Supplementary Material: ColorPeel: Color Prompt Learning with Diffusion Models via Color and Shape Disentanglement

Muhammad Atif Butt[1], Kai Wang[1⋆], Javier Vazquez-Corral[1,2], and Joost van de Weijer[1]

[1] Computer Vision Center, Spain
[2] Universitat Autonoma de Barcelona, Spain
{mabutt,kwang,jvazquez,joost}@cvc.uab.es

A Learning Colors with Existing Methods

As elucidated in the paper, current T2I diffusion models offer users the ability to generate objects in desired colors by incorporating linguistic color descriptions in the prompts. Although these diffusion models have showcased remarkable capabilities in generating images from textual prompts, they encounter challenges in accurately reproducing specific colors. One of the primary reasons for this limitation is the broad spectrum of colors encompassed by linguistic color names (e.g., pink, blue, green), which can represent numerous combinations of hues and shades. For example, color blue alone encompasses various shades such as navy blue, sky blue, and royal blue. Consequently, generated images may not exactly match the intended color. As depicted in Fig. S1(a) and Fig. S1(c), even when prompted with standard color names, stable diffusion model [6] and Rich-Text method [2] struggle to distinguish between different color variants.

Another method of specifying precise colors is by including exact color values or hex color codes in the prompts to synthesize objects. The results depicted in Fig. S1(b) reveal that diffusion model struggles to interpret hex or RGB values directly, as these models lack embeddings for such representations. Next, we consider T2I personalization methods to learn color embeddings based on new text tokens. We use seminal T2I personalization baselines i.e., TI [1], DB [7], and CD [5]. First, we learn color embeddings from fully-colored images using RGB/hex values. However, as shown in Fig. S1(d), these methods employ a naive transfer learning approach. While they are able to learn shapes or objects, they fail to learn from plain color images. To gain further insight into this issue, we extract attention maps from final timestep of the training process, as shown in Fig. S2. It is evident from Fig. S2(a) that the employed personalization methods struggle to focus on color region when entire image is colored. To address this limitation, we generate basic 2D/3D shapes using RGB/hex values. With this training setup, we observe that employed methods, while still performing poorly, show some focus on color regions. However, as demonstrated in the main paper, these methods tend to mix colors due to non-aligned token initialization.

⋆ Corresponding Author

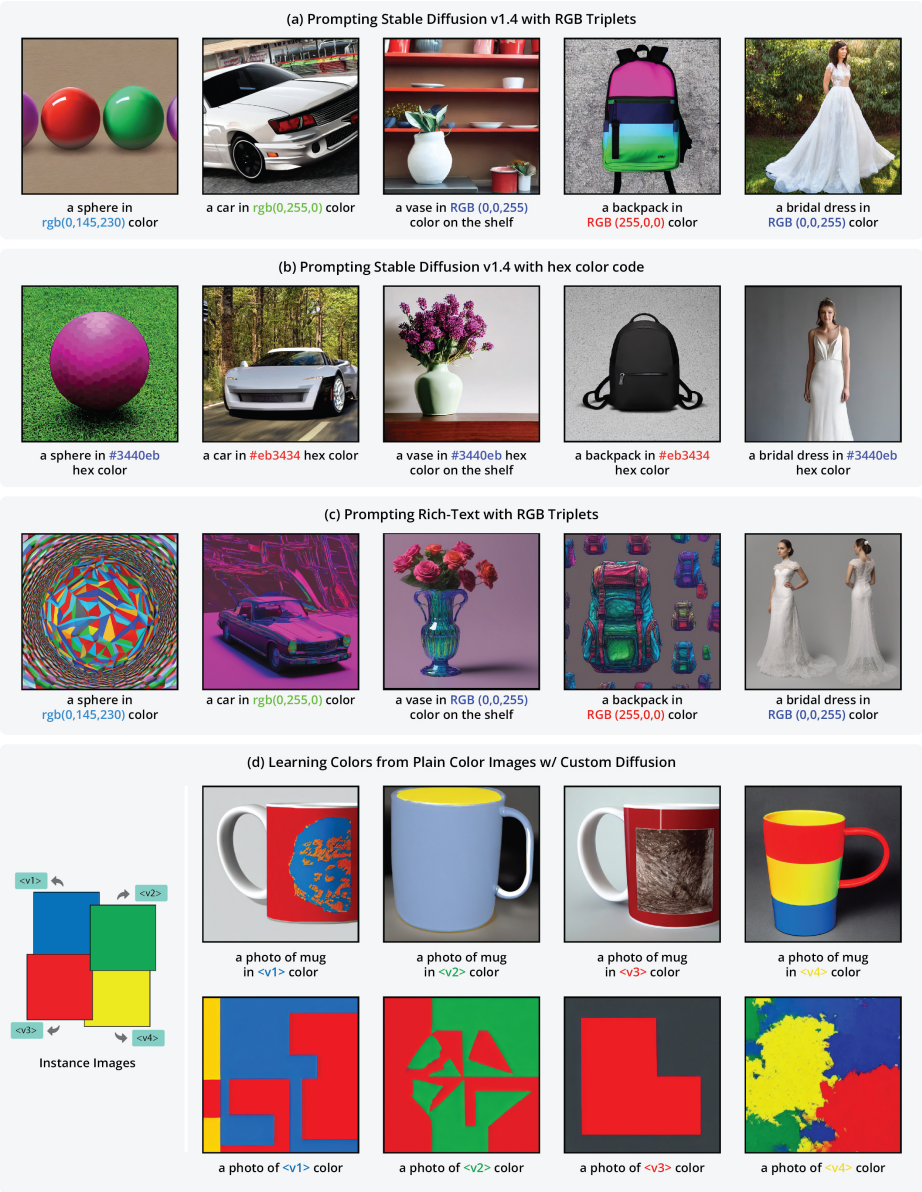

Fig. S1: Visualizing failure compositions with existing methods. It can be noticed from the results that *Stable Diffusion fails to comprehend* ***(a)*** ***(c)*** *RGB Triplets and* ***(b)*** *hex color code directly.* Whereas, multi-concept personalization method i.e., **(d)** Custom Diffusion struggles to learn color embeddings from plain images, which results in color intermixing in the generated outputs.

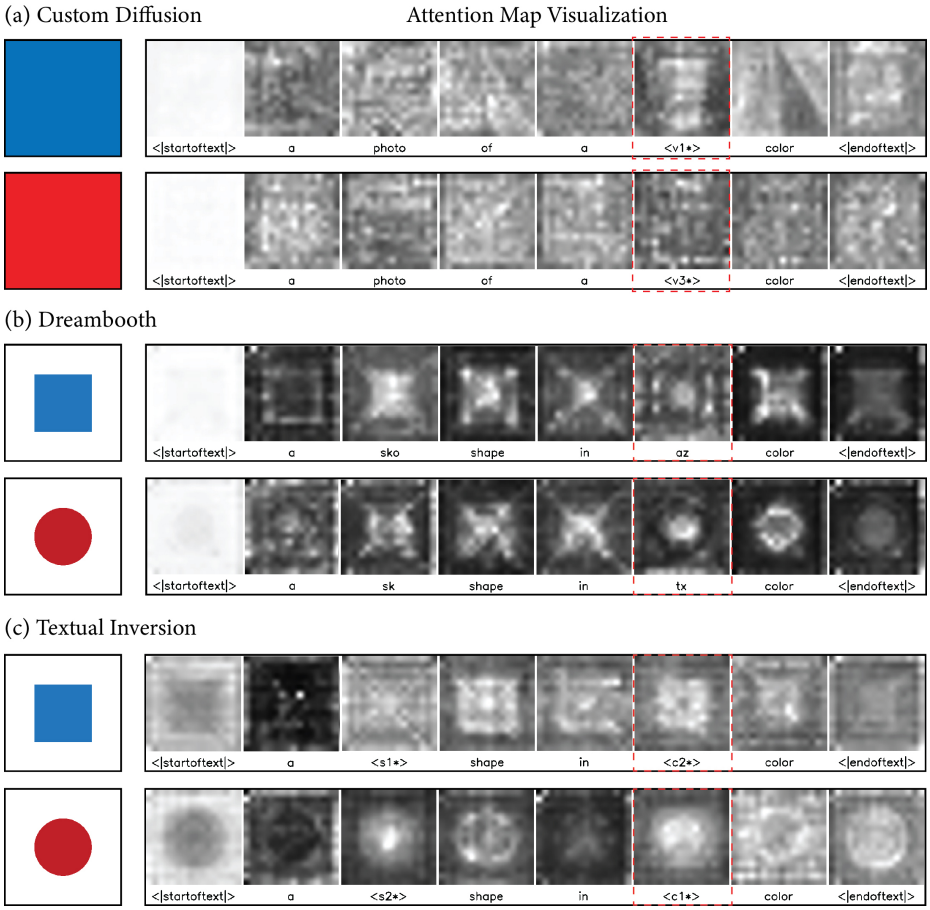

Fig. S2: Visualizing cross-attention maps from last time-step. **(a) Custom Diffusion**—fails to focus on the color in case of fully colored images. In particular, *<v1*>* and *<v3*>* which are supposed to learn colors, are not precisely learning. **(b) Dreambooth**—with basic colored shapes in the train images, the shape and color text tokens i.e., *sko*, *sk*, *az*, and *tx* are focusing on the color region, however, overlapping with the other tokens. Similarly, in **(c) Textual Inversion**, the new text-tokens *<s1*>*, *<s2*>*, *<c1*>*, and *<c2*>* are overlapping with other tokens. This can be one of the reasons which lead to the inaccurate color syntheses in the generated images.

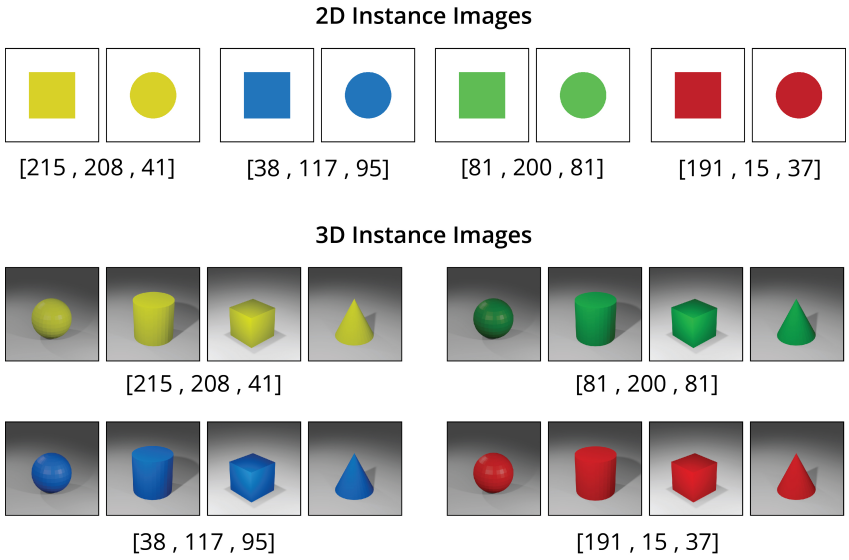

Fig. S3: 2D and 3D instance images for the coarse-grained color learning task

B Experiments

B.1 Implementation Details

We train our *ColorPeel* with a batch size of 1 and a learning rate of 10^{-5}. For coarse-color learning, we train the model for 1500 steps. However, we increase the training steps to 6000 steps for fine-grained color learning. The source code is provided in the supplementary material and will be made publicly available upon acceptance. It is important to mention that, we follow the same training schema for learning coarse and fine-grained colors as discussed in the section 3, for analyzing the disentanglement between the shapes and colors, along with the transferability of colors to unknown shapes/objects. To ensure faster convergence, we only back-propagate the valid regions' loss combined with the cross-attention alignment loss to improve learning.

B.2 Dataset Details

As discussed in the previous section, colors have a countless range of combinations, due to different hues and shadings. Therefore, it is also crucial to prepare training-data/instance-images in the precise desired color. To handle this challenge, we introduce a data synthesizer in our *ColorPeel*, which acts as a processing step in the pipeline. In particular, our method is capable of creating basic 2D and 3D shapes, given the desired RGB-Triplet, hex-Code, or Color-Coordinates. In 2D-Instances, rectangle and circle shapes are used, whereas, in the case of 3D, five simple 3D-shapes are used including *sphere*, *cube*, *cylinder*, *cone*, and *hexagon*. It is important to mention here that 3d shapes provide more accurate

Fig. S4: 3D instance images for the fine-grained color prompt learning task.

representation, allowing for better understanding of color variations in different spatial dimensions. Therefore, we ensure that our *ColorPeel* can learn color embeddings with both the 2D and 3D instance images. For 3D instance image creation, initially object files containing a scene graph with the shape positioned in the center of the plane, with three directional area lighting to ensure appropriate visibility, are created in the Blender. Our *ColorPeel* can render these shapes, given the RGBs in real-time.

For color prompt learning, we design two color learning tasks: (i) coarse-grained color learning, which contains four basic colors—red, green, blue, and yellow, and (ii) fine-grained color learning which covers 18 colors related to less common color names, including 'salmon','beige', etc. The sample instance 2D and 3D images for coarse-grained color learning task are shown in the Fig. S3,

Table S1: RGB values of colors used to generate coarse and fine-grained color sets. Coarse-grained color learning tasks covers four basic colors, whereas, fine-grained color learning task includes 18 colors.

Color	RGB Code
Coarse-grained Color Set	
Red	191, 15, 37
Green	81, 200, 81
Blue	38, 117, 195
Yellow	215, 208, 41
Fine-grained Color Set	
Red	255, 0, 0
Maroon	128, 0, 0
Orange	255, 165, 0
Coral	255, 127, 80
Pink	255, 192, 203
Green	0, 128, 0
Lime	0, 255, 0
Olive	128, 128, 0
Blue	0, 0, 255
Navy	0, 0, 128
Cyan	0, 255, 255
Turquise	64, 255, 208
Indigo	75, 0, 130
Purple	128, 0, 128
Yellow	255, 255, 0
Gold	255, 215, 0
Bisque	255, 228, 196
Wheat	245, 222, 179
Beige	245, 245, 220

and the 3d instance images are demonstrated in Fig. S4 and the list of the color names along with corresponding RGB values are enlisted in Table S1.

B.3 Additional Qualitative Results

The results our method *ColorPeel* for the coarse color and fine color learning tasks are demonstrated in Fig. S5, and Fig. S6, respectively. In addition to generating high-quality images, *ColorPeel* also showcases effective personalization of diverse elements, including customizing attire like clothing, footwear, gloves, and glasses, as well as toys and objects within different settings. In the next step, we carefully designed the prompts to evaluate the color transferability in terms of consistency and fidelity to a wide range of attributes. These attributes span from image background color customization to personalized elements such as clothing, eye, hair colors, as well as other objects like chair, dustbin, etc.

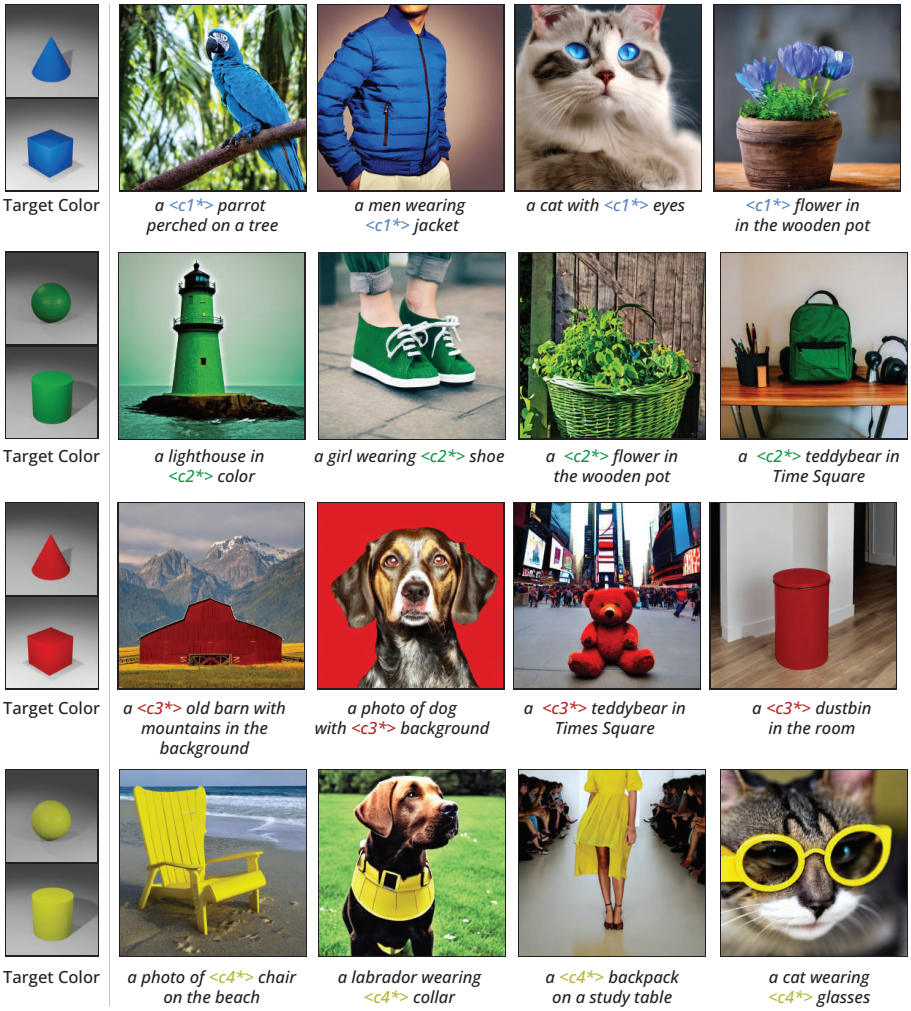

Fig. S5: Qualitative color generation results of the coarse-grained color learning task.

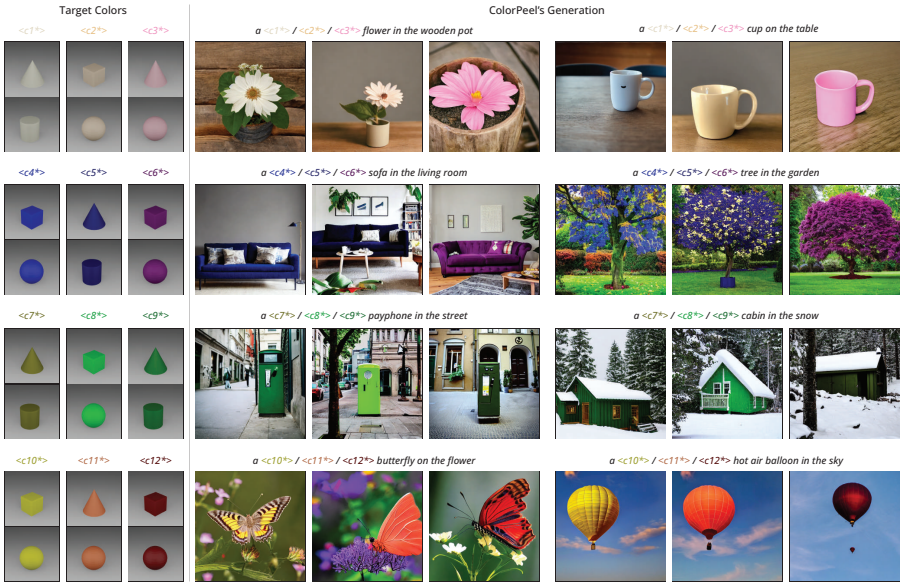

Fig. S6: Qualitative color generation results of the fine-grained color learning task.

B.4 Prompt Templates

Evaluation prompt templates. Here is the list of text prompts used in evaluating our proposed method and comparing it with the baseline methods i.e., Stable Diffusion, Custom Diffusion [5], Textual Inversion [1], Rich Text [2], and Dreambooth [7].

- a {color} bowl on the table
- a {color} bowling ball in a bowling alley
- a {color} plate on the table
- a {color} vase on the shelf
- a women wearing {color} pants
- a {color} teddy-bear in Time Square
- a {color} snooker ball on the table
- a {color} parrot perched on a tree
- a {color} sofa in living room
- a {color} rose blooming in a wooden pot

Training prompt templates. Initially, we tried optimizing the new text-tokens with multiple training prompt examples, enlisted below.

- a photo of <s*> shape in <c*> color
- a <s*> shape in <c*> color
- a <c*> colored <s*> shape
- a photo of <c*><s*>

However, *ColorPeel* achieved better results with single prompt i.e., "*a photo of <c*> shape in <s*> shape*" for each instance image, where <s*> and <c*> are new text-tokens, corresponding to shape and color, respectively.

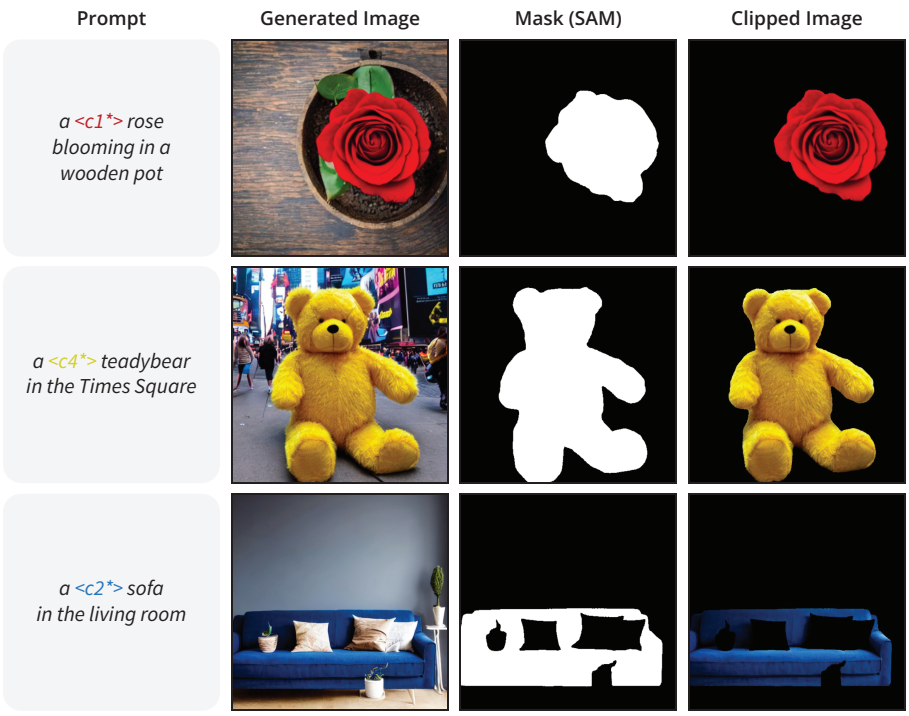

Fig. S7: Mask Generation for quantitative evaluation. Given an image generated with a prompt, we compute a mask using Segment Anything model [4] in order to consider only those pixels that belong to the object.

B.5 Evaluation pipelines

Fig. S7 shows the masking computation. An image is generated given a prompt. Then, the Segment Anything Model [4] is used to compute the mask of the generated object —in this figure the red rose, the yellow teddy-bear, or the blue sofa—. This mask is then used to consider only the object's pixels for the computation of the metrics. This said, for some objects they might be some small parts that are not supposed to have the desired color; for example the eyes of the teddy-bear. For this reason, in the main paper we computed some measures considering also the 10%, or 50% of most correct pixels inside the mask. Let us remind the reader that our method outperformed all the others in all metrics at under all conditions.

B.6 User study

Fig. S8 shows our user's study setup. In the left, we see a picture of the room, that was completely black and the monitor set to sRGB. The monitor —Fujitsu B-24-8— was the only light source during the experiment. Observers were seated

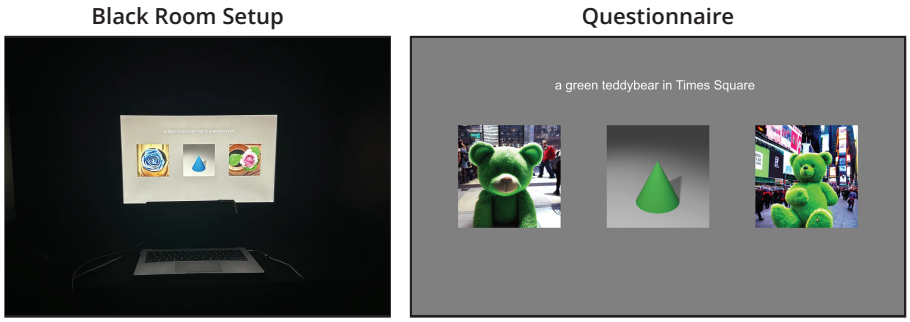

Fig. S8: Setup of our Human Study. *Left*: Observers were sitting in a black room, where the only light was the one provided by the monitor. *Right*: The background of the monitor screen was fixed to middle gray, and the observers needed to select which image —left or right— was a better match considering the prompt and the color shown in the central image.

approximately 60 cm away from the monitor to ensure a 7-degree visual angle. The monitor background was set to middle gray. The monitor displayed a prompt and a central image with the reference color. Left and right from the central image we randomly showed the results for our method and a competing one. The observer needed to select which image from the two represented better the prompt given the color in the middle image. An example of the set-up as seen by the observer is shown in the right part of Fig. S8. A total of 15 observers participated in the study, and none of the authors took part in the study.

B.7 Verification of the Color Prompt Learning

We conduct experiments over coarse-grained concepts using the training schema as shown in Fig. S9 to analyze the learning of color prompts from given colored shapes and their transferability to other objects. To evaluate if our method *ColorPeel* is correctly disentangling the colors from shapes and can transfer to the unknown ones, as devised in training schema, we learn colors and shapes given in coarse-grained training set in c_i^* and s_i^* text-tokens, respectively. The results are illustrated in Fig. S9, we showcase that with only eight images in the 4×4 training scheme, *ColorPeel* can successfully infer the geometries not included in the training set. That further proves the effectiveness of our method *ColorPeel*.

B.8 Image Editing

Here we demonstrate a few examples of image editing following the P2P method [3] with our *ColorPeel*. The corresponding image editing results are shown in Fig. S10, where we successfully modified the color of the objects to our learned colors.

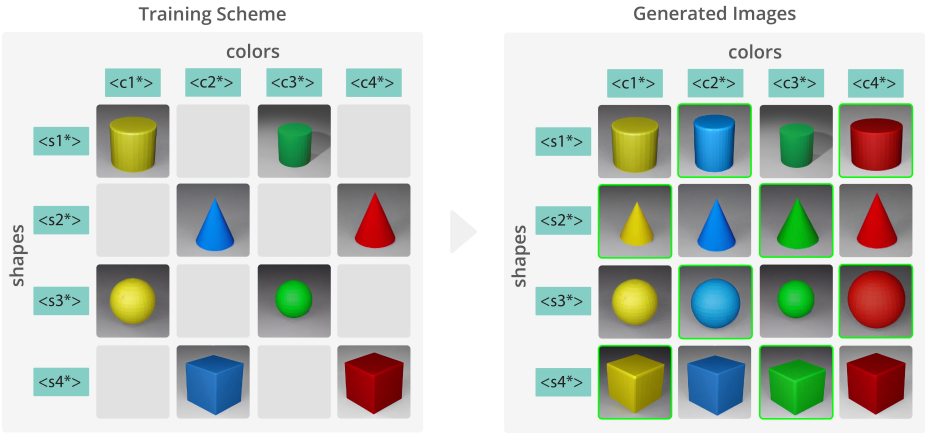

Fig. S9: Training scheme and generating unknown shapes. The instance prompt—"*a photo of <s*> shape in <c*> color*" is used to reproduce the shapes with unknown colors. The generated images are outlined in green border.

B.9 Color Interpolation

Here we demonstrate a few examples of linear interpolation between two newly learned color tokens. The results are shown in Fig. S11 which shows that our *ColorPeel* can represent the colors continuously between the learned color prompts, which ultimately can avoid the training for new colors.

B.10 Additional Qualitative Comparison

We show the additional qualitative comparison between the generated images from baselines including Stable Diffusion (SD), Rich-Text, Textual Inversion (TI), Dreambooth (DB), and Custom Diffusion (CD) against our method *ColorPeel*. The results are shown in Fig. S12 which shows that our *ColorPeel* can learn and transfer better photo-realistic colors as compared to the baselines.

C Ablation Study

We ablate various components of our method to show its contribution, which are demonstrated in Fig. S13. Firstly, we remove cross attention loss (CAA) and train the model with the default baseline settings. The results (see Fig. S13a) demonstrate that the model fails to disentangle the color from the shape, and replicates the shapes while ignoring the target prompt. We also notice that the default transformation techniques (resize, zoom, etc.) in the baseline are highly influencing the reconstruction performance, resulting in inaccurate generation. Secondly, we use fully weighted cross attention loss combined with the reconstruction loss to train the model. The results (see Fig. S13b) demonstrate that model ensures efficient learning and transferability of colors, however, model

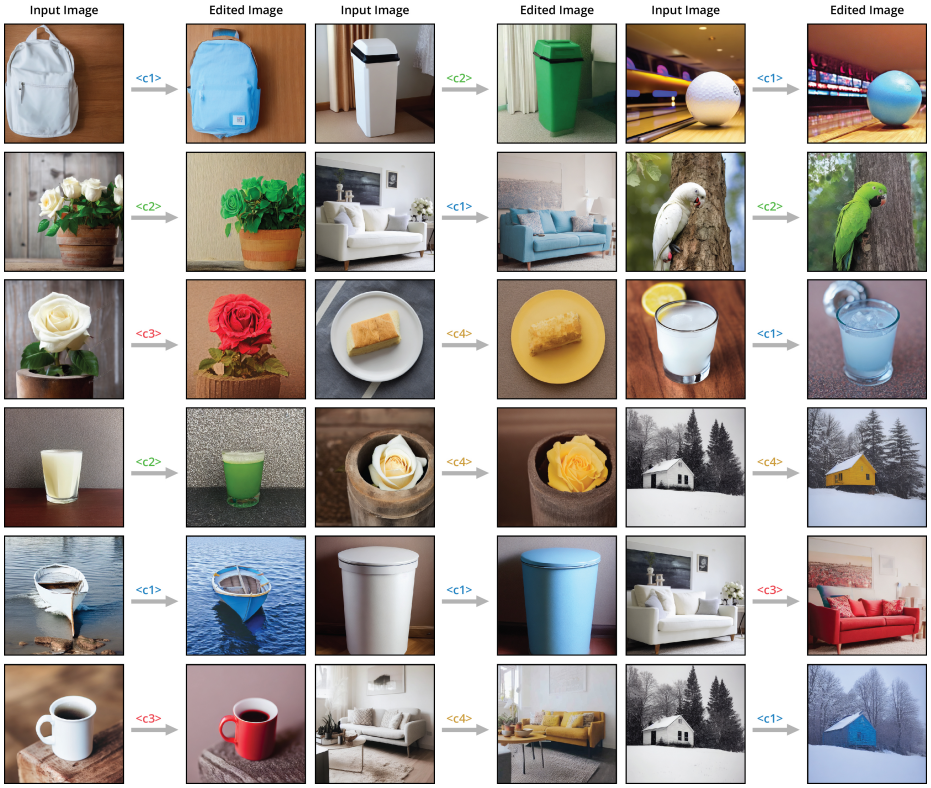

Fig. S10: Demonstrating the results of image editing. <c1*>, <c2*>, <c3*>, and <c4*> corresponds to the embeddings of newly learned colors for the coarse-grained color learning task. The results show that the object colors can be modified given our learned embeddings.

fails in accurate shape reconstruction. Thirdly, we trained the model by scaling down the lambda to 0.7 in CAA loss which improves the color transferability and shape reconstruction as compared to fully weighted CAA loss. Lastly, we show the results of our *ColorPeel* where we train the model by setting lambda to 0.2 in CAA loss. With this setting, our method achieves comparatively better performance in terms of color fidelity and consistency, and shape reconstruction.

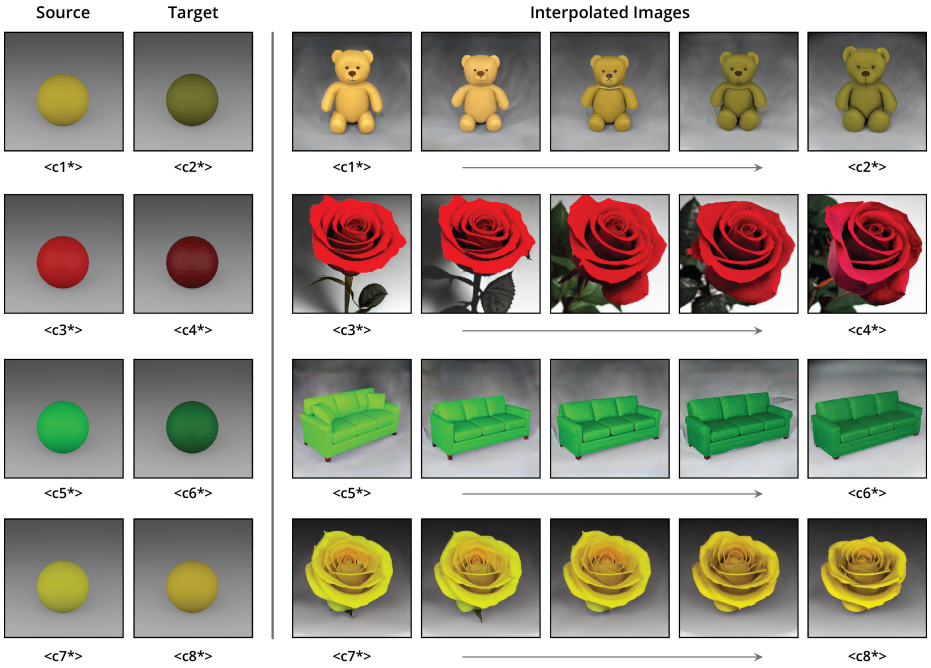

Fig. S11: Linear interpolation between two color tokens.

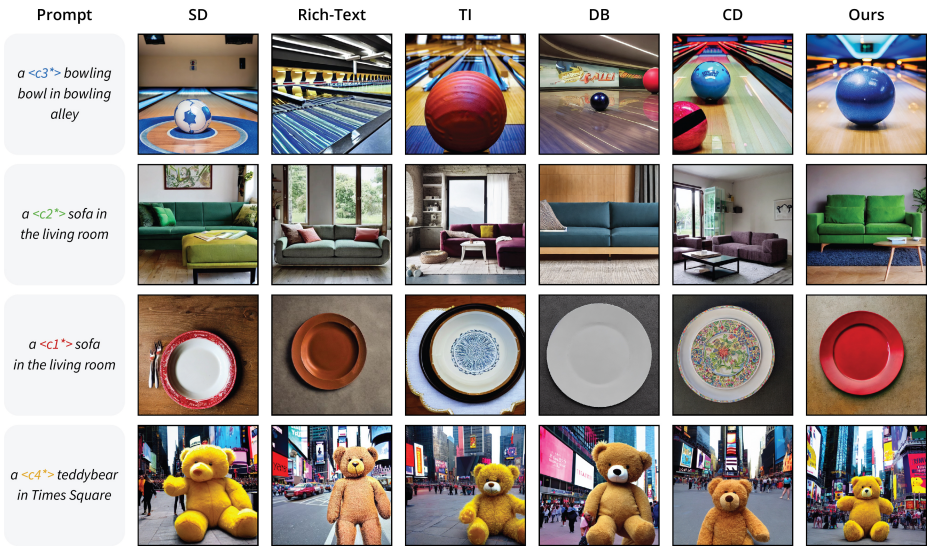

Fig. S12: Demonstrating qualitative comparison of the generated images with the baselines and our method.

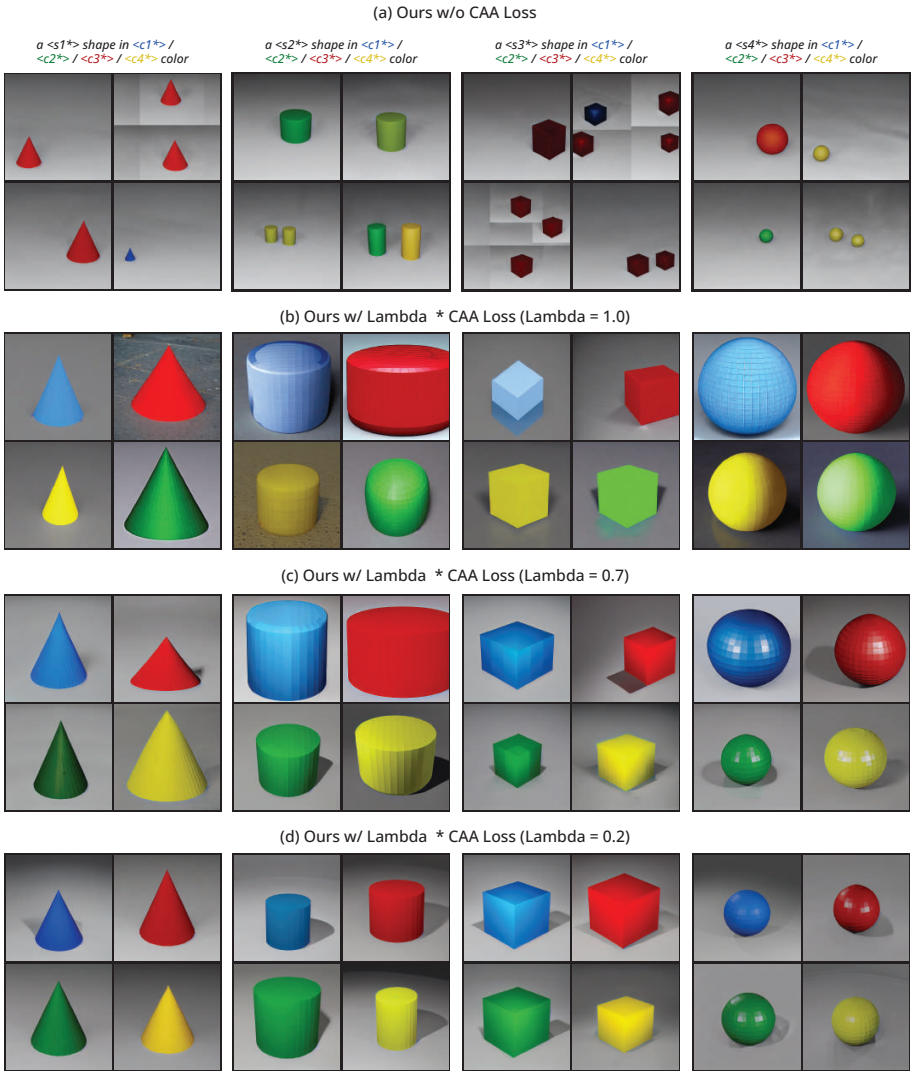

Fig. S13: Ablation Study. We ablate various components of our method *ColorPeel* to demonstrate their contributions.

References

1. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. ICLR (2023)
2. Ge, S., Park, T., Zhu, J.Y., Huang, J.B.: Expressive text-to-image generation with rich text. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7545–7556 (2023)
3. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. ICLR (2023)
4. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. ICCV (2023)
5. Kumari, N., Zhang, B., Zhang, R., Shechtman, E., Zhu, J.Y.: Multi-concept customization of text-to-image diffusion. CVPR (2023)
6. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022)
7. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. CVPR (2023)