Appendix

Table of Contents

A. Discussions

A.1. Intuitive Difference Between `AugCLIP` and Directional CLIP Similarity

The major difference between CLIP_{dir} and `AugCLIP` stems from the flexibility of the evaluation standard. Unlike directional CLIP similarity that relies on the fixed standard of 'Target text - Source text' as shown in Fig. 5(a), our metric `AugCLIP` estimates the contextual difference between the source and the target to flexibly adjust the evaluation standard as '$M(\text{Target, Source}) - \text{Source}$'. More specifically, Fig. 5(b) shows flexibility of `AugCLIP`. On the left side, the direction of the evaluation standard (red line) is close to the direction of 'Target - Source' as CLIP_{dir} does. On the right side, the direction of the evaluation standard (red line) does not align with 'Target - Source', indicating that the evaluation standard inclines toward preserving the source image, rather than modifying the image into the semantic space of the target text.

In real-world evaluation cases, this is an important property since some tasks might require a small amount of editing that transforms only part of the image whereas some tasks focus on editing the whole image with large modifications. Evaluation metrics should be flexibly applicable to all such cases, deciding the evaluation standard to focus on preservation or modification on a case-by-case basis. However, regardless of the editing context requiring different modification or preservation levels, existing metrics blindly apply the same standard that overly focuses on preservation or modification.

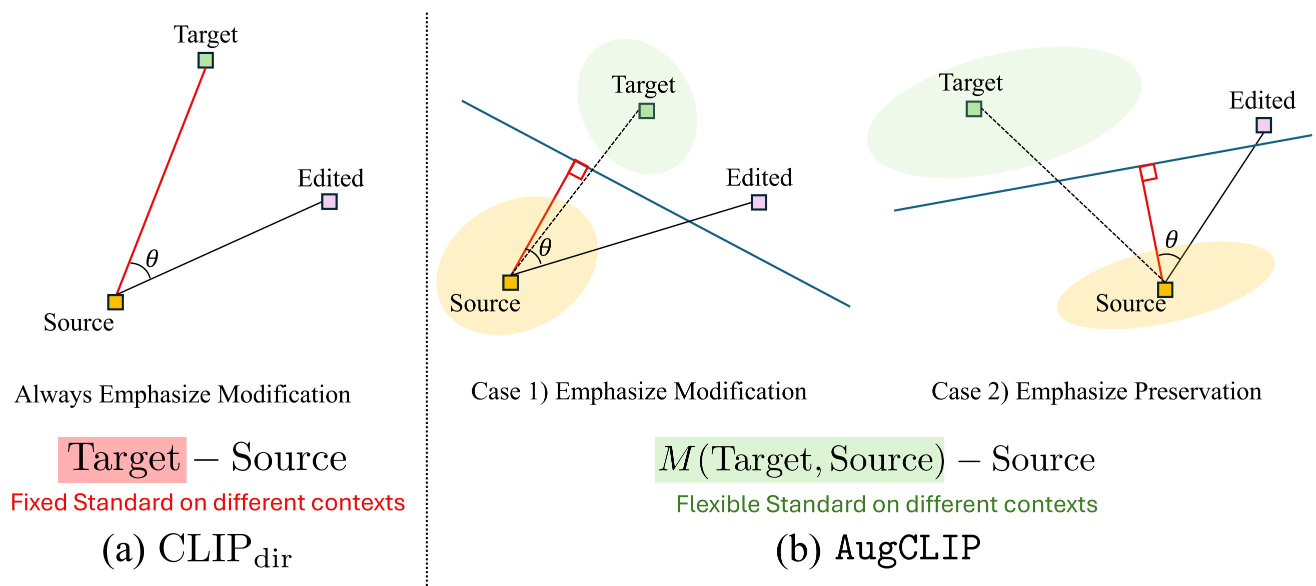


Figure 5. Difference between CLIP_{dir} and `AugCLIP`. The red line indicates the evaluation standard and the black line indicates the change in image from source to target. Both CLIP_{dir} and `AugCLIP` measure the quality of the edited image according to the corresponding red lines. In **(b)**, the green and yellow circles indicate the distribution of target and source attributes, respectively, and dotted black lines indicate the evaluation standard of CLIP_{dir}.

A.2. Randomness of Descriptions

As discussed in Sec. 6, the randomness of extracted descriptions does not impact the alignment with human judgment (merely 0.0197 in variance). In addition to this observation, we provide description samples for the target text, varying the seed over five configurations in Fig. 6, 7. The examples show that generated descriptions have overlapping semantics across different seeds. For example, Fig. 7a across five seeds describe black and white stripes, with small textual differences but identical in terms of semantics. This is observed across all four examples of the target text, proving that randomness in the description extraction process does not create semantically distinct samples, thus the evaluation results of `AugCLIP` are robust. the difference across random seeds is almost negligible since source descriptions are directly extracted from the source image caption.

Seed 7	"A dog is in a sitting position with its hind legs on the ground.", "A dog is with its front legs positioned forward, resting on the ground.", "A dog is having its back straightened and upright as it sits.", "A dog is looking slightly upward as its head is elevated in a sitting posture.", "A dog is with its tail positioned behind or curled to the side while sitting.", "A dog is with its ears perked up or relaxed, depending on the breed, while sitting."
Seed 83	"A dog is in a sitting position with its hind legs bent and resting on the ground.", "A dog has its front legs positioned forward, supporting its weight while sitting up straight.", "A dog is looking forward or slightly upwards instead of standing tall with its body extended.", "A dog has a relaxed posture, with its body lower to the ground compared to a standing position.", "A dog has its tail positioned behind it, possibly resting on the ground or curled up to the side.", "A dog has its ears in a neutral position, possibly perked up or relaxed depending on its mood while sitting."
Seed 87	"A dog is positioned with its hind legs on the ground and its front legs resting on its thighs.", "A dog is sitting upright with its back straight and head held high.", "A dog is displaying a relaxed posture with its tail possibly curled around its body.", "A dog is not on its toes or standing on its legs, but rather has its paws flat on the ground.", "A dog is showing its belly more towards the ground rather than being elevated."
Seed 88	"A dog is positioned with its hind legs on the ground and its front legs resting on its thighs.", "A dog is in a relaxed posture with its body lower to the ground.", "A dog is displaying a sitting stance with its tail possibly curled behind it.", "A dog is likely to have its head held upright or slightly tilted forward, looking alert or content.", "A dog is showing its belly facing the ground instead of upright as in a standing position."
Seed 94	"A dog is in a sitting position.", "A dog is with its hind legs bent underneath its body.", "A dog is with its front legs straight and placed in front of its body.", "A dog is with its tail resting on the ground instead of lifted up.", "A dog is with its body weight shifted backward onto its hind quarters."

(a) Target text: Change a standing dog into a sitting dog.

Seed 7	"A box is open at the top, revealing the interior space.", "A box has its flaps or lid raised or removed, indicating it is no longer sealed.", "A box shows visible contents or emptiness inside, contrasting with a closed position.", "A box has its edges or corners slightly bent outward, suggesting movement from a closed to an open state."
Seed 83	"A box is open at the top, allowing access to the inside.", "A box has flaps that are lifted or tilted back instead of closed flat.", "A box is showing the interior, potentially with contents visible.", "A box is revealing the inner structure and material, contrasting with the exterior.", "A box is lacking a closed lid, indicating it is in an open position."
Seed 87	"A box is open at the top, revealing the interior space.", "A box has flaps that are lifted upwards instead of being closed down.", "A box is showing contents inside, which may include various items or empty space.", "A box has a visible gap at the top where the flaps no longer meet.", "A box is positioned such that one or more sides may be slightly ajar."
Seed 88	"A box is open at the top, revealing its interior.", "A box is tilted slightly to one side, creating a sense of depth and accessibility.", "A box is showing its flaps folded back, indicating it is no longer closed.", "A box is displaying an empty space inside, suggesting it could hold items.", "A box is having a wider perspective, allowing for a view of its inner walls."
Seed 94	"A box is open, revealing its interior contents.", "A box is missing its top lid, showing the inside.", "A box is positioned at an angle that showcases the inside.", "A box is slightly tilted to the side, indicating it is not fully closed.", "A box is showing visible gaps where the lid would normally fit.", "A box is displaying items inside it, making it clear it is open."

(b) Target text: Change a closed box into an open box.

Figure 6. Target Descriptions Generated with Five Random Seeds.

Seed 7

```
"A zebra is black and white striped.",
"A zebra has a more robust, stocky body compared to a horse.",
"A zebra has a shorter, more upright mane.",
"A zebra has larger, rounded ears.",
"A zebra has a more pronounced muzzle with a broader face.",
"A zebra has a more compact build with shorter legs than a horse.",
"A zebra has a distinctive, unique pattern of stripes that varies from individual to individual."
```

Seed 83

```
"A zebra is black and white striped.",
"A zebra has a long mane that stands upright.",
"A zebra has a stocky build with a rounded belly.",
"A zebra has large, rounded ears that are more prominent than a horse's.",
"A zebra has a shorter muzzle compared to a horse.",
"A zebra has a more compact body shape than a horse.",
"A zebra has distinctively shaped hooves that are more rounded.",
"A zebra has a thick tail with a tuft of hair at the end."
```

Seed 87

```
"A zebra is characterized by its distinctive black and white stripes.",
"A zebra has a more robust body structure compared to a horse.",
"A zebra has a mane that is more upright and bristly than that of a horse.",
"A zebra's ears are typically larger and more rounded than those of a horse.",
"A zebra has a shorter and stockier build than a horse, with a thicker neck.",
"A zebra's face is more elongated with a rounded muzzle compared to that of a horse.",
"A zebra has a different tail with a tuft of hair at the end, while a horse's tail is long and flowing."
```

Seed 88

```
"A zebra is black and white striped.",
"A zebra has a mane that is short and upright.",
"A zebra has a more rounded body shape compared to a horse.",
"A zebra has a larger head in proportion to its body.",
"A zebra has a tail with a tuft of hair at the end.",
"A zebra has more pronounced facial features, with a broader muzzle.",
"A zebra's ears are more rounded at the tips.",
"A zebra has distinctive curved hooves that are smaller than a horse's."
```

Seed 94

```
"A zebra is black and white striped.",
"A zebra has a mane that is short and upright.",
"A zebra has a stocky body and a rounded belly.",
"A zebra has large, rounded ears that are often pointed upwards.",
"A zebra has a long tail with a tuft of hair at the end.",
"A zebra has a more horse-like face, but with a broader muzzle.",
"A zebra's legs are typically shorter and sturdier than a horse's legs."
```

(a) Target text: Change a horse into a zebra.

Seed 7

```
"A cat is in a yawning pose with its mouth wide open.",
"A cat is showing its teeth and tongue during the yawn.",
"A cat is tilting its head slightly backward as it yawns.",
"A cat is exhibiting closed eyes or partially closed eyes while yawning.",
"A cat is displaying a relaxed body posture with a slightly extended neck."
```

Seed 83

```
"A cat is in a yawning position.",
"A cat is showing its teeth while yawning.",
"A cat is tilting its head slightly backward as it yawns.",
"A cat is having its eyes half-closed during the yawn.",
"A cat is extending its front legs and stretching its body while yawning."
```

Seed 87

```
"A cat is positioned with its mouth open in a yawning expression.",
"A cat is showing its teeth as it yawns.",
"A cat is tilting its head slightly back while yawning.",
"A cat is displaying its tongue during the yawn.",
"A cat is having its eyes partially closed as it yawns.",
"A cat is having its ears in a relaxed position while yawning."
```

Seed 88

```
"A cat is sitting upright with its body relaxed.",
"A cat is yawning with its mouth wide open and tongue visible.",
"A cat has its eyes closed tightly while yawning.",
"A cat has its ears slightly back as it yawns.",
"A cat's whiskers are more pronounced and spread apart during the yawn."
```

Seed 94

```
"A cat is in a yawning posture with its mouth open wide.",
"A cat is showing its teeth and tongue as it yawns.",
"A cat is tilting its head slightly backward while yawning.",
"A cat is appearing more relaxed and less alert than when sitting.",
"A cat is displaying droopy eyelids, suggesting drowsiness during the yawn."
```

(b) Target text: Make a cat yawn.

Figure 7. Target Descriptions Generated with Five Random Seeds.

A.3. Optimization Objective

In Tab. 9, we test three variants of optimization objective for deriving the classifier function $g(x) = \mathbf{w}^T x + b$. Since source and target descriptions encoded into CLIP are separable by a simple linear function, we set the hyperplane as $\mathbf{w}^T x + b = 0$. Moreover, in optimizing the parameters $\mathbf{w}$ and b, we select the hinge loss objective with L2 regularization, namely SVM objective.

Reason for Choosing SVM Objective. We have emonstrate in Tab. 9, that this SVM objective is the best option. Moreover, the reasons for choosing this objective are threefold. First, the SVM objective is compatible with various source and target cases. Since SVM allows for misclassification to some extent with the slack variable, it extends to editing cases where the source image and target text do not show distinct discrepancies. Second, support vectors maximize the margin, which is defined as the distance between the hyperplane and the closest support vectors on either source and target descriptions. Third, the influence of data points far from the margin is reduced, since only the closest points (support vectors) determine the decision boundary. This helps avoid bias caused by outlying data points, ensuring the hyperplane is not skewed toward either the preservation or modification side.

Comparing Optimization Objectives. In order to test if choosing other hyperplane optimization objective impacts the level of alignment with human judgment, we compare latent discriminant analysis (LDA), logistic regression, and linear SVM objective in finding a separating hyperplane. More specifically, the objective functions are as follows,

$$\text{LDA} : \frac{\det(N_S(\mu_S - \mu)(\mu_S - \mu)^T + N_T(\mu_T - \mu)(\mu_T - \mu)^T)}{\det(\sum_{\boldsymbol{s}_i \in \mathcal{D}_S}(\boldsymbol{s}_i - \mu_S)(\boldsymbol{s}_i - \mu_S)^T + \sum_{\boldsymbol{t}_j \in \mathcal{D}_T}(\boldsymbol{t}_j - \mu_T)(\boldsymbol{t}_j - \mu_T)^T)}$$

$$\textsc{Logistic} : -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\sigma(g(x_i))) + (1 - y_i)\log(1 - \sigma(g(x_i)))\right]$$

$$\text{SVM} : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y_i \cdot g(x_i))$$

where the optimization targets to find $\mathbf{w}$ and b that minimizes the given objective functions. A total of N pairs of data points (x, y) is employed in the optimization process, where x in these objective functions signifies the CLIP-encoded source or target attributes with the corresponding label $y \in \{-1, 1\}$.

In Tab. 9, we report the human judgment alignment score $\boldsymbol{s}_{2\text{AFC}}$ and ground truth test accuracy $\mathbf{Acc}_{\text{Both}}$. Linear SVM shows the best $\boldsymbol{s}_{2\text{AFC}}$ and $\mathbf{Acc}_{\text{Both}}$, except for CelebA, and the difference between optimization functions do not largely impact the final judgment. Therefore, `AugCLIP` is robust to the optimization of hyperplane.

Table 9. Comparison on difference optimization function for the classifier.

	CelebA	EditVal	DreamBooth	TEdBench	MagicBrush	Average Misc.
LDA	**0.884**	0.827	0.821	0.545	0.863	3.37%
Logistic	0.849	0.830	0.821	0.550	0.866	1.38%
Linear SVM	0.883	**0.831**	**0.857**	**0.570**	**0.889**	**1.35%**

Misclassification Rate. Our metric `AugCLIP` first encodes the source and target attributes into CLIP space, denoted as $\mathcal{D}_S$ and $\mathcal{D}_T$ respectively. We then employ a linear function $g(x) = \mathbf{W}^T x + b$ to estimate a decision boundary that separates the source and target distribution. The average percentage of source and target attributes that are wrongly classified by the hyperplane across five benchmark datasets is reported in Tab. 9, denoted as 'Average Misc.'. We observe that simple linear decision function $g(x)$ shows a small misclassification rate, 1.35%, which signifies its ability to separate the source and target distributions. Specifically, linear SVM achieves the lowest misclassification rate, successfully distinguishing between source and target attributes. Given that the source image and target text may share visual similarities, the extracted source and target attributes cannot always be perfectly separable by a linear hyperplane (*e.g.*, when editing an orange to a tangerine, both the source and edited images share a round shape.). In such cases, these attributes are closely positioned in the embedding space and do not require complete separation. SVM's ability to manage overlapping factors more flexibly allows it to find a more accurate hyperplane, leading to superior performance.

UMAP Visualization. Additionally, we visualize that CLIP features of source and target attributes can be separated by a linear hyperplane in 2D projected space using UMAP [21] in Fig. 8, in which randomly chosen subset of TEdBench samples are plotted. 'S' represents source attributes encoded into CLIP, while 'T' represents the target attributes. The line signifies the linear hyperplane $g(x) = \mathbf{W}^T x + b = 0$ that separates the two classes, which are source and target. In `AugCLIP`, the linear hyperplane $g(x) = 0$ is d-dimensional following the original dimension of CLIP, but to visualize in Fig. 8, the dimension shrinks into $d = 2$ by UMAP fitting.

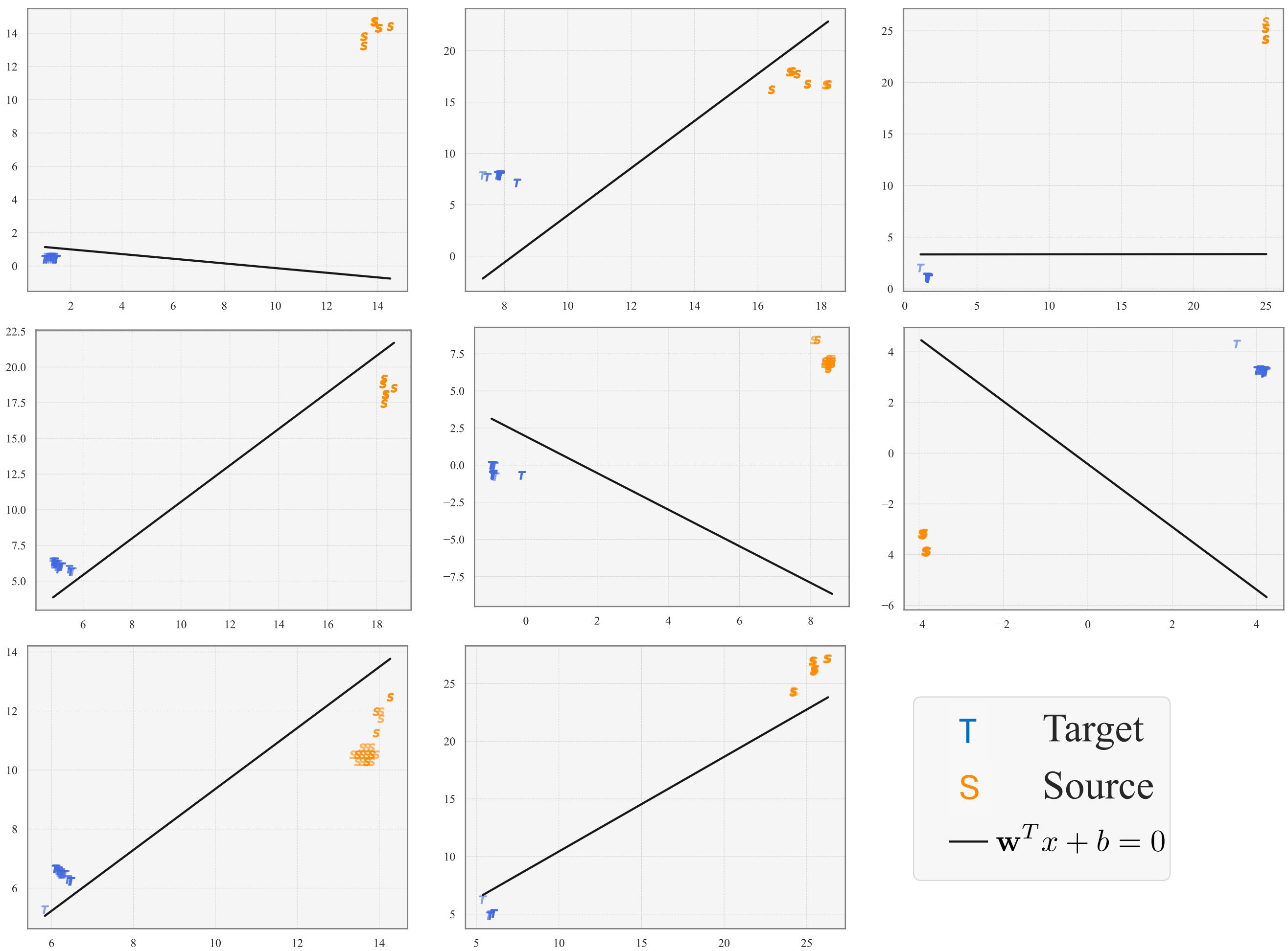


Figure 8. Visualization of Source and Target Attributes in 2D Space. S indicates the source attributes encoded into CLIP space and T indicates the target attributes. The separating hyperplane is a linear function $\mathbf{W}^T x + b = 0$.

A.4. Analyzing the Modification Vector

The goal of text-guided image editing is to apply a necessary transformation to the source image, preserving the original property as much as possible. Our evaluation process follows this protocol by estimating the modification vector $\mathbf{v}$ as a minimum modification that makes the source image look like a target text, ensuring that essential source attributes remain unchanged but the resulting edited image resembles the target text. In this section, we qualitatively analyze the effect of $\mathbf{v}$ on source and target attributes.

Figure 9. Effect of the Modification Vector v on Source Attributes. The source attributes are ranked based on the magnitude of change induced by the modification vector $\mathbf{v}$. Attributes at the top of the list exhibit the most significant adjustments toward the target text, indicating that these characteristics are considered important source attributes that should be modified in the given editing context. Conversely, attributes lower down the list are determined to remain intact in an ideal editing.

Effect of the Modification Vector v on Source Attributes Source attributes that need to be preserved should not be affected by the modification vector $\mathbf{v}$, while those requiring adjustment according to the target text should be altered. To demonstrate that $\mathbf{v}$ drives significant changes in the attributes requiring modification while minimally impacting those that should remain intact, we analyze several cases using the TEdBench and EditVal datasets, as shown in Fig. 9. We measure the difference in cosine similarity between each source attribute $\boldsymbol{s}_i$ and both the source image and the ideally edited image. Specifically, we calculate the increase in similarity between the ideal edited image $E(I_{\text{src}}) + \mathbf{v}$, and the source image $E(I_{\text{src}})$ as

$$\Delta \boldsymbol{s}_i = \texttt{cs}\Big(E(I_{\text{src}}) + \mathbf{v}, \boldsymbol{s}_i\Big) - \texttt{cs}\Big(E(I_{\text{src}}), \boldsymbol{s}_i\Big), \tag{11}$$

where $\boldsymbol{s}_i$ indicate the CLIP-encoded source attribute in $\mathcal{D}_S$. As demonstrated in Fig. 9, source attributes that need to be preserved exhibit only small changes in similarity (small $\Delta\mathbf{s}_i$), while attributes that need modification (large $\Delta\mathbf{s}_i$) show a significant change in similarity.

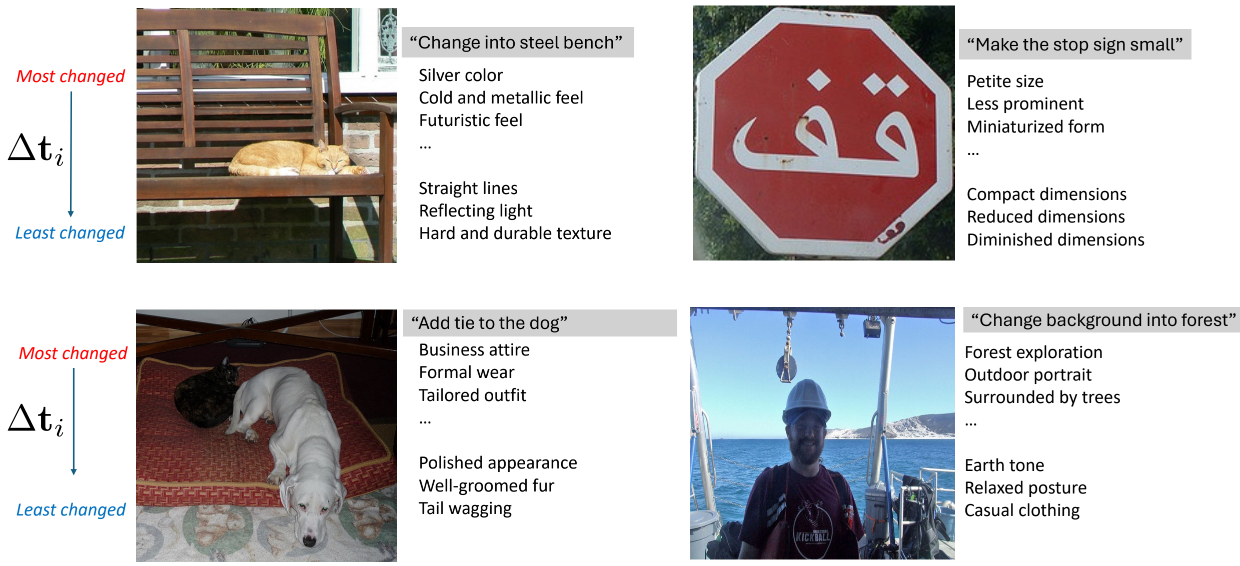


Figure 10. Effect of the Modification Vector v on Target Attributes. The target attributes are ranked based on the magnitude of change induced by the modification vector $\mathbf{v}$. Attributes at the top of the list are determined by `AugCLIP` standard to be key to modification aspects. Attributes lower on the list exhibit smaller adjustments, which are deemed less important by `AugCLIP` standard.

Effect of the Modification Vector v on Target Attributes We demonstrate how target attributes are affected by the modification vector $\mathbf{v}$. To demonstrate that $\mathbf{v}$ causes significant changes in the attributes that are central to make the image resemble the target text while having minimal impact on the rather peripheral attributes, we analyze several cases using the EditVal dataset, as shown in Fig. 10. The difference in cosine similarity between the source image and the ideally edited image is measured for each target attribute $\boldsymbol{t}$. Similar to the previous paragraph, we compare the increase of target attribute $\boldsymbol{t}_i$ in the ideal edited image, $E(I_{\text{src}}) + \mathbf{v}$, compared to the source image $E(I_{\text{src}})$ as

$$\Delta \boldsymbol{t}_i = \texttt{cs}\Big(E(I_{\text{src}}) + \mathbf{v}, \boldsymbol{t}_i\Big) - \texttt{cs}\Big(E(I_{\text{src}}), \boldsymbol{t}_i\Big), \tag{12}$$

where $\boldsymbol{t}_i$ indicate the CLIP-encoded target attribute in $\mathcal{D}_T$. As illustrated in Fig. 10, attributes essential for matching the target text display a significant increase in similarity, while secondary attributes experience only minor changes.

B. Algorithm

We provide the algorithm of `AugCLIP` in Python code style in the following block.

```python
# Step 0: Get CLIP features
src_img_feat, tgt_img_feat = CLIP(src_img), CLIP(tgt_img)
src_text_feat, tgt_text_feat = CLIP(src_text), CLIP(tgt_text)
src_desc_feat, tgt_desc_feat = CLIP(src_desc), CLIP(tgt_desc)

# Step 1: Compute importance weighting for each desc
src_dist = [src_img_feat, src_desc_feat]
tgt_dist = [tgt_img_feat, tgt_desc_feat]

src_weight, tgt_weight = compute_weight(src_dist, tgt_dist)
weight = [src_weight, tgt_weight]

# Step 2: Fit the classifier model
X = [src_dist, tgt_dist]
y = [-1] * src_dist.shape[0] + [1] * tgt_dist.shape[0]
svc_classifier.fit(X, y, sample_weight=weight)

# Step 3: Retrieve the hyperplane parameters
w = svc_classifier.coef_ # Hyperplane coefficients
b = svc_classifier.intercept_ # Hyperplane intercept

# Step 4: Compute the modification vector v by calculating the projection of src_img_feat onto the hyperplane
numerator = -(np.dot(w.T, src_img_feat) + b)
denominator = np.linalg.norm(w)**2
v = (numerator / denominator) * w

# Step 5: Calculate the alignment score
score = cosine_similarity(src_img_feat + v, tgt_img_feat)
```

C. Evaluation Details

C.1. Assets

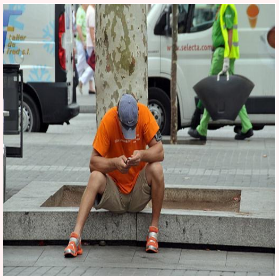
Table 10. Assets Employed in Our Experiments. List of pre-trained models, benchmark datasets, and metrics employed in this paper.

Category		Asset	URL
Benchmarks		CelebA [20]	https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
		TedBench [12]	https://github.com/imagic-editing/imagic-editing.github.io/tree/main/tedbench
		EditVal [1]	https://github.com/deep-ml-research/editval_code
		DreamBooth [26]	https://github.com/google/dreambooth
		MagicBrush [28]	https://github.com/OSU-NLP-Group/MagicBrush
Editing Models	TEdBench EditVal MagicBrush	InstructPix2Pix [2]	https://github.com/timothybrooks/instruct-pix2pix
		DiffEdit [5]	https://github.com/Xiang-cd/DiffEdit-stable-diffusion.git
		Prompt-to-Prompt [8]	https://github.com/google/prompt-to-prompt.git
		DDS [9]	https://github.com/google/prompt-to-prompt/blob/main/DDS_zeroshot.ipynb
		Imagic [12]	https://github.com/huggingface/diffusers/tree/main/examples/community#imagic-stable-diffusion
	CelebA	DiffusionCLIP [13]	https://github.com/gwang-kim/DiffusionCLIP
		Multi2One [14]	https://github.com/akatigre/multi2one
		Asyrp [17]	https://github.com/kwonminki/Asyrp_official
		StyleCLIP [23]	https://github.com/orpatashnik/StyleCLIP
	DreamBooth	Custom Diffusion [16]	https://github.com/adobe-research/custom-diffusion
		BLIP-Diffusion [18]	https://github.com/salesforce/LAVIS/tree/main/projects/blip-diffusion
		ELITE [27]	https://github.com/csyxwei/ELITE
Metrics		CLIP [24]	https://github.com/openai/CLIP
		LPIPS [29]	https://pypi.org/project/lpips/
		SC [13]	Implemented by the Authors

C.2. Source Description Generation

As described in Sec. 4.1, we prompt GPT-4V with the source image to first generate a detailed caption. Specifically, the prompt orders to focus on the visual and objective aspects of the image, disregarding any opinions or non-visual descriptions. With the generated caption, we provide an example of another source image caption and its parsed list of visual characteristics. With these examples, GPT-4V parses the source image caption into a Python list of visual characteristics. The example prompting result is shown as follows.

User



Describe the image in detail. Do not include opinions or non-visual aspects. Make the descriptions easy and intuitive.

GPT-4V

The image shows a person sitting on a low ledge. He is wearing an orange T-shirt and tan shorts. His legs are spread apart, and he appears to be focused on his phone, which he is holding with both hands. He has a gray cap on his head, and his athletic shoes are orange with gray accents. His posture is slightly hunched over as he engages with his device.

User

You are a helpful text-generation assistant. Given detailed textual descriptions about an image, your goal is to parse into specific visual attributes.
EXAMPLE INPUT: “The image features a large, dark-colored dog standing in a grassy field. The dog appears to be alert and attentive, possibly observing its surroundings. The grass is lush and green, providing a natural backdrop for the dog.”

EXAMPLE OUTPUT: [“A dog is large”, “A dog is dark-colored”, “A dog is standing”, “A dog is standing on a grassy field”, “A dog is alert”, “A dog is attentive”, “A dog is observing its surroundings”, “A grass is lush”, “A grass is green”, “A grass is providing a natural backdrop”]

Give me a Python list of visual characteristics included in the following sentences:
INPUT: “The image shows a person sitting on a low ledge. He is wearing an orange T-shirt and tan shorts. His legs are spread apart, and he appears to be focused on his phone, which he is holding with both hands. He has a gray cap on his head, and his athletic shoes are orange with gray accents. His posture is slightly hunched over as he engages with his device.”

GPT-4V

[“He is sitting on a low ledge”, “He is wearing an orange t-shirt”, “He is wearing tan shorts”, “He has his legs spread apart”, “He is focused on his phone”, “He is holding the phone with both hands”, “He has a gray cap on his head”, “He has athletic shoes”, “He has orange shoes with gray accents”, “He is slightly hunched over”]

C.3. Target Description Generation

The descriptions for the target text are generated by providing the target text and the source image caption. The prompt instructs the model to avoid repetition of similar visual attributes. The example depicts a case of changing a sitting person into a standing posture. The output is structured into a Python list of descriptions, starting with “Standing person”. From the target text “Change a man into Ironman,” the generated list of attributes includes descriptions of the suit, weapons, and armor. These also depict that Ironman takes heroic poses and appears in intense battles. The example prompting result is shown as follows.

User

You are a helpful text-generation assistant. Given a detailed textual description of an image, your goal is to parse it into specific visual attributes. If the visual attributes with similar meanings are repeated, only use one. For example, “A dog is large” and “A dog is big” are similar, so only use one.

EXAMPLE INPUT: In order to make a person into a standing posture, what visual attributes of the image should be changed? Answer in the format of “Standing person is ATTRIBUTE”.

EXAMPLE OUTPUT: [“Standing person has straight legs”, “Standing person is upright”, “Standing person is on their feet”, “Standing person is tall”]

INPUT: In order to make a person into Ironman, what visual attributes of the image should be changed? Answer in the format of “Ironman is ATTRIBUTE”.

GPT-4V

[“Ironman has powered exoskeleton suit”, “Ironman has a red and gold color scheme”, “Ironman has Arc reactor on chest”, “Ironman wears metallic armor”, “Ironman wears helmet with facial display”, “Ironman has repulsor technology on hands”, “Ironman has jet boots for flight”, “Ironman has advanced weapon systems”, “Ironman has a strong and muscular build”, “Ironman has heroic poses”, “Ironman appears in intense battles”]

C.4. User Study Details

Criteria)

1. *Realism*: The manipulated image should show high realism, aiming to appear as authentic as possible.
2. *Relevance to Text Prompt*: The manipulated image should be closely aligned with the accompanying target text
3. *Preservation of Source Image*: The manipulated image should preserve the original image's essence.

Example)

If a text prompt is "Change a dog into a cat",
then the color and the posture of the dog should be preserved while making the image look like a cat.

Instruction)

Choose Model A or Model B with the that better satisfies the criteria.

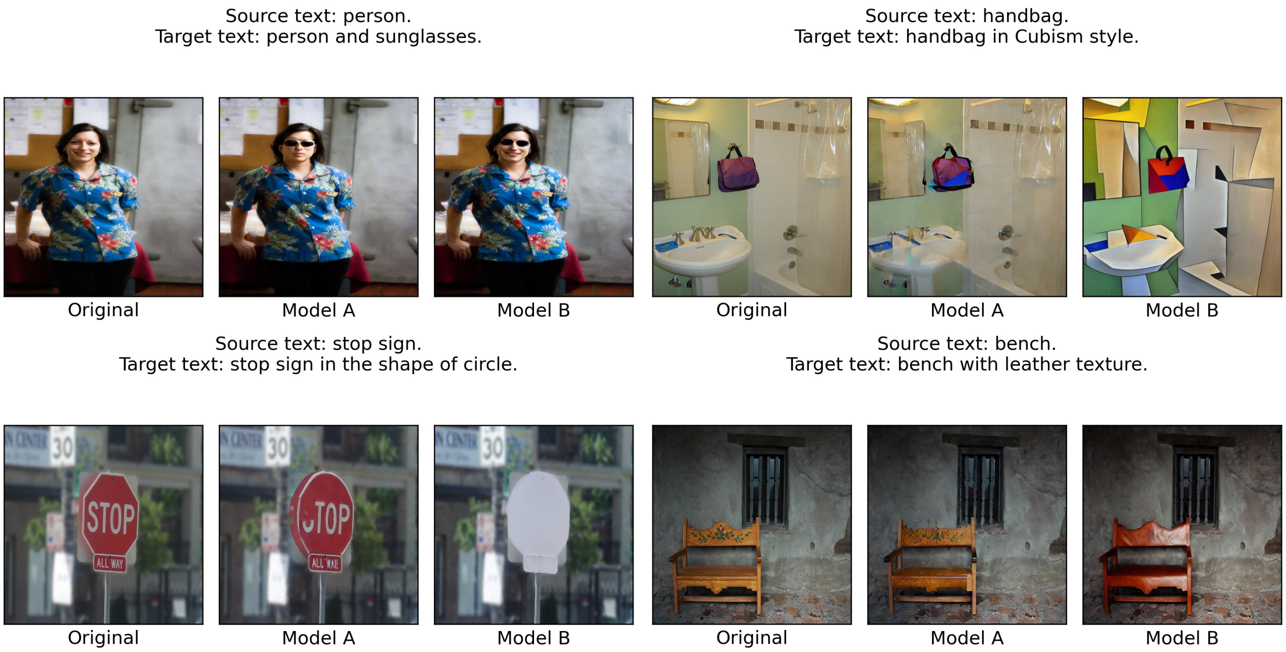


Figure 11. User Study Details. The figure shows the instructions and example questions of our user study.

As existing text-guided image editing models do not guarantee adequate visual quality, we manually select the images that show sufficient change in the image to conduct a user study. Each participant is provided with a source image, its corresponding target text, and two variants of edited images. Then, human evaluators are instructed to choose the image with better editing quality. As shown in Fig. 11, clear guidelines are provided to instruct the participants to evaluate the images based on both the preservation of the source image and the modifications toward the target text.

C.5. Benchmark Datasets

TEdBench comprises 100 pairs of source image and target text. It focuses on specific settings where the source image has a single object at the center, and the corresponding target text only modifies some attributes of that object.

EditVal contains 648 image-text pairs that cover 13 different types of edits, including object addition, object replacement, and size modification. Since it has such complicated editing scenarios, the models that we use for editing could not properly edit the majority of the cases, leaving almost no samples with enough quality for user study. Therefore, we use the subset of EditVal, which encompasses eight editing types that show adequate modification for proper evaluation.

MagicBrush is a benchmark specifically designed to evaluate sequential editing tasks, where iterative modifications are made to different parts of the source image.

Dreambooth enables the modification of specific instances within the source image by providing corresponding masks along with image-text pairs; however, since typical editing models do not utilize masks as input, we only consider the image-text pairs in our evaluation.

CelebA dataset consists of 50 image-text pairs that guide changes specific to facial attributes. We create target texts by swapping attributes of human faces.

D. Additional Results

D.1. Combination of Preservation and Modification Centric Metrics

As discussed in Sec. 3.1, combining modification-centric metric (CLIP-T) with existing preservation-centric metrics (DINO similarity, Segment Consistency, CLIP-I) shows negligible improvement or rather deteriorates in terms of alignment with human judgment and ground truth selection test. Due to the spatial constraint, we have shown two of the datasets, EditVal and CelebA, in the main paper. In Fig. 12, we demonstrate the results of the other three datasets. Notably, the combination of CLIP-I and CLIP-T shows improvement in the TEdBench dataset. Since TEdBench is the dataset of the simplest setting, where source images are highly object-centric and the target text instructs relatively simple modification, a simple strategy of combining these two metrics could be a viable option for evaluation. However, as shown in most editing cases, such a simplistic combination approach fails to show large improvement, underscoring the need for a metric tailored for text-guided image editing.

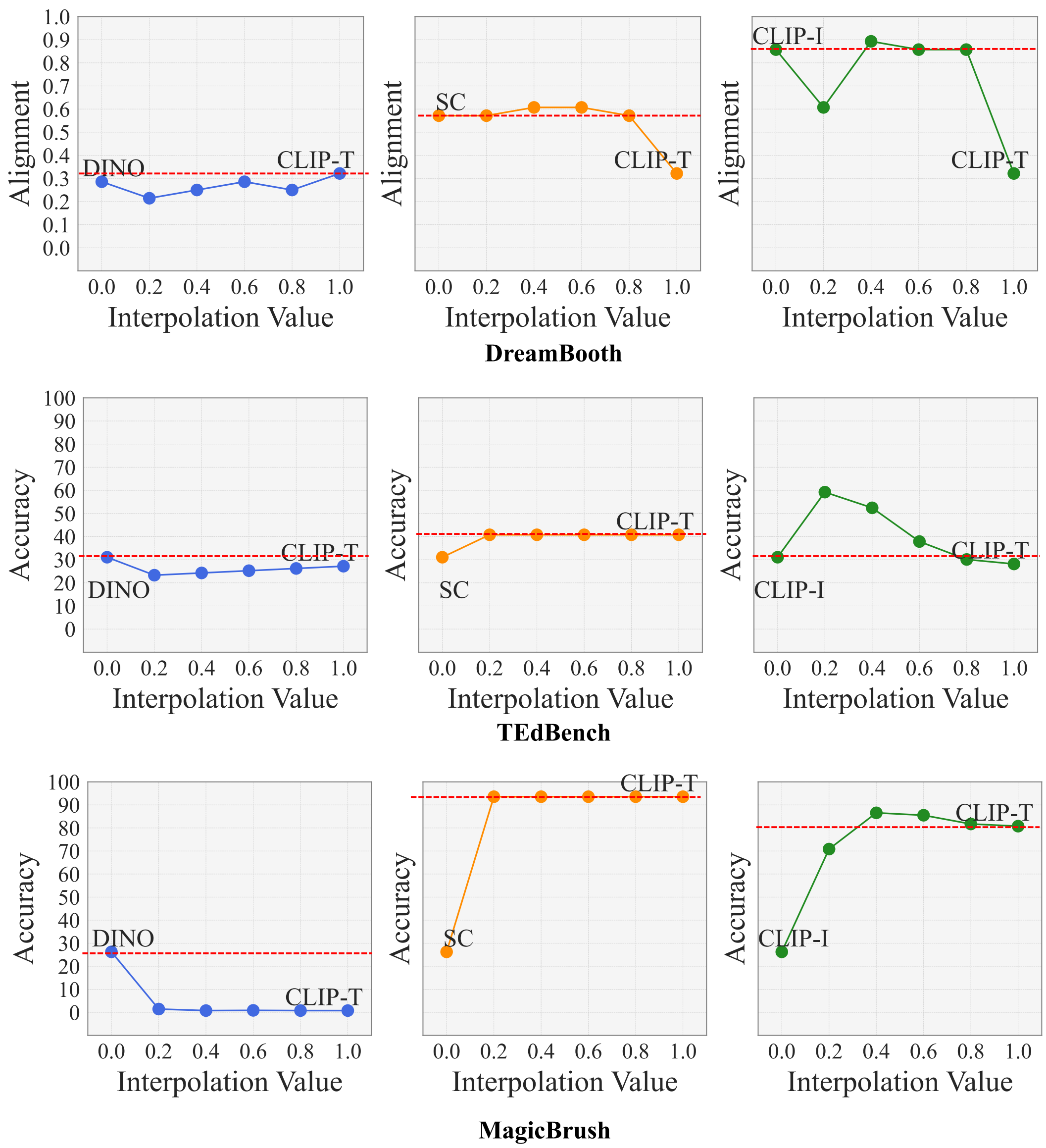


Figure 12. Interpolation of Preservation- and Modification-Centric Metrics.

D.2. Additional Result on Augmenting Directional CLIP Similarity

Table 11. Effect of Augmenting Attributes into $\text{CLIP}_{\mathbf{dir}}$**.** We use $\boldsymbol{s}_{2\text{AFC}}$ for CelebA, EditVal, and Dreambooth, and $\mathbf{Acc}_{\text{Both}}$ for TEdBench and MagicBrush.

	Weighting	CelebA	EditVal	DreamBooth	TEdBench	MagicBrush
$\text{CLIP}_{\mathbf{dir}}$		0.673	0.697	0.357	0.350	<u>0.601</u>
+ **src attr.**	✗	0.816	0.629	0.357	0.400	0.429
	✓	0.796	<u>0.725</u>	0.464	0.440	0.523
+ **trg attr.**	✗	<u>0.819</u>	0.708	<u>0.536</u>	0.420	0.533
	✓	0.734	0.513	0.464	0.450	0.443
+ **src & trg attr.**	✗	0.816	0.607	<u>0.536</u>	0.440	0.402
	✓	0.636	0.600	<u>0.536</u>	<u>0.500</u>	0.407
`AugCLIP`	✓	**0.883**	**0.831**	**0.857**	**0.570**	**0.889**

In Tab. 6, we demonstrate that the simple strategy of directly augmenting CLIP_{dir} with the source and target attributes fails to outperform `AugCLIP`. Additionally, we show the effect of applying the weighting strategy (✓) of Eq. (6) when aggregating attributes into CLIP_{dir} in Tab. 11. Note that augmentation of CLIP_{dir} without weighting (✗) is already reported in Tab. 6.

Formally, for augmenting the source text T_{src} with source attributes in $\mathcal{D}_S$, directional CLIP similarity is redefined as

$$\texttt{cs}\Big(E(I_{\text{edit}}) - E(I_{\text{src}}), E(T_{\text{trg}}) - \mathbb{E}_{\mathbf{s}_i \in \mathcal{D}_S}(s_i)\Big), \tag{13}$$

where $\mathbb{E}$ means expectation. Using the weighting strategy with α defined in Eq. (6), CLIP_{dir} is reformulated as

$$\texttt{cs}\left(E(I_{\text{edit}}) - E(I_{\text{src}}), E(T_{\text{trg}}) - \mathbb{E}_{\mathbf{s}_i \in \mathcal{D}_S}\left(\alpha(\mathbf{s}_i) \cdot \mathbf{s}_i\right)\right). \tag{14}$$

The same formulation applies for the target text T_{trg} as well.

Across all configurations, with and without weighting, `AugCLIP` outperforms CLIP_{dir} in terms of alignment with human judgment and ground truth selection test accuracy. This emphasizes that our metric, `AugCLIP`, notably well-performs compared to CLIP_{dir}.

D.3. Comparison with GPT-4V

Table 12. Comparison with GPT-4V. We use $\boldsymbol{s}_{2\text{AFC}}$ for CelebA, EditVal, and Dreambooth, and $\mathbf{Acc}_{\text{Both}}$ for TEdBench and MagicBrush.

	CelebA	EditVal	DreamBooth	TEdBench	MagicBrush
GPT-4V	0.876	**0.933**	0.821	**0.620**	0.703
`AugCLIP`	**0.883**	0.831	**0.857**	0.570	**0.889**

Recently, GPT-4V [22] has been employed in evaluating various tasks, including text-guided image editing, text-to-image generation, and image quality assessment. Since GPT-4V is one of the best-performing multi-modal large language models, we test the ability of GPT-4V's effectiveness in evaluating the quality of text-guided edited images. For evaluation, we use the following prompt: "Given a source image and two edited images, you should choose a better edited one based on the source and target text. Source text describes the source image, and target text describes the editing. A well-edited image should preserve the essence of the source image while following the target text."

As shown in Tab. 12, GPT-4V outperforms `AugCLIP` in tasks such as EditVal and TEdBench, which involve simple edits like modifying a single object's attribute. In contrast, our proposed metric, `AugCLIP`, effectively captures minor differences by augmenting attributes of the source image and target text and shows better performance in other benchmarks with complex scenarios. This finding is consistent with prior research [30], which suggests that GPT-4V struggles to differentiate between images with subtle differences.

D.4. Additional Examples on Problem 1 of Directional CLIP Similarity

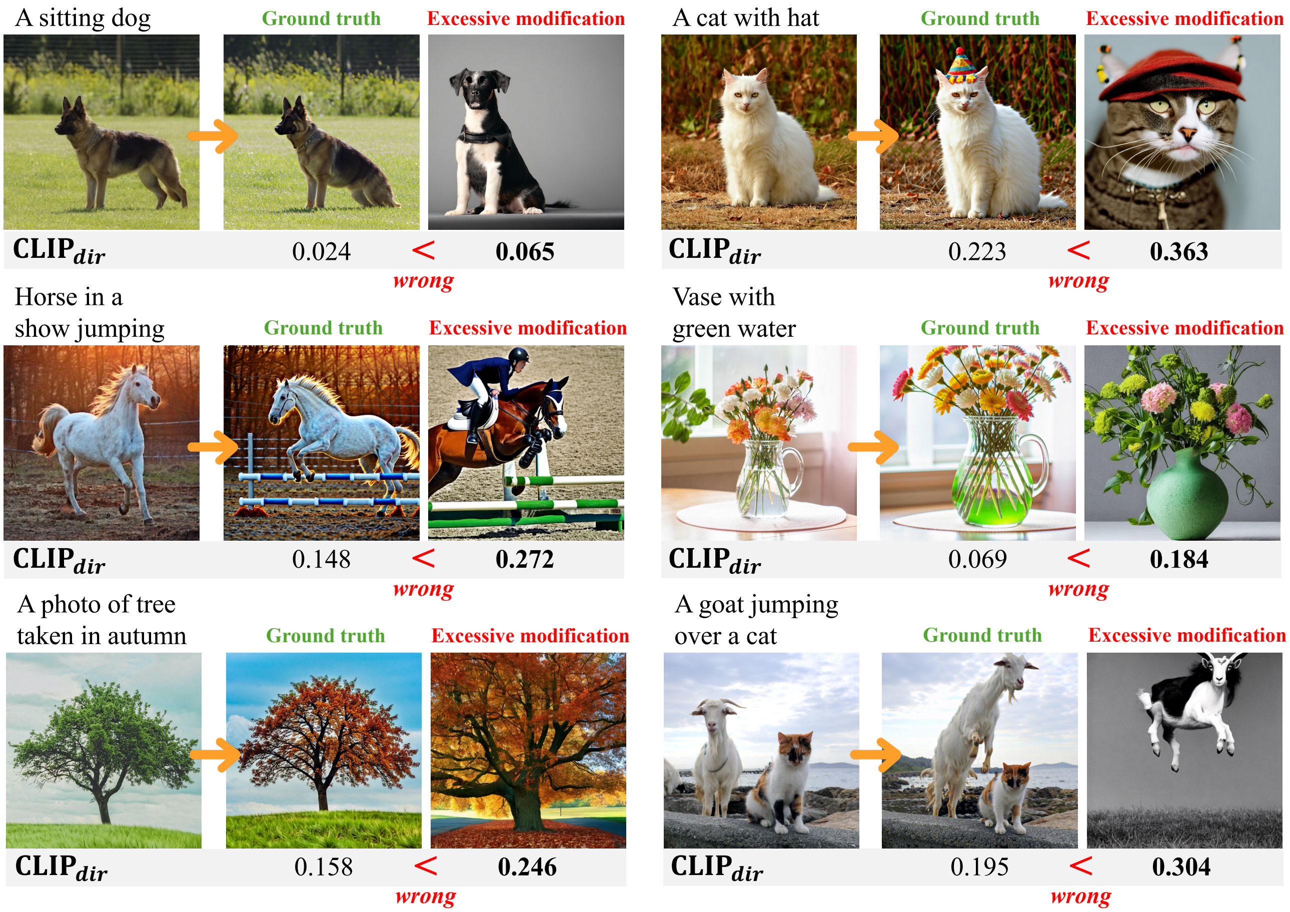


Figure 13. Additional Examples on Problem 1 of Directional CLIP Similarity. CLIP$_{\text{dir}}$ assigns higher scores to excessive modification, over well-edited ground truth images.

D.5. Additional Examples on Problem 2 of Directional CLIP Similarity

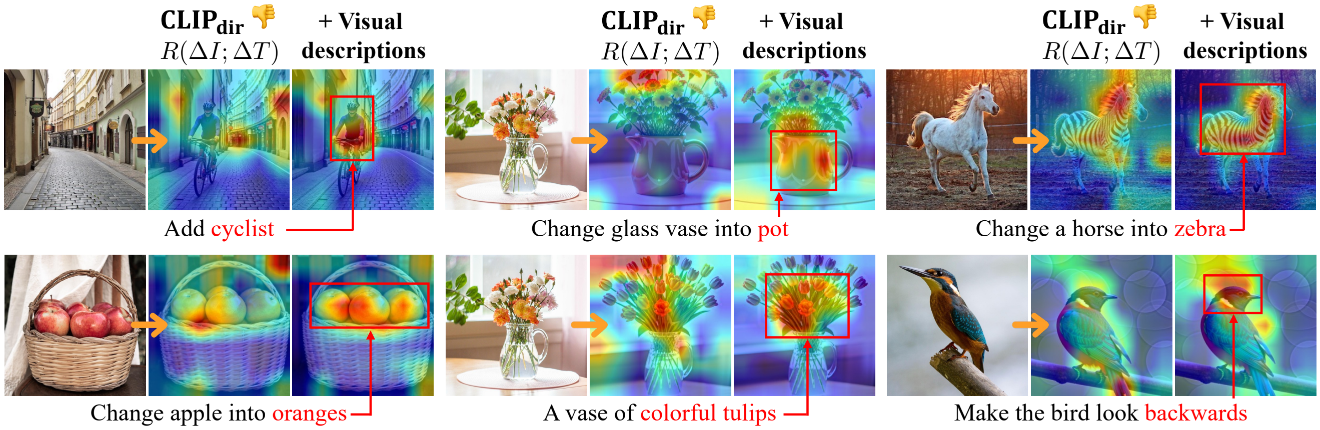


Figure 14. Additional Examples on Problem 2 of Directional CLIP Similarity. CLIP_{dir} evaluates edited images by attending to irrelevant regions of the image. Adding visual annotations helps CLIP_{dir} properly attend to edited regions.

To assess directional CLIP similarity's capability to focus on the image regions modified following the target text rather than peripheral or unchanged regions, we use the relevancy map [4], $\boldsymbol{R}$. The relevancy map visualizes the transformer's attention on an image corresponding to a given text depending on their cosine similarity. Specifically, for an image $I \in \mathbb{R}^{h\times w}$ and text T, the relevancy map is computed as

$$\boldsymbol{R}(I;T) = \nabla_{\boldsymbol{A}} \texttt{cs}(E(I), E(T); \boldsymbol{A}) \odot \boldsymbol{A} \in \mathbb{R}^{h\times w},$$

where $\boldsymbol{A}$ represents the attention scores of the CLIP visual encoder and $\odot$ denotes the Hadamard product. To visualize the relevancy map of CLIP_{dir}, which is a cosine similarity between ΔI and ΔT, we subtract the two relevancy maps as

$$\boldsymbol{R}(\Delta I; \Delta T) = \boldsymbol{R}(I_{\text{edit}}; \Delta T) - \boldsymbol{R}(I_{\text{src}}; \Delta T).$$

Fig. 3(b) and Fig. 14 illustrate the relevancy maps of CLIP_{dir} across multiple cases and their improvement achieved by incorporating manually annotated visual descriptions. Unlike CLIP_{dir}, `AugCLIP` measures the cosine similarity between the estimated well-edited *image* and the edited *image*, rather than between an *image* and *text*. As a result, the relevancy map, which requires direct comparison of the image and text, cannot be applied to `AugCLIP`.

E. Qualitative Results

We present qualitative samples of the **2AFC Test**, as reported in Tab. 3, using the CelebA, EditVal, and DreamBooth datasets. For each dataset, we randomly select triplets consisting of a source image, target text, and edited images to demonstrate how `AugCLIP` consistently assigns higher scores to the edited image preferred by human evaluators. The preferred image, highlighted with a red box, appears in the middle. Each case represents a two-alternative forced choice (2AFC) survey, where the source image on the far left is altered into the middle and rightmost images. We observe that directional CLIP similarity often favors excessively modified images. For instance, in the second row of Fig. 15, where the target text is "high arch of the eyebrows," directional CLIP similarity prefers an edited image that changes the gender of the source image into a man. Similarly, when the target text is "wrinkle-free skin," directional CLIP similarity assigns a higher score to an image where the hair bangs are missing.

E.1. CelebA

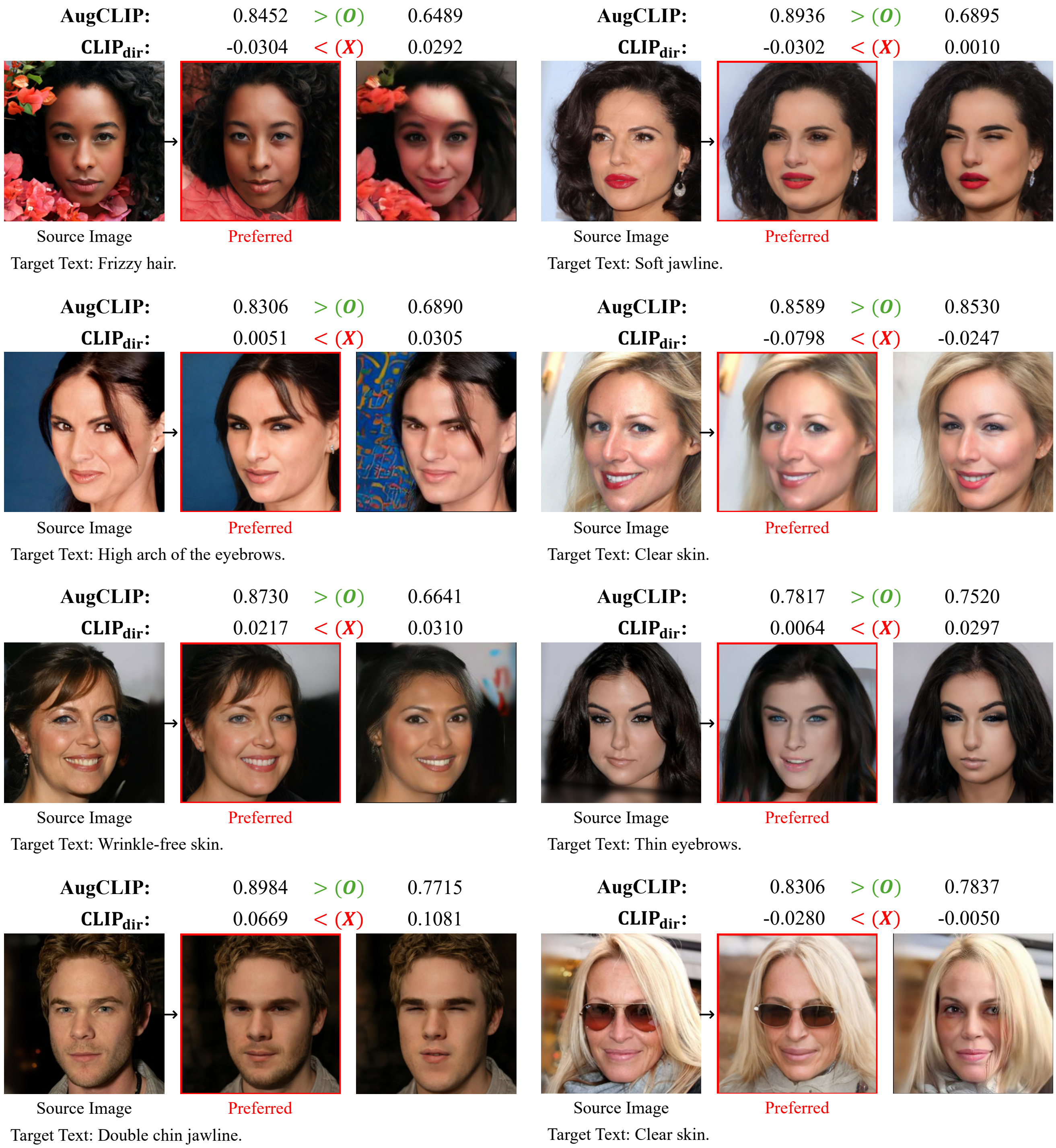


Figure 15. Qualitative Results on CelebA (2AFC Test).

E.2. EditVal

AugCLIP: 0.9683 > (*O*) 0.8525

CLIP$_{\mathbf{dir}}$: 0.1019 < (*X*) 0.1252

Source Image Preferred

Target Text: Change the texture of bicycle into wooden.

AugCLIP: 0.9326 > (*O*) 0.9160

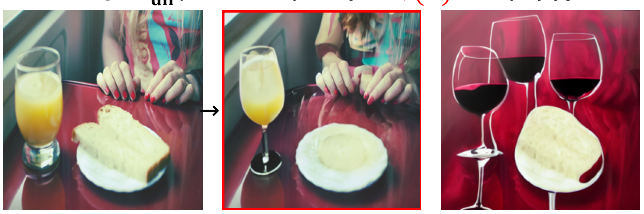
CLIP$_{\mathbf{dir}}$: 0.1112 < (*X*) 0.1853

Source Image Preferred

Target Text: Add chocolate toppings to donut.

AugCLIP: 0.8076 > (*O*) 0.6226

CLIP$_{\mathbf{dir}}$: 0.1416 < (*X*) 0.1968



Source Image Preferred

Target Text: Change a cup into wine glass.

AugCLIP: 0.8013 > (*O*) 0.5605

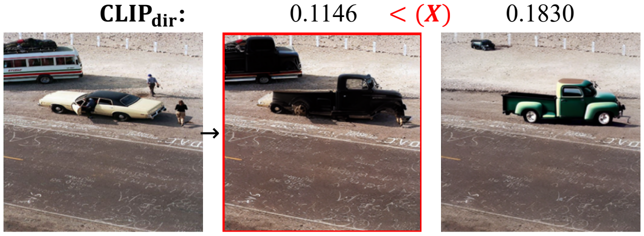
CLIP$_{\mathbf{dir}}$: 0.1572 < (*X*) 0.2178

Source Image Preferred

Target Text: Change the background into grassland.

AugCLIP: 0.8765 > (*O*) 0.8672

CLIP$_{\mathbf{dir}}$: 0.1146 < (*X*) 0.1830

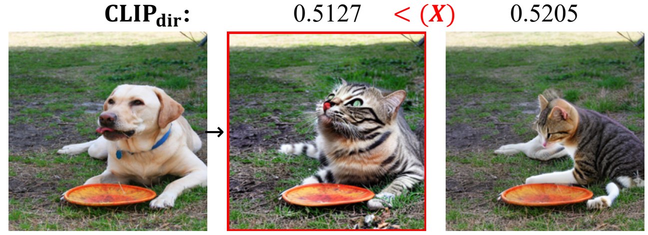


Source Image Preferred

Target Text: Change a car into pickup truck.

AugCLIP: 0.8618 > (*O*) 0.8599

CLIP$_{\mathbf{dir}}$: 0.5127 < (*X*) 0.5205

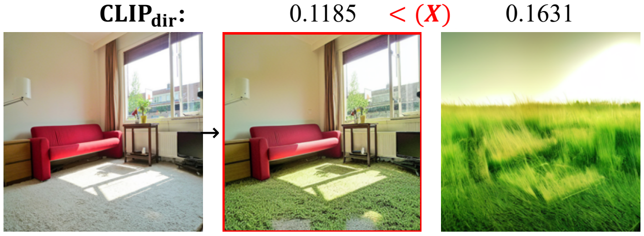


Source Image Preferred

Target Text: Change a dog into cat.

AugCLIP: 0.9072 > (*O*) 0.5903

CLIP$_{\mathbf{dir}}$: 0.1185 < (*X*) 0.1631

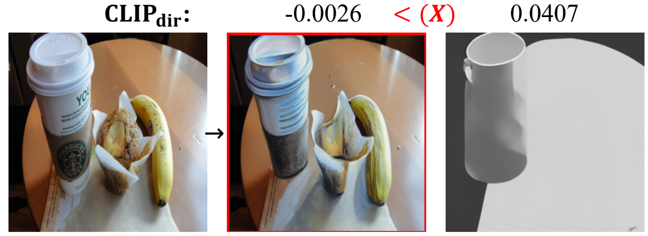


Source Image Preferred

Target Text: Change the background into grassfield.

AugCLIP: 0.6411 > (*O*) 0.5469

CLIP$_{\mathbf{dir}}$: -0.0026 < (*X*) 0.0407



Source Image Preferred

Target Text: Change an art painting of a cup in minimalism style.

Figure 16. Qualitative Results on EditVal (2AFC Test).

E.3. DreamBooth

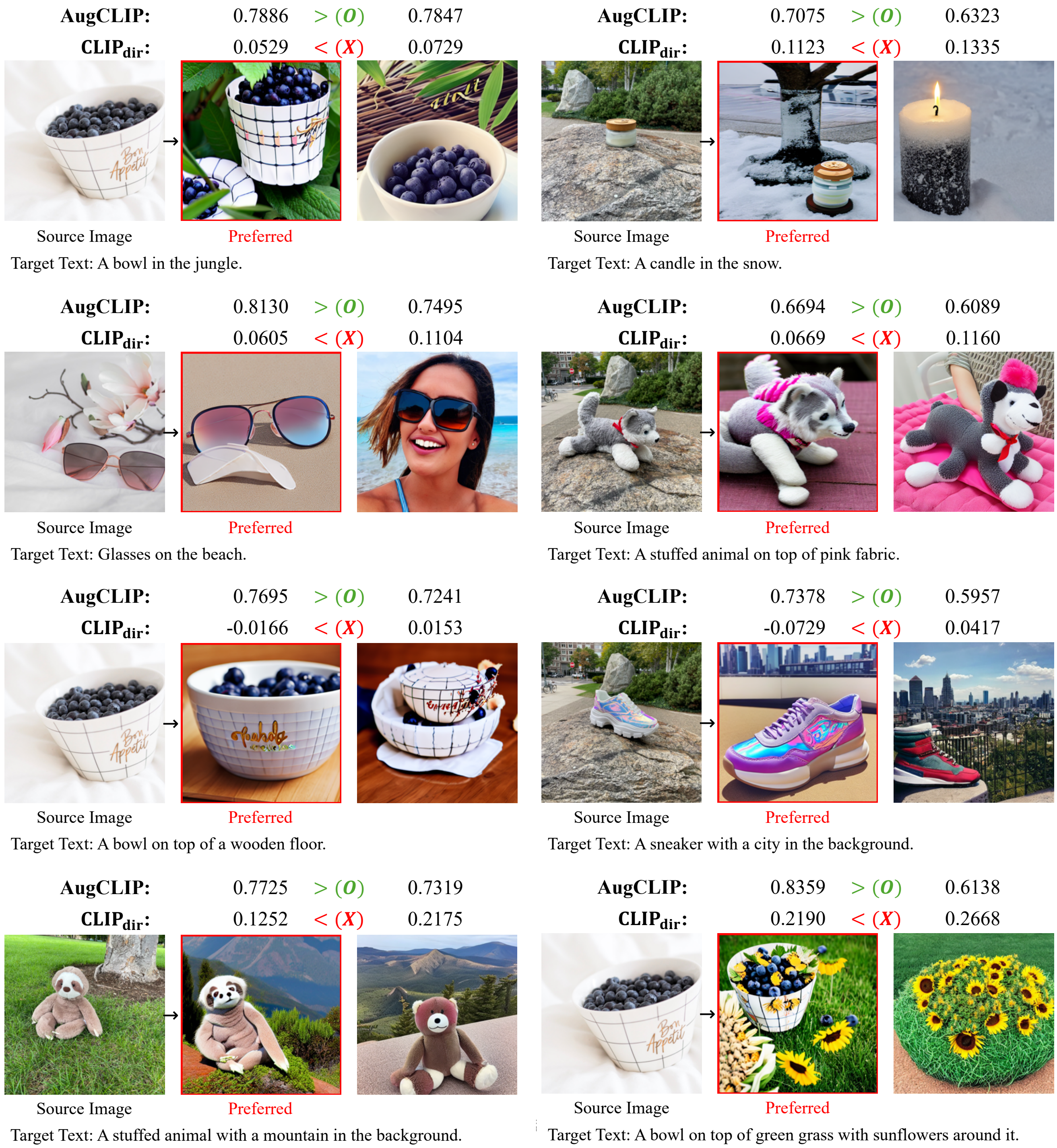


Figure 17. Qualitative Results on DreamBooth dataset (2AFC test).

Additionally, we provide qualitative samples from the **Ground Truth Selection Test**, reported in Tab. 4, using the TEdBench and MagicBrush datasets (Fig. 18, 19 and Fig. 20, 21). In these cases, the ground truth image is located in the second column, the excessively preserved image in the third column, and the excessively modified image in the fourth column. Once again, we observe that directional CLIP similarity tends to prefer excessive modifications.

E.4. TEdBench

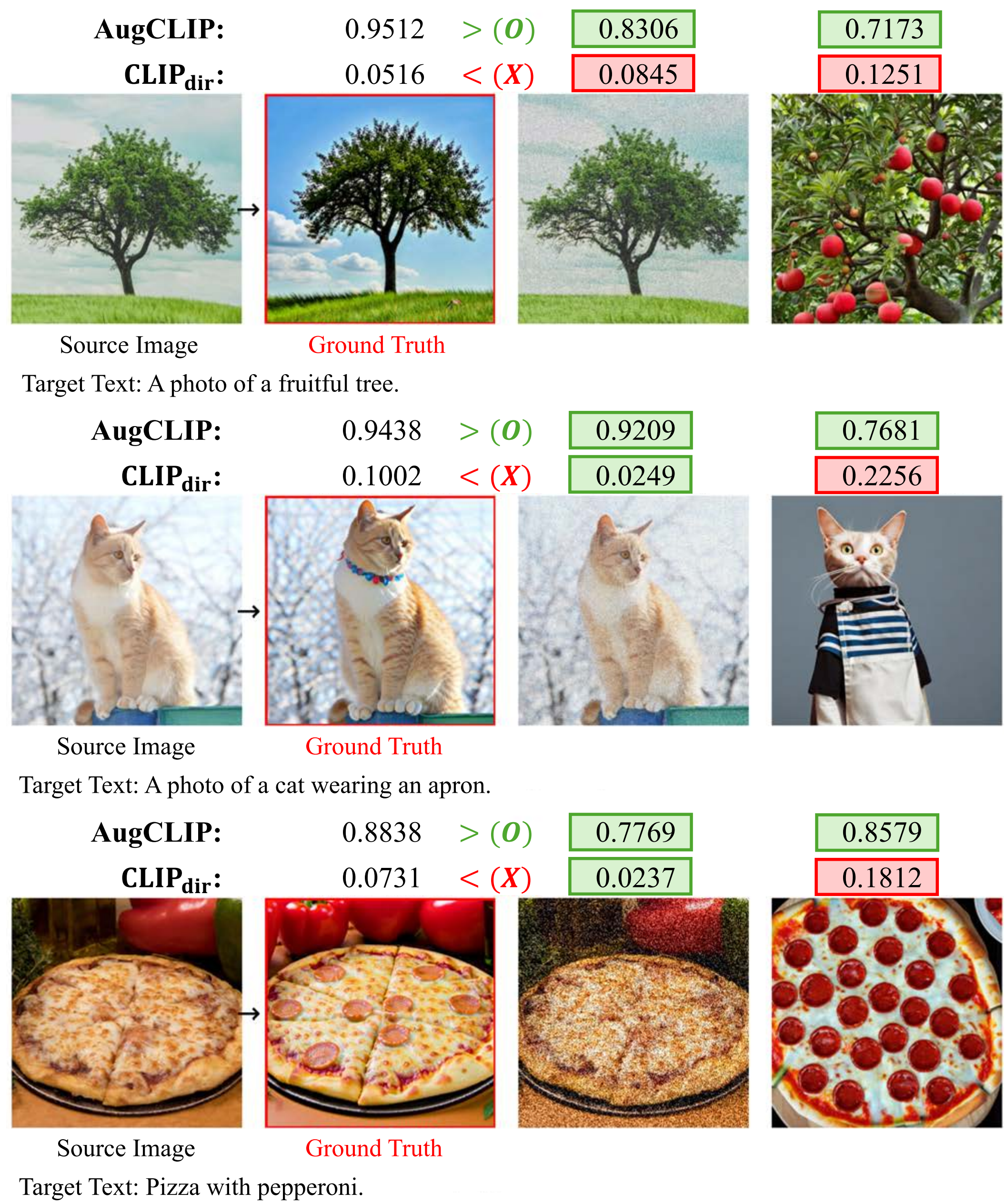


Figure 18. Qualitative Results on TEdBench (Ground Truth Selection Test).

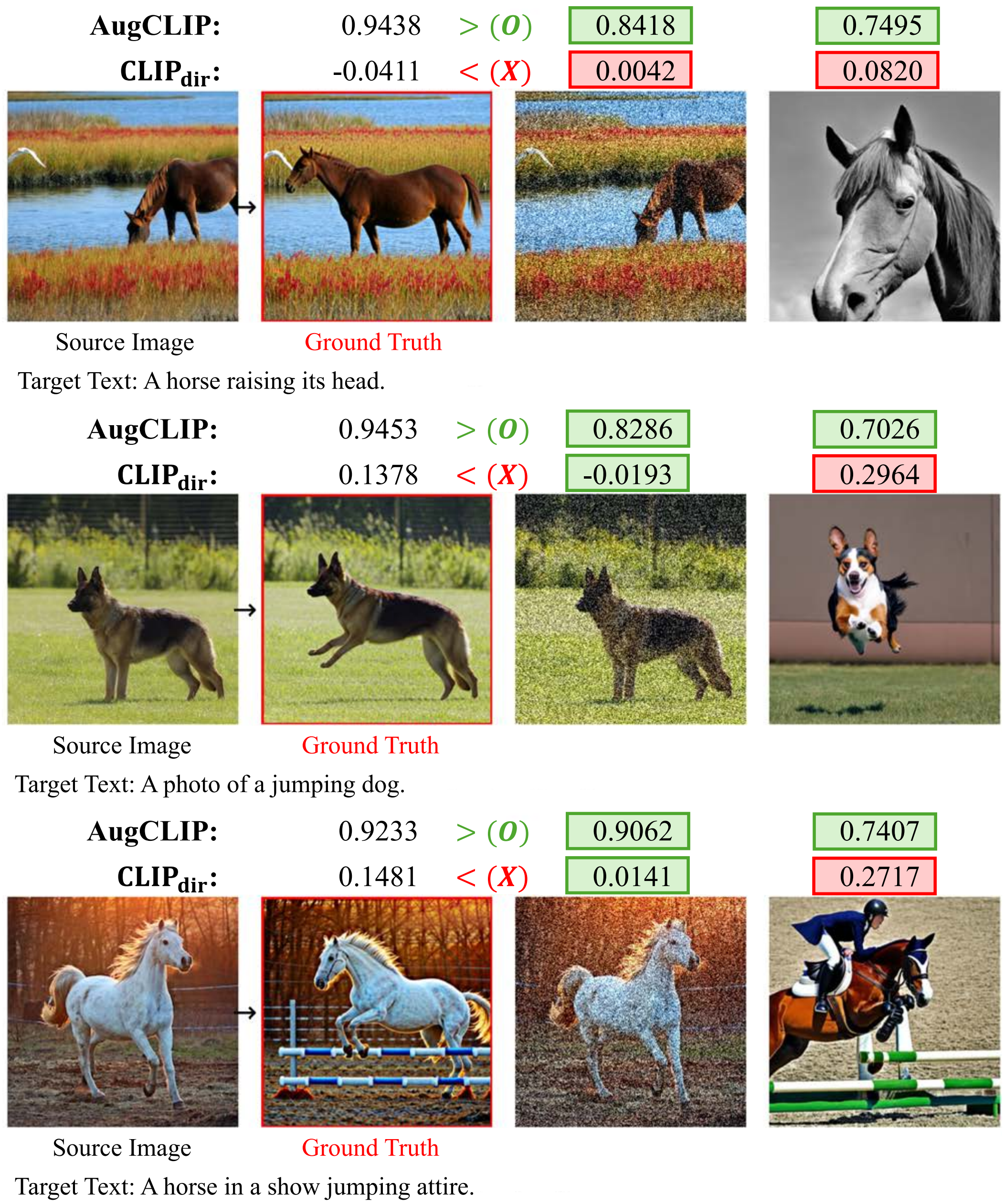


Figure 19. Qualitative Results on TEdBench (Ground Truth Selection Test).

E.5. MagicBrush

Figure 20. Qualitative Results on MagicBrush (Ground Truth Selection Test).

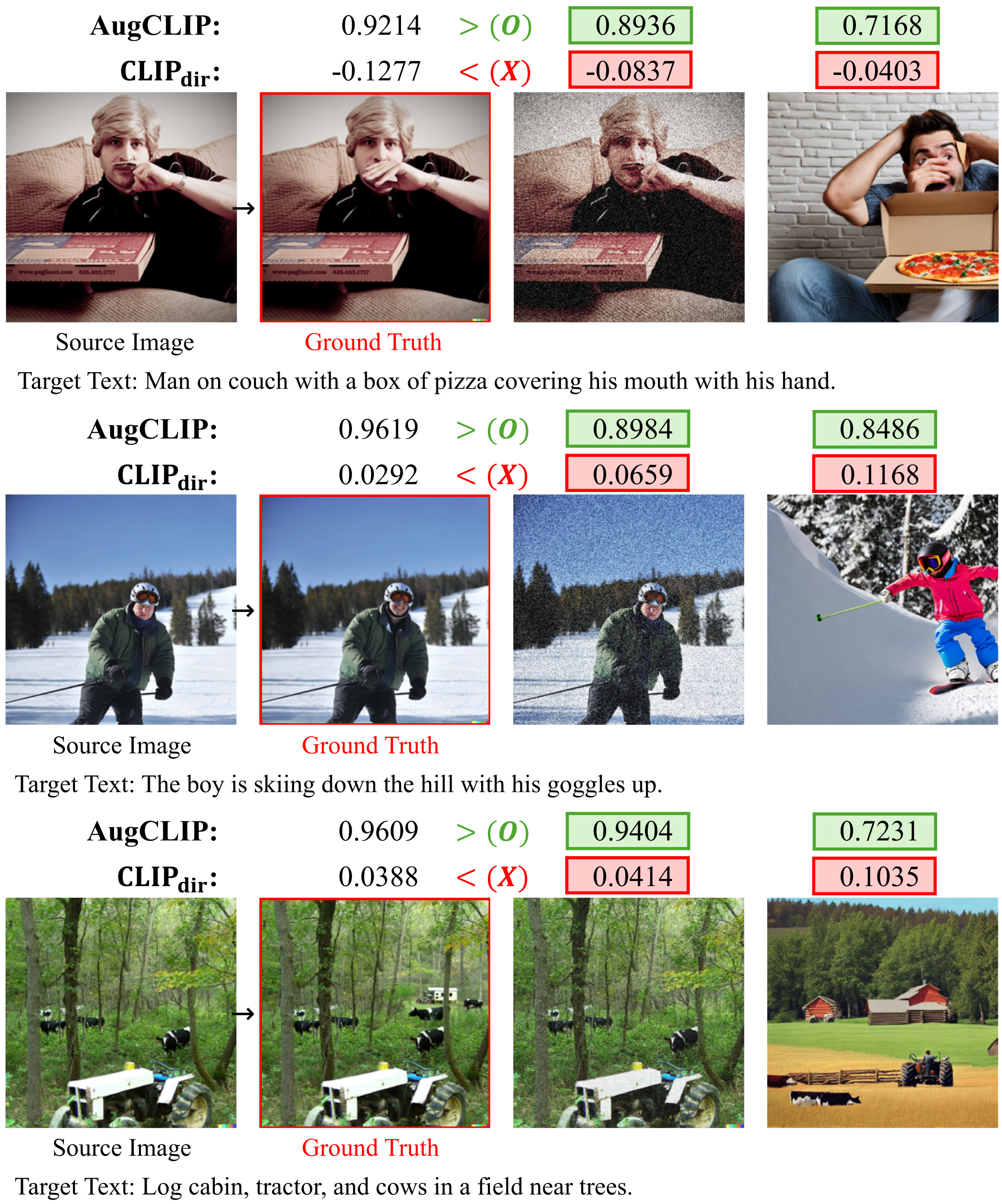


Figure 21. Qualitative Results on MagicBrush (Ground Truth Selection Test).

E.6. Failure Cases of `AugCLIP`

Compared to directional CLIP similarity, `AugCLIP` shows superior alignment with human evaluation and a stronger ability to classify ground truth images. However, there are several cases where directional CLIP similarity aligns closely with human preferences. Fig. 22 illustrates examples where `AugCLIP` diverges from human judgment.

For instance, in the first-row example, both edited images are adequately modified from the source to resemble the target text "dog." However, the middle image emphasizes dog-like features more prominently while the right image exhibits subtler changes. Human evaluators tend to favor the more prominently modified one. In the example of adding fruit toppings to donuts, both edited images accurately depict fruit toppings while preserving the original content. Yet, human evaluators prefer the middle image, which better retains the original donut's color and texture. Here, preference is skewed toward better preservation.

Although the edits in these examples are well-executed in terms of balancing preservation and modification, human preferences remain inherently subjective and vary significantly from case to case. This highlights the limitation of evaluation metrics in fully capturing the nuances of human judgment.

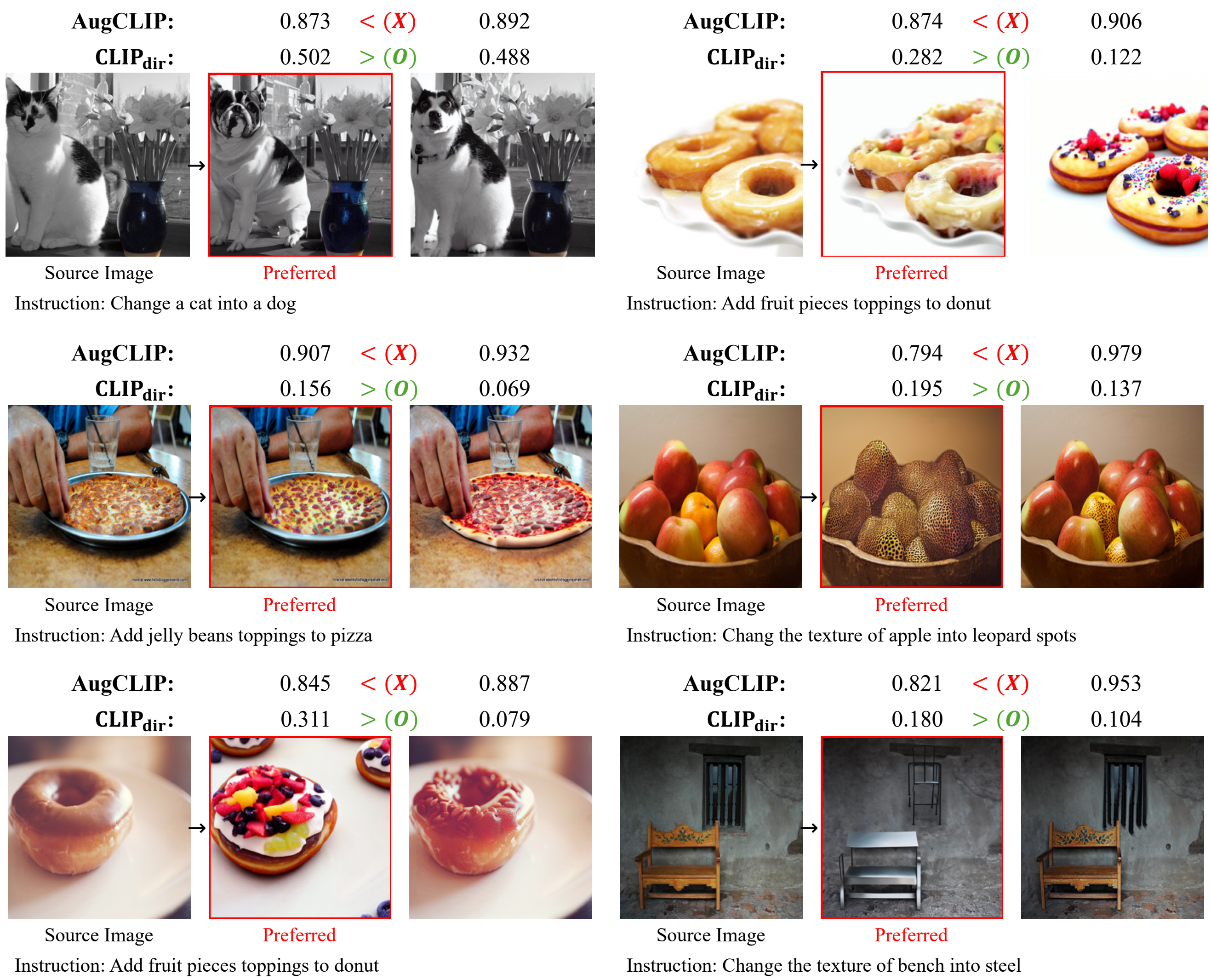


Figure 22. Failure Cases of `AugCLIP`. Failure cases where directional CLIP similarity correctly assigns higher evaluation scores to images that human evaluators prefer, while `AugCLIP` fails to.