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

Abstract

We study the problem of automatically generating images that may cause illusions when viewed in a certain way using a frozen text-to-image diffusion model. Firstly, we propose a formal definition of generating illusion images computationally. Then we introduce Diffusion Illusions, a general pipeline designed for generating a diverse array of illusions without tuning any parameters of the diffusion network. We conduct comprehensive experiments across multiple aspects and verify the effectiveness of our proposed method qualitatively and quantitatively. We also highlight the successful creation of physically fabricated versions of our illusions. Our code and results will be released publicly. Please visit our interactive project website: https://diffusionillusions.com

1. Introduction

An image that 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, that when superimposed form a biohazard symbol. These types of illusions have long required immense time and skill to create, but we have developed a flexible pipeline capable of generating appealing illusions automatically.

Generating such images is not the sole domain of play. 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.

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 [3, 14] and philosophers [41] since the 1800s. Ambiguous images have been used experimentally to understand how category bias during perception varies as people age [22], and families of ambiguous images, such as ambigrams [12], are often constructed as a way of better understanding the domains they belong to. We present some relevant examples in classical illusions.

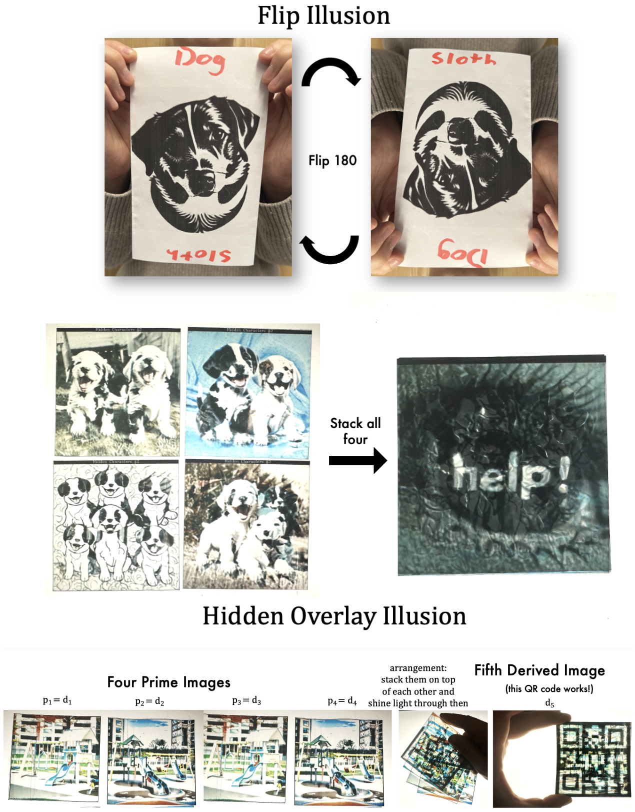


Figure 1. Diffusion Illusions are a new class of automatically generated optical illusions. The top image demonstrates our Flip illusion, and the bottom one showcases four transparent images that when stacked on top of each other and put on a backlight reveal fifth images. *Please note that these are all real photographs.*

Computationally-generated Illusions: A growing stream of research has focused on computationally generating specific types of illusions. One early example is hybrid images [23]. 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.

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 [13], a single 3D wire may be viewed or lit from multiple angles to obtain different clean line drawings; and in view-dependent surfaces [27], a colored 3D-printed heightfield may be viewed from different angle to obtain different color images.

An additional 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 [25], seemingly meaningless dots are generated such that, when viewed through an intricate refractive lens, they will comprise a specified image.

Diffusion-based Image Generation: Diffusion Probabilistic Models [37] resulted in rapid advances for image generation tasks, including text-to-image generation [7, 21, 31–35, 43]. Recent works [4, 29] sample pre-trained diffusion models without re-training to generate outputs in novel domains. Score Distillation introduced in DreamFusion [29] 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 [1, 2, 10, 15, 42], which we use to evaluate our proposed framework.

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 [38] or ambigrams [36]. While these approaches appear to yield appealing results, they are narrowly focused on specific illusions, may require substantial cherry-picking, and have not been formally presented or published. In contrast, we present a formalized, generic approach capable of generating variable types of illusions followed by extensive evaluation (both quantitative and qualitative) of our approach. Contemporary work in [11] presents a formal framework for efficient (fast inference) illusion generation, but operate on a subset of our illusions and do not explore generality for real-world transfer (i.e. fabrication of illusions in real world).

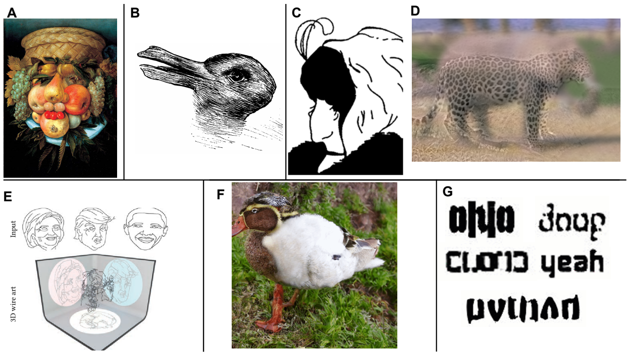


Figure 2. A brief history of illusions. **Classical illusions**: (A) “Fruit Basket” (1500s) by Giuseppe Arcimboldo provides a very early example, depicting a face when viewed in one orientation and a fruit basket when viewed in the other. (B) When viewed directly, “Kaninchen und Ente” (1892) is ambiguous; 45° rotations make it appear as a rabbit or a duck [14, 41]. (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 [3, 22]. **Computationally-generated illusions**: (D) a hybrid image which appears to be a leopard when viewed close-up and an elephant when viewed from a distance [23]. (E) a wire sculpture which depicts three different 2010s American politicians when viewed from different angles [13]. **Diffusion-based illusions**: (F) an image depicts a duck when viewed upright and a rabbit when rotated 90° ccw [38]. (G) a set of computationally-generated ambigrams reading ‘Ohio’, ‘doug’, ‘cloud’, ‘yeah’, and ‘python’ [36].

Contributions: In this work, we make three main contributions: 1) we provide the first formal definition for the problem of generating illusions; 2) we present Diffusion Illusions, a flexible tool for generating multiple types of illusions; and 3) we outline an automated framework for assessing the quality of computer-generated illusions and conduct comprehensive experiments in multiple aspects.

2. Problem Statement

We define an illusion as the situation that occurs when a set of physical images called *prime images* are viewed or *arranged* in multiple ways, with each arrangement yielding a unique perceived image, referred to as a *derived* image, that represents a specific object or scene. Most of the existing illusions we have 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 [23], on the other hand, might be best modeled by high-pass and low-pass spatial frequency filters.

In an effort to find a fully general definition of illusions and leverage the new possibilities afforded by text-to-image

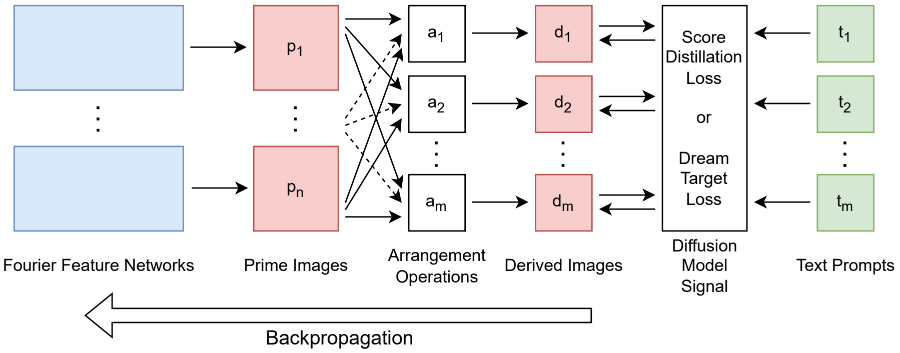


Figure 3. Architecture overview. Trainable components shown in blue, intermediate variables in red, non-trainable functions in white, and inputs in green. A diffusion network provides two different loss signals pulling the derived images towards the text prompts. Only a single loss signal, either Score 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.

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 a multiply operation to model the light-filtering step.

Formally, the illusion process is described as follows. Consider some prime image space $\mathcal{P}$ representing physically realizable visual stimuli, and some 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 $A = \{a_1, a_2, \ldots, a_m\}, a_j : \mathcal{P}^n \to \mathcal{D}$. 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 a wire frame viewed through a stencil; in this case each prime image p_i belongs to its own prime image space $\mathcal{P}_i$.)

This framing is complementary to the existing literature on “ambiguous images”. The illusion process is not intended to cover images which 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 of ambiguous images 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 step is sufficiently concrete that it may be practically automated, as we discuss in Sec. 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 parameterized prime images ($\mathcal{P}$), a set of specific arrangement processes (A, that derive images from primes), and 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 Sec. 3.4) to suitably optimize the prime images, which in turn modifies the derived images. The whole pipeline is demonstrated in Fig. 3.

3.1. Prime Images

As described in Sec. 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) [39] to represent prime images in parametric form. For each prime image, the learnable weights of a single MLP network acts as its representation. The MLP network weights map image-space coordinates to corresponding RGB values similar to [5], forming an implicit image representation. We further discuss the advantages of FFN in Sec. 4.3.

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. 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.

3.3. Specific Illusions

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 (derived image 1) and as another object when viewed upside-down (derived image 2).
- **Rotation Overlay Illusion** is a minimal 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 with 0, 90, 180, and 270 degree angles and superimposed on the base image; each rotation yields a derived image interpreted as a different object.
- **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, maybe 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. The Hidden Overlay Illusion demonstrated in real life. Please note that these are all real photographs.

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 Tab. 1.

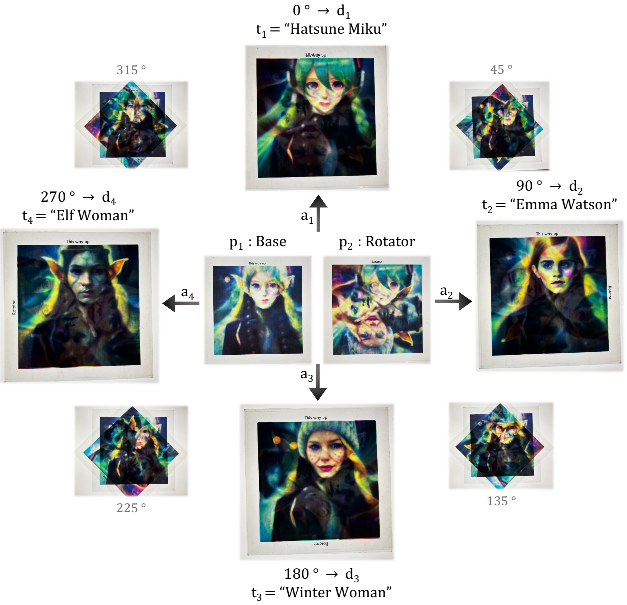


Figure 4. This figure shows the rotation overlay illusion arrangement process. *Please note that these are all real photographs.* The “rotator” image is placed on a “base” image over a backlight, both printed out onto transparent sheets. Then, as the rotator spins, we derive four different images.

Illusion	n	m	$\mathbf{a}$
Flip	1	2	$a_1(\mathbf{p}) = p_1$ $a_2(\mathbf{p}) = \text{rot}(p_2, 180)$
Rotation Overlay	2	4	$a_j(\mathbf{p}) = p_1 * \text{rot}(p_2, 90j)$
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.

3.4. Learning Process

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 methods to provide suitable alignment signals to the derived images, that in turn would update the prime images. First, we use Score Distillation Loss [29], a high-fidelity but expensive ithm that applies a conditional denoising model to the input at every image update step. Second, we introduce the complementary Dream Target Loss, a faster technique that pulls the derived images towards periodically updated *target images*.

For each derived image (d_i), we introduce a target (t_i) that describes in natural language the expected visual appearance of its final form. This natural language description, t_i is used

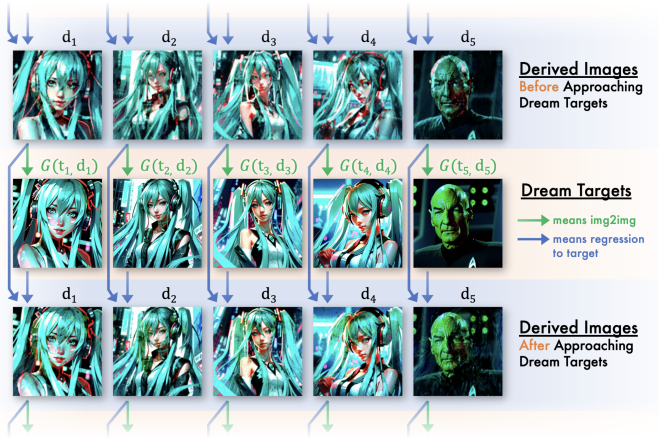


Figure 5. We depict the dream-target loss above. It is an iterative process, refining derived images using SDEdit to create target images, which the derived images are then regressed to with gradient descent. Note how the derived images look more like the targets after approaching them than before, such as the man's green face.

Score Distillation Loss: is a widely-used technique to align images with external conditioning such as textual prompts. In essence, Score Distillation Loss randomly selects a timestep τ of the denoising process, adds noise η proportionate to the timestep τ to the derived image x, and applies the denoising process to $x + \eta$ to obtain an estimated x_0 and estimated noise $\hat{\eta}$. The difference between the estimated noise $\hat{\eta}$ and actual noise η provides a signal for the discrepancy between the derived image and the target category for the derived image. This difference is normalized by τ and then provided as a gradient to the derived image and backpropagated through the view process to the prime image. Importantly, this process does not require any backpropagation through the diffusion model. In summary, we may treat score distillation as an operation $\mathcal{L}_{\text{SD}}(\text{Image}, \text{Text})$ that can provide gradients to optimize the image such that iterative updates to the image converge its appearance towards the paired text caption.

$$\mathcal{L}_i = \mathcal{L}_{\text{SD}}(t_i, d_i) \tag{2}$$

Minimizing this loss in Eq. (2) and propagating gradients provides a minimal framework with strong results. We next explore how compute efficiency of this setup can be improved.

3.4.1 Dream Target Loss

Our novel Dream Target Loss is an optimized version of the Score Distillation Loss for circumstances where it is not trivial for input image(s) to follow their gradients. In the situation where the gradients from multiple derived images are competing to influence the prime images, it is inefficient to apply a full diffusion model at every iteration while this competition is being resolved.

Instead, Dream Target Loss periodically applies a conditional image-to-image process ($\mathcal{G}$) to obtain a target image for each derived image, and then gradually pulls each derived image towards its target image using a combination of the structural image similarity loss ($\mathcal{L}_{SSIM}$) and a pixelwise mean squared error loss ($\mathcal{L}_2$). In practice, we implement $\mathcal{G}$ using SDEdit [19] (please see appendix for details). We define new image targets as $\mathcal{G}(t_i)$ for each derived image using the same textual targets. Therein, we obtain a joint loss,

$$\mathcal{L}_i = \mathcal{L}_{\text{SSIM}}(z_i, d_i) + \mathcal{L}_2(z_i, d_i), \text{where } z_i = \mathcal{G}(t_i, d_i) \tag{3}$$

Minimizing this loss allows us to similarly learn optimal prime images p_i resulting in derived images aligned to each of our target concepts. An additional feature of the Dream Target Loss relative to the SD variant is that it tends to introduce less noise. We illustrate this in Fig. 5.

Our final loss is an average across all per derived image loss terms, $\mathcal{L}_i$ to obtain,

$$\mathcal{L} = \sum w_i \mathcal{L}_i \tag{4}$$

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 $w_5 = 3$.

Note that in both variants of losses, we propagate gradients all the way to the prime images, updating their parametric representing (i.e. the weights of the MLP Fourier Feature Networks).

3.4.2 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 printing their prime images on overhead display sheets on a color laser printer, before being laminated to protect them from scratches. It's a really simple process! With a strong enough backlight, the hidden overlays and rotation overlay illusions can be performed on regular pieces of paper as well.

4. Experiments

4.1. Qualitative Evaluation

We showcase many outputs of our Diffusion Illusions. Flip Illusions are shown in Sec. 4.1, Hidden Overlay Illusions are shown in Fig. 7, and Rotation Overlay Illusions are shown in Fig. 8.

Please see the appendix for more.

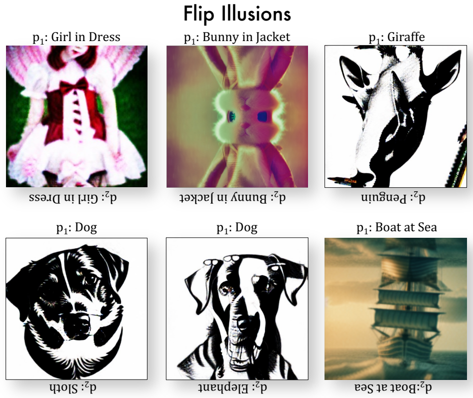


Figure 6. **Flip Illusion Examples:** *Please zoom in!* Please view these images upside-down as well as right-side-up to see two different subjects. Note: In this illusion, $d_1 = p_1$

4.2. Quantitative Evaluation

In this section, 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 proposed framework.

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 $i \in \{1, \dots, 5\}$ from T^o. Then we substitute the *subject* token in t^s with t_i^o to construct the textual prompt t_i. Finally, $t_1, \dots, t_4$ are used to generate the prime images respectively, and t_5 is used to generate the overlay image.

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 in PASCAL VOC [9] ($|T^o| = 20$), and $N = 64$. Please refer to the Appendix for the complete list of subjects and styles.

Evaluation Metrics Inspired by recent works on diffusion model evaluation [16, 42], 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 [30] model.
- *Diversity* the variety of generated images conditioned on the same prompt. For images generated by the same textual prompt, we calculate two *Venti scores* [10] independently based on two visual embeddings: the [CLS] embeddings of DINOv2 [24] 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 [20] 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 indepen-

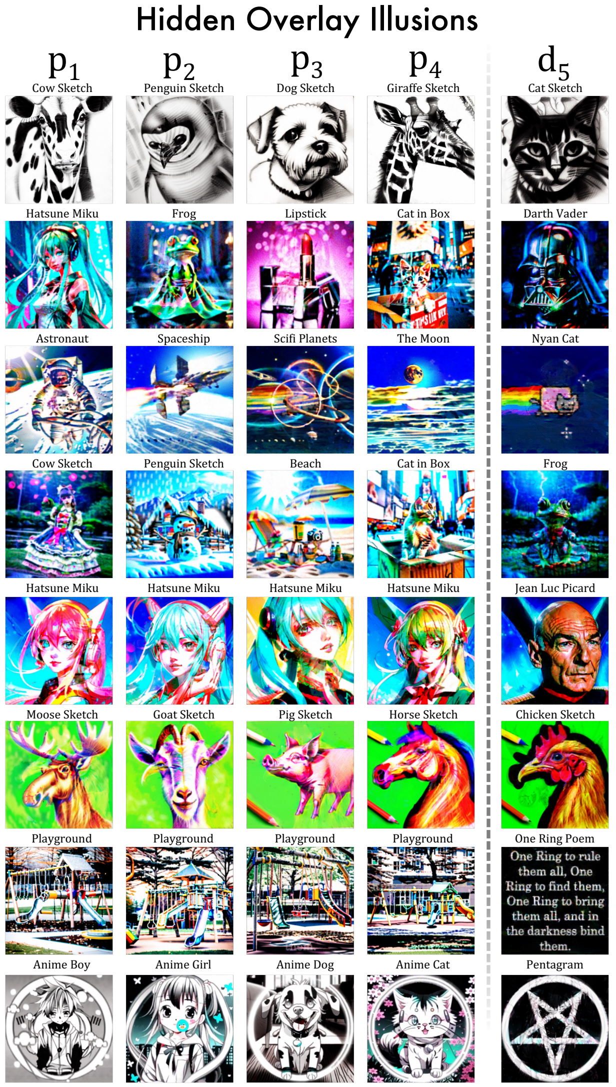


Figure 7. **Hidden Overlay Examples:** *Please zoom in!* On the left are the four prime images p_1, p_2, p_3, p_4 and on the right is the derived image $d_5 = p_1 \cdot p_2 \cdot p_3 \cdot p_4$, which simulates overlaying them over a backlight. Note: In this illusion, $d_{1\dots4} = p_{1\dots4}$

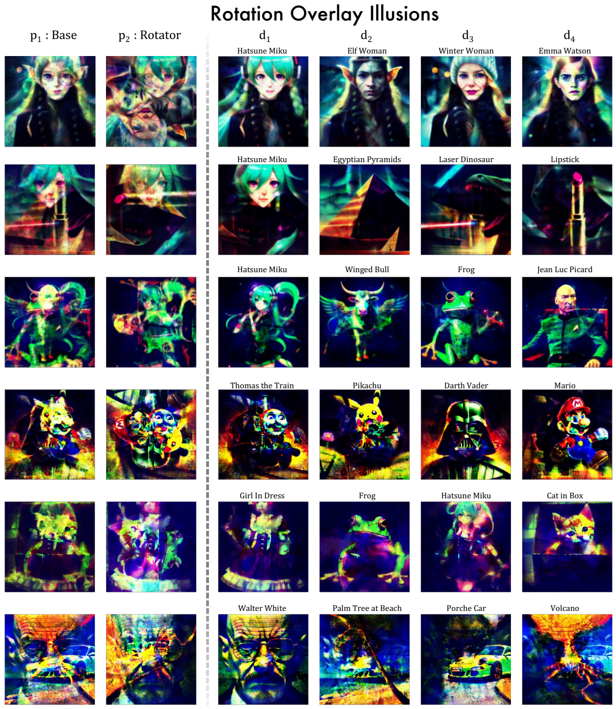


Figure 8. **Rotation Overlay Examples:** *Please zoom in!* On the left are the two prime images p_0, p_1 and on the right are the four derived images d_5 that obtained by taking the product of the primes, simulation of them overlaid on a backlight.

dence 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 [30] 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.

$$K_0 = \text{Softmax}(K/\tau, 0) \tag{5}$$

$$K_1 = \text{Softmax}(K/\tau, 1) \tag{6}$$

$$S_{\text{IS}} := \min(\text{diag}(K_0) \cup \text{diag}(K_1)) \tag{7}$$

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.

Methods The baseline method of our experiments is a vanilla SDXL generating target images with corresponding textual prompts independently for one step. We benchmark four variants of our methods named A, B, C, and D. Method C is our default method, and involves 500 steps of score distillation loss followed by 8 steps of dream target loss, and applies relative weights [1,1,1,1,3] respectively - which prioritizes the quality derived hidden image over its constituent primes. In addition, Method A uses Stable Diffusion 1.5 instead of SDXL, which is used by all other methods. Method C uses equal weights for all derived images, using weights [1,1,1,1,1] respectively. Lastly, method D uses 4000 steps of score distillation loss followed by 1 step of dream target loss for smoothness, to evaluate the ability of score distillation loss alone in this task. For fairness, all methods were constrained to run in a 15-minute time window.

Results For all metrics, we report the score distributions achieved by our default method and the baseline in Fig. 9.

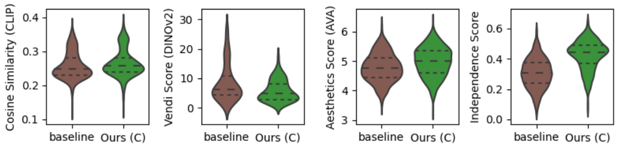


Figure 9. Comparison of multiple score distributions

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. 10. Each row of Fig. 10 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. 10). Method C shows significant advantages in Aesthetics (Aesthetics Score) and Methods C and D achieve relatively higher Independence Score.

A detailed look at different art styles is presented on the right-hand side of each row of Fig. 10, 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, the prompts used are far more important than the chosen implementation. There is no clear one-

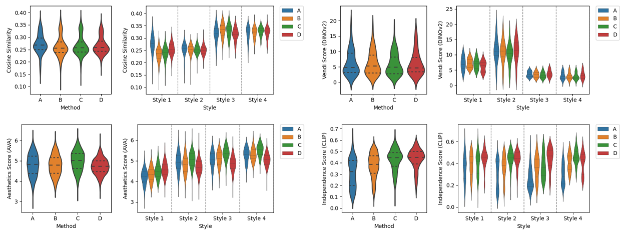


Figure 10. Score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

size-fits-all method indicated by our quantitative evaluations, however, we observe that depending on the art styles and subjects used, a different method will be optimal. One should carefully pick up a method when generating illusions in a specific art style. A further study on subjects is available in the Appendix.

4.3. Discussions

In this section, we discuss several observations that may inspire future investigation.

Q1: Can Diffusion Illusion yield better images when running for a longer time?
Yes. Fig. 11 presents the trend of Controllability (Cosine Similarity) and Aesthetics (Aesthetics Score) as the images used in Sec. 4.2 are getting optimized. The term 'relative time' is employed to denote the progression of wall-clock time during the optimization process. A relative time value of 0 means the beginning of optimization, whereas a value of 1 marks its conclusion. Fig. 11 reveals a notable trend: there is a consistent increase in metrics as the optimization process advances.

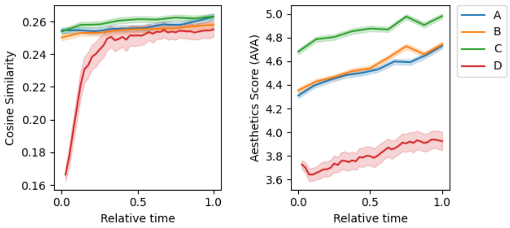


Figure 11. CLIP Cosine Similarity (left) and Aesthetics Score (right) increase when optimizing for a longer time.

Q2: Is Independence Score a qualitatively valid metric?
Generally yes. Fig. 12 shows four illusions randomly selected with diverse independence scores. For each row, the subject of each image is listed above, and the method, style, and independence scores are listed on the left-hand side. The four images grouped in the middle are prime images and they derive the overlay image on the right-hand side. For the first two examples where the independence score is relatively high, each image aligns with its corresponding textual prompt. However, for the third example, the overlay image is not closely related to the subject 'sofa', resulting in a lower independence score. Furthermore, in the last example of Fig. 12, the overlay image visually biases more towards 'cow' instead of 'bottle', leading to the lowest independence score.

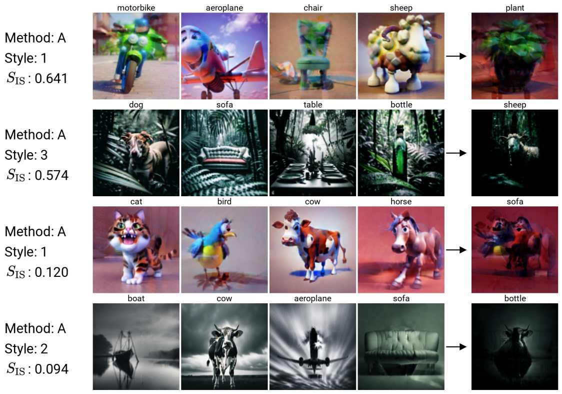


Figure 12. Examples with diverse independence scores

Q3: What are the reasons to use Fourier Features Network?
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 [4, 5], we elect to use Fourier Features Network [39] based parametric image representations.

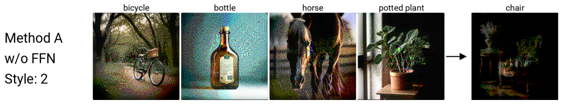


Figure 13. A Hidden Overlay image with prime images optimized directly in pixel space. While high-frequency encoding of the hidden image results in less perceivable interference in each individual image, it results in a brittle illusion that is disrupted without pixel-perfect printing and alignment.

5. 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. Furthermore, as noted, the accuracy of the Rotation Overlay or Hidden Overlay illusions (Tab. 1) might vary due to physical imperfections in the printing process (refer to Appendix for details) - leaving room for further exploration. Other areas to explore include more types of illusion generation and creative ways to take advantage of diffusion models.

Limitations: The main limitation of our framework is the relatively high inference time required for generating illusions. While our framework improves over plain score-distillation in terms of inference time, we are still slow. Improving speed of illusion generation frameworks such as ours presents an interesting future direction. We note that contemporary work has already explored ways to minimize this inference time. Other limitations include biases contained in our models (discussed in detail under ethics statement).

Reproducibility Statement Our work builds off open-source models whose pre-trained weights are publicly available. Our framework simpy 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.

Contributions: RB led the project, conceived the prime image / derived image illusion relationship, invented the classes of hidden and rotation overlay illusions, and designed & implemented the Diffusion Illusions pipeline. XL designed and performed all quantitative evaluation experiments. AL formalized and wrote the Illusion problem statement and contributed to paper writing. KR discussed multiple aspects of the project, supported designing a prior framework (Peekaboo) important for building our setup, and contributed to paper writing. MR supervised the project, advised on research direction, and discussed all aspects of the project.

References

[1] Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129:1712 – 1731, 2020. 2

[2] Eyal Betzalel, Coby Penso, Aviv Navon, and Ethan Fetaya. A study on the evaluation of generative models. *ArXiv*, abs/2206.10935, 2022. 2

[3] E. G. Boring. A new ambiguous figure. *The American Journal of Psychology*, 42:444–445, 1930. Place: US Publisher: Univ of Illinois Press. 1, 2

[4] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S. Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. *ArXiv*, abs/2211.13224, 2022. 2, 8, 1

[5] Ryan Burgert, Jinghuan Shang, Xiang Li, and Michael Ryoo. Neural neural textures make sim2real consistent. In *Proceedings of the 6th Conference on Robot Learning*, 2022. 3, 8, 1

[6] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 3

[7] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 6, 3

[10] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022. 2, 6

[11] Daniel Geng, Inbum Park, and Andrew Owens. Visual anagrams: Generating multi-view optical illusions with diffusion models. *arXiv:2311.17919*, 2023. 2

[12] Douglas R Hofstadter. 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*, pages 274–278, 1985. 1

[13] Kai-Wen Hsiao, Jia-Bin Huang, and Hung-Kuo Chu. Multi-view wire art. *ACM Transactions on Graphics*, 37(6):1–11, 2018. 2

[14] Joseph Jastrow. The mind's eye. *Popular Science Monthly*, pages 299–312, 1899. 1, 2

[15] 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. Holistic Evaluation of Text-To-Image Models, 2023. arXiv:2311.04287 [cs]. 2

[16] Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Benita Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *arXiv preprint arXiv:2311.04287*, 2023. 6

[17] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 3

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 3

[19] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022. 5

[20] Naila Murray, Luca Marchesotti, and Florent Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2408–2415. IEEE, 2012. 6

[21] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 2

[22] Michael E. R. Nicholls, Owen Churches, and Tobias Loetscher. Perception of an ambiguous figure is affected by own-age social biases. *Scientific Reports*, 8:12661, 2018. 1, 2

[23] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. *ACM Transactions on Graphics (TOG)*, 25(3): 527–532, 2006. 2

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6

[25] Marios Papas, Thomas Houit, Derek Nowrouzezahrai, Markus Gross, and Wojciech Jarosz. The magic lens: refractive steganography. *ACM Transactions on Graphics*, 31 (6):1–10, 2012. 2

[26] 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. Pytorch: An imperative style, high-performance deep learning library, 2019. 1

[27] Maxine Perroni-Scharf and Szymon Rusinkiewicz. Constructing Printable Surfaces with View-Dependent Appearance. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–10, New York, NY, USA, 2023. Association for Computing Machinery. 2

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv*, 2023. 1

[29] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, abs/2209.14988, 2022. 2, 4

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6, 7

[31] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ICML*, 2021. 2

[32] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[33] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2021.

[34] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021.

[35] 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. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv:2205.11487*, 2022. 2

[36] Noufal Samsudin. Generating ambigrams using deep learning: A typography approach, 2023. unpublished work. 2

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015. 2

[38] Matthew Tancik. Illusion diffusion: optical illusions using stable diffusion, 2023. unpublished work. 2

[39] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020. 3, 8

[40] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022. 1

[41] Ludwig Wittgenstein. *Philosophical investigations.* Macmillan, Oxford, England, 1953. (Part 2, Section 11). 1, 2

[42] Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard BW Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lycoris fine-tuning to model evaluation. *arXiv preprint arXiv:2309.14859*, 2023. 2, 6

[43] 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. Scaling autoregressive models for content-rich text-to-image generation. *arXiv:2206.10789*, 2022. 2

Diffusion Illusions: Hiding Images in Plain Sight

Supplementary Material

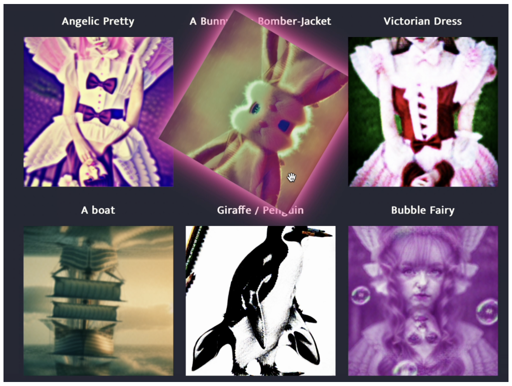


Figure 14. Please visit https://diffusionillusion.github.io/

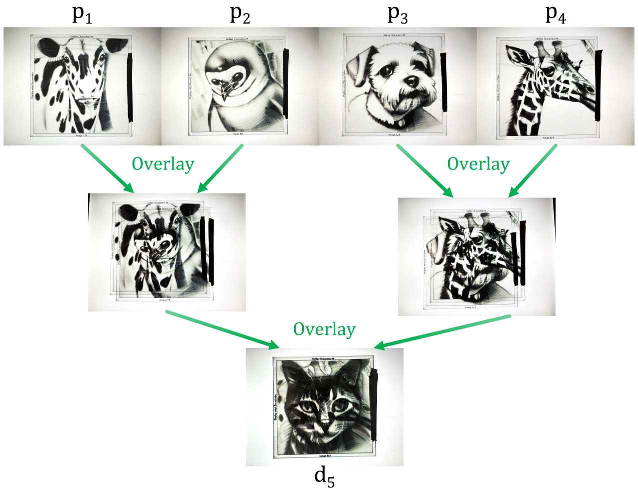


Figure 15. **Hidden Overlay Illusion:** Photographs of different prime images as we overlay them in real life.

A. More Examples

We demonstrate again how to use the Hidden Overlay Illusion in Fig. 15.

We kindly ask the reader to check our website at https://diffusionillusion.github.io/ for more examples.

B. Implementation Details

In this section, we outline our algorithm using python-style pseudo-code.

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. 7). 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 [28]. Our SDEdit implementation of SDXL comes from [40], using PyTorch [26]. Our implementation of fourier feature networks is directly adapted from the TRITON [5], using the default parameters for their Neural Neural Textures. Our implementation of Score Distillation Loss comes from Peekaboo [4].

B.4. Pseudocode

In this subsection we show python-like pseudocode that outlines the exact process of creating the algorithm.

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


###### 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(10000):
    loss = 0
    for a,t,w in zip(A,T,W):
        #Derived image d is 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, .89, .88 ... .03, .02, .01]

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!
send_to_laser_printer(P)

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

C. Extended Quantitative Evaluation

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:

> 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 PASCAL VOC dataset [9]: *aeroplane*, *bicycle*, *bird*, *boat*, *bottle*, *bus*, *car*, *cat*, *chair*, *cow*, *dining table*, *dog*, *horse*, *motorbike*, *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 [17, 18] to measure the similarity between the image and the textual prompt. The instruction sent to the VLM is

  > *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 [6] suggests that the feature from the original DINOv2 might suffer from abnormal patches corresponding to the plain areas of the image. Therefore, we report a new *Vendi Score* using the feature from DINOv2+reg [6].
- *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-Large [8] with a patch size of 14. The version of LLaVA-1.5 we utilized is Vicuna-13B.

Hidden Overlay Illusion Results Fig. 16 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.

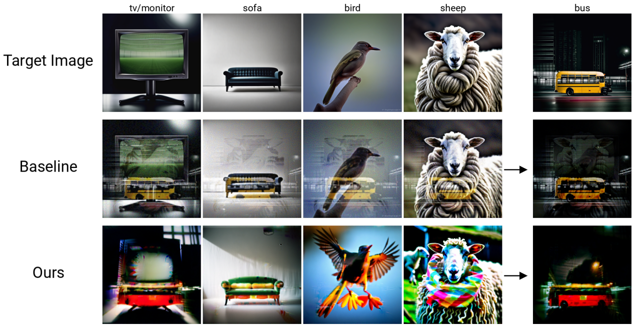


Figure 16. Examples of our method and the baseline, starting from the same target image. Note how in the baseline, you can see the sheep in the bus image and the bus in the sheep image - which is why its independence score is lower.

Fig. 17, Fig. 18, Fig. 19 and Fig. 20 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. 17). Meanwhile, LLaVA has relatively less bias on art styles and different subjects (Fig. 18 and Fig. 20)

A detailed study over different subject is presented as Fig. 21, Fig. 22, Fig. 23, and Fig. 24

Rotation Overlay Illusion Results We further benchmark the performance on 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. The result is presented in Fig. 25. Our method is significantly better than the baseline in terms of controllability (CLIP cosine similarity) and Aesthetics Score from LLaVA and our Aesthetics Score (AVA) is comparable to the baseline.

D. Limitation

As briefly discussed in the conclusion section of the main paper, the effectiveness of visual illusion in the real world may vary a lot due to the errors introduced in the printing process. Fig. 26 and Fig. 27 present the effect of the color shifts when printing the images.

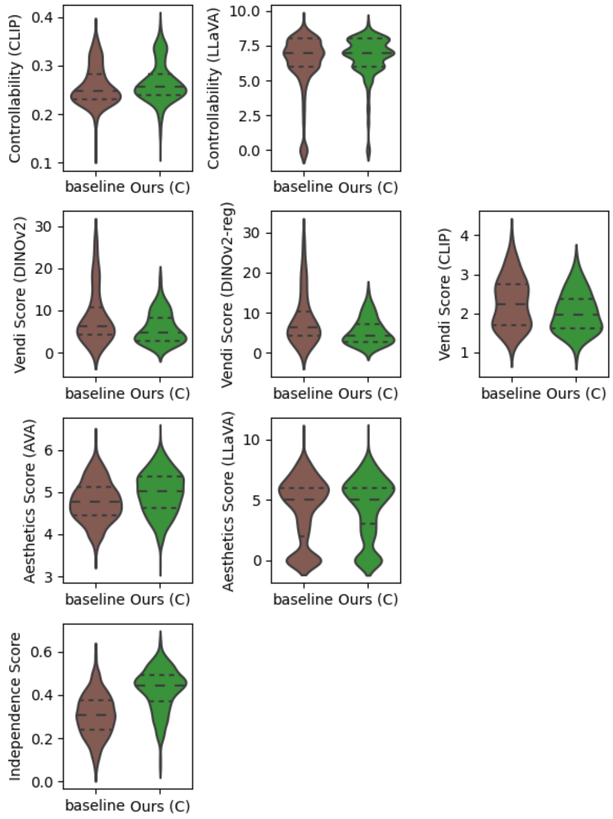


Figure 17. Full evaluation on Hidden Overlay Illusion, each row is a group of thematically-aligned figures.

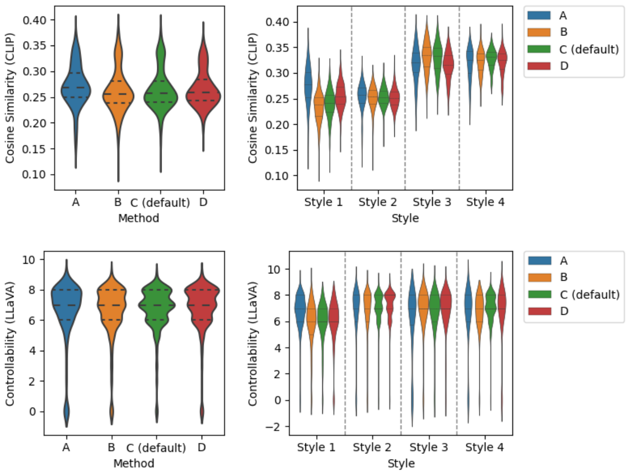


Figure 18. Controllability score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

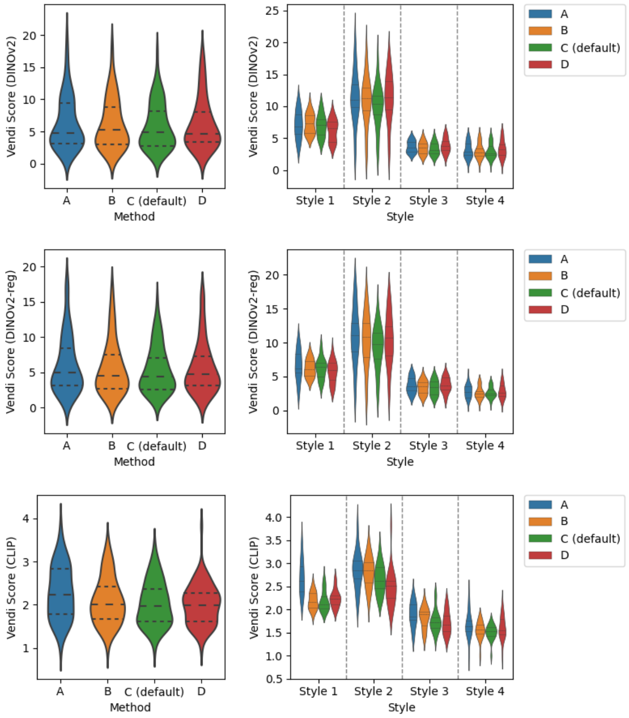


Figure 19. Diversity score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

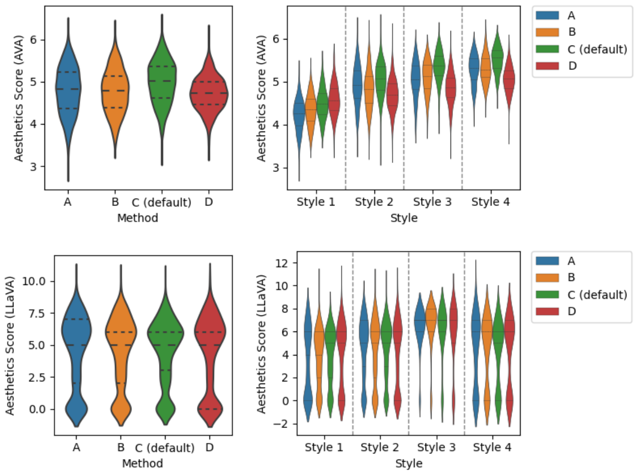


Figure 20. Aesthetics score distributions over methods (left) and styles (right). A, B, C, D stands for four variants of our method

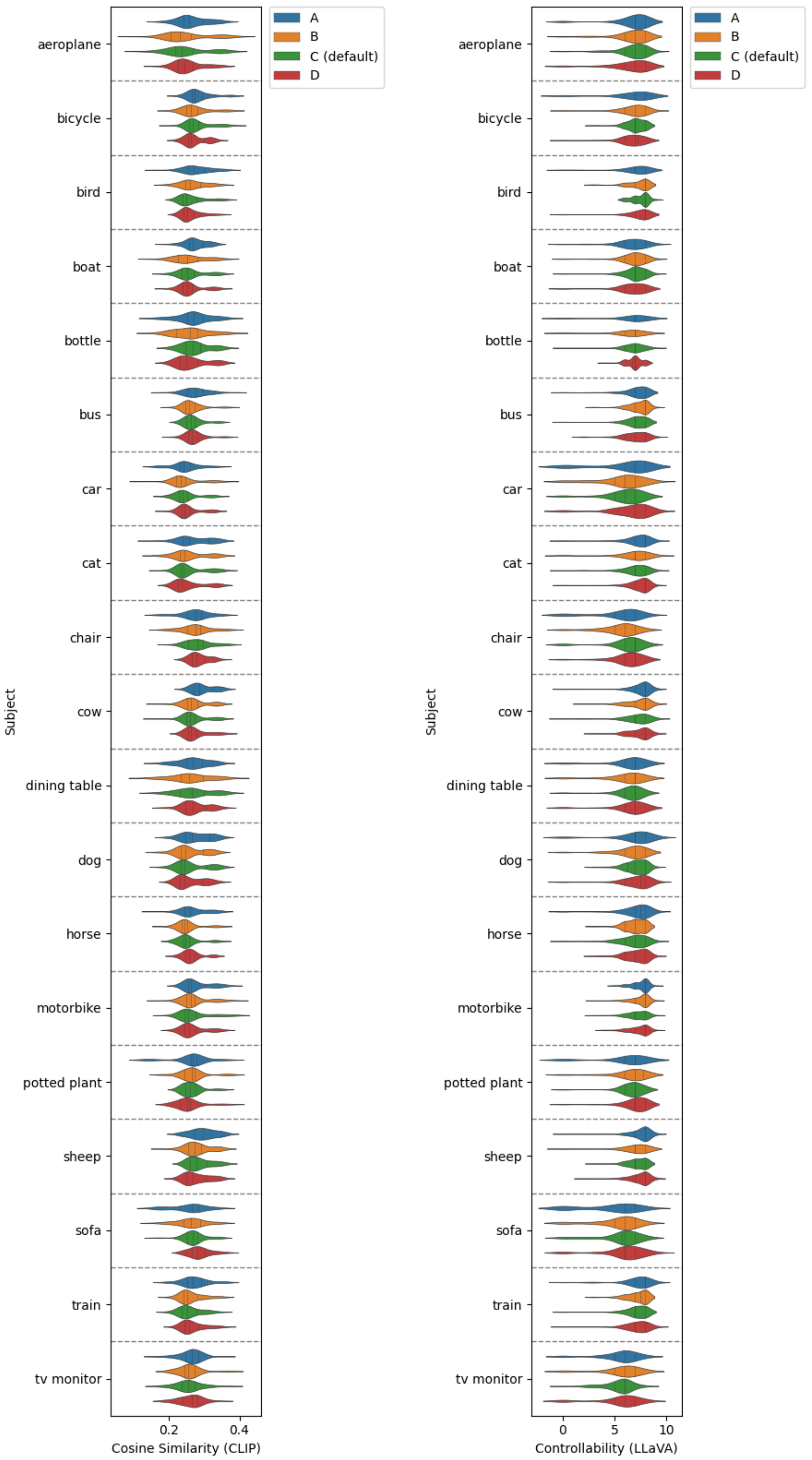


Figure 21. Controllability of Hidden Overlay Illusion over different subjects.

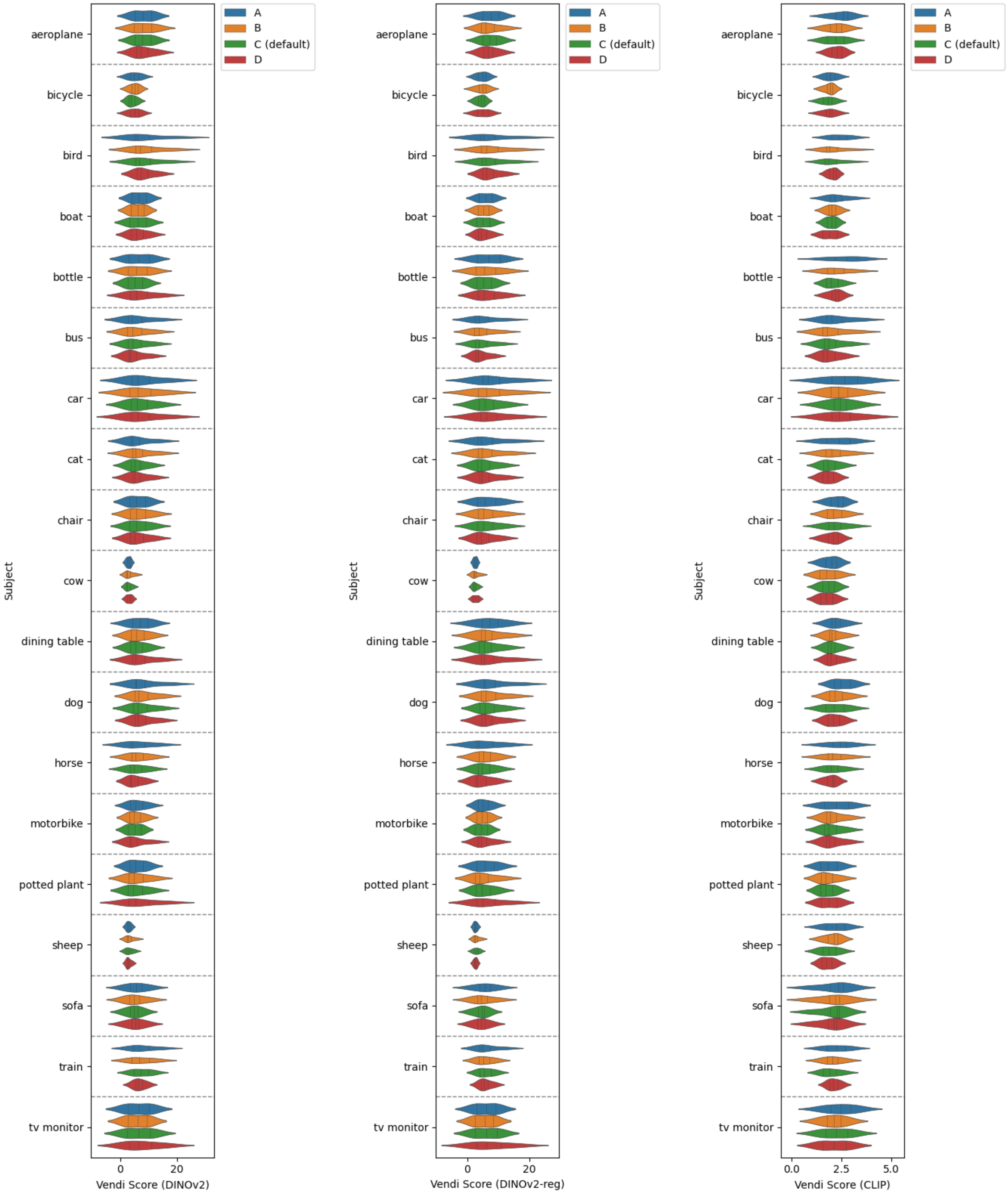


Figure 22. Diversity of Hidden Overlay Illusion over different subjects.

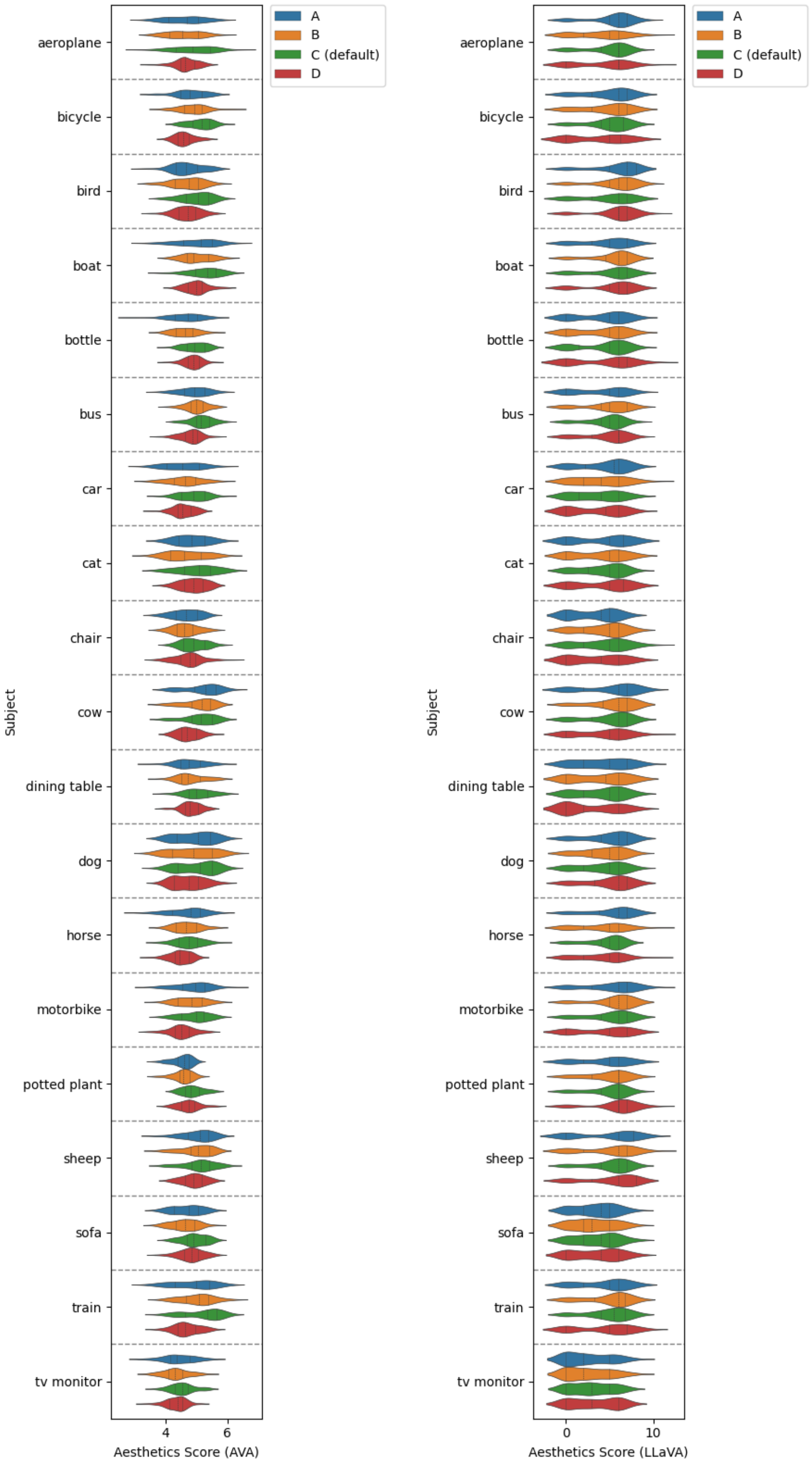


Figure 23. Aesthetics of Hidden Overlay Illusion over different subjects.

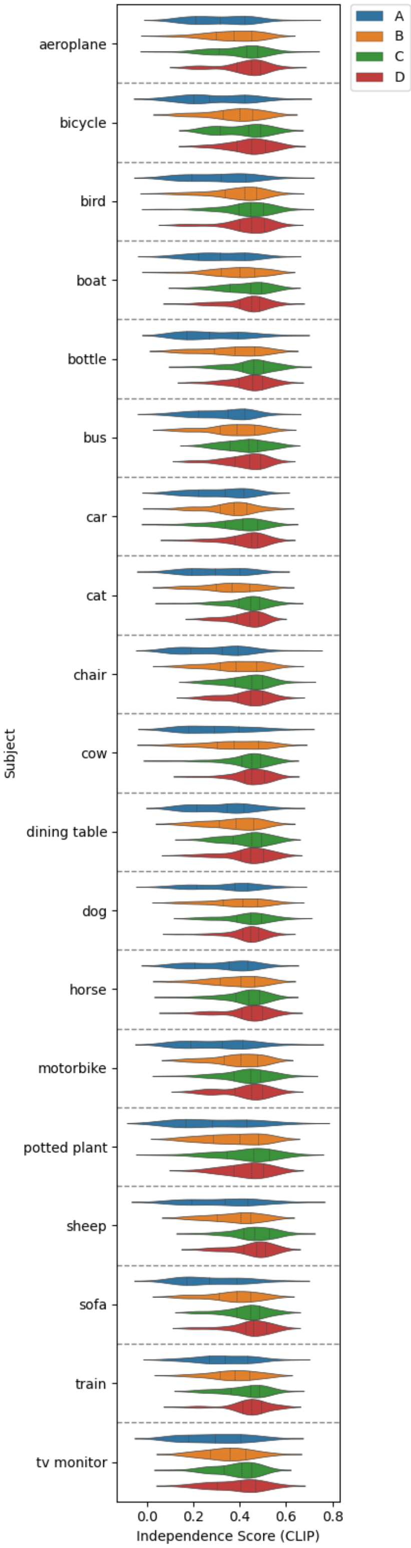


Figure 24. Independence Score of Hidden Overlay Illusion over different subjects.

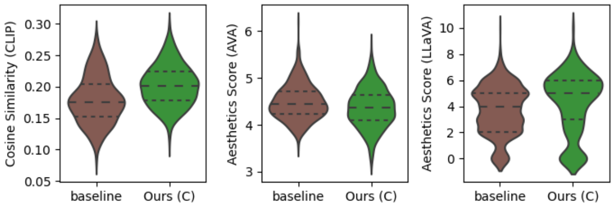


Figure 25. Evaluation on Rotation Overlay Illusion

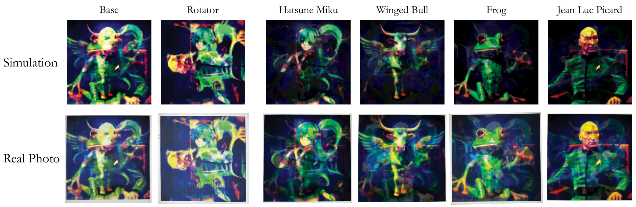


Figure 26. The colors shift after printing out Rotation Overlay Illusion images. First row: digital copy of the images and the overlay simulation. Second row: real-world photos of the printed images.

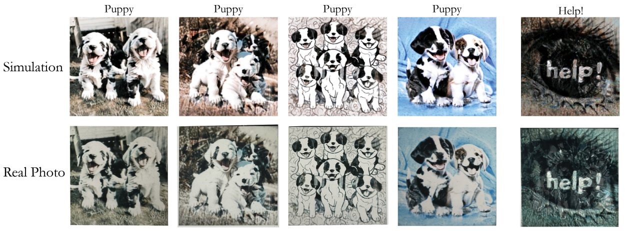
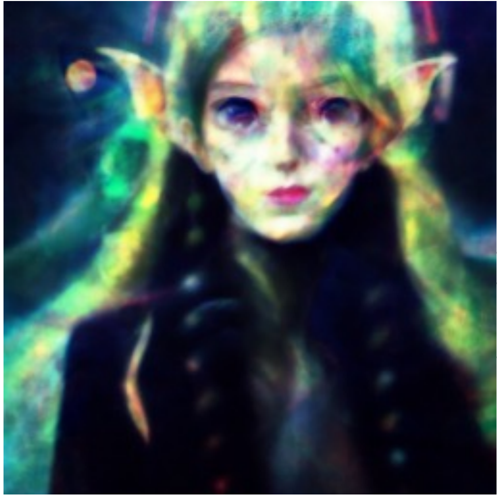


Figure 27. The colors shift after printing out Hidden Overlay Illusion images. First row: digital copy of the images and the overlay simulation. Second row: real-world photos of the printed images.

Make your own Rotation Overlay Illusions!

Print these onto a transparency film using a laser printer and cut them out!
Then, with a bright light behind them, hold the base image in place while you rotate its respective rotator image in 90° intervals

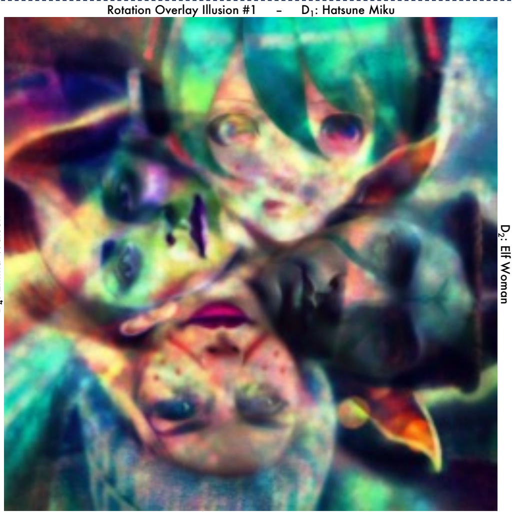
Rotation Overlay Illusion #1



Diffusion Illusions Hiding Images in Plain Sight

Base: Prime Image 1/2

Rotation Overlay Illusion #1 – D_1: Hatsune Miku

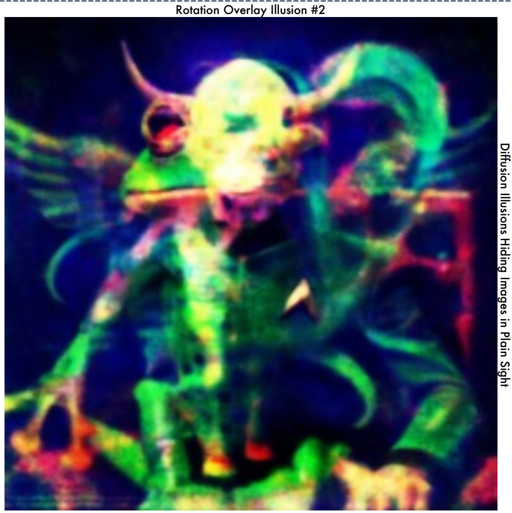


D_4: Emma Watson

D_2: Elf Woman

Rotator: Prime Image 2/2 – D_3: Winter Woman

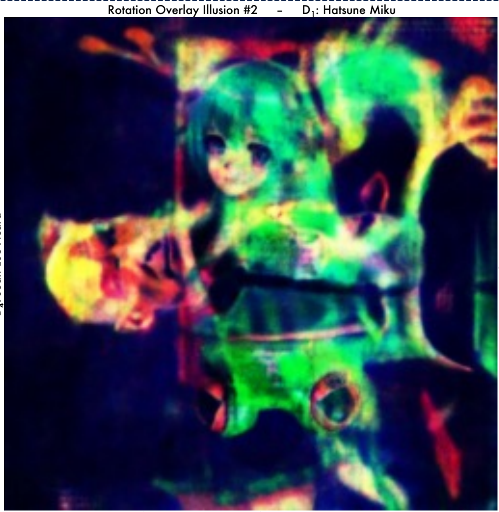
Rotation Overlay Illusion #2



Diffusion Illusions Hiding Images in Plain Sight

Base: Prime Image 1/2

Rotation Overlay Illusion #2 – D_1: Hatsune Miku

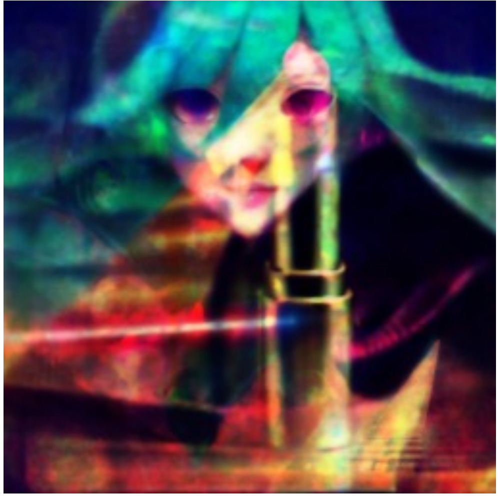


D_4: Jean Luc Picard

D_2: Winged Bull

Rotator: Prime Image 2/2 – D_3: Frog

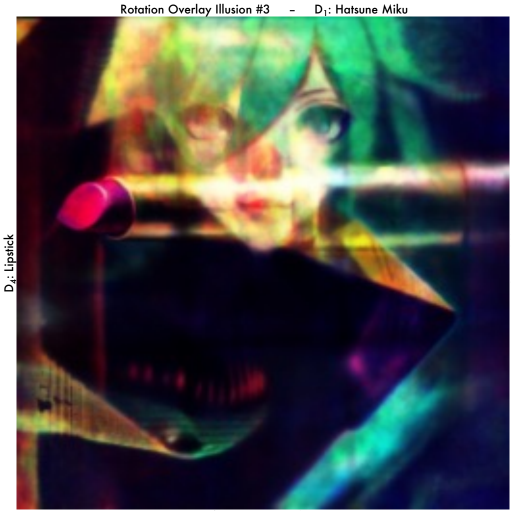
Rotation Overlay Illusion #3



Diffusion Illusions Hiding Images in Plain Sight

Base: Prime Image 1/2

Rotation Overlay Illusion #3 – D_1: Hatsune Miku



D_4: Lipstick

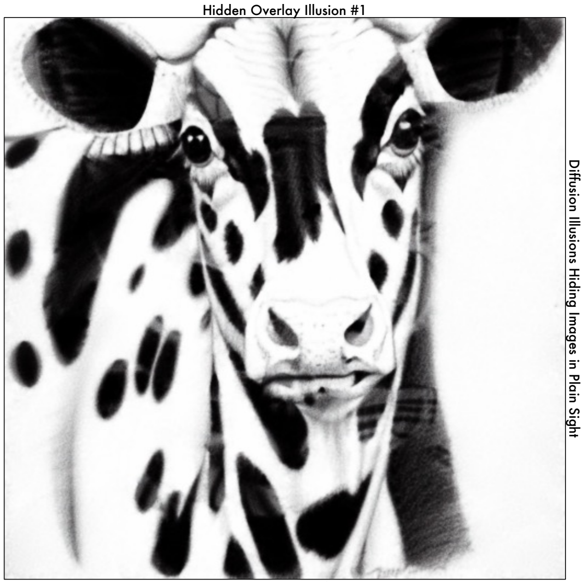
D_2: Egyptian Pyramids

Rotator: Prime Image 2/2 – D_3: Laser Dinosaur

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

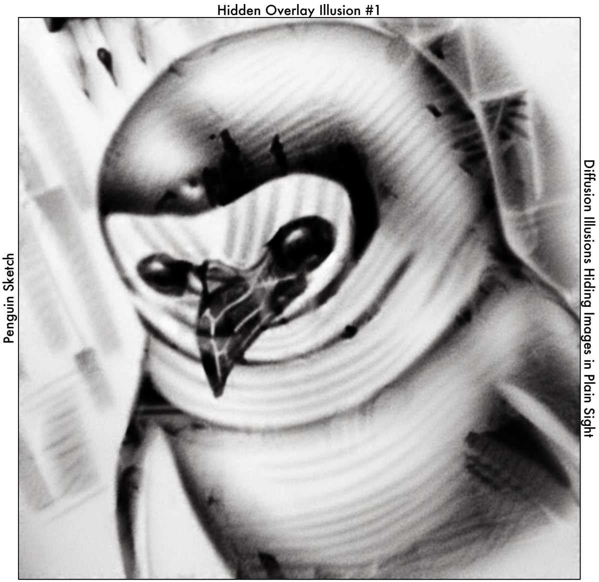


Cow Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 1/4

Hidden Overlay Illusion #1

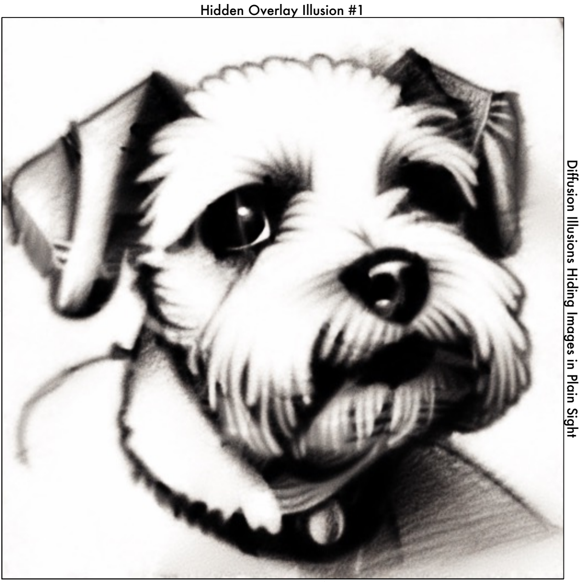


Penguin Sketch

Diffusion Illusions Hiding Images in Plain Sight

Prime Image 2/4

Hidden Overlay Illusion #1

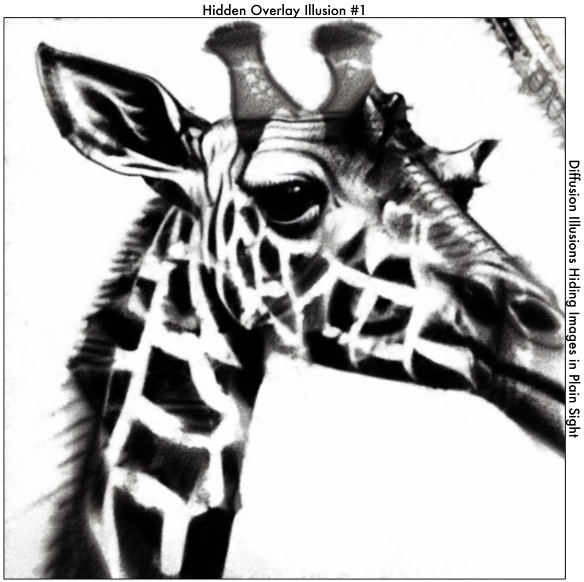


Dog Sketch

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