A. Implementation Details

To help reproduce our results, we report the detailed hyperparameter settings and model architectures used in this work in Table 2.

In terms of optimization, we use different loss balancing weight β and different initialization of combination weights λ_t for attributes on person and scenes, since edits on person require more strict content preservation than edits on scenes. Specifically, when editing person, we adopt a larger β on perceptual loss and initialize λ_t such that $\boldsymbol{c}_{1:T}$ is more similar to the style-neutral description $\boldsymbol{c}^{(0)}$. Both these settings encourage content preservation when disentangling attributes on person. For the directional CLIP loss, we use the pre-trained `ViT-B/32` [58]. For the model used to compute perceptual loss, we adopt pre-trained `VGG-16` [29]. The stable diffusion model we use is the pre-trained `stable-diffusion-v1-4` [4]. We use all pre-trained models without changing any parameters.

B. Text Descriptions Used for Attribute Disentanglement and Image Editing

In this section, we provide the exact text descriptions we use to disentangle target attributes and perform image editing in this paper. For each target attribute or edit, the text description consists of a style-neutral description and a description with explicit styles, whose embeddings are denoted as $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ respectively. For brevity, we use these notations to represent their corresponding text descriptions and list them in Table 3. For each attribute in the table, we also provide the corresponding figure that demonstrates the visual effects.

We emphasize that our method is generally robust to the choice of text descriptions and is not restricted to the text listed here. Please refer to Sec. 4.3 and Sec. G for more analyses on the robustness of our method to different choices of text descriptions.

C. Additional Examples on Partially Replacing Text Embeddings

As described in Sec.3.2, the stable diffusion model is inherently capable of disentangling attributes, and we demonstrate that such disentanglement can be triggered by partially replacing the text embeddings from a style-neutral one to the one with explicit styles. In this section, we provide more examples to better illustrate this phenomenon.

The examples are shown in Fig. 8. In the figure, each row demonstrates an example of replacing the text embedding at later denoising steps. In other words, we use the style-neutral description $\boldsymbol{c}^{(0)}$ during early denoising steps (T to t'), and replace it with the one containing explicit styles $\boldsymbol{c}^{(1)}$ during later steps (t' to 0). $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ are listed on the left and right of each row, respectively. In the first column, t' is set to 0, which corresponds to using $\boldsymbol{c}^{(0)}$ for all denoising steps. On the other hand, $t' = T$ in the last column means the text embedding replacement happens at the beginning, and the denoising is entirely conditioned on $\boldsymbol{c}^{(1)}$.

From these results, we observe that only replacing $\boldsymbol{c}^{(0)}$ with $\boldsymbol{c}^{(1)}$ in later denoising steps can maintain the contents in the style-neutral image, and more replaced steps lead to stronger modification effects on the target attribute. Thus for some specific time steps t' (*e.g.,* $t' = 0.8T$ for "red brick" and $t' = 0.9T$ for "renaissance style"), the target attribute can be successfully disentangled. This verifies the inherent disentanglement ability in the stable diffusion model. Furthermore, we observe that for different attributes, the optimal t' could be different. For example, $t' = 0.8T$ disentangles the "red brick" and "in sunset" attributes but does not bring successful modifications for other two attributes. In other words, one has to search the best t' for optimal disentanglement. This motivates a more principled optimization scheme to combine text embeddings for the best disentanglement, which is described in Sec. 3.3.

D. More Examples of Disentangled Attributes

We now provide more examples of attributes that can be disentangled by our method in Fig. 9. These results show that our method is generalizable to a broad range of attributes, and it satisfies the two criteria for disentanglement discussed in Sec. 1.

E. Details of Subjective Evaluation

In this section, we detail our subjective evaluation process. We compare the performance of our method with DIFFUSIONCLIP [35] on the image editing task. Specifically, we consider two existing datasets used by DIFFUSIONCLIP: Celeb-A [71] that focuses on person and LSUN-church [76] that focuses on churches. For each dataset, we select the first 20 images as source images[2] and evaluate 4 types of edits used in [35], *i.e.,* tanned, male, sketch, and pixar for human faces, and golden, wooden, snowy, and red brick for churches. We compare our editing results with the results generated by the official checkpoints of DIFFUSIONCLIP that are fine-tuned for each target edit. We conduct evaluation on Amazon Mechanical Turk, and we require all participants to be master workers in order to answer our questions. In total, 11 workers participate in the study. For each edit, we present participants with the source image, as well as images edited by two methods in random order. We ask participants to answer the following three questions:

(1) Which one is perceptually consistent with the target edit attribute and looks like a natural image?

[2]For the "male" attribute, we use 20 female images from Celeb-A dataset as source images.

		Value
Optimization	β	0.05 (attributes on person) 0.03 (attributes on scenes)
	Optimizer	Adam [37]
	Learning rate	0.05
	λ_t initialization	0.0 for $t \geq 0.8T$, 1.0 for $t < 0.8T$ (attributes on person) 0.0 for $t \geq 0.9T$, 1.2 for $t < 0.9T$ (attributes on scenes)
	Checkpoint for CLIP loss	`ViT-B/32` [58]
	Checkpoint for perceptual loss	`VGG-16` [29]
Diffusion Model	Model checkpoint	`stable-diffusion-v1-4` [4]
	Sampling steps	50
	Sampling variance	0.0
	Resolution	512×512
	Latent channels	4
	Latent down-sampling factor	8
	Conditional guidance scale	7.5

Table 2. Hyperparameter settings and model architectures used in this paper.

Type	Attribute	$\boldsymbol{c}^{(0)}$	$\boldsymbol{c}^{(1)}$	Example
Scenes	**Global attributs:**			
	Children Drawing	A castle	A castle, children drawing style	Fig. 4
	Cyberpunk Style	A street view	A street view, Cyberpunk style	Fig. 4
	Anime Style	A lake in mountains	A lake in mountains, anime style	Fig. 17
	Wooden Building	A photo of church exterior	A photo of church exterior, wooden style	Fig. 11
	Golden Building	A photo of church exterior	A photo of church exterior, golden style	Fig. 4
	Red Brick Building	A photo of church exterior	A photo of church exterior, red brick	Fig. 1
	In Sunset	A photo of church exterior	A photo of church exterior, in sunset	Fig. 11
	At Dark Night	A photo of seaside	A photo of seaside, dark night	Fig. 6
	At Starry Night	A photo of railway	A photo of railway, in milky galaxy	Fig. 9
	Covered by Snow	A photo of church exterior	A photo of church exterior, covered by snow	Fig. 17
	Local attributes:			
	Cherry Blossom	A forest	A forest, cherry blossom	Fig. 4
	Rainbow	A lake in mountains	A lake in mountains, rainbow	Fig. 6
	Foothills	A man sitting on grass	A man sitting on grass, in mountains	Fig. 6
Person	**Global attributes:**			
	Renaissance Style	A photo of person	A photo of person, renaissance style	Fig. 1
	Egyptian Mural Style	A photo of person	A photo of person, Egyptian mural style	Fig. 4
	Sketch	A photo of person	A photo of person, sketch style	Fig. 10
	Pixar	A photo of person	A photo of person, pixar style	Fig. 10
	Young	A photo of person	A photo of person, young	Fig. 4
	Tanned	A photo of face	A photo of face, tanned	Fig. 10
	Male	A photo of face	A photo of face, male	Fig. 10
	Local attributes:			
	Smiling	A photo of person	A photo of person, smiling	Fig. 17
	Crying	A photo of person	A photo of person, crying	Fig. 9
	Angry	A photo of person	A photo of person, angry	Fig. 9

Table 3. Text descriptions used for attributes disentanglement and image editing in this paper. For each attribute, we report the descriptions used for $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ as well as the corresponding visual example in the paper.

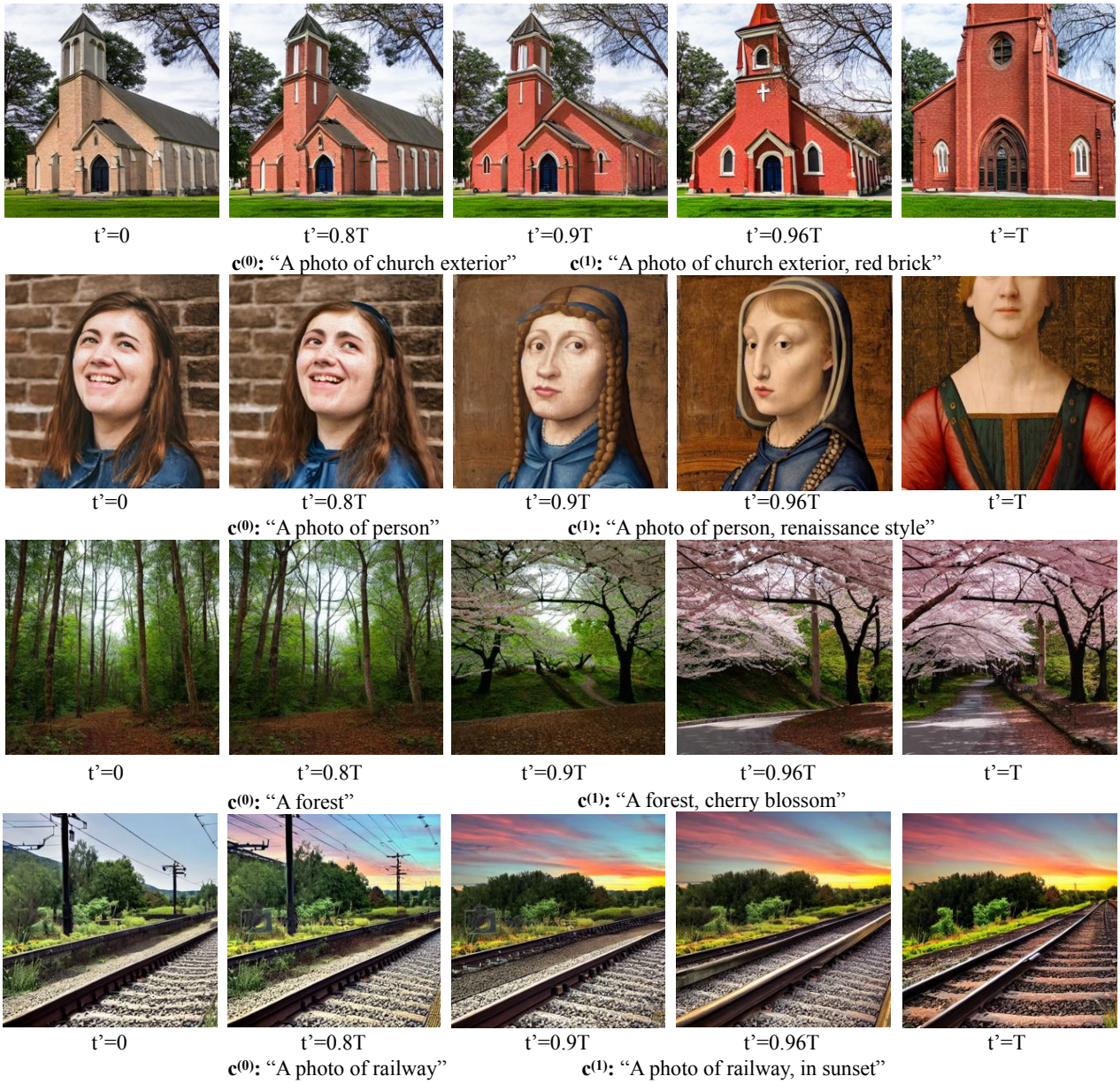

Figure 8. Inherent disentanglement capability in the stable diffusion model. For each row, we partially replace the style-neutral text description $\boldsymbol{c}^{(0)}$ with another description $\boldsymbol{c}^{(1)}$ that includes the explicit style. Particularly, the denoising process conditions on $\boldsymbol{c}^{(0)}$ from T to t' and $\boldsymbol{c}^{(1)}$ from t' to 0.

(2) Which one better preserves the information of original images (e.g. background, shape)?
(3) Overall, which editing result is better?

Fig. 5 shows the results of subjective evaluation. We also provide all generated images by both methods in Fig. 10 and Fig. 11. We observe that DIFFUSIONCLIP tends to over-change the attribute in the image to the extent that introduces artifacts in the image and modifies other contents (*e.g.,* when making the church golden, it changes the sky and ground into gold). By contrast, our method generates more natural images and only modifies the target object.

F. More Comparisons with Baselines

In this section, we perform more qualitative comparison between our method and the state-of-the-art diffusion-model-based image editing methods. As mentioned in Sec. 4.2, since these methods either have not released code by the time of our submission or require auxiliary labels that are unavailable, we cannot include them in the subjective evaluation. Instead, we collect the source and edited

Figure 9. Examples of disentangled attributes. Text description with style ($\boldsymbol{c}^{(1)}$) is shown below each row, which consists of the style-neutral text description ($\boldsymbol{c}^{(0)}$) and target attribute description, separated by comma. Within each attribute, **first row:** results on optimization images; **second row:** results of transferring to unseen images; **left column:** source images; **right column:** modified images.

images in their papers and perform the same edit using our method for comparison. The results are shown in Fig. 12. Overall, our method achieves comparable editing results with the baselines. More specifically, our method produces stronger and more natural editing results for global target attributes (*e.g.,* rainy, snowy), while our method has difficulties disentangling attributes for small edits such as cake decorations. Meanwhile, we comment that the results of comparing with DIFFUSIONCLIP are blurry due to the low-resolution inputs obtained from the original paper. Please refer to Fig. 10 and Fig. 11 for a more comprehensive comparison with DIFFUSIONCLIP.

G. Effects of Varying Text Descriptions

Here we provide more analyses on whether our method is robust to different choices of text descriptions. We consider three types of variations in the following section.

The way of appending target attributes in $\boldsymbol{c}^{(1)}$: We explore how the way that $\boldsymbol{c}^{(1)}$ appends the target attribute descriptions can influence the results. Concretely, we fix the style-neutral text description $\boldsymbol{c}^{(0)}$ and explore three ways of concatenating the target attribute description in $\boldsymbol{c}^{(1)}$: direct concatenation, concatenation separated by a comma, and concatenation separated by proper prepositions ("with" and "in"). As can be observed in Fig. 17, for all three variations, our method consistently produces images with desired attributes on people, buildings, and natural scenes. We comment that this is not meant to be an exhaustive list of all the concatenation ways, but generally, our method is robust to how the target attribute descriptions are appended in $\boldsymbol{c}^{(1)}$.

More complex target attribute description in $\boldsymbol{c}^{(1)}$: We further investigate how would the complexity of the target attribute description in $\boldsymbol{c}^{(1)}$ affect the image editing results. In this experiment, we again fix $\boldsymbol{c}^{(0)}$ but gradually increase the complexity of target attribute descriptions by adding more correlated modifiers (*e.g.,* pink flower, pink tree). As shown in Fig. 13, using one modifier (the second column) is sufficient for successful edits. Meanwhile, we are able to achieve stronger editing effects with more correlated modifiers. For example, for the "cherry blossom" attribute, as we increase the number of correlated modifiers, the flowers

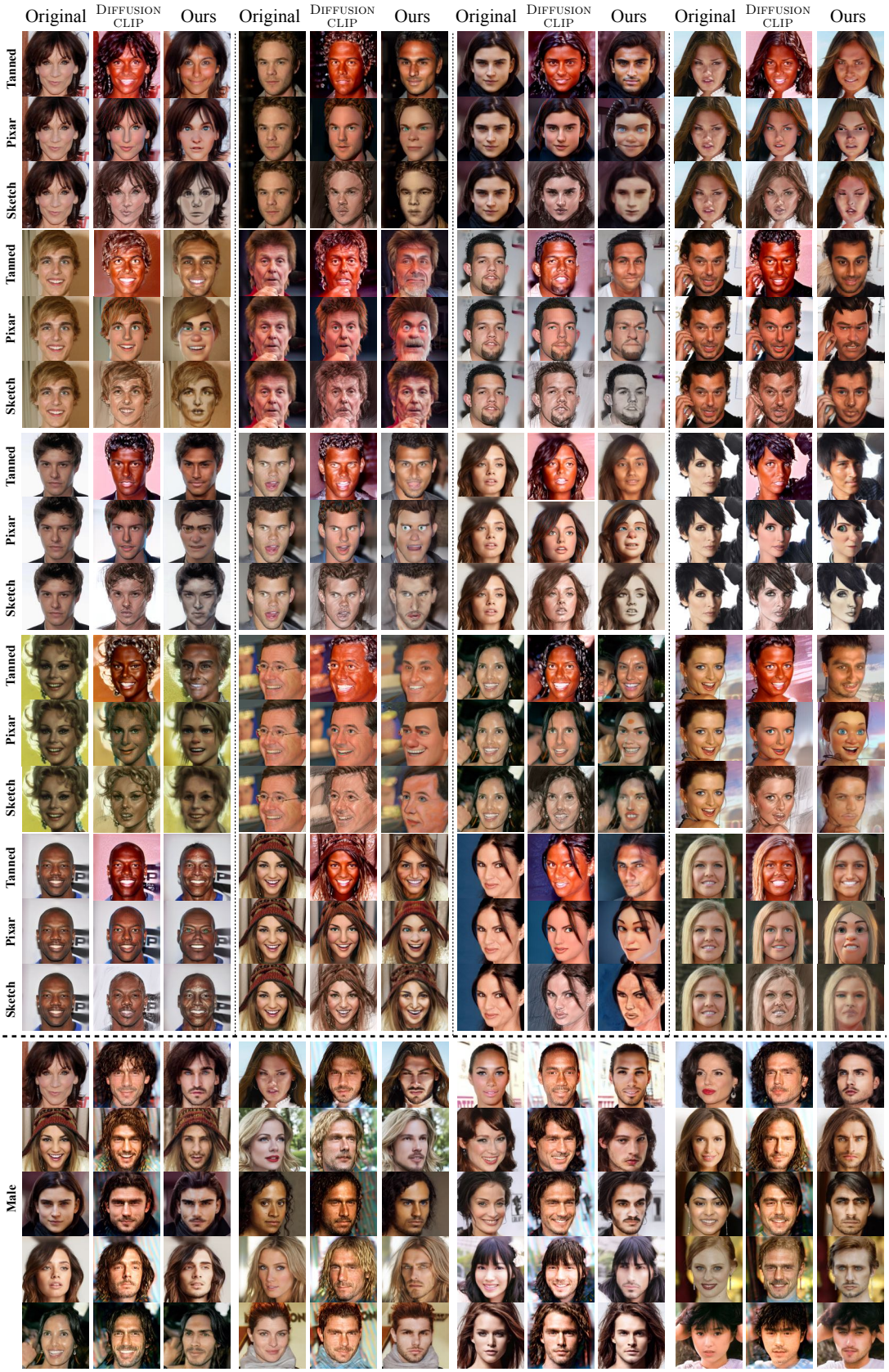

Figure 10. Generated images for subjective evaluation on Celeb-A dataset. Different from other attributes, 20 female images are used as source images for the attribute "male".

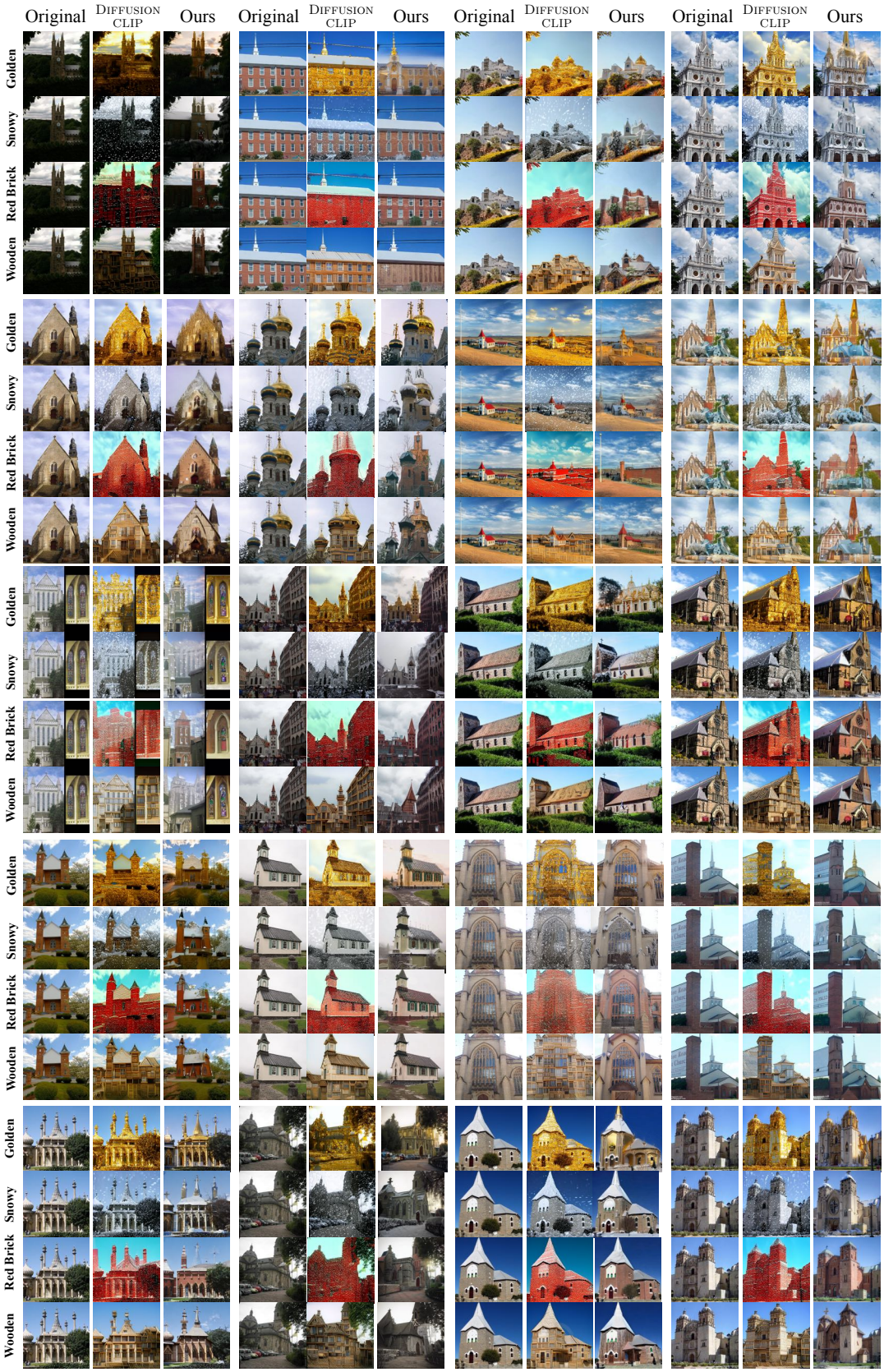

Figure 11. Generated images for subjective evaluation on LSUN-church dataset.

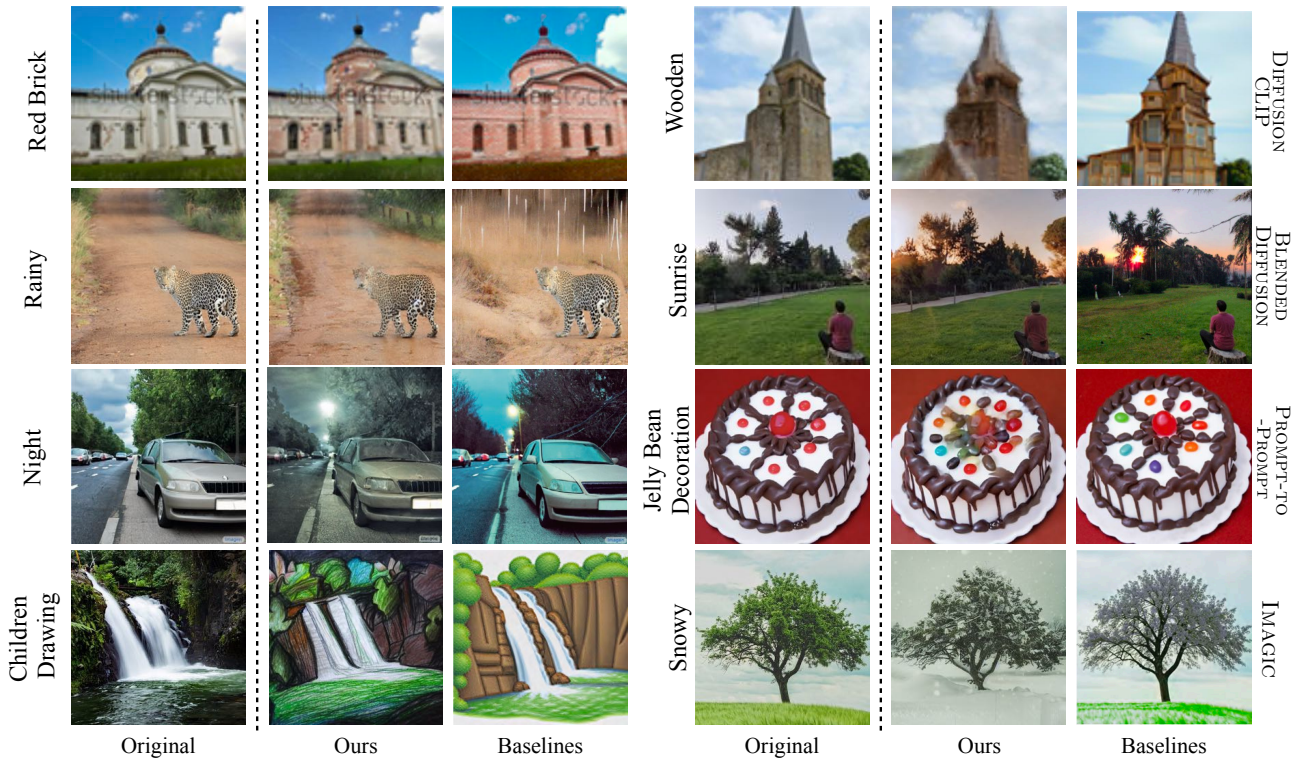

Figure 12. Comparisons with baselines. The source image and corresponding baseline result are taken from the original papers. Each row illustrates the comparison with one particular baseline.

become more colorful and bright.

The variations of $\boldsymbol{c}^{(0)}$: Lastly, we examine the effects of varying $\boldsymbol{c}^{(0)}$. We fix the way of appending target attribute descriptions (concatenated by comma) in $\boldsymbol{c}^{(1)}$, and we examine three variants of $\boldsymbol{c}^{(0)}$, including a short description, a longer description by adding non-informative words, and a description generated by the state-of-the-art image captioning model [75]. Results are shown in Fig. 18. To better compare the effects of variations on $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$, we use the same input images and target attributes as in Fig. 17. From this figure, we observe the image editing results are largely robust to different choices of $\boldsymbol{c}^{(0)}$, except that when using the outputs from the captioning model, the editing effects are sometimes not significant. For example, in the last row, the anime style is not shown in the last image. One possible reason of failure in this case is that the generated caption is long and contains many details, which outweigh the target attribute description.

H. Effects of Varying Optimization Images

We also investigate if our method is robust to the images used for optimization. We examine whether a successful disentanglement depends on (1) the choice of a particular image used for optimization; and (2) number of images used for optimization. Fig. 19 illustrates the results for the first factor. For each attribute, we optimize $\lambda_{1:T}$ on 3 different images separately. We then apply the resulting $\lambda_{1:T}$ on the same unseen image and compare their transferring results. As can be seen, different optimization images result in similar transferring results, showing the robustness of our method to the choice of optimization image. We further quantitatively measure the similarity of optimized combination weights by calculating their cosine similarity. As shown in Fig. 20, we visualize the similarity of $\lambda_{1:T}$ between every pair of optimization images. The high similarity score again demonstrates our method's robustness to the choice of optimization image.

The second factor is illustrated in Fig. 21, where for each target attribute, we optimize $\lambda_{1:T}$ on 1, 3, and 5 images respectively and show their transferring results on unseen images. We observe that optimizing on more images leads to better disentanglement. For example, when optimizing on 5 images, the identity of the person is better preserved (*e.g.,* the badge in the first row and the beard in the second row). We therefore use 5 optimization images in Sec. 4.1.

I. Computation Cost

In this section, we report the computation cost of our method and compare with other baselines in terms of time and memory costs. Specifically, we measure the cost to edit one image on a single Nvidia RTX A6000 GPU, and we report the statistics in Table 4. In the table, "Ours-learning"

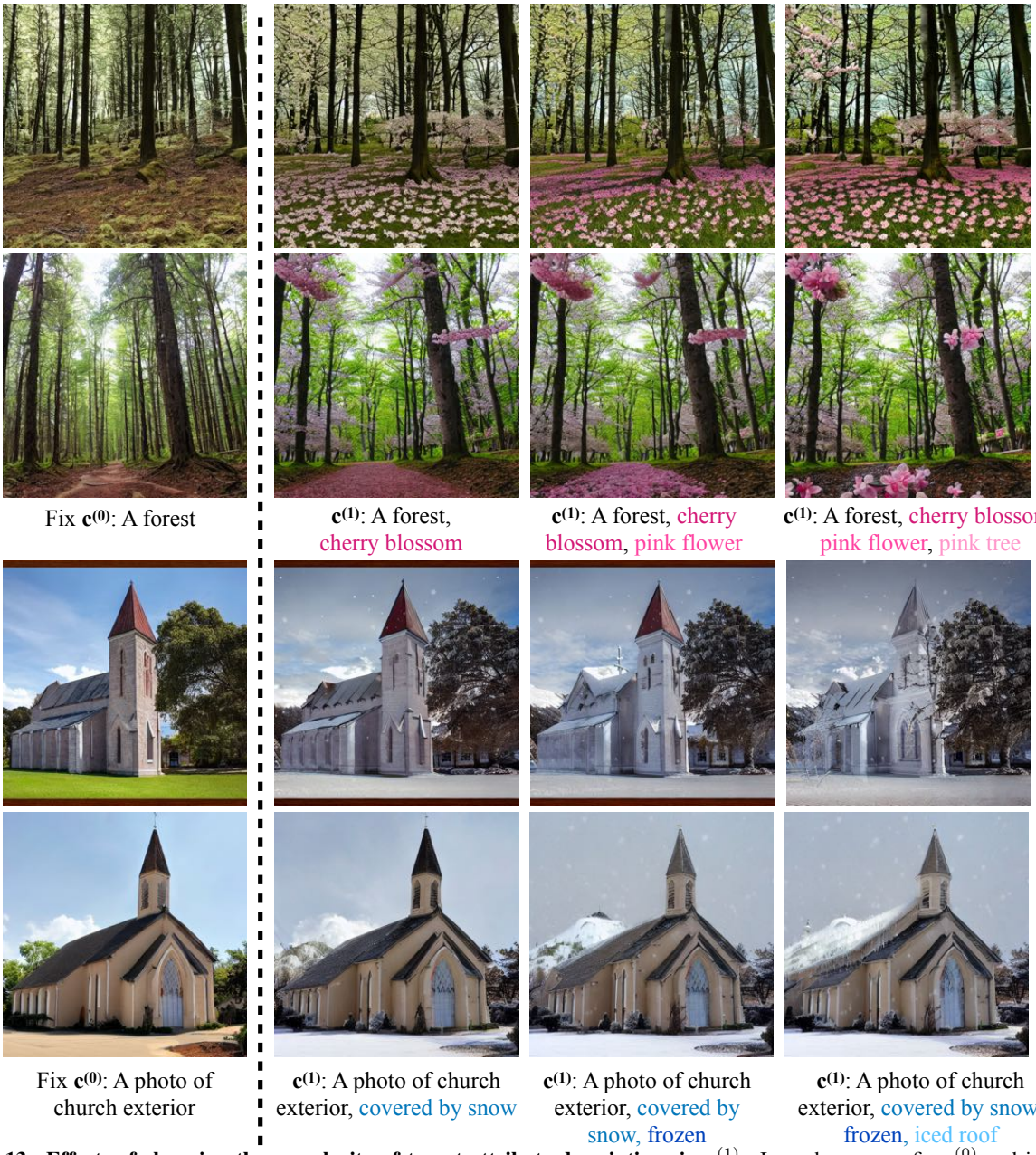

Figure 13. Effects of changing the complexity of target attribute descriptions in $\boldsymbol{c}^{(1)}$. In each row, we fix $\boldsymbol{c}^{(0)}$ and increase the complexity of attribute description by adding more correlated modifiers in $\boldsymbol{c}^{(1)}$.

denotes optimizing combination weights λ; "Ours-edit" denotes applying learned λ to other images. We observe that our method is faster than others except for two that do not need fine-tuning. Besides, once λ is learned, we can apply it to other images to achieve a similar editing effect efficiently.

Method	Time	Memory	Fine-tune?
`StyleCLIP`[53]	0.22 min	7 GB	N
`StyleGAN-NADA`[17]	5 min	9 GB	Y
`BlendedDiffusion`[8]	8 min	17 GB	N
`DiffusionCLIP`[35]	6 min	23 GB	Y
`Prompt-to-Prompt`[20]	0.18 min	8 GB	N
`Imagic`[34]	Code not available		Y
Ours-learning	5 min	42 GB	N
Ours-edit	0.13 min	8 GB	N

Table 4. The computation costs of different methods. "Fine-tune" means if the generator needs to be fine-tuned.

J. Limitations

While our method can modify a wide range of attributes, we find that some attributes are harder to be disentangled. Specifically, as discussed in Table 1, our method has diffi-

culties performing "small edits" on images, such as adding toppings to a cake or adding hats to a person in a portrait. When editing these attributes, our observation is that those small target attributes usually correlate with other parts of the images, thus edits performed on these attributes will also change the irrelevant parts of the images. For example, in the bottom panel of Figure 4, we observe that when the target attribute involves small objects, the model tends to also change some other correlated attributes, such as the style of the cake or the identity of the person. This may be ascribed to the model's weaker control of finer-grained details.

K. Effects of Partially Combined Prompts

In this section, we review the way we combine two text prompts in detail, from where we investigate an alternative way of *partially* combining text prompts and contrast it with our original combining methods.

Recall that in Sec. 3.3, we discussed how we combined a given pair of $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$ at denoising step t:

$$\boldsymbol{c}_t = \lambda_t \boldsymbol{c}^{(1)} + (1-\lambda_t)\boldsymbol{c}^{(0)} \equiv \boldsymbol{c}_t^{(\lambda)}. \tag{15}$$

Here, $\boldsymbol{c}^{(0)}$ is the embedding of a style-neutral description (*e.g.*, "A forest"), and $\boldsymbol{c}^{(1)}$ is the embedding of a description with an explicit style (*e.g.*, "A forest, cherry blossom"). When encoding these descriptions, both texts are padded to the same token length T and embedded by a text encoder. This allows us to directly weighted-sum the two sentences token-wise. Specifically, denote $\boldsymbol{c}^{(0)} = [\boldsymbol{c}_n, \boldsymbol{p}]$, $\boldsymbol{c}^{(1)} = [\boldsymbol{c}'_n, \boldsymbol{c}_a]$, where $\boldsymbol{p}$ is the embedding of the pad sequence, $\boldsymbol{c}_a$ is the extra description (*e.g.*, "cherry blossom"), $\boldsymbol{c}_n \approx \boldsymbol{c}'_n$ are the embeddings of the original neutral description, which are different because their contexts are different. Then the weighted sum becomes

$$\boldsymbol{c}_t^{(\lambda)} = [\lambda \boldsymbol{c}'_n + (1-\lambda)\boldsymbol{c}_n, \lambda \boldsymbol{p} + (1-\lambda)\boldsymbol{c}_a]. \tag{16}$$

Now, we analyze an alternative way to *partially* combine text prompts embedding $\boldsymbol{c}^{(0)}$ and $\boldsymbol{c}^{(1)}$. To start with, notice that if we assume $\boldsymbol{c}_n = \boldsymbol{c}'_n$ in Eq. 16, then $\boldsymbol{c}^{(\lambda)}$ and $\boldsymbol{c}^{(0)}$ would differ only at the second half. Thus, the *attention score* summed over $\boldsymbol{c}^{(\lambda)}$ would exactly equal that over $\boldsymbol{c}^{(0)}$ plus the difference of the second half renormalized. In other words, similar to prompt-to-prompt, we can just re-compute the attention scores of the second-half prompts.

In practice, we have $\boldsymbol{c}_n \approx \boldsymbol{c}'_n$. We conduct experiments to validate that, by computing the partially combined prompts, the edited image would still have similar quality. In the experiment, we edit 300 images in CelebA-HQ and LSUN-Church, and we measure the CLIP Similarity (S_{dir}) between the edited image and the target attribute. The result of partially combining prompts gives a CLIP score of 0.119, while our original method has a CLIP score of 0.112. This result illustrates that partially combined prompts give a similar performance compared with our original method.

L. More Editing Applications

Attribute Preservation In many cases, by explicitly stating an attribute in neutral description $\boldsymbol{c}^{(0)}$, we can better preserve the attribute that would otherwise be altered, as demonstrated in Fig. 14 Left. Here, sequentially adding "*yellow skin*" and "*no makeup*" in $\boldsymbol{c}^{(0)}$ incrementally preserves contents in the original image. However, this method would fail when the DDIM is inherently unable to disentangle the attribute to be kept and that to be altered. In these cases, either the original attribute cannot be preserved or the target attribute won't change, as shown in Fig. 14 Right.

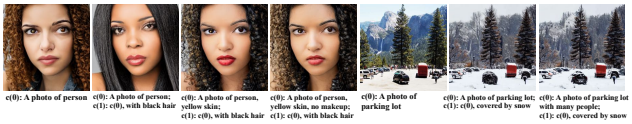

Figure 14. Effects of adding attributes in neutral description.

Attribute Removal We can also extend our method to remove attributes in an image. For example, to remove a smile from a person, we need to specify the smile in $\boldsymbol{c}^{(0)}$ (*i.e.,* "A photo of person, smiling"), and then *exclude* it in $\boldsymbol{c}^{(1)}$, (*i.e.,* "A photo of person"), as shown in Fig. 15. We show that our method can remove both global attributes such as snow and local attributes like leaves and facial expressions.

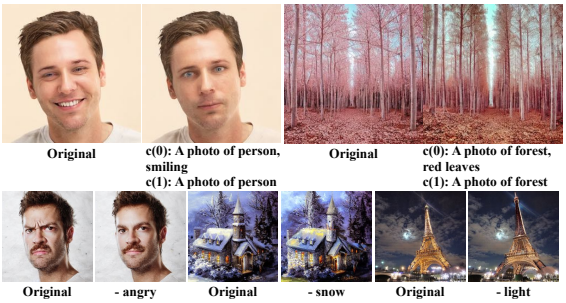

Figure 15. Results of removing attributes.

Sequential Editing To achieve sequential editing, we take the output from the last editing step as the input image, use the same $\boldsymbol{c}^{(0)}$ as the last editing step (*e.g.,* "A forest"), and add the target attribute of the current step in $\boldsymbol{c}^{(1)}$ (*e.g.,* "A forest, children drawing style"). Fig. 16 shows that resulting images gradually incorporate attributes from small objects like cherry blossom to global styles.

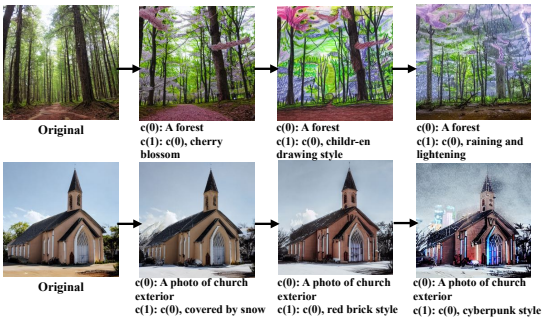

Figure 16. Sequential editing effects.

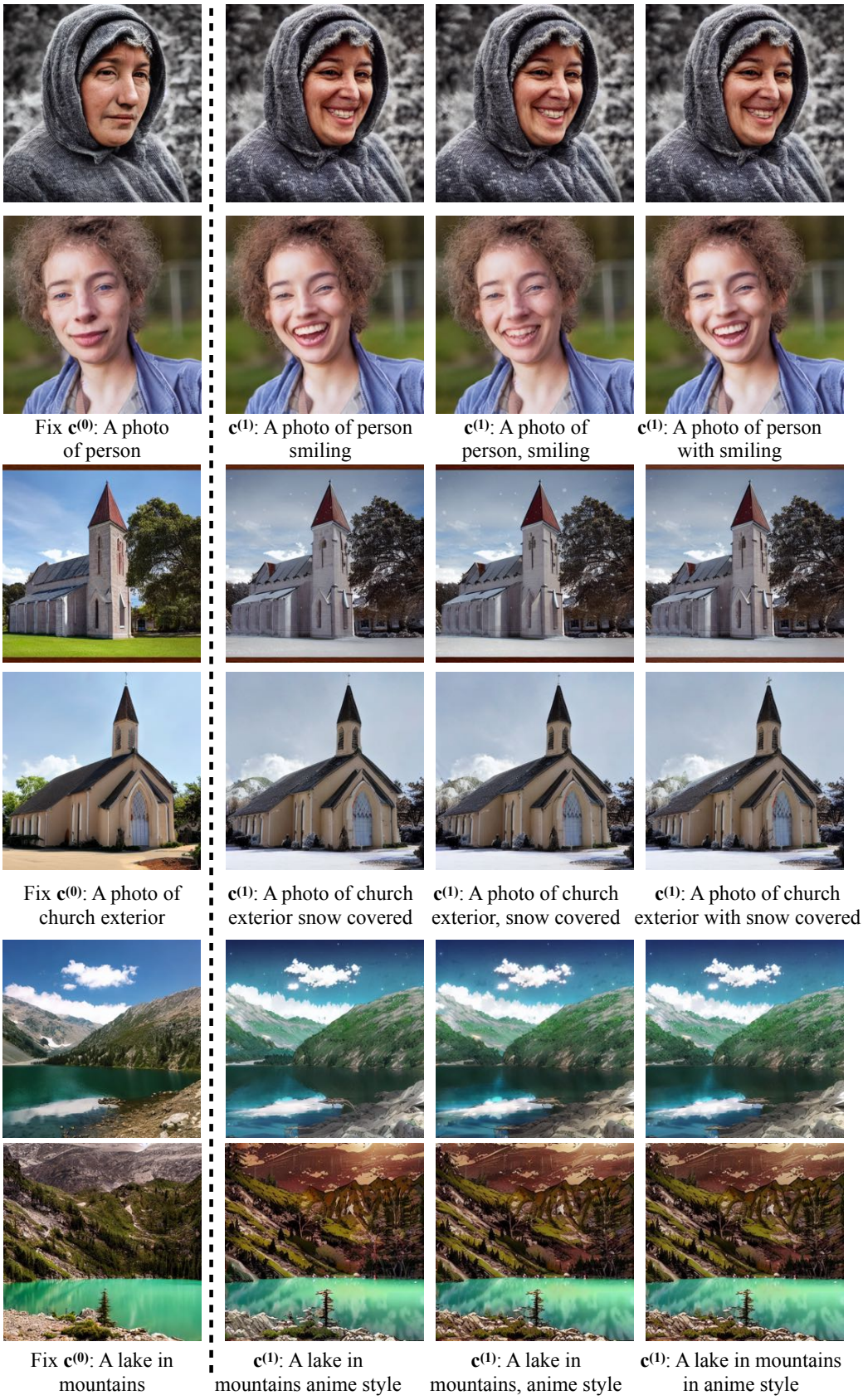

Figure 17. Effects of varying the way that $c^{(1)}$ appends the target attribute description. In each row, we fix $c^{(0)}$ and change the way of concatenating target attribute descriptions in $c^{(1)}$.

Figure 18. Effects of choosing different style-neutral descriptions $\boldsymbol{c}^{(0)}$. For each row, we fix the target attribute description and consider three variations of $\boldsymbol{c}^{(0)}$ that describe the same object: a short description, a longer description by adding non-informative words, and a description generated by image captioning model.

Figure 19. Results of using different optimization images. For each attribute, I_1, I_2, I_3 represent 3 different optimization images. **Left column:** images used for training and their corresponding disentanglement results. **Right column:** disentanglement results of applying the learned weights to unseen images.

Figure 20. Cosine similarity between combination weights $\lambda_{1:T}$ optimized on different images. For each attribute, I_1, I_2, I_3 represent 3 different optimization images.

Figure 21. Results of using different numbers of images for optimization. The left column is the original unseen image, and the remaining columns demonstrate the results of applying the weights optimized on 1, 3, and 5 images respectively.