End-To-End Illusion Generation with Latent Diffusion Models

AUTHOR
Erik Xie, Xin Qi Liu

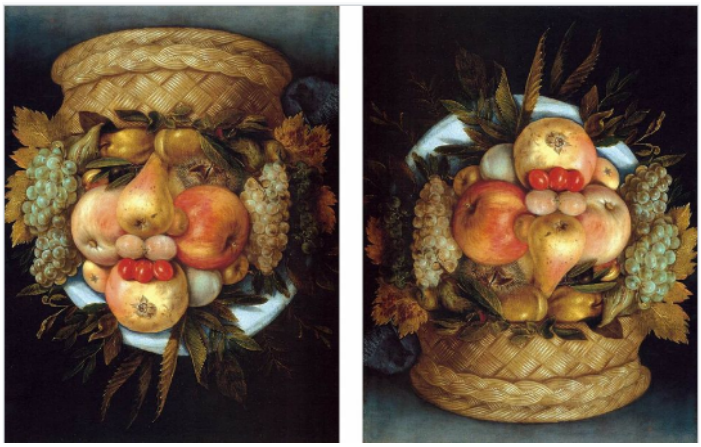
PUBLISHED
May 12, 2025

Abstract

Visual illusions serve as a bridge between perception and creativity. From the ancient Greeks' trompe-l'œil techniques, where painters like Zeuxis and Parrhasius crafted images so realistic they could deceive both humans and animals, to the Renaissance's mastery of perspective in works by Andrea Mantegna and Antonio da Correggio, challenging our visual perception has been a constant in art history. In contemporary times, computational technology has given rise to innovative methods for generating visual illusions. Recent advancements in diffusion techniques have enabled the creation of compelling illusions, however, at the cost of expensive optimization. This paper aims to contribute to this growing field by redesigning the diffusion process, eliminating the need for complex optimization, and achieving faster, aesthetically pleasing results through seamless integration into latent diffusion models like Stable Diffusion. Furthermore, we showcase the effectiveness of our approach in generating modern visual illusions in comparison to existing methods, especially in terms of efficiency. Code is available at the end of the paper in colab.

Background and Related Work

Images that can be interpreted in different ways under transformations have long been the interest of artists and researchers. The capacity to hide semantic meanings in plain sight takes careful engineering, which has been explored since the 1500s by Guiseppe Arcimboldo. In this paper, we explore the challenge of generation them computationally. Specifically, we focus on two types of illusion, flip illusions, which are images that have two appearances rightside-up and upside-down, and overlay illusions, which are images that each have their own appearances, but give rise to another appearance when combined on top of each other.



The Fruit Basket, a c.1590 oil-on-panel still life by the Italian painter Giuseppe Arcimboldo, source: Wikipedia

Previous works in computational illusion generation mostly rely on a powerful class of generative models called diffusion models, which convert samples from a noise distribution to an image distribution. They work by iteratively estimating the added noise from the noisy sample, and removing it, tyically using rules given by specially designed schedulers, like DDPM (Ho, Jain, and Abbeel 2020). Diffusion models have been applied universally in text-based image generation, which is a task highly correlated with ours of generation illusions, where different arrangments of images correspond to different text prompts. Previously, Burgert et al. (Burgert et al. 2023) used post-processing to generate images with distillation of diffusion models, forcing the output images to align with different views when given different prompts. Their method achieves good results but requires explicit optimization, which consumes large amount of compute. Geng et al. (Geng, Park, and Owens 2024) have integrated generation into pixel diffusion models like DeepFlyod. They have achieved state of the art results, but suffer from similarly significant compute requirements, often taking longer to generate single images of decent quality. This is due to pixel space diffusion models process uncompressed image data, which is more expensive. In this paper, we explore the integration of illusion generation into latent diffusion models, which denoise in a compressed latent space, to improve the speed and compute requirement. In this article, we propose a technically similar approach, but improve the computation requirements by generalizing the generation pipeline to diffusion models in the latent space. Most diffusion models released recently falls under the class of latent diffusion models (Podell et al. 2023), which uses autoencoders to compress the images down to more efficient representations. Our framework can thereby provide more options and flexibility, along with more efficient generation of illusions of user chosen prompts.

Method

There are several key challenges when it comes to rearrangment illusions involving a latent diffusion model, such as Stable Diffusion. The main one being that the encoder does not conserve the spatial properties of the image (i.e. flipping and encoding do not commute). This is due to the architecture involved in the variational autoencoder, which is typically used in latent diffusion models. Mathematically speaking, let I denote the space of images (e.g., $R^{H\times W\times C}$) and Z denote the latent space. The encoder is then a mapping $E : I \to Z$. Let $F : I \to I$ represent a spatial flip operation, Mathematically, E would be F-equivariant if there existed a corresponding transformation $F_Z : Z \to Z$ in the latent space such that for all images

$$I \in I : E(F(I)) = F_Z(E(I))$$

As shown by a line of research (Weiler and Cesa 2019), this property is not a part of a conventional conventional

neural network for common rearrangments like rotations, flipping, or overlaying. The non-linear mapping between image and latent spaces, however, serves as the backbone of the autoencoders used by the most used diffusion models in the market. Simply adapting existing method will not work with them.

We propose several ways to overcome this problem. (1) The naive way is to flip the latent representation directly, which suffers from artifacts, as shown in previous studies for aforementioned reasons, but benefits from its computational simplicity. (2) Another approach is to decode the latent into the pixel space, in which it would be rotated. However, repetitively encoding and decoding the latents at each timestep results in loss of information, as some details are lost each time at the bottleneck of the autoencoder. (3) Furthermore, re-training the autoencoder using rotation equivariant architectures, such as `e2cnn`, to enforce the F-equivariance property. This approach is promising but requires large scale of compute (hundreds of GPUs and billions of images, i.e. `LAION-5b`). (4) Lastly, a regularization approach (Kouzelis et al. 2025) can be considered, where the pretrained autoencoder is finetuned on augmented data (i.e. rotated, noised) with specialized loss functions to enforce equivariance in the latent space. Since it can be difficult to ensure the retrained or finetuned VAE does not deviate from the denoising network's expected distribution in approaches (3) and (4), we adopt approach (2) in this paper and leave the two other approaches to further research.

The approach can be formally formulated as the following: Let I represent the image space, and let N be the space of latent images. The tasks consists:

1. A set of target images (or conceptual targets) $T = \{I_i\}$. For simplicity, consider a single conceptual target I, as in a flip illusion.

2. A set m of invertible spatial transformations $\{a_1, a_2, a_3, \cdots\}$, where each $a_i : I \to I$ has an inverse $a_i^{-1} : I \to I$ (acting spatially on image space).

3. A set of prompts or conditioning signals $P = \{p_i\}$, including negative prompts, if applicable.

4. A noise prediction model $\epsilon_\theta(x_t, t, p)$, parameterized by θ, which predicts the noise added to an image x_t at timestep t given prompt p, in the space of N.

The objective is to determine the overall noise estimate ϵ' for the current noisy latent state x_t at a given diffusion timestep t, while considering guidance from all transformations m and prompts P. Our approach involves the following steps, which are also outlined in figure 2:

1. Noise Predictions: For each transformation $a_i \in m$, predict the noise for $a_i(I_i)$ using ϵ_θ. Let p_i be the conditioning corresponding to transformation a_i. The predicted noise associated with the i-th transformation is:

$$\epsilon^i = \epsilon_\theta(a_i(x_t), t, p_i), i = 1, \ldots, m$$

2. Clean Images: Use the scheduler's timestep information to compute the clean latent image. For DDPM, this formula is :

$$x_0^i = \frac{x_t^i - \sqrt{1 - \alpha_t \cdot \epsilon^i}}{\sqrt{\alpha_t}}$$

   The clean images are then decoded into the pixel space.
3. Inverse Transformation: Apply the inverse transformation a_i^{-1} to each corresponding predicted clean image x_0^i

4. Image Blending: Compute the updated images by blending the inversely transformed clean images. For simple transformations like rotation, simply taking the average suffices: $x_0' = \frac{1}{m'} \sum_{i=1}^{m'} x_0^i$, where $m' = |m|$.
5. Noise Update: Using the encoder, x_0' is mapped back to N. For each latent view of the illusion, the effective noise, ϵ', is simply the diffference between the original latents and its respective x_0'.

The resulting ϵ' is then used in the diffusion sampling step to estimate the less noisy image x_{t-1}. This process is repeated for each timestep t during the reverse diffusion process.

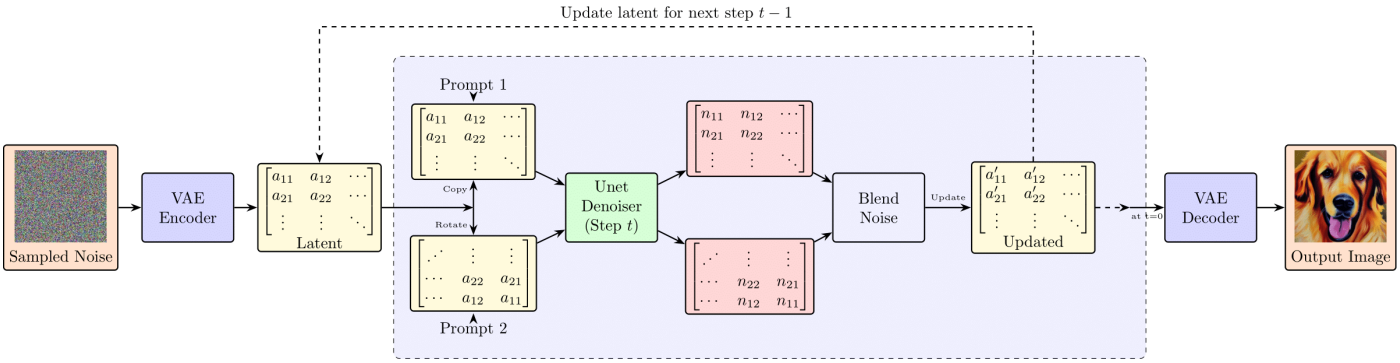


Figure 2: Diffusion Pipeline for a Flip Illusion, the above procedure describes the Blend Noise module

We can rationalize this approach with a few reasons: first, our framework benefits from the efficiency of latent diffusion models, as described above. Second, manipulating the images outside the latent space provides our model with less constraints. In the latent space, simple rotations results can result in significant artifacts (See figure 3). Whereas most applicable transformations are straightforward in the latent space. Second, this allows us to introduce one more augmentation – pixel shuffling. For overlay illusion, we can shift the images by a small, randomized amount of pixels prior to overlaying, which simulates how illusions can be produced in the real world, and ensures the robustness of the results that do not rely on solutions that require pixel-accurate overlaying. In addition, we can take advantage of negative prompting to better conceal the illusions. The negative prompts discourage one rearrangment to look like the others, which can achieve more compelling results (Geng, Park, and Owens 2024)

Figure 3: artifacts in the latent space, where the denoised latent is rotated 180 degrees prior to decoding

We provide another modification to our approach to generate other illusion effects such as Overlay Illusions. An example of overlay illusion is provided below, using 4 images to generate the 5-th image (without loss of generality):

1. Prepare 5 copies the same latent that are sampled from N.

2. For each image, we will calculate the noise predictions using Classifer-Free Guidance (CFG), using the corresponding prompt for each.

3. Using the noise prediction, we will now calculate the clean-image prediction for each of the images, using the method mentioned above. We use denote the clean-image predictions as x.

4. We will now use the 5 clean-image prediction to solve a $\mathbf{L}^2$ minimization problem, instead of simple averaging for rotation illusions. Specifically, we want to make sure images L_1, L_2, L_3, L_4 have the property such that $L_1 L_2 L_3 L_4 = k L_5$ for some $k \in \mathbb{R}$. This will gave us a new set of images x' that that follows the constraint as closely as possible. We use an iterative solver with gradient steps for this, which was also used in Diffusion Illusion (Burgert et al. 2023) (see Appendix for further details). And we choose $k = 3$ empirically, as it represents the weight of the overlaid image with respect to the individual images.

5. In order to update each image under the constraint that the overlay of each image is equal to the 5-th image, we must recalculate the noise predictions. To do this, we calculate the sample prediction (see appendix) to explicity calculate the new noises $\hat{\epsilon}'$ given the updated latents.

6. Each images are updated with their respective noise $\hat{\epsilon}_i{}'$, and the steps are repeated until $t = 0$.

Experiments

We conduct experiments using `Stable Diffusion v1.5` as ϵ_θ and an RTX 2080 GPU. We generated illusions of resolution [512, 512] using 80 timesteps, which is sufficient for our model to produce reasonable outputs. Note that the results can likely be significantly improved by using more recent models like `Stable Diffusion v3.5`, but we are limited by computational resources, and show results here mostly as a proof of concept. Integration into large models should require no further modifications.

Baselines

Two contending baselines are Visual Anagrams (Geng, Park, and Owens 2024), a similar illusion generation model, that uses a pixel diffusion backbone. As well as Diffusion Illusion (Burgert et al. 2023), which uses a latent diffusion model with distillation techniques. Due to the limit to our compute, we will be limiting our runtime to a single GPU on each baseline. This will also help standardize and visualize how much faster our algorithm is at generating illusions compared to comtemporary work. The runtime of illusion generation is reported in figure 4:

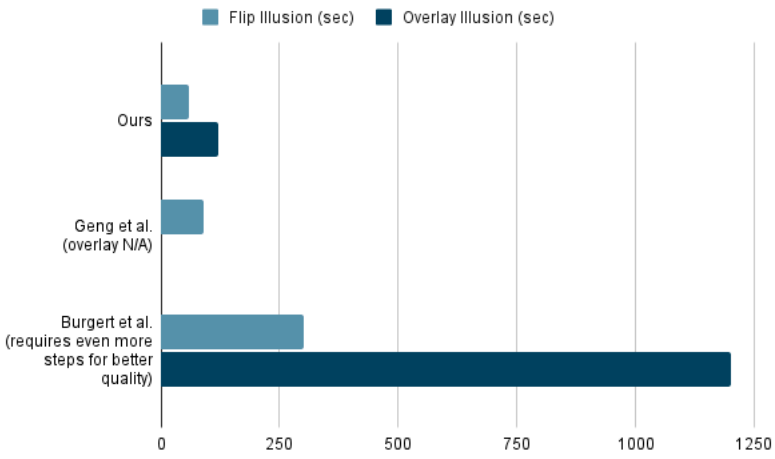


Figure 4: time (seconds) to generate illusions

Flip Illusion Gallery

Figure 5: Left: ours (prompts: dog vs sloth, old man vs campfire, kitch vs garden), Middle: diffusion illusion (prompt: boat vs boat), Right: visual anagrams (prmopt: campfire vs old man)

Overlay Illusions Gallery

The primary contemporary work addressing overlay illusions is Diffusion Illusions (Burgert et al. 2023), whereas Visual Anagrams, as its name suggests, focuses primarily on flip illusion. We present images generated by our model and those produced by Diffusion Illusions at iteration step $t = 80$, which already takes significantly longer. We measure the average CLIP score (Radford et al. 2021) for image-prompt correspondance for our method and diffusion illusion and report an average of 32.869 and 21.2416, respectively on the prompts in the gallery. This illustrates that our method achieves higher quality in the same timestep than diffusion illusion.

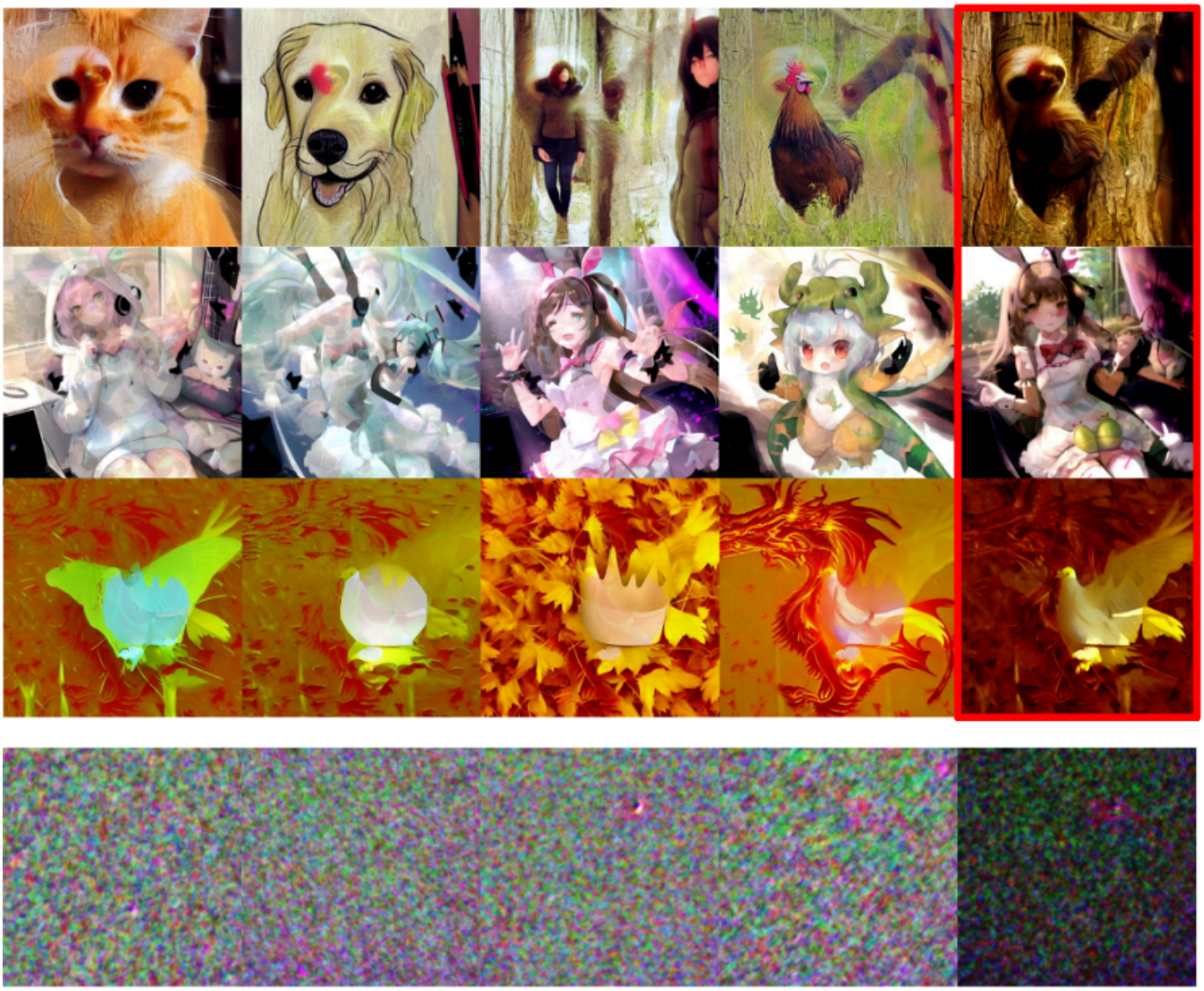


Figure 6: Overlay images highlighted in red. Top: ours (prompts: 1. cat, dog, lady, chicken, sloth, 2. anime girls (using a finetuned version of SD1.5)), Bottom: Diffusion Illusion

Discussion

The primary finding from our experiments is that our method significantly outperforms previous optimization-based approaches in terms of speed, without compromising image quality. Our model generates satisfactory results in approximately 40–80 iterations, whereas prior methods often produce noisy images even after extensive computation. This performance disparity is largely due to the computational intensity of traditional optimization processes, which involve calculating gradients and backpropagating through the image—either in pixel space or using Fourier feature representations. These operations are computationally expensive and time-consuming.

In contrast, our approach eliminates the need for such optimization steps by directly modifying the diffusion process to generate images. By avoiding iterative optimization, we achieve faster inference times while maintaining high-quality outputs. This efficiency makes our method more suitable for real-time applications and resource-constrained environments. In addition, adaptation to latent diffusion models allows users to choose from most contemporary diffusion models without the previous constraint.

Future Directions

Encoding and decoding the image representations can result in a loss of details with long timesteps, which can degrade the quality of illusions in our framework. In our methods section, we proposed equivariance-oriented approaches, which require more fundamental changes to the standard latent diffusion models. Future research can focus on creating latent models that can create illusions without relying on the autoencoder during denoising. In addition, aspects of our images other than image-prompt correspondance can be evaluated using multimodal models like CLIP to provide more comprehensive benchmarks, which we did not do due to limited compute resources. Lastly, additional illusion types, like 90 degree rotations, folding tiles, etc. can be explored in future studies for more diversity in illusion generation.

Appendix

In diffusion models, the prediction type parameter determines the target the model aims to predict during the denoising process. The most common prediction types are:

- ϵ: The model predicts the noise added to the image.
- c_0: The model predicts the original, clean image directly.
- $v_{\text{prediction}}$: The model predicts a combination of the noise and the original image, often leading to improved performance.

Here, we explain methods of conversion from different prediction objectives to calculate the predicted clean image. Let x be the latent and c be the clean image prediction. Let $\bar{\alpha}$ be the cumulative noise product, and $\bar{\beta} = 1 - \bar{\alpha}$. We have:

$$\epsilon \to c = \frac{x - \sqrt{\bar{\beta}}\epsilon}{\sqrt{\bar{\alpha}}}$$

$$v_{\text{prediction}} \to c = \sqrt{\bar{\alpha}}x - \sqrt{\bar{\beta}}\epsilon$$

The second method of explicity calculting the clean image is the clean image prediction and need not be modified.

Likewise, we can explicity calculate the noise as follows:

$$c_0 \to \epsilon = \frac{x - \sqrt{\bar{\alpha}}\epsilon}{\sqrt{\bar{\beta}}}$$

$$v_{\text{prediction}} \to \epsilon = \sqrt{\bar{\alpha}}\epsilon + \sqrt{\bar{\beta}}x$$

Simiarly, this is no need to convert the ϵ prediction. This allows our method to adapt to various models in existence without major modifications.

Algorithm 1 (Flip) Code Availability

https://colab.research.google.com/drive/12n89l8AW--ttID4NLCT_VMaOZ_R2BKOK?usp=sharing

Algorithm 2 (Overlay) Code Availability

https://colab.research.google.com/drive/1nFqpHsuhn0HgjqNzmeEpFroM6kSWIVMr?usp=sharing

---

References

Burgert, Ryan, Xiang Li, Abe Leite, Kanchana Ranasinghe, and Michael S. Ryoo. 2023. "Diffusion Illusions: Hiding Images in Plain Sight." https://arxiv.org/abs/2312.03817.

Geng, Daniel, Inbum Park, and Andrew Owens. 2024. "Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models." In *Conference on Computer Vision and Pattern Recognition (CVPR)*. https://arxiv.org/abs/2311.17919.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. "Denoising Diffusion Probabilistic Models."

Kouzelis, Theodoros, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. 2025. "EQ-VAE: Equivariance Regularized Latent Space for Improved Generative Image Modeling." In *Arxiv*.

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

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, et al. 2021. "Learning Transferable Visual Models from Natural Language Supervision."

Weiler, Maurice, and Gabriele Cesa. 2019. "General E(2)-Equivariant Steerable CNNs." In *Conference on Neural Information Processing Systems (NeurIPS)*.