Concept Decomposition for Visual Exploration and Inspiration - Supplementary Material

Anonymous Authors

Table of Contents

1. Implementation details

We rely on the diffusers [4] implementation of Textual Inversion [1], based on Stable Diffusion v1.5 text-to-image model [3]. We used the default training parameters provided in this implementation, except for changing the batch size to 2 (which scales the learning rate to 0.004 respectively). We used four different seeds $\{0, 1000, 1234, 111\}$ for each sibling nodes optimization. To generate the set of 10 images for each new node we first generated a random set of 40 images, and used our proposed CLIP consistency measurement to choose a subset of 10 images that are most consistent with each other. Our code will be made available to facilitate future research.

2. Baselines

In the absence of existing works attempting to achieve our goal of decomposition into different aspects, we compare our performance in intra-tree combination with two existing relevant works.

We consider Textual Inversion [1], and its more advanced modification – Extended Textual Inversion [5] – designed specifically for appearance mixing (which is most similar to our "intra-tree combination").

We provide a qualitative comparison to these methods in Figure 1. On the left we aim to combine the aspect of a wooden saucer and the creature on the cup from the objects presented on top. On the right we aim to combine a part of the stone statue with some specific style aspects of the cat sculpture.

Gal et al. [1] propose a style transfer application, in which their method can be used to find pseudo words representing a specific style taken from a given concept, and can then be applied in combination with other concepts. To extract the style code from a given concept, they replace the training texts with prompts of the form: "A painting in the style of S^*".

For the TI baseline, we applied the original Textual Inversion for the first concept (from which we wish to take the structure), and for the appearance concept we used their proposed style extraction application described above. This results in a pair of textual tokens S_1^{TI}, S_2^{TI} that represent each concept. We explicitly combine these tokens in a sentence, providing the desired mixing description (e.g. "S_1^{TI} in the style of S_2^{TI}" and use it to generate an image.

Voynov et al. [5] propose an extended textual conditioning space for a diffusion model that can be used to control style and geometry disentanglement. The main idea is to provide each diffusion UNet cross-attention layer with an independent textual prompt. The authors notice that low-resolution UNet layers are commonly responsible for geometrical attributes, while high-resolution input and output layers are responsible for style-related attributes.

For the task of style mixing, given a pair of objects, the method performs two independent Textual Inversions to this extended prompt space (called XTI in the paper). Then, the low-resolution layers are provided with the inversion of the object that donors the shape, and the high-resolution layers are provided with the inversions of the object that donors the appearance.

For the comparison to XTI [5], we use their recommended hyperparameters. We apply two independent Textual Inversions to the extended prompt space, which brings the pair of textual tokens S_1^{XTI}, S_2^{XTI}. To use the geometry from S_1^{XTI} and appearance of S_2^{XTI}, we provided the prompt "a photo of S_1^{XTI}" to the deeper (low-res) layers and "a photo of S_2^{XTI}" to the shallower (high-res) UNet layers. We tried to combine the concepts using different layers split to achieve the best possible performance.

From Figure 1, we can see that these baselines fail to

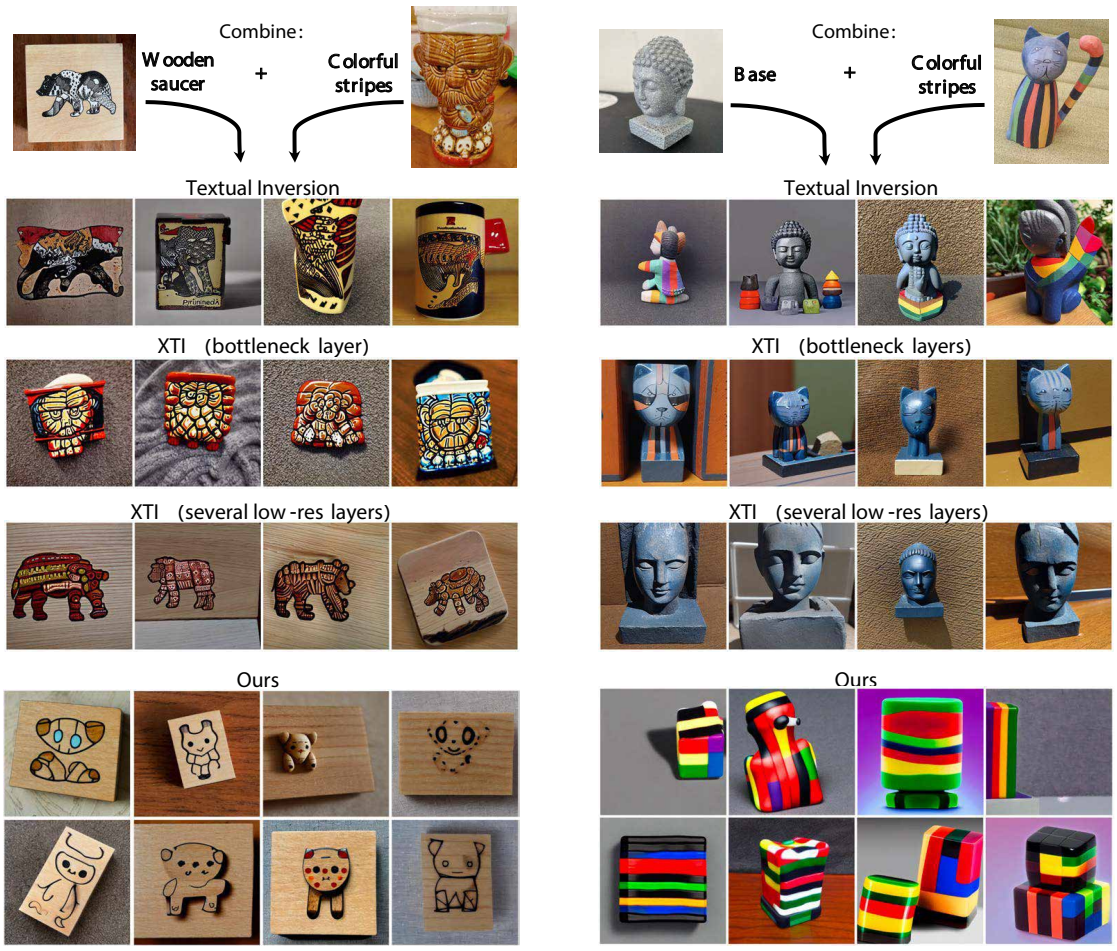


Figure 1. Comparison of blending specific aspects of concepts. Top row are the source objects and a description of which aspect is taken. Second row are the results of blending the chosen concepts using Textual Inversion [1]. Third row are the results of blending with Extended Textual Inversion [5] (XTI) when only bottleneck layers are provided with the left object. Fourth row are results of XTI where a wider range of the low-resolution layers are provided with the left object. Last two rows are images generated with our proposed approach.

combine the very specific aspects of the source objects. Textual Inversion commonly blends the attributes, while XTI is able to transfer either the whole creature's appearance, or texture only, failing to extract only the shape. In contrast, using our approach it is possible to pick the two distinct aspects and combine them naturally to depict a new concept.

3. Ablation and Analysis

3.1. Binary Tree

Our choice to use a binary tree stems from two main reasons: (1) complexity, and (2) consistency.

It is technically possible with our method to build a tree with more than two children per node, however, we believe this may add redundant complexity to the method. The use of more than two children will result in a longer running time in each level since we will have to split more nodes. Additionally, after two levels we will receive 12 aspects (for a simple scenario of three nodes), which may be difficult to visualize and navigate.

In terms of consistency, we observe that when optimizing more than two nodes at a time, the chance of receiving inconsistent nodes increases. Often, two nodes will be consistent, and the third node is inconsistent or may depict irrelevant concepts such as background. We visually demonstrate this in Figure 2, on the "red teapot" object. We present the aspects obtained from the optimal seed after 200 iterations, for the case of two nodes (left) and for the case of three nodes (right). As can be seen, the sub-concept in v_3 for the 3 nodes optimization does not appear to be consistent or comprehensible, and therefore is not useful in achieving our goal of extracting aspects from the parent concept. For the two-node case, however, the aspects obtained provide a coherent concept in addition to decomposing the object well. At the bottom of Figure 2, we show the concepts depicted by two other seeds. The results show that when using two nodes, the rest of the seeds also produce relatively consistent results, compared to the case where three nodes were used.

The following quantitative experiment further confirms this observation. We obtained 52 trees for our set of 13 objects (using four seeds for each object as described in the main paper). Each tree is a 3-node tree with one level, resulting in a total number of 156 nodes.

For each node, we then applied our CLIP-based consistency test to determine its average consistency score. For each tree, we sorted the 3 nodes according to their consistency and received a set of $\{v_1, v_2, v_3\}$, where v_1 is the most consistent node of the three, v_2 is the second most consistent and v_3 is the least consistent. We then average the scores of $\{v_1, v_2, v_3\}$ across all objects. Results are shown in Figure 4; observe that there is a noticeable consistency gap between the top 2 nodes (achieving average scores of $0.804, 0.742$) and the third node who achieved a score of 0.633. This indicates that, on average, two of the three nodes are consistent, while the last may contain incoherent information. This experiment correlates well with our visual observation (as demonstrated in Figure 2).

Figure 2. Comparison of optimizing for two child nodes (left) v.s. three child nodes (right). Using three nodes increases the chance of arriving at inconsistent or irrelevant concepts. At the top we show the results of the chosen seed among the four seeds, and below we demonstrate how two of the other seeds provide similar results, demonstrating that this trend is general.

3.2. Timestep Sampling

As discussed in Section 4.1 of the main paper, we use the timestep sampling approach proposed in ReVersion [2], favoring larger t values. This sampling approach plays a significant role in the success of our method, as demonstrated in Figure 3.

The left side of Figure 3 shows the results obtained when using a uniform sampling approach (which is the more common approach in LDM-based optimization), the right side shows the results obtained when using the sampling method we selected from ReVersion.

In both cases, the results were obtained after 500 iterations with the same seed and settings. As can be seen, the uniform timestep sampling approach negatively affects both reconstruction quality (see "v1 v2") and decomposition quality, where for example for the cat sculpture the aspect depicted in "v1" is unrelated to the original concept.

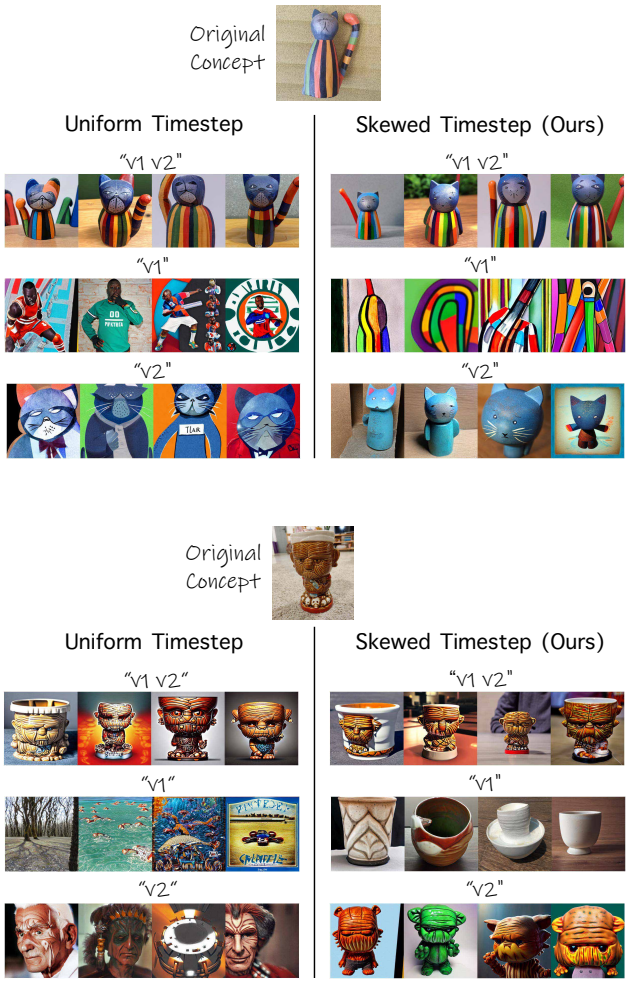


Figure 3. Timestep sampling approach ablation. We show the effect of using a uniform sampling (left), compared to using the sampling approach from ReVersion [2], which favor larger values of t.

3.3. Consistency Test

In this section we provide examples and details regarding our proposed CLIP-based consistency test presented in Section 4.1 in the main paper. First, we visually demonstrate the effect of using $k = 4$ seeds in each run. We observe that 4 seeds are generally enough for most of the concepts, and in most cases also 2 seeds may be good enough. However we do note that the variability in results among the different seeds can be quite meaningful in some cases. We demonstrate this in Figures 5 and 6, where we show the original concept on top, along with the random set of images generated for each nodes in each of the seeds.

The seed that was chosen using our CLIP-based consistency measurement is marked in green. While the results depicted in Figure 5 were reasonable for most of the seeds, in Figure 6 we can see that seed1 and seed2 are failure cases,

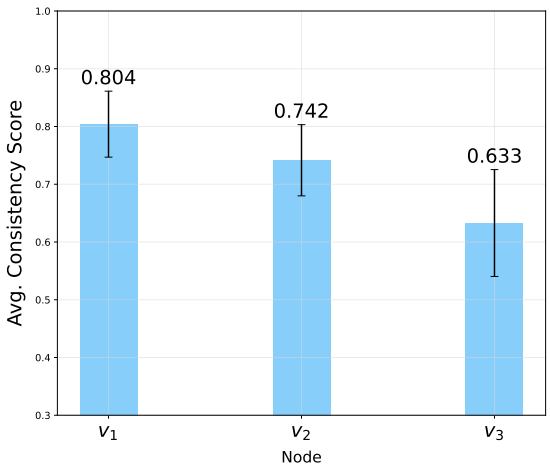


Figure 4. Average CLIP-based consistency scores of the three child nodes sorted from left to right. These scores were measured for 3-node trees obtained for 13 objects using 4 seeds per object. Observe that, on average, the third node tend to encode incoherent information, which encourages us to choose a binary tree structure.

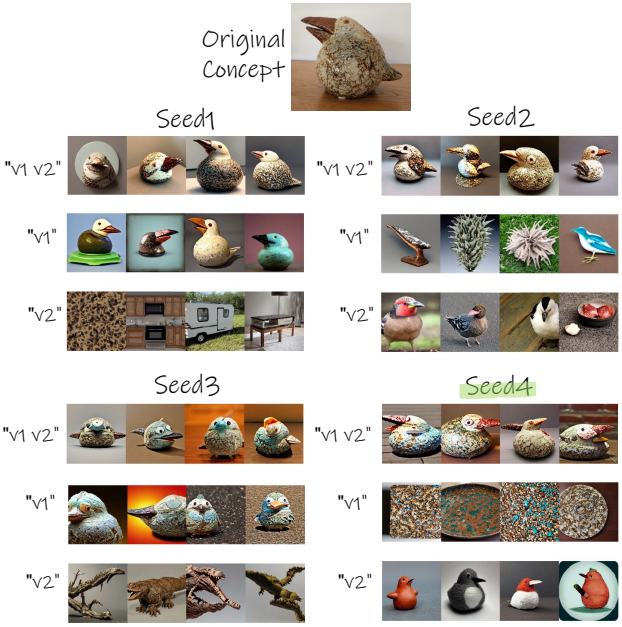


Figure 5. Results of four different seeds after 200 steps. The best seed is marked in green.

where in seed 1 the concept depicted in v_2 is inconsistent and not interpretable, and in seed4 we have a case of one dominant node (v_1).

Additionally, we provide an illustration to better clarify the significance of the consistency scores and their relationship to the patterns observed visually in the trees. In Fig-

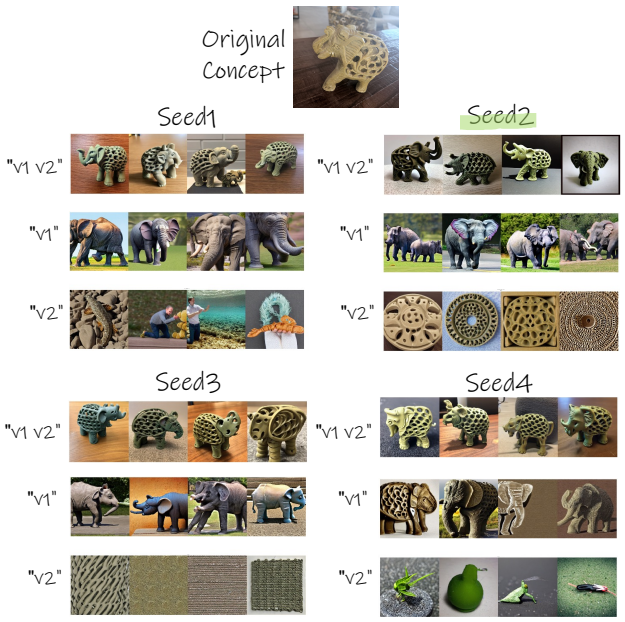


Figure 6. Results of four different seeds after 200 steps. The best seed is marked in green.

ure 7 we show examples of two trees with different characteristics. Next to each node are the self-consistency score (marked in red) and the siblings consistency score (marked in green). The first score measures the degree to which the images depicted in a specific node are consistent withing themselves. The second score indicates the similarity between sibling nodes.

First, observe that the self consistency score for the root node (0.89) is the highest, since the images depicted in that node originated from the set provided by the user. This indicates the highest consistency score possible in our settings. In addition, we observe that the self consistency score across most nodes is relatively high and does not vary significantly as we go deeper in the tree. However, v_5 in the right tree obtained a self consistency score of 0.66, which is relatively low, and in our scale it means that the set is not considered consistent.

Considering that this node is not consistent with itself, it is obvious that it is not consistent with its sibling node, which is why, in such cases, we can ignore the score obtained in green for that node in this discussion.

We now examine the scores in green, which indicate consistency across siblings. First, note that in both trees, the consistency across siblings is low (0.6 and 0.61) in the first level, suggesting that a good separation has been achieved. However, at the second level we can see that this score generally increased, indicating that the quality of separation decreases as we go deeper in the tree. Additionally, the sibling similarity correlates well with the visual information, with v_3, v_4 in the left tree and v_3, v_4 and v_5, v_6 in the right tree

Table 1. Average self consistency (left) and sibling consistency (right) scores. The scores were obtained for 13 trees.

Node	Self Cons.	Avg. Level1	Node	Sibling Cons.	Avg. Level2
v1	0.790	0.792	v1	0.580	0.58
v2	0.794		v2	0.580	
v3	0.781	0.783	v3	0.711	0.69
v4	0.780		v4	0.711	
v5	0.768		v5	0.669	
v6	0.803		v6	0.669	

appearing to be more consistent than the other pairs.

It is important to note that in these cases, when the consistency among siblings is high, or when one node is inconsistent within itself, the split will be stopped at this particular level.

In order to confirm this observation, we measured these scores for the set of 13 trees that were used for the other evaluations. For each node, we calculated the self consistency score as well as the sibling consistency score, and averaged these scores across the trees. The results are presented in Table 1. In both levels, the average self consistency score is high, while the average siblings consistency score increased with the transition from the first to the second level, indicating that the splits are less distinct on average. The reason for this is that as we go deeper into the tree, the components are becoming increasingly simple, making it more challenging to further split them.

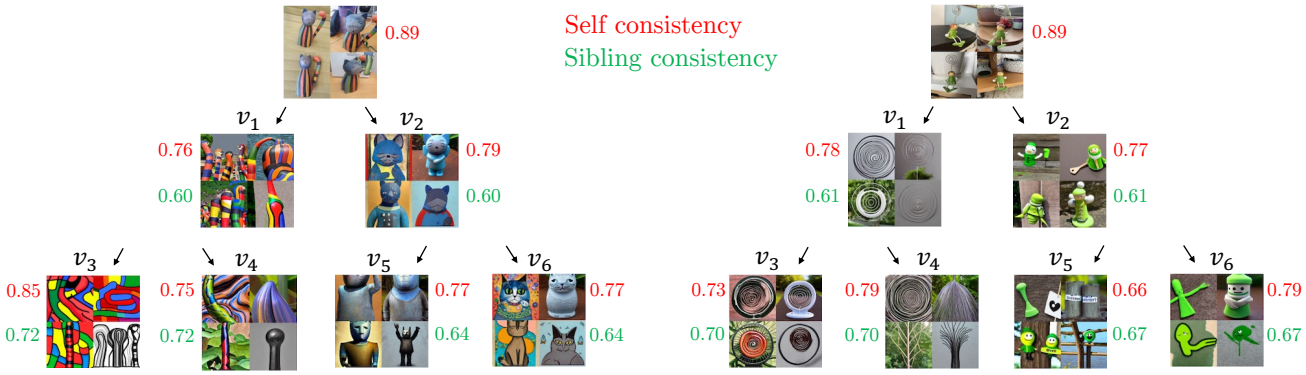


Figure 7. An illustration of two trees with different characteristics. The original training set is depicted at the root of the trees. Next to each node we present its self-consistency score (in red) and the consistency score of that node with its brother node (in green). The scores were obtained using our CLIP-based consistency measurement described in the main paper.

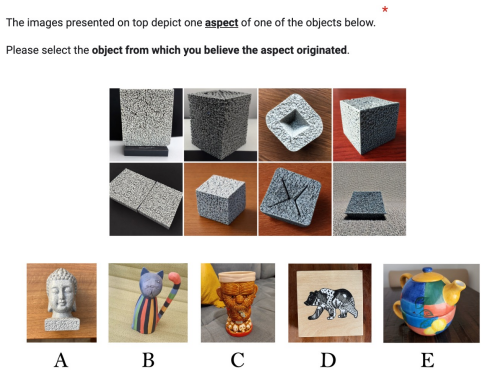


Figure 8. Example of a question we presented in the aspect relevance survey.

3.4. Perceptual Study

The following section provides additional details regarding our perceptual study described in section 5.2 of the main paper. For the consistency evaluation, we collected answers from 35 participants. Participants were presented with 15 pairs of random image sets, and they were asked to determine which set in each pair is more consistent. In order to handle cases where the sets are similar, we have also added two options to choose from - "Both sets are equally consistent", and "Both sets are equally not consistent". Figure 9 contains a few examples of the survey questions. On the left of each set, we also present the results in percentages, indicating which answer was selected by the majority of people. In the aspect relevance experiment, we collected answers from 35 participants and asked each participant 15 questions. Figure 8 provides an example of the questions. The question were obtained from 5 chosen objects, shown at the top of Figure 8.

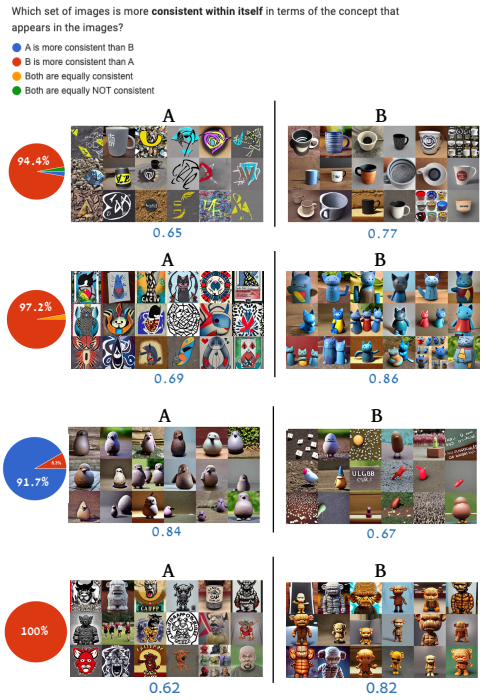


Figure 9. Examples of questions asked in the consistency evaluation survey. On the left we show the results in percentages, indicating which answer was selected by the majority of people.

4. Additional Qualitative Results

In Figures 10 and 11 we show more examples of inter-tree combinations. At the top part of Figures 12 to 19 we show examples of trees on various objects.

At the bottom part of Figures 12 to 15 and in Figures 20 and 21 we show visual examples of intra tree combinations.

At the bottom part of Figures 12, 16 and 17 and in Figures 22 to 27 and 28 we show examples of text based generation.

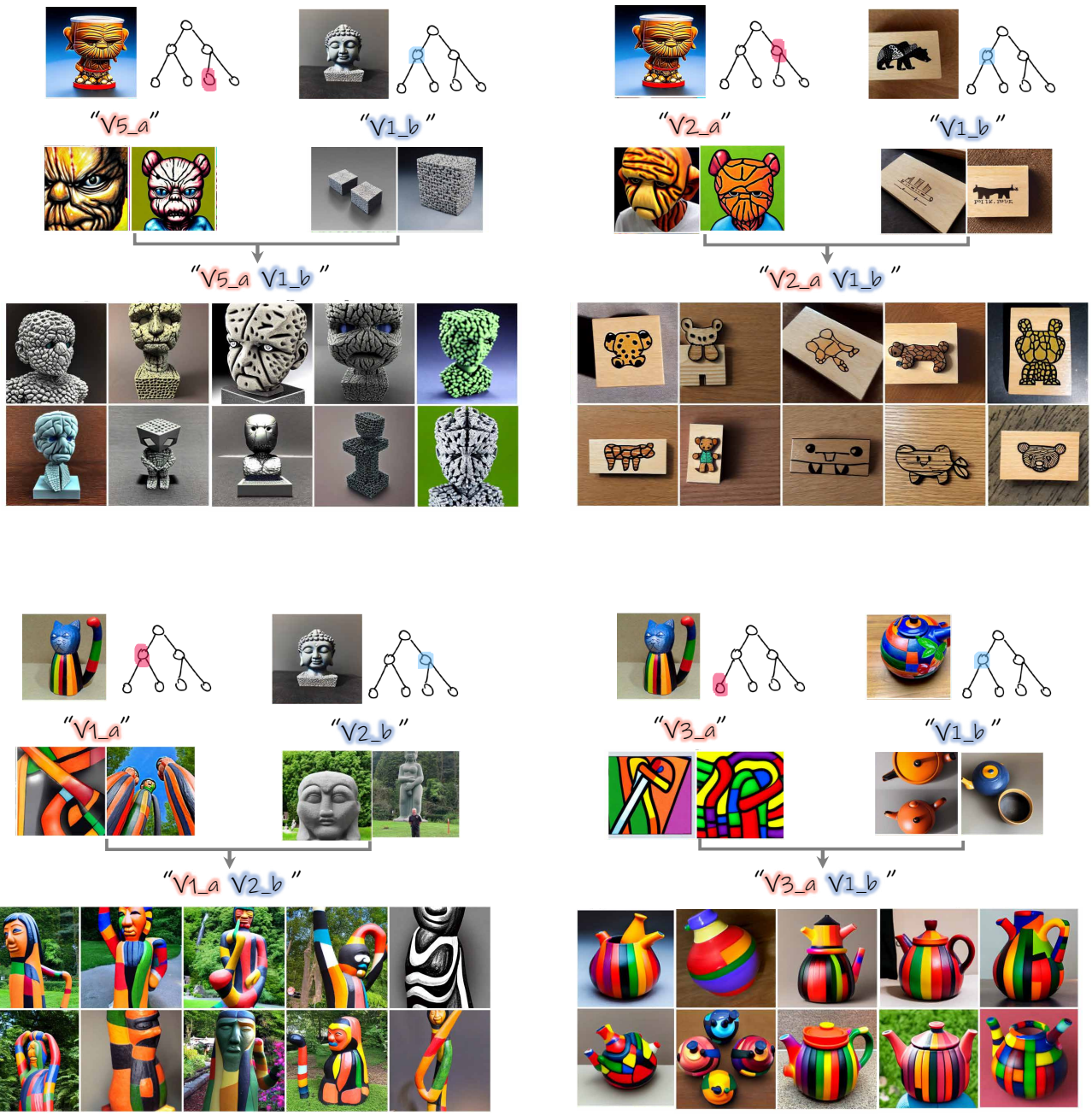


Figure 10. More examples of inter-tree combinations.

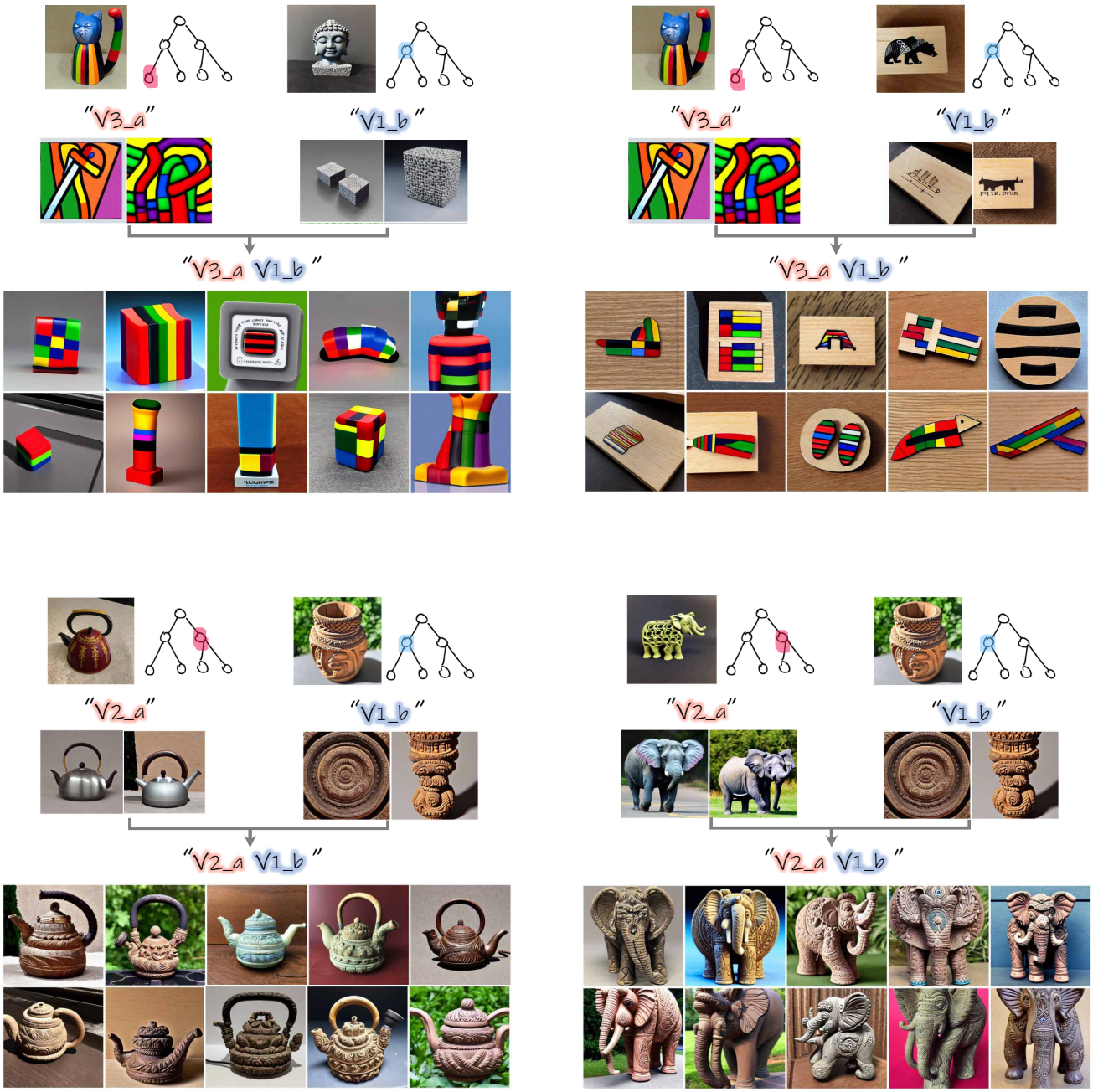


Figure 11. More examples of inter-tree combinations.

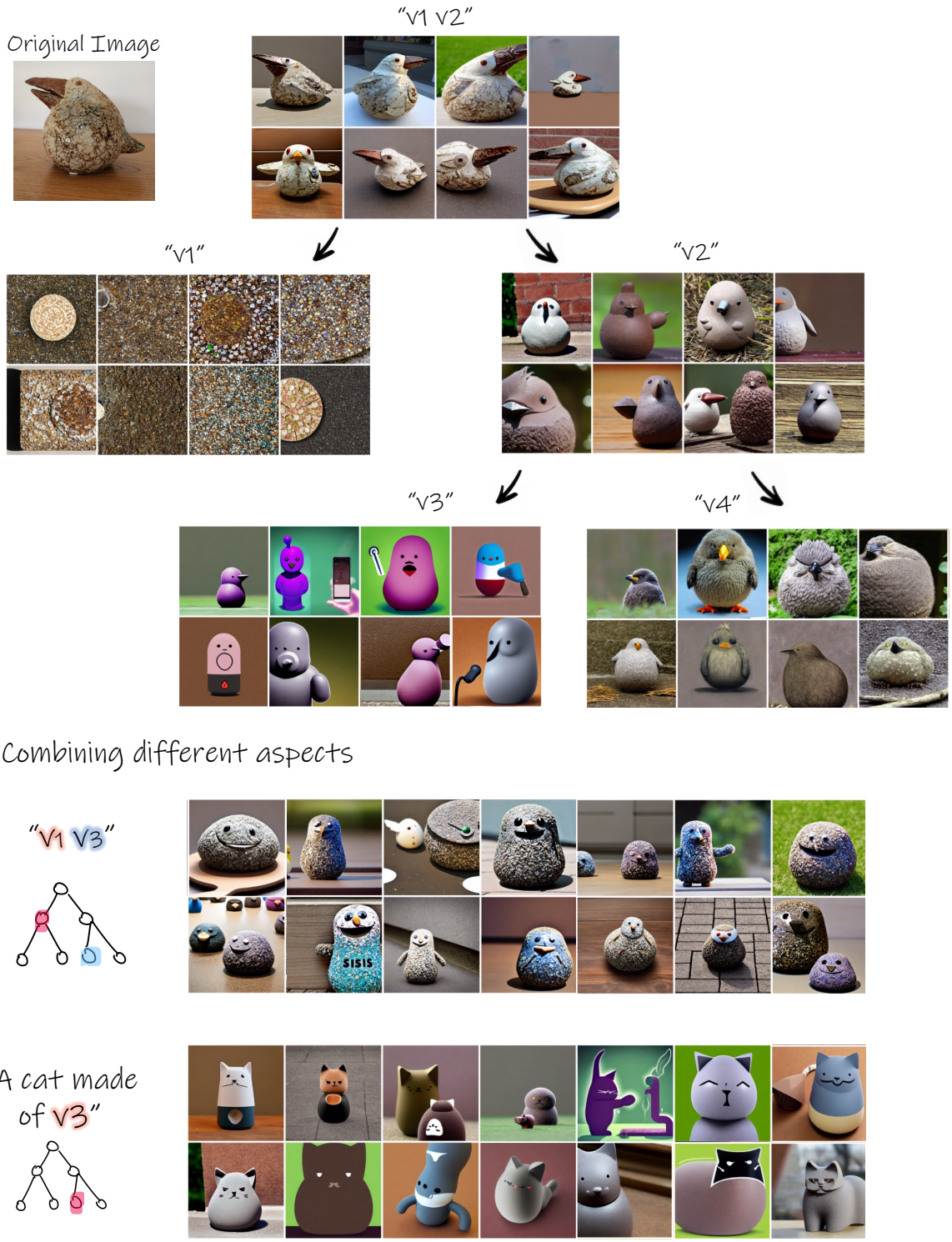
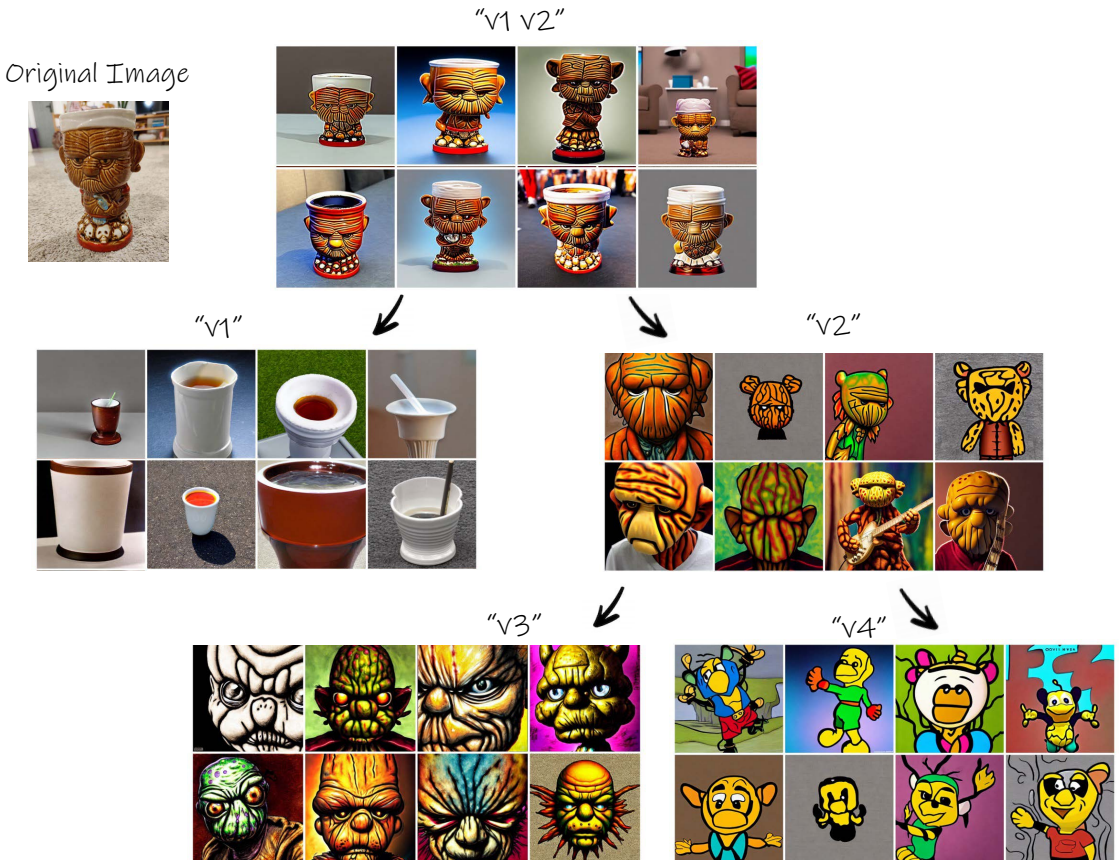


Figure 12. Exploration tree for the "round bird" object. At the bottom we show examples of possible intra-tree combinations and text-based generation.



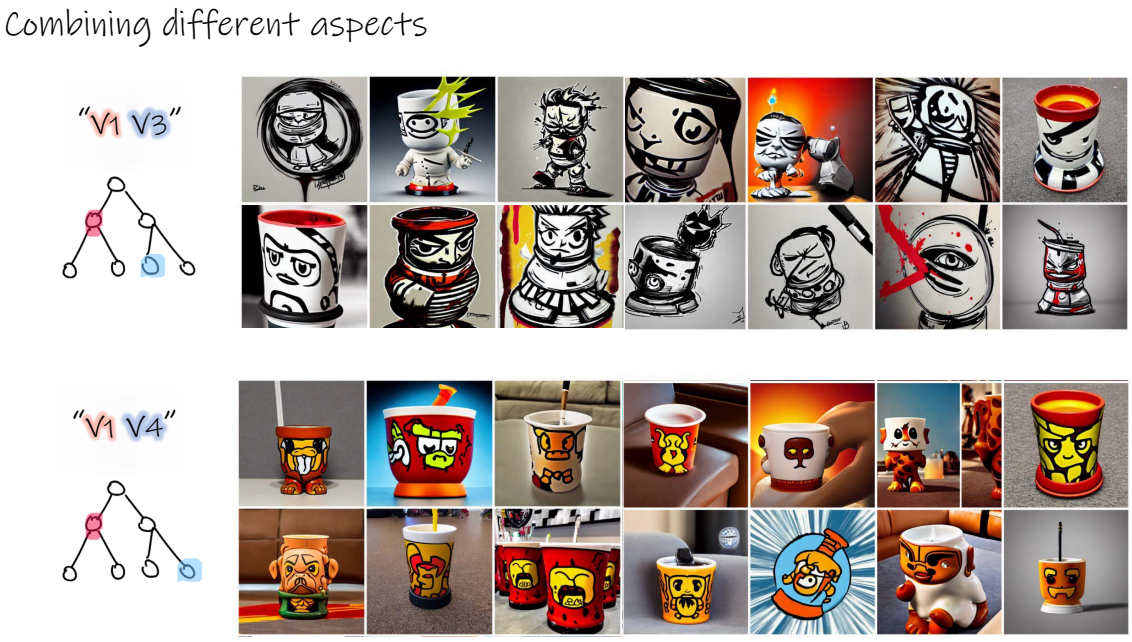
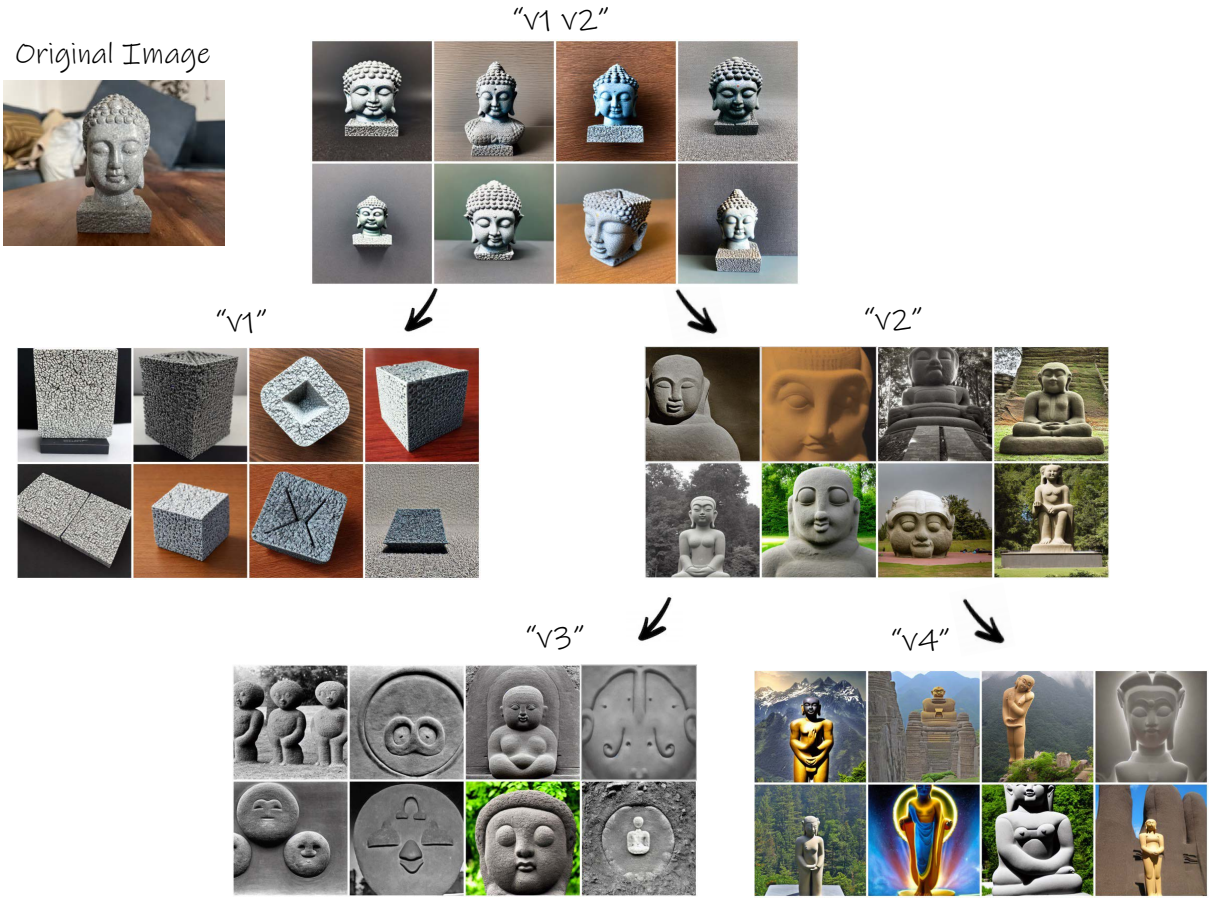


Figure 13. Exploration tree for the "scary mug" object. At the bottom we show examples of possible intra-tree combinations.



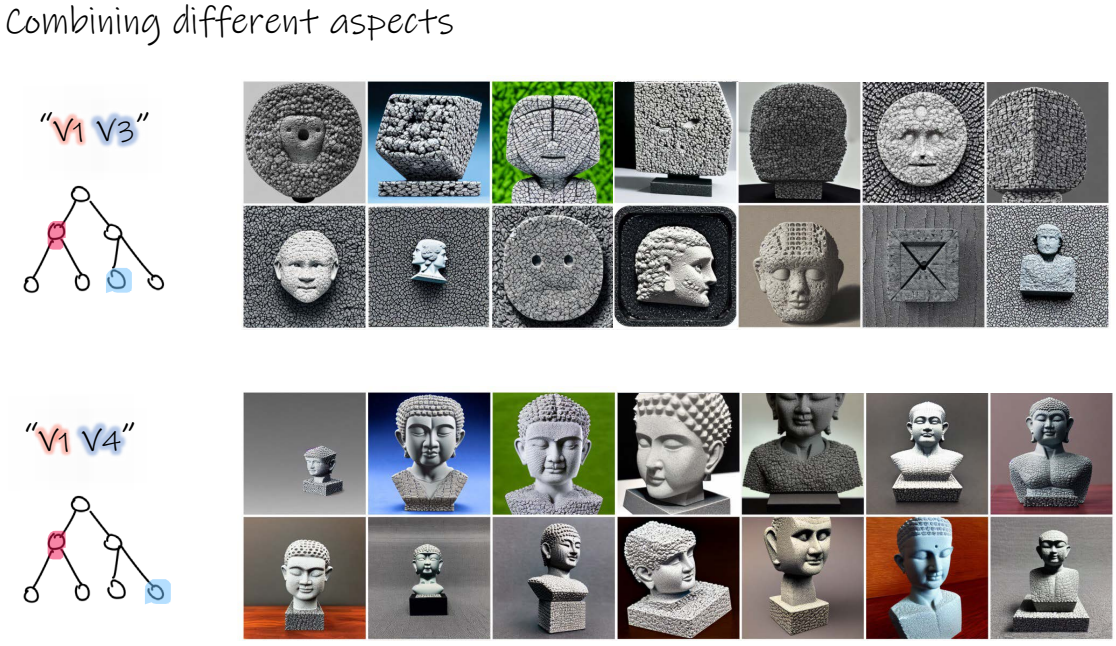
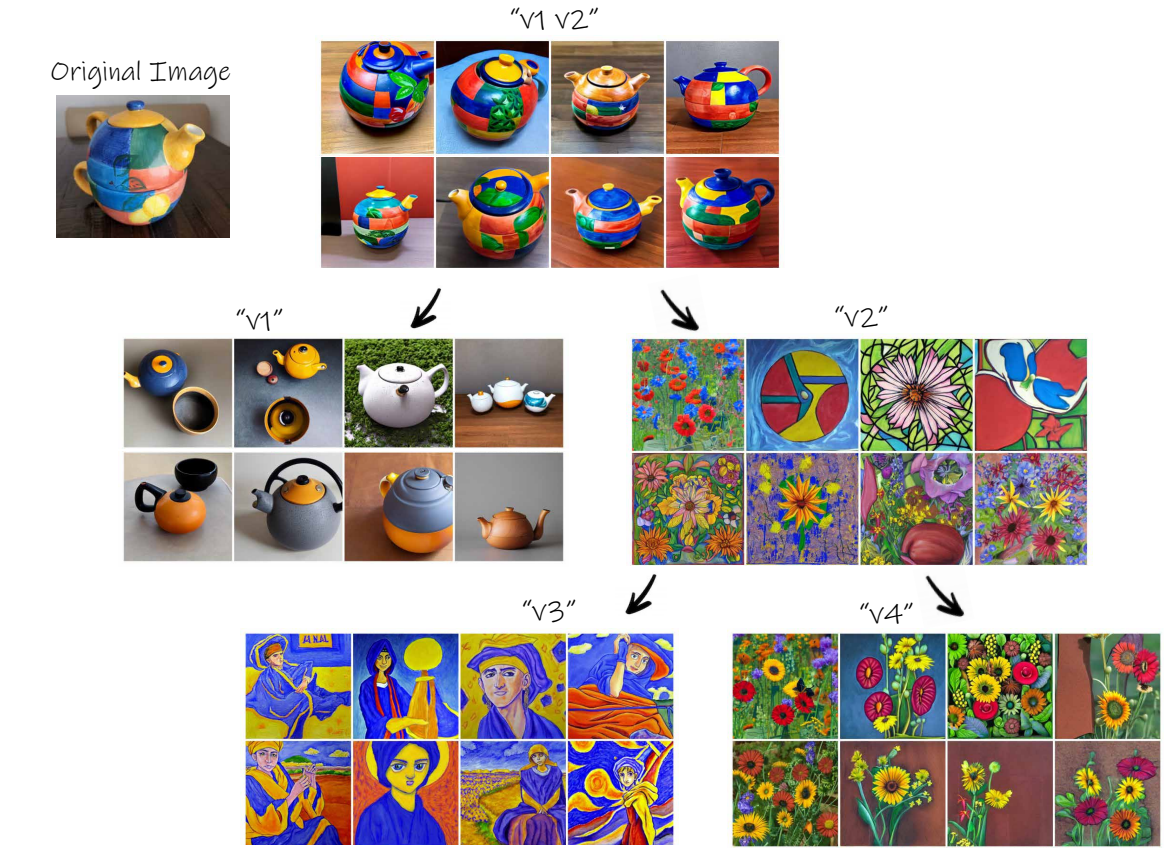


Figure 14. Exploration tree for the "Buddha sculpture" object. At the bottom we show examples of possible intra-tree combinations.



Combining different aspects

Figure 15. Exploration tree for the "colorful teapot" object. At the bottom we show examples of possible intra-tree combinations.

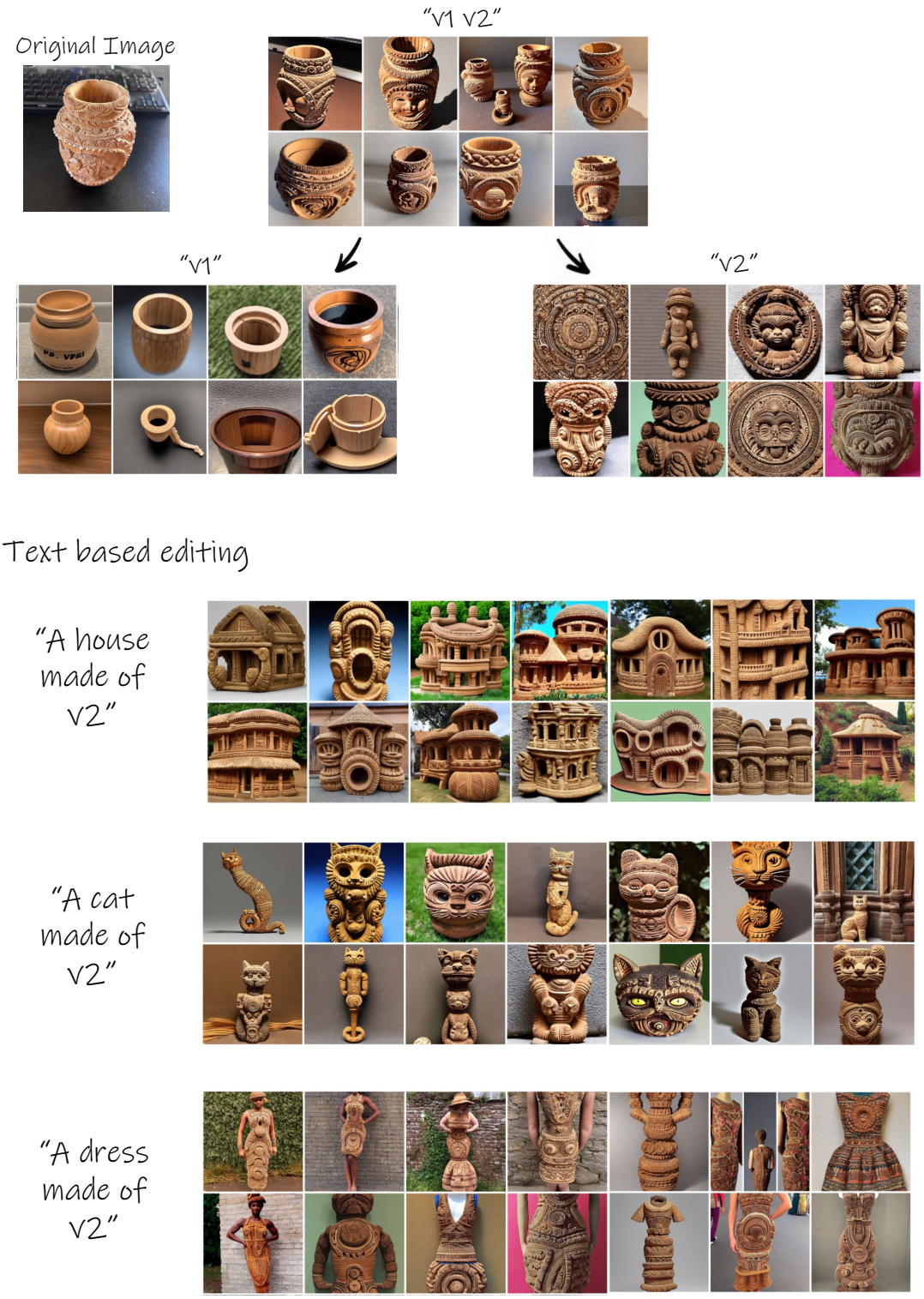


Figure 16. Exploration tree for the "wooden pot" object. At the bottom we show examples of possible text-based generation.

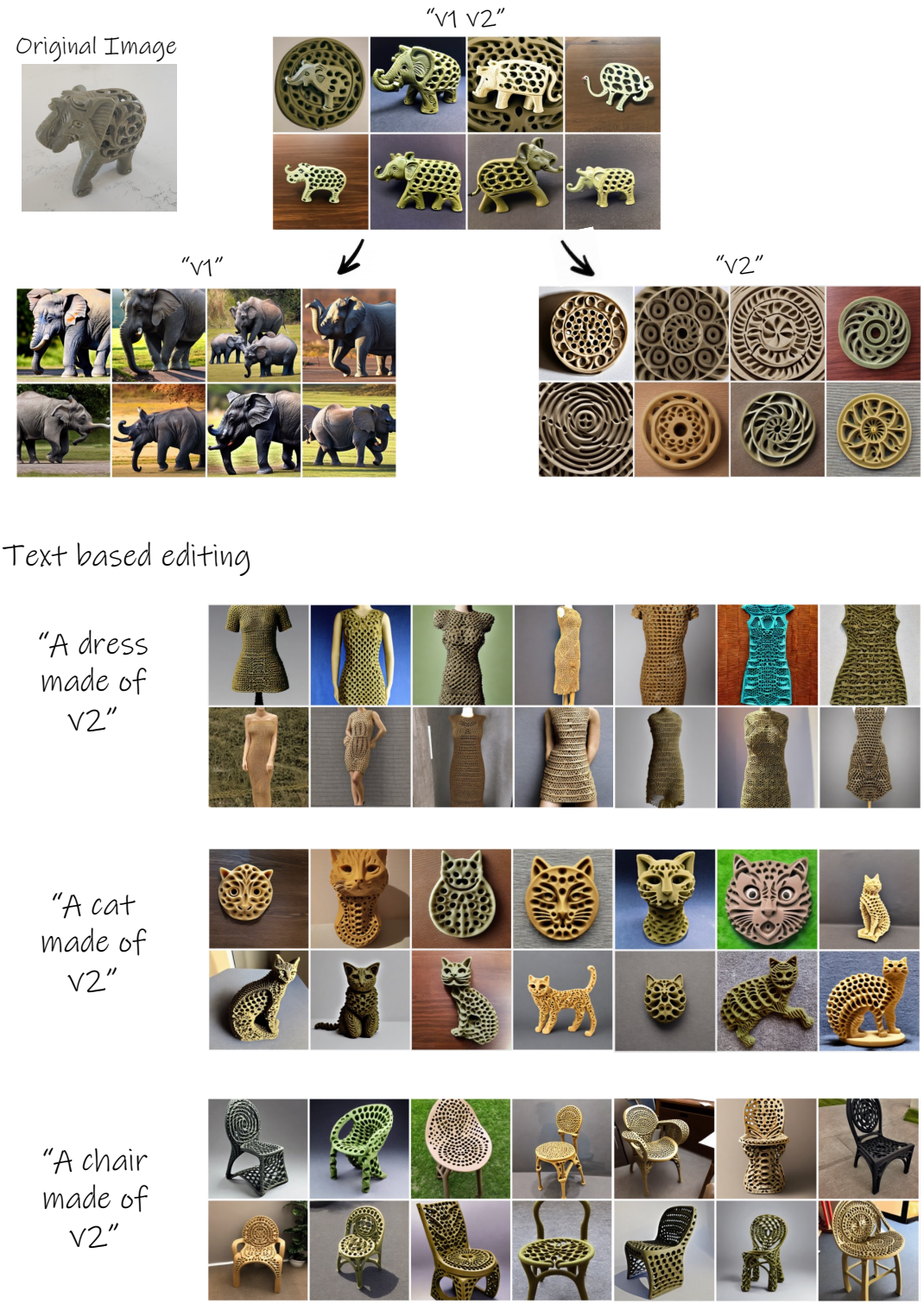


Figure 17. Exploration tree for the "elephant" object. At the bottom we show examples of possible text-based generation.

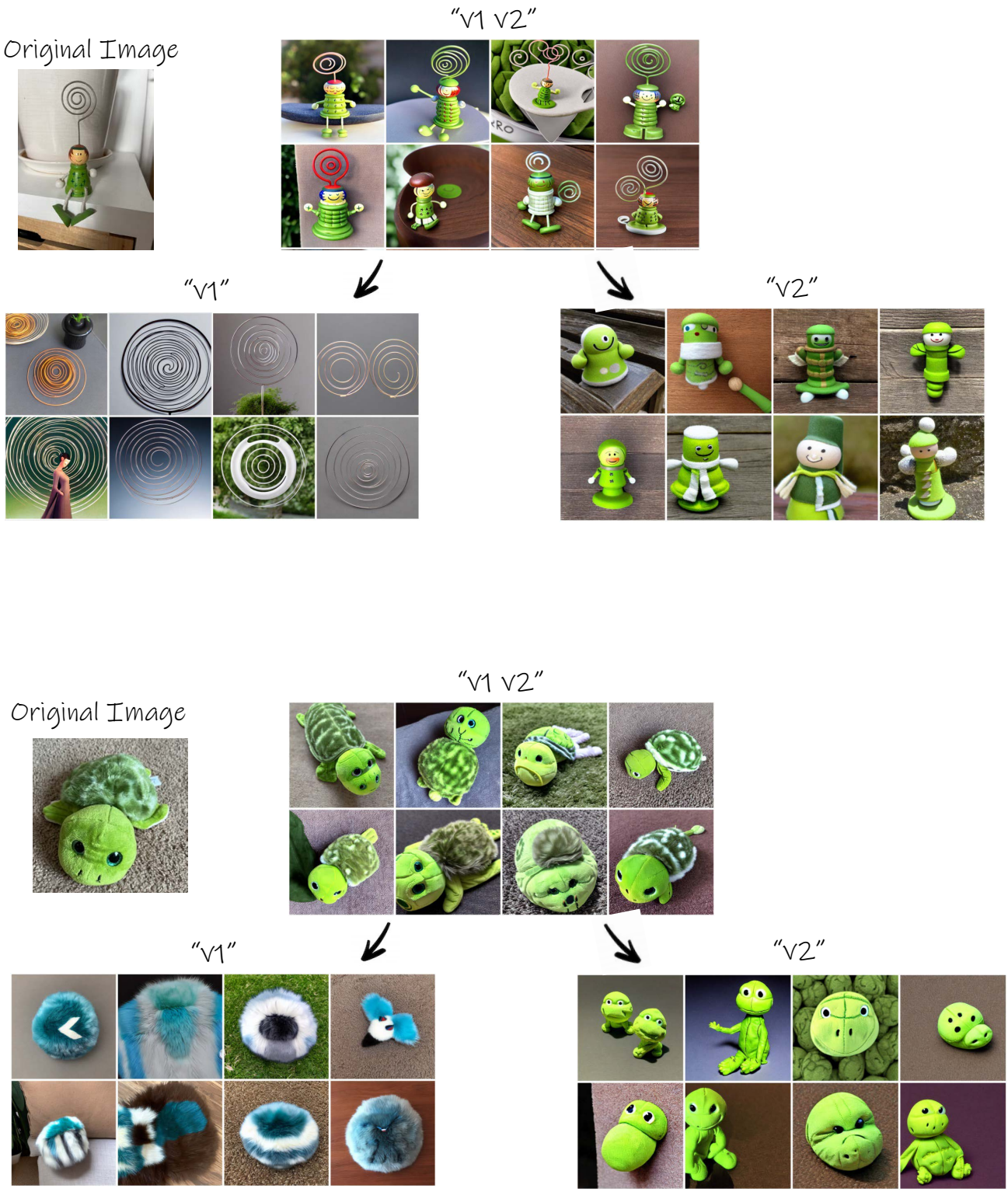


Figure 18. Exploration trees for the “green doll” and the “turtle” objects.

Figure 19. Exploration trees for the “Girona mug” and the “physics mug”.

Figure 20. More examples of intra-tree combinations for the “cat sculpture” object. The full original tree is shown in the main paper.

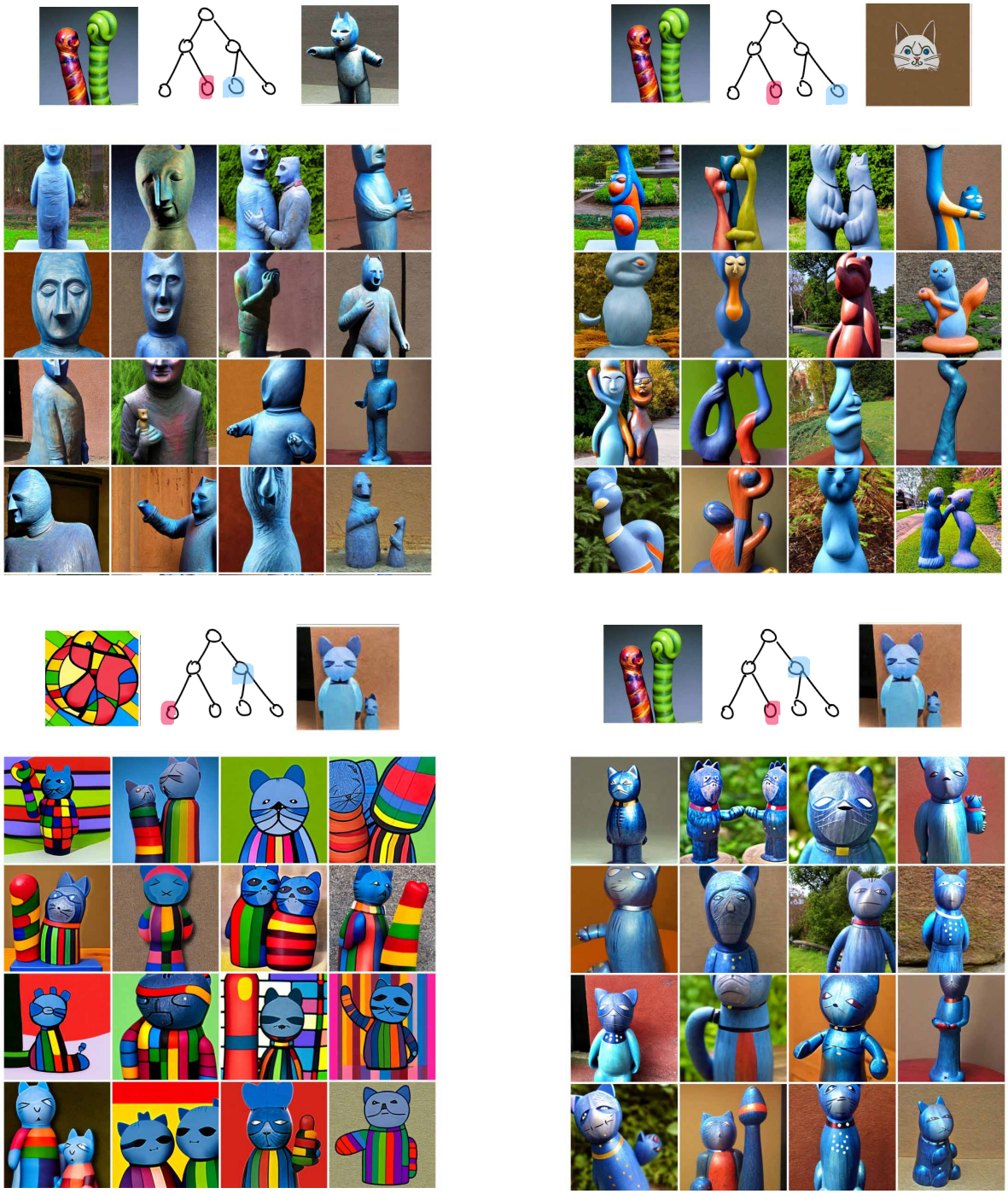


Figure 21. More examples of intra-tree combinations for the “cat sculpture” object. The full original tree is shown in the main paper.

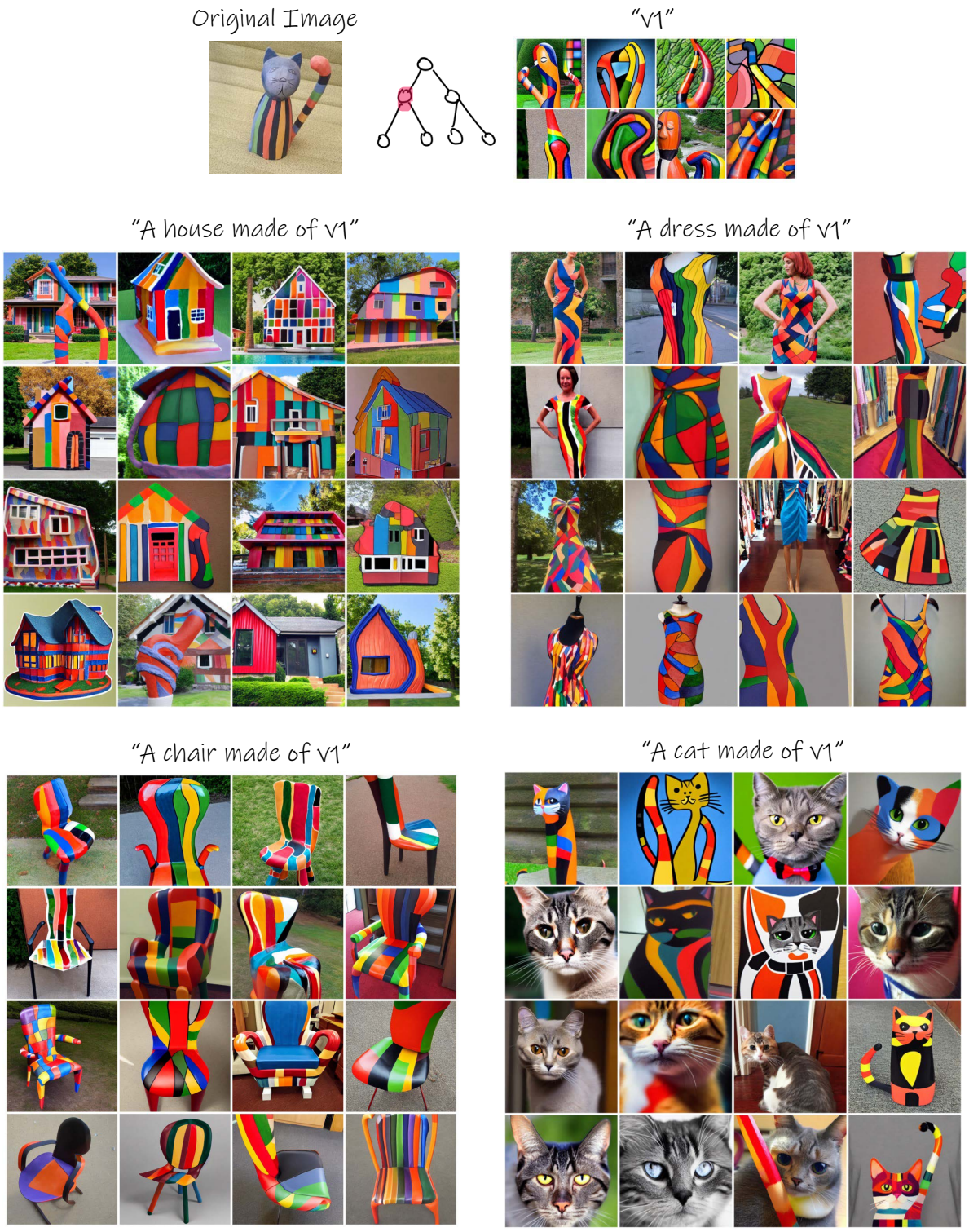


Figure 22. More examples of text based generation for the "cat sculpture" object. The full original tree is shown in the main paper.

Figure 23. More examples of text based generation for the "cat sculpture" object. The full original tree is shown in the main paper.

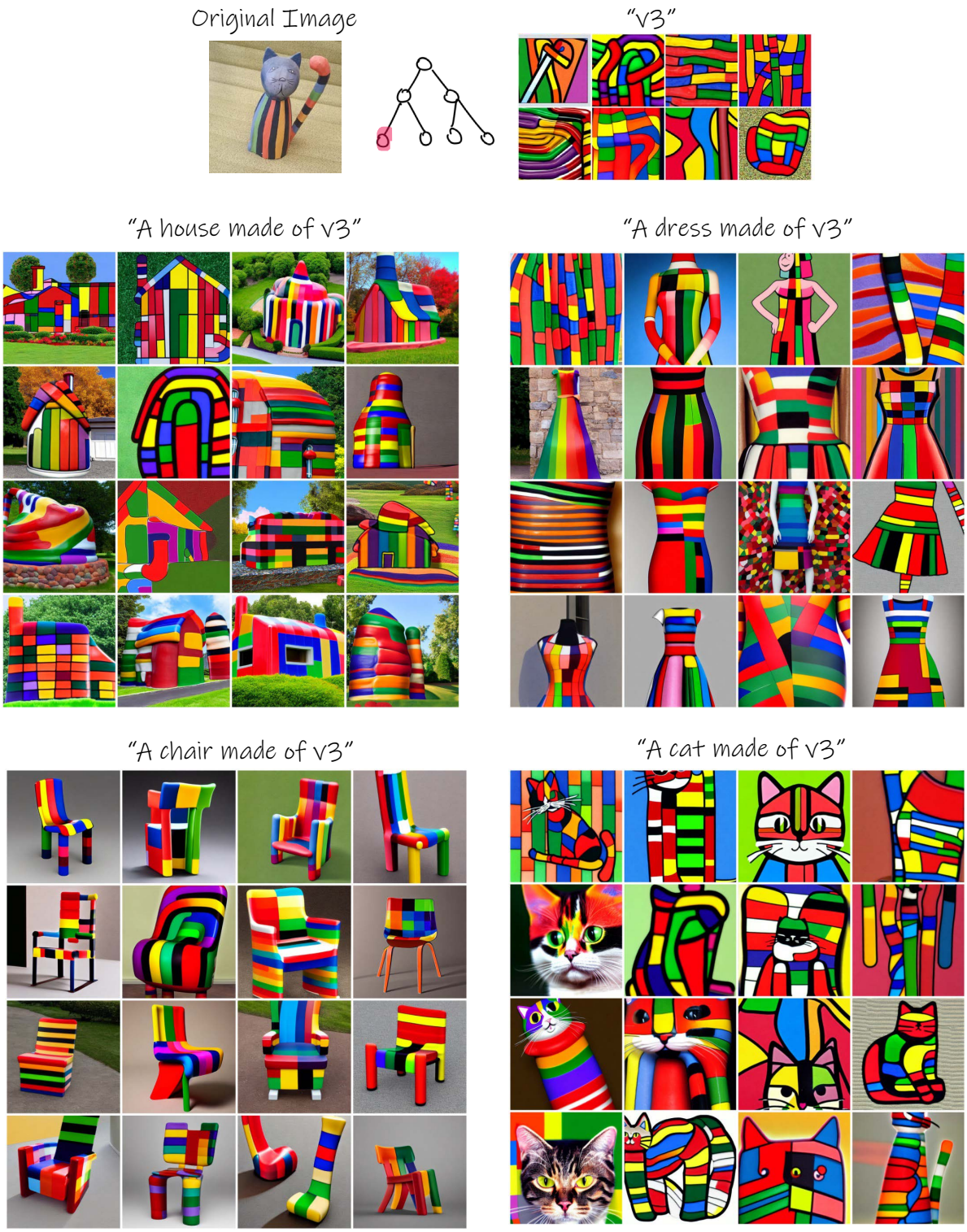


Figure 24. More examples of text based generation for the "cat sculpture" object. The full original tree is shown in the main paper.

Figure 25. More examples of text based generation for the "wooden saucer bear" object. The full original tree is shown in the main paper.

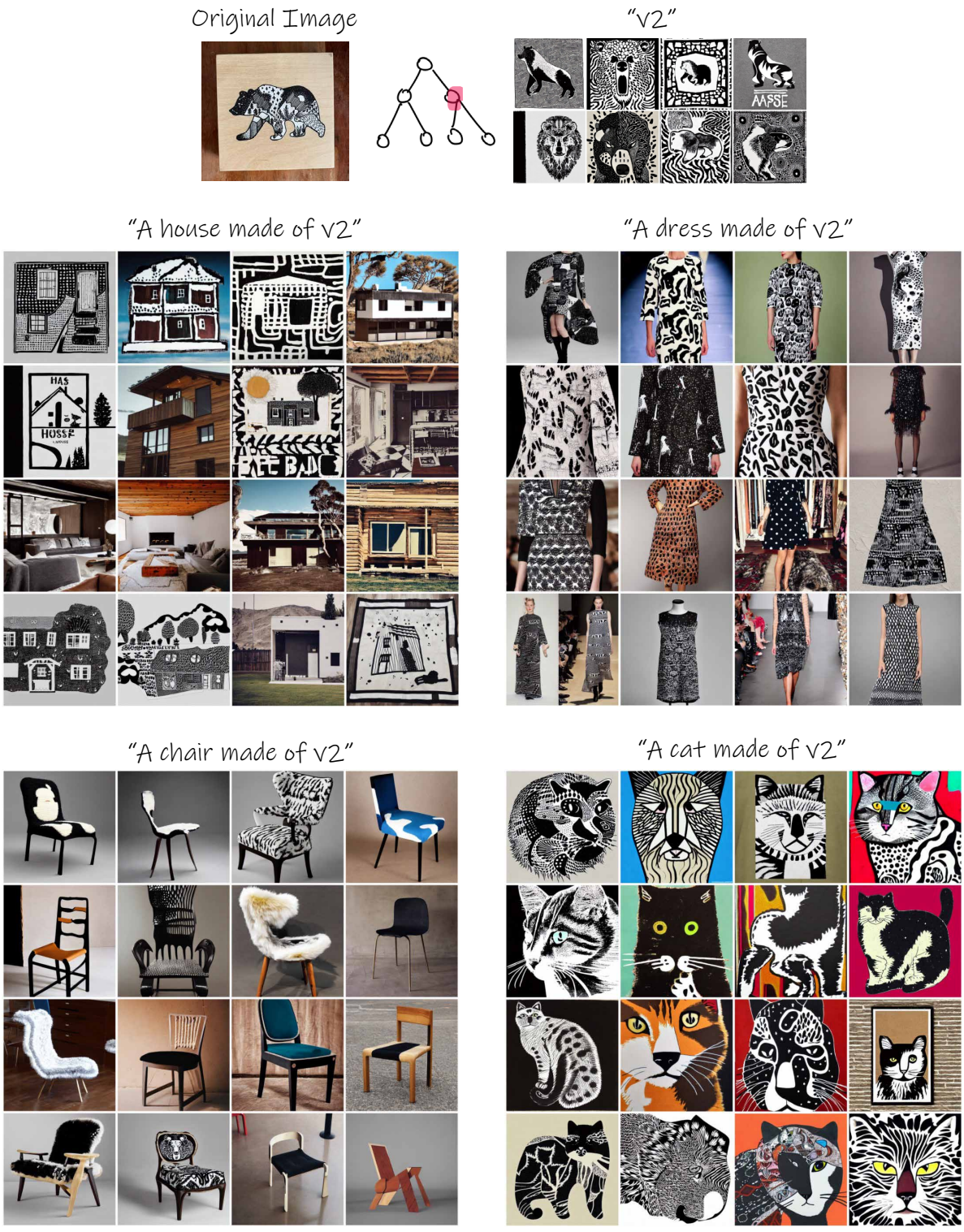


Figure 26. More examples of text based generation for the "wooden saucer bear" object. The full original tree is shown in the main paper.

Figure 27. More examples of text based generation for the "wooden saucer bear" object. The full original tree is shown in the main paper.

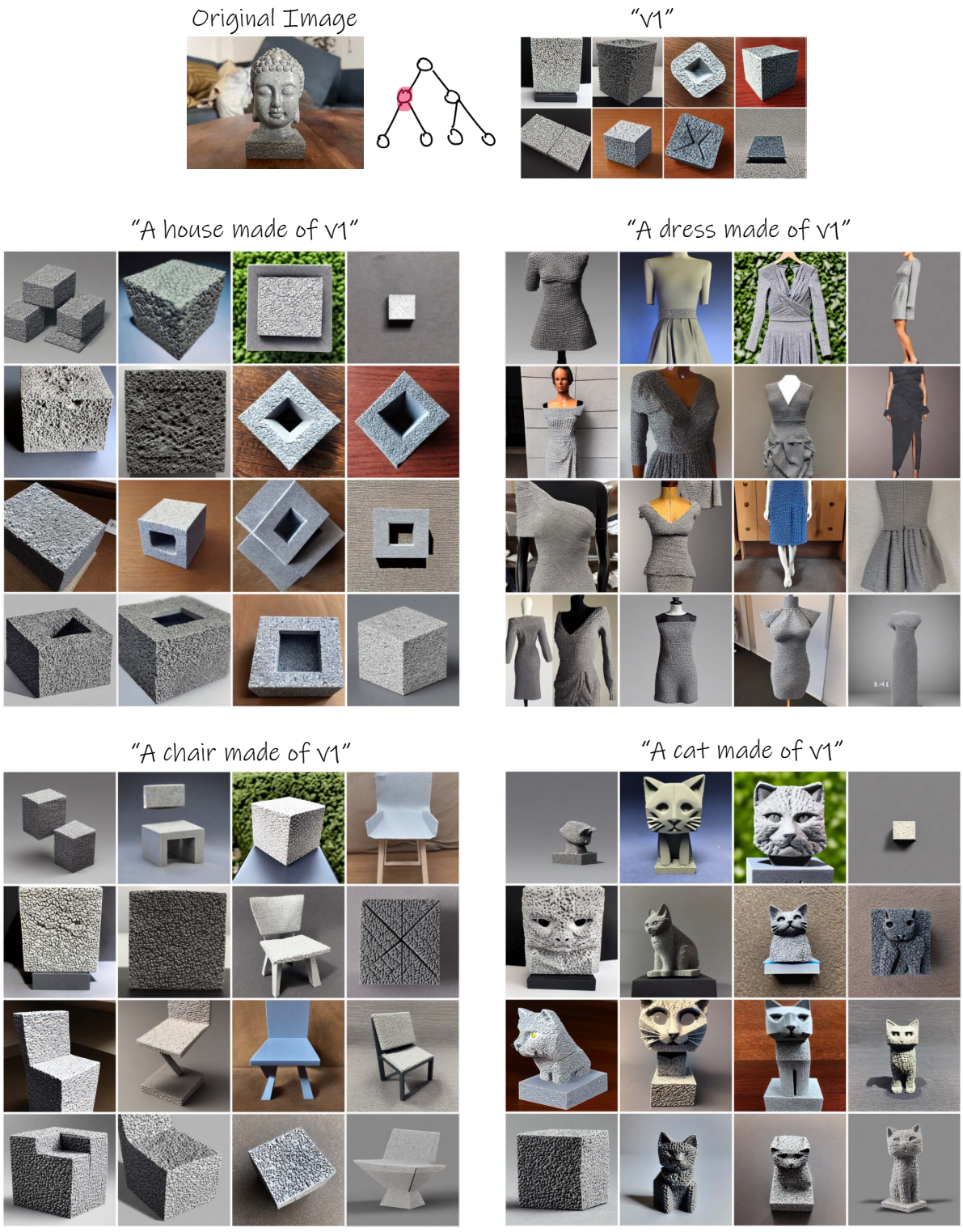


Figure 28. More examples of text based generation for the "Buddha sculpture" object.

References

[1] 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. 1, 2

[2] Ziqi Huang, Tianxing Wu, Yuming Jiang, Kelvin C.K. Chan, and Ziwei Liu. ReVersion: Diffusion-based relation inversion from images. *arXiv preprint arXiv:2303.13495*, 2023. 3, 4

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[4] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 1

[5] Andrey Voynov, Q. Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-to-image generation. *ArXiv*, abs/2303.09522, 2023. 1, 2