Does Feasibility Matter? Understanding In- and Out-of-Distribution Data Impact in Synthetic Training Datasets for Classification

Yiwen Liu[1], Jessica Bader[1,2,3,4], Jae Myung Kim[1,2,3,4]

[1]Technical University of Munich [2]Helmholtz Munich [3]MCML [4]MDSI

Figure 1. To investigate the necessity of ensuring *feasibility* (attributes assigned to an object in a synthetic image could exist in the real-world training domain) within synthetic training data, we propose our minimal-change pipeline VariReal. We isolate target attributes in three categories: background, color, and texture, allowing us to compare CLIP [34] classification performance with *feasible* and *infeasible* synthetic data from different perspectives. We also compare images generated by VariReal to those from prior text-guided editing methods [6, 35], covering both feasible and infeasible attributes in each category. Edition prompts are shown below the generated images.

Abstract

With the development of increasingly 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 cat floating above the ground or with unrealistic texture artifacts. We consider these types of images ***infeasible****. Intuitively, training with infeasible images should be detrimental to a model's ability to generalize to real data; hence, infeasible images are typically treated as out-of-distribution (OOD) and removed from the training set whenever possible via filtering techniques. But does feasibility really matter? In this paper, we investigate the necessity of feasibility when generating synthetic training data for classifiers by using an LLM to define per-class in-distribution (ID) and OOD attributes realting to the three target categories: background, color, and texture. We introduce a minimal-change generation pipeline, VariReal, to create feasible and infeasible comparison pairs from real images. In this way, we isolate the target attribute from other information in the synthetic data. We show that feasibility of the synthetic data does not majorly affect performance on several fine-grained classification datasets when LoRA fine-tuning CLIP on synthetic data, showing less than 1 percentage point difference in top-1 accuracy between feasible and infeasible datasets across almost all test settings when evaluated on Oxford Pets, FGVC Aircraft, and Stanford Cars. More importantly, we show that mixing feasible and infeasible data within synthetic training datasets does not significantly impact performance when compared with models trained on only feasibles or infeasible synthetic images.*

1. Introduction

In recent years, large-scale pre-trained models [7, 24, 31, 37, 48, 57] have significantly surpassed traditional deep learning and machine learning approaches in various tasks. However, as the scale of training data grows, access to high-quality data has become increasingly limited [61], posing

challenges to further improving these large models' capabilities. With the popularity of generative models [19, 30] like Stable Diffusion [48], researchers are increasingly leveraging these models to generate high-fidelity synthetic data that closely resembles real-world data, offering a solution to data scarcity [14, 18].

Prior studies have explored synthetic data generation under a limited few-shot real image setting [8, 13, 20, 22, 27, 51, 52, 56]. These works aim to create synthetic data that approximates the real-world data distribution while avoiding overfitting to the limited available examples. Some studies [22, 27] suggest that synthetic data can offer benefits beyond those of real data. However, the inherent randomness in the image generation process of diffusion process [24, 48] can introduce domain shifts [22] or implausible scenarios like "a dog floating in the sky" [51] that do not reflect realistic patterns, which might intuitively be counterproductive.

Interestingly, some studies [4, 9, 17] suggest that OOD data can positively impact downstream tasks when mixed with real data in certain proportions. A typical example is data augmentation [17], where some augment methods introduce OOD data relative to the original distribution yet still provide benefits. While the advantages of OOD data generally diminish as divergence from the original distribution increases [9], these findings demonstrate OOD data is not always harmful. Conversely, incorporating feasible content similar to the training domain is naturally beneficial. The ALIA method [13] augments datasets with "feasible backgrounds", demonstrating performance improvement with ID data. This raises a key question: does training data feasibility affect downstream tasks, and could control the incorporation of such OOD data improve performance?

This work introduces an automatic minimal-change generation pipeline, **VariReal**, based on Stable Diffusion [48]. VariReal allows us to control object feasibility to create targeted synthetic comparison pairs, as the example shows in Figure 1. We evaluate feasibility effectiveness by employing the CLIP [34] classifier and fine-tuning it on the synthetic dataset generated using VariReal. Precisely, we manipulate three key object attributes—background, color, and texture—to examine classifier performance under two conditions: (1) fine-tuning with synthetic data only and (2) mixed training with real and synthetic data. For each attribute, we consider feasible data as ID and infeasible data as OOD. For example, a black "Bombay" dog is a plausible real-world instance (ID), while a "Bombay" dog with white fur is infeasible and thus categorized as OOD.

Our experiments on three fine-grained datasets reveal several key insights. We also show that modifications similar to ALIA [13] do not necessarily need to select only feasible scenarios. Regardless of feasibility, changing the background can enhance the classifier's focus on the primary task, while foreground modifications for color and texture often challenge the classifier's learning process. We also demonstrate that mixing synthetic data can yield performance benefits when paired with real data.

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 at a time. This approach can be applied out-of-the-box to any object-centric classification dataset without additional fine-tuning.
- We generate and provide feasible (ID) and infeasible (OOD) dataset comparison pairs based on real images, covering three controlled attributes.
- To explore feasible and OOD data roles, we fine-tune CLIP with LoRA modules. Analyzing classification scores, we offer new insights into the impact of feasibility and the strategic use of OOD data for enhancing downstream task performance.

2. Related Work

Effect of out-of-distribution data. OOD data, defined relative to ID data, introduces a distribution shift between train and test data. OOD data is generally categorized into semantic and covariance shifts [54]; here, we focus on covariance shifts. Early works [4, 17] 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 using only simple OOD data types.

Silva *et al*. [9] and Geiping *et al*. [17] 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 [21]) and datasets (e.g., MNIST [11]), our work seeks to extend OOD data study to more complex scenarios with diffusion models, utilizing advanced architectures like CLIP to deepen understanding of OOD effects.

Learning with synthetic data. Several studies [8, 22, 27, 51, 56] focus on generating synthetic data that approximates real-world distributions. This approach aims to create a dataset larger than the few-shot samples. Generated data supports various tasks, including object recognition [8, 13, 27, 51], object detection [15], and semantic segmentation [52]. Its effectiveness is demonstrated by training models exclusively on synthetic data or in combination with real data [22, 27]. In this work, we focus specifically on object classification.

Automatic approach for minimal change generation. Unlike synthetic data generation methods that focus on creating novel and diverse in-distribution images to expand limited real training sets [27], minimal change generation aims only to modify specific areas or attributes of existing images. Generative models, particularly diffusion-based

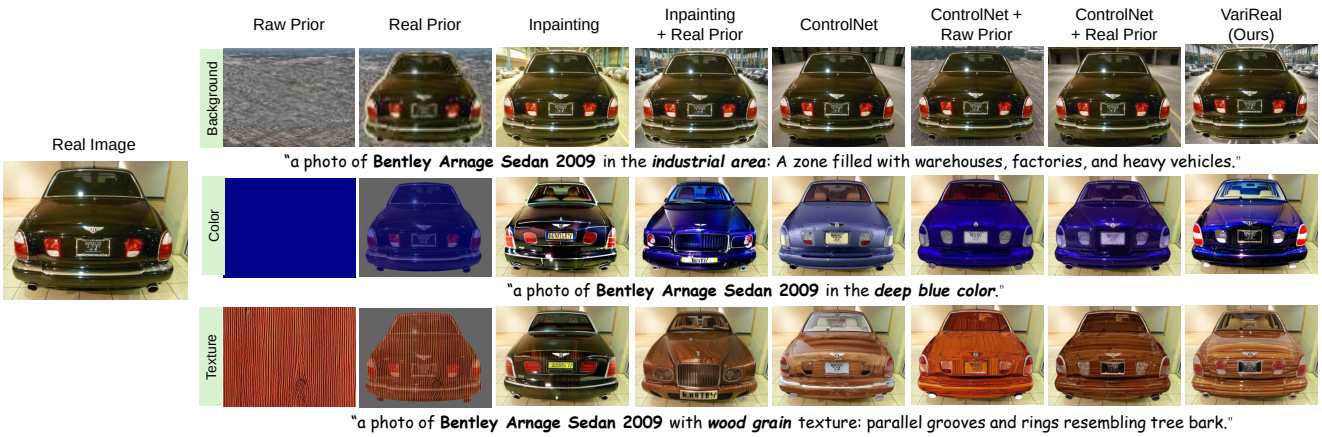

Figure 2. We compare generated images for various potential methods: inpainting [39] only, inpainting with Real Prior, ControlNet [58] only, ControlNet with Raw Prior, ControlNet with Real Prior, and our final generation results under three settings. The generation prompts are listed below the generated images, and the Raw Prior and Real Prior are shown in columns 2-3.

approaches [43, 46, 48, 50], facilitate efficient image editing without requiring manual annotation [22] or physical graphics engines [3, 45]. 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 for IP-Adaptor [55] and Canny edge maps for ControlNet [58].

These methods fall into two main categories: fine-tuning [6, 16, 60] and attention- or mask-based diffusion approaches [23, 35]. 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 [35] and P2P [23], substitute certain self- or cross-attention layers in the U-Net [49]'s denoising process, leveraging the interpretability of attention maps. However, these methods may not perform well in all scenarios, particularly with real images. Mask-based diffusion models, including inpainting [38] and specialized mask-driven editing methods [28, 42, 53], allow controlled modifications within specified areas, preserving regions outside the mask. When editing object attributes, however, mask-based models can sometimes alter subtle shape details of the object. Methods like ControlNet [58] can help maintain an object's original structure during edits.

The most comparable work to ours is VisMin [2], which generates minimal-change data to enhance vision-language model comprehension. However, their approach does not address background modifications and shows a high failure rate in other settings. In this study, we introduce an automatic, off-the-shelf approach that enables minimal, photo-realistic edits for any combination of real images and instructions.

3. Method

3.1. Preliminaries

Task formulation. Our goal is to analyze the impact of feasible (ID) or infeasible (OOD) synthetic data, I_{Syn}, where feasibility is specified in the context of each individual class c_i, $i = \{1, \ldots, C\}$. Our diffusion-based VariReal pipeline generates I_{Syn} comparison pairs by modifying a shared real-image base, I_{Real}, with unique prompt guidances. In this way, we isolate the feasibility of target attributes while otherwise minimally modifying the image (i.e. we train with the *same* dog in black and blue). Guidances are LLM-generated sets of class-feasible P_f (ID) and infeasible P_{if} (OOD) prompts. We combine each $r \in I_{\text{Real}}$ with every $p \in P_f, P_{if}$, such that every real image is repeated $|P_f| = |P_{if}|$ times. Using number $|P_f| > 1$, we can evaluate the impact of additional synthetic augmentations without requiring more real images. We generate datasets to evaluate feasibility in the context of three isolated categories: background, color, and texture. Note that texture properties also encompass color characteristics. We then LoRA fine-tune CLIP on the ID and OOD synthetic data to compare the downstream training impact of ID and OOD synthetic data on classification task.

Fine-tuning with low-rank adaptation. The Low-Rank Adaptation [25] introduces low-rank decomposition into the pre-trained weight matrix to reduce the number of learnable parameters. The final weights after fine-tuning could be expressed by $h = W_0 x + BAx$, 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 [48] 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:

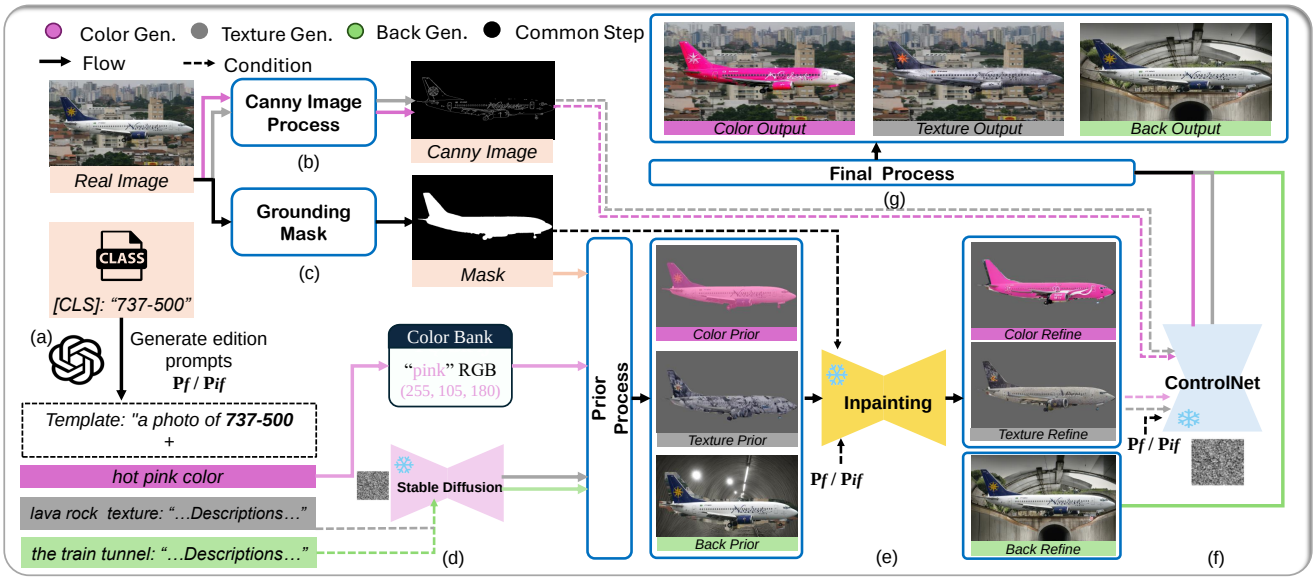

Figure 3. In our VariReal synthetic image generation pipeline, common shared steps are shown in black, while the processing of color, texture, and background are represented in pink, gray, and green, respectively. Solid lines indicate the primary workflow of our method, and dotted lines denote conditional model processing steps.

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

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 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 [38] (e.g., SDXL Inpainting) or Canny-edge-based ControlNet [58] models (e.g., SDXL ControlNet), using a base prompt P_{base} = "a photo of a [CLS]". However, the original image heavily influences Inpainting modifications. For instance, in the third column of Figure 2, when attempting to change the color of a black car, the output image often retains a dark hue, thereby limiting the model's effectiveness in altering attributes.

On the other hand, ControlNet-generated images can preserve the object's structure without being affected by the original attributes, as shown in the fifth column. However, they are often less natural. Furthermore, objects in these images sometimes appear floating in the background, detracting from overall 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 of the components, i.e. combining Inpainting's realism with ControlNet's preciseness.

3.2. VariReal: Generating Minimal-Change Data

In this section, we present our zero-shot minimal-change image generation pipeline. We first generate prompts P_f and P_{if} (Figure 3 (a)) as described in Section 3.2.1. The required mask and Canny images are created as shown in (b) and (c) under **Grounding mask** and **Canny images** in Section 3.2.2. We use prior information (d) to guide the VariReal generation, explained in **Prior generation** of Section 3.2.2. The diffusion Inpainting model and ControlNet then generate images based on P_f and P_{if} (Section **Minimal change**). Finally, **Final process** (g) and Section 3.2.3 describe the final steps and MLLM model filtering.

3.2.1. Guidance Prompt

P_f and P_{if} will guide the Stable Diffusion model to edit desired content. To generate as many accurate P_f and P_{if} 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. The prompt example is shown in the lower left corner of Figure 3.

Although large language models possess broad knowledge across various domains, ChatGPT still regularly designates 'feasible' attributes for a target object that do not exist in the real world, particularly for fine-grained classes for which it has limited knowledge. To address this issue, we design additional prompts to instruct the model

to perform preliminary checks and filtering on its outputs. Specifically, for the attributes of a given category, the model filters out infeasible features that do not align with the real attributes, ensuring that the generated prompts meet our requirements. 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 Figure 3. Details of the generation process are provided in the Appendix.

3.2.2. Prior-Guided Minimal Change Generation

Grounding mask. For models requiring mask information, we use Grounding Dino [36] to generate bounding boxes bbox_i, which are then fed into the SAM2 [47] 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.

Canny images. In our method, we use the Canny image ControlNet [58] model. For background editing, the Canny image is created by extracting the foreground Foreground_i from mask_i. For color and texture modifications, we generate a complete Canny map from the real image.

Prior generation. For the "Background Prior" and "Texture Prior", we use prompts P_f or P_{if}. For color edits, RGB values are selected from a predefined Color Bank. We refer to the initial outputs as *Raw Prior* and combine them with real images to produce the *Real Prior* as shown in column 2-3 in Figure 2.

We replace the original image's background with the generated Background Prior, using mask dilation to preserve context and maintain a natural spatial relationship (e.g., keeping a pet grounded). For color or texture changes, we overlay the generated prior as an alpha channel to retain the subject's shape and details. Both Raw Prior and Real Prior are tested with the ControlNet model, conditioned by IP-Adaptor [55], while Inpainting uses only the Real Prior. Comparison results are shown in Figure 2.

Final process. The final step in our generation process is to copy the invariant areas from the original image and paste on the generated image to ensure minimal change.

Minimal Change for Background. Figure 3 shows that incorporating prior information significantly improves success rates and generation details for background settings. Using Inpainting with Real Prior, a background region mask, and the corresponding prompt P, our background modification approach meets desired requirements and achieves the best performance.

Minimal change for foreground. Conversely, color and texture require foreground modifications. In Figure 2, we show that single-stage Inpainting and ControlNet are insufficient: Inpainting may apply unintended object shape modifications, while ControlNet may produce unnatural results.

To address this, we first generate an initial refined image with SDXL Inpainting, which is then used as a conditional input for ControlNet to generate the final image. This approach in Figure 3 combines the strengths of Inpainting and ControlNet, ensuring the main object's shape remains intact while achieving the desired, natural color or texture.

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, and exclude unrealistic object-background combinations, such as "a flying plane in a hangar." Although a hangar is feasible for aircraft, it is considered infeasible in cases where the original image shows the plane flying. More details about the filtering questions can be found in the Appendix.

3.3. Feasibility Effectiveness Validation

We evaluate the impact of data feasibility by training CLIP [34] on the synthetic data and testing on real images. Building on prior research [27], we fine-tune both CLIP's image and the text encoder by incorporating LoRA [25] modules. For the text encoder, we use the prompt "a photo of [CLS]" for each class in the set of classes C. We employ a supervised learning strategy to train with a classification loss.

When the classifier is trained using only the synthetic dataset, the loss function is the cross-entropy loss. In the mixed training scenario, the loss function is an average of the real and synthetic data losses, weighted by a parameter λ. The loss function is formulated as follows:

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

where λ is the weight assigned to the loss from real data, and the function CE denotes cross-entropy loss.

4. Experiments

4.1. Experiments Setup

Dataset. Because our modifications for background, color, and texture require both a well-defined foreground object and a visible background, datasets with images dominated solely by foreground objects, such as ImageNet [10], are unsuitable for our experiments. Fine-grained variations provide a better basis for comparing feasible and infeasible attribute changes. Thus, we select three base datasets to generate our minimal-change synthetic datasets: Oxford Pets [44], FGVC Aircraft [40], and Stanford Cars [32], To further validate the background modification setting, we selected the binary classification WaterBirds [13] dataset,

which includes landbirds and waterbirds against land or water backgrounds. To prevent the classifier from relying on background cues, the dataset introduces a 5% bias, with some landbirds on water and waterbirds on land.

Implementation details. In our VariReal pipeline, we use Stable Diffusion [48] v2.1 to generate prior images for background and texture modifications. We employ SDXL Inpainting v0.1 and the SDXL ControlNet [58] v1.0 model based on Canny edge images. For automatic filtering, Llava-1.6-7B model is used. The real images for modification are sourced from each dataset's training set. Detailed generation parameters for each dataset and class can be found in Appendix. Unless noted otherwise, we train with 100 real images per class and $|P_f| = |P_{if}| = 5$, meaning we generate 5 synthetic images corresponding to each real image base.

We use the AdamW [29] optimizer to train the CLIP ViT-B/16 classifier with LoRA applied to both the image and text encoders, using a rank of 16. The scale factor λ is set to 0.5 to balance the contribution of real and synthetic cross-entropy losses. Additional training parameters are detailed in the Appendix. Notably, since the dataset sizes vary for different settings, namely only real, only synthetic and synth + real training, we ensure a fair comparison by setting the same training iterations and adjusting the number of training iterations to the equal iterations for 70 epochs of the synth + real dataset setting.

Baseline methods. Our primary approach focuses on self-comparisons based on our defined feasible and infeasible settings. Specifically, we analyze CLIP classification performance by training on either only synthetic or synth + real data. To assess the impact of feasible or infeasible data on classification performance, we use zero-shot CLIP performance as a baseline. We also fine-tune CLIP on the original real data; however, it is important to note that the purpose of this work is to compare ID and OOD data, rather than propose an image augmentation pipeline. Hence, the real images should not be considered a competing method.

Evaluation protocol.

We use top-1 accuracy(%) to assess our model's performance. or a given test sample i, let $\hat{y}_i^{(k)}$ denote the k-th highest-ranked predicted label, and let y_i represent the true label. To validate whether the correctly learned set of one model is a subset of another, we use the inclusion coefficient, defined as: Inclusion Coefficient $= \frac{|A \cap B|}{|A|}$, where a value closer to 1 indicates higher overlap between the two models' correct predictions. Additionally, we assess the overlap in learned knowledge between the two models by calculating the Jaccard index for the sets of correctly predicted samples across the test set, defined as $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, where A and B represent the correctly classified samples in two different training configurations. This metric evaluates the extent of overlap between the correct prediction sets from two models.

	R	S	Pets [44]			AirC [40]			Cars [32]			Avg	
			F	IF	diff	F	IF	diff	F	IF	diff	F	IF
0-shot			91.0			23.8			63.2			59.3	
Back.			95.4	95.3	0.1	86.8	84.1	0.7	93.7	93.8	-0.1	92.0	91.1
Color		✓	94.5	94.4	0.1	80.8	81.6	0.2	91.6	91.5	0.1	89.0	89.1
Text.			93.8	93.3	0.5	81.6	81.9	-0.3	90.9	86.8	4.1	88.8	87.3
Back.			95.3	95.3	0.0	88.0	88.4	-0.4	93.8	93.7	0.1	92.4	92.5
Color	✓	✓	95.3	95.2	0.1	84.6	84.0	0.6	92.7	92.5	0.2	90.9	90.5
Text.			95.3	95.2	0.1	83.9	83.8	0.1	93.0	92.8	0.2	90.7	90.6
Real	✓		95.2			84.5			92.6			90.8	

Table 1. Top-1 performance using the full training set and synthetic images generated by VariReal, with training setups including synthetic-only and synth + real. All results use CLIP ViT-B/16 as the base classification model, with the number of synthetic images set to five times the number of real images. Datasets include Oxford Pets (Pets), FGVC Aircraft (AirC), and Stanford Cars (Cars). R/S means using real/synthetic images for fine-tuning.

For dataset distribution analysis, we employ a common metric for evaluating the distance between generated and real data distributions: Fréchet Inception Distance score [41]. Additionally, we calculate the CLIP Score, Dino Score and LPIPS Score to measure data similarity: 1) **CLIP Score**: We used the ViT-L/14 model [34]. 2) **Dino Score**: We employed the DINOv2-Base [57] model for feature extraction. 3) **LPIPS Score**: [59]: The Learned Perceptual Image Patch Similarity score is used to capture fine-grained visual differences between images.

4.2. Classification Performance with Minimal-change Data

4.2.1. Accuracy Comparison

Table 1 compares model performance across four training methods: baseline, synthetic-only, synth + real, and real-only training. For synth + real training, we fix the synthetic data volume at five times the number of real images, using all available real images from the training set. First, we notice that in almost all settings, the difference between the ID and OOD data is less than 1%. Furthermore, ID data is not universally better: for the Cars dataset [32], infeasible OOD data performs better in three out of six scenarios. Our modified synthetic data is also able to outperform the original real data in five out of eighteen synthetic-only settings.

To further validate the positive effect of minimal background changes, we experimented with feasible and infeasible background modifications on the binary classification task using the WaterBirds [13] dataset. As shown in Table 2, both feasible and infeasible background modifications enhance performance, with infeasible backgrounds yielding even greater improvement than feasible ones by 5.79 percentage points in the synthetic-only setting and 1.68 percentage points in the real + synth setting.

4.2.2. Training with All Data

Next, we evaluate the effect of mixing feasibile and infeasible attributes. We create a synthetic dataset with balanced

	R	S	WaterBirds [13] F	IF
0-shot Base			78.95	
Back.		✓	86.59	92.38
	✓	✓	92.85	94.53
Real	✓		85.66	

Table 2. The top-1 performance using the full training set and synthetic data, with training setups including synthetic-only and synth. + real data. All results use CLIP ViT-B/16 as the base classification model, with synthetic images set to five times the number of real images. The attribute of experimented dataset WaterBirds [13] is background.

	R	S	Pets [44]	AirC [40]	Cars [32]	Avg
0-shot Base			91.00	23.80	63.18	59.33
Back.			95.18	86.58	93.84	**91.87**
Color		✓	94.14	81.85	92.14	89.38
Text.			92.78	82.04	91.84	88.89
Back.			95.29	87.99	93.60	**92.29**
Color	✓	✓	95.02	83.36	92.77	90.37
Text.			95.15	83.83	92.56	90.51
Real		✓	95.23	84.54	95.59	90.79

Table 3. The top-1 performance results using the full training set combined with a balanced mix of feasible and infeasible synthetic data, with training setups including synthetic-only and synthetic + full real data. All results use CLIP ViT-B/16 as the base classification model, with synthetic images set to five times the number of real images. Underlined values indicate performance improvements over Table 1 after mixing feasible and infeasible data.

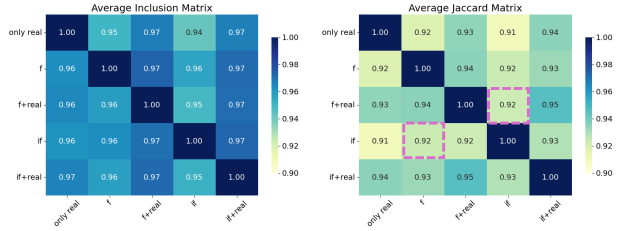

Figure 4. The averaged Inclusion and Jaccard index matrix for three editing settings across three datasets. "f" = feasible, "if" = infeasible, "real" = training with real images.

samples from both the feasible and infeasible datasets, with a total volume five times that of the real data for training.

In Table 3, we find similar overall classification results as in Table 1. Combining feasible and infeasible data even marginally improves final classification metrics compared to separate training for synthetic-only training on the AirC [40] and Cars [32] datasets, specifically for color and texture on Aircraft and Cars.

4.3. Classification Results Analysis

We evaluate prediction correctness for each test sample to determine if models learn similar knowledge under different training settings. As shown in Figure 4 the Inclusion Matrix shows that the knowledge learned by different model do not have a subset relationship, so the pink box on the Jaccard

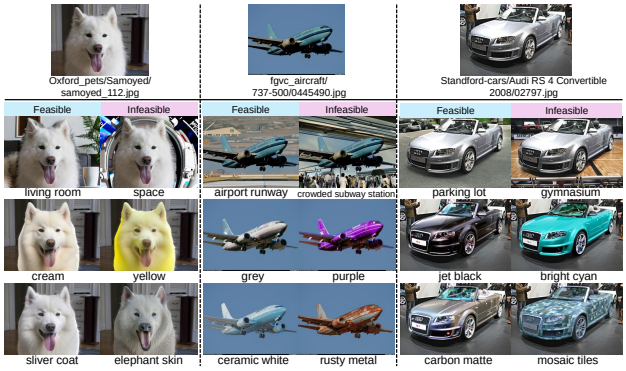

Figure 5. The selected generation visualizations for the three datasets. We only display the target prompt words, omitting detailed descriptions for background and texture.

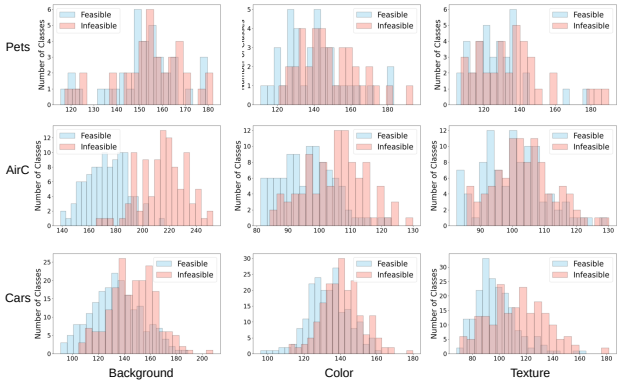

Figure 6. The FID score settings compared using feasible and infeasible settings acorss different dataset.

Matrix proves that while performance metrics are similar, the underlying learned representations differ. Synth + real training aligns predictions more closely with the baseline, though the Jaccard index still diverges, indicating influence from synthetic data.

4.4. Analysis of Minimal-Change Data

4.4.1. Qualitative Results

In addition to the data samples in the methodology section, we sampled two more examples from each of the three datasets. Due to space constraints, we present samples from the Stanford Cars [32] dataset here; additional samples are shown in Appendix. As shown in Figure 5, the generated data meets our minimal-change criteria.

4.4.2. Distribution Analysis

We calculated the FID score [41] between generated and real data per class as a measure of distribution similarity. Additionally, we used CLIP Score, Dino Score, and LPIPS for each synthetic-real data pair as similarity measures, noting these provide only qualitative insights.

Figure 6 shows feasible samples resemble ID data due to their similarity to the real training set. However, as dis-

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

Table 4. The average DINO score, CLIP score and LPIPS scores calculated between generated synthetic image and corresponding real images for three datasets. F means feasibility.

cussed in Section 4.3, color and texture modifications do not improve model performance, likely because they disrupt the original distribution. For instance, white is dominant in our aircraft dataset, and adding multiple "red" or "yellow" samples shifts color proportions, creating distribution shifts. Even within matching colors, slight variations (e.g., different shades of red) can alter the distribution.

This trend is also reflected in metrics in Table 4: feasible data is generally closer to real data than infeasible data. While CLIP and Dino Scores are similar across settings due to their insensitivity to fine-grained details, LPIPS captures subtle differences. However, as feasible data is derived from real images, neither feasible nor infeasible data scores closely match real data.

4.4.3. Scaling the Number of Training Images

All previous experiments used a ratio of five synthetic images per real image. However, similar to prior studies [26], the synthetic-to-real data ratio can influence results. We conducted a scaling experiment on the AirC [40] dataset (and on the Pets [44] dataset, where results showed minimal variation). For AirC, we used all real images and varied the synthetic data from a 1:1 to a 5:1 ratio.

Results reveal a nonlinear trend with scale increases, though turning points vary across settings. For color and texture settings, peak performance slightly exceeds the baseline, suggesting that when synthetic data diverges significantly from real data, the benefits of OOD data may be limited to a narrow range.

4.5. Ablation Study

In Figure 8, we ablate the expanded mask, which preserves the spatial relationship between the object and background. Without this adjustment, the generated image often displays a "floating" effect, where the object appears unnaturally integrated into its environment.

5. Conclusion

In this work, we use our VariReal pipeline to generate minimal-change ID and OOD samples, so that we may investigate the essentialness of synthetic data feasibility towards generating classifier training data. By systematically altering background, color, and texture attributes, we create

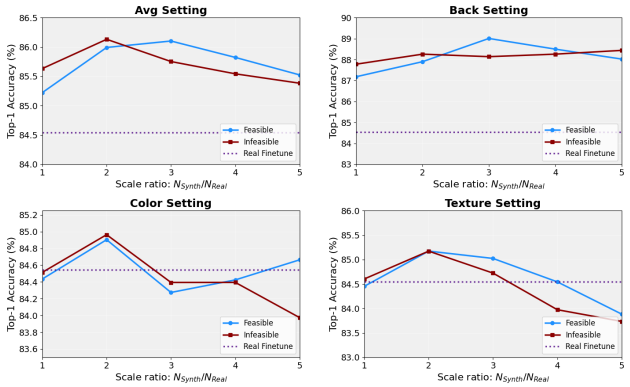

Figure 7. The scaling experiment results for the FGVC-Aircraft [40] dataset are shown for background, color, and texture settings. The horizontal axis represents the scale factor for synthetic images relative to real images. Here, the total real image training set is used, with scale factors ranging from 1 to 5.

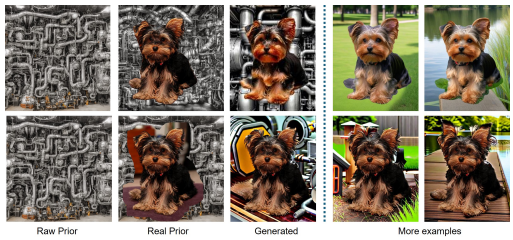

Figure 8. The ablation study for the usage to expand object mask for background edition setting. We show the real generated prior background on the left, and then present the different combined image with real and prior image.

feasible and infeasible scenarios across three fine-grained datasets. We evaluate the generated data by LoRA finetune a CLIP model for classification. Although previous literature assumes the importance of removing OOD data, we show that feasibility in terms of background, color, and texture may not significantly impact classification accuracy when training with the generated data. Furthermore, mixing ID and OOD samples also produces similar results. On the other hand, proxy metrics such as FID, CLIP score, DINOv2 score, and LPIPS show that ID data is closer to the original dataset than the infeasible data; from this, we may reconsider using these metrics as proxies for the effectiveness of synthetic training data for classification.

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. 4, 1, 2

[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. Bria 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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

[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 [best of the web]. *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] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in neural information processing systems*, 36:79024–79034, 2023. 2, 5, 6, 7, 1

[14] 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

[15] Chengjian Feng, Yujie Zhong, Zequn Jie, Weidi Xie, and Lin Ma. Instagen: Enhancing object detection by training on synthetic dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14121–14130, 2024. 2

[16] 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

[17] 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

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

[19] 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. 2

[20] 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

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[22] 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

[23] 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

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[25] 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. 3, 5

[26] Chip Huyen. Data distribution shifts and monitoring, 2022. https://huyenchip.com/2022/02/07/data-distribution-shifts-and-monitoring.html#data-shift-types. 8

[27] 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, 5

[28] 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

[29] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015. 6, 5

[30] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[32] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5, 6, 7

[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] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 1, 2, 5, 6

[35] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7817–7826, 2024. 1, 3

[36] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: 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] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and L Repaint Van Gool. Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471. 3, 4

[39] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 3, 5, 6

[40] 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

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

[42] 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

[43] 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

[44] 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, 7, 8, 2, 3

[45] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21783–21794, 2024. 3

[46] 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

[47] 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

[48] 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. 1, 2, 3, 6

[49] 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

[50] 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. 3

[51] Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8011–8021, 2023. 2

[52] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023. 2

[53] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18381–18391, 2023. 3

[54] 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

[55] 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, 6

[56] 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

[57] 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, 6

[58] 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. 3, 4, 5, 6

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9026–9036, 2024. 3

[61] 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

Does Feasibility Matter? Understanding In- and Out-of-Distribution Data Impact in Synthetic Training Datasets for Classification

Supplementary Material

6. Broader Impacts and Limitations

Our VariReal generation pipeline focuses on creating feasible and infeasible image pairs for downstream tasks but has the potential to extend to other applications. It provides a robust method for modifying backgrounds, colors, and textures in arbitrary input prompts and real images. For instance, it can be used in image editing tasks requiring background, color, and texture changes while preserving other areas. VariReal could also serve as a dataset generation pipeline to fine-tune Stable Diffusion models for text-guided image editing, enabling precise modifications based on prompts.

This work highlights the importance of feasibility in classification tasks, showing that strict adherence to the real data domain for backgrounds, colors, and textures is unnecessary. Text-guided synthetic data generation can prioritize style and significant object descriptions. VariReal can also be used for data augmentation, demonstrating that augmenting both feasible and infeasible backgrounds improves classification performance, unlike ALIA [13], which focuses only on feasible backgrounds.

Our approach primarily targets datasets with clear foreground and background and focuses on classification tasks due to its minimal-change setting. Future work could explore its applicability to more diverse and larger datasets, as well as general tasks. Due to resource limitations, we tested three attributes (background, color, texture) but additional attributes, such as lighting, could be investigated for feasibility. Developing a unified method for single-step minimal changes across multiple attributes would further enhance scalability and applicability.

7. Other Image Editing Methods

As shown in Figure 1, we compare our VariReal method with InstructPix2Pix [6] and FPE [35]. To ensure fairness and maximize the advantages of each model, we follow the original usage guidelines. For FPE, we retain the aspect ratio through resizing and padding to the specified size, and we use the original model training or best-performing prompts. For instance, FPE employs prompts like `"a [CLS] in the [ATTRIBUTE] background"` for background changes and `"a [ATTRIBUTE] [CLS]"` for color or texture edits. The [CLS] here is the corresponding class names, while the [ATTRIBUTE] represents the feasible or infeasible prompts generated as Section 3.2.1. Similarly, InstructPix2Pix uses prompts such as `"put it in [ATTRIBUTE] background"` for background changes and `"make it a [ATTRIBUTE] aircraft"` for color and texture modifications. We conducted multiple experiments and selected the best-generated images for comparison.

8. Method Details

8.1. Guidance Prompt

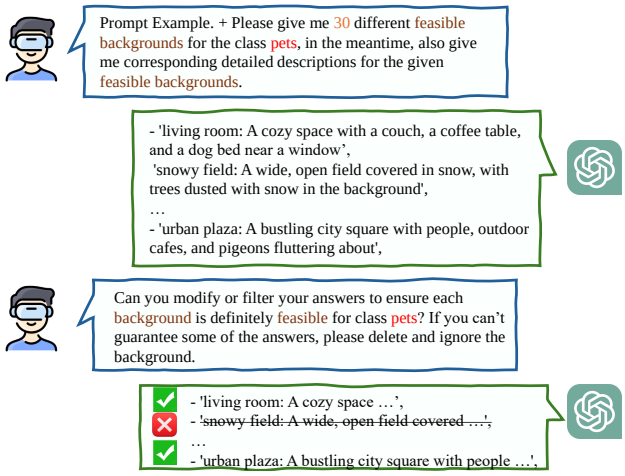

Figure 9. The generated attributes(prompt words) and self-filtering process using ChatGPT-4 [1].

As detailed in Section 3.2.1 and shown in Figure 9, the prompt generation process includes initial prompt generation and preliminary checks. The following prompt is used for preliminary checks:

```
"Can you modify or filter your
answers to ensure each
[background/color/texture] is
definitely [feasible/infeasible]
for class [CLASS]? Please delete
and ignore some of the answers
if you can't guarantee them."
```

For example, "snowy field" is not a feasible background for the pets class in the initial generation results and is filtered out by ChatGPT. To ensure feasible attributes align with the training set, we manually check the existing backgrounds, colors, and textures in the training data and remove those absent from it. Table 1 shows the acceptance ratio at each stage.

	Background						Color(Per CLS)						Texture					
	Pets		AirC		Cars		Pets		AirC		Cars		Pets(Per CLS)		AirC		Cars	
	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF
Raw output	50	70	50	70	50	70	10	10	10	10	10	10	8	50	30	50	15	70
Auto-filtering	47	64	36	68	44	67	6∼7	8∼9	7∼8	8∼9	7∼8	8∼10	7	42	25	46	12	64
Manual-filtering	43	50	22	50	31	50	5	5	5∼8	5∼6	5	5	5	27	24	44	7	57
Final Accept Rate	0.86	0.714286	0.44	0.71429	0.62	0.71429	0.5	0.5	0.5∼0.8	0.5∼0.8	0.5	0.5	0.625	0.54	0.8	0.88	0.467	0.814

Table 5. The number of prompts which are generated initially by LLM, after self-filtering and manual-filtering for each specific settings and some datasets. The Pets, AirC, Cars refer to our experimental dataset introduced in 4.1.

Specifically, for the guidance prompt generation, We use ChatGPT [1] to generate feasible or infeasible attributes (prompt words), which are then combined into a final prompt using our template: "a photo of a [CLS]", as shown in Figure 9. An example of generated attributes is the following, where the placeholders [ATTRIBUTE] represents the feasible/infeasible background, color, or texture, and [CLASS] represents a specific class.

Prompt Example. *"Task: As an AI language model, generate [Attribute] where the given class of objects typically exists ('feasible') and where they absolutely cannot exist ('unfeasible'). For each [Attribute], provide a one-sentence description detailing its visual appearance. You should adhere to the specified criteria.*

Criteria:

1. Unique [Attribute]: Ensure each listed [Attribute] is distinct and not synonymous with others provided.
2. Empty List Handling: If no unfeasible backgrounds can be identified, use 'EMPTY' to denote this.
3. Format Requirement: Answers must be formatted as a Python list, following the structure shown in the 'Answer' section of the 'Example'.

Positive Example:

- **Object Class:** [CLASS]
- **Question:** Provide five different [Attribute] for the object class, each accompanied by a concise visual description.
- **Answer:**
  - ...

Negative Examples:

- The answers are not acceptable as follows:
  - ...
- **Reasons:** ...

Question: Please give me [NUMBER] different [Attribute] for the class [CLASS]; in the meantime, also give me corresponding detailed descriptions for the given [Attribute].

Here we also give one specific example for generating feasible and infeasible background for Oxford Pets dataset [44] after replacing the placeholders in the above template.

Task: As an AI language model, generate backgrounds where the given class of objects typically exists ('feasible') and where they absolutely cannot exist ('unfeasible'). For each background, provide a one-sentence description detailing its visual appearance. The description should be vivid and adhere to the specified criteria.

Criteria:

1. **Feasible Backgrounds:** Identify environments where the object class naturally occurs in the real world.
2. **Unfeasible Backgrounds:** Identify environments where the object class cannot naturally or logically be present. Avoid fantastical or scientifically impossible scenarios (e.g., "inside a sun").
3. **Unique Backgrounds:** Ensure each background is distinct and does not overlap in meaning with others.
4. **Empty List Handling:** If no unfeasible backgrounds can be identified, use 'EMPTY' to denote this.
5. **Format Requirement:** Responses must be formatted as a Python list, following the structure provided in the 'Example' section.

Positive Example:

1. **Object Class:** Dog
2. **Question:** Provide five different unfeasible backgrounds for a dog, each accompanied by a concise visual description.
3. **Answer:**
   (a) 'underwater coral reef: A vibrant underwater scene filled

with colorful corals, schools of fish, and shimmering light filtering through the water surface.'
   (b) 'volcano crater: A rugged, rocky landscape with molten lava, steam vents, and an eerie red glow from the molten rock below.'
   (c) 'deep space station: A sterile, futuristic interior filled with advanced technology, floating objects, and a view of the infinite void of space outside.'
   (d) 'airplane cockpit: A confined, high-tech space with multiple control panels, screens, and a view of the clouds through the windshield.'
   (e) 'desert dunes: A vast, arid landscape with rolling sand dunes, scorching heat, and sparse vegetation under a blazing sun.'

Negative Examples:

1. **The following answers are not acceptable:**
   (a) 'industrial furnace room: A high-temperature environment with large furnaces used for metal smelting, filled with intense heat and noise.'
   (b) 'operating theater: A sterile room in a hospital where surgeries are performed, requiring a clean and controlled environment.'
2. **Reasons:**
   (a) Responses are not in a proper Python list format (e.g., ['', '', ..., '']).
   (b) Descriptions should focus on specific visual elements (e.g., objects, colors, lighting) instead of abstract concepts like "unsuitable for pets."
   (c) Example descriptions should include more visual details, e.g., "a large furnace with workers and glowing red-hot objects."

Question:

1. Please generate 20 different feasible and unfeasible backgrounds, respectively, for the class 'pets.'
2. Additionally, provide detailed visual descriptions for each background.

By using the prompts describe above, we also select some generated attributes (prompt words) to replace the placeholder in the prompt template. Due to space limitations, we provide up to five attributes as an example for the Oxford Pets [44] dataset. The following lists the feasible attributes:

Feasible Prompt Word Examples from Pets. Background:

- **suburban backyard:** A grassy area with a wooden fence, a few trees, and a doghouse in one corner.
- **city park:** A green space with open fields, walking paths, and other people walking their dogs.
- ...
- **rural countryside:** Rolling hills with grazing cows, wooden fences, and a distant farmhouse.
- **patio:** A stone patio with outdoor furniture, potted plants, and a view of the garden.

Color:

- **Abyssinian:** ruddy, blue gray, silver, fawn, fawn.
- **American Bulldog:** white, brindle, brown, fawn, brown.
- **American Pit Bull Terrier:** blue gray, fawn, black, white, brown.
- ...
- **Wheaten Terrier:** wheaten, golden, wheaten, wheaten, golden.
- **Yorkshire Terrier:** blue gray, tan, black, gold, tan.

Texture:

- **Abyssinian:**
  - ruddy ticked coat: warm ruddy brown fur with black ticking throughout.
  - sorrel coat: light reddish-brown fur with coppery tones.
  - blue coat: soft blue-gray fur with warm undertones.
  - fawn coat: light cream-colored fur with a gentle rose tint.
  - chocolate ticked coat: rich chocolate fur with lighter ticking.
- ...
- **Yorkshire Terrier:**
  - steel blue and tan coat: long,

silky fur in steel blue with tan points.
- black and tan coat: shiny black fur with tan points.
- golden tan coat: long fur in a rich golden tan color.
- blue and gold coat: dark blue fur with golden tan accents.
- silver and tan coat: light silver fur with warm tan points.

The following gives us the infeasible attributes examples:

Infeasible Prompt Word Examples from Pets.

Background:

- **space station:** A high-tech interior with floating objects, control panels, and a view of Earth through a window.
- **deep sea:** A dark, underwater environment with bioluminescent creatures and no sunlight.
- **volcano interior:** A fiery landscape with flowing lava, molten rocks, and intense heat.
- ...
- **mars surface:** A barren, reddish landscape with rocks, dust, and no signs of life.

Color:

- **Abyssinian:** purple, blue, pink, orange, neon green.
- **American Bulldog:** purple, pink, blue, green, yellow.
- **American Pit Bull Terrier:** purple, green, blue, orange, pink.
- ...
- **Wheaten Terrier:** green, purple, blue, yellow, pink.
- **Yorkshire Terrier:** green, purple, blue, yellow, orange.

Texture:

- **elephant skin texture:** characterized by thick, rough, and wrinkled surfaces, with deep creases.
- **wood grain:** parallel grooves and rings resembling tree bark, with a natural flow pattern typically seen in wooden planks.
- ...
- **metallic scales:** small, shiny scales arranged in an overlapping pattern.

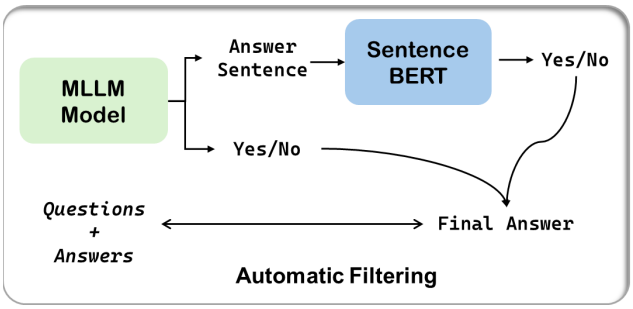

Figure 10. The automatic filtering process using a MLLM model to filter the generated images using pre-defined qustions to check certain aspect for the generated image and ground truth answers.

8.2. Automatic Filtering

As introduced in Section 3.2.3, we present the filtering questions for background, color, and texture changes. These checks ensure that the generated attributes align with the text prompt. For background attributes, we also verify if the foreground objects are feasible within the given background. Using placeholders for each background, color, texture prompt, object class, and feasibility information, we formulate questions based on the following filtering question template.

Background-related questions:

- **Question 1:** Is the object in the image located in the [BACKGROUND] environment? *Choices:* ['yes', 'no'] *Answer:* 'yes'
- **Question 2:** Does the image background represent [BACKGROUND]? *Choices:* ['yes', 'no'] *Answer:* 'yes'
- **Question 3:** Does the [BACKGROUND] look feasible for the [CLS]? *Choices:* ['yes', 'no'] *Answer:* 'yes' if [FEASIBLE] else 'no'
- **Question 4:** Is it possible for the [CLS] in this image to exist in the real world with its background? *Choices:* ['yes', 'no'] *Answer:* 'yes' if [FEASIBLE] else 'no'

Note: The placeholder [CLS] represents the current class name, [BACKGROUND] represents the target background being generated, and [FEASIBLE] denotes its feasibility.

If we change the color and texture, we use the following questions:

Color and Texture-related questions:

- **Question 1:** Does the image show a [COLOR/TEXTURE] [CLS]? *Choices:* ['yes', 'no'] *Answer:* 'yes'

- **Question 2:** Is the [COLOR/TEXTURE] feasible for the [CLS]? *Choices:* ['yes', 'no'] *Answer:* 'yes' if [FEASIBLE] else 'no'

Note: The placeholders retain similar meanings as above, where [COLOR/TEXTURE] indicates the current target appearance being generated.

9. Implementation Details

We provide additional implementation details for VariReal generation in Table 6. For instance, noise strength is a key parameter for the SDXL Inpainting model [39], and the strength of the IP-Adaptor [55] conditions ControlNet [58]. Different datasets and the generation of feasible vs. infeasible datasets often vary in difficulty, so we use dataset-specific parameters.

Following the approach in DataDream [27] for classification tasks, we experiment with different learning rates and weight decay. Specifically, we use a batch size of 64, AdamW [29] as the optimizer, and a cosine annealing scheduler. Table 7 details the CLIP [34] fine-tuning parameters. For learning rates and weight decay, we search within a range and select the best-performing configuration as the final result. Additionally, we fix the number of iterations as mentioned in Section 4.1, with the table specifying iteration counts for each dataset.

10. Qualitative Examples

We provide additional qualitative examples to demonstrate the generation quality of our VariReal method. One additional example from the Oxford Pets [44], FGVC Aircraft [40], and Stanford Cars [32] datasets is included, along with one randomly selected example across these datasets.

Figure 11 shows the Abyssinian pet generation results, where our VariReal method produces more detailed backgrounds, such as "active war zone." Figure 12 presents a Spitfire aircraft sample, illustrating snow in the background "arctic tundra landing strip." Figure 13 features a BMW X3 SUV 2012 example. All color and texture changes align with the text prompt requirements. Finally, Figure 14 provides randomly selected examples from the three datasets for further visualization.

Parameters	Back.						Color						Texture					
	Pets		AirC		Cars		Pets		AirC		Cars		Pets		AirC		Cars	
	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF	F	IF
Guidance Scale for SDXL Inpainting [39]	40		7.5		7.5		12		12		30		12		8		30	
Guidance Scale for ControNet [58]	-						7.5						7.5					
Strength for SDXL	0.99		0.95		0.9		0.3		0.8		0.85		0.3	0.3	0.65	0.3	0.65	0.3
IP-Adptor [55] Strength	-						0.7		0.4		0.4		0.2	0.5	0.65	0.4	0.65	0.4
Inference Step for SD	20						-						15					
Inference Step for SDXL Inpainting	30						20						20					
Inference Step for ControlNet	-						30						30					
Mask dilated factor/alpha factor	120		50		25		0.3		0.6		0.6		0.5	0.4	0.5	0.65	0.65	0.65

Table 6. The detailed generation parameters for VariReal. We introduce the parameters for feasible and infeasible settings of three dataset respectively.

HyperParameters	**lamda**	**lr**	**Min_lr**	**Weight decay**	**Warm up steps**	**CLIP LoRA rank**	**CLIP LoRA alpha**
Values	0.5	{1e-3,5e-4,1e-4,5e-5,1e-5}	1e-08	1e-03, 1e-4, 5e-5	5% total iterations	16	32
HyperParameters	**Training bs**	**Test bs**	**Train iterations**	**Val iterations**	**Data augmentation**		
Values	64	8	Pets:20700/AirC:72000/Cars:91840	1/70 Train iterations	random resized crop, random horizontal flip, random color jitter, and random gray scale		

Table 7. The hyper-parameter details for CLIP [34] model fine-tuning.

Figure 11. Qualitative results of the class Abyssinian from Oxford Pets dataset [44], created the same as Figure 5.

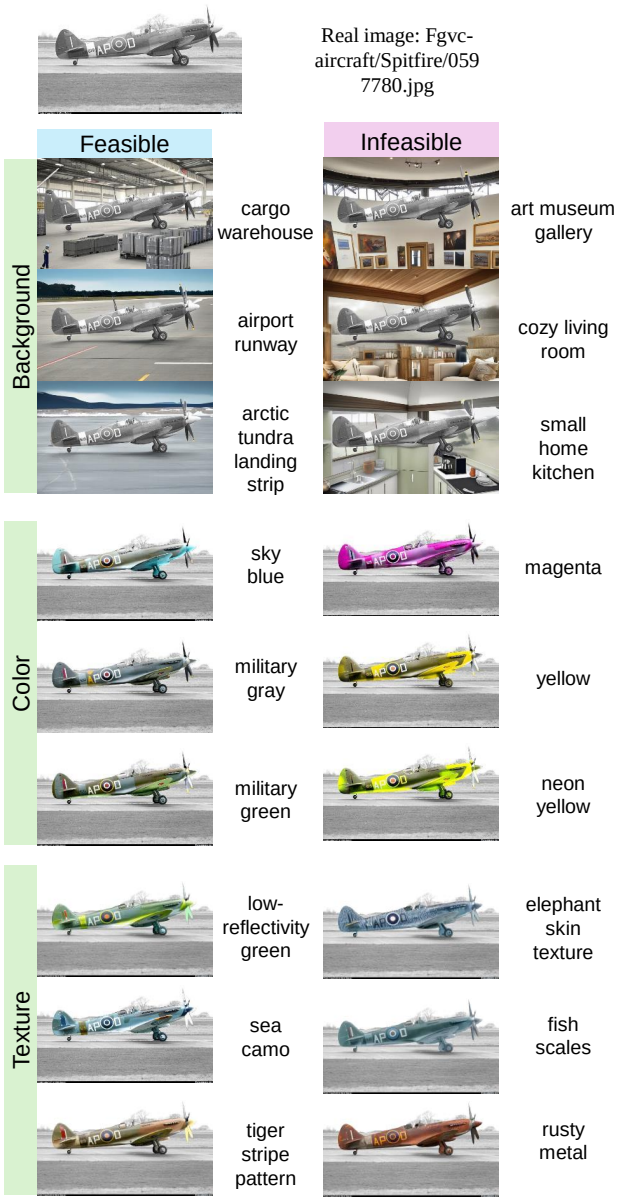

Figure 12. Qualitative results of the class Spitfire from Fgvc-Aircraft dataset [40], created the same as Figure 5.

Figure 13. TQualitative results of the class BMW X3 SUV 2012 from Stanford Cars dataset [32], created the same as Figure 5.

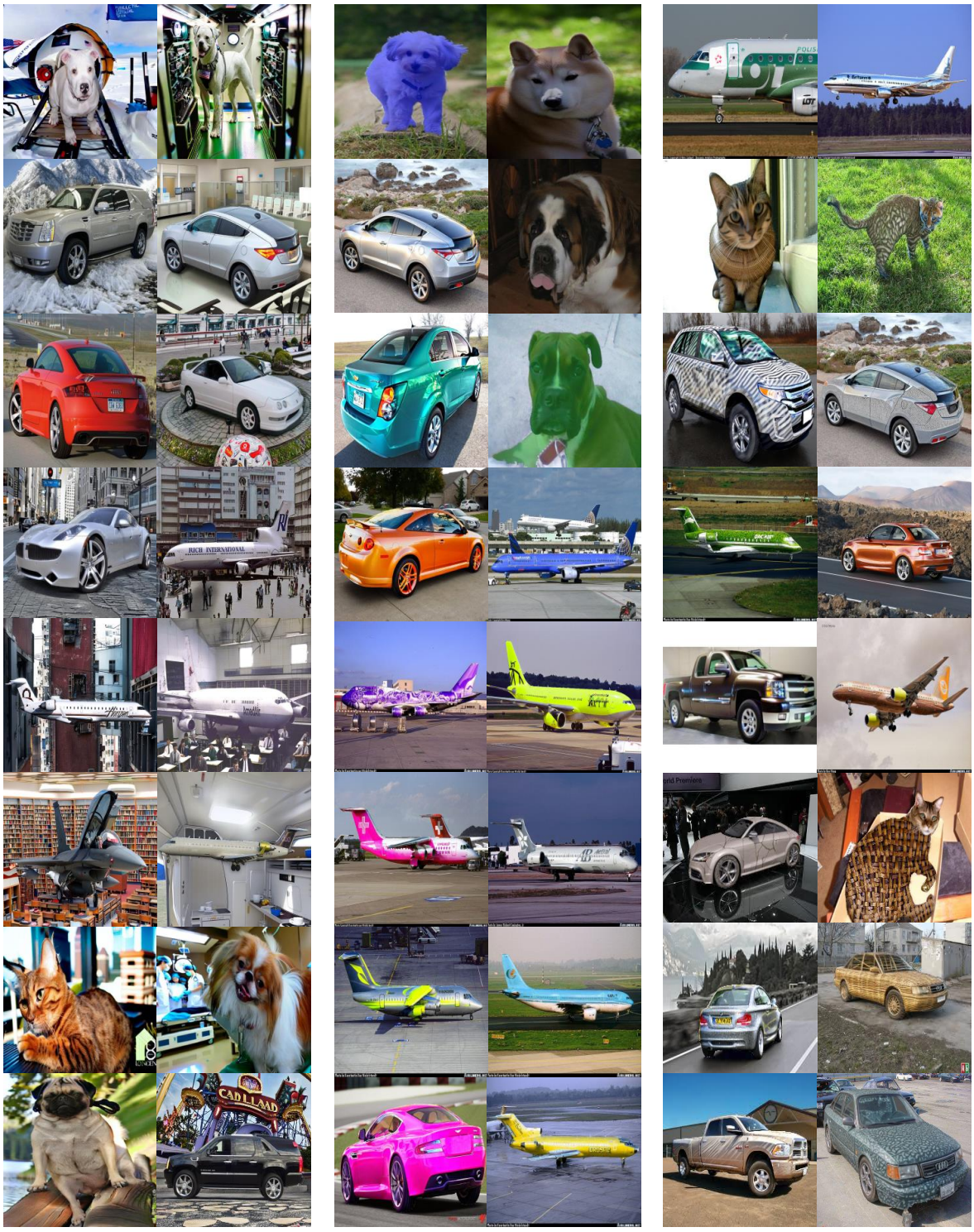

Figure 14. Randomly selected generated samples across three datasets and feasibility attributes are shown. For visualization purposes, all images are resized to the same dimensions.