This CVPR Findings paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

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

Muhammad Ahmed Ullah Khan[1,2†] Muhammad Haris Bin Amir[2] Didier Stricker[1] Muhammad Zeshan Afzal[1]

[1]DFKI [2]RPTU Kaiserslautern-Landau

muhammad_ahmed_ullah.khan@dfki.de haris.amir@edu.rptu.de

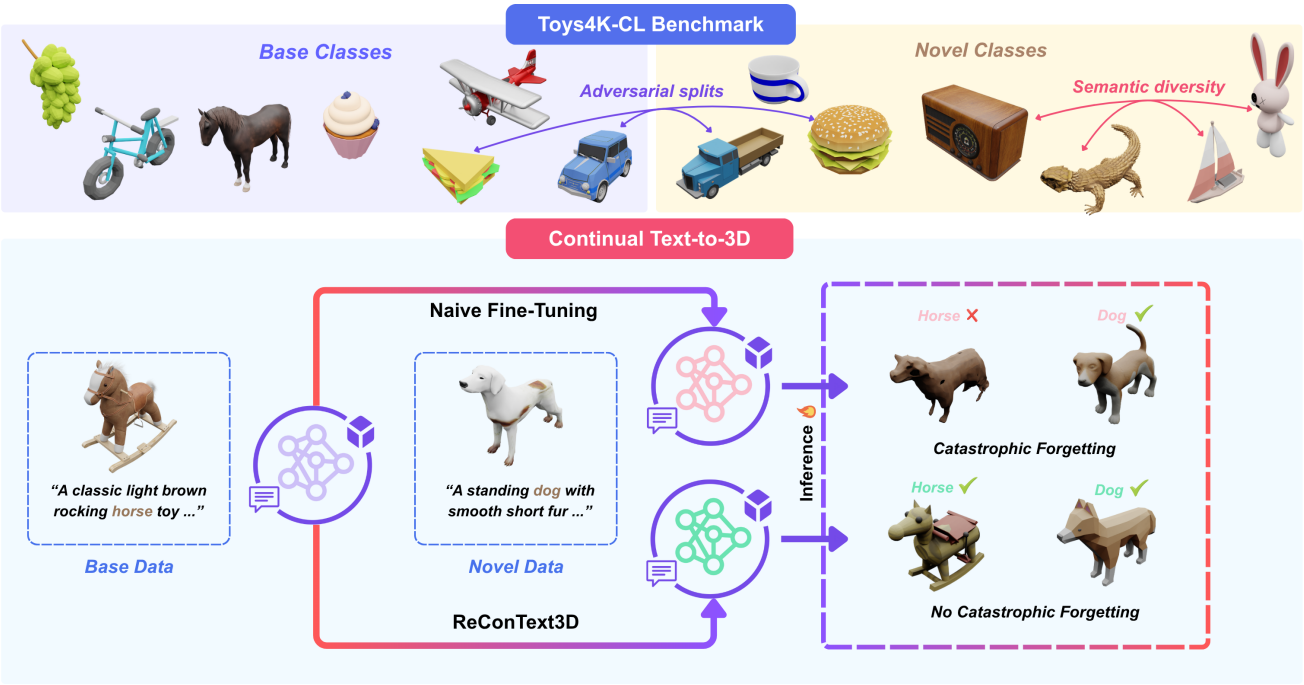

Figure 1. **Top:** The Toys4K-CL benchmark splits data into base classes and novel classes with adversarial splits and semantic diversity while maintaining a comparable number of classes and total assets across both stages. **Bottom:** Continual text-to-3D generation. Naive fine-tuning on novel data, in this case dog, leads to catastrophic forgetting on base class horse, while continual learning mitigates forgetting, preserving previously learned 3D concepts during model updates.

Abstract

*Continual learning enables models to acquire new knowledge over time while retaining previously learned capabilities. However, its application to text-to-3D generation remains unexplored. We present **ReConText3D**, the first framework for continual text-to-3D generation. We first demonstrate that existing text-to-3D models suffer from catastrophic forgetting under incremental training. ReConText3D enables generative models to incrementally learn new 3D categories from textual descriptions while preserving the ability to synthesize previously seen assets. Our method constructs a compact and diverse replay memory through text-embedding k-Center selection, allowing representative rehearsal of prior knowledge without modifying the underlying architecture. To systematically evaluate continual text-to-3D learning, we introduce **Toys4K-CL**, a benchmark derived from the Toys4K dataset that provides balanced and semantically diverse class-incremental splits. Extensive experiments on the Toys4K-CL benchmark show that **ReConText3D** consistently outperforms all baselines across different generative backbones, maintaining high-quality generation for both old and new classes. To the best of our knowledge, this work establishes the first continual learning framework and benchmark for text-to-3D generation, opening a new direction for incremental 3D generative modeling. Project page is available at:*

†Corresponding Author.

https://mauk95.github.io/ReConText3D/.

1. Introduction

Generative models have made rapid progress in synthesizing 3D assets directly from text, enabling compelling applications in content creation, simulation, and robotics [13, 45]. Text-to-3D pipelines based on diffusion or flow objectives can now produce high-quality 3D assets with increasingly faithful text alignment and geometric fidelity [5, 17, 26, 43]. Despite these advances, the dominant training paradigm remains *static*; models are trained once on large, curated datasets and then frozen. In real-world settings, however, a 3D generator must continuously learn new concepts, i.e, categories, shapes, materials, etc., without sacrificing performance on previously learned assets.

Continual learning (CL) addresses this challenge by training models sequentially over tasks while mitigating *catastrophic forgetting*, the degradation of prior knowledge when learning new data [23, 25]. A rich body of work has explored CL for recognition, including rehearsal-based methods that store exemplars [30], knowledge distillation [15], and regularization [4, 15, 47]. In 3D perception, recent efforts such as SDCoT [48] and SDCoT++ [49] extend class-incremental learning to 3D object detection with teacher–student designs and model-agnostic evaluations.

Generative settings pose distinct challenges under continual learning, i.e, distribution shift emerges in both *text* and *shape* spaces, supervision is weakly aligned with perception metrics, and stability–plasticity trade-offs must be managed without sacrificing sample diversity. We argue that continual text-to-3D generation is both practically important and scientifically distinct. Practical pipelines must expand to new product lines or styles over time, while maintaining backward compatibility with existing asset libraries for games, AR/VR, and simulation [12, 14]. Scientifically, the conditioning pathway (text encoder) and the generative pathway (3D decoder) co-evolve during fine-tuning; naive adaptation on novel classes often drifts the shared conditioning manifold, degrading alignment and appearance for old classes even when the architecture is expressive.

To the best of our knowledge, **continual learning for *text-to-3D generation*** has not been studied. In this work, we introduce **ReConText3D**, the first framework for *continual text-to-3D generation*. ReConText3D is *model-agnostic* and can be plugged into any text-to-3D backbone. The core idea is a simple but effective *semantic replay*: we construct a compact memory of base exemplars using (i) a *count-aware budget allocation* to respect long-tailed data and (ii) *text-embedding k-center selection* to maximize semantic coverage in the prompt space. During novel-stage training, replayed base prompts are mixed with novel data, anchoring the text–shape mapping and mitigating drift, without changing the underlying generative objective or architecture.

To evaluate this new setting, we propose **Toys4K-CL**, a class-incremental benchmark derived from the captioned Toys4K subset of TRELLIS-500K [37, 45]. We retain 90 well-populated classes and form balanced, semantically diverse base/novel splits, including adversarial arrangements that separate closely related categories to stress interference. We report text alignment, appearance, and geometry metrics following prior work [28, 29, 45], and analyze forgetting explicitly.

Our **contributions** can be summarized as follows:

- **Problem & framework.** We are the first to formulate *continual text-to-3D generation* and present **ReConText3D**, a simple, model-agnostic framework for continual 3D asset synthesis.
- **Forgetting analysis.** We empirically demonstrate that state-of-the-art text-to-3D models suffer from severe catastrophic forgetting under class-incremental training.
- **Replay method.** We propose a novel replay-based CL method that combines *count-aware budget allocation* with *text-embedding k-center* selection to build compact, semantically diverse memories that mitigate forgetting.
- **Benchmark.** We introduce **Toys4K-CL**, the first benchmark for *continual text-to-3D generation* with balanced and adversarial base/novel splits .
- **Results.** ReConText3D significantly reduces forgetting and improves overall generation quality across multiple backbones, validating its model-agnostic design.

2. Related Work

Text-to-3D. Recent advances in text-to-3D generation largely fall into two architectural categories, diffusion-based and flow-based generative models. Diffusion-based approaches build upon the success of 2D diffusion models by distilling their learned priors into 3D space. DreamFusion [26] pioneered this direction using Score Distillation Sampling (SDS) to optimize neural radiance fields under 2D diffusion guidance [32, 34] without requiring 3D supervision. Follow-up works [16, 17, 40, 41, 43] improved geometric consistency and efficiency. Recent models explore native 3D diffusion architectures that operate directly in learned 3D latent spaces. Shap-E [13] introduces a conditional diffusion model that maps text embeddings to implicit 3D representations, enabling rapid text-to-3D asset synthesis after training an encoder on large 3D datasets. 3DTopia-XL [6] extends this paradigm by scaling diffusion transformers on a novel primitive-based 3D representation which jointly encodes shape, texture, and material.

Flow-based approaches [2, 18, 19] have more recently emerged as a powerful class of generative models challenging the dominance of diffusion-based approaches. TRELLIS [45] employs a flow-transformer architecture trained on large-scale structured 3D latents.

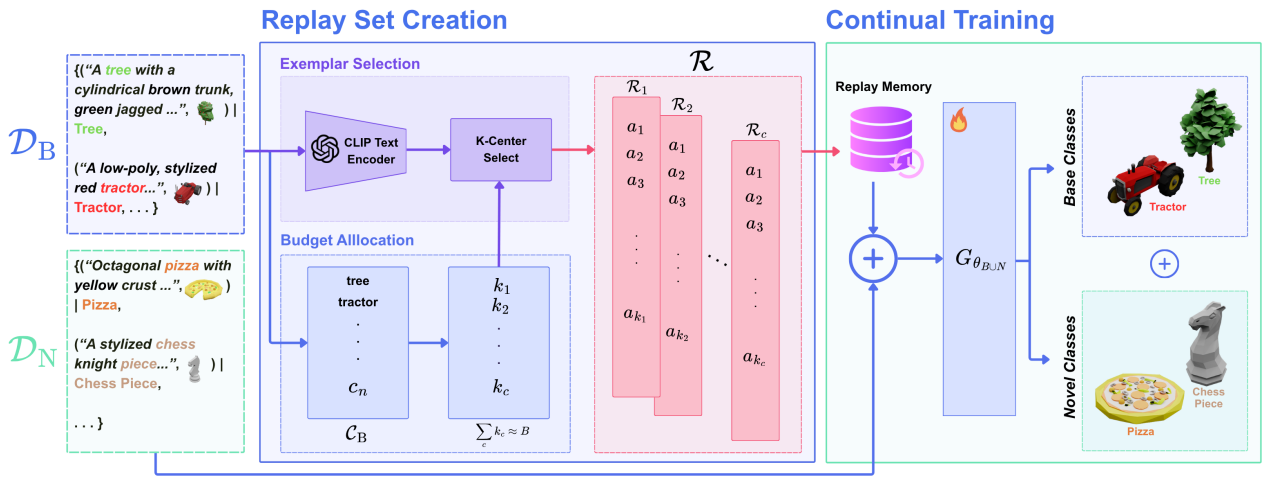

Figure 2. **Overview of the ReConText3D framework.** Base captions are encoded with CLIP and selected via k-center sampling under a count-aware budget to form replay memory $\mathcal{R}$, which is combined with novel data to incrementally train $G_{\theta_{B\cup N}}$ while preserving base-class synthesis.

Continual Learning. Continual learning aims to develop models capable of learning sequentially while mitigating catastrophic forgetting. Major strategies that alleviate forgetting include (1) Replay methods [3, 10, 24, 30, 35, 44] that replay old data exemplars during new task training, (2) Regularization-based methods [4, 7, 15, 47] that impose constraints on parameter updates to protect knowledge acquired from previous tasks (3) Structural methods [21, 22, 33] that expand the network architecture or isolate parameters for different tasks to prevent interference.

Continual Learning in 3D. Although continual learning strategies have been explored in the 2D visual domain, including recent advances in text-to-image diffusion models [9, 36, 38], their application in 3D is still relatively underexplored. 3D incremental learning has been explored mainly for classification and reconstruction tasks. For classification, methods such as I3DOL [8] introduce geometry-aware modules to preserve features across incremental classes, while others employ basic-shape pre-training [27], foundation-model adapters [1], or spectral exemplar selection [31] to reduce forgetting. For reconstruction, Thai et al. [42] reveal a positive knowledge transfer effect when sequentially learning 3D shapes from visual inputs. Despite these advances, to the best of our knowledge no prior work addresses continual learning in text-to-3D generation.

3. Methodology

3.1. Problem Definition

In the continual text-to-3D generation setting, we adopt a two-stage (Base→Novel) protocol following standard practice in 3D continual learning [48, 50]. We consider two non-overlapping sets of classes: a **base class set** $\mathcal{C}_{\text{base}}$ and a **novel class set** $\mathcal{C}_{\text{novel}}$, where $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \varnothing$. The corresponding datasets are denoted by $\mathcal{D}_{\text{base}} = \{(t_i, O_i) \mid y_i \in \mathcal{C}_{\text{base}}\}$ and $\mathcal{D}_{\text{novel}} = \{(t_i, O_i) \mid y_i \in \mathcal{C}_{\text{novel}}\}$, where t_i represents a textual description (caption) and O_i is the corresponding 3D asset (mesh).

We define the task of **continual text-to-3D generation** as: given a text-to-3D generator G_{θ_B} (the base model) pre-trained on $\mathcal{D}_{\text{base}}$, our goal is to obtain an incremental model $G_{\theta_{B\cup N}}$ by training on $\mathcal{D}_{\text{novel}}$ such that the resulting model can generate 3D assets corresponding to both the base and novel classes, i.e., $\mathcal{C}_{\text{base}} \cup \mathcal{C}_{\text{novel}}$.

In this class-incremental setup, the model must learn new classes from $\mathcal{C}_{\text{novel}}$ while retaining generation capability for previously learned classes in $\mathcal{C}_{\text{base}}$. Following the criteria used in class-incremental learning for classification [30], during novel training, access to the complete base dataset is not permitted, although limited replay exemplars, i.e, captions may be used depending on the continual learning strategy. This setting simulates a realistic, continual 3D content creation scenario, where models encounter new object categories over time but must avoid catastrophic forgetting of previously learned categories.

3.2. ReConText3D Framework

The goal of the proposed **ReConText3D** framework is to enable **incremental 3D asset generation from textual descriptions** while mitigating catastrophic forgetting of previously learned classes. As described in Sec. 3.1, given the base dataset $\mathcal{D}_{\text{base}}$ and the novel dataset $\mathcal{D}_{\text{novel}}$, our objective is to train a text-to-3D generator $G_{\theta_{B\cup N}}$ that can generate high-quality 3D meshes for both $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$.

We first train a text-conditioned 3D generative model G_{θ_B} on the base dataset $\mathcal{D}_{\text{base}}$, learning to generate meshes $O_i \in \mathcal{O}$ conditioned on their textual descriptions $t_i \in \mathcal{T}$ for the base class set $\mathcal{C}_{\text{base}}$. Once G_{θ_B} converges, we enter the

continual learning stage, where the model is incrementally adapted to the novel dataset $\mathcal{D}_{\text{novel}}$ while retaining its ability to generate objects from $\mathcal{C}_{\text{base}}$.

Formally, this adaptation produces an updated generator:

$$\theta_{B\cup N} \leftarrow \text{CLTrain}\big(G_{\theta_B}, \mathcal{D}_{\text{novel}}, \mathcal{M}_{\text{replay}}\big),$$

where $\mathcal{M}_{\text{replay}}$ denotes a replay memory containing a subset of caption and 3D mesh pairs from $\mathcal{D}_{\text{base}}$. The combined data from the replay memory and the novel dataset, $\mathcal{M}_{\text{replay}} \cup \mathcal{D}_{\text{novel}}$, is used to train the incremental model $G_{\theta_{B\cup N}}$, enabling it to revisit representative base-class distributions and preserve generation quality across previously learned concepts.

Our framework is inherently **model-agnostic** and can be integrated with any text-conditioned 3D generative backbone, independent of the underlying architecture or training objective. It requires only that the model accepts textual prompts as conditioning input and outputs a 3D representation (e.g., mesh, point cloud, or implicit field). Recent text-to-3D backbones either use rectified flow-based methods or latent diffusion-based methods. In our experiments (Sec. 4), we demonstrate the versatility of ReConText3D by applying it to one representative model from each type of method, including *TRELLIS-XL* [45] from rectified flow methods and *Shap-E* [13] from latent diffusion methods. The overall **ReConText3D** framework, including our proposed replay strategy, is illustrated in Fig. 2, showing the incremental training process (base $\rightarrow$ novel).

3.3. ReConText3D: Replay Set Creation

To mitigate catastrophic forgetting during novel-class adaptation, we propose **ReConText3D**, a novel replay-based continual generation strategy that selectively reuses representative samples from the base dataset. The core idea is to construct a compact replay memory from the base training set $\mathcal{D}_{\text{base}}$ that preserves semantic diversity (**text-embedding k-center selection**) while respecting class imbalance (**count-aware allocation**) inherent to real-world 3D datasets such as Toys4K-CL.

ReConText3D builds the replay memory in two stages: (1) **budget allocation** and (2) **exemplar selection**, as illustrated in Algorithm 2 and Algorithm 3, respectively. The overall replay set creation process is summarized in Algorithm 1 and illustrated in Fig. 2.

Budget Allocation. In the allocation stage, we assign replay quotas to each base class based on the square-root of its sample count, using per-class minimum, maximum, and maximum percentage caps $(m_{\min}, m_{\max}, p_{\max})$. This $\sqrt{\textbf{count}}$-based allocation ensures that frequent classes do not dominate the replay memory while still maintaining adequate representation of underrepresented categories. The resulting per-class allocation $\{k_c\}$ satisfies $\sum_c k_c \approx B$,

Algorithm 1 ReConText3D CREATEREPLAYSET

Inputs: Base metadata with captions $\mathcal{A}$, replay percentage r, novel-set size N_{novel}, caps $(m_{\min}, m_{\max}, p_{\max})$
Outputs: Replay set $\mathcal{R}$ and associated metadata
$B \leftarrow \lfloor (r/100)\, N_{\text{novel}} \rfloor$
$B \leftarrow \min(B, \text{total available})$
$\{k_c\} \leftarrow$ ALLOCATEBUDGET($\mathcal{A}, B, m_{\min}, m_{\max}, p_{\max}$)
$\mathcal{R} \leftarrow$ SELECTKCENTER(A, $\{k_c\}$, $E(\cdot)$, M)
return $\mathcal{R}$

where B is the global replay budget.

Exemplar Selection. In the selection stage, ReConText3D performs **k-center selection** in the **text-embedding space** of asset captions to identify semantically diverse exemplars per class. We use the same text encoder as employed by most text-to-3D backbones, i.e., *CLIP ViT-L/14* [29], ensuring that the replay memory remains aligned with the textual conditioning space actually perceived by the generator during training. Each asset a_i is represented by the average of its caption embeddings. To allow generality, we denote by M the maximum number of captions considered per asset. We use all available captions ($M = 11$) for every asset in the Toys4K-CL benchmark. Given the per-class quota k_c, captions are embedded and normalized, and assets are greedily selected to maximize coverage of the text-embedding manifold using cosine similarity in CLIP space.

This design provides a principled balance between **semantic coverage** and **class proportionality**. Count-aware allocation mitigates long-tail bias by allowing smooth budget growth while preventing large classes from monopolizing memory. The k-center strategy ensures diverse semantic coverage so that the replay memory exposes the model to varied textual conditions during continual training.

Considering the constraints of replay-based continual learning, we maintain $B = 248$ samples, i.e replay ratio $r = 20\%$ of the novel-set size N_{novel}, with $m_{\min} = 3$, $m_{\max} = 20$, and $p_{\max} = 30\%$. These heuristics were empirically tuned to preserve diversity, avoid overrepresentation of high-frequency classes, and ensure smooth scaling of per-class allocations.

3.4. Toys4K-CL Benchmark

To systematically evaluate continual text-to-3D generation, we introduce the **Toys4K-CL** benchmark, a class-incremental benchmark derived from the Toys4K [37] subset of the TRELLIS-500K dataset [45]. We select Toys4K as our base due to its clear class annotations, diverse object categories, and its independence from the training data of recent text-to-3D models, making it well-suited for continual evaluation.

Algorithm 2 ReConText3D ALLOCATEBUDGET

Inputs: Base metadata grouped by class $\mathcal{A} = \{(c, \{a_i\})\}$, global budget B, caps $m_{\min}, m_{\max}, p_{\max} \in (0, 1]$
Outputs: Per-class replay allocation $\{k_c\}$ such that $\sum_c k_c \approx B$

```
Initialize n_c ← |{a_i ∈ c}|                    ▷ Class availability
TOTAL α
s ← 0
for all class c do
    u_c ← min(m_max, n_c, ⌊p_max n_c⌋)          ▷ Effective cap
    w ← clip(⌊α√n_c⌉, m_min, u_c)
    s ← s + w
end for
return s
Find α such that TOTAL(α) ≈ B
    Grow or shrink α until TOTAL(α) ≤ B
for all class c do
    k_c ← clip(⌊α√n_c⌉, m_min, u_c)
end for
Greedy rounding adjustment: if Σ_c k_c > B decrement
largest k_c, else increment smallest
return {k_c}
```

Algorithm 3 ReConText3D SELECTKCENTER

Inputs: Per-class assets $\{(c, \{(a_i, \mathcal{T}_i)\})\}$, allocations $\{k_c\}$, CLIP text encoder $E(\cdot)$, max captions per asset M
Outputs: Replay set $\mathcal{R}$ of selected base assets

```
EmbedAsset a, T
Select first M valid captions t_j ∈ T
z_j ← E(t_j) / ‖E(t_j)‖_2,    z̄ ← (1/M') Σ_j z_j
v(a) ← z̄ / ‖z̄‖_2                    ▷ Average caption embeddings
return v(a)
for all class c do
    V ← {v(a_i)}_{i=1}^{N_c},    k ← k_c
    if k ≥ N_c then
        R_c ← {a_i}; continue
    end if
    μ ← (Σ_i v_i) / ‖Σ_i v_i‖_2          ▷ Class mean in text space
    s ← arg max_i ⟨v_i, μ⟩,    S ← {s}
    d_min(i) ← 1 − ⟨v_i, v_s⟩
    for t = 2 to k do
        u ← arg max_i d_min(i)
        S ← S ∪ {u}
        d_new(i) ← 1 − ⟨v_i, v_u⟩
        d_min(i) ← min(d_min(i), d_new(i))
    end for
    R_c ← {a_i : i ∈ S}
end for
R ← ⋃_c R_c
return R
```

Preprocessing and Class Filtering. The Toys4K captioned dataset [45] contains 3,180 captioned 3D assets (meshes) belonging to 109 natural object classes. To ensure sufficient samples per class for both training and testing, we remove classes containing less than 15 assets, retaining the top 90 classes. Similar to other real-world 3D datasets, our **Toys4K-CL** dataset exhibits a long-tailed distribution.

Base and Novel Class Splits. Following the notation introduced in Sec. 3.1, we divide the dataset into disjoint *base* and *novel* class sets, $\mathcal{C}_{\text{base}}$ and $\mathcal{C}_{\text{novel}}$, each containing **45** classes. To construct these splits, we adopt a balanced and semantically diverse partitioning strategy guided by the following goals: (i) maintain a comparable number of classes and total assets across stages; (ii) preserve semantic diversity within each stage, ensuring exposure to varied structures, shapes, and materials; (iii) provide sufficient training and test samples for each class; and (iv) create *adversarial splits* that distribute semantically similar categories (e.g., *dog*, *cat*, *fox*) across the two stages to increase interference and challenge continual learning methods.

To design these splits, we prompted GPT4o [11] to propose balanced partitions from the full class distribution under the above constraints. This results in two 45-class splits, Base and Novel, with near-uniform class counts and complementary semantic coverage. Each class contributes up to five assets for testing, with the remaining samples used for training. Our test split consists of **450** samples, with **225** samples for both base and novel classes. The training set has **1352** samples for base class assets and **1243** samples for novel class assets.

Benchmark Properties. Toys4K-CL provides a compact yet challenging continual generation benchmark featuring realistic long-tailed data statistics, class imbalance, and inter-class similarity—factors that collectively stress-test catastrophic forgetting and knowledge transfer in generative models. Detailed class lists and per-split statistics are provided in the Supplementary Sec. 8.

4. Experiments

4.1. Implementation Details

Backbones. We experiment with two representative text-to-3D generation backbones: *TRELLIS-XL* (flow-based model) and *Shap-E* (diffusion-based model), enabling analysis across archi- tectures rather than model variants. For both models, we only train their text-conditioned parts of the models. For TRELLIS, we train its text-conditioned generative models, the Sparse-Structure (SS) Flow model and Structured-Latent (SLAT) Flow model. The SS and SLAT representations required for training are generated us-

Table 1. **Quantitative comparision on the Toys4K-CL benchmark.** We report CLIP similarity (↑) and Fréchet Distance scores computed on Inception and PointNet++ features (↓). Forgetting (%) measures base-class degradation after novel training. Results are reported on TRELLIS-XL and Shap-E backbones. The best results are in bold, and the second-best are underlined.

Method	CLIP (↑)				FD_{Incep} (↓)				FD_{Point} (↓)			
	Base	Novel	All	F (%)	Base	Novel	All	F (%)	Base	Novel	All	F (%)
TRELLIS-XL (Flow-based)												
Base Training	29.60	–	–	-	73.04	–	–	-	75.22	–	–	-
Fine-tuning	24.46	29.79	27.12	17.36	101.00	75.33	56.66	38.18	72.01	69.05	70.13	–4.27
L2-SP	24.50	29.63	27.06	17.23	102.88	75.24	57.65	40.85	**68.47**	69.05	**68.33**	**–8.98**
Ours	28.41	**29.86**	**29.14**	4.02	87.28	75.34	52.42	19.50	70.20	70.28	69.78	–6.68
Ours + L2-SP	**28.44**	29.52	28.98	**3.92**	**84.85**	**74.56**	**51.59**	**16.17**	71.57	**68.83**	69.79	–4.86
Joint Training	29.57	29.45	29.51	0.10	78.56	75.54	50.20	7.56	74.34	68.87	71.05	–1.18
Shap-E (Diffusion-based)												
Base Training	28.75	–	–	-	107.11	–	–	-	25.39	–	–	-
Fine-tuning	27.80	**28.70**	28.25	3.30	118.28	108.99	85.43	10.43	25.46	22.05	23.37	0.28
L2-SP	27.38	28.27	27.83	4.77	123.98	112.87	89.92	15.75	25.76	22.25	23.58	1.49
Ours	**28.34**	28.54	**28.44**	**1.43**	**110.49**	111.77	**83.26**	**3.15**	**25.21**	**21.78**	**23.17**	**–0.67**
Ours + L2-SP	28.33	28.47	28.40	1.46	112.31	111.00	83.84	4.85	25.25	22.35	23.45	–0.55
Joint Training	28.45	28.53	28.49	1.04	112.23	110.16	83.33	4.77	25.65	22.28	23.63	1.05

ing the pretrained VAEs. We also use the pretrained mesh decoder to decode the generated SLATs into 3D meshes. Both VAEs and mesh decoder are already pre-trained on the TRELLIS-500K train dataset (Toys4K not included) and open-sourced [45]. For Shap-E, we use the pretrained SDF-VAE, provided by [20] to obtain latent volumes and train only its text-conditioned diffusion model.

Training setup. For TRELLIS-XL, the base model is trained from scratch for 360k and 150k steps for the SS and SLAT models, respectively, with a learning rate of $1e{-}4$. Continual-learning baselines are fine-tuned for 200k (SS) and 120k (SLAT) steps with a learning rate of $1e{-}5$. All the models are trained on 4 A100-80GB GPUs with a batch size of 8 and 24 for the SS and SLAT flow model, respectively. For Shap-E, the text-conditioned diffusion model is trained using the weights from a pre-trained checkpoint provided by [20]. Both the base model training and the continual learning baselines are trained for 1000 epochs with a learning rate of $1e{-}5$, and the best checkpoint is used for evaluation. All models are trained on a single H100 GPU with a batch size of 16. All other training hyperparameters are the same as mentioned in the original papers for both TRELLIS [45] and Shap-E [13]. All the models were trained on our **Toys4K-CL** training set. Originally, each asset has 11 captions (different levels of details). Similar to TRELLIS, we randomly select 1 out of 11 during training. For evaluation, the most descriptive caption is used.

Remarks. Unlike TRELLIS, we did not train the diffusion-based Shap-E generative model from scratch, as it failed to converge on the relatively small Toys4K-CL dataset. The flow-based TRELLIS generators use simpler objectives and two-stage generation (sparse→structured), whereas Shap-E's one-stage diffusion objective is considerably more complex, motivating the use of pretrained checkpoints.

4.2. Baselines

We compare **ReConText3D** against the following continual-learning and reference baselines:

- **Base Training:** model trained only on $\mathcal{D}_{\text{base}}$; serves as the initial representation and upper-bound performance for base classes.
- **Fine-tuning:** direct fine-tuning of the base model on $\mathcal{D}_{\text{novel}}$ without any constraint. Updates all parameters of the base model.
- **L2-SP** [46]: fine-tuning with an additional L2-SP regularization term applied to all learnable parameters except bias and normalization layers. Serves as the comparision with a known continual learning method.
- **ReConText3D (ours) + L2-SP:** combination of L2-SP with our ReConText3D replay strategy.
- **Joint Training:** model trained from scratch on $\mathcal{D}_{\text{base}} \cup \mathcal{D}_{\text{novel}}$; represents the upper performance bound.

4.3. Quantitative Results

Evaluation Metrics. For quantitative evaluation of the generated 3D assets, we use the metrics reported by TRELLIS [45]. We report CLIP score [29] to measure text-mesh alignment and Frechet Distance (FD), using Inception-

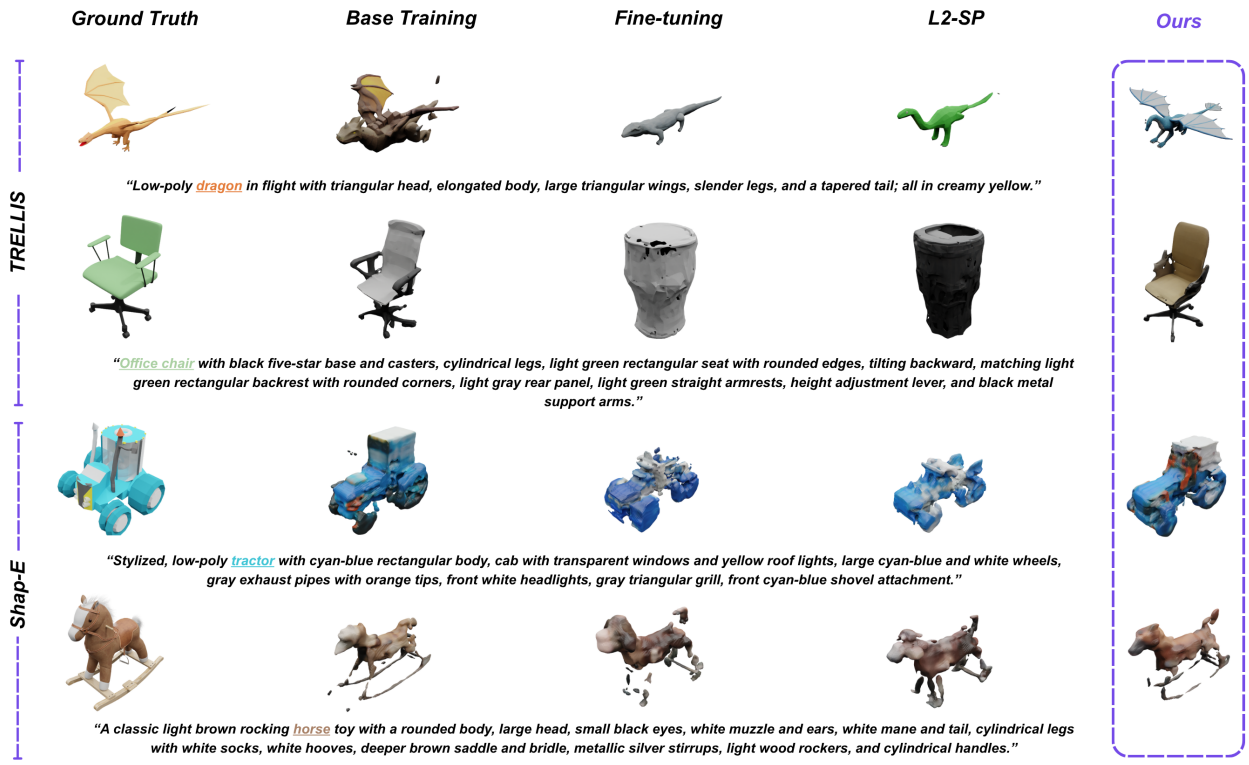

Figure 3. **Qualitative comparision of continual text-to-3D baselines on base class assets:** From left to right: Ground Truth, Base Training, Fine-tuning, L2-SP, and our method. Results on TRELLIS are shown on the top, while Shap-E is shown on the bottom.

v3 [39] and PointNet++ [28] feature embeddings, for appearance and geometric quality, respectively. Forgetting (F) is measured as the relative percentage drop in base-class performance after novel training.

Results on TRELLIS-XL and Shap-E. Table 1 presents the quantitative comparision on the test set of our **Toys4K-CL** benchmark for both TRELLIS-XL and Shap-E models. Naive fine-tuning exhibits clear *catastrophic forgetting*, with strong performance degradation on previously learned (Base) classes, i.e., a **17.4%** drop in CLIP score and **38.2%** forgetting in FD_{Incep} on TRELLIS-XL, and smaller but consistent losses of **3.3%** in CLIP score and **10.4%** in FD_{Incep} on Shap-E. In contrast, our proposed **ReConText3D** replay strategy markedly reduces forgetting and improves overall generative quality. On TRELLIS-XL, ReConText3D reduces the CLIP forgetting by ≈**77%** and FD_{Incep} forgetting by ≈**50%** compared to naive fine-tuning. Similarly, on Shap-E, it lowers the forgetting by ≈**56%** in CLIP score and ≈**70%** in FD_{Incep} forgetting compared to the fine-tuning baseline, approaching the joint-training upper bound.

Beyond mitigating forgetting, **ReConText3D** consistently enhances the overall performance across the base and novel classes, indicating that exemplar-based replay not only preserves past knowledge but also facilitates better generalization to new classes. We attribute this to a *semantic rehearsal effect*; by retaining a compact and diverse set of representative Base samples in text-embedding space, the model maintains alignment between the learned text-conditioned priors and its generative latent distribution, preventing drift in the shared conditioning space. This replayed supervision anchors the text-to-3D mapping and stabilizes training dynamics, especially under continual fine-tuning on limited novel-class data.

When combined with a mild **L2-SP** regularization term, the hybrid **ReConText3D + L2-SP** variant further reduces the forgetting, achieving the best scores for base classes on CLIP and FD_{Incep} metric for TRELLIS-XL.

Interestingly, we observe *negative forgetting* (i.e., positive backward transfer) on the geometric FD_{Point} metric across both models, suggesting that learning novel structural categories helps refine global 3D shape priors, improving geometry fidelity even for Base classes. Class-wise results are presented in the Supplementary Sec. 10.

4.4. Qualitative Results

Fig. 3 and Fig. 6 present qualitative comparisions of meshes generated from representative text prompts of the base and novel classes, respectively. Fine-tuning exhibits visible degradation on base classes, while L2-SP partially preserves geometry but fails to maintain texture fidelity. Fine-tuning fails to keep the semantic understanding of the base classes and confuses them with novel classes. In the first ex-

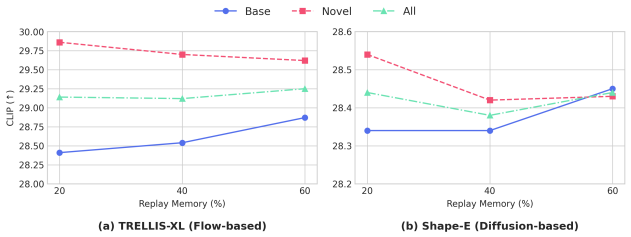

Figure 4. **Ablation on replay memory size.** Performance of our replay strategy under varying replay memory budgets/sizes on TRELLIS-XL and Shap-E backbones.

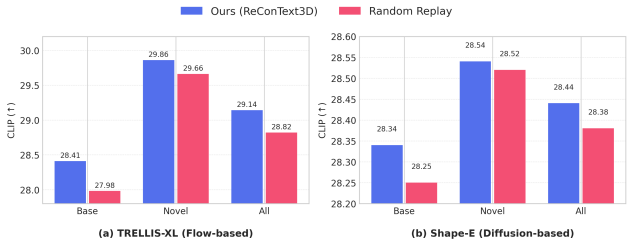

Figure 5. **Ablation on replay strategy.** comparision with random replay strategy on TRELLIS-XL and Shap-E backbones.

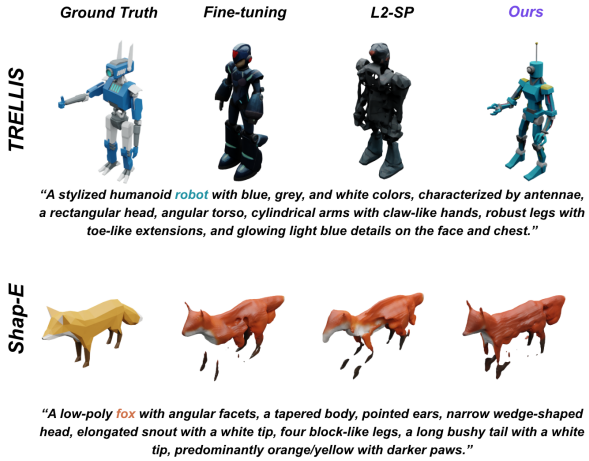

Figure 6. **Qualitative comparision** of novel class assets across baselines.

ample (row 1), it generates a lizard, which is a novel class, instead of the original dragon class from the base set. ReConText3D, on the contrary, retains semantic understanding and structural detail for old classes. Additionally, as seen in Fig. 6, ReConText3D successfully synthesizes plausible novel objects, with better quality compared to fine-tuning baseline, illustrating an effective balance between stability and plasticity in 3D generation. Extensive qualitative results are presented in the Supplementary Sec. 11.

4.5. Ablation Studies

We present ablation studies of our ReConText3D method in Fig 4 and Fig 5 to investigate the effects of: size of replay memory and type of replay strategy.

Replay Memory Size. We analyze the influence of the replay memory budget B in our method using TRELLIS-XL and Shap-E model. For this, we create three replay sets with $r \in \{20\%, 40\%, 60\%\}$ of the novel-set size N_{novel}. As shown in Fig. 4, CLIP performance on the base set improves steadily with larger replay buffers. Suggesting that a compact memory is sufficient for stable, continual generation.

Replay Type. To evaluate our k-Center selection, we compare it against a simple replay-based strategy, *Random Replay*, where replay samples are randomly selected from the entire base dataset. Results in Fig. 5 show that our text-embedding k-Center replay achieves the best overall balance between base retention and novel adaptation in terms of CLIP scores, demonstrating the importance of semantically diverse and representative replay construction in text-conditioned 3D generation.

5. Limitations

While **ReConText3D** shows strong performance and cross-architecture generalizability, our study is limited to a two-stage class-incremental setup with a fixed replay budget. Extending to multi-stage continual learning would require larger 3D datasets to maintain sufficient samples per class, particularly crucial for diffusion-based models like Shap-E, which demand substantial data for stable convergence. Broader object domains may also introduce challenges such as replay-memory saturation and semantic drift over longer training horizons. Moreover, current text-to-3D evaluations still rely on 2D-adapted metrics, future work should therefore explore perceptual and physical realism metrics defined directly in 3D space for more faithful assessment.

6. Conclusion

This work, for the first time, studies catastrophic forgetting in existing text-to-3D models under continual learning. We presented **ReConText3D**, the first framework for continual text-to-3D generation, enabling generative models to incrementally learn new 3D categories while retaining prior knowledge. Our replay-based strategy combines count-aware allocation with text-embedding k-center selection, achieving a strong balance between stability and plasticity without altering the base architecture. To support systematic evaluation, we introduced the **Toys4K-CL** benchmark with balanced and semantically diverse base–novel splits. Experiments across multiple backbones demonstrate that ReConText3D significantly mitigates catastrophic forgetting, confirming its model-agnostic effectiveness. We hope this work lays a foundation for future research on continual 3D generation.

7. Acknowledgment

This work was co-funded by the European Union under Horizon Europe, grant number 101135724, project LUMINOUS. However, the views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible.

References

[1] Sahar Ahmadi, Ali Cheraghian, Morteza Saberi, Md Towsif Abir, Hamidreza Dastmalchi, Farookh Hussain, and Shafin Rahman. Foundation model-powered 3d few-shot class incremental learning via training-free adaptor. In *Proceedings of the Asian Conference on Computer Vision*, pages 2282–2299, 2024. 3

[2] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022. 2

[3] Francisco Manuel Castro Payán, Manuel J Marín-Jiménez, Nicolás Guil-Mata, Cordelia Schmid, Karteek Alahari, et al. End-to-end incremental learning. 2018. 3

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. 2, 3

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22246–22256, 2023. 2

[6] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26576–26586, 2025. 2

[7] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. 3

[8] Jiahua Dong, Yang Cong, Gan Sun, Bingtao Ma, and Lichen Wang. I3dol: Incremental 3d object learning without catastrophic forgetting. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6066–6074, 2021. 3

[9] Jiahua Dong, Wenqi Liang, Hongliu Li, Duzhen Zhang, Meng Cao, Henghui Ding, Salman H Khan, and Fahad Shahbaz Khan. How to continually adapt text-to-image diffusion models for flexible customization? *Advances in Neural Information Processing Systems*, 37:130057–130083, 2024. 3

[10] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 831–839, 2019. 3

[11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 5

[12] Chenhan Jiang. A survey on text-to-3d contents generation in the wild. *arXiv preprint arXiv:2405.09431*, 2024. 2

[13] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 4, 6, 1

[14] Chenghao Li, Chaoning Zhang, Joseph Cho, Atish Waghwase, Lik-Hang Lee, Francois Rameau, Yang Yang, Sung-Ho Bae, and Choong Seon Hong. Generative ai meets 3d: A survey on text-to-3d in aigc era. *arXiv preprint arXiv:2305.06131*, 2023. 2

[15] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 3

[16] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6517–6526, 2024. 2

[17] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 300–309, 2023. 2

[18] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[19] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2

[20] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*, 36: 75307–75337, 2023. 6

[21] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 3

[22] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–82, 2018. 3

[23] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 2

[24] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11321–11329, 2019. 3

[25] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with

neural networks: A review. *Neural networks*, 113:54–71, 2019. 2

[26] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[27] Chao Qi, Jianqin Yin, Meng Chen, Yingchun Niu, and Yuan Sun. Boosting the class-incremental learning in 3d point clouds via zero-collection-cost basic shape pre-training. *arXiv preprint arXiv:2504.08412*, 2025. 3

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2, 7

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2, 4, 6, 1

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 3

[31] Hossein Resani, Behrooz Nasihatkon, and Mohammadreza Alimoradi Jazi. Continual learning in 3d point clouds: Employing spectral techniques for exemplar selection. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2921–2931. IEEE, 2025. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[33] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[34] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[35] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 3

[36] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023. 3

[37] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021. 2, 4, 1

[38] Gan Sun, Wenqi Liang, Jiahua Dong, Jun Li, Zhengming Ding, and Yang Cong. Create your world: Lifelong text-to-image diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):6454–6470, 2024. 3

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 7

[40] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[41] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22819–22829, 2023. 2

[42] Anh Thai, Stefan Stojanov, Zixuan Huang, and James M Rehg. The surprising positive knowledge transfer in continual 3d object shape reconstruction. In *2022 International Conference on 3D Vision (3DV)*, pages 209–218. IEEE, 2022. 3

[43] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36: 8406–8441, 2023. 2

[44] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *Advances in neural information processing systems*, 31, 2018. 3

[45] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 4, 5, 6, 1

[46] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International conference on machine learning*, pages 2825–2834. PMLR, 2018. 6, 1

[47] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 2, 3

[48] Na Zhao and Gim Hee Lee. Static-dynamic co-teaching for class-incremental 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3436–3445, 2022. 2, 3

[49] Na Zhao, Peisheng Qian, Fang Wu, Xun Xu, Xulei Yang, and Gim Hee Lee. Sdcot++: Improved static-dynamic co-teaching for class-incremental 3d object detection. *IEEE Transactions on Image Processing*, 2024. 2

[50] Ziyuan Zhao, Mingxi Xu, Peisheng Qian, Ramanpreet Singh Pahwa, and Richard Chang. Da-cil: Towards domain adaptive class-incremental 3d object detection. *arXiv preprint arXiv:2212.02057*, 2022. 3