Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models

Daniel Geng Inbum Park Andrew Owens
University of Michigan
https://dangeng.github.io/visual_anagrams/

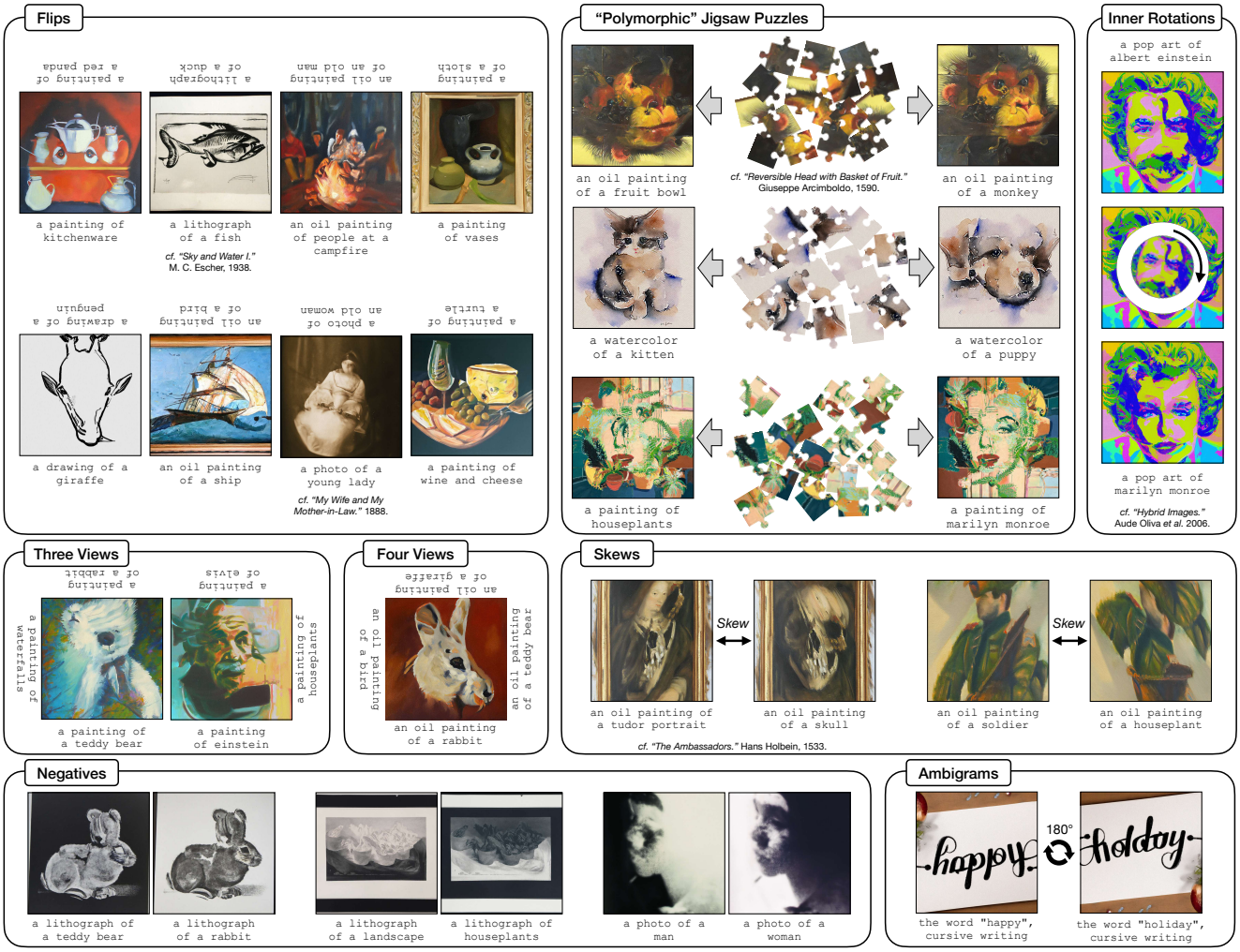

Figure 1. **Generating Multi-View Illusions.** We propose a method for generating optical illusions from an off-the-shelf text-to-image diffusion model. We create images that match different prompts after undergoing a transformation. Our approach supports a variety of transformations, including flips, rotations, skews, color inversions, and jigsaw rearrangements. All images are hand selected. For random samples, please see Fig. 8 and Appendix D. **For easier viewing, please see our webpage for animated versions of these illusions.**

Abstract

We address the problem of synthesizing multi-view optical illusions: images that change appearance upon a transformation, such as a flip or rotation. We propose a simple, zero-shot method for obtaining these illusions from off-the-shelf text-to-image diffusion models. During the reverse diffusion process, we estimate the noise from different views of a noisy image, and then combine these noise estimates together and denoise the image. A theoretical analysis sug-

gests that this method works precisely for views that can be written as orthogonal transformations, of which permutations are a subset. This leads to the idea of a visual anagram*—an image that changes appearance under some rearrangement of pixels. This includes rotations and flips, but also more exotic pixel permutations such as a jigsaw rearrangement. Our approach also naturally extends to illusions with more than two views. We provide both qualitative and quantitative results demonstrating the effectiveness and flexibility of our method. Please see our project webpage for additional visualizations and results:* https://dangeng.github.io/visual_anagrams/

1. Introduction

Images that change their appearance under a transformation, such as a rotation or a flip, have long fascinated students of perception, from Salvador Dalí to M. C. Escher. The appeal of these *multi-view optical illusions* lies partly in the challenge of arranging visual elements such that they may be understood in multiple different ways. Creating these illusions requires accurately modeling—and then subverting—visual perception.

In this paper, we propose a simple, zero-shot method for creating multi-view illusions with off-the-shelf text-to-image diffusion models. In contrast to most previous work on computationally generating optical illusions [3–5, 10, 12, 15, 18, 20, 28, 31, 32, 38], our method does not require an explicit model of human perception. Rather, our approach builds on work that suggests generative models may process optical illusions in a way similar to humans [14, 23, 29]. In this way, our method is similar to recent work that uses diffusion models to create optical illusions by Burgert *et al*. [2] and Tancik [42].

Our method can generate many types of classic illusions, such as images that change appearance when flipped or rotated (Fig. 1), as well as a new class of illusions which we term *visual anagrams*. These are images that change appearance under a permutation of their pixels. Image flips and rotations are a subset of these, as they can both be expressed as a permutation of pixels, but we also consider more exotic permutations. For example, we generate jigsaw puzzles that can be solved in two different ways, which we call "polymorphic jigsaws." In addition, we successfully apply our approach to generating illusions with three and four views (Fig. 1).

Our method works by using a diffusion model to denoise an image from multiple views, obtaining multiple noise estimates. These noise estimates are then combined to form a single noise estimate which is used to perform a step in the reverse diffusion process. However, we show that care must be taken in choosing these views. For one, the transformation must keep the statistics of the noise intact, as the diffusion model is trained under the assumption of i.i.d. Gaussian noise. We provide an analysis of these conditions and give an exact specification of the class of transformations supported. Our contributions are as follows:

- We present a simple yet effective method for generating multi-view optical illusions using diffusion models.
- We derive a precise description of the set of views that our method supports and provide empirical evidence that these views work.
- We consider practical design decisions, crucial to optimizing the quality of generated illusions, and report ablations on our choices.
- We provide quantitative and qualitative results, showcasing both the efficacy and flexibility of our method.

2. Related Work

Diffusion Models. Diffusion models [6, 17, 22, 35–37, 39–41] are a class of powerful generative models that iteratively convert a sample from a noise distribution to a sample from some data distribution. These models work by estimating the noise in a noisy sample, and removing the estimated noise following some update rule such as DDPM [22] or DDIM [40]. A prominent application of diffusion models has been text-conditioned image synthesis [24, 30, 36, 37]. In addition to a noisy image and a timestep, these models take a language model embedding of a text prompt as conditioning. Our approach is closely related to recent works that experiment with composing energy-based models and diffusion models [7–9, 13, 26, 27]. These approaches [9, 27] have shown that noise estimates from multiple conditional distributions can be combined together to obtain samples from compositions of the learned distributions. Our method uses a similar approach, and we apply it to the problem of multi-view illusion generation.

Computational Optical Illusions. Optical illusions serve as a testbed for understanding both human and machine perception [14, 19, 23, 29, 45]. We focus on generating illusions computationally, an area which has primarily relied on models of how our brains process external stimuli. Freeman *et al*. [12] create the illusion of constant motion in a desired direction by locally applying a filter with continuously shifting phase, relying on the observation that local phase-shifts are interpreted as global movement. Oliva *et al*. [31] propose a method to make "hybrid images," which change appearance depending on the distance they are viewed from. This method takes advantage of the multiscale nature of human perception by blending high frequencies of one image with low frequencies from another. Chu *et al*. [5] camouflage objects in a scene through re-texturing, with additional constraints on luminance as to preserve salient features of the object, and other work camouflages objects from multiple viewpoints in 3D scenes [18, 32]. Recently, Chandra *et al*. [3] design color-constancy, size constancy, and face

perception illusions by differentiating through a Bayesian model of human vision. Our method likewise generates illusions, but does not depend on an explicit model of human perception. Instead, our method works by leveraging visual priors in diffusion models learned implicitly through data. This aligns with observations [14, 23, 29] that generative models process illusions similarly to humans, and predict the same ambiguities. From this perspective, we can view our method as leveraging generative, rather than discriminative, models to synthesize adversarial examples [16] against humans [11].

Illusions with Diffusion Models. Very recently, artists and researchers have taken steps that show the potential of using diffusion models to create illusions. An artist under the pseudonym MrUgleh [43] repurposed a model fine-tuned for generating QR codes [25, 47] to create images whose global structure subtly matches a given template image. In contrast, we study multi-view illusions that can be created zero-shot from off-the-shelf diffusion models, and our illusions are specified via text rather than images. Burgert *et al*. [2] use score distillation sampling (SDS) [33, 44] to create images that align with different prompts from different views. While in principle this approach supports a superset of our views, the use of SDS results in significantly lower quality results, and the need for explicit optimization leads to long sampling times. Our method is most similar to a proof-of-concept by Tancik [42], which creates rotation illusions by sampling from a latent diffusion model [36] while alternating noise estimates between different views and prompts. While our technical approach is similar, by contrast we systematically study multi-view illusions, both by experimentally evaluating many different types of illusions and by providing a theoretical analysis of which views are (and are not) supported. In doing so, we go beyond just rotation views. We also make a number of improvements that result in qualitatively and quantitatively better illusions, such as by identifying a source of artifacts from latent diffusion, and by adding support for an arbitrary number of views. To our knowledge, we are the first to systematically evaluate illusions generated by these approaches.

3. Method

Our goal is to produce multi-view optical illusions using a pretrained diffusion model. That is, we seek to synthesize images that change appearance or identity when transformed, such as when flipped or rotated.

3.1. Text-conditioned Diffusion Models

Diffusion models [22, 39, 41] take i.i.d. Gaussian noise, $\mathbf{x}_T$, and iteratively denoise it to produce a sample, $\mathbf{x}_0$, from some data distribution. These models are parameterized by a neural network which estimates the noise in some intermediate, partially denoised data point $\mathbf{x}_t$, denoted as $\epsilon_\theta(\mathbf{x}_t, y, t)$, where y is some conditioning such as text prompts and t is the timestep in the diffusion process. The estimated noise is then used in an update rule [22, 40], from which $\mathbf{x}_{t-1}$ is computed from $\mathbf{x}_t$.

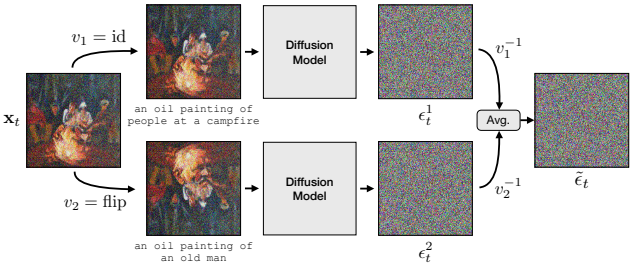

Figure 2. **Algorithm Overview.** Our method works by simultaneously denoising multiple views of an image. Given a noisy image $\mathbf{x}_t$, we compute noise estimates, ϵ_t^i, conditioned on different prompts, after applying views v_i. We then apply the inverse view v_i^{-1} to align estimates, average the estimates, and perform a reverse diffusion step. The final output is an optical illusion.

To condition the diffusion model on another input, such as a text prompt, a common approach is to use classifier-free guidance [21]. With this method, unconditional noise estimates (usually obtained by passing the null text prompt as conditioning) and conditional noise estimates are combined together:

$$\epsilon_t^{\text{CFG}} = \epsilon_\theta(\mathbf{x}_t, t, \varnothing) + \gamma(\epsilon_\theta(\mathbf{x}_t, t, y) - \epsilon_\theta(\mathbf{x}_t, t, \varnothing)). \quad (1)$$

Here, $\varnothing$ denotes the embedding of the empty string and γ is a parameter that controls the strength of the guidance. Classifier-free guidance acts to sharpen the distribution of generated images to produce higher quality results. It also enables *negative prompting* [1], in which the empty text prompt embedding, $\varnothing$, is replaced by a text prompt that we would like to discourage the model from generating.

3.2. Parallel Denoising

We produce multi-view illusions by using a diffusion model to simultaneously denoise multiple views of an image. Concretely, we take a set of N prompts, y_i, each associated with a view function $v_i(\cdot)$, which applies a transformation to an image. These transformations may be, for example, the identity function, an image flip, or a permutation of pixels. Then given a diffusion model, $\epsilon_\theta(\cdot)$, and a partially denoised image, $\mathbf{x}_t$, we combine noise estimates from different views into a single noise estimate by averaging:

$$\tilde{\epsilon}_t = \frac{1}{N} \sum_i v_i^{-1} \left(\epsilon_\theta(v_i(\mathbf{x}_t), y_i, t)\right). \quad (2)$$

Effectively, we use each view v_i to transform the noisy image $\mathbf{x}_t$, estimate the noise in the transformed images, and then apply v_i^{-1} to the estimates in order to transform them back to the original view. Taking an average of these noise

estimates gives us our combined noise estimate, which we can then use with our choice of diffusion sampler. We note that this technique of combining noise estimates is similar to previous work on compositionality [7–9, 13, 26, 27], where the idea is studied in further detail. In order to incorporate classifier-free guidance we simply replace the estimates $\epsilon_\theta(v_i(\mathbf{x}_t), y_i, t)$ with their classifier-free estimates, ϵ_t^{CFG}.

3.3. Conditions on Views

One straightforward condition for the views is that they must be invertible. But diffusion models also implicitly impose other conditions on the views $v_i(\cdot)$. We describe two such conditions below. We find that if these conditions are not satisfied, the denoising process produces poor results.

Linearity. The diffusion model, ϵ_θ, acts on noisy images, $\mathbf{x}_t$. That is, specifically images of the form:

$$\mathbf{x}_t = w_t^{\text{signal}} \underbrace{\mathbf{x}_0}_{\text{signal}} + w_t^{\text{noise}} \underbrace{\epsilon}_{\text{noise}}. \tag{3}$$

The exact values of w_t^{signal} and w_t^{noise} depend on model implementation details such as the variance schedule, but are unimportant for our work, so we omit them for clarity. What is important is that $\mathbf{x}_t$ is a linear combination of pure signal, $\mathbf{x}_0$, and pure noise, ϵ, for some specific w_t^{signal} and w_t^{noise}. Therefore our view v_i must take a noisy image $\mathbf{x}_t$ and transform it into a new noisy image $v_i(\mathbf{x}_t)$ that is also a linear combination of pure signal and pure noise *with the same weighting*. This can be achieved by requiring v_i to be a linear transformation, of the form

$$v_i(\mathbf{x}_t) = \mathbf{A}_i \mathbf{x}_t, \tag{4}$$

for some matrix $\mathbf{A}_i$, and some flattened noisy image $\mathbf{x}_t$. By linearity, we are effectively applying the view v_i to the signal and the noise separately:

$$v_i(\mathbf{x}_t) = \mathbf{A}_i(w_t^{\text{signal}} \mathbf{x}_0 + w_t^{\text{noise}} \epsilon) \tag{5}$$

$$= w_t^{\text{signal}} \underbrace{\mathbf{A}_i \mathbf{x}_0}_{\text{new signal}} + w_t^{\text{noise}} \underbrace{\mathbf{A}_i \epsilon}_{\text{new noise}}. \tag{6}$$

This results in a linear combination of transformed signal, $\mathbf{A}_i\mathbf{x}_0$, and transformed noise, $\mathbf{A}_i\epsilon$, weighted with the correct scaling factors. For further discussion, please see Appendix H.

Statistical Consistency. In addition to expecting a linear combination of signal and noise at a specific weighting, the diffusion model also expects the noise to have a precise distribution. In particular, most diffusion networks are trained with $\epsilon \sim \mathcal{N}(0, I)$. Therefore, we must ensure that our transformed noise, $\mathbf{A}_i\epsilon$, is likewise drawn from $\mathcal{N}(0, I)$. This is true if and only if $\mathbf{A}_i$ is an orthogonal matrix. We provide a proof in Appendix I, but intuitively this fact reflects the spherical symmetry of the standard Gaussian density. Orthogonal transformations, being generalizations of rotations and flips to higher dimensions, preserve this spherically symmetric density. Note that these are rotations *in the pixel values* as opposed to spatial rotations.

3.4. Views Considered

The vast majority of orthogonal transformations applied to an image will not correspond to an intuitive image transformation. However, a number of these transformations do. Below, we enumerate the orthogonal transformations which we consider, all of which can be seen in the illusions in Fig. 1 unless otherwise specified.

Identity. The simplest transformation we consider is the identity transformation. Using this view allows us to optimize the untransformed image to align with a chosen prompt.

Standard Image Manipulations. We also consider **spatial rotations** of an image, which can be viewed as permutations of pixels. This works because permutations are in turn orthogonal. However, caution must be exercised when applying a rotation view, as common anti-aliasing operations such as bilinear sampling will modify the statistics of the noise. We discuss this further in Sec. 4.4. **Spatial reflections** are also permutations of pixels. As such we can use these views to generate illusions. Finally, we implement an approximation to **skewing** by rolling columns of pixels by different displacements.

General Permutations. We have already considered the special cases of spatial rotation, reflection, and skews but we can also consider other permutations. For example, we can divide an image into jigsaw pieces and rearrange these pieces to generate jigsaw puzzles with two solutions—what we call **"polymorphic" jigsaw puzzles**. Implementation details can be found in Appendix F.

We also consider the extreme case of sampling a **completely random permutation** of pixels and treating it as our view. Additionally, we can reduce the complexity of this by considering **permutations of square patches**, rather than pixels. Examples of these illusions can be found in Fig. 6 and are discussed in Sec. 4.3.

Finally, we consider rotating a circle within an image while leaving the rest of the image stationary, which we term **inner rotations**. Note that the permutations we consider are certainly not exhaustive, and many clever transformations exist which we do not study.

Color Inversion. Negation is an orthogonal transform; it is intuitively a 180 degree rotation generalized to higher dimensions. This allows us to generate illusions that change appearance upon color inversion, assuming pixel values are centered at 0 (*e.g.*, in the range [-1, 1]).

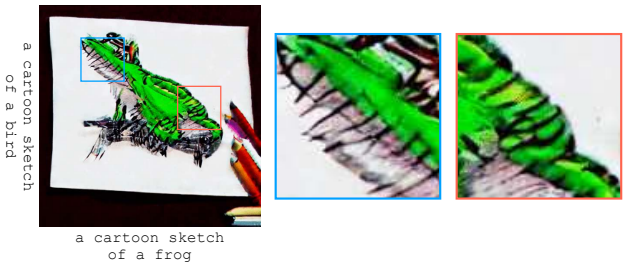

Figure 3. **Latent-Based Artifacts.** Manipulating the *location* of latent codes does not change the *orientation* of the blocks for which they encode. Therefore, when using latent diffusion models we see artifacts as shown above, in which straight lines are thatched under a rotation.

Arbitrary Orthogonal Transformations. An arbitrary rotation of an image in pixel space is uninterpretable. Nevertheless, we demonstrate that our method works for these transformations as well. While these "illusions" are inscrutable to the human eye, they serve as confirmation that any orthogonal transformation works as a view with our method. These can be found in Fig. 7 and are discussed in Sec. 4.3.

3.5. Design Decisions

Beyond the core method, we also consider various design decisions aimed at maximizing illusion quality.

Pixel Diffusion Model. Previous work [42] performed multi-view denoising using Stable Diffusion [36], a latent diffusion model. However, the latent representation effectively encodes patches of pixels. This leads to artifacts under rotations or flips, where the *location* of latents change, but the *content* and *orientation* of these blocks do not. We show a qualitative example of this in Fig. 3, in which the model is forced to generate thatched lines to produce straight lines under a 90° rotation.

To ameliorate this issue, we implement our method using a pixel-based diffusion model, DeepFloyd IF [24]. DeepFloyd denoises directly on pixels, effectively side-stepping the problem of orientation in latent code blocks.

Combining Noise Estimates. In addition to taking a mean of noise estimates from different views, we also consider alternating through them by timestep, using the estimate

$$\tilde{\epsilon}_t = v_{t \bmod N}^{-1}\left(\epsilon_\theta(v_{t \bmod N}(\mathbf{x}_t), t, y)\right). \tag{7}$$

This is the reduction strategy used by [42], but we show in ablations in Sec. 4.2 that it performs worse than averaging.

Negative Prompting. We experiment with negative prompting [1] in the 2-view case by using one view's prompt as a negative for the other view, and vice versa. This encourages the model to hide the other view's prompt for a given view. For a discussion, please see the ablations in Sec. 4.2.

Table 1. **Quantitative Results.** We report the alignment score, $\mathcal{A}$, and the concealment score, $\mathcal{C}$, as well as quantiles of these scores. For a discussion, please see Sec. 4.1.

Prompt Pair	Method	$\mathcal{A}\uparrow$	$\mathcal{A}_{0.9}\uparrow$	$\mathcal{A}_{0.95}\uparrow$	$\mathcal{C}\uparrow$	$\mathcal{C}_{0.9}\uparrow$	$\mathcal{C}_{0.95}\uparrow$
CIFAR	Burgert *et al.* [2]	0.225	0.253	0.260	0.501	0.526	0.537
	Tancik [42]	0.278	0.310	0.316	0.595	0.692	0.712
	Ours	**0.287**	**0.321**	**0.327**	**0.624**	**0.717**	**0.739**
Ours	Burgert *et al.* [2]	0.233	0.270	0.283	0.501	0.526	0.538
	Tancik [42]	0.256	0.294	0.309	0.545	0.621	0.655
	Ours	**0.275**	**0.315**	**0.326**	**0.574**	**0.668**	**0.694**

Table 2. **Ablations.** We ablate negative prompting, reduction methods, and guidance scales on our dataset.

Ablation	$\mathcal{A}\uparrow$	$\mathcal{A}_{0.9}\uparrow$	$\mathcal{A}_{0.95}\uparrow$	$\mathcal{C}\uparrow$	$\mathcal{C}_{0.9}\uparrow$	$\mathcal{C}_{0.95}\uparrow$
Negative Prompting	0.24	0.27	0.276	**0.576**	**0.659**	**0.683**
No Negative Prompting	**0.255**	**0.285**	**0.295**	0.567	0.643	0.679
Alternating Reduction	0.252	**0.286**	0.292	0.560	0.639	0.664
Mean Reduction	**0.255**	0.285	**0.295**	**0.567**	**0.643**	**0.679**
$\gamma = 3.0$	0.239	0.271	0.285	0.537	0.610	0.629
$\gamma = 7.0$	0.255	0.285	0.295	0.567	0.643	0.679
$\gamma = 10.0$	**0.259**	**0.290**	**0.297**	**0.576**	**0.664**	**0.702**

4. Results

We provide quantitative and qualitative results, and quantitative ablations. If not specified, qualitative results have been hand picked for quality. For **random samples** please see Fig. 8 and Appendix D. All implementation details can be found in Appendix A.

4.1. Quantitative Results

Metrics. We use CLIP [34] to measure how well views align with the desired prompts. We consider two metrics derived from a score matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$, defined as

$$\mathbf{S}_{ij} = \phi_{\text{img}}(v_i(\mathbf{x}))^T \phi_{\text{text}}(p_j), \tag{8}$$

where ϕ_{img} and ϕ_{text} are the CLIP visual and textual encoders respectively, returning a unit-norm vector embedding. $\mathbf{x}$ is our generated illusion, and v_i are our views with associated prompts p_i. A higher dot product indicates higher similarity between the image and text.

The first metric we consider is $\min \text{diag}(\mathbf{S})$, which intuitively measures the worst alignment of all the views. We term this metric $\mathcal{A}$, the **alignment score**. However, this metric does not account for the possibility of seeing prompt p_i in view v_j for $i \neq j$. This is an occasional failure case of our method and to quantify this we propose a second derived metric which we term $\mathcal{C}$, the **concealment score**, computed as

$$\frac{1}{N}\,\text{tr}(\text{softmax}(S/\tau)), \tag{9}$$

where τ is the temperature parameter of CLIP. In computing this metric we average both directions of the softmax, so that this metric measures how well CLIP can classify a view

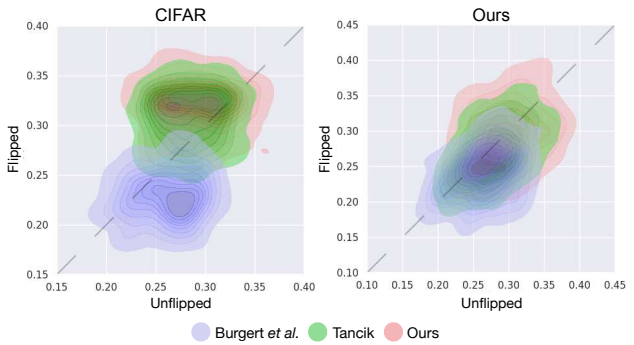

Figure 4. **Flip View CLIP Score Distribution.** We visualize trade-offs between flipped and unflipped views by plotting the distribution of CLIP scores on the datasets. Note that the quality of the flipped image is as good as the unflipped image, with parity indicated by the dashed line.

as one of the N prompts and vice versa.

Dataset. To evaluate our method and baselines we compile two datasets of prompt pairs for 2-view illusions. One dataset uses the 10 classes from CIFAR-10 and contains a prompt per pair of classes, for a total of 45 prompt pairs. We refer to this as **CIFAR**. The other dataset we compile by hand, with the process documented in Appendix B. This dataset consists of 50 prompt pairs, which we refer to as **Ours**.

Baselines. We use two baselines that generate illusions using off-the-shelf diffusion models. One, which we denote "Burgert *et al*. [2]," uses Score Distillation Sampling. The other, which we denote "Tancik [42]," is an earlier version of our method, with differences discussed in more detail in Sec. 2

Results. We show results comparing our method to baselines on both datasets in Tab. 1 using vertical flips. We use vertical flips because it is a transformation supported by our method as well as the baselines. We use 10 samples per prompt, for a total of 450 and 500 samples for the CIFAR dataset and our dataset respectively. It is hard to perform a fair comparison with more samples because the Burgert *et al*. method uses SDS, which is quite slow[1]. Because we are particularly interested in the "best-case" performance, we also report quantiles of metrics, which we denote as $\mathcal{A}_{0.9}$ for the 90th percentile, for example. As can be seen, our method performs consistently better than the baselines, in both the alignment score and the concealment score.

In order to give a clearer understanding of trade-offs when optimizing two views, we show density plots which plot the CLIP scores of each of the two views of an illusion in Fig. 4. As can be seen, we do better than the baselines

[1]Sampling just 10 images per prompt already takes more than a week of GPU-hours.

Figure 5. **Qualitative Comparisons.** We compare illusions generated by baselines to our illusions. We show examples from both our prompt dataset and the CIFAR prompt dataset.

on average and in the best-case. Moreover, flipping during denoising does not hurt performance. The quality of the flipped images is as high as the unflipped images.

4.2. Ablations

We ablate out the noise estimate reduction strategy, negative prompting, and the guidance scale in Tab. 2. We use our dataset, with 10 samples for each prompt for a total of 500 illusions.

Reduction Strategy. We find that mean reduction does better than alternating. Our hypothesis is that alternating the noise estimates results in "thrashing," causing poor convergence. Moreover, we find that the alternating strategy gives poor results on illusions with more than 2 views, as each view has fewer denoising steps. Qualitative examples of this can be found in Appendix G.

Negative Prompting. When using negative prompting, care must be taken to omit any overlap between the negative and positive prompt. For example, given the two prompts `"oil painting of a dog"` and `"oil painting of a cat"`, using one prompt as the negative for the other would simultaneously encourage and discourage the style `"oil painting"`. Rather, the negative prompts should be `"a cat"` and `"a dog"` respectively. We find that negative prompting can improve the concealment score, indi-

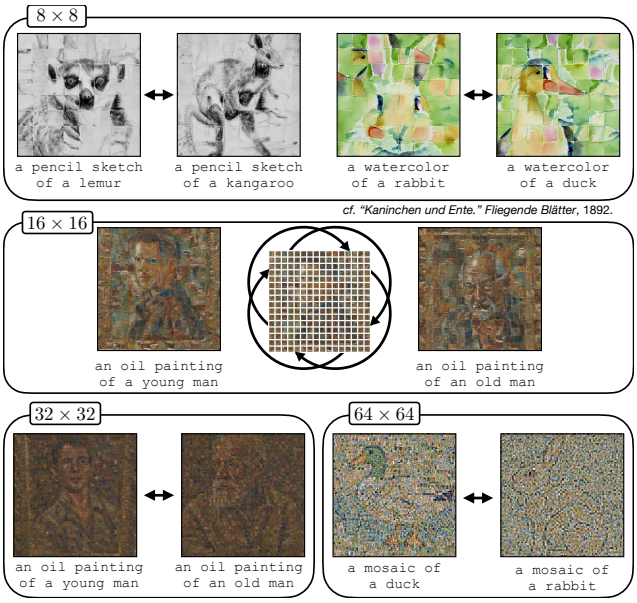

Figure 6. **Permutation Illusions.** We synthesize images whose appearance changes upon permutation of patches. Even in the difficult case of a 64×64 grid of patches, in which every pixel is effectively shuffled, we are able to generate meaningful images.

cating that it is working as intended. But this comes at the cost of worse alignment score. This is because the negative and positive prompt may have fundamental similarities. For example, using `"a cat"` as the negative prompt for the prompt `"an oil painting of a dog"` may discourage the model from synthesizing anything remotely cat-like—such as fur, four legs, or a tail—even if it helps in producing a dog. For this reason we opt not to use negative prompting with our method.

Guidance Scale. We also ablate out various guidance scales, γ, for our method. We find that a higher guidance scale tends to do better. This is presumably because a higher guidance scale results in a sharper sampling distribution.

4.3. Qualitative Results

We show qualitative results in Fig. 1, Fig. 5, Fig. 6, and Fig. 7. Again, random samples may be found in Fig. 8 and Appendix D. Additional qualitative samples can be found in Appendix C. Overall, we find that our method can produce very high quality optical illusions for a wide range of views. Interestingly, our method often finds clever ways of reusing elements from one view for another, such as in the `"waterfalls"/"rabbit"/"teddy bear"` three-view illusion in Fig. 1, in which the nose of the teddy bear is the eye of the rabbit, and a rock on the waterfalls.

Baselines. We provide qualitative comparisons of our method to baselines in Fig. 5, where we pick the best images out of 100 samples for each method. As can be seen,

Figure 7. **Orthogonal Illusions.** We show that our method works, even when the view is a randomly sampled orthogonal transformation $\mathbf{A}$. While these "illusions" are incomprehensible to human perception, they serve as a confirmation for our mathematical analysis.

images generated using our method match the prompts in both views equally well and are of higher quality.

Permutations. Pixel and patch permutations, being a subset of orthogonal transformations, should work with our method. We show that this is indeed the case in Fig. 6, where we have results on patch grids of various sizes under randomly sampled permutations. The 64×64 case is quite hard, yet our method is able to generate images that satisfy the constraint, albeit at lower quality.

Arbitrary Orthogonal Transformations. As discussed in Sec. 3.3, our method works for any orthogonal transformation. So far, we have shown illusions based on a subset of orthogonal views that correspond to intuitive image transformations. In Fig. 7, we show "illusions" using an arbitrary orthogonal transformation as a view. We use Stable Diffusion [36] and sample a random orthogonal matrix $\mathbf{A} \in \mathbb{R}^{16384 \times 16384}$ by projecting an i.i.d. random Gaussian matrix with an SVD. These dimensions correspond to the size of the Stable Diffusion latent space. We note that this is an incredibly hard and unnatural transformation of an image, and results are accordingly of lower quality, but our method is still able to produce reasonable images.

Random Samples. We show random samples for selected prompts in Fig. 8. As can be seen, these random samples, while not as good as those in Fig. 1, are still very high quality. Some failure cases can be seen where the model prefers one prompt over another. We add further discussion and present more random samples in Appendix D.

4.4. Failures

We highlight three interesting failure cases of our method in Fig. 9.

Independent Synthesis. The first of these cases involves the model synthesizing prompts separately, without combining elements of the two to form an illusion. Empirically, this happens surprisingly rarely, especially given that it seems to be such an easy shortcut solution. We hypothesize that this is because the diffusion model is biased toward centering its content, resulting in far more images with content that is integrated and centered as opposed to separate and off-center.

Figure 8. **Random Samples.** We show random samples, along with their corresponding view, for selected prompts. For more random samples please see Appendix D. **For best quality, view digitally and zoom-in.**

Noise Shift. Using views that preserve noise statistics is critical to our method's success. For example, we attempted to recreate the "Dress" illusion [46], in which a dress can be seen as either "blue and black" or "white and gold." We used simple white balancing as our view, in which pixel values were scaled by a constant factor. While this transformation is linear, it does not preserve the statistics of Gaussian noise. As a result, we see artifacts in the forms of spots, which we hypothesize is the result of the model interpreting the scaled Gaussian noise as signal and actively denoising peaks in the scaled noise.

Correlated Noise. While our method supports rotations as transformations, as demonstrated with the "3-view," "4-view," and "Inner Rotation" illusions in Fig. 1, care must be taken that the rotation does not introduce correlations in the noise, such as through anti-aliasing. For example, bilinear sampling introduces significant correlations in the noise, as it is a linear combination of four adjacent pixels. Therefore, seemingly innocuous rotations may result in divergent samples if transformations are not carefully kept correlation free, as shown with the 45 degree bilinear rotation in Fig. 9.

Figure 9. **Failures.** We highlight three interesting failure cases, which are discussed in Sec. 4.4.

5. Limitations and Conclusions

We present a method to produce compelling and diverse optical illusions. Our method is simple and straightforward to implement, and additionally amenable to theoretical analysis. We prove that our method works for a broad set of transformations, and qualitatively show that it can generate a wide array of optical illusions. However, at the same time many possible illusions and transformations are still not possible using our method, such as color constancy illusions, homographies, stretches, and more generally non-volume-preserving deformations. We leave implementation of these views for future work. Moreover, our method does not consistently produce perfect illusions. This may be a symptom of the difficulty of producing good illusions, but may indicate future work to be done to improve consistency.

Acknowledgements We thank William Henning, Trenton Chang, Kimball Strong, Jeongsoo Park, Patrick Chao, Kurtland Chua, and Mohamed El Banani for their feedback on early drafts. Daniel is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1841052.

References

[1] AUTOMATIC1111. Negative prompt. https://github.com/AUTOMATIC1111/stable-diffusion-webui/wiki/Negative-prompt, 2022. Accessed: November 7, 2023. 3, 5

[2] Ryan Burgert, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael Ryoo. Diffusion illusions: Hiding images in plain sight. https://ryanndagreat.github.io/Diffusion-Illusions, 2023. 2, 3, 5, 6

[3] Kartik Chandra, Tzu-Mao Li, Joshua Tenenbaum, and Jonathan Ragan-Kelley. Designing perceptual puzzles by differentiating probabilistic programs. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 2

[4] Ming-Te Chi, Chih-Yuan Yao, Eugene Zhang, and Tong-Yee Lee. Optical illusion shape texturing using repeated asymmetric patterns. *The Visual Computer*, 30:809–819, 2014.

[5] Hung-Kuo Chu, Wei-Hsin Hsu, Niloy J Mitra, Daniel Cohen-Or, Tien-Tsin Wong, and Tong-Yee Lee. Camouflage images. *ACM Trans. Graph.*, 29(4):51–1, 2010. 2

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[7] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 2, 4

[8] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

[9] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pages 8489–8510. PMLR, 2023. 2, 4

[10] Werner Ehm. A variational approach to geometric-optical illusions modeling. *Proceedings of Fechner Day*, 27(1):41–46, 2011. 2

[11] Gamaleldin Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial examples that fool both computer vision and time-limited humans. *Advances in neural information processing systems*, 31, 2018. 3

[12] William T Freeman, Edward H Adelson, and David J Heeger. Motion without movement. *ACM Siggraph Computer Graphics*, 25(4):27–30, 1991. 2

[13] Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. *arXiv preprint arXiv:2309.16115*, 2023. 2, 4

[14] Alexander Gomez-Villa, Adrian Martin, Javier Vazquez-Corral, and Marcelo Bertalmío. Convolutional neural networks can be deceived by visual illusions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12309–12317, 2019. 2, 3

[15] Alex Gomez-Villa, Adrián Martín, Javier Vazquez-Corral, Marcelo Bertalmío, and Jesús Malo. On the synthesis of visual illusions using deep generative models. *Journal of Vision*, 22(8):2–2, 2022. 2

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 3

[17] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022. 2

[18] Rui Guo, Jasmine Collins, Oscar de Lima, and Andrew Owens. Ganmouflage: 3d object nondetection with texture fields. *Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[19] Aaron Hertzmann. Visual indeterminacy in gan art. In *ACM SIGGRAPH 2020 Art Gallery*, pages 424–428. 2020. 2

[20] Elad Hirsch and Ayellet Tal. Color visual illusions: A statistics-based computational model. *Advances in neural information processing systems*, 33:9447–9458, 2020. 2

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 3

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 2, 3

[23] Priyank Jaini, Kevin Clark, and Robert Geirhos. Intriguing properties of generative classifiers. *arXiv preprint arXiv:2309.16779*, 2023. 2, 3

[24] Mikhail Konstantinov, Alex Shonenkov, Daria Bakshandaeva, and Ksenia Ivanova. If by deepfloyd lab at stabilityai, 2023. GitHub repository. 2, 5, 11

[25] Monster Labs. Controlnet qr code monster v2 for sd-1.5, 2023. 3

[26] Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. *Advances in Neural Information Processing Systems*, 34: 23166–23178, 2021. 2, 4

[27] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022. 2, 4

[28] Dominique Makowski, Zen J Lau, Tam Pham, W Paul Boyce, and SH Annabel Chen. A parametric framework to generate visual illusions using python. *Perception*, 50(11):950–965, 2021. 2

[29] Jerry Ngo, Swami Sankaranarayanan, and Phillip Isola. Is clip fooled by optical illusions? 2023. 2, 3

[30] 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, 2021. 2

[31] Aude Oliva, Antonio Torralba, and Philippe G. Schyns. Hybrid images. *ACM Trans. Graph.*, 25(3):527–532, 2006. 2

[32] Andrew Owens, Connelly Barnes, Alex Flint, Hanumant Singh, and William Freeman. Camouflaging an object from many viewpoints. 2014. 2

[33] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 5, 7

[37] 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, 2022. 2

[38] Troy Shinbrot, Miguel Vivar Lazo, and Theo Siu. Network simulations of optical illusions. *International Journal of Modern Physics C*, 28(02):1750018, 2017. 2

[39] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, Lille, France, 2015. PMLR. 2, 3

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. 2, 3

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2, 3

[42] Matthew Tancik. Illusion diffusion. https://github.com/tancik/Illusion-Diffusion, 2023. 2, 3, 5, 6, 12

[43] Ugleh. Spiral town - different approach to qr monster. https://www.reddit.com/r/StableDiffusion/comments/16ew9fz/spiral_town_different_approach_to_qr_monster/, 2023. 3

[44] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 3

[45] Xi Wang, Zoya Bylinskii, Aaron Hertzmann, and Robert Pepperell. Toward quantifying ambiguities in artistic images. *ACM Transactions on Applied Perception (TAP)*, 17(4):1–10, 2020. 2

[46] Wikipedia contributors. The dress. https://en.wikipedia.org/wiki/The_dress. Accessed: November 9, 2023. 8

[47] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

A. Implementation Details

We use the first two pixel-based stages of the DeepFloyd IF [24] diffusion model. Specifically, we use the first stage which produces images of size 64×64, and the second stage which upsamples images to 256×256. Our method is applied in both stages, by implementing view transformations for both resolutions. DeepFloyd IF additionally predicts the variance, along with a noise estimate. We reduce multiple variance estimates by also taking a mean. We use a classifier guidance strength between 7 and 10, and between 30 and 100 inference steps depending on the prompt. We use the M size models for both stages.

Because DeepFloyd IF also estimates variances, we need to apply inverse views to these variance estimates, in addition to the noise estimates. For pixel permutation based views, we simply apply the inverse permutation to the variance estimates. For inversion, the inverse transformation would be negating the predicted logged variance, which does not make sense. We find that simply not inverting the variance estimates works well in this case.

DeepFloyd IF additionally uses a third super resolution stage, which is the Stable Diffusion x4 upscaler. This model upscales from 256×256 to 1024×1024. Because this model is a latent model, we do not apply our method to it. However, we find that we can use it with no modification to upscale our illusions without any loss in quality in the different views. We do this by upsampling conditioned on the prompt associated with the identity view. All results in Fig. 1 have been upsampled in this way.

B. Dataset Collection

Our dataset consists of a list of styles, such as `"a street art of..."` or `"an oil painting of..."`, and a list of subjects such as `"an old man"` or `"a snowy mountain village"`. Subjects and styles were chosen by hand, using GPT-3.5 for inspiration. Prompt pairs are generated by randomly sampling a style prompt and prepending it to two randomly chosen subject prompts.

The CIFAR dataset was constructed by taking the 10 classes of CIFAR-10 as our subjects, and using the prompt `"a painting of"` as the style prompt. We take all 45 pairs of subjects, and prepend the style prompt to the subject prompts, resulting in 45 prompt pairs.

C. Additional Results

We provide additional qualitative results in this section. In Fig. 10, we compare our method to baselines, using prompts from our dataset and the CIFAR prompt dataset. This is an extension of Fig. 5. We also generate more illusions with 90° and 180° rotations, ambigrams, "polymorphic" jigsaw puzzles, color inversion, and vertical flips, which can all be found in Fig. 12. In Fig. 13, we generate

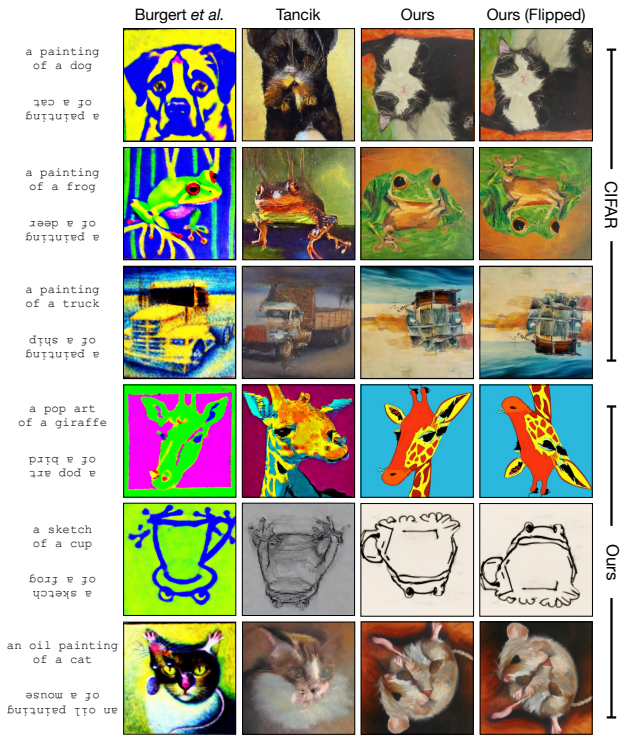

Figure 10. **Qualitative Comparisons.** We compare more illusions generated by baselines to our illusions. We show examples from both our prompt dataset and the CIFAR prompt dataset.

Figure 11. **Combining Noise Estimates.** We show that mean reduction does better than alternating with an example of a 4-view sample image.

several flip illusions with the same flipped prompt, and different unflipped prompts, and we show flipped versions of these illusions in Fig. 14.

D. Random Samples

We provide more random samples generated using our method. For rotations, color inversion, and vertical flips, please refer to Fig. 16. For three-view, inner rotation, "polymorphic" jigsaw puzzles, and patch and pixel permutation views, please refer to Fig. 17. We also provide random samples generated with prompts from the CIFAR dataset in Fig. 15.

The CIFAR prompt pair results, in Fig. 15 as well

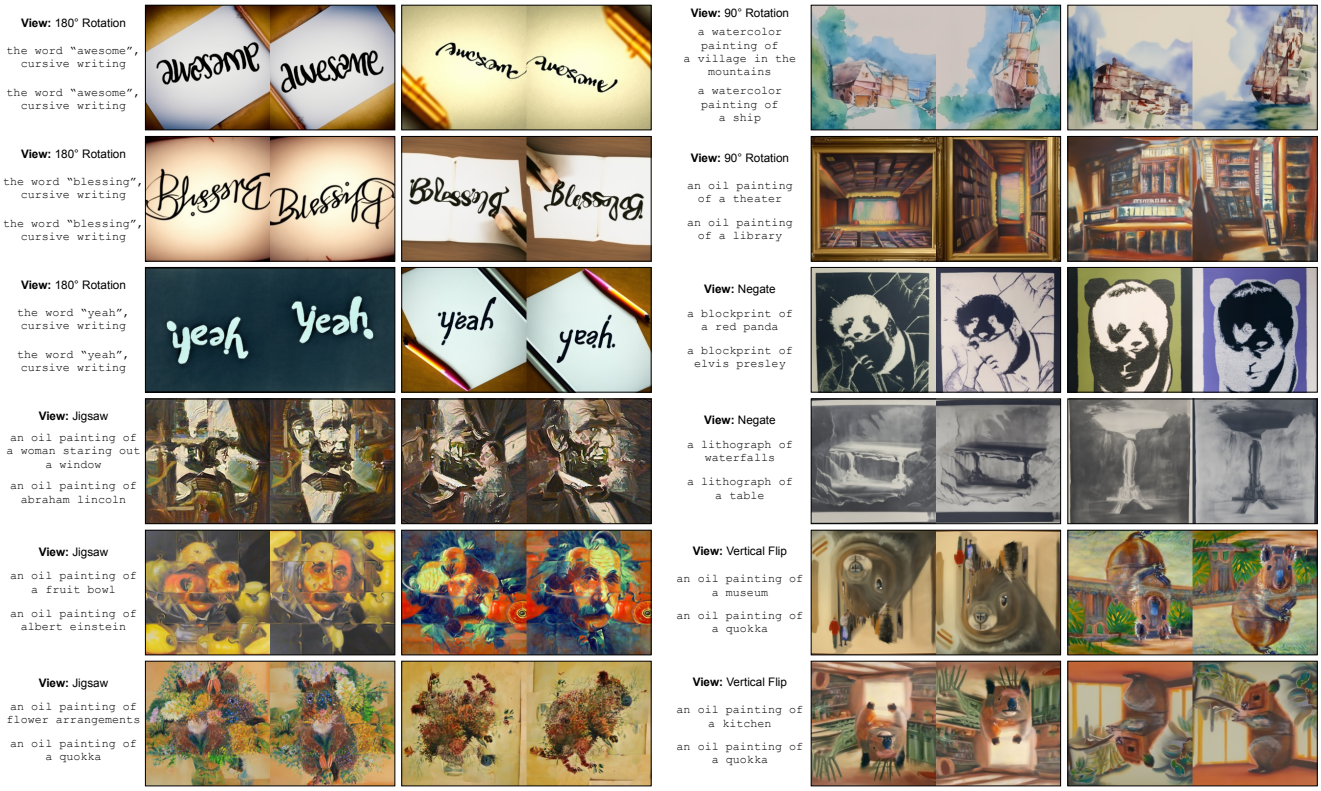

Figure 12. **Qualitative Samples.** We show more illusions with views such as rotations, flips, color inversion, and jigsaw puzzles.

as Tab. 1, are included as a proxy for random prompts. We note that systematically sampling truly random prompts for evaluation is tricky. Firstly, there is no standard method for sampling a random prompt. And secondly, not all prompt pairs make for good illusions. A straightforward example of this would be prompt pairs that differ in style. Therefore, evaluating illusion generation on completely random prompts may result in meaningless or misleading results, and as such prompts in Fig. 8, Fig. 16, and Fig. 17 are to some extent curated.

E. The Art of Choosing Prompts

We find that choosing good prompts is important to achieving good illusions. We lay out a few rules of thumb here. Firstly, it is very hard to reason as to what will make good illusions. Prompts that one may believe to work easily can fail consistently, and prompts that one may believe to have no chance of working may work fantastically. We find that more abstract styles, such as `"a painting"` or `"a drawing"` work much better than realistic styles such as `"a photo of"`. We believe this is because the constraints on realistic styles is too strong for illusions to work well. We also find that human faces make for good illusions, perhaps due to the sensitivity of the human visual system to face-like stimuli.

F. Jigsaw Puzzle Implementation

We produce jigsaw puzzles by implementing a rearrangement of puzzle pieces as a permutation of pixels. We first hand-draw three puzzle pieces—a corner, edge, and center piece—such that they can disjointly tile a 64×64, a 256×256, or a 1024×1024 image. All pieces in the puzzle are one of these three pieces, in different orientations. We then sample a random permutation of corner, edge, and center pieces respectively, and translate this permutation of pieces to a permutation of pixels.

G. Combining Noise Estimates

Rather than taking the mean of noise estimates, we also experimented with alternating or cycling through noise estimates by timestep, as is done in [42]. However, we find that this can lead to "thrashing," in which the sample is optimized in different directions at different timesteps, leading to poor quality. Moreover, in illusions with more than two views, each view gets fewer denoising steps, resulting in lower quality illusions. For example, given four prompts each matched to a rotation of the image (i.e., `"a teddy"`, `"a bird"`, `"a rabbit"`, and `"a giraffe"`), the mean reduction outputs images with higher quality than the alternating method as shown in Fig. 11.

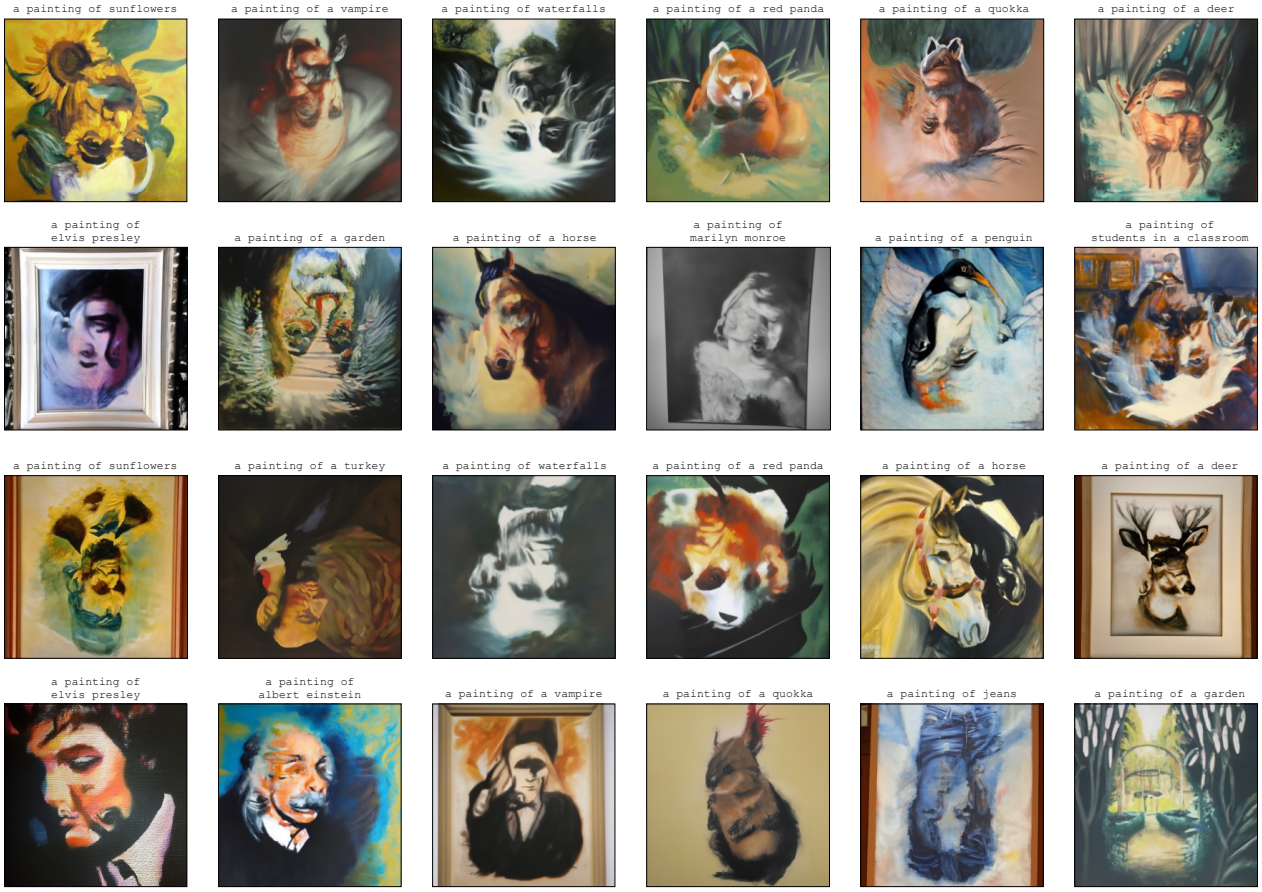

Figure 13. **Flip illusions.** For each row, the prompt of the flipped image is the same. We encourage the reader to guess what the flipped prompt is. For an answer and flipped illusions, please see Fig. 14.

H. Linearity of Views

As discussed in Sec. 3.3, when a view v is a linear transformation, it satisfies:

$$v(\mathbf{x}_t) = v(w_t^{\text{signal}}\mathbf{x}_0 + w_t^{\text{noise}}\epsilon) \tag{10}$$

$$= w_t^{\text{signal}} v(\mathbf{x}_0) + w_t^{\text{noise}} v(\epsilon). \tag{11}$$

This is convenient because applying v to the noisy image $\mathbf{x}_t$ is equivalent to applying v to the signal, $\mathbf{x}_0$, and the noise, ϵ, independently. In addition, the result is a linear combination of transformed signal and transformed noise, and is weighted as the diffusion model expects for timestep t.

However, there may be other conditions that work. For example, we could enforce

$$v(\mathbf{x}_t) = v(w_t^{\text{signal}}\mathbf{x}_0 + w_t^{\text{noise}}\epsilon) \tag{12}$$

$$= w_t^{\text{signal}} v_1(\mathbf{x}_0) + w_t^{\text{noise}} v_2(\epsilon), \tag{13}$$

with the interpretation being that v somehow acts on the signal and noise in different ways, through v_1 and v_2, and combines them with the correct weightings. We leave this for future work.

I. Statistical Consistency

We provide a proof that for $\epsilon \sim \mathcal{N}(0, I)$ and square matrix $\mathbf{A}$, $\mathbf{A}\epsilon \sim \mathcal{N}(0, I)$ if and only if $\mathbf{A}$ is orthogonal, stated in Sec. 3.3. By properties of Gaussians, $\mathbf{A}\epsilon$ is also Gaussian, so we need only compute mean and covariances. The mean is given by

$$\mathbb{E}[\mathbf{A}\epsilon] = \mathbf{A}\mathbb{E}[\epsilon] = 0. \tag{14}$$

Because the mean is 0, the covariance is given by

$$\text{Cov}(\mathbf{A}\epsilon) = \mathbb{E}[(\mathbf{A}\epsilon)(\mathbf{A}\epsilon)^\intercal] \tag{15}$$

$$= \mathbf{A}\mathbb{E}[\epsilon\epsilon^\intercal]\mathbf{A}^\intercal \tag{16}$$

$$= \mathbf{A}\mathbf{A}^\intercal \tag{17}$$

So if $\mathbf{A}\epsilon \sim \mathcal{N}(0, I)$, then we must have $\text{Cov}(\mathbf{A}\epsilon) = \mathbf{A}\mathbf{A}^\intercal = I$, or equivalently $\mathbf{A}$ must be orthogonal. And if $\mathbf{A}$ is orthogonal, then $\mathbf{A}\mathbf{A}^\intercal = I$ and $\mathbf{A}\epsilon \sim \mathcal{N}(0, I)$.

Figure 14. **Flip illusions.** Flipped illusions from Fig. 13, revealing the flipped prompt. Please refer to Fig. 13 for the unflipped images.

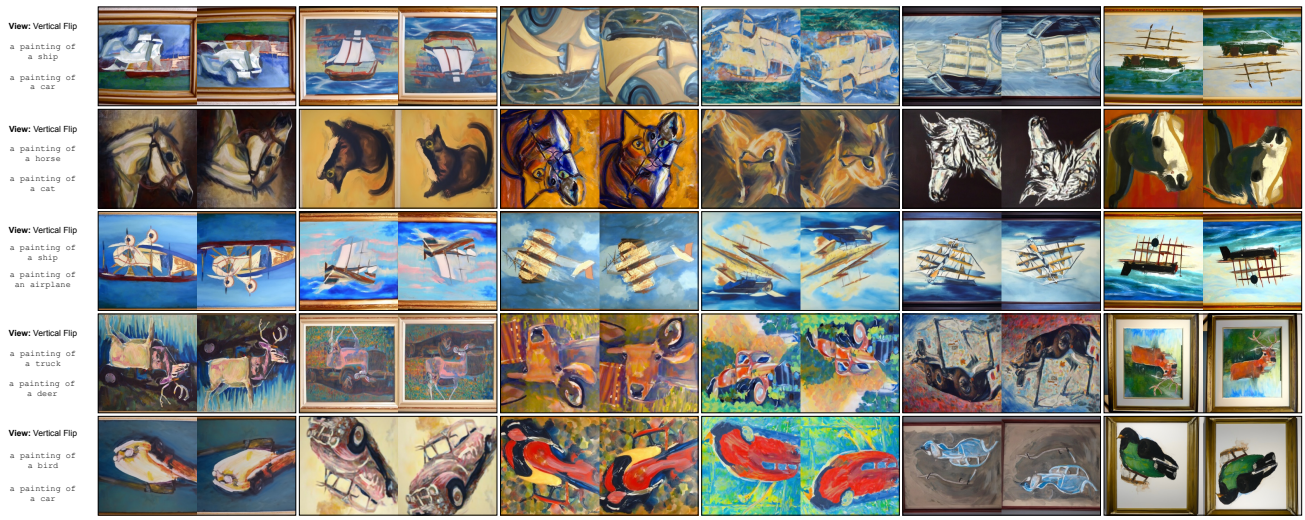

Figure 15. **Random Samples.** We provide random samples for vertical flips using prompts from the CIFAR dataset. We show both views of the illusions side-by-side.

Figure 16. **Random Samples.** We provide random samples for rotations, negations, and vertical flip views. We show both views of the illusions side-by-side.

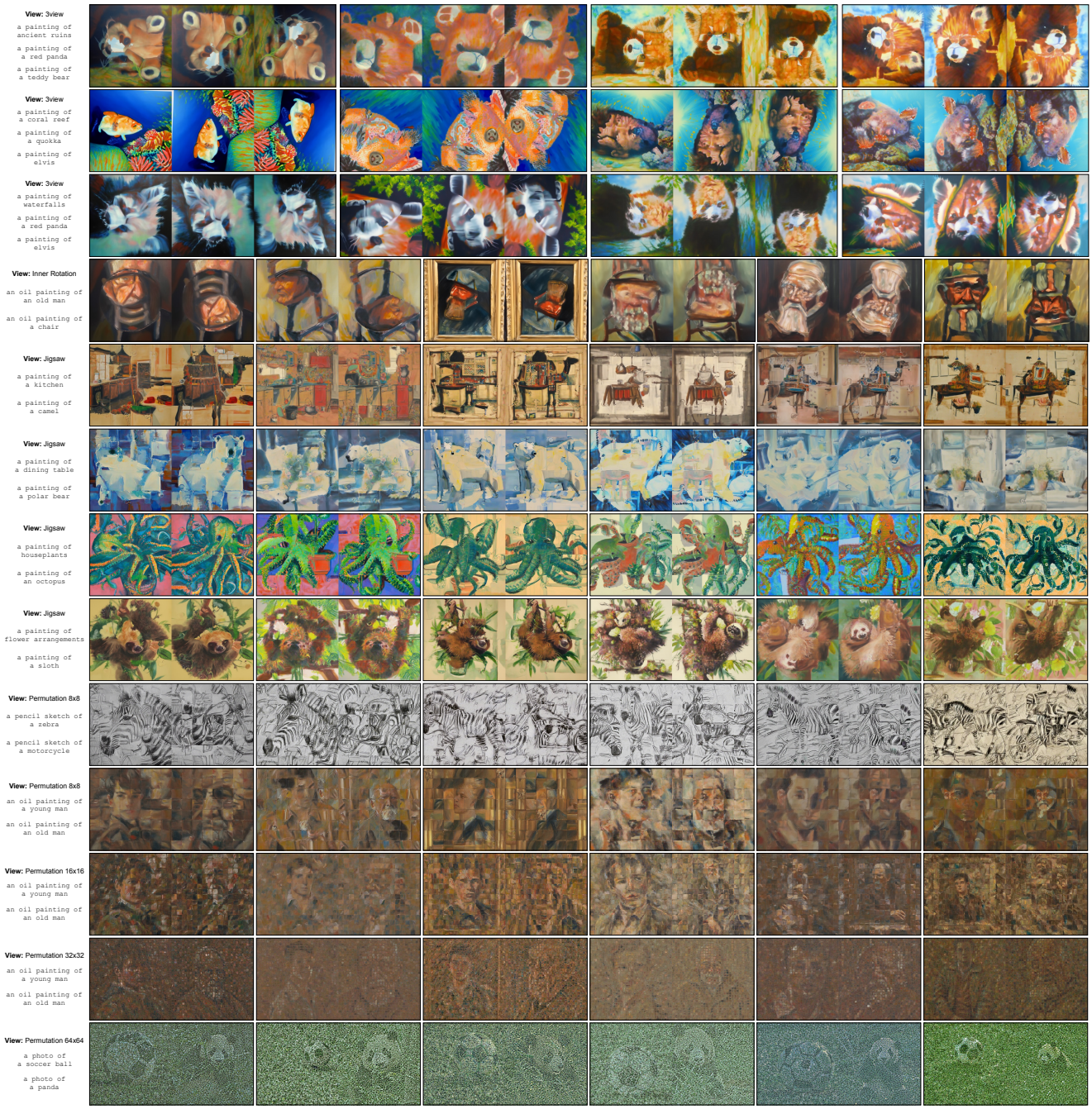

Figure 17. **Random Samples.** We provide random samples for 3-view, inner rotation, jigsaw puzzle, and patch and pixel permutation views. We show all views of the illusions side-by-side.

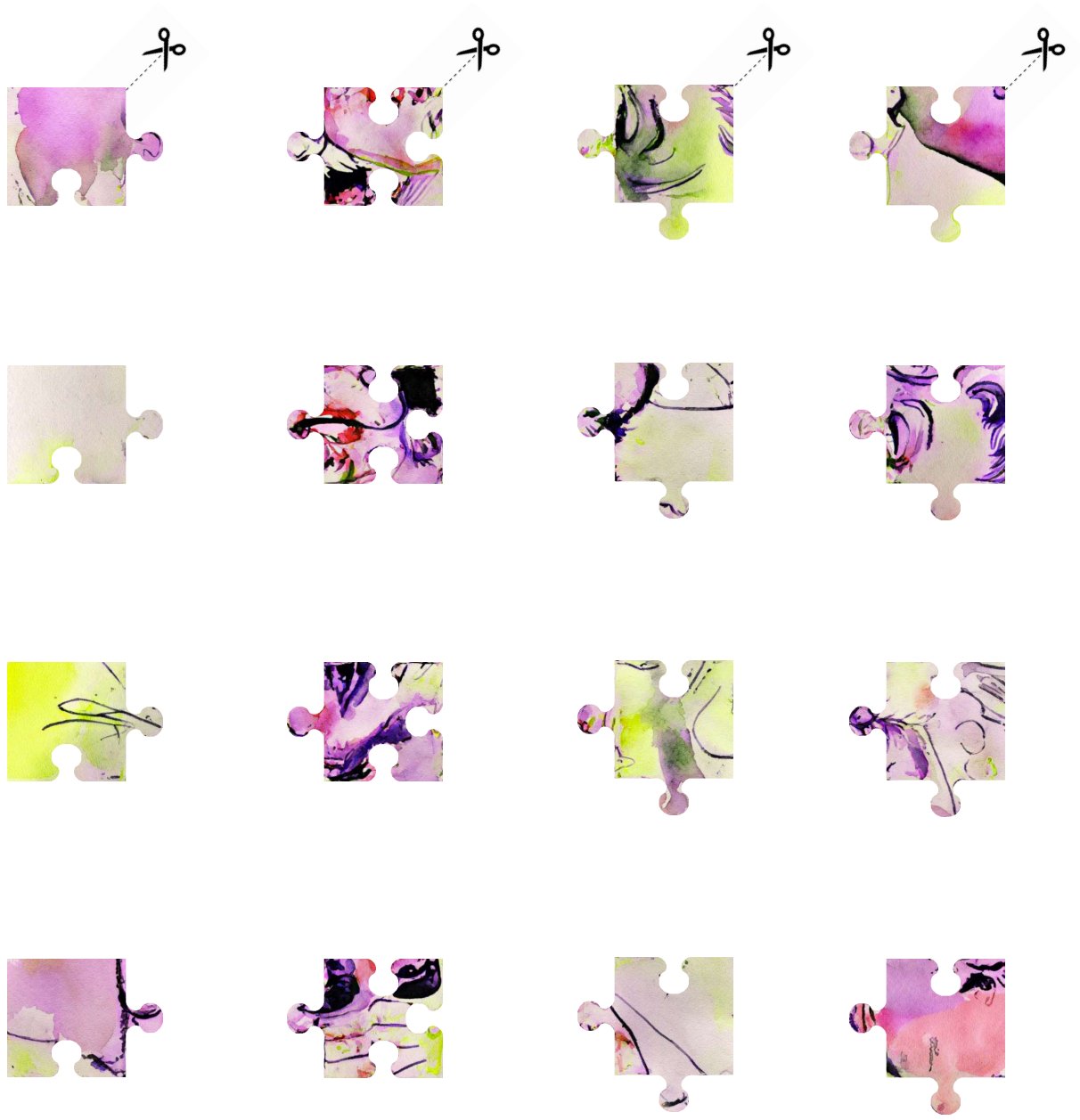

Figure 18. **Cut Your Own Polymorphic Jigsaw!** We invite the reader to cut out their own polymorphic jigsaw puzzle, and try to discover both solutions.