Does Feasibility Matter?
Understanding the Impact of Feasibility on Synthetic Training Data

Yiwen Liu[1] Jessica Bader[1,3] Jae Myung Kim[2,3]

[1]Technical University of Munich [2]University of Tübingen [3]Helmholtz Munich

Figure 1. We propose VariReal, a pipeline for minimal-change editing of real images, enabling isolation of target attributes in three categories: background, color, and texture. We compare images generated by VariReal to those produced by prior text-guided editing methods [6, 34], examining both feasible and infeasible attributes. The editing prompts are provided below each generated image.

Abstract

With the development of photorealistic diffusion models, models trained in part or fully on synthetic data achieve progressively better results. However, diffusion models still routinely generate images that would not exist in reality, such as a dog floating above the ground or with unrealistic texture artifacts. We define the concept of ***feasibility*** *as whether attributes in a synthetic image could realistically exist in the real-world domain; synthetic images containing attributes that violate this criterion are considered* ***infeasible****. Intuitively, infeasible images are typically considered out-of-distribution; thus, training on such images is expected to hinder a model's ability to generalize to real-world data, and they should therefore be excluded from the training set whenever possible. However, does feasibility really matter? In this paper, we investigate whether enforcing feasibility is necessary when generating synthetic training data for CLIP-based classifiers, focusing on three target attributes: background, color, and texture. We introduce VariReal, a pipeline that minimally edits a given source image to include feasible or infeasible attributes given by the textual prompt generated by a large language model. Our experiments show that feasibility minimally affects LoRA-fine-tuned CLIP performance, with mostly less than 0.3 % difference in top-1 accuracy across three fine-grained datasets. Also, the attribute matters on whether the feasible/infeasible images adversarially influence the classification performance. Finally, mixing feasible and infeasible images in training datasets does not significantly impact performance compared to using purely feasible or infeasible datasets. Code is available at https://github.com/Yiveen/SyntheticDataFeasibility.*

1. Introduction

In recent years, large-scale pre-trained models [7, 26, 31, 37, 51, 60] have significantly surpassed traditional learning approaches in various tasks. However, as the scale of training data grows, access to high-quality data has become increasingly limited [64], posing challenges to further improving these large models' capabilities. With the popularity of generative models [21, 30] like Stable Diffusion [51], researchers are increasingly leveraging these models to generate high-fidelity synthetic data that closely resembles real-

world data, offering a solution to data scarcity [15, 20].

Prior studies have explored synthetic data generation under a limited few-shot real image setting [8, 14, 22, 24, 28, 54, 55, 59]. These works aim to create synthetic data that approximates the real-world data distribution while avoiding overfitting to the limited available examples. Some studies [24, 28] suggest that synthetic data can offer benefits beyond those of real data. However, the inherent randomness in the diffusion-based image generation process [26, 51] can introduce domain shifts [24] or implausible scenarios, such as "a dog floating in the sky" [54], which fail to reflect realistic patterns. Such data could intuitively be perceived as out-of-distribution (OOD), potentially becoming counterproductive for downstream tasks.

Interestingly, previous studies [4, 9, 19] suggest that OOD data can positively impact downstream tasks when mixed with real data in certain proportions. A typical example is data augmentation [19], where some data augment methods introduce OOD data relative to the original distribution yet still provide benefits. While these advantages generally diminish as divergence from the original distribution increases [9], these findings demonstrate OOD data is not always harmful. Conversely, incorporating feasible content, which is considered in-distribution, is naturally beneficial. For instance, Dunlap et al. [14] propose augmenting training data by synthesizing data with diverse feasible backgrounds and show performance gain. This raises a question: *does the feasibility matter for synthetic training data?*

In this paper, we study the impact of the feasibility on synthesized data when using them as training data for the classification task. We define feasibility as whether class-specific attributes could realistically occur in the real world. Attributes that meet this criterion are considered feasible while others are infeasible. For instance, given a Yorkshire terrier in Figure 1, it is likely to find it at the lake shore, while not at the oil rig platform. Therefore, we assume an image of Yorkshire terrier at the lake shore background as a feasible image, while an image of it at the oil rig platform as an infeasible image.

To generate feasible and infeasible images and study their impact of a downstream classification task, we propose VariReal, an editing pipeline with minimal change of attributes given a real image. We first generate a list of feasible and infeasible attribute names for each class by using GPT-4 [1], with generated attributes further being validated through a user study. We then use a proposed image-editing pipeline based on Stable Diffusion [51] that generates feasible (or infeasible) images given a source real image and a prompt with a feasible (or infeasible) attribute name. We then assess the impact of the feasibility of images to downstream tasks by fine-tuning CLIP-based classifiers under two conditions: synthetic-only training and mixed real-synthetic training.

Our study of feasibility for a downstream task in three different attributes (background, color, texture) on three fine-grained datasets reveals the following insights. First, we show that changing the background regardless of feasibility brings performance gain, which loosens a restriction considered in ALIA [14] where it only uses a feasible background scenario. Second, foreground modifications, like color or texture attributes, often challenge the classifier's learning process especially when the training datasets are infeasible inputs.

In summary, our contributions are as follows:

- We propose VariReal, an automated generation pipeline for producing minimal-change synthetic data by altering only one attribute from real images at a time. This approach can be applied out-of-the-box to any object-centric classification dataset without additional fine-tuning.
- We define and generate feasible and infeasible dataset comparison pairs based on real images, covering three controlled attributes.
- To explore feasible and infeasible data roles, we fine-tune CLIP with LoRA [27]. Analyzing classification scores, we offer new insights into the impact of feasibility and the strategic use of synthetic data for enhancing downstream classification performance.

2. Related Work

Effect of out-of-distribution data. OOD data, defined relative to in-distribution data, introduces a distribution shift between train and test data. OOD data is generally categorized into semantic and covariance shifts [57]; here, we focus on covariance shifts. The impact of OOD data is commonly evaluated using classification tasks [4, 9, 19, 19]. Early works [4, 19] attributed OOD data's benefits to feature invariance and the stochasticity it adds in gradient descent, helping avoid local minima and improving optimization. However, this conclusion was drawn only using simple OOD data types like rotation.

Silva *et al*. [9] and Geiping *et al*. [19] show that, for small domain shifts, adding OOD data reduces generalization error on the original test set and exhibits non-monotonic behavior. While most research has relied on basic models (e.g., ResNet [23]) and datasets (e.g., MNIST [11]), our work seeks to produce OOD data study to more complex scenarios with diffusion models, utilizing advanced classification architectures to deepen the understanding of OOD effects.

Learning with synthetic data. Several studies [8, 24, 28, 54, 59] focus on generating synthetic data that approximates real-world distributions. These approaches aim to create a dataset larger than the few-shot samples. Generated data supports various tasks, including object recogni-

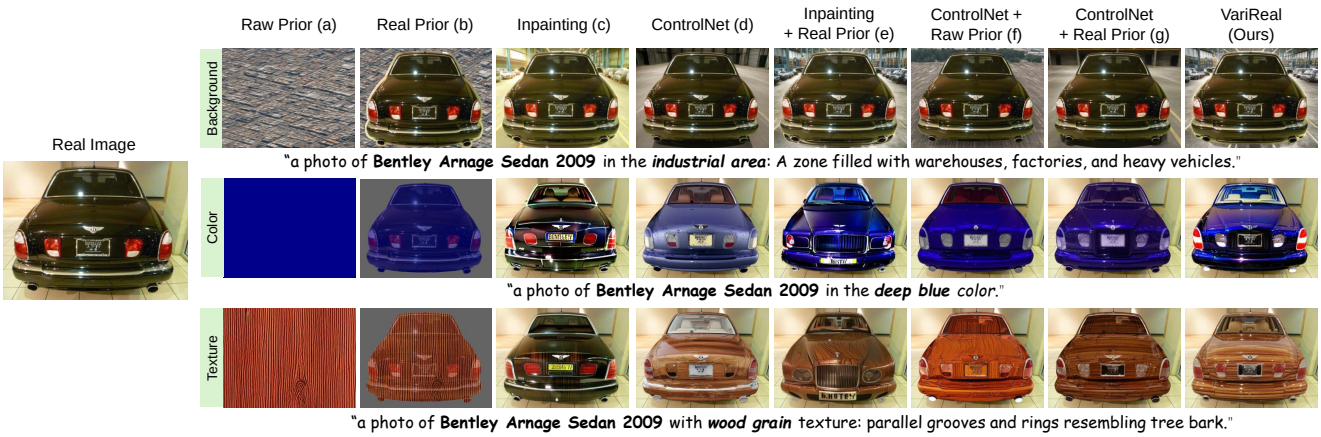

Figure 2. We compare images generated by various candidate methods: Inpainting model [40] alone, ControlNet [61] alone, Inpainting model with Real Prior, ControlNet with Raw Prior or Real Prior, and our final results for three attribute modifications. The first two columns illustrate the priors used (Raw Prior and Real Prior), and generation prompts used are listed beneath each image.

tion [8, 14, 28, 54], object detection [17], and semantic segmentation [55]. Its effectiveness is demonstrated by training CLIP [47] models exclusively on synthetic data or in combination with real data [16, 24, 28]. As a result, we focus specifically on object classification using CLIP model.

Automatic approach for minimal change generation. Unlike synthetic data generation methods that focus on creating novel and diverse in-distribution images [28], minimal change generation aims only to modify specific areas or attributes of existing real images. Generative models, particularly diffusion-based approaches [44, 49, 51, 53], facilitate efficient image editing without requiring manual annotation [24] or physical graphics engines [3, 48]. In particular, text-to-image stable diffusion methods are popular for minimal-change editing due to their high fidelity generation. Beyond text guidance, these models also support diverse conditioning inputs, such as reference images through IP-Adaptor [58] and Canny edge maps through ControlNet [61].

These methods fall into two main categories: fine-tuning needed approaches [6, 18, 63], and non-fine-tuning needed approaches such as attention- or mask-based diffusion methods [25, 34]. Fine-tuned methods, such as InstructPix2Pix [6], require model retraining to achieve desired edits across new input domains. In contrast, attention- and mask-based diffusion models can target specific modifications without further fine-tuning. Attention-based methods, like FPE [34] and P2P [25], substitute certain self- or cross-attention layers in the U-Net [52]'s denoising process, leveraging the interpretability of attention maps. However, these methods may not perform well in all scenarios, particularly with real images [34]. Mask-based diffusion models, such as inpainting methods [29, 39, 43, 56], offer strong generalization and method versatility by enabling controlled edits within specified areas while preserving unmasked regions. However, when modifying objects itself, these models may occasionally alter subtle shape details. Methods like ControlNet [61] can help maintain an object's original structure during edits.

The most closely related work is VisMin [2], which generates minimal-change data to improve vision-language model comprehension. However, VisMin does not support controlled edits across our targeted three attributes. In contrast, we introduce an automatic, off-the-shelf approach enabling minimal, photorealistic edits for arbitrary combinations of real images and textual instructions.

3. Method

3.1. Preliminaries

Task formulation. Our goal is to analyze the impact of feasible and infeasible synthetic data (I_{Syn}), with feasibility defined per individual class c_i, where $i \in 1, \ldots, C$. Our VariReal method generates minimal-change I_{Syn} pairs from a shared real-image base (I_{Real}) using distinct textual prompts. Our approach isolates feasibility across three targeted attribute categories—background, color, and texture—while minimally altering other image content (e.g., the same dog depicted with different colors). The textual guidances are class-specific, LLM-generated prompts categorized as feasible (P_f) and infeasible (P_{if}). Each real image (I_{Real}) is combined with all prompts from both categories, ensuring every real image is repeated equally, $|P_f| = |P_{if}|$. By varying the number of prompts ($|P_f| \geq 1$), we assess the impact of additional synthetic augmentations. Note that the texture attribute inherently includes color characteristics. Finally, we LoRA fine-tune CLIP models on in-distribution and OOD synthetic datasets to compare how each data type influences downstream classification performance.

Fine-tuning with low-rank adaptation. The Low-Rank Adaptation [27] introduces low-rank decomposition into the

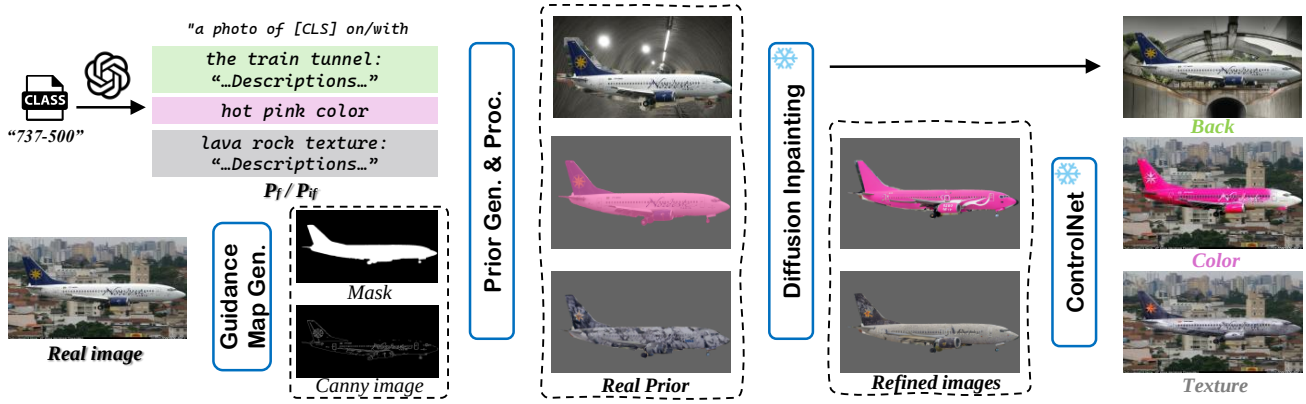

Figure 3. **An overview of VariReal pipeline.** Minimal-change steps for background, color, and texture are highlighted in green, pink, and grey, respectively. Real images are processed to generate guidance maps (e.g., masks, Canny edges) for Inpainting and ControlNet. GPT-4 generates feasible and infeasible prompts (P_f and P_{if}), which guide color retrieval or prior image generation via Stable Diffusion. These Real Priors, combined with masks and prompts, are input to the inpainting model. For color and texture, ControlNet with Canny conditioning ensures precise foreground shapes. A final refinement step produces the optimal output for each setting.

pre-trained weight matrix to reduce the number of learnable parameters. The final weights after fine-tuning could be expressed by $h = W_0x + BAx$, where W_0 represents the pre-trained weights. The decomposed weights $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, with LoRA rank $r \ll \min(d, k)$.

Latent diffusion models. Latent Stable Diffusion [51] encodes an image into a latent space using an encoder, defined as $z_0 = E(x_0)$, and learns a conditional distribution $p(z|c)$ by predicting the Gaussian noise added to the latent vector. The objective function can be expressed as:

$$\min_{\theta} \mathbb{E}_{(x,c)\sim\mathcal{D},\, \epsilon\sim\mathcal{N}(0,1),\, t}\left[\|\epsilon - \epsilon_\theta(z_t, c, t)\|_2^2\right] \quad (1)$$

where z_t is the noisy latent representation, c is corresponding conditions and ϵ represents the Gaussian noise added at each time step t. For the inference process, a randomly noised vector is sampled and denoised over total T steps to obtain the final latent representation z_0, which is then decoded back into pixel space using the decoder $D(z_0)$ of the VAE [30].

A naive solution. Naive solutions could employ text-guided Inpainting models [39] (e.g., SDXL Inpainting) or Canny-edge-based ControlNet [61] models (e.g., SDXL ControlNet), using a base prompt P_{base} = "a photo of a [CLS]". Inpainting methods generate more natural images but are heavily influenced by the original attributes, limiting their effectiveness for substantial changes, as illustrated by the persistent dark hue when changing a black car's color of Figure 2 (c).

Conversely, ControlNet preserves object structure independently from original attributes in Figure 2 (d) but often produces less natural edits for color and texture and can cause objects to appear floating when modifying backgrounds, reducing realism.

Motivation. To overcome the limitations of existing methods, we design a pipeline which overcomes the individual weaknesses of out-of-the-box methods by combining the individual strengths, i.e. combining Inpainting's realism with ControlNet's preciseness.

3.2. VariReal: Generating minimal-change data

We present a zero-shot pipeline for minimal-change image generation. Sec. 3.2.1 details prompt generation P_f and P_{if}, followed by our prior-based generation process in Sec. 3.2.2, including key steps like guidance maps and final processing. We also compare candidate models to determine the optimal modification strategy. Lastly, Sec. 3.2.3 covers MLLM-based filtering.

3.2.1. Guidance prompt

P_f and P_{if} are as text prompts for Stable Diffusion model to guide desired content. To generate as many accurate P_f and P_{if} per fine-grained class as possible, we utilize ChatGPT-4 [1] with In-Context Learning [12], providing the model with positive examples $Example+$ and negative examples $Example-$ to help avoid errors and repetitive content. To improve the fine-grained detail and realism of the generated backgrounds or textures, we instruct GPT to append a brief explanatory description when generating prompts, providing more detailed guidance for image generation.

Although large language models possess broad knowledge across various domains, ChatGPT still regularly designates attributes as 'feasible' for a target object that do not exist in the real world, particularly for fine-grained classes for which it has limited knowledge. For example, fine-grained airplane class "737-500" normally do not have color in purple. To address this issue, we design additional prompts to instruct the model to perform preliminary checks and filtering on its outputs. Manual verification ensures that feasible prompts align with the training domain. Using the same base prompt and ChatGPT-generated results, we form

our final prompts shown in the Figure 1 and Figure 2. Details of the generation process and filtered ratios are provided in the Supplementary Sec. D.1.

3.2.2. Prior-guided minimal change generation

Guidance map generation. The guided mask and canny images are for inpainting model and ControlNet respectively. We use Grounding DINO [36] to generate bounding boxes bbox_i, which are then fed into the SAM2 [50] model to produce masks m_i for each category c_i. For samples without detectable bounding boxes, we use the RMBG1.4 [5] foreground segmentation model as a fallback to ensure each sample has a mask.

In our method, we use the Canny image ControlNet [61] model. For all settings, the Canny image is created by extracting the foreground Foreground_i from mask_i.

Prior generation and process. We use prompts P_f or P_{if} with Stable Diffusion to generate "Background" and "Texture" Priors, and predefined RGB values from a Color Bank for the "Color Prior." These initial outputs are termed *Raw Priors*.

To integrate these priors with real images, we merge the original object's region with the Background Prior, applying mask dilation to preserve spatial context and realism (e.g., ensuring pets remain grounded). We ablate this operation effect in the Supplementary Sec. H. For color and texture edits, the generated prior is overlaid via an alpha channel to retain the original shape and details of the subject. These refined results are referred to as *Real Priors*. The Figure 2 (a-b) illustrate these Priors. ControlNet leverages both Raw and Real Priors as conditions via IP-Adaptor [58], whereas Inpainting exclusively employs Real Prior to retain unchanged original information.

Final process. Before outputting the final images, the last step involves copying invariant regions from the original image and pasting them onto the generated image, ensuring minimal alterations.

Minimal change for background. Figure 2 (e) demonstrates that incorporating prior information significantly enhances background editing quality, fulfilling our minimal-change requirement. Our optimal results are obtained using Inpainting with the Real Prior, a background-region mask, and the corresponding prompt P shown in Figure 3.

Minimal change for foreground. In contrast, color and texture edits require foreground modifications. As shown in Figure 2 (e-g), single-stage Inpainting and ControlNet models are insufficient under either Raw or Real Priors: Inpainting may distort object shapes, while ControlNet can produce unnatural results. To address this, we first produce an initial refined image using SDXL Inpainting, then use it as a conditional input for ControlNet to generate the final image. This combined approach (Figure 3) leverages the strengths of both methods, preserving the object's shape while achieving natural and precise color or texture changes.

3.2.3. Automatic filtering

To ensure generated images meet prompt requirements, the MLLM Llava-Next [33] model checks each image's feasibility and attributes. Using predefined questions, we filter out images that do not match the specified background, color, or texture. More details and example about the filtering questions can be found in the Supplementary Sec. D.2.

3.3. Feasibility effectiveness validation

Following the common practice [16, 28] to evaluate the impact of data feasibility, we fine-tune a CLIP [47] classifier, which encodes images and corresponding text prompts to calculate similarity scores for classification. We apply LoRA [27] modules to fine-tune both CLIP's image and text encoders. For each class $c_i \in C$, we use the prompt "a photo of [CLS]" as text input. Training is performed via supervised learning using cross-entropy loss, updating only the LoRA modules while keeping pretrained weights frozen.

In mixed training scenarios (real and synthetic data), the loss is a weighted combination defined as:

$$\mathcal{L}_C = \lambda \cdot \text{CE}(\text{Real}) + (1 - \lambda) \cdot \text{CE}(\text{Synth}) \tag{2}$$

where λ balances the contribution from real data, and CE denotes cross-entropy loss.

4. Experiments

4.1. Experiments setup

Dataset. Our synthetic data for background, color, and texture modifications require images with clearly defined foreground objects and visible backgrounds; hence, datasets dominated by foreground-only images, such as ImageNet [10], are unsuitable. Fine-grained datasets offer clearer comparisons between feasible and infeasible attribute variations. Therefore, we generate our minimal-change synthetic datasets from three fine-grained sources: Oxford Pets [46], FGVC Aircraft [41], and Stanford Cars [32]. Additionally, to specifically evaluate background modifications, we use the binary classification WaterBirds dataset [14], which pairs landbirds and waterbirds with water or land backgrounds.

Implementation details. Our VariReal pipeline utilizes SDXL Inpainting v0.1 and SDXL ControlNet v1.0 [61] based on Canny-edge conditioning, along with Stable Diffusion v2.1 [51] for prior image generation in background and texture modifications. The Llava-1.6-7B [35] model is employed for automatic filtering. Real images used for modification are sourced from the training split of each dataset, and performance is evaluated on the original test

	R	S	Pets [46]					AirC [41]					Cars [32]					Average				
			F	IF	Mix	Δ_1	Δ_2	F	IF	Mix	Δ_1	Δ_2	F	IF	Mix	Δ_1	Δ_2	F	IF	Mix	Δ_1	Δ_2
0-shot			— 91.0 —					— 23.8 —					— 63.2 —					— 59.3 —				
Real	✓		— 95.2 —					— 84.5 —					— 92.6 —					— 90.8 —				
Back.		✓	95.4	95.3	95.2	+0.1	-0.2	86.8	85.0	87.1	+1.8	+1.2	93.7	93.8	93.8	-0.1	+0.1	92.0	91.4	92.0	+0.6	+0.4
Color		✓	94.5	94.4	94.1	+0.1	-0.4	80.8	81.6	81.9	-0.8	+0.7	91.6	91.5	91.6	+0.1	+0.1	89.0	89.1	89.2	-0.1	+0.2
Text.		✓	93.8	93.3	92.8	+0.5	-0.8	81.6	81.9	82.0	-0.3	+0.3	90.9	87.7	91.8	+3.2	+3.0	88.8	87.6	88.9	+0.2	+0.7
Back.	✓	✓	95.3	95.3	95.3	+0.0	+0.0	88.0	88.4	88.6	-0.4	+0.4	93.8	93.7	93.6	+0.1	-0.2	92.4	92.5	92.5	-0.1	+0.1
Color	✓	✓	95.3	95.2	95.0	+0.1	-0.3	84.6	84.0	83.6	+0.6	-0.7	92.7	92.5	92.8	+0.2	+0.2	90.9	90.5	90.4	+0.4	-0.2
Text.	✓	✓	95.3	95.2	95.2	+0.1	-0.1	83.9	83.8	83.8	+0.1	-0.1	93.0	92.8	92.6	+0.2	-0.3	90.7	90.6	90.5	+0.1	-0.1

Table 1. Top-1 performance using the full training set and synthetic images generated by VariReal, including baseline, synthetic-only and synth + real. The number of synthetic images is set to five times the number of real images across all experiments. R/S indicates real/synthetic fine-tuning. F/IF denotes feasible/infeasible inputs, Mix indicates training with both. $\Delta_1 = F - IF$, and $\Delta_2 = Mix - \frac{F+IF}{2}$ measures the gain/loss of mixing compared to the average of individual setting.

set. We use $|P_f| = |P_{if}| = 5$ prompts per class, thus generating five synthetic images per real-image base.

We fine-tune a CLIP ViT-B/16 [13] classifier using LoRA modules with the rank of 16 applied to both image and text encoders, optimized with AdamW [38]. The scale factor λ is set to 0.5 to equally weight real and synthetic cross-entropy losses. To ensure fair training budget despite varying dataset sizes (real-only, synthetic-only, and mixed synth+real training), we fix the total maximum training iterations to ensure same optimizer update steps. Detailed generation and training hyperparameters are provided in the Supplementary Sec. B.

Baseline methods. To evaluate the impact of feasible versus infeasible synthetic data, we use zero-shot CLIP and CLIP fine-tuned on real images as baselines. We compare these baselines with CLIP trained exclusively on synthetic data and on combinations of synthetic and real data.

Evaluation protocol. We measure classification performance using top-1 accuracy (%). For dataset distribution analysis in Sec. 4.3.2, we report FID [42], CLIP score [47], DINO score [45], and LPIPS [62] scores. More details on those metrics are described in Sec. 4.3.2.

4.2. Classification with minimally changed data

4.2.1. The role of feasibility

Table 1 compares model performance across four training settings: (1) two baselines, (2) synthetic-only, and (3) real + synthetic training. To assess the role of feasibility, we define the metric $\Delta_1 = F - IF$, where F and IF denote performance using feasible and infeasible data, respectively. As shown in the second-to-last column, 4 out of 6 cases yield positive Δ_1, suggesting that feasible data generally performs slightly better than infeasible.

Under the synthetic-only setting, 56% of Δ_1 values (5/9) fall within 0.3% across each dataset column. Specifically, in the AirC [41] dataset, feasible data outperforms infeasible by 1.8% under the background setting, while infeasible data performs better by 0.8% and 0.3% under the color and texture settings, respectively. After incorporating real data (real + synthetic), 78% of Δ_1 values (5/9) remain within 0.3%, indicating that the performance gap between feasible and infeasible data is consistently small across settings.

> *Observation 1: Although feasible images perform slightly better, feasibility shows no clear impact on classification performance.*

4.2.2. The role of attribute

Although all settings in Table 1 outperform the zero-shot baseline, synthetic color and texture data remain less effective compared to the real data. For instance, in the synthetic-only setting, feasible and infeasible color data achieve 89.0% and 89.1% average performance, both below the real-only fine-tuning baseline of 90.8%. Even when combined with real data, color edits perform slightly worse by 0.1% and 0.2%.

In contrast, background modifications consistently improve performance. For instance, under synthetic-only training, feasible and infeasible backgrounds yield average accuracy gains of 1.2% and 0.6%, respectively, and 1.6% and 1.7% in the real + synthetic setting.

We further validate the benefits of background modifications on the WaterBirds [14] dataset (see Supplementary Sec. E). Both feasible and infeasible background edits outperform real-only setting, with improvements of 0.9% and 6.7% respectively in the synthetic-only setting, and 7.2% and 8.8% in the real + synthetic setting.

> *Observation 2: Compared to fine-tuning on real data alone, adding synthetic data with background modifications improves performance, whereas synthetic foreground edits (color and texture) are less effective.*

4.2.3. The role of mixed training

To assess the effect of mixing feasible and infeasible data, we construct a balanced synthetic dataset (third column

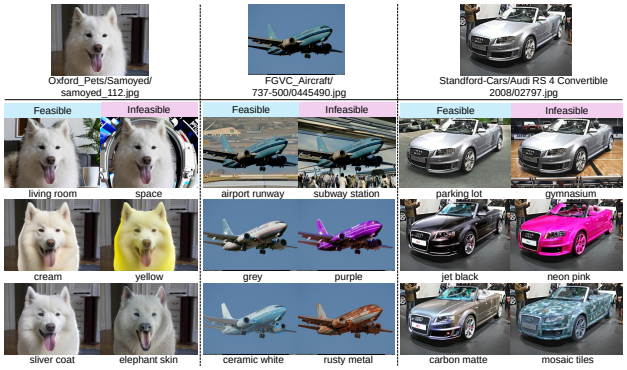

Figure 4. Selected generation results from the three datasets. Only target prompt keywords are shown; detailed background and texture descriptions are omitted. Please zoom in for visual details.

of each subtable in Table 1), with a total size five times that of the real training set. We define the metric $\Delta_2 = \text{Mix} - \frac{F+IF}{2}$ to measure the performance gain from mixing, relative to the average of using feasible and infeasible data separately.

In the real + synthetic setting, mixed training yields comparable performance, with average Δ_2 deviations within 0.2%. In contrast, under the synthetic-only setting, mixing leads to greater gains—0.4%, 0.2%, and 0.7% improvements on average for background, color, and texture edits—indicating stronger complementarity between feasible and infeasible data. Further analysis for this is provided in Supplementary Sec. F. This suggests that, unlike ALIA [14], modifications need not be strictly feasible.

Observation 3: It is not necessary to strictly generate only feasible synthetic images to achieve performance gain.

4.3. Analysis of minimally changed data

4.3.1. Qualitative results

To assess the quality of VariReal-generated images, Figure 4 presents qualitative examples from all three datasets. These examples demonstrate that the edits follow the text prompts with minimal changes and align with our feasibility definition—existing in real world. For instance, "neon pink" is not a released color for the "Audi RS 4 Convertible 2008" and is thus treated as infeasible. The images show the expected modification, while the rest of the image remains unchanged from the real source. More examples are provided in Supplementary Sec. G.

To further validate image quality, we conducted a user study with six human annotators using a questionnaire. Evaluators assessed each image on two aspects: (1) feasibility—whether feasible images appear realistic and infeasible ones do not—and (2) naturalness, rated on a 1–5 scale, where 5 indicates the most natural appearance. More details about the scoring setup are included in Supplementary Sec. G.

	Back		Color		Texture		Averaged	
	F	IF	F	IF	F	IF	F	IF
Feasibility Correctness/%	92.1	87.5	94.4	85.2	90.1	80.9	92.2	84.2
Naturalness Score(0.0-5.0)	4.5	4.1	3.62	3.90	3.70	3.88	3.94	3.96

Table 2. Human evaluation of the generated dataset based on feasibility correctness and naturalness scores, validating its suitability for downstream tasks.

Feasibility is central to our pipeline, ensuring a clear distinction between feasible and infeasible subsets. As shown in Table 2, feasibility correctness is high, with error rates below 8% for feasible and 16% for infeasible data. The slightly lower accuracy for infeasible cases stems from occasional mismatched background-object combinations and difficulty capturing fine-grained texture details—particularly in the AirC dataset, as noted in annotator feedback (see Supplementary Sec. G). These results support the effectiveness of our approach, with VariReal reliably generating high-quality edits, further refined by automatic filtering (Sec. 3.2.3).

Regarding how natural the generated images are, VariReal images received acceptable naturalness scores from human annotators—averaging 3.94 for feasible and 3.96 for infeasible data. For failure cases, some generated images appear less natural (see Supplementary Sec. G) because of a dramatic change from the original color to a new color, such as red to white.

4.3.2. Distribution analysis

We analyze the dataset using several similarity metrics to better understand the distributional differences between feasible and infeasible data and their relation to in- and out-of-distribution. We compute the Fréchet Inception Distance (FID) [42] to quantify the distributional similarity between generated and real data. Additionally, we use: **CLIP Score**: calculated cosine similarity for feature from the ViT-L/14 model [13]. **DINO Score**: computed cosine similarity for feature from the DINOv2-Base model [45] for feature extraction. And **LPIPS Score** [62].

Figure 5 shows that feasible samples generally resemble in-distribution data more closely than infeasible ones, aligning better with the real data distribution. This observation is supported by the metrics in Table 3, which reports average scores across the three datasets. All three metrics indicate that feasible data is closer to real data. While CLIP and DINO scores show limited sensitivity to fine-grained differences, LPIPS captures subtle variations more effectively.

Interestingly, both feasible and infeasible foreground modifications (color and texture) are closer to real data than background edits. For instance, in the AirC [41] dataset, FID peak scores for foreground edits are much

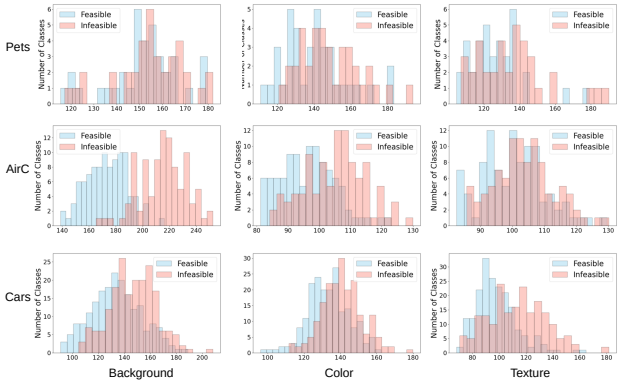

Figure 5. The FID score settings compared using feasible and infeasible settings across different datasets.

Settings	Inputs	CLIP (↑)	DINO (↑)	LPIPS (↓)
Background	F	**0.914**	**0.861**	**0.447**
	IF	0.886	0.830	0.477
Color	F	**0.951**	**0.956**	**0.189**
	IF	0.904	0.939	0.254
Texture	F	**0.936**	**0.949**	**0.207**
	IF	0.898	0.925	0.218

Table 3. The average DINO, CLIP and LPIPS scores calculated between generated synthetic image and corresponding real images for three datasets. F/IF denotes feasible/infeasible inputs.

lower (around 95 and 110 for feasible/infeasible) than for background edits (around 170 and 220). Table 3 shows similar trends—for instance, the average DINO score for color is about 10% higher than for background. However, as discussed in Sec. 4.2.2, only background modifications consistently improve classification performance. This highlights the following:

> *Observation 4: Classification tasks are object-centric: although foreground (color and texture) modifications align more closely with real data distributions, changing them may deviate from meaningful class-relevant features, leading to weaker effects.*

4.3.3. Scaling the number of training images

To further understand the impact of synthetic data by VariReal, we conducted a scaling analysis on the AirC dataset [41], adjusting feasible/infeasible synthetic-to-real ratios from 1:1 to 5:1.

Our results in Figure 6 reveal a nonlinear relationship between performance and data scale. While background modifications always benefit the downstream tasks, color and texture modifications achieve peak accuracy at smaller scales. Notably, performance slightly exceeds the baseline at these peaks but declines as more synthetic images are added. This indicates that both feasible and infeasible color

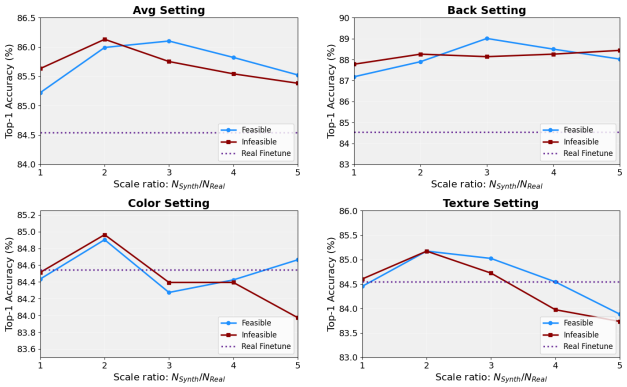

Figure 6. The scaling experiment results for the FGVC-Aircraft [41] dataset are shown for background, color, and texture settings. The horizontal axis represents the scale factor for synthetic images relative to real images. Here, the total real image training set is used, with scale factors ranging from 1 to 5.

and texture data behave similar to OOD data, while feasible data being relatively closer to the real distribution. Large-scale use of such data does not provide meaningful in-distribution information for downstream tasks. However, a limited amount can serve as effective augmentation, enhancing model performance and robustness.

> *Observation 5: Synthetic data with color and texture modifications can enhance classification performance as augmentation, but their effectiveness is limited to specific scaling ranges. In contrast, background modifications consistently yield performance gains.*

5. Conclusion

In this work, we present VariReal, a pipeline for systematically investigating the impact of minimal-change feasible and infeasible synthetic data. By introducing controlled variations in background, color, and texture across three fine-grained datasets, we assess the role of feasibility through LoRA-based fine-tuning of a CLIP classifier. Our findings reveal a counter-intuitive result: feasibility does not significantly affect classification performance. Although typically assumed to benefit downstream tasks, feasible synthetic variations in color and texture are no more effective than real data—and in some cases, even degrade performance. In contrast, background modifications consistently improve accuracy, regardless of feasibility. This suggests that, for object-centric classification, altering foreground attributes may disrupt class-relevant signals and yield limited gains. Overall, our results underscore the nuanced effects of different attribute modifications and offer new insights for designing effective synthetic data generation strategies.

Acknowledgements. Jae Myung Kim thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) and the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD programs for support.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2, 4, 1, 3

[2] Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. VisMin: Visual minimal-change understanding. *arXiv preprint arXiv:2407.16772*, 2024. 3

[3] Kaixin Bai, Huajian Zeng, Lei Zhang, Yiwen Liu, Hongli Xu, Zhaopeng Chen, and Jianwei Zhang. ClearDepth: enhanced stereo perception of transparent objects for robotic manipulation. *arXiv preprint arXiv:2409.08926*, 2024. 3

[4] Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 164–172. JMLR Workshop and Conference Proceedings, 2011. 2

[5] Briaai. Briaai background removal v1.4 model, 2024. https://huggingface.co/briaai/RMBG-1.4. 5

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 1, 3

[7] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198, 2024. 1

[8] Victor G Turrisi da Costa, Nicola Dall'Asen, Yiming Wang, Nicu Sebe, and Elisa Ricci. Diversified in-domain synthesis with efficient fine-tuning for few-shot classification. *arXiv preprint arXiv:2312.03046*, 2023. 2, 3

[9] Ashwin De Silva, Rahul Ramesh, Carey Priebe, Pratik Chaudhari, and Joshua T Vogelstein. The value of out-of-distribution data. In *International Conference on Machine Learning*, pages 7366–7389. PMLR, 2023. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. 5

[11] Li Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE signal processing magazine*, 29(6):141–142, 2012. 2

[12] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 4

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6, 7

[14] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E. Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *NeurIPS*, 36:79024–79034, 2023. 2, 3, 5, 6, 7, 1, 4

[15] Peter Eigenschink, Thomas Reutterer, Stefan Vamosi, Ralf Vamosi, Chang Sun, and Klaudius Kalcher. Deep generative models for synthetic data: A survey. *IEEE Access*, 11: 47304–47320, 2023. 2

[16] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392, 2024. 3, 5

[17] Chengjian Feng, Yujie Zhong, Zequn Jie, Weidi Xie, and Lin Ma. Instagen: Enhancing object detection by training on synthetic dataset. In *CVPR*, pages 14121–14130, 2024. 3

[18] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023. 3

[19] Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. *arXiv preprint arXiv:2210.06441*, 2022. 2

[20] Aldren Gonzales, Guruprabha Guruswamy, and Scott R Smith. Synthetic data in health care: A narrative review. *PLOS Digital Health*, 2(1):e0000082, 2023. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[22] Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Philip Torr, Adel Bibi, and Bernard Ghanem. SynthCLIP: Are we Ready for a fully synthetic CLIP training? *arXiv preprint arXiv:2402.01832*, 2024. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[24] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 2, 3

[25] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2

[27] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3, 5

[28] Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. DataDream: Few-shot guided dataset generation. *arXiv preprint arXiv:2407.10910*, 2024. 2, 3, 5, 1

[29] Sungnyun Kim, Junsoo Lee, Kibeom Hong, Daesik Kim, and Namhyuk Ahn. Diffblender: Scalable and composable multimodal text-to-image diffusion models. *arXiv preprint arXiv:2305.15194*, 2023. 3

[30] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 4

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 1

[32] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, pages 554–561, 2013. 5, 6, 4, 8

[33] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 5

[34] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards understanding cross and self-attention in stable diffusion for text-guided image editing. In *CVPR*, pages 7817–7826, 2024. 1, 3

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024. 5

[36] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. GroundingDino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5

[37] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, page 100017, 2023. 1

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6, 1

[39] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and L Repaint Van Gool. Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471. 3, 4

[40] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022. 3, 1, 2

[41] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5, 6, 7, 8, 4

[42] Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fr\'echet inception distance. *arXiv preprint arXiv:2009.14075*, 2020. 6, 7

[43] Ivona Najdenkoska, Animesh Sinha, Abhimanyu Dubey, Dhruv Mahajan, Vignesh Ramanathan, and Filip Radenovic. Context diffusion: In-context aware image generation. *arXiv preprint arXiv:2312.03584*, 5, 2023. 3

[44] 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. 3

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 6, 7

[46] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5, 6, 3, 4, 7

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3, 5, 6, 1, 2

[48] Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *CVPR*, pages 21783–21794, 2024. 3

[49] 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. 3

[50] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[51] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 3, 4, 5

[52] 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

[53] 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. *NeurIPS*, 35:36479–36494, 2022. 3

[54] Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *CVPR*, pages 8011–8021, 2023. 2, 3

[55] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *ICCV*, pages 1206–1217, 2023. 2, 3

[56] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, pages 18381–18391, 2023. 3

[57] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pages 1–28, 2024. 2

[58] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-Image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 3, 5, 1, 2

[59] Zhuoran Yu, Chenchen Zhu, Sean Culatana, Raghuraman Krishnamoorthi, Fanyi Xiao, and Yong Jae Lee. Diversify, don't fine-tune: Scaling up visual recognition training with synthetic images. *arXiv preprint arXiv:2312.02253*, 2023. 2

[60] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1

[61] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 3, 4, 5, 1, 2

[62] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6, 7

[63] Shu Zhang, Xinyi Yang, Yihao Feng, Can Qin, Chia-Chih Chen, Ning Yu, Zeyuan Chen, Huan Wang, Silvio Savarese, Stefano Ermon, et al. Hive: Harnessing human feedback for instructional visual editing. In *CVPR*, pages 9026–9036, 2024. 3

[64] Fan Zhou, Zengzhi Wang, Qian Liu, Junlong Li, and Pengfei Liu. Programming every example: Lifting pre-training data quality like experts at scale. *arXiv preprint arXiv:2409.17115*, 2024. 1