LANGUAGE-INFORMED VISUAL CONCEPT LEARNING

AUTHOR RESPONSES

Anonymous authors
Paper under double-blind review

CONTENTS

1 INFERENCE ON REAL-WORLD IMAGES

Despite being trained on images generated by diffusion-based models, the concept encoders generalize well to diverse, complex real-world images, including *unseen* types of object `category`, `material`, and `color` from casual photo captures (Figures 1 to 4) and unseen types of artwork `style` (Figure 5).

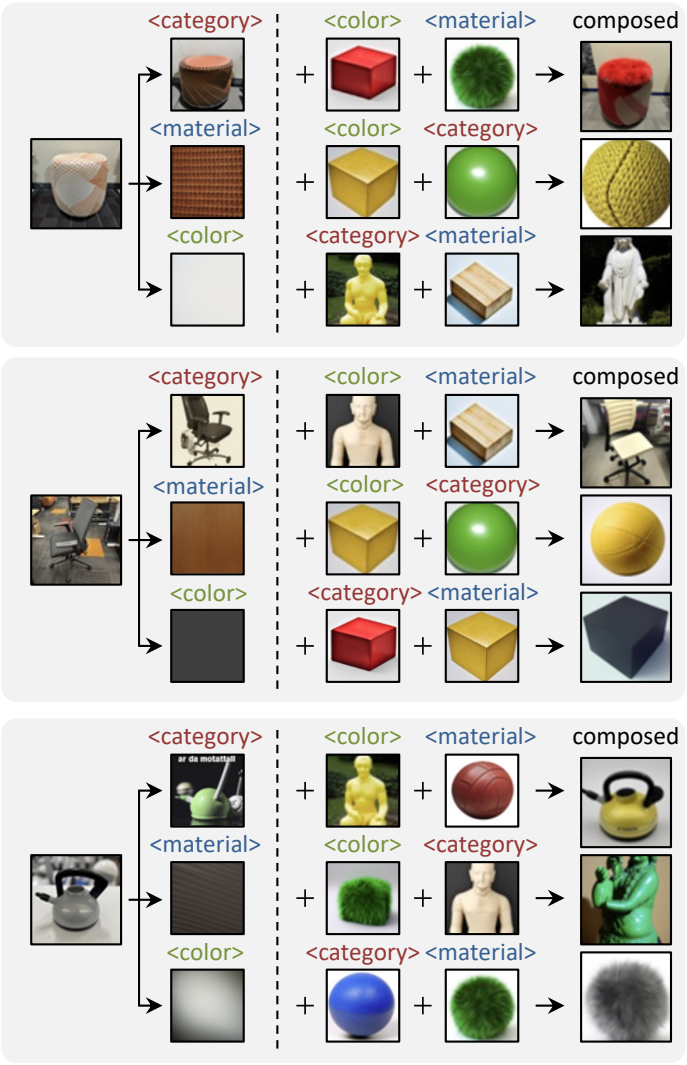

Figure 1: Visual Concept Extraction and Recomposition Results on Real-world Images of various objects.

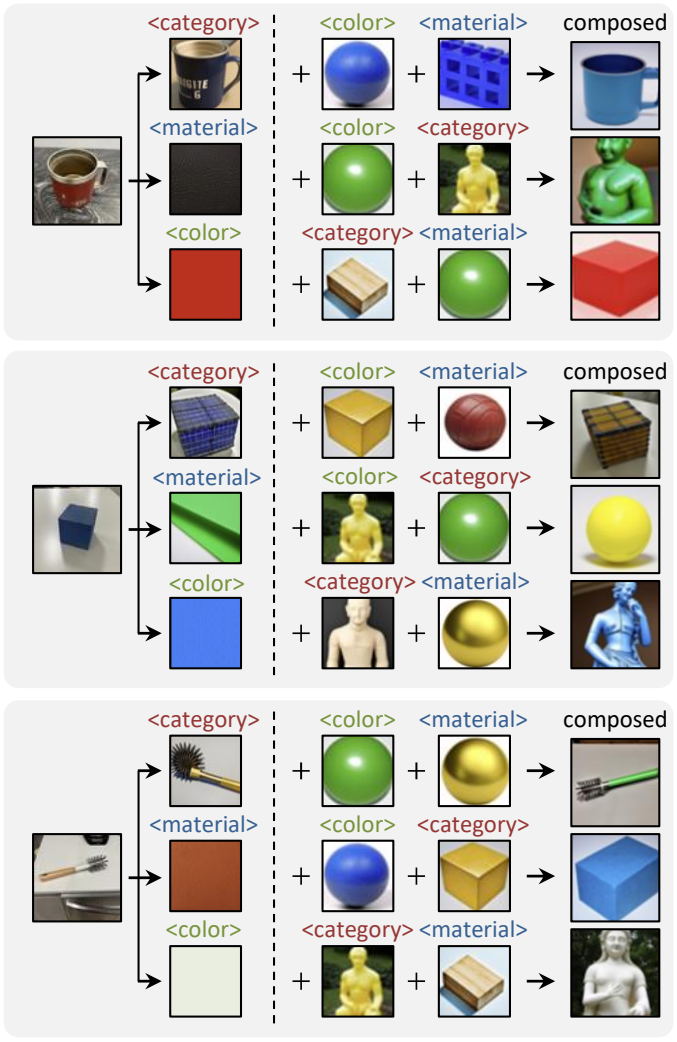

Figure 2: Additional Visual Concept Extraction and Recomposition Results on Real-world Images of various objects.

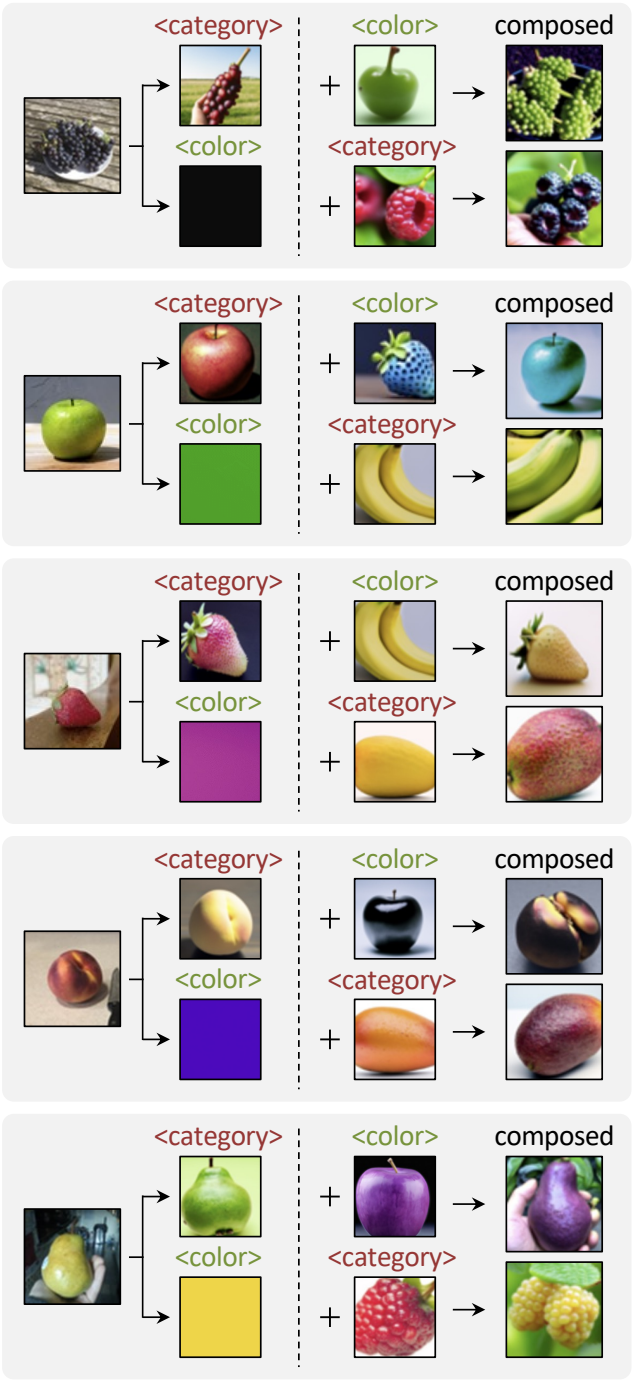

Figure 3: Visual Concept Extraction and Recomposition Results on Real-world Images of various fruits.

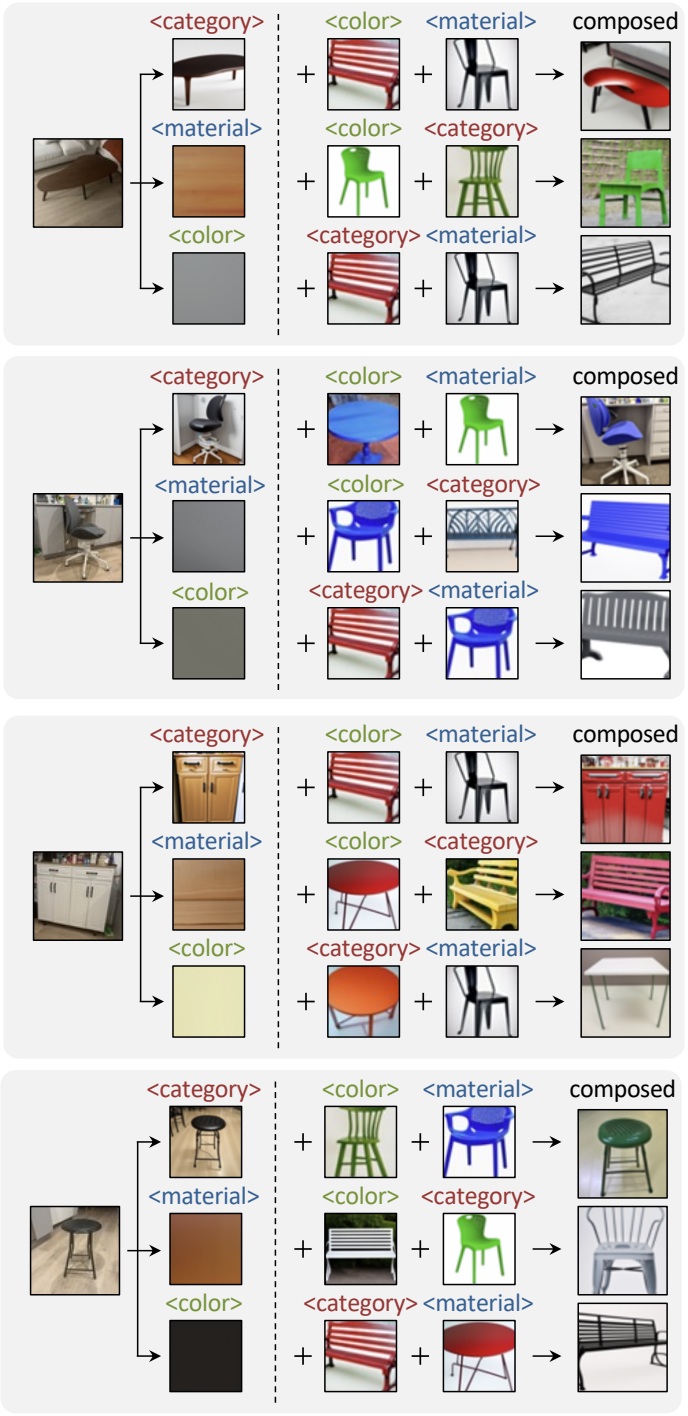

Figure 4: Visual Concept Extraction and Recomposition Results on Real-world Images of various pieces of furniture.

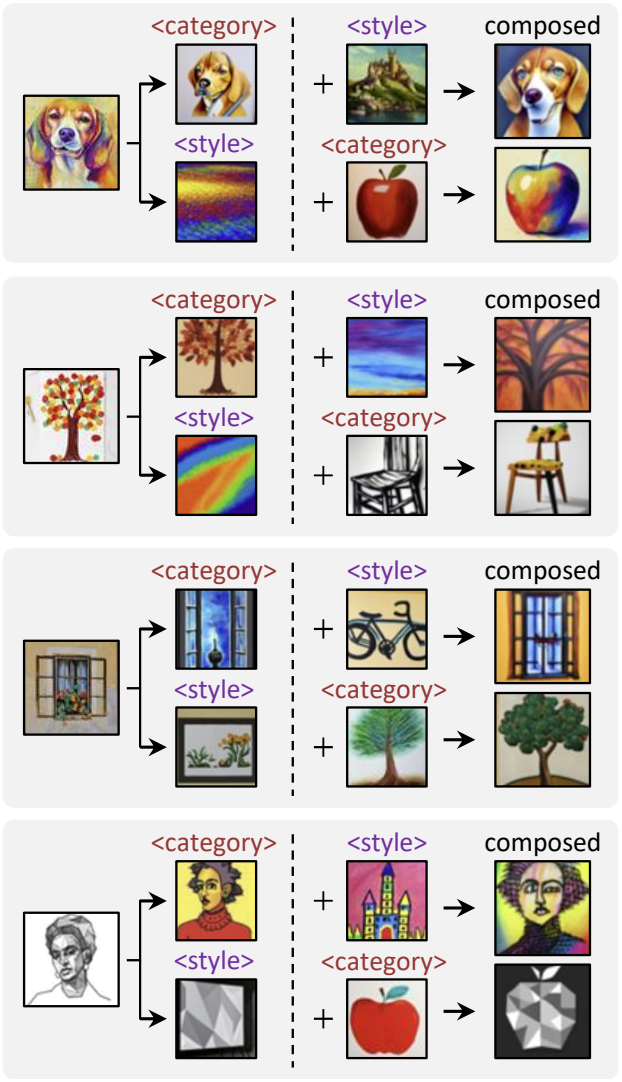

Figure 5: Visual Concept Extraction and Recomposition Results on Real-world Images of artwork.

2 INTERPOLATION OF CONCEPT EMBEDDINGS

We further demonstrate concept interpolation results. By interpolating between two concept embeddings, our model can generate meaningful images depicting gradual changes from one concept to another, such as the hybrid fruit of cherries and bananas shown in Figure 6. To interpolate between two input images, first, we extract CLIP features for both images, giving us two vectors of size 12×1024. We interpolate the two vectors using Spherical interpolation (SLERP). Specifically, given two normalized vectors A_{norm} and B_{norm} of dimensions 12×1024, we compute the dot product to find the cosine of the angle θ between them as $\cos(\theta) = A_{norm} \cdot B_{norm}$. For 12 interpolation points, each point i is calculated using $\alpha_i = \frac{i}{11}$ and the interpolated vector is $slerp(A, B, \alpha_i) = \frac{\sin((1-\alpha_i)\theta)}{\sin(\theta)} A_{norm} + \frac{\sin(\alpha_i \theta)}{\sin(\theta)} B_{norm}$. This ensures the resultant vectors maintain a constant magnitude.

With a set of trained encoders, each encoder takes in the interpolated CLIP features, and their outputs are combined with a training-time textual template to generate the interpolation results as shown in Figures 6 and 7.

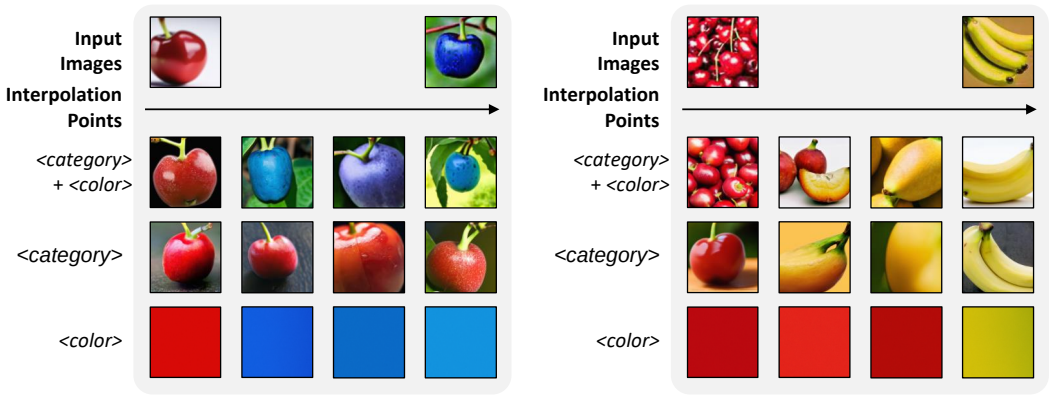

Figure 6: Interpolation on *Fruit* dataset.

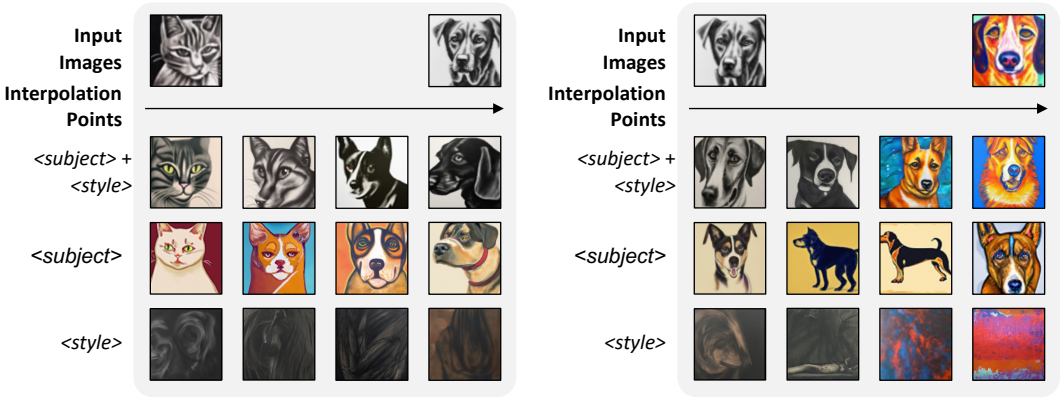

Figure 7: Interpolation on *Art* dataset.

3 The Effect of the Anchor Loss

During training time, the anchor loss (Equation (2)) encourages the encoder predictions to converge to a *meaningful* subspace within the word embedding space Gal et al. (2023). This ensures that these embeddings can be readily visualized by a pre-trained text-to-image generation model, and improves the compositionality across different concept axes, as shown in Figure 12.

Empirically, we find that simply setting a small weight on this loss can effectively achieve this objective, allowing the model to capture nuances along each axis, without collapsing to the word embeddings. In Figure 8, we empirically show such examples, where we compare the concept embeddings predicted by the color encoder to the text embedding of the training-time BLIP-2 label, *e.g.* "blue" from Figure 8, and the former preserves the specific color of the input image while the latter does not.

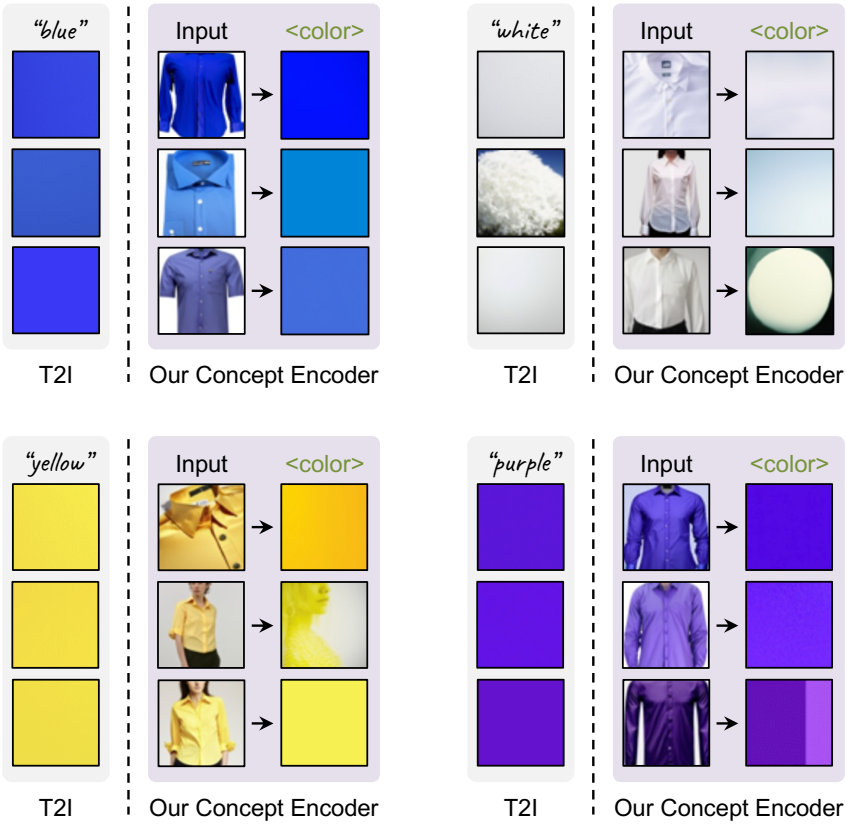

Figure 8: The concept embeddings extracted by our concept encoders capture various shades of colors instead of generic 'blue', 'white' *etc*. directly generated by the T2I model DeepFloyd.

4 EXTENSION WITH NEW CONCEPT AXES

We show that we are able to extend to additional concept axes by training new concept encoders *without retraining the existing ones*. In the experiments shown in Figures 9 and 10, given two trained encoders for `category` and `color` and the corresponding training dataset, we train the `material` encoder while keeping the other two frozen. We show that such procedural training maintains the disentanglement of the frozen encoders while allowing the framework to extend to a new concept axis.

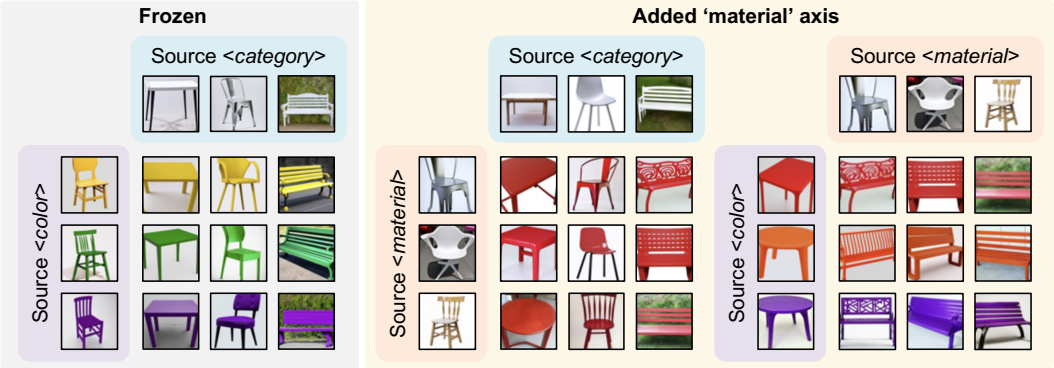

Figure 9: Concept Recomposition after adding a new `material` axis to the model trained with `category` and `color` axes.

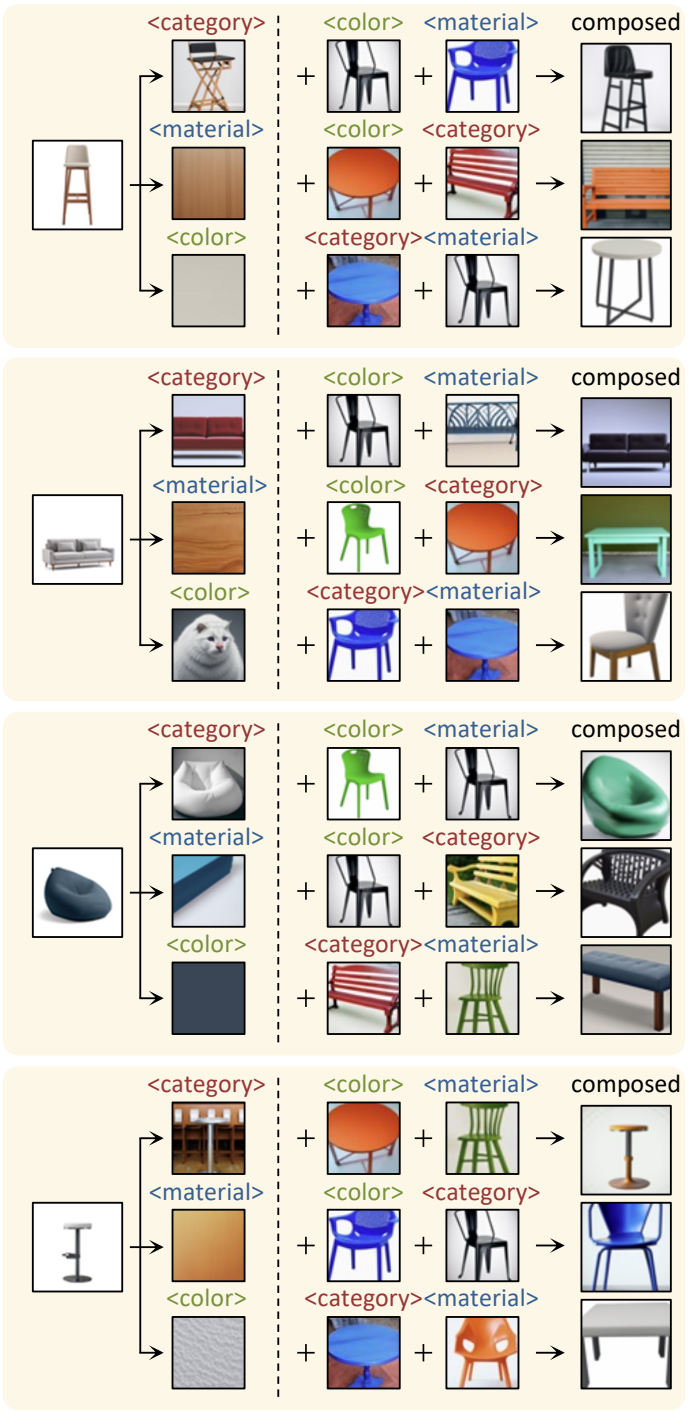

Figure 10: Test-Time Generalization Results on images of furniture, where the `material` encoder is additionally trained on top of the frozen `category` and `color` encoders.

5 Additional Baseline Comparisons

In Figure 11, we provide additional qualitative examples accompanying the experiments in Table 2. From these visual examples, we observed that the color nuances captured by ours are more accurate compared to the BLIP-2 baseline. However, since the CLIP-based metric specified in Section 4.3 cannot fully capture the minute differences, the BLIP-2 baseline still achieves comparable scores to our method despite this evident gap in visual results.

To quantify such visual differences in colors, we compare the color of a 16×16 centered patch from the input image and the image generated by the method being evaluated, both of resolution 64×64, and report the L2 error of the mean color of the two patches. We report the average of metric across all examples in Figure 11 in Table 1. Results suggest that our method captures colors more accurately compared to the baselines.

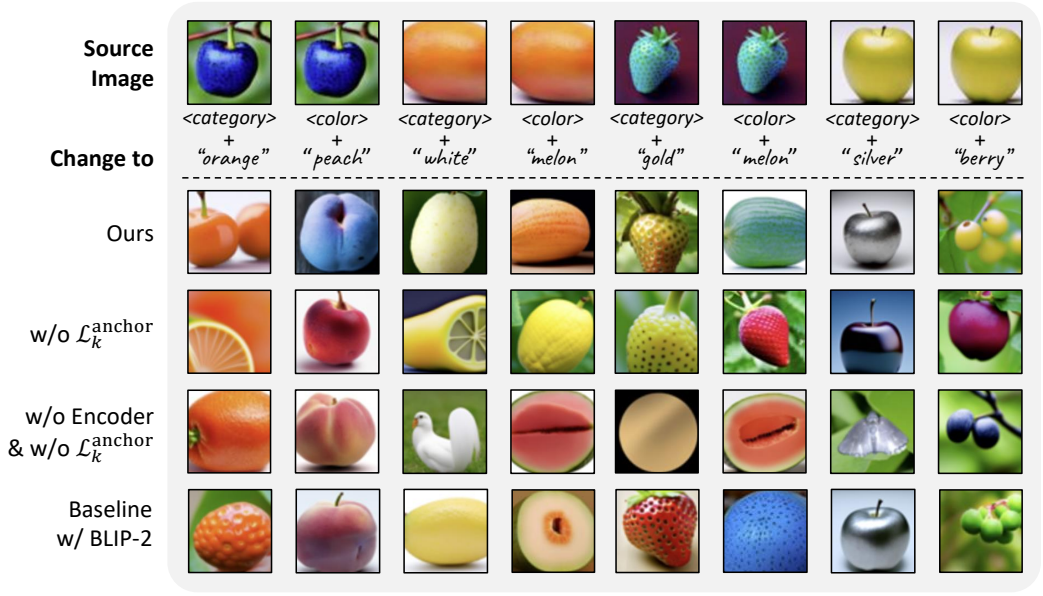

Figure 11: More qualitative results for ablations.

	CIELAB ΔE* $\downarrow$			
	Cherry	Mango	Strawberry	Apple
Ours	**35.50**	**4.76**	**16.12**	**15.64**
w/o $\mathcal{L}_k^{\text{anchor}}$	101.34	86.34	85.31	122.01
w/o Encoder & $\mathcal{L}_k^{\text{anchor}}$	85.86	82.46	82.07	127.80
Baseline w/ BLIP-2	79.90	25.20	47.30	73.62

Table 1: **Quantitative Comparisons on Color Editing.** To quantify the differences in color seen in Figure 11, we use CIELAB ΔE*, the color-distance metric recommended by the International Commision on Illumination (Fraser et al., 2004).

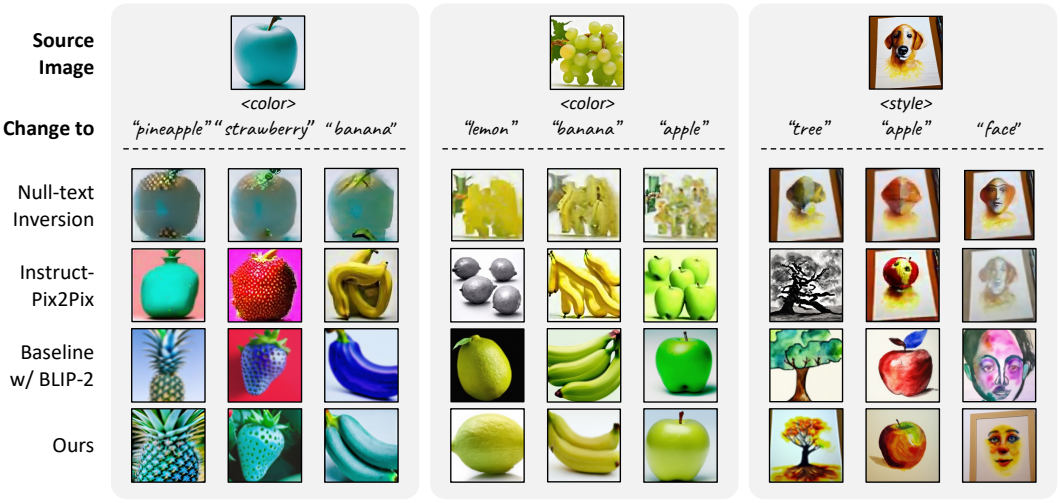

Figure 12: **Baselines Comparisons.** Compared the text-based image editing methods, our method achieves significantly better compositionality due to the disentangled concept representation, and captures fine-grained color variations, which the baseline struggles to encode with language.

	CLIP-Score ↑						Human Evaluation ↑	
	Edit Category			Edit Color			Edit Cat.	Edit Clr.
	Cat.&Clr.	Cat.	Clr.	Cat.&Clr.	Cat.	Clr.	Score	Score
Null-text Inversion	0.258	0.249	**0.249**	0.259	0.265	0.223	0.287	0.316
InstructPix2Pix	0.267	0.277	0.226	0.270	0.245	**0.268**	0.233	0.648
Baseline w/ BLIP-2	**0.313**	0.294	0.248	0.287	0.271	0.237	0.448	0.379
Ours	0.308	0.297	0.238	**0.302**	**0.287**	0.236	**0.968**	**0.840**
w/o $\mathcal{L}_k^{\text{anchor}}$	0.268	0.276	0.219	0.263	0.257	0.236	-	-
w/o Encoder & $\mathcal{L}_k^{\text{anchor}}$	0.288	**0.300**	0.214	0.242	0.213	0.265	-	-

Table 2: **Quantitative Comparisons on Visual Concept Editing.** Compared to existing image editing baselines (Brooks et al., 2023; Mokady et al., 2022), our method achieves better overall CLIP score when editing either axis, and is particularly effective at retaining `category`-related concepts as reflected in human evaluation. 'Cat' denotes Category and 'Clr' denotes Color.

6 Discussion on Textual Inversion

As discussed in Section 3.1, compared to directly using text as inputs for image generation, using techniques like Textual Inversion (Gal et al., 2022), our model is able to capture more nuanced visual details of a particular image with continuous embeddings, instead of discrete words. This can be illustrated in the empirical results in Figures 11 and 12 as well as the results in the main paper, which show that our method preserves the nuances from input images more accurately than the BLIP-2 baseline which uses texts for conditional generation.

REFERENCES

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Bruce Fraser, Fred Bunting, and Chris Murphy. Real world color management, 2004.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

Rinon Gal, Moab Arar, Yuval Atzmon, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *ACM Trans. Graph.*, 2023.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.