

PartStickers: Generating Parts of Objects for Rapid Prototyping

Mo Zhou^{1*}, Josh Myers-Dean^{1*}, Danna Gurari^{1,2}

¹University of Colorado Boulder, ²The University of Texas at Austin

*Denotes Equal Contribution

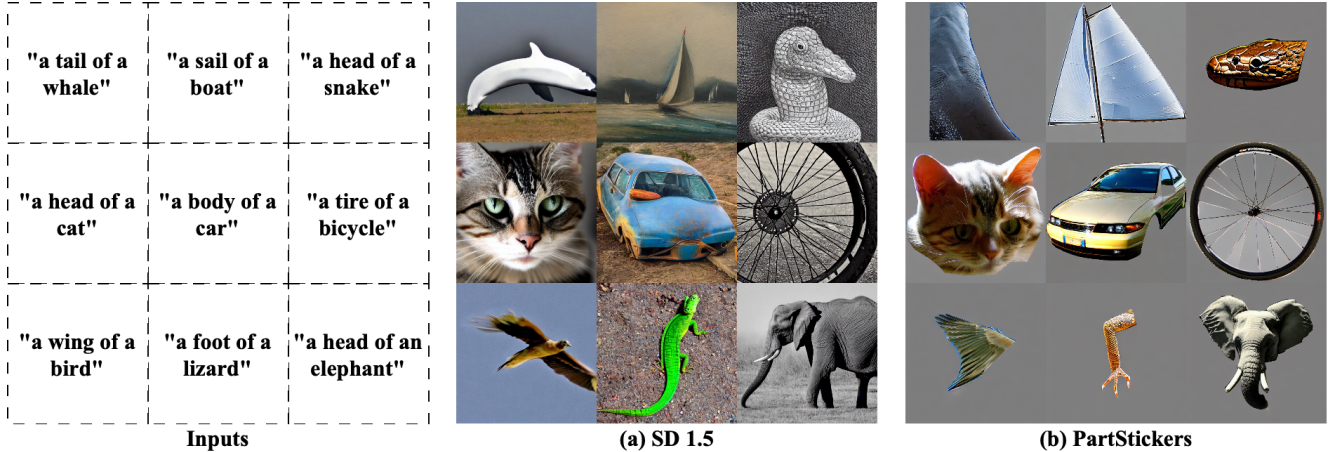


Figure 1. Given text prompts on the left of parts of objects, we show results from (a) a baseline, Stable Diffusion (SD) 1.5, showing that: generated parts can 1) be unrealistic (top), 2) include unwanted information, such as obfuscated content (middle) and 3) show the entire object, rather than just the part of interest (bottom). In contrast, results from (b) our method consistently produce realistic parts on neutral backgrounds, isolated from their parent objects.

Abstract

Design prototyping involves creating mockups of products or concepts to gather feedback and iterate on ideas. While prototyping often requires specific parts of objects, such as when constructing a novel creature for a video game, existing text-to-image methods tend to only generate entire objects. To address this, we propose a novel task and method of “part sticker generation”, which entails generating an isolated part of an object on a neutral background. Experiments demonstrate our method outperforms state-of-the-art baselines with respect to realism and text alignment, while preserving object-level generation capabilities. We publicly share our code and models to encourage community-wide progress on this new task: <https://partsticker.github.io>.

1. Introduction

Design prototyping involves creating mockups of products or concepts to gather feedback and rapidly iterate on ideas, ultimately refining the final outcome [6, 8, 64]. This low-

fidelity approach minimizes the need for polished assets at each step, accelerating creativity and decision-making in fields such as product design, education, and entertainment [3, 12, 54, 55]. Generative AI tools have emerged as a popular tool in rapid prototyping due to their proficiency in generating diverse objects and scenes from linguistic input [4, 7, 23, 37, 48, 58].

Towards increasing the applicability of generative AI for design prototyping, we aim to address three limitations of existing methods. *First*, most text-to-image pipelines cannot directly generate *isolated parts* (e.g., a wing detached from a bird). Yet, such finer-grained, part-level renderings can also be valuable for prototyping, such as for product development. *Second*, existing approaches supporting part-level generation often either compromise on realism or include extra, unwanted regions (e.g., head and shoulders instead of just the head), necessitating extra work before such rendered parts can be incorporated into a design. *Third*, text-to-image pipelines typically produce non-neutral backgrounds, necessitating extra work to isolate rendered parts from the backgrounds. These three limitations are highlighted in **Figure 1(a)**, with results from an existing state-

of-the-art model (Stable Diffusion): the top row shows unrealistic parts; the middle row includes incomplete regions, such as a fraction of the desired tire of the bicycle; and the bottom row displays unwanted extra regions, such as the entire object.

To overcome these limitations, we propose the novel task of *part sticker generation*, which requires a system to create a single, accurately isolated part on a plain background given only a textual prompt. While numerous pay-per-use datasets exist for object-level “stickers” (e.g., Adobe Stock, Noun Project), free alternatives are limited, and few, if any, provide parts (or offer the ability to generate parts on demand). Our contributions are three-fold. *First*, we introduce **PartStickers**, the *first dedicated method* for isolated part generation that simultaneously provides realism and diversity. *Second*, by generating parts against a neutral background, PartStickers removes the need for manual segmentation to isolate rendered parts, enabling drag-and-drop capabilities that could streamline rapid design prototyping. *Third*, we benchmark our approach against state-of-the-art text-to-image models, identifying common failure modes and discussing avenues for future improvements. As shown in **Figure 1(b)**, our method produces realistic, complete, and precisely isolated parts.

Success in our proposed task can yield numerous benefits for the design space. For instance, designers can assemble different generated parts to rapidly prototype novel concepts, similar to how physical LEGO blocks can be combined to create unique structures. Additionally, by simply varying the text prompt, users can quickly produce multiple variations (e.g., shapes, textures, or styles) of a specific part and select the best fit. This low-cost, on-the-fly variation supports rapid A/B testing of design ideas in a visual manner. It also encourages remixing and exploration: for example, a game artist could generate bird, dragon, or airplane wings and attach them to a creature concept, thereby expanding creative possibilities beyond conventional assets.

2. Related Work

Text-to-Image Generation. Text-to-image generation methods aim to synthesize realistic images from natural language descriptions. There has been rapid progress on this task since the introduction of Generative Adversarial Networks [14] (GANs), an adversarial training framework that pits a generator against a discriminator to produce increasingly plausible images. Building on these foundations, subsequent works incorporated text encoders [27, 44, 47, 57, 66] and attention mechanisms [10, 13, 36, 65] to better align generated content with linguistic input, improving visual fidelity and text relevance. More recently, diffusion [2]-based models have achieved impressive results by iteratively denoising randomly sampled noise [19, 33, 53, 68], leveraging large-scale image-text datasets to produce high-fidelity out-

puts [43, 49]. While these methods excel at generating entire scenes or objects, they offer limited control when users only need to generate specific components, such as parts of an object [51, 60]. In contrast, our work focuses on generating standalone parts directly against a neutral background, eliminating the need for manual intervention while maintaining fidelity to the text prompt.

Segmentation-Based Generation Models. Numerous approaches leverage segmentation to guide image synthesis [9, 15, 31, 67]. Early works introduced semantic masks or label maps as constraints to control layouts [34, 62], enabling users to define rough outlines of objects before rendering them [24]. For example, segmentation-aware GANs [25] and conditional architectures can produce images aligned with user-provided masks [41, 52, 68, 69], improving structural accuracy. Although these methods improve realism and controllability, they generally expect explicit segmentation inputs from users (i.e., the models rely on this guidance) and still focus on generating entire objects or complex scenes, rather than just requiring text input. In contrast, our approach capitalizes on part-level annotations during training to directly generate isolated object parts on plain backgrounds when *only given text prompts* (i.e., no segmentation mask inputs).

Generation of Images with Plain Backgrounds. Many works aim to produce images with plain or fully controlled backgrounds to support diverse, often creative, tasks like e-commerce photography and compositing. Early techniques often relied on segmentation masks or ‘green screen’ setups, requiring users to remove unwanted backgrounds by hand before compositing the subject on a plain canvas [1, 39]. More recently, GAN-based methods like SPADE [34] have enabled semantic image synthesis from user-defined segmentation maps, effectively granting fine-grained control over foreground and background regions. Inpainting approaches like InstructPix2Pix [5] use textual prompts to refine or erase backgrounds without altering the main subject. Although these methods provide adjustable backgrounds, many still require substantial user input (e.g., drawing segmentation outlines) or do not offer mechanisms for generating *plain* backgrounds. In contrast, our work automates the plain-background process by directly generating ‘stickers’, isolated parts placed against a neutral canvas that can be trivially separated by automated methods, thus removing the manual overhead of removing backgrounds by hand.

3. Method

We now introduce our **PartStickers** framework, describing its model architecture, data generation pipeline, and training strategies. An overview is shown in **Figure 2**.

3.1. Background: Diffusion

Our proposed framework leverages fundamental advancements in diffusion modeling [2, 45]. Diffusion approaches frame image generation as a process of gradually denoising samples, beginning from pure noise and iteratively refining them into coherent images. Following [19], let $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ be an image sampled from the ground truth distribution (e.g., dataset) q . Forward diffusion processes progressively add Gaussian noise over T timesteps, forming a Markov Process with latent variables $\mathbf{x}_1, \dots, \mathbf{x}_T$. For every time step $t \in \{1, 2, \dots, T\}$, we sample

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) \sim \mathcal{N}(\sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \quad (1)$$

where β_t is the variance scheduler controlling the noise at time step t and \mathbf{I} is the identity matrix with the same dimension as the latents, ensuring that noise is added isotropically (all latent channels are treated equally). This process yields a nearly Gaussian latent vector \mathbf{x}_T after T timesteps, regardless of \mathbf{x}_0 .

Diffusion models generally learn the reverse process:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t), \quad (2)$$

where p is parametrized by the parameters θ belonging to a denoising network (e.g., a UNet [46] in our setting). This reverse process iteratively removes noise to reconstruct a de-noised latent vector, \mathbf{x}_0 . A common realization of this objective is expressed via the simplified denoising loss:

$$\mathcal{L}_{denoise} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), t \sim \mathcal{U}(\{1, \dots, T\})} [\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2], \quad (3)$$

where ϵ_θ is a time-conditioned network that predicts the Gaussian noise ϵ at timestep t , where t is uniformly sampled from $\{1, \dots, T\}$. Once the model learns to remove the added noise, the resulting vector \mathbf{x}_0 can be decoded back to image space, thereby completing image generation.

3.2. PartStickers Framework

The core contribution of our framework is our training data generation pipeline. We design it to support our task of part sticker generation, by generating a single part specified in a text prompt that is ‘pasted’ on a neutral, gray background. The pipeline is exemplified in **Figure 2**, and consists of two key steps. First, given an image with associated object and part segmentation masks, we localize each desired part and place it in the center of a gray background. Second, for each localized image region, we pair it with a prompt describing the region. All resulting image-prompt pairs are then used to train a text-to-image diffusion model. We hypothesize that such a pipeline would enable a trained model to generate diverse renderings for each part type because existing datasets offer many diverse part segmentations, including of the same types observed across different objects (e.g., an eye can be found in a frog and dog).

Sticker Construction. Given a part segmentation dataset with associated images, we create our part stickers by using a templated approach. First, we obtain a binary segmentation for a desired part (e.g., the leg of a dog). We then multiply this binary segmentation with the associated image to obtain a RGB image where pixels outside of the desired region are black and pixels inside are the original content from the image. We then create a blank canvas of square aspect ratio where the background of the canvas is blank and ‘paste’ the masked content onto the center of the canvas. This design choice stems from the fact that saliency variations are undesired in our setting. Rather, we aim to generate a single, easy-to-extract salient part per image.

Prompt Generation. To provide textual conditioning during training, we construct prompts that reflect both the part and the object. Specifically, given an (object, part) pair, such as (dog, leg), we construct our input prompts with the template “a [PART] of a [OBJECT].” We choose this prompt over other choices (e.g., “[OBJECT] [PART].”) as language models have shown better hierarchical understanding with the former prompt [30]. However, we hypothesize at test time that users do not need to strictly abide by these prompts due to the model learning the object-part association, and we provide ablation study results that support this hypothesis in **Section 4**.

Training Approach. We fine-tune Stable Diffusion 1.5 with our training data in order to extend its powerful base capabilities. We achieve this with Low-Rank Adaptation (LoRA) fine-tuning, by attaching LoRA adapters [20, 40] to the attention layers of the U-Net. LoRA adds low-rank trainable parameters (i.e., matrices of rank r) into Q, K, V matrices and output projections (i.e., last linear layer of the attention blocks), which we unfreeze for gradient updates. The rest of the network remains frozen, reducing memory consumption and training time compared to full fine-tuning while preventing catastrophic forgetting. Importantly, this fine-tuning strategy allows us to adapt the network to our part-sticker generation task without discarding the foundational parameters. This approach maintains the model’s general “world knowledge” while enabling it to learn domain-specific behaviors pertinent to generating isolated parts on a gray background. During training, we optimize the standard diffusion loss presented in **Equation 3**.

4. Experiments

We now assess the performance of our proposed PartSticker framework against modern text-to-image generation baselines, both qualitatively and quantitatively.

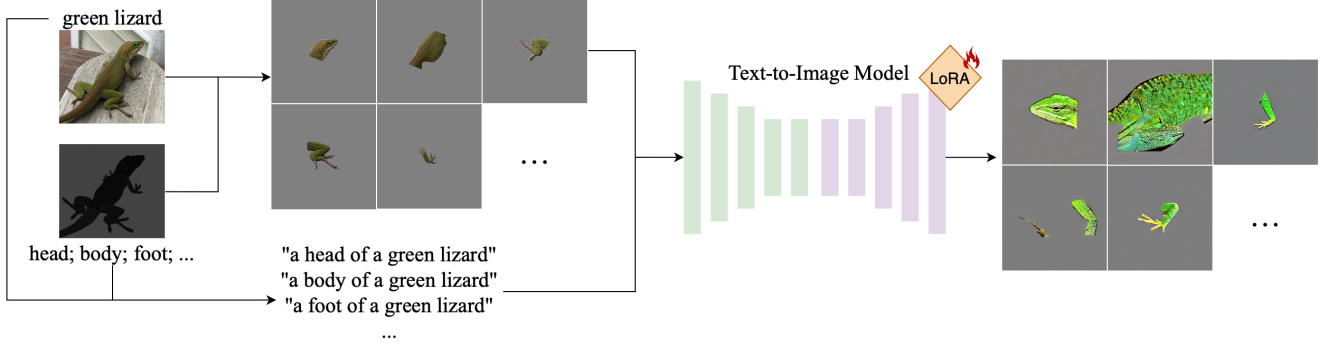


Figure 2. Overview of our proposed **PartStickers** framework. We train a base model on ‘part stickers’ (i.e., masked out parts of an object pasted on a neutral background) with text prompts describing the region, both of which are derived from existing part segmentation datasets. Text prompts are created by combining the part class labels with their object-level superclasses, leveraging the following template: “a [PART] of a [OBJECT]”. We leverage LoRA [20] to achieve parameter-efficient fine-tuning.

Model	Ours	SD1.5 [45]	SDXL [38]	InstanceDiffusion [61]	GLIGEN [26]
# Total Parameters	1.06B	1.06B	3.2B	1.2B	1.2B
Neutral Background Generation	✓	✗	✗	✗	✗
Part-only Generation	✓	✗	✗	✗	✗

Table 1. Comparison of our proposed method with other baselines in terms of total parameters, ability to generate regions of interest on a neutral background, and ability to generate solely the region of interest. Out of all methods, PartStickers is the only method capable of achieving the latter two capabilities, and achieves this while also relying on the fewest amount of parameters.

4.1. Experimental Design

We evaluate our framework by providing a model with a text prompt describing the part to be generated. We use the prompt introduced in **Section 3**. For each prompt, we consider 100 generated samples from each baseline method.

Baselines. We compare our proposed PartStickers framework against numerous popular text-to-image models. A summary of each model’s capacity and capabilities is provided in **Table 1**.

First, we leverage Stable Diffusion 1.5 [45] (SD 1.5), which also serves as our base model. SD 1.5 follows a latent diffusion architecture, that compresses images into lower-dimensional latent space via a variational autoencoder [22], then denoises these latent codes with a time-condition U-Net [46], and conditions on text using CLIP [42]. We use the variant of SD 1.5 trained on the large-scale LAION 5B dataset [50], which contains roughly five billion image-text pairs. We include SD 1.5 as a baseline for its proven competence in generating high-fidelity images from text alone while maintaining a relatively lightweight architecture.

We also evaluate Stable Diffusion XL [38] (SDXL), a more parameter-rich extension of SD 1.5 that introduces a refined architecture, multiple text encoders for better language alignment, and a refiner module for enhanced image detail. Trained on an expanded LAION-5B dataset with

higher-quality image-text pairs, SDXL offers near state-of-the-art text-to-image performance but demands significantly more computational resources. In our experiments, we compare using the full model, which includes a refinement module.

Besides text-to-image foundation models, we also compare our method with controllable generative models, which typically offer more precise manipulation of entities in a generated image. We leverage two top-performing models, GLIGEN [26] and InstanceDiffusion [61]. GLIGEN augments a U-Net backbone with learnable ‘condition’ embeddings and auxiliary modules that take in as input spatial annotations. We use the variant of GLIGEN trained on multiple image-text datasets with spatial annotations. Similarly, InstanceDiffusion integrates a module to handle four types of grounded conditions: bounding boxes, segmentation masks, scribbles, and points, relying on large-scale multimodal annotations. Both models accept text prompts plus one or more conditioning inputs. We use a centered bounding box in all experiments, to yield spatially constrained images.

Implementation Details. We implement our PartStickers model using the Diffusers [59] library, alongside PyTorch [35] for model training. To increase training speed, we leverage mixed-precision. We train our model for 10 epochs using a batch size of 16 text-part sticker pairs and

Methods	FID↓	SSIM↑
<i>PartStickers</i>	39.52	0.74
<i>Stable Diffusion 1.5</i> [45]	81.93	0.34
<i>InstanceDiffusion</i> [61]	85.66	0.35
<i>GLIGEN</i> [26]	81.62	0.35
<i>SDXL</i> [38]	55.32	0.49

Table 2. We evaluate using FID and SSIM. Our method outperforms prior work across both metrics, indicating its effectiveness on generating isolated part stickers.

keep the model with the best validation loss. For our LoRA fine-tuning, we use a rank of 16 using a learning rate of $1e^{-4}$ with the AdamW [28] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We resize all images to a spatial resolution of 512x512. All experiment were conducted on two NVIDIA A100 GPUs. At training time our model takes as input a text prompt and image, while at inference it takes as input a prompt alone.

Datasets. To generate our training data, we leverage the high-quality part segmentation dataset, PartImageNet [17], based off of the seminal ImageNet [11] paper. This datasets contains 20k images for training, 1k for validation, and 2k for testing. PartImageNet contains 11 object super-categories which can be further expanded into 158 specific categories (e.g., quadruped to deer) using their WordNet [29] names. After performing our sticker construction algorithm from Section 3.2, we end up with 94k samples for training, 5.6k for validation, and 11k for testing.

Through training and validation, we rely on WordNet-derived object names to construct precise text prompts that preserve rich semantic detail. However, anticipating that real-world users may describe an object region with more general terms (e.g., “car” rather than a specific make like “Subaru”), we have our test-time prompts adopt these more generalized descriptors.

Evaluation Metrics. We evaluate our generated part stickers using Fréchet Inception Distance [18] (FID) and Structural Similarity Index Measure [63] (SSIM). FID compares the feature distributions of real and generated images using features extracted from InceptionV3 [56]; values closer to 0 indicate greater similarity to the real distribution, whereas higher values represent signal poorer fidelity. SSIM assesses pixel-wise structural consistency between pairs of images on a scale from -1 to 1, where 1 denotes identical images and -1 denotes perfect anti-correlation (e.g., an inverted version of the image). By examining both FID (global distribution alignment) and SSIM (local structural fidelity), we capture complementary aspects of realism and structural accuracy in our generated outputs.

4.2. Results

We now analyze the results of experiments, both quantitatively using traditional image generation evaluation metrics, and qualitatively, using representative generated samples from our baseline methods. We provide quantitative results in Table 2, and qualitative results in Figure 3.

Overall Performance. We report the quantitative performance for our method and all baselines in Table 2. For both metrics, FID and SSM, PartStickers consistently outperforms the baseline methods by a considerable margin. For example, we achieve a 106% relative percentage point increase in FID over the next best method, GLIGEN, suggesting that our approach more correctly produces images that have a higher degree of realism, as measured by comparing the feature distributions of InceptionV3. Similarly, when observing the next best SSIM scores (i.e., InstanceDiffusion and GLIGEN), we observe a 111% relative percentage point increase in performance. This discrepancy in performance highlights a crucial limitation of existing approaches: they are unable to match patterns in generated images of localized parts, as evidenced by both the low SSIM scores as well as the qualitative examples shown in Figure 3.

When observing the quantitative performance of baseline methods, we observe little variance in both SSIM and FID scores, with the exception of InstanceDiffusion, which performs nearly four percentage points worse than the remaining baselines. This poor performance suggests that, while these methods may be able to generate out of distribution objects such as the dinosaur head in Figure 3, they are unable to generalize outside of their training data to novel structures such as solely generating a part. The one exception to this trend is SDXL, which attains significantly better FID and SSIM scores than the other baselines, although it still does not surpass our results. Notably, SDXL uses nearly three times as many parameters, granting it substantially more expressive power.

Analysis with Respect to Specific Parts. We analyze the effectiveness of our method compared to baseline approaches for generating specific object parts, such as “the tire of a car” or “the wing of a bird,” as shown in Figure 3. Across all evaluations, our proposed framework demonstrates consistent results, whereas baseline methods yield mixed outcomes. PartStickers consistently generates only the intended part for various object classes. For example, for the prompt “the sail of a boat,” our method reliably generates only the sail without including the boat body. Similarly, with the prompt “a foot of a lizard”, our method correctly does not include the lizard’s body in the generated content. In contrast, baseline methods frequently generate the entire boat and lizard. Similarly, when prompted to gen-

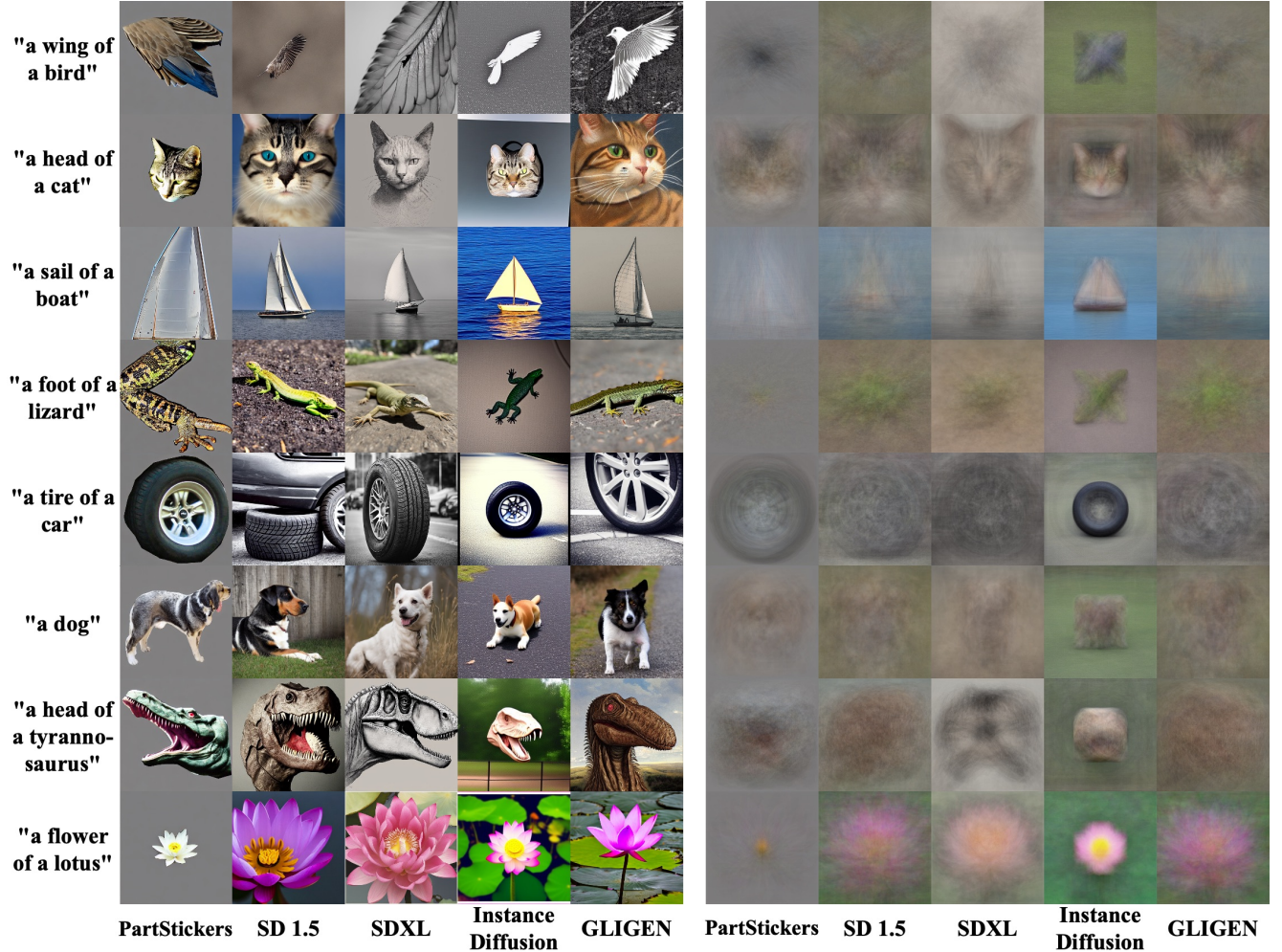


Figure 3. Qualitative results showing examples of generated images given text prompts (left) and the average image of 100 generated samples from a given method (left). Overall, we observe that PartStickers is the only method capable of consistently generating only the requested part on a neutral background with a high degree of realism. The bottom three rows represent out-of-distribution scenarios for PartStickers: generation of an object and two out-of-distribution parts. (SD stands for Stable Diffusion).

erate “the tire of a car,” PartStickers consistently isolates the tire, whereas other methods often include extra portions of a car, such as its fender in one instance. While Stable Diffusion 1.5 and Stable Diffusion XL can occasionally generate only the target part, such as for the prompt “the wing of a bird,” such results are inconsistent with many generations containing the full bird. This inconsistency across baseline methods highlights a significant drawback: users cannot reliably predict whether a method will generate solely the desired part. Such unpredictability may necessitate additional effort to isolate the desired parts from the generated images, which we hypothesize is cumbersome for users. In contrast, our method provides consistent and predictable results, reducing the potential need for extra intervention.

Analysis with Respect to Realism. We assess realism in terms of physical accuracy and stylistic consistency between photorealistic and artistic outputs. PartStickers reliably produces physically plausible results. For example, when prompted with “the foot of a lizard,” it generates anatomically correct outputs without extra limbs or unrealistic artifacts. In contrast, baseline methods like Stable Diffusion 1.5 and InstanceDiffusion occasionally produce errors such as duplicated limbs or misplaced features (e.g., a foot replacing a tail). GLIGEN and SDXL similarly exhibit issues like distorted parts or incorrect spatial placement, such as a wing rendered on a bird’s head.

In terms of photorealism, PartStickers consistently outputs realistic imagery, likely due to fine-tuning on real-world images. Baseline models, however, often alternate between photorealistic and stylized results, includ-

ing sketches or illustrations. While stylistic diversity may be desirable in some contexts, we argue that photorealism should be the default unless otherwise specified, as it better supports our target application.

One limitation of PartStickers is the presence of harsh edges in some outputs, particularly for round parts like tires, which may appear polygonal. This likely stems from segmentation ground truths being imperfect at pixel boundaries. Still, we believe minor edge artifacts are a reasonable trade-off, especially in early-stage prototyping where speed outweighs polish. Future improvements could focus on enhancing segmentation quality or integrating boundary refinement directly into the generation process.

Analysis with Respect to Saliency. We assess part saliency by averaging 100 generated images per prompt, as shown in **Figure 3 (right)**. Results reveal how consistently each method renders target parts across samples.

Our method maintains a clear focus on the specified part while capturing real-world variability. For instance, prompts like “a head of a cat” and “a tire of a car” yield blurry yet recognizable shapes, indicating strong saliency. In contrast, “a wing of a bird” shows broader variation in pose and angle, which can support design exploration. In the out-of-distribution case, “a flower of a lotus,” the stamen appears centered, but the petals are faint or missing. We hypothesize this results from variations in petal shape, size, and viewpoint, which become indistinct when averaged.

Baseline models generally show greater positional variation. For example, Stable Diffusion 1.5, SDXL, and GLIGEN lead to diffuse average images, such as dominated by a tire color for “a tire of a car” that gets generated at varying positions. The one exception is InstanceDiffusion, which is expected because InstanceDiffusion relies on a user-provided bounding box to determine where the generated part should appear, thus consistently centering the target in the image. Interestingly, despite GLIGEN receiving similar spatial cues (i.e., a bounding box), its content exhibits more positional variation than InstanceDiffusion. While such variability may benefit tasks like recognition or segmentation, we argue it is less suitable for rapid prototyping, where designers may prefer predictable placement.

Although one could envision a future hybrid approach that combines bounding-box guidance from InstanceDiffusion with our PartStickers framework to offer explicit location control, it would require extra user input (i.e., the bounding box). In comparison, PartStickers centers each generated part by default, balancing user needs for predictable placement and minimal configuration overhead.

Analysis with Respect to Background. We next analyze the backgrounds associated with the generated content. PartStickers consistently places each generated region of interest onto a neutral, gray background across all prompts.

Although slight hue variations occur between samples and prompts, PartStickers remains the only method that reliably maintains neutral backgrounds. Baseline methods, in contrast, generally produce non-neutral backgrounds. Stable Diffusion XL and GLIGEN occasionally produce relatively neutral backgrounds, notably when generating “a head of a cat,” but this consistency does not extend to other prompts, creating unpredictable results for end users. InstanceDiffusion also occasionally generates simpler backgrounds, as seen with in the lizard example, however they often exhibit color gradients or subtle patterns that could complicate automated masking tasks for ‘sticker’ removal.

Analysis with Respect to Object Generation. In the third row from the bottom in **Figure 3**, we show a generated example of an entire object (i.e., a dog) on the left and its ‘average image’ on the right. Notably, even though our model was not fine-tuned on full-object data, it retains the base model’s capacity to generate complete objects. Moreover, it exhibits a new capability by placing the dog at the center of a neutral background, suggesting our approach is suitable for rapid prototyping beyond part-level requirements. The resulting output can be used *as is* with minimal user intervention, unlike the other methods, which depict the dog against a plausible but non-neutral environment.

Observing the average image confirms that most of the generated dog pixels appear near the center, likely a consequence of the center-pasting strategy used in our part-sticker training pipeline (**Section 3**). In comparison, the baseline methods (other than InstanceDiffusion, which conditions on a bounding box in the center) exhibit broader variation in the dog’s position, leaving the user to locate or isolate the object. As expected, all baselines also generate full objects effectively, aligning with their original tasks.

Analysis with Respect to Out-of-distribution Performance. In the bottom two rows of **Figure 3**, we present out-of-distribution examples for all baseline methods. These prompts are considered out-of-distribution for PartStickers because neither the categories (i.e., tyrannosaurus and lotus) nor the text prompts were seen during fine-tuning. They are also out-of-distribution for the baseline methods because their prompts follow the form “a [PART] of [OBJECT].” Notably, when generating “a head of a tyrannosaurus”, most methods depict the creature accurately, with only GLIGEN adding extraneous elements such as the neck. When examining background contents, PartStickers remains the sole method to produce a fully neutral background. While Stable Diffusion 1.5 and SDXL exhibit gray backdrops, they incorporate vignette effects that cannot be easily removed without access to the RAW image, which is unavailable with generated images since they are not captured from the sensor of a camera. For the lotus example, PartStickers successfully generates the flower alone,

Rank	FID↓	SSIM↑
4	42.93	0.69
16	39.52	0.74
32	41.48	0.71

Table 3. Ablation study on the effect of LoRA rank and data augmentation strategy. A rank of 16 achieves the best performance.

whereas most other methods also produce its stem. Despite these categories being unseen during fine-tuning, PartStickers maintains strong image quality while preserving a consistent, neutral background.

4.3. Model Design Analysis

In this section we ablate key design choices for our PartStickers framework to establish their importance.

Effect of LoRA Rank. We analyze the impact on performance of the LoRA rank for fine-tuning our base model. Results are shown in **Table 3**.

Intuitively, one might expect higher ranks to yield better performance, since larger LoRA matrices can provide increased capacity. However, our results indicate a more nuanced relationship. When using a rank of 4, the model appears to lack sufficient representational power, leading to the worst performance. Conversely, at rank 32, the model exhibits the second poorest results, suggesting it may be *underfitting* due to excessive capacity relative to our dataset. The best outcomes arise with an intermediate rank of 16, presumably because this configuration strikes an effective balance between representational capacity and demands for our novel task. Notably, all the aforementioned configurations still outperform all baselines by a large margin.

Effect of Amount of Training Data. We analyze the impact of training data size, comparing performance when fine-tuning with 50% of the available data versus the full dataset. Results are shown in **Table 4**.

As expected, reducing the training data results in a performance drop, consistent with the general trend that foundation models benefit from larger datasets. However, a notable observation is that even with only half the data, our model still outperforms all baselines. This finding suggests that while additional data improves performance, modern models require only a fraction of the samples to effectively learn our target task.

Effect of Cropping Strategy. We evaluate two cropping strategies for creating part stickers. In the default *center-paste* strategy (see **Section 3**), the extracted part is placed at the center of a neutral background. The *non-center-paste* strategy instead preserves the part’s original position. Results are shown in **Table 5**.

Amount of Data	FID↓	SSIM↑
50%	51.67	0.63
100%	39.52	0.74

Table 4. Ablation study on the effect of the amount of available training data for fine-tuning. We observe a decrease in performance when using half the available data, but an increase compared to the baseline methods.

Crop Strategy	FID↓	SSIM↑
Non-center Paste	41.52	0.72
Center Paste	39.52	0.74

Table 5. Ablation study on the effect of center crop.

As shown, center-paste yields better quantitative performance. This suggests the consistent positioning aids model learning. In contrast, non-center-paste stickers retain contextual cues like real-world scale and framing. For instance, a cat’s head near the center often indicates a close-up, while one near the edge often suggests a wide-angle view. Our findings offer evidence that such context is less helpful when generating a part for our target prototyping task.

5. Conclusion

We introduced PartStickers, a generative framework that directly produces isolated object parts with neutral backgrounds from textual prompts. By leveraging LoRA-based fine-tuning on part-segmentation data, PartStickers outperforms state-of-the-art methods both quantitatively and qualitatively, which we argue could considerably reduce human effort in rapid prototyping workflows. Future work could explore method refinements, such as achieving smoother boundaries, as well as investigate compositional strategies for creating entirely new objects from generated parts.

Our method accelerates iterative design but, like other generative AI tools, carries ethical risks such as misuse for disinformation or low-quality content [21]. Unlike image-level generation, which can instantly create misleading media like Deep Fakes [16, 32], our framework requires manual assembly, reducing the risk of mass misinformation. Additionally, it focuses on non-human animals and rigid objects [17], limiting its ability to generate deceptive human imagery while facilitating in rapid prototyping.

Acknowledgments. Josh Myers-Dean is supported by a NSF GRFP fellowship (#1917573). This work utilized the Blanca condo computing resource at the University of Colorado Boulder. Blanca is jointly funded by computing users and the University of Colorado Boulder.

References

- [1] Yağız Aksoy, Tunç Ozan Aydın, Marc Pollefeys, and Aljoša Smolić. Interactive high-quality green-screen keying via color unmixing. *ACM Transactions on Graphics (TOG)*, 36(4):1, 2016. 2
- [2] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. 2, 3
- [3] Volker Bilgram and Felix Laarmann. Accelerating innovation with generative ai: Ai-augmented digital prototyping and innovation methods. *IEEE Engineering Management Review*, 51(2):18–25, 2023. 1
- [4] Ali Borji. Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2. *arXiv preprint arXiv:2210.00586*, 2022. 1
- [5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18392–18402, 2023. 2
- [6] Bradley Camburn, Vimal Viswanathan, Julie Linsey, David Anderson, Daniel Jensen, Richard Crawford, Kevin Otto, and Kristin Wood. Design prototyping methods: state of the art in strategies, techniques, and guidelines. *Design Science*, 3:e13, 2017. 1
- [7] Anpei Chen, Ruiyang Liu, Ling Xie, Zhang Chen, Hao Su, and Jingyi Yu. Sofgan: A portrait image generator with dynamic styling. *ACM Transactions on Graphics (TOG)*, 41(1):1–26, 2022. 1
- [8] Chee Kai Chua, Kah Fai Leong, and Chu Sing Lim. *Rapid prototyping: principles and applications*. World scientific, 2010. 1
- [9] Guillaume Couairon, Marlene Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2174–2183, 2023. 2
- [10] Giannis Daras, Augustus Odena, Han Zhang, and Alexandros G Dimakis. Your local gan: Designing two dimensional local attention mechanisms for generative models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14531–14539, 2020. 2
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5
- [12] Kristen M Edwards, Brandon Man, and Faez Ahmed. Sketch2prototype: rapid conceptual design exploration and prototyping with generative ai. *Proceedings of the Design Society*, 4:1989–1998, 2024. 1
- [13] Hajar Emami, Majid Moradi Aliabadi, Ming Dong, and Ratna Babu Chinnam. Spa-gan: Spatial attention gan for image-to-image translation. *IEEE Transactions on Multimedia*, 23:391–401, 2020. 2
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2
- [15] Jing Gu, Nanxuan Zhao, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, HyunJoon Jung, Yilin Wang, and Xin Eric Wang. Swapanything: Enabling arbitrary object swapping in personalized image editing. In *European Conference on Computer Vision*, pages 402–418. Springer, 2024. 2
- [16] Jeffrey T Hancock and Jeremy N Bailenson. The social impact of deepfakes, 2021. 8
- [17] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts, 2022. 5, 8
- [18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017. 5
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020. 2, 3
- [20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3, 4
- [21] Laura Illia, Elanor Colleoni, and Stelios Zyglidopoulos. Ethical implications of text generation in the age of artificial intelligence. *Business Ethics, the Environment & Responsibility*, 32(1):201–210, 2023. 8
- [22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 4
- [23] Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Text-to-image generation grounded by fine-grained user attention. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 237–246, 2021. 1
- [24] Sehyun Lee, Mijung Kim, Yeongnam Chae, and Björn Stenger. Linearly controllable gan: Unsupervised feature categorization and decomposition for image generation and manipulation. In *European Conference on Computer Vision*, pages 229–245. Springer, 2024. 2
- [25] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P Xing. Semantic-aware grad-gan for virtual-to-real urban scene adaption. *arXiv preprint arXiv:1801.01726*, 2018. 2
- [26] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation, 2023. 4, 5
- [27] Wentong Liao, Kai Hu, Michael Ying Yang, and Bodo Rosenhahn. Text to image generation with semantic-spatial aware gan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18187–18196, 2022. 2
- [28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5
- [29] George A. Miller. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. 5

- [30] Josh Myers-Dean, Jarek Reynolds, Brian Price, Yifei Fan, and Danna Gurari. Spin: Hierarchical segmentation with subpart granularity in natural images. In *European Conference on Computer Vision*, pages 275–292. Springer, 2024. [3](#)
- [31] Nithin Gopalakrishnan Nair, Jeya Maria Jose Valanarasu, and Vishal M Patel. Maxfusion: Plug&play multi-modal generation in text-to-image diffusion models. In *European Conference on Computer Vision*, pages 93–110. Springer, 2024. [2](#)
- [32] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. [8](#)
- [33] 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. *arXiv preprint arXiv:2112.10741*, 2021. [2](#)
- [34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. [2](#)
- [35] 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. [4](#)
- [36] Dunlu Peng, Wuchen Yang, Cong Liu, and Shuairui Lü. Sam-gan: Self-attention supporting multi-stage generative adversarial networks for text-to-image synthesis. *Neural Networks*, 138:57–67, 2021. [2](#)
- [37] Zongming Peng, Tie Liu, Yangqianqian Chen, Yue Yang, Keren Fu, Fan Pan, and Qijun Zhao. Identity-preserving animal image generation for animal individual identification. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 349–362. Springer, 2024. [1](#)
- [38] 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, 2023. [4](#), [5](#)
- [39] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984. [2](#)
- [40] Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning. *arXiv preprint arXiv:2311.11077*, 2023. [3](#)
- [41] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6305–6315, 2024. [2](#)
- [42] 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, 2021. [4](#)
- [43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. [2](#)
- [44] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016. [2](#)
- [45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. [3](#), [4](#), [5](#)
- [46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. [3](#), [4](#)
- [47] Shulan Ruan, Yong Zhang, Kun Zhang, Yanbo Fan, Fan Tang, Qi Liu, and Enhong Chen. Dae-gan: Dynamic aspect-aware gan for text-to-image synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13960–13969, 2021. [2](#)
- [48] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. [1](#)
- [49] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. [2](#)
- [50] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. [4](#)
- [51] Artur Shagidanov, Hayk Poghosyan, Xinyu Gong, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Grounded-instruct-pix2pix: Improving instruction based image editing with automatic target grounding. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6585–6589. IEEE, 2024. [2](#)

- [52] Yichun Shi, Xiao Yang, Yangyue Wan, and Xiaohui Shen. Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11264, 2022. 2
- [53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2
- [54] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. Protoai: Model-informed prototyping for ai-powered interfaces. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, pages 48–58, 2021. 1
- [55] Hari Subramonyam, Divy Thakkar, Jürgen Dieber, and Anoop Sinha. Content-centric prototyping of generative ai applications: Emerging approaches and challenges in collaborative software teams. *arXiv preprint arXiv:2402.17721*, 2024. 1
- [56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. 5
- [57] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. Df-gan: A simple and effective baseline for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16515–16525, 2022. 2
- [58] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016. 1
- [59] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. <https://github.com/huggingface/diffusers>, 2022. 4
- [60] Qian Wang, Biao Zhang, Michael Birsak, and Peter Wonka. Instructedit: Improving automatic masks for diffusion-based image editing with user instructions. *arXiv preprint arXiv:2305.18047*, 2023. 2
- [61] Xudong Wang, Trevor Darrell, Sai Saketh Rambhatla, Rohit Girdhar, and Ishan Misra. Instancediffusion: Instance-level control for image generation, 2024. 4, 5
- [62] Yi Wang, Lu Qi, Ying-Cong Chen, Xiangyu Zhang, and Jiaya Jia. Image synthesis via semantic composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13749–13758, 2021. 2
- [63] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5
- [64] James Wilson and Daniel Rosenberg. Rapid prototyping for user interface design. In *Handbook of human-computer interaction*, pages 859–875. Elsevier, 1988. 1
- [65] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018. 2
- [66] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, and John Collomosse. Text-to-image generation via implicit visual guidance and hypernetwork. *arXiv preprint arXiv:2208.08493*, 2022. 2
- [67] Yu Zeng, Zhe Lin, Jianming Zhang, Qing Liu, John Collomosse, Jason Kuen, and Vishal M Patel. Scenecomposer: Any-level semantic image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22468–22478, 2023. 2
- [68] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2
- [69] Yinan Zhao, Brian Price, Scott Cohen, and Danna Gurari. Unconstrained foreground object search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2030–2039, 2019. 2