ReConText3D: Replay-based Continual Text-to-3D Generation

Supplementary Material

This supplementary document provides additional details supporting the main paper. We first present further information on the design and statistics of the **Toys4K-CL** benchmark (Sec. 8), including semantic grouping and long-tailed class distributions. We then describe our **ReConText3D replay construction** and illustrate how budget allocation affects class coverage (Sec. 9). Next, we report full **class-wise quantitative results** for both TRELLIS-XL and Shap-E across base and novel categories (Sec. 10). Finally, we include extended **qualitative comparisons** and highlight representative failure cases (Sec. 11). Together, these materials complement the main paper with detailed analysis and visualizations.

8. Additional Details on Toys4K-CL Benchmark

In this section, we present the additional details regarding our **Toys4K-CL Benchmark**.

Benchmark details. Toys4K-CL consists of 45 base and 45 novel classes selected to maximize semantic diversity while preserving the natural long-tailed distribution of the Toys4K dataset [37]. Based on an analysis of the dataset's taxonomy, we observed that assets naturally cluster into the following broad semantic groups: **furniture/households** (*chair*, *sofa*, *fridge*), **tools/utensils** (*hammer*, *screwdriver*, *pencil*), **vehicles** (*airplane*, *car*, *truck*), **animals/creatures** (*dog*, *fox*, *dragon*), **food** (*banana*, *cookie*, *pizza*), and **instruments/others** (*guitar*, *piano*, *monitor*), and constructed splits such that each contains categories spanning all groups. This ensures broad variation in geometry, appearance, and topology across both base and novel classes. Tab. 2 reports per-split statistics, including the counts of classes, train, test, and total samples. Fig. 7 visualizes the class-frequency distributions for base and novel splits. Both splits exhibit similar long-tailed behaviour, reflecting realistic class imbalance conditions.

Table 2. **Toys4K-CL Benchmark Statistics.**

Split	Classes	Train	Test	Total
Base	45	1352	225	1577
Novel	45	1243	225	1468

9. Additional Details on ReConText3D Replay Creation

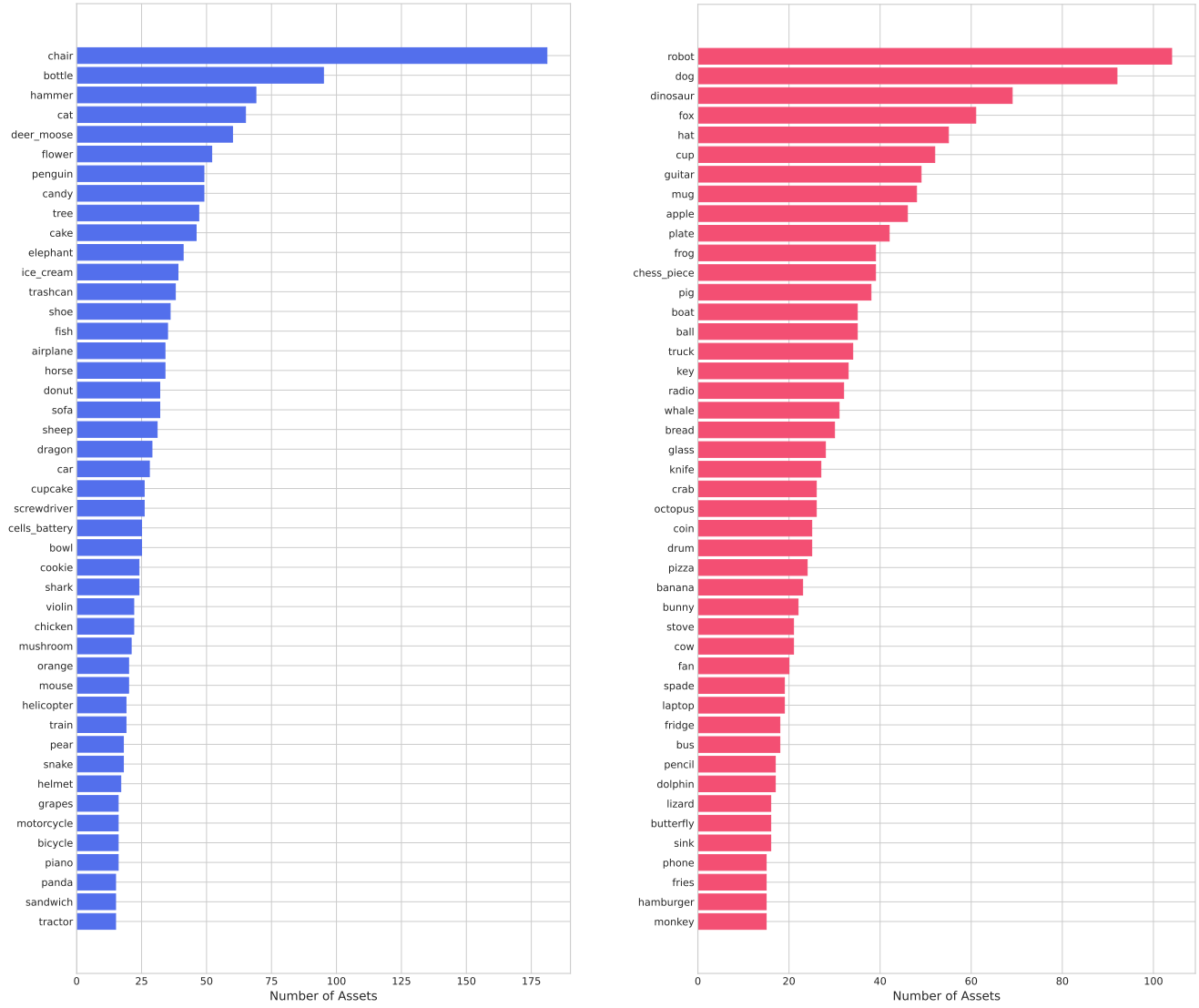
Our ReConText3D replay strategy provides a principled balance between semantic coverage and class proportionality of the replayed examplers. Figure 8 shows that our count-aware allocation mitigates long-tail bias by allowing smooth budget growth while preventing large classes from monopolizing memory.

10. Additional Quantitative Results

In this section, we provide detailed class-wise CLIP scores for both base and novel categories on the Toys4K-CL benchmark, complementing the aggregated results in the main paper.

Base-class performance. Table 3 reports class-wise CLIP [29] scores on base-class assets for both **TRELLIS-XL** [45] (left) and **Shap-E** [13] (right) across the baselines. For each backbone, the *Ours* column (ReConText3D replay) is highlighted in light purple. Overall, ReConText3D either matches or improves over naive fine-tuning and L2-SP [46] on the vast majority of classes, while remaining competitive with or close to the base-training upper bound. On TRELLIS-XL, this trend is particularly visible for categories that are prone to strong forgetting, such as *airplane*, *chair* and *grapes*, where fine-tuning substantially degrades CLIP similarity, whereas ReConText3D restores or surpasses the original base-training scores. On Shap-E, ReConText3D similarly offers consistent gains or stability across diverse object types (e.g. *hammer*, *panda*, *pear*), confirming that text-space replay remains effective even for diffusion-based generators. The joint behavior of L2-SP+Ours also illustrates the complementary effect of weight regularization and semantic replay: in some classes L2-SP+Ours can be slightly higher, whereas in many others, ReConText3D alone achieves the best performance.

Novel-class performance. Table 4 presents the corresponding CLIP scores for novel-class assets. Here again, we compare fine-tuning, L2-SP, L2-SP+Ours, and our replay strategy for both TRELLIS-XL and Shap-E, with the ReConText3D columns highlighted. While the primary goal of replay is to preserve base-class performance, the results show that ReConText3D also maintains strong or even improved performance on novel classes. For many categories (e.g., *apple*, *fries*, *pencil*, *whale*), the ReConText3D configuration either achieves the best CLIP score or remains very



(a) Base classes

(b) Novel classes

Figure 7. **Class distribution statistics for the Toys4K-CL benchmark (train and test).** Both splits exhibit similar long-tailed behaviour.

close to the fine-tuning baseline, indicating that replay does not overly constrain plasticity. This pattern is consistent across both backbones and across both simple and visually complex categories (such as *chess_piece*, *glass*, *radio*), supporting our claim that semantic replay can balance stability and plasticity without sacrificing generative alignment for novel concepts.

Overall, the class-wise results on base and novel assets demonstrate that ReConText3D yields robust improvements across a wide range of object categories and model families. Rather than benefiting only a small subset of classes, the replay mechanism consistently stabilizes base performance while preserving (and often enhancing) novel-class quality, providing fine-grained evidence for the global trends reported in the main paper.

Failure cases and challenging categories. While ReConText3D delivers consistent improvements across most classes, the class-wise results also reveal a small number of categories where replay remains challenging. These cases typically arise in highly fine-grained or visually ambiguous object types, or in categories with inherently low intra-class consistency. For example, on TRELLIS-XL, classes such as *dragon*, *mouse*, and *tractor* show only modest gains or slight reductions compared to the strongest baseline, suggesting that their visual and geometric variability may reduce the benefit of text-embedding replay. Similarly, on Shap-E, categories such as *helicopter*, *cat*, and *orange* ex-

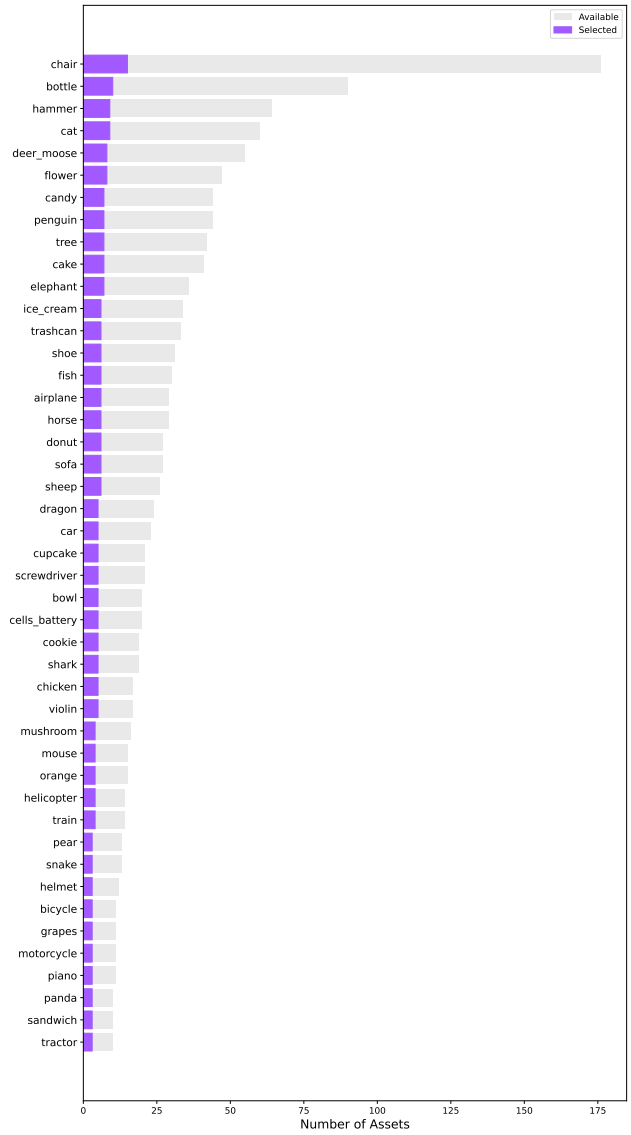


Figure 8. Number of base class assets (replay examplers) returned by our count-aware budget allocation

hibit limited improvements.

11. Additional Qualitative Results

In this section, we provide extended qualitative comparisons. While CLIP scores are a useful metric for semantic alignment, qualitative inspection remains essential for assessing geometric fidelity, texture realism, structural consistency, and the extent of catastrophic forgetting in continual text-to-3D generation.

Base-class performance. Figures 9 and 11 visualize generations for base classes on TRELLIS-XL and Shap-E, respectively. When fine-tuned on novel classes, both models exhibit clear signs of catastrophic forgetting, but in distinct ways. TRELLIS-XL frequently mixes or shifts the semantic class of the object, indicating that category-level information is easily overwritten during continual updates. For example, in row 2 of Figure 9, the fine-tuned TRELLIS-XL model, and even L2-SP, changes a *donut* into a *bread*-like object. In row 3 both these models again alter a *tractor* into something closer to a *truck*. In contrast, our ReConText3D approach correctly preserves the original base-class semantics in both cases, reliably generating a *donut* and a *tractor* with appropriate geometry and structure even after novel-class training.

Shap-E behaves slightly differently. It tends to retain the correct class identity more often but suffers from degraded geometry, yielding structurally weakened outputs. Nonetheless, Shap-E is not fully immune to semantic drift. Row 3 of Figure 11 shows a notable case where the model appears to confuse an *airplane* with a *tractor*, blending features from both categories.

Novel-class performance. Figures 10 and 12 present qualitative results for novel classes on TRELLIS-XL and Shap-E. Overall, fine-tuning performs well on many newly introduced categories and often produces good stage-2 assets. However, there are notable cases where our ReConText3D approach surpasses the fine-tuned models in visual fidelity or semantic accuracy. For example, in row 6 of Figure 10, the fine-tuned TRELLIS-XL model generates an *octopus* with an incorrect orange coloration, whereas our method produces the intended purple version with a more coherent overall structure.

Failure Cases. While ReConText3D largely preserves semantic and structural fidelity for both base and novel classes, there are still occasional failure modes. For instance, Figure 9 row 3 shows a *tractor* that is correctly classified but generated in grey rather than the intended red, indicating a loss of color information. Similarly, Figure 10 row 4 depicts a *bus* generated in green instead of red.

These examples highlight that while our semantic replay strategy effectively stabilizes class identity and structure, it can sometimes fail to retain fine-grained visual attributes such as color. Addressing such limitations could further enhance the fidelity of stage-1 and stage-2 assets in future work.

Table 3. **Class-wise Evaluation (CLIP (↑))** on Base-Class assets for TRELLIS-XL and Shap-E. Best scores are highlighted in **bold**, whereas our ReConText3D results are highlighted in purple. Best scores are shown excluding Base Training.

Class	*TRELLIS-XL*					*Shap-E*				
	Base Training	Fine-tuning	L2-SP	L2SP + Ours	**Ours**	Base Training	Fine-tuning	L2-SP	L2SP + Ours	**Ours**
airplane	29.37	22.83	21.52	**30.85**	30.77	30.83	25.14	23.09	**30.54**	29.76
bicycle	27.21	18.12	19.04	**25.88**	25.36	27.18	26.17	23.33	25.40	**26.19**
bottle	32.96	28.54	24.38	**30.66**	29.75	31.52	**31.37**	31.30	30.53	31.18
bowl	33.50	30.97	31.38	**33.94**	33.78	33.86	32.81	32.81	33.68	**33.85**
cake	28.12	21.60	21.68	27.06	**27.35**	29.28	28.63	28.70	**29.57**	29.49
candy	34.72	26.95	28.99	**29.06**	28.33	34.64	32.20	32.60	**34.08**	33.88
car	26.87	**28.36**	28.13	27.71	28.35	28.32	27.29	27.01	28.02	**28.40**
cat	26.18	25.21	26.18	24.48	**26.94**	28.45	27.78	27.85	**27.90**	27.20
cells_battery	30.87	26.71	26.75	31.16	**31.58**	29.61	29.46	28.72	**30.94**	30.15
chair	30.51	20.55	20.57	**29.66**	29.31	31.18	29.94	28.60	**32.16**	31.32
chicken	28.31	21.44	21.78	**27.47**	27.43	27.31	26.66	25.32	**26.40**	26.34
cookie	29.98	27.08	29.99	**31.74**	30.58	29.38	29.29	29.59	29.58	**30.23**
cupcake	29.17	20.63	21.68	23.69	**24.18**	31.92	**30.87**	29.80	29.89	29.42
deer_moose	31.80	29.22	28.65	29.63	**30.26**	30.29	28.92	28.30	29.45	**30.31**
donut	29.77	25.15	22.05	**30.54**	28.06	32.71	**31.72**	31.63	31.29	31.15
dragon	28.27	24.85	24.95	25.21	**25.24**	26.22	**26.68**	25.11	26.26	25.97
elephant	32.75	27.79	24.62	**31.15**	30.88	32.57	30.35	29.36	**31.32**	31.08
fish	28.71	25.62	25.56	**27.66**	27.39	28.90	27.53	25.96	**27.98**	27.19
flower	28.70	21.81	22.29	27.91	**28.09**	25.46	25.25	25.12	**25.59**	25.37
grapes	29.61	19.45	19.17	31.02	**31.07**	25.24	24.36	24.55	**25.45**	25.07
hammer	29.48	24.39	27.22	**28.60**	27.38	27.27	28.86	28.70	28.70	**29.69**
helicopter	30.33	20.38	20.27	29.01	**30.03**	25.10	23.34	22.56	**24.22**	22.52
helmet	26.64	23.13	23.26	26.28	**27.11**	24.42	23.87	23.78	**24.06**	23.96
horse	28.13	23.33	23.28	25.38	**26.65**	26.19	24.89	25.56	24.73	25.71
ice_cream	31.47	25.74	25.92	28.21	**30.07**	34.74	**35.20**	35.17	33.46	34.27
motorcycle	30.47	22.35	22.41	**31.03**	**31.03**	27.06	24.91	24.70	**25.47**	25.33
mouse	31.39	28.86	27.71	**29.10**	27.99	29.56	27.92	28.45	28.45	**28.88**
mushroom	31.97	27.57	28.14	29.79	**29.89**	34.40	33.92	32.14	33.93	**34.63**
orange	30.58	27.48	26.99	27.92	**28.03**	29.54	30.37	**30.63**	29.48	29.37
panda	29.64	27.27	26.64	**31.62**	30.94	29.26	25.53	26.74	27.26	**28.72**
pear	33.55	28.20	28.44	**33.33**	32.99	32.94	31.48	31.29	32.46	**33.67**
penguin	29.55	21.92	22.57	**29.01**	28.52	29.44	27.00	27.15	29.38	**30.01**
piano	29.73	18.69	21.43	**24.14**	23.62	16.98	**19.86**	19.19	19.77	18.24
sandwich	29.23	22.64	25.30	**27.53**	24.81	30.74	29.03	28.85	**31.36**	30.64
screwdriver	28.08	26.28	25.62	26.08	**26.44**	29.77	**29.64**	29.56	29.43	29.52
shark	31.83	30.29	30.30	30.35	**30.65**	31.92	**31.33**	31.17	30.91	31.02
sheep	28.89	25.82	25.68	27.42	**28.43**	28.07	28.46	27.99	**29.13**	28.86
shoe	25.52	15.87	16.06	**23.59**	23.38	22.41	21.13	19.90	22.70	**22.87**
snake	28.67	24.19	23.40	**27.22**	26.59	27.61	24.00	23.80	**25.27**	25.03
sofa	26.56	17.98	16.64	27.90	**27.94**	26.46	25.51	24.55	25.45	**26.09**
tractor	31.25	27.66	26.10	26.82	**27.72**	29.19	24.85	24.75	**26.63**	26.09
train	22.65	**26.15**	25.00	22.18	22.36	24.63	**25.55**	25.11	25.19	24.95
trashcan	32.19	26.56	28.35	31.53	**32.14**	30.26	**32.05**	31.46	31.31	31.70
tree	27.98	18.59	18.98	**28.13**	27.94	22.44	22.13	**23.01**	22.57	21.56
violin	28.68	26.58	27.58	**31.27**	31.00	28.51	27.73	27.32	27.65	**28.28**

Table 4. **Class-wise Evaluation on Novel-Class Assets (CLIP (↑))** for both TRELLIS-XL and Shap-E. Best scores are highlighted in **bold**, whereas our ReConText3D results are highlighted in purple.

Class	*TRELLIS-XL*				*Shap-E*			
	Fine-tuning	L2-SP	L2SP + Ours	**Ours**	Fine-tuning	L2-SP	L2SP + Ours	**Ours**
apple	33.14	34.09	32.03	**33.35**	32.17	31.56	32.27	**32.37**
ball	29.32	31.29	31.24	**31.41**	28.29	28.15	27.54	**28.44**
banana	33.03	33.00	**33.15**	33.03	32.23	31.79	32.66	**33.14**
boat	29.36	27.53	29.00	**29.99**	**23.82**	22.92	21.67	21.46
bread	30.65	**30.86**	30.35	30.75	**27.42**	26.68	26.10	26.01
bunny	28.53	**29.43**	28.86	28.50	**29.94**	29.26	28.67	29.30
bus	26.24	**28.38**	27.16	27.57	**23.83**	23.20	22.79	22.65
butterfly	**29.04**	28.55	28.52	27.96	27.03	27.00	27.77	**29.04**
chess_piece	**36.07**	35.81	34.40	35.82	35.16	34.89	**35.30**	35.18
coin	26.37	**26.72**	24.41	26.21	25.80	24.73	**26.72**	25.29
cow	**32.96**	32.43	30.41	31.40	29.58	**29.76**	29.01	29.69
crab	32.11	29.72	**32.84**	32.50	**28.92**	27.78	27.25	28.57
cup	29.57	29.93	**29.98**	29.60	30.92	30.25	**31.71**	31.13
dinosaur	27.63	27.23	26.59	**28.19**	**27.59**	27.04	27.43	26.93
dog	**28.10**	27.56	26.79	26.72	**27.21**	27.20	27.06	27.13
dolphin	32.09	31.94	**32.42**	31.79	31.93	31.87	31.16	**32.11**
drum	26.58	26.52	26.68	**28.45**	28.62	27.97	**29.25**	28.12
fan	**28.05**	27.82	27.80	27.75	25.12	24.77	25.46	**26.63**
fox	29.61	**30.11**	30.00	29.44	**29.58**	28.68	29.52	28.88
fridge	**28.04**	27.18	26.75	27.46	**25.63**	25.39	25.22	25.21
fries	30.96	30.81	31.41	**31.79**	32.18	30.30	32.09	**32.41**
frog	30.05	29.98	30.89	**31.58**	**27.75**	26.34	26.21	27.11
glass	34.16	33.87	**34.70**	34.46	32.90	**33.08**	32.38	32.97
guitar	28.87	28.66	**29.17**	28.90	28.25	28.17	**28.33**	28.08
hamburger	32.83	**33.30**	31.95	33.12	**32.86**	32.39	31.06	30.97
hat	**28.28**	25.74	27.51	27.31	27.28	26.63	27.69	**27.72**
key	28.37	**28.94**	28.12	28.50	28.89	28.84	29.90	**30.01**
knife	27.53	27.23	**27.80**	26.77	28.33	27.89	27.96	**28.33**
laptop	29.51	29.92	30.11	**30.15**	26.11	26.01	26.84	**26.91**
lizard	**27.48**	27.00	27.43	26.57	**26.45**	25.97	25.87	25.47
monkey	29.55	**29.62**	28.48	28.92	26.75	26.75	**27.00**	26.88
mug	30.15	29.81	30.09	**30.24**	**31.34**	31.12	31.01	31.23
octopus	**31.38**	29.15	29.30	30.19	28.12	28.18	**28.96**	28.68
pencil	31.27	31.81	30.87	**32.67**	30.45	30.00	31.16	**31.31**
phone	26.88	21.89	24.50	**27.33**	21.47	22.70	21.14	**22.07**
pig	**33.83**	32.20	33.22	32.19	**32.97**	32.32	32.18	31.89
pizza	31.12	**31.87**	31.32	29.56	**32.77**	32.44	32.43	32.41
plate	31.57	32.08	**33.12**	32.46	32.55	32.08	**33.13**	33.06
radio	24.07	**25.56**	23.82	25.45	24.71	23.39	**24.99**	23.82
robot	29.16	28.70	28.14	**30.08**	**29.02**	28.49	26.81	27.47
sink	**31.12**	31.11	30.05	31.11	27.95	27.01	**29.23**	28.43
spade	28.88	29.72	29.12	**30.08**	**30.71**	30.08	29.21	30.07
stove	28.69	**29.70**	28.24	28.80	25.34	25.16	25.37	**25.62**
truck	27.75	26.40	**28.08**	25.59	**24.89**	24.76	24.65	24.01
whale	31.33	32.01	31.42	**32.16**	30.47	31.23	**31.69**	31.01

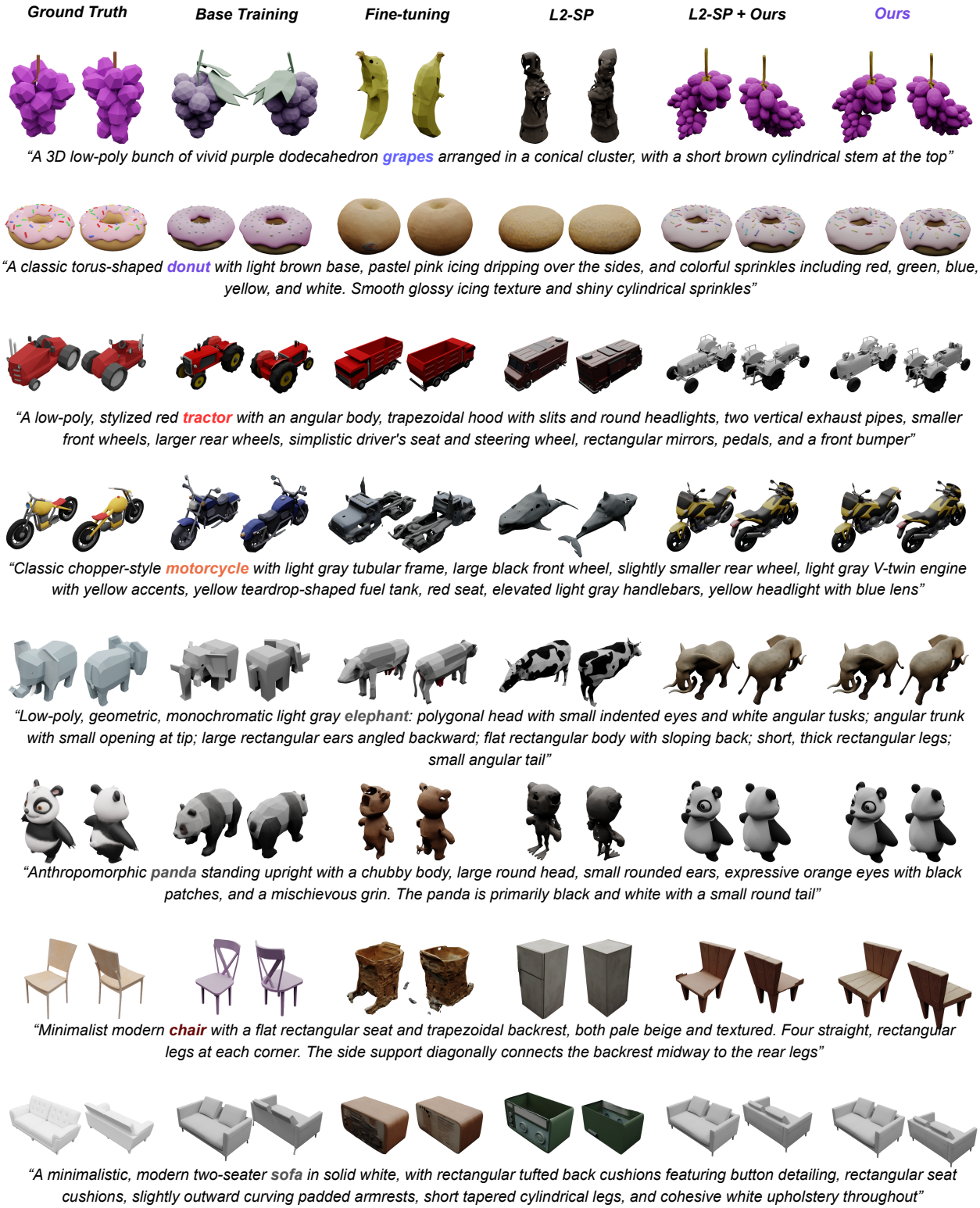


Figure 9. Qualitative comparison of continual text-to-3D baselines against ReConText3D (Ours) on **base class** assets using **TRELLIS-XL**.

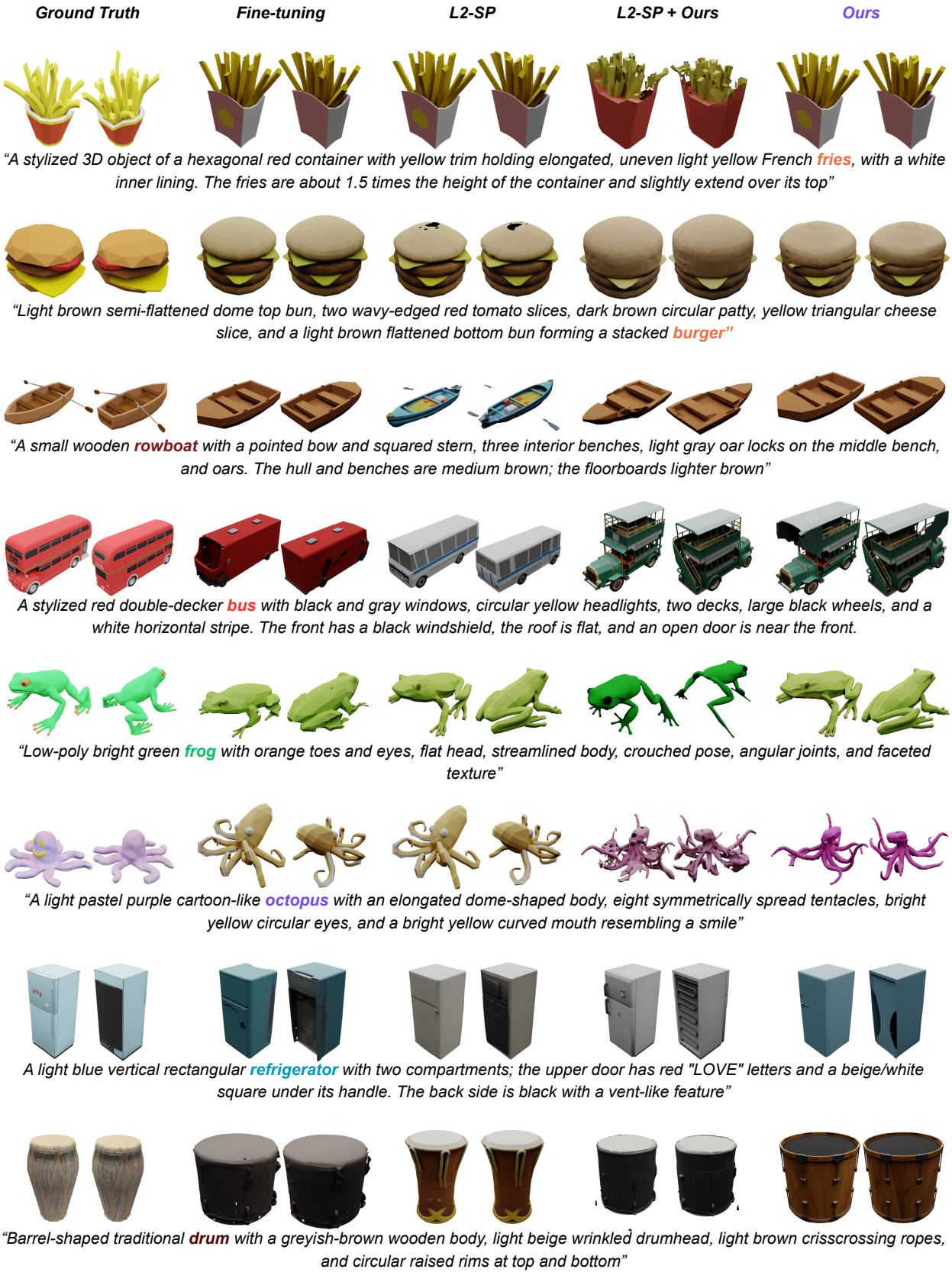


Figure 10. Qualitative comparison of continual text-to-3D baselines against ReConText3D (Ours) on **novel class** assets using **TRELLIS-XL**.

Figure 11. Qualitative comparison of continual text-to-3D baselines against ReConText3D (Ours) on **base class** assets using **Shap-E**.

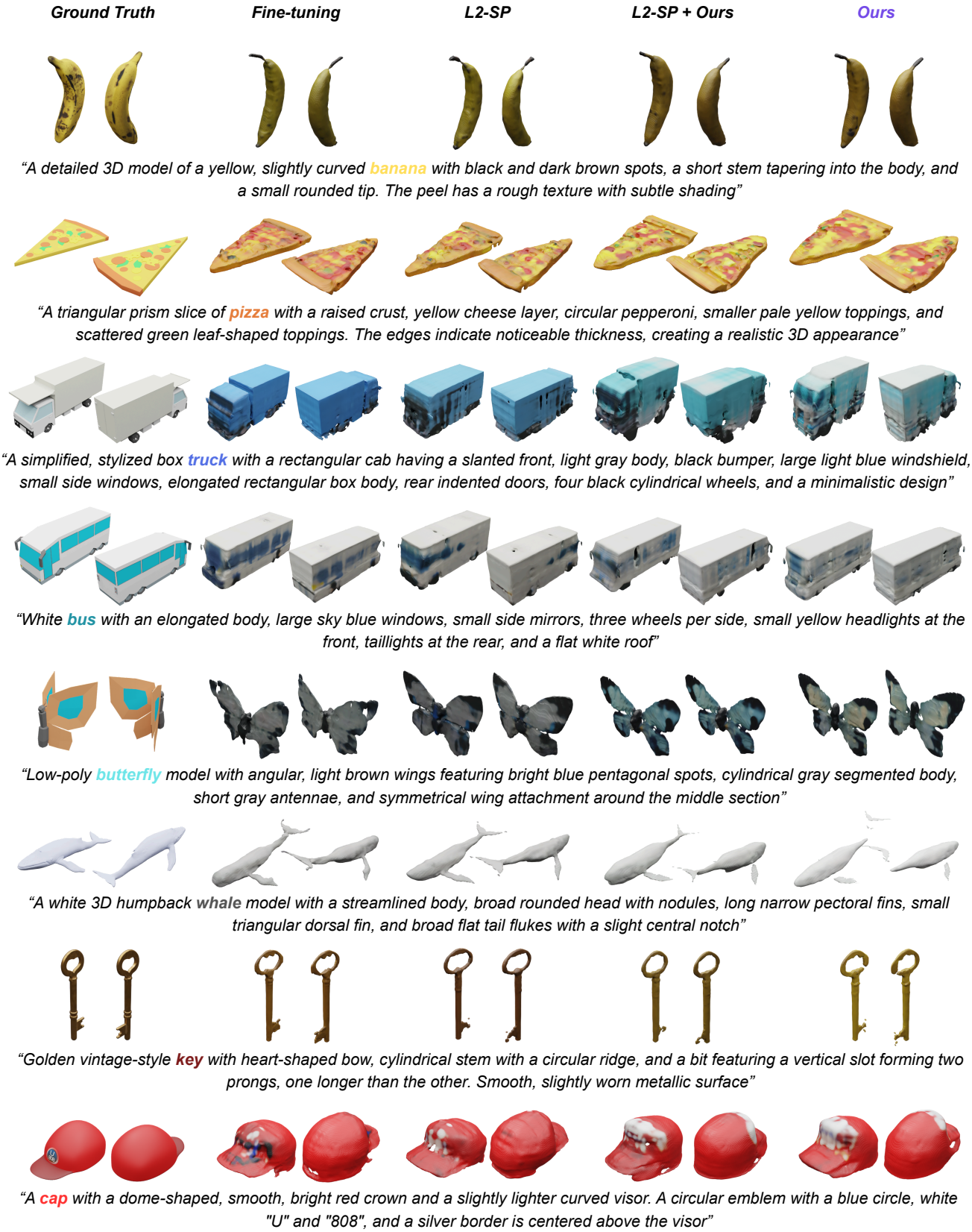


Figure 12. Qualitative comparison of continual text-to-3D baselines against ReConText3D (Ours) on **novel class** assets using **Shap-E**.