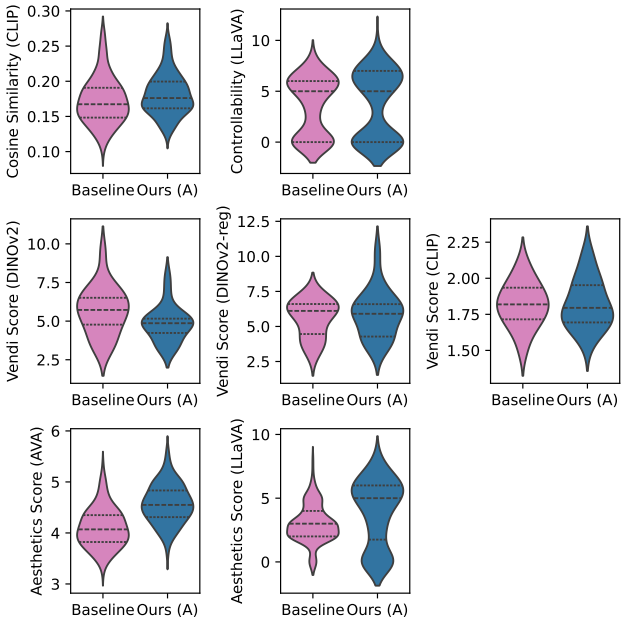


Fig. 31. Full evaluation on Rotation Overlay Illusion, each row is a group of thematically-aligned figures.

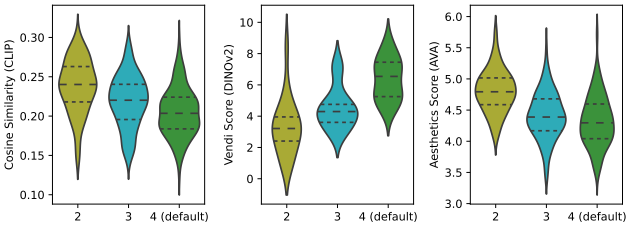


Fig. 32. Ablation on the number of the derived images in Rotation Overlay Illusion

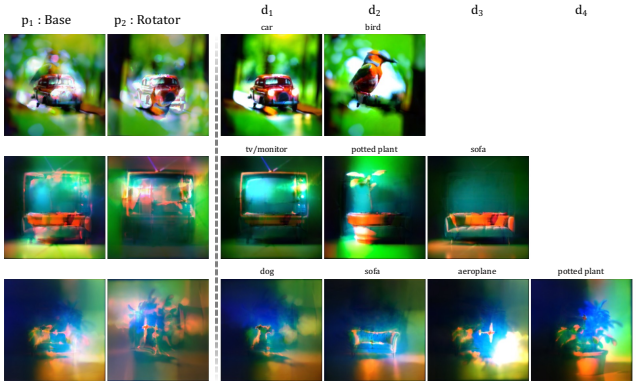


Fig. 33. Qualitative results of Rotation Overlay Illusions with different numbers of derived images

#	A	B	C	D	Z
1	aeroplane	sheep	cow	chair	car
2	bicycle	chair	train	sheep	boat
3	bird	car	cat	horse	person
4	boat	dog	table	bus	cow
5	bottle	television	motorbike	cat	bicycle
6	bus	cow	boat	car	plant
7	car	cat	bird	bottle	horse
8	cat	table	car	aeroplane	sheep
9	chair	bird	television	dog	aeroplane
10	cow	motorbike	chair	train	bird
11	dog	sofa	aeroplane	plant	bottle
12	horse	bottle	dog	sofa	television
13	motorbike	train	bus	television	dog
14	person	bicycle	bottle	bird	bus
15	plant	horse	bicycle	motorbike	cat
16	sheep	bus	person	bicycle	train
17	sofa	boat	plant	person	chair
18	table	aeroplane	horse	boat	motorbike
19	television	plant	sofa	cow	table
20	train	person	sheep	table	sofa

Table 4. This table shows the sets of subjects used for our hidden overlay illusion failure-rate experiment. All prompts use Style 2, i.e "an award winning photograph of a <s>" where the subject is shown in this table. There are 20 sets of prompts used. Columns 'A','B','C','D' refer to the four prime images, and 'Z' refers to the fifth derived image obtained by overlaying the four primes.

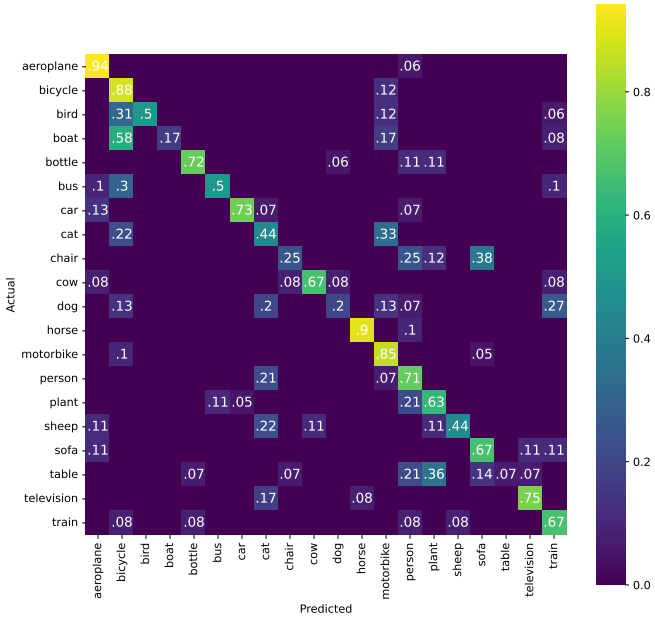


Fig. 34. A confusion matrix for the hidden images in our GPT4-recognition experiment, normalized by row. This confusion matrix is only for the hidden images, because the prime images were almost always correctly classified.

```
Your job is to classify this image as correctly
as possible. It may be abstract or it may
be realistic. But in any case, your job is
to guess the subject of this image. Your
response must be exactly one word, from the
following choices: 1. aeroplane 2. bicycle 3.
bird 4. boat 5. bottle 6. bus 7. car 8. cat 9.
chair 10. cow 11. dog 12. horse 13. motorbike
14. person 15. plant 16. sheep 17. sofa 18.
table 19. television 20. train
```

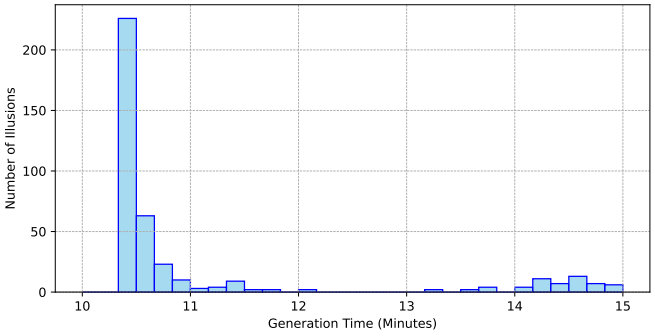


Fig. 35. Generation times for our hidden-overlay failure-rate experiment. On average, it took 11 minutes per illusion on a single NVIDIA A100 GPU.

D Fabrication Details

All of the illusions we present are realizable in the real world in physical form. To create a flip illusion in real life is quite easy - just print out one of the images onto a sheet of paper using a regular color laser printer.

The hidden overlay and rotation overlay can also be created with a basic color laser printer, and this is how we made all of the photographic examples in this paper. Searching "transparency film" online will yield many cheap transparent plastic films that are laser-printer compatible (a pack of 100 sheets sells for about $20 USD). However, after printing onto these overlays, it is useful to laminate them, as the ink can be easily scratched off. We do this with a basic thermal lamination machine that can also be purchased cheaply online.

After printing, laminating, and cutting the transparencies, stack them and put them over a light source. In this paper, we use a backlight taken from an old LCD monitor for our photos. However, any backlight will work, and holding the stack up to a bright window with sunlight also works well.

Since we model the light filtering process as multiplication, and multiplication is commutative, our modeling process assumes that the ordering of the layers does not matter. This is true in real life as well - with sufficient backlighting, you will get the same visual result whether transparency p_1 is on the top or on the bottom. However in practice, since some light reflects off the top transparency, it will not be perfectly identical.

Additionally, we found that inserting a thin layer of water between the transparent overlay sheets further enhances the visual effect, and slightly reduces the need for as strong of a backlight. We suspect this is because it eliminates the air gap between the sheets,

Stony Brook University
rburgert@cs.stonybrook.edu

leading to a smaller difference in the index of refraction. This is not necessary, but can somewhat enhance the clarity of the illusion.

We would like to point out, however: *transparencies are not strictly needed* to create overlay illusions. Regular paper can also be used, provided a strong enough backlight and use a sufficient amount of ink. We've included a comparison in figure Fig. 16.

E Additional Analysis

In Fig. 40, we show the optimization timeline for a hidden overlay illusion with timestamps and phases labeled. Artifacts are present throughout the score distillation phase, which are cleaned up in the dream target loss phase.

F Reproducibility and Ethics Statement

Reproducibility Statement Our work builds off open-source models whose pre-trained weights are publicly available. Our framework simply performs inference time optimizations to generate illusions. In our paper, we detail all specifics of our implementation (including PyTorch style pseudo-code) necessary to generate such illusions. Our code (and all material necessary to replicate results in paper) will be released publicly.

Ethics Statement A main ethical concern for any generative art model is that it will reduce the demand for human artists in its domain. Generating optical illusion artwork is a very difficult artistic task, and there are few artists that attempt it. Thus, the genre of illusions is currently relatively small and there is limited demand for illusions at present. Diffusion Illusions makes the generation of optical illusions accessible to the general public, making illusions more accessible to the layperson. We believe that, if anything, Diffusion Illusions and related works are likely to increase interest in illusions and the demand for human-created illusions as a result.

Secondly, our experiments utilize Stable Diffusion 1.5 and Stable Diffusion-XL models, and thus our reference implementation of the Diffusion Illusions pipeline will replicate any biases contained within these models. These models are trained on the LAION-2B(en) and LAION-5B datasets, and may over-represent English-language or Western content. The Stable Diffusion 1.5 and Stable Diffusion-XL models are intended for research purposes only, and thus our reference implementation should also be used exclusively for research and informative purposes. Some recent models, including DeepFloyd, are licensed for limited production use and our pipeline easily generalizes to them; however, they have higher system requirements.

Another potential ethical concern is that this will be used to hide information stenographically for nefarious purposes, although we deem this situation unlikely to be an issue in practice.

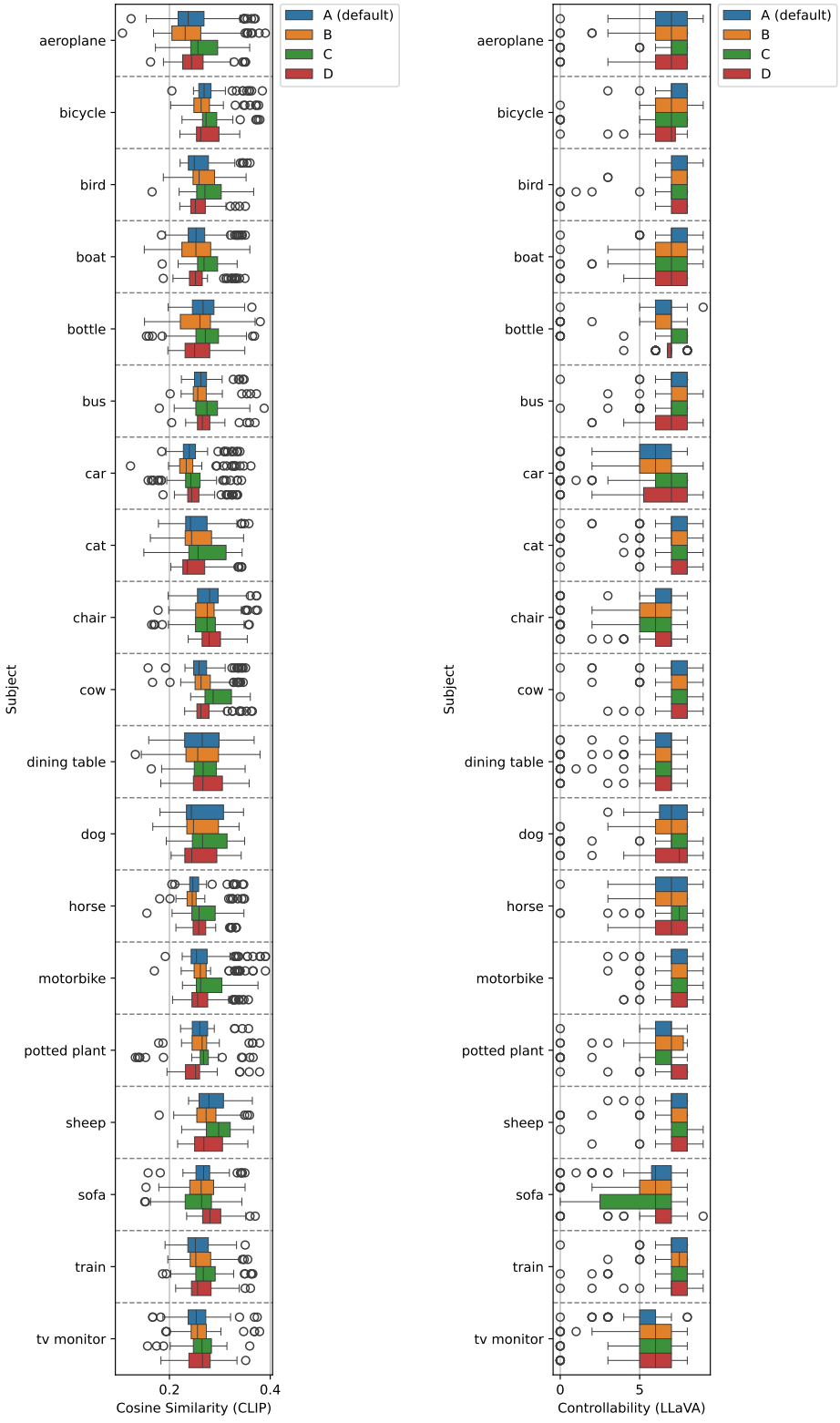
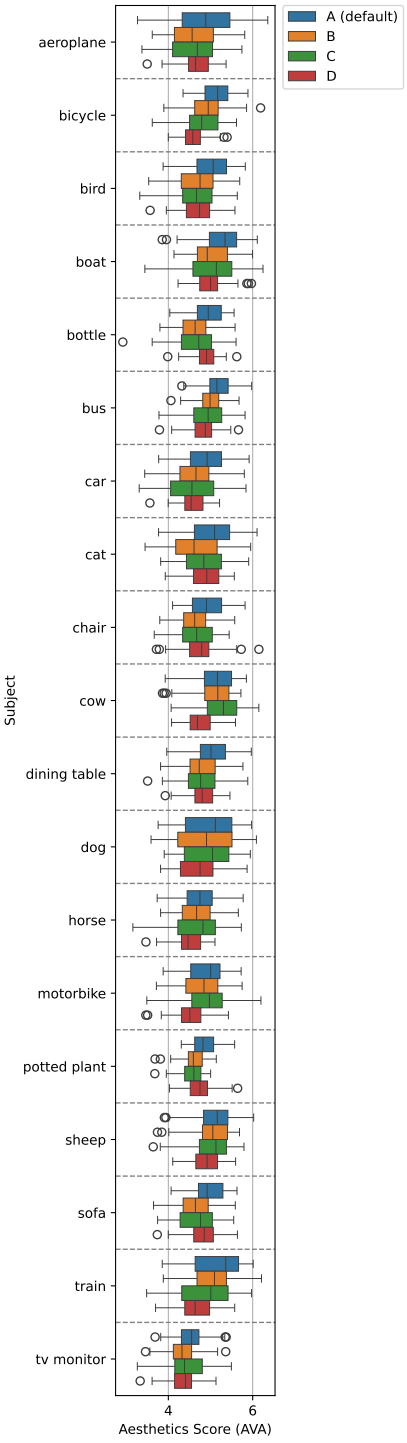


Fig. 36. Controllability of Hidden Overlay Illusion over different subjects.

Stony Brook University
rburgert@cs.stonybrook.edu

Fig. 37. Diversity of Hidden Overlay Illusion over different subjects.



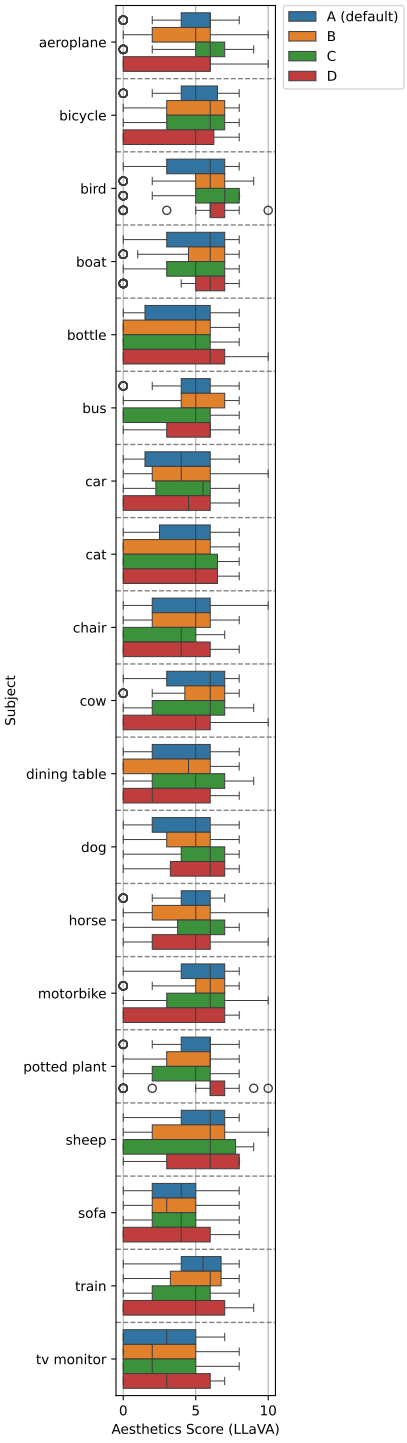


Fig. 38. Aesthetics of Hidden Overlay Illusion over different subjects.

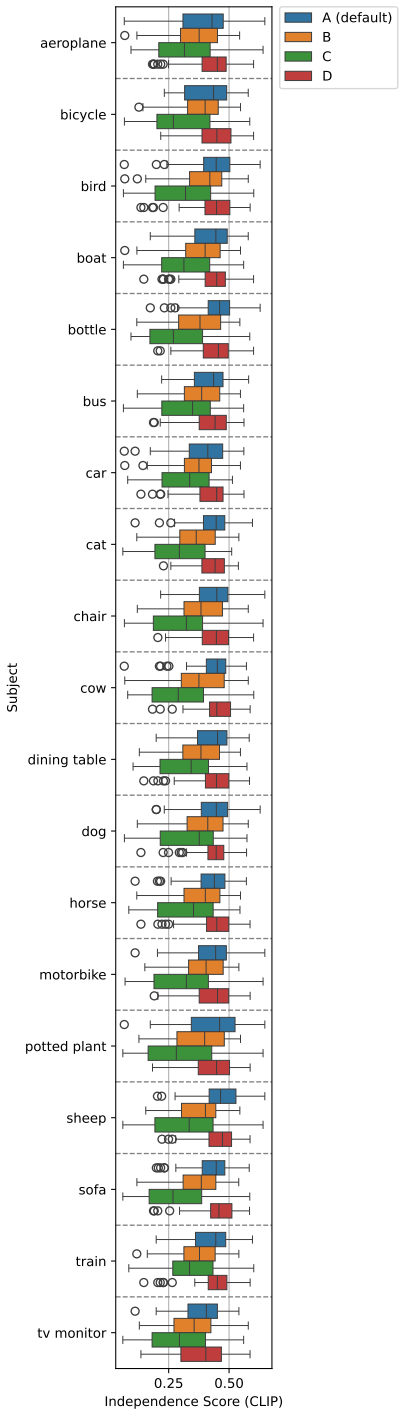


Fig. 39. Independence Score of Hidden Overlay Illusion over different subjects.

Stony Brook University
rburgert@cs.stonybrook.edu

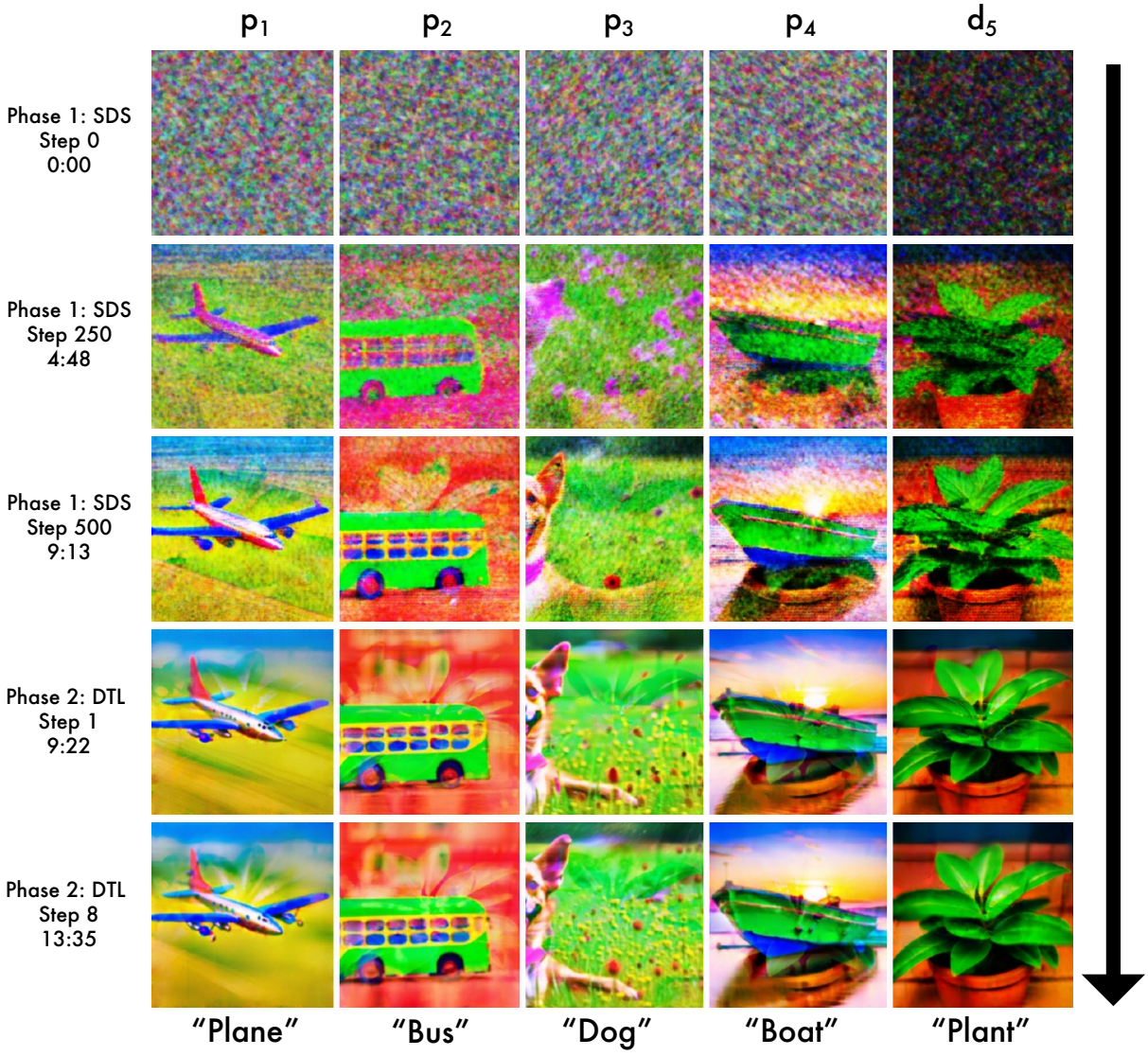
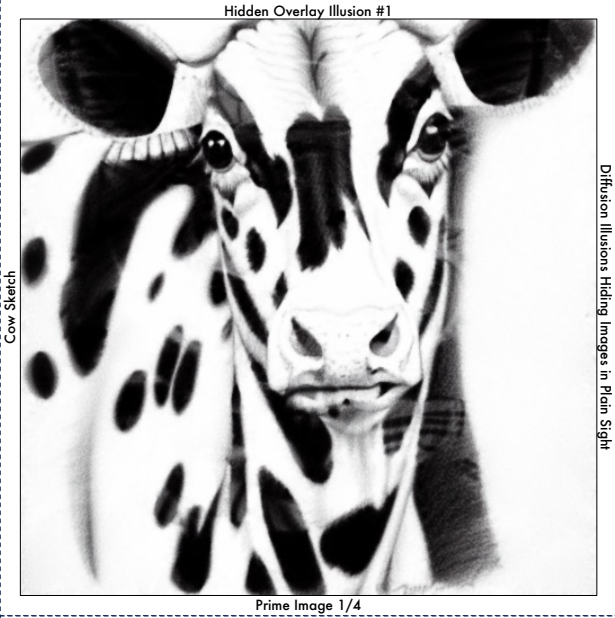


Fig. 40. A sample optimization timelapse for a hidden overlay illusion. Times are denoted in 'minute:second' format. DTL stands for Dream Target Loss, and SDS refers to Score Distillation Sampling.

Make your own Hidden Overlay Illusions!
Print these onto a transparency film using a laser printer and cut them out!
Then, with a bright light behind them, overlay and align all four images.

Hidden Overlay Illusion #1



Cow Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 1/4

Hidden Overlay Illusion #1

Penguin Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 2/4

Hidden Overlay Illusion #1

Dog Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 3/4

Hidden Overlay Illusion #1

Giraffe Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 4/4