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

Supplementary Material

In this supplementary material, we first discuss the broader impacts and limitations of our analysis in Sec. A. Experimental setups for our method are provided in Sec. B, and configurations for other image editing models are detailed in Sec. C. Sec. D describes our method in detail, including guidance prompts and automatic filtering. Additionally, we present background-specific classification results on the WaterBird [14] dataset in Sec. E. Further classification result analysis is provided in Sec. F, followed by additional qualitative examples and user study details in Sec. G. Finally, an ablation study of the VariReal pipeline is included in Sec. H.

A. Broader Impact and Limitation

Our VariReal pipeline focuses on generating feasible and infeasible image pairs for downstream tasks, with potential applications beyond classification. It offers a robust method for modifying backgrounds, colors, and textures in both prompts and real images, making it suitable for image editing tasks that require precise changes while preserving other regions. VariReal can also serve as a dataset generation tool to fine-tune Stable Diffusion models for text-guided image editing, enabling targeted modifications. Additionally, it supports data augmentation, showing that augmenting both feasible and infeasible backgrounds improves classification performance—unlike ALIA [14], which only uses feasible backgrounds.

We define feasibility as alignment with real-world plausibility. For instance, feasible car colors are those officially released by manufacturers. Rare custom paint jobs—such as a ”cyan” Audi RS 4 Convertible 2008—are excluded, as they do not reflect typical production offerings. Within our scope, such extreme cases are treated as infeasible settings.

Our approach targets datasets with clear foreground-background separation and focuses on classification tasks under minimal-change settings. Although we strive to preserve structure, slight deviations—particularly in color and texture edits—are sometimes unavoidable due to current image editing limitations. In the meantime, our method requires adjusting hyperparameters when modifying images to meet specific requirements. We believe advances in image editing techniques will make our experimental setup more effective and easier to implement. Due to resource constraints, we explored only three attributes (background, color, texture), but future work could extend to others, such as lighting. Developing a unified method for minimal, single-step edits across multiple attributes would enhance scalability and enable broader application to diverse datasets and tasks.

B. Implementation Details

We provide additional implementation details for VariReal in Table 5. Key parameters include noise strength for the SDXL Inpainting model [40] and conditioning strength for IP-Adapter [58] with ControlNet [61]. Due to varying difficulty across datasets and between feasible and infeasible generation, we use dataset-specific settings.

Following DataDream [28], we tune learning rates and weight decay for classification tasks. We use a batch size of 64, AdamW [38] optimizer, and a cosine annealing scheduler. Table 6 lists the CLIP [47] fine-tuning parameters. Learning rates and weight decay are selected from a predefined range based on validation performance. The number of training iterations is fixed as described in Sec. 4.1, with dataset-specific counts provided in the table.

C. Other Image Editing Method Setups

As shown in Figure 1, we compare VariReal with InstructPix2Pix [6] and FPE [34]. To ensure fairness and leverage each model's strengths, we follow their original usage guidelines. For FPE, we maintain aspect ratio via resizing and padding, and use the original training setup with recommended prompts—e.g., `"a [CLS] in the [ATTRIBUTE] background"` for background changes and `"a [ATTRIBUTE] [CLS]"` for color or texture edits, where [CLS] denotes the class name and [ATTRIBUTE] refers to feasible or infeasible prompts from Sec. 3.2.1. InstructPix2Pix uses prompts like `"put it in [ATTRIBUTE] background"` for background changes and `"make it a [ATTRIBUTE] aircraft"` for foreground edits. We conducted multiple trials and selected the best outputs for comparison.

D. Method Details

D.1. Guidance prompt

As detailed in Sec. 3.2.1 and shown in Figure 7, the prompt generation process includes initial prompt generation and preliminary checks.

Specifically, we use GPT-4 [1] to generate feasible or infeasible initial attributes (prompt words), which are then combined into a final prompt using our template: ”a photo of a [CLS]”, as shown in Figure 7. These initial attributes are then preliminarily checked by:

	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 4. 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.

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 [40]	40		7.5		7.5		12		12		30		12		8		30	
Guidance Scale for ControNet [61]	-						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 [58] 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 5. 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-3, 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 6. The hyper-parameter details for CLIP [47] model fine-tuning.

```
"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, "deep cave" is not a feasible background for the pets class in the initial generation results and is filtered out by GPT-4. 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.

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.

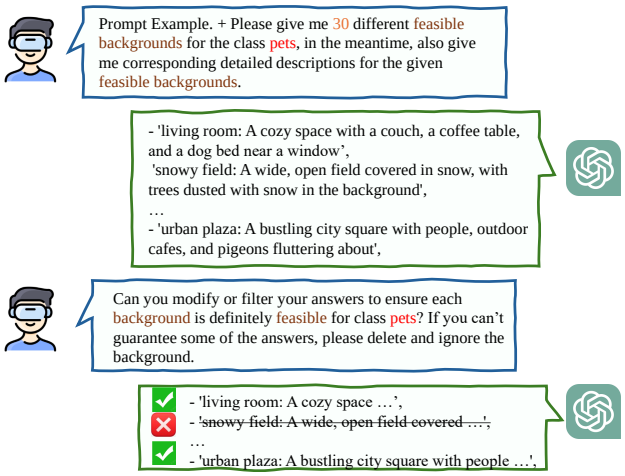

Figure 7. The generated attributes(prompt words) and self-filtering process using ChatGPT-4 [1].

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 [46] after replacing the placeholders in the above template in Figure 8.

By using the prompts described 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 [46] dataset. Some generated feasible and infeasible prompt words can be found in Figure 9.

D.2. Automatic filtering

As introduced in Sec. 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.

We show an example process for the automatic filtering in the Figure 14.

E. WaterBird Experiment Details

In this section, we present detailed experimental results for the WaterBird [14] dataset under background modification settings, as shown in Table 7. Notably, infeasible background edits improve performance by 5.8 percentage points in the synthetic-only setting and 1.6 percentage points in the real + synthetic setting.

F. Classification Results Analysis

In Sec. 4.2.1, we analyze mixing the feasible and infeasible data has no clear impact on classification tasks but some times will help the model learn complementary knowledge. We evaluate prediction correctness per test sample to compare knowledge learned by models trained under different settings. To measure whether one model's correctly predicted set is a subset of another's, we use: Inclusion Coefficient $= \frac{|A \cap B|}{|A|}$, with values closer to 1 indicating greater overlap. Additionally, we quantify the overlap of correctly predicted samples between models using the Jaccard index:$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$, where A and B, where A and B represent correct predictions from two training configurations.

The Inclusion matrix in Figure 15 shows no subset relationship exists between model predictions.Notably, the feasible-only and infeasible-only settings labeled with dashed lines yield the lowest Jaccard scores, indicating minimal similarity.

> *Observation: The feasible and infeasible data lead the model to learn different directions, while they achieve very similar performance.*

G. Qualitative Examples and User Study

We provide additional qualitative examples to demonstrate the generation quality of our VariReal method. One additional example from the Oxford Pets [46], FGVC Aircraft [41], 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. Finally, Figure 16 provides randomly selected examples from the three datasets for further visualization. The instruction for the questionnaire is shown in Figure 17.

	R	S	WaterBirds [14] F	WaterBirds [14] IF
0-shot			79.0	
Back.		✓	86.6	92.4
	✓	✓	92.9	94.5
Real	✓		85.7	

Table 7. The top-1 performance using the full training set and synthetic data, with training setups including synthetic-only and synth. + real data. The attribute of experimented dataset WaterBirds [14] is background. All results use synthetic images set to five times the number of real images.

Figure 18 presents examples of correctly and incorrectly classified feasibility cases. More detail can be seen by zooming into the figures. For infeasible texture modifications, failure cases often like infeasible texture change of *fish scale* or *brick wall*, which are fine-grained and hard to represent clearly. In such cases, the output may only reflect the color rather than the intended texture, so human evaluators will classify these to the incorrect cases. Another source of error involves implausible object-background combinations—for example, a "flying aircraft in an airplane hangar" shown in the lower part of Figure 18.

For the naturalness criterion, some images—such as those in Figure 19 where the feasible color is changed from red to gray or white—receive lower scores, as the resulting colors appear less natural.

H. Ablation Study

We ablate the mask dilation step introduced in Sec. 3.2.2, which helps maintain spatial coherence between objects and backgrounds. Without mask dilation, generated images often exhibit a "floating" effect shown in Figure 20, where objects appear unnaturally integrated into their backgrounds.

Task and Criteria

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.

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.

In Context Learning 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.'
4. **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.'
5. **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.

Figure 8. A specific prompt example used for background prompt words generation of Oxford Pets [46] dataset.

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.
- ...
- **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.
- ...
- **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.

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.
- ...
- **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.
- ...
- **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 9. Final accepted prompt word examples for Oxford Pets [46].

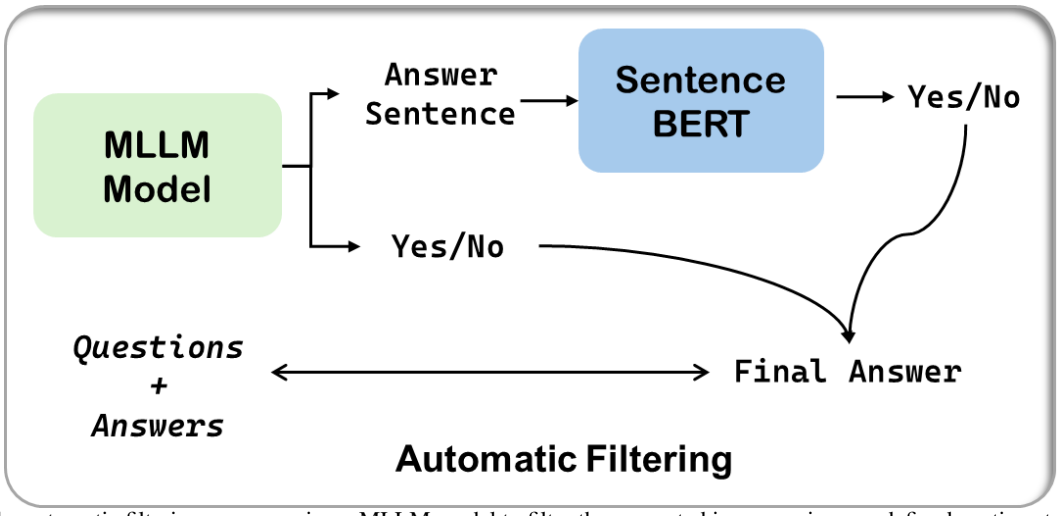

Figure 10. The automatic filtering process using a MLLM model to filter the generated images using pre-defined qustions to check certain aspect for the generated image and ground truth answers.

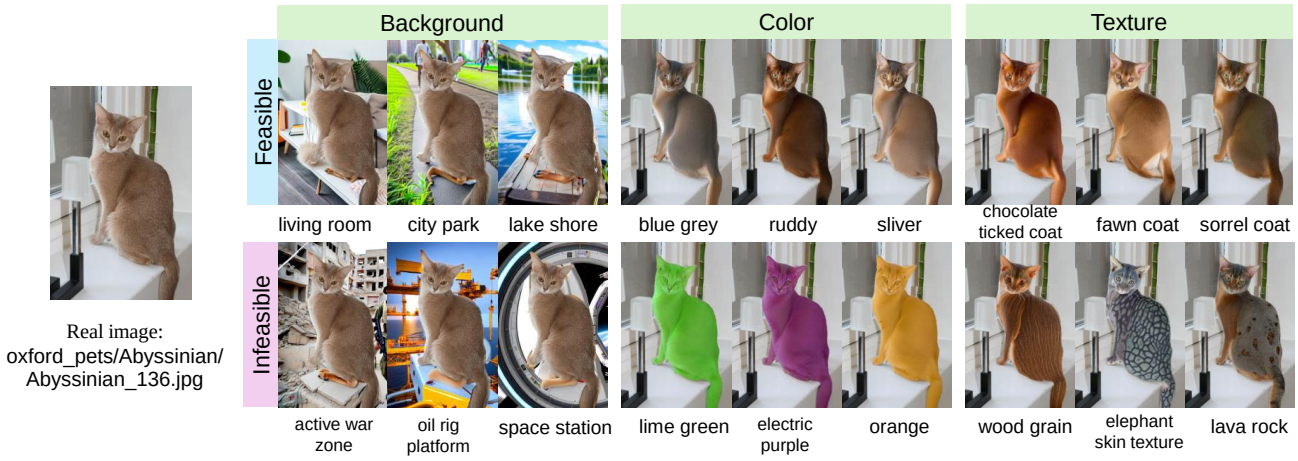

Figure 11. Qualitative results of the class Abyssinian from Oxford Pets dataset [46], as a supplement for Figure 4.

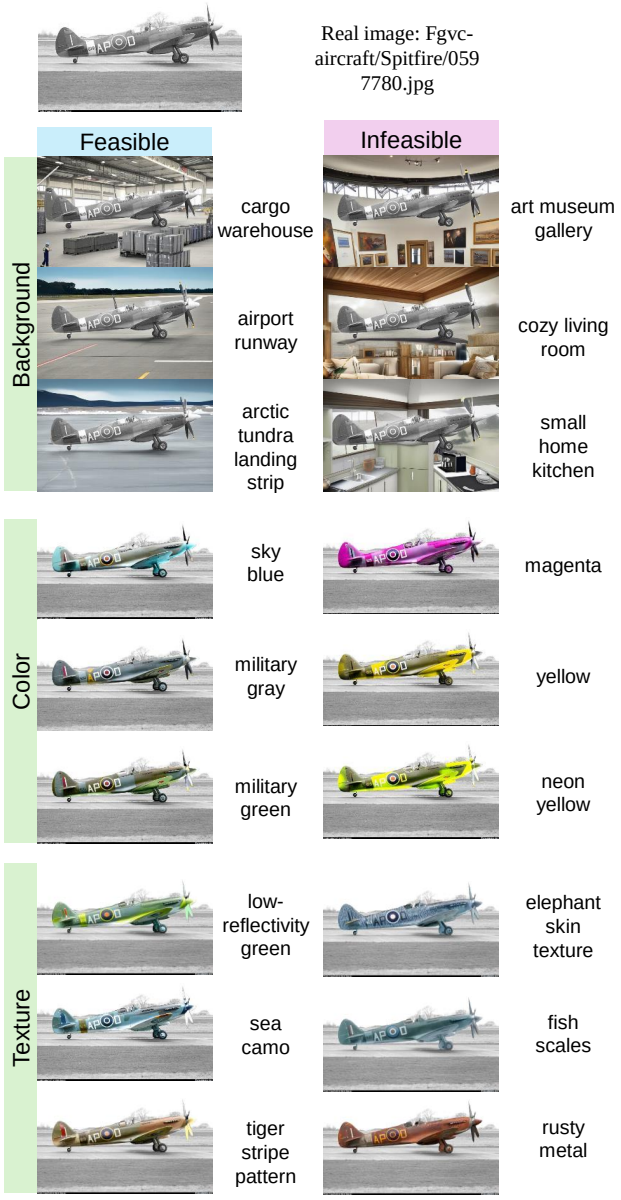

Figure 12. Qualitative results of the class Spitfire from Fgvc-Aircraft dataset [41], as a supplement for Figure 4.

Figure 13. TQualitative results of the class BMW X3 SUV 2012 from Stanford Cars dataset [32], as a supplement for Figure 4.

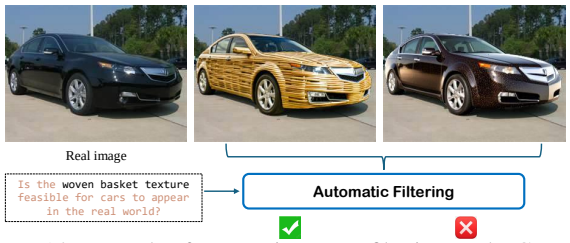

Figure 14. Example of automatic texture filtering on the Cars [32] dataset.

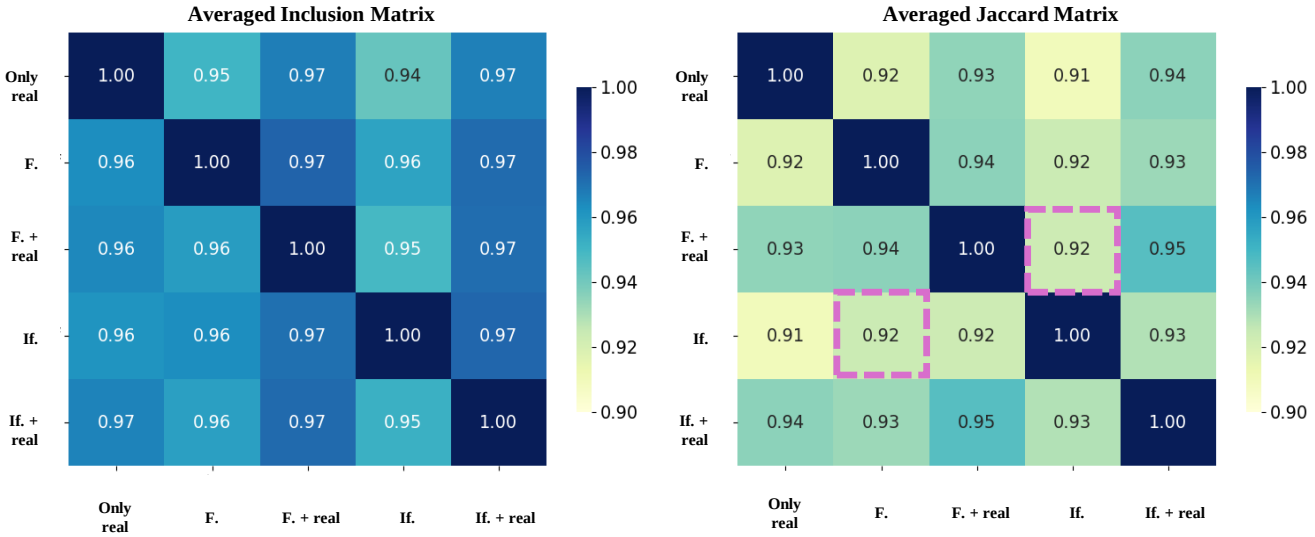

Figure 15. The averaged Inclusion and Jaccard index matrix for three editing settings across three datasets. ”f” = feasible, ”if” = infeasible, ”real” = training with real images.

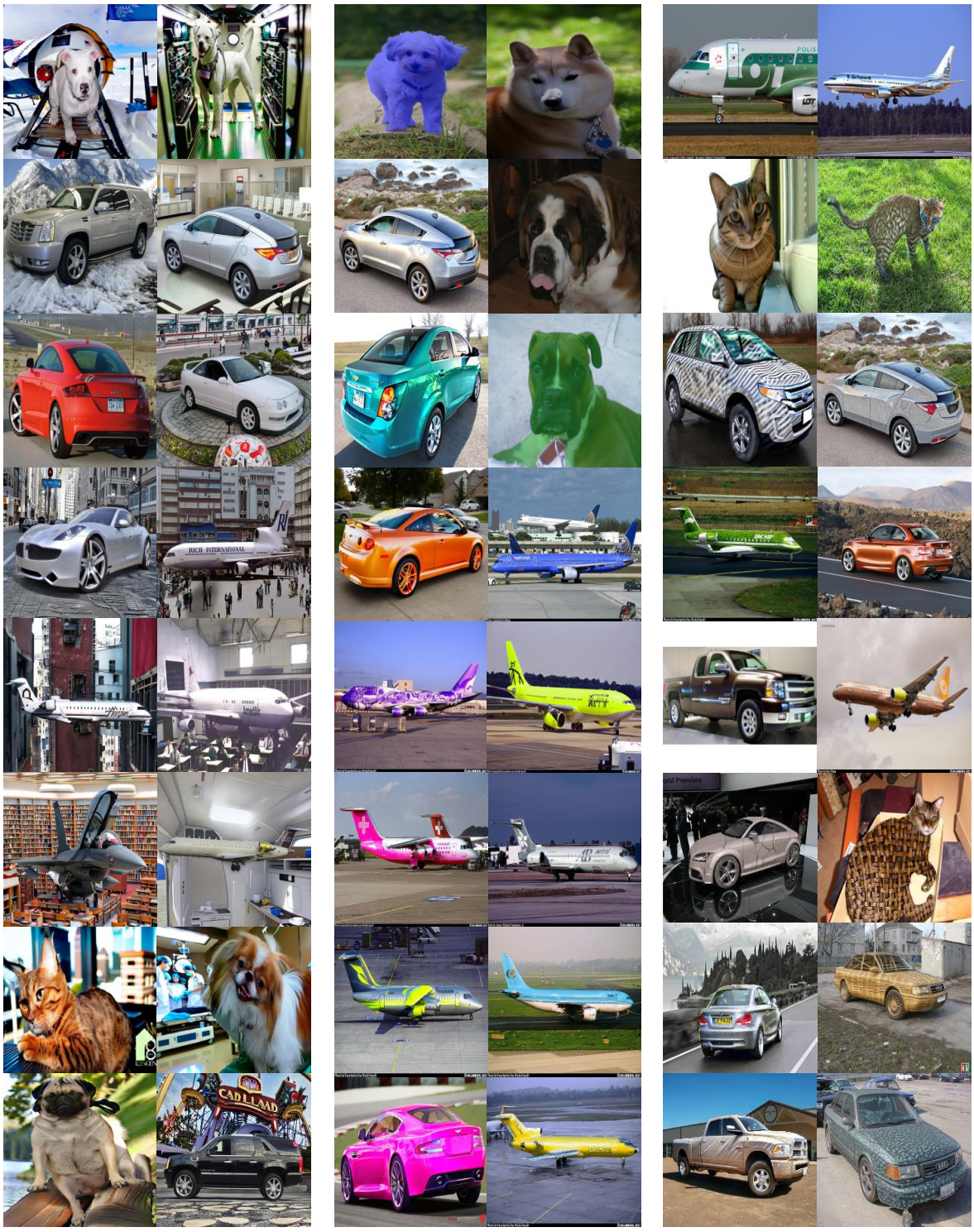

Figure 16. Randomly selected generated samples across three datasets and feasibility attributes are shown. For visualization purposes, all images are resized to the same dimensions.

In this questionnaire, you will be shown an image and instructions that specify some edits to be made to the image (we call this "edit instruction"). You will also be shown the edited image. Your task is to evaluate the edited image's correctness/feasibility/naturalness.

- The edited image is judged to be **feasible** if attributes assigned to an object in the synthetic image could realistically exist in the real-domain with high probability; On the contrary, it is infeasible.
- Please rate the **naturalness** of the image subjectively, with 1 being the lowest score and 5 being the most natural.

Please see the examples below to understand the task better: (Left: original; Right: edited)

Example 1(Back):

Is the edited image feasible?

☐ YES ☒ No

Please rate the **naturalness** of the image: 4.5

Example 2(Color):


Is the edited image feasible?

☒ YES ☐ No

Please rate the **naturalness** of the image: 4

Example 3(Texture):


Is the edited image feasible?

☐ YES ☒ No

Please rate the **naturalness** of the image: 4

Please answer the questions below to the best of your knowledge. Thank you for your careful attention to detail and your valuable contribution!

Figure 17. Instructions for feasibility and naturalness generated images human study.

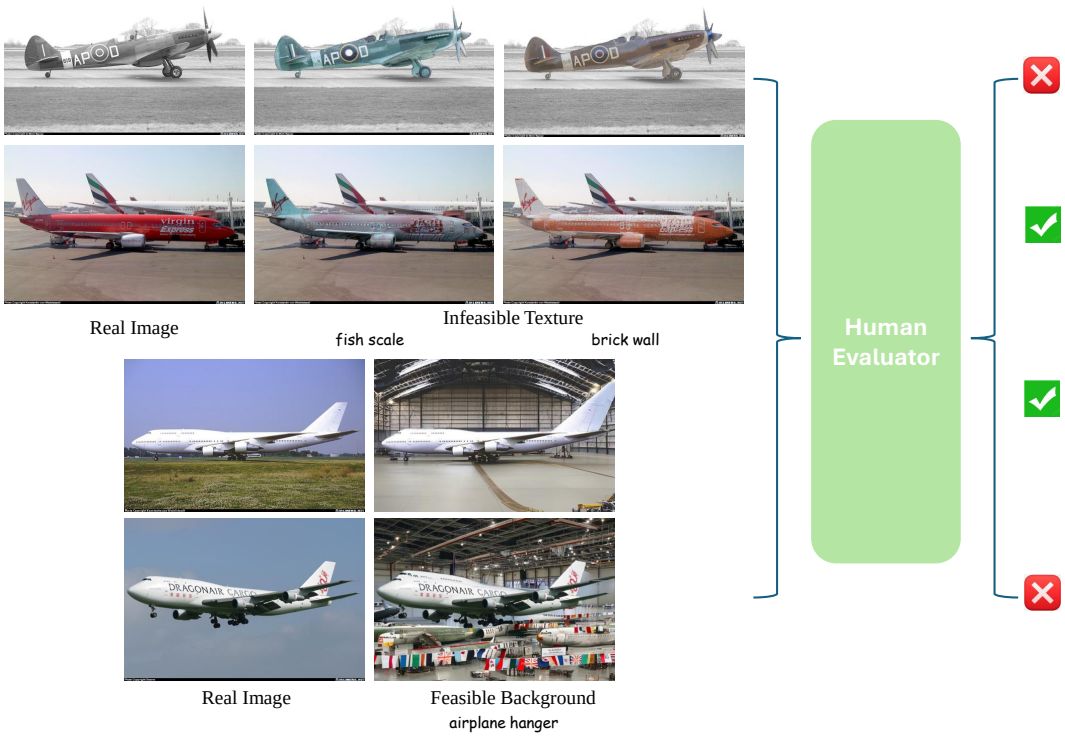

Figure 18. Examples assessed as incorrect feasibility by human evaluators, including unclear fine-grained textures (e.g., ”fish scale”) and implausible object-background combinations (e.g., a flying aircraft inside a hangar).

Figure 19. Examples assessed by human evaluators as having lower naturalness, often due to unnatural color modifications or unrealistic visual appearance.

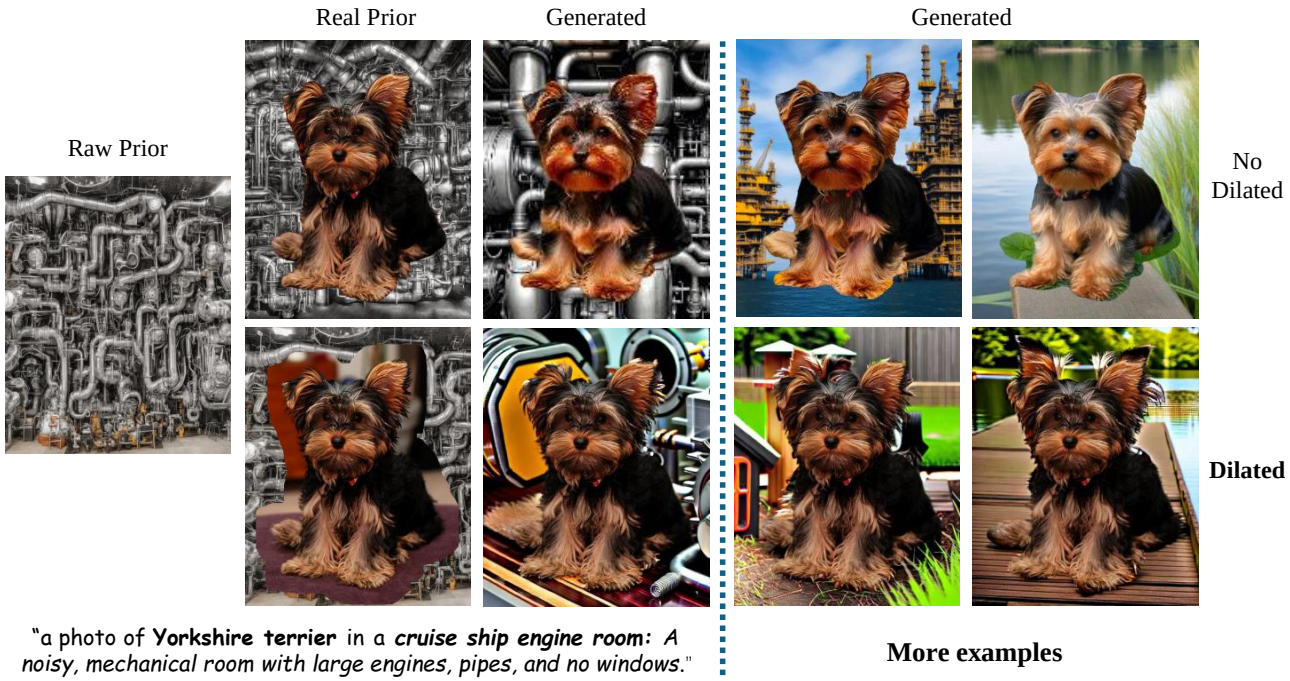

Figure 20. The ablation study for the usage to expand object mask for background edition setting. We show the real generated prior background on the left, and then present the different combined image with real and prior image.