Objaverse-XL: A Universe of 10M+ 3D Objects

Matt Deitke[†ψ], **Ruoshi Liu**[γ], **Matthew Wallingford**[ψ], **Huong Ngo**[ψ], **Oscar Michel**[†], **Aditya Kusupati**[ψ], **Alan Fan**[ψ], **Christian Laforte**[σ], **Vikram Voleti**[σ], **Samir Yitzhak Gadre**[γ], **Eli VanderBilt**[†], **Aniruddha Kembhavi**[†ψ], **Carl Vondrick**[γ], **Georgia Gkioxari**[δ], **Kiana Ehsani**[†], [*]**Ludwig Schmidt**[†ψℓ], [*]**Ali Farhadi**[ψ]

[†]Allen Institute for AI [ψ]University of Washington, Seattle [γ]Columbia University
[σ]Stability AI [δ]California Institute of Technology [ℓ]LAION
[*]Equal Senior Contribution

Figure 1: Objaverse-XL includes a ginormous collection of diverse 3D objects from a variety of sources. Here, we show examples of objects in Objaverse-XL rendered in a scene.

Abstract

Natural language processing and 2D vision models have attained remarkable proficiency on many tasks primarily by escalating the scale of training data. However, 3D vision tasks have not seen the same progress, in part due to the challenges of acquiring high-quality 3D data. In this work, we present Objaverse-XL, a dataset of over 10 million 3D objects. Our dataset comprises deduplicated 3D objects from a diverse set of sources, including manually designed objects, photogrammetry scans of landmarks and everyday items, and professional scans of historic and antique artifacts. Representing the largest scale and diversity in the realm of 3D datasets, Objaverse-XL enables significant new possibilities for 3D vision. Our experiments demonstrate the improvements enabled with the scale provided by Objaverse-XL. We show that by training Zero123 on novel view synthesis, utilizing over 100 million multi-view rendered images, we achieve strong zero-shot generalization abilities. We hope that releasing Objaverse-XL will enable further innovations in the field of 3D vision at scale.

Preprint. Under review.

1 Introduction

Scale has been paramount to recent advances in AI. Large models have produced breakthroughs in language comprehension and generation [7, 43], representation learning [50], multimodal task completion [3, 37], image generation [51, 54], and more. With an increasing number of learnable parameters, modern neural networks consume increasingly large volumes of data. As data has scaled up, the capabilities exhibited by models has dramatically increased.

Just a few years ago, GPT-2 [49] broke data barriers by consuming roughly 30 billion language tokens and demonstrated promising zero shot results on NLP benchmarks. Now, models like Chinchilla [25] and LLaMA [57] consume trillions of web crawled tokens and easily surpass GPT-2 at benchmarks and capabilities. In computer vision, ImageNet [15], with 1 million images, was the gold standard for representation learning until scaling to billions of images, via web crawled datasets like LAION-5B [55], produced powerful visual representations like CLIP [50]. Key to scaling up from millions of data points to billions and beyond has been the shift from assembling datasets manually to assembling them from diverse sources via the web.

As language and image data has scaled up, applications that require other forms of data have been left behind. Notable are applications in 3D computer vision, with tasks like 3D object generation and reconstruction, continue to consume small handcrafted datasets. 3D datasets such as ShapeNet [9] rely on professional 3D designers using expensive software to create assets, making the process tremendously difficult to crowdsource and scale. The resulting data scarcity has become a bottleneck for learning-driven methods in 3D computer vision. For instance, 3D object generation currently lags far behind 2D image generation, and current 3D generation approaches often still leverage models trained on large 2D datasets instead of being trained on 3D data from scratch. As demand and interest in AR and VR technologies goes up, scaling up 3D data is going to be increasingly crucial.

We introduce Objaverse-XL, a large-scale, web-crawled dataset of 3D assets. Advances in 3D authoring tools, demand, and photogrammetry, have substantially increased the amount of 3D data on the Internet. This data is spread across numerous locations including software hosting services like Github, specialized sites for 3D assets like Sketchfab, 3D printing asset sources like Thingiverse, 3D scanning platforms like Polycam, and specialized sites like the Smithsonian Institute. Objaverse-XL crawls such sources for 3D objects, providing a significantly richer variety and quality of 3D data than previously available, see Figure 1. Overall, Objaverse-XL comprises of over 10 million 3D objects, representing an order of magnitude more data than the recently proposed Objaverse 1.0 [14] and is two orders of magnitude larger than ShapeNet.

The scale and diversity of assets in Objaverse-XL significantly expands the performance of state-of-the-art 3D models. The recently proposed Zero123 [36] model for novel view synthesis, when pre-trained with Objaverse-XL, shows significantly better zero-shot generalization to challenging and complex modalities including photorealistic assets, cartoons, drawings and sketches. Similar improvements are also seen with PixelNerf which is trained to synthesize novel views given a small set of images. On each of these tasks, scaling pre-training data continues to show improvements from a thousand assets all the way up to 10 million, with few signs of slowing down, showing the promise and opportunities enabled with web scale data.

2 Related Work

Pre-training Datasets. Massive datasets have a prevalent role in modern, data-driven AI as they have produced powerful and general representations when paired with large-scale training. In computer vision, ImageNet [15], introduced nearly 14 years ago, has become the standard pre-training dataset of state-of-the-art visual models in object detection [53, 8], instance segmentation [24, 11] and more. More recently, large image datasets, such as LAION-5B [55], have powered exciting advances in generative AI, such as Stable Diffusion [54], and have given rise to new general-purpose vision and language representations with models like CLIP [50] and Flamingo [3]. This year, SAM [31]

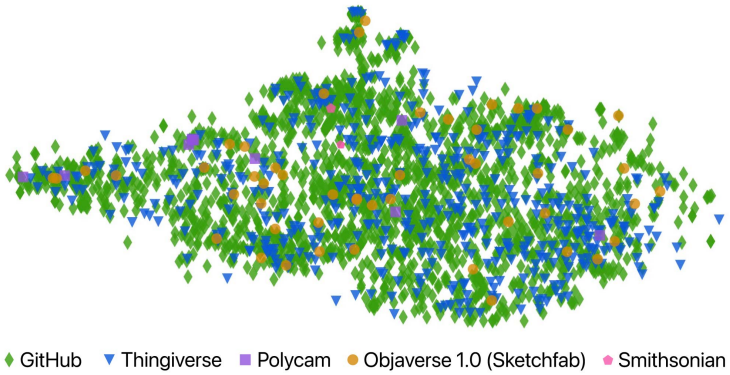


Figure 2: t-SNE projection of CLIP L/14 embeddings on a subset of rendered objects. Compared to Objaverse 1.0 (orange), Objaverse-XL more densely captures the distribution of 3D assets.

Source	# Objects
IKEA [32]	219
GSO [17]	1K
EGAD [41]	2K
OmniObject3D [63]	6K
PhotoShape [46]	5K
ABO [13]	8K
Thingi10K [67]	10K
3d-Future [19]	10K
ShapeNet [9]	51K
Objaverse 1.0 [14]	800K
Objaverse-XL	**10.2M**

Table 1: Number of 3D models in common datasets. Objaverse-XL is over an order of magnitude larger than prior datasets.

introduced a dataset of one billion object masks used to train a model capable of segmenting any object from an image. In language understanding, datasets like Common Crawl [1] have culminated in unprecedented capabilities of large language models such as GPT-4 [43], which in turn power mainstream applications like ChatGPT. The impact of large datasets is undeniable. However, current efforts to collect massive datasets focus on image and language modalities. In this work we introduce and release publically a massive dataset of 3D objects, called Objaverse-XL. Given the promise of large datasets for 2D vision and language, we believe Objaverse-XL will accelerate research in large-scale training for 3D understanding.

3D Datasets. Existing 3D datasets have been instrumental in yielding significant findings in 3D over the years. ShapeNet [9] has served as the testbed for modeling, representing and predicting 3D shapes in the era of deep learning. ShapeNet provides a collection of 3D shapes, in the form of textured CAD models labeled with semantic categories from WordNet [40]. In theory, it contains 3M CAD models with textures. In practice, a small subset of 51K models is used after filtering by mesh and texture quality. Notwithstanding its impact, ShapeNet objects are of low resolution and textures are often overly simplistic. Other datasets such as ABO [13], GSO [17], and OmniObjects3D [63] improve on the texture quality of their CAD models but are significantly smaller in size with the largest constituting 15K CAD models. Recently, Objaverse 1.0 [14] introduced a 3D dataset of 800K 3D models with high quality and diverse textures, geometry and object types, making it 15$\times$ larger than prior 3D datasets. While impressive and a step toward a large-scale 3D dataset, Objaverse 1.0 remains several magnitudes smaller than dominant datasets in vision and language. As seen in Figure 2 and Table 1, Objaverse-XL extends Objaverse 1.0 to an even larger 3D dataset of 10.2M unique objects from a diverse set of sources, object shapes, and categories. We discuss Objaverse-XL and its properties in Section 3.

3D Applications. The potential of a massive 3D dataset like Objaverse-XL promises exciting novel applications in computer vision, graphics, augmented reality and generative AI. Reconstructing 3D objects from images is a longstanding problem in computer vision and graphics. Here, several methods explore novel representations [12, 59, 38, 39], network architectures [22, 64] and differentiable rendering techniques [30, 10, 52, 34, 35] to predict the 3D shapes and textures of objects from images with or without 3D supervision. All of the aforementioned projects experiment on the small scale ShapeNet. The significantly larger Objaverse-XL could pave the way to new levels of performance, and increase generalization to new domains in a zero-shot fashion. Over the past year, generative AI has made its foray into 3D. MCC [62] learns a generalizable representation with self-supervised learning for 3D reconstruction. DreamFusion [48] and later on Magic3D [33] demonstrated that 3D shapes could be generated from language prompts with the help of text-to-image models. Point-E [42] and Shape-E [28] also train for text-to-3D with the help of 3D models from an undisclosed source. Recently, Zero123 [36] introduced an image-conditioned diffusion model which generates novel object views and is trained on Objaverse 1.0. Stable Dreamfusion [56] replaces the text-to-image model in DreamFusion with the 3D-informed Zero123 and shows improved 3D generations. Recent findings in AI and scaling laws [29, 25] suggest that both generative and predictive models benefit

Figure 3: **Examples of 3D objects from various sources of Objaverse-XL** spanning GitHub, Thingiverse, Polycam, the Smithsonian Institution, and Sketchfab. Objects from Thingiverse do not include color information, so each object's primary color is randomized during rendering.

from larger models and larger pre-training datasets. For 3D, Objaverse-XL is by far the largest 3D dataset to date and has the potential to facilitate large-scale training for new applications in 3D.

3 Objaverse-XL

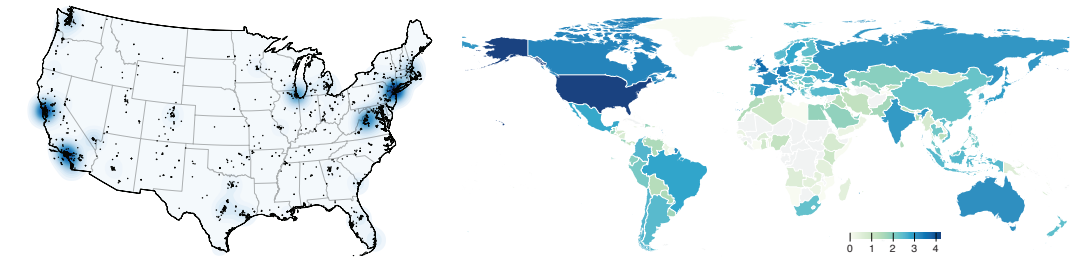
Objaverse-XL is a web scale 3D object dataset composed of a highly diverse set of 3D data sources on the internet. In this section, we discuss the sources, metadata of the objects, and provide an analysis of the objects.

3.1 Composition

Objaverse-XL is composed of 3D objects coming from several sources, including GitHub, Thingiverse, Sketchfab, Polycam, and the Smithsonian Institution. We detail each source below.

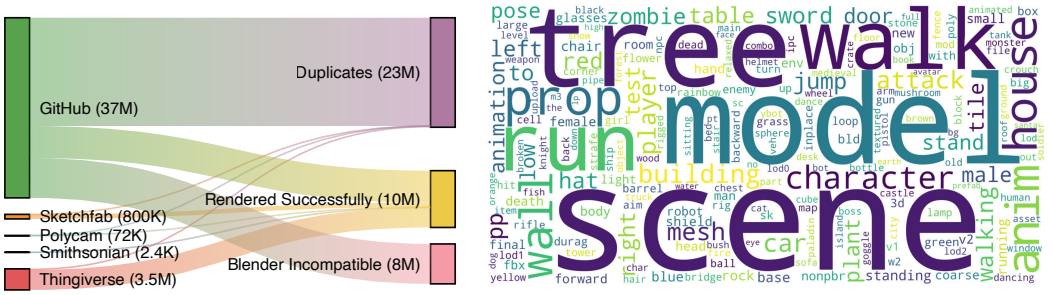
GitHub is a popular online platform for hosting code. We index 37M public files that contain common 3D object extensions; in particular, `.obj`, `.glb`, `.gltf`, `.usdz`, `.usd`, `.usda`, `.fbx`, `.stl`, `.dae`, `.ply`, `.abc`, and `.blend`. These extensions were chosen as they are best supported in Blender, which we use to render 2D images of the 3D objects. We only index objects that come from "base" GitHub repositories (*i.e.* non-forked repos, excluding forks that had more stars than the original repo). In total, the files come from over 500K repositories.

Across all of Objaverse-XL, objects are deduplicated by file content hash, which removes approximately 23 million files. Among the remaining files, we were able to import and successfully render



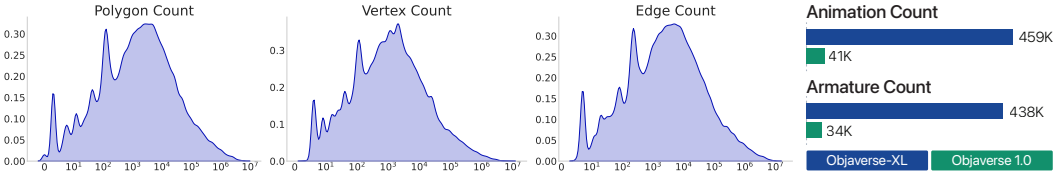
(a) Object location density in the United States

(b) Choropleth map of objects per country (log scale)



(c) Sankey diagram of Objaverse-XL sources

(d) Word cloud of file names for GitHub



(e) Statistics of the metadata extracted from the 3D objects.

Figure 4: **Analysis of metadata from Objaverse-XL.** Locations of geotagged objects in (a) the United States and (b) around the world. (c) Various sources and their contribution to Objaverse-XL. (d) Frequency of filenames of GitHub objects. (e) Further statistics of collected 3D objects.

5.5 million of those files. Files that were not successfully rendered were either caused by import compatibility issues (*i.e.* FBX ASCII files are not natively importable to Blender), no meshes are in the files, or the file is not a valid 3D file (*e.g.* an `.obj` file may be a C compiler file instead of a Wavefront Object file). Moving forward, we expect a solution for converting 3D file formats into a consolidated representation may yield several million more unique 3D objects.

Thingiverse is a platform for sharing objects most commonly used for 3D printing. We index and download around 3.5 million objects from the platform, which are predominantly released under Creative Commons licenses. The vast majority of the objects are STL files, which are often watertight meshes that are untextured, and serve as useful data for learning a shape prior. During rendering, we randomize the colors to broaden the distribution of the images.

Sketchfab is an online platform where users can publish and share 3D models, encompassing a broad variety of categories. The data sourced from Sketchfab for our project is specifically from Objaverse 1.0, a dataset of 800K objects consisting of Creative Commons-licensed 3D models. Each model is distributed as a standardized GLB file. The 3D models are freely usable and modifiable, covering an array of object types, from real-world 3D scans to intricate designs created in 3D software.

Polycam is a 3D scanning mobile application designed to facilitate the acquisition and sharing of 3D data. One of its salient features is the *explore* functionality, which enables members of the user community to contribute their 3D scans to a publicly accessible database. In the context of our dataset, we focus specifically on the subset of objects within the explore page that are designated as savable. These savable objects are governed by a Creative Commons Attribution 4.0 International License

(CC-BY 4.0). We indexed 72K objects that were marked as savable and licensed under a CC-BY 4.0 license. Following deduplication, we obtain 71K unique objects.

Smithsonian 3D Digitization is a project by the Smithsonian Institution dedicated to digitizing their vast collection of historical and cultural artifacts. The project has provided us with a set of 2.4K models, all licensed under a CC0 license, which signifies that these works are fully in the public domain and free for use without any restrictions. The objects in this collection are primarily scans of real-world artifacts. Each model is distributed in a standardized compressed GLB format.

3.2 Metadata

Each object comes with metadata from its source, and we also extract metadata from it in Blender and from its CLIP ViT-L/14 features. We describe the metadata acquisition process below.

Source Metadata. From the source, we often get metadata such as its popularity, license, and some textual description. For example, on GitHub, the popularity is represented by the stars of the object's repository and the file name serves as the object's textual pair.

Blender Metadata. For each object that we render, we obtain the following metadata for it: `sha256`, `file-size`, `polygon-count`, `vertex-count`, `edge-count`, `material-count`, `texture-count`, `object-count`, `animation-count`, `linked-files`, `scene-dimensions`, and `missing-textures`. During rendering, for objects that have a missing texture, we randomize the color of that texture. Figure 4 shows some charts extracted from the metadata, including density plots over the number of polygons, vertex counts, and edge counts.

Animated Objects. From the Blender metadata, we find that the number of animated objects and those with armature (a digital skeleton used to animate 3D models) significantly increases from Objaverse 1.0 to Objaverse-XL. Figure 4e (right) shows a bar chart of the increase, specifically from 41K to 459K animated objects and from 34K to 438K objects with armature.

Model Metadata. For each object, we extract its CLIP ViT-L/14 [50] image embedding by averaging the CLIP embedding from 12 different renders of the object at random camera positions inside of a hollow sphere. We use the CLIP embeddings to predict different metadata properties, including aesthetic scores, NSFW predictions, face detection, and for detecting holes in the photogrammetry renderings. Section C.3 provides more details on the analysis.

3.3 Analysis

NSFW annotations. Most data sources used for the creation of Objaverse-XL already have either a strict NSFW policy or strong self-filtering. However, owing to the web scale of Objaverse-XL we performed NSFW filtering using the rendered images of the objects. Each 3D object is rendered in 12 random views and each rendered image is passed through an NSFW classifier trained on the NSFW dataset introduced in LAION-5B [55] by Gadre et al. [20] using the CLIP ViT-L/14 [50] features. After careful analysis and manual inspection, we marked a rendered image as NSFW if it has an NSFW score above 0.9 and a 3D object is marked as NSFW if at least 3 rendered images are deemed to be NSFW. Overall, only 815 objects out of the 10M are filtered out as NSFW objects. Note that the high threshold and multi-view consistency are needed due to the distribution shift between LAION-5B and Objaverse-XL along with NSFW classification of certain viewpoint renders of harmless 3D objects.

Face detection. We analyze the presence of faces in Objaverse-XL using a detector trained by Gadre et al. [20]. Like NSFW filtering, we count the objects where at least 3 images contain a detected face. Out of 10M assets, we estimate 266K objects include faces. However, unlike most web-scale datasets, the faces present in Objaverse-XL often come from the scans of dolls, historical sculptures, and anthropomorphic animations. Hence, there are less privacy concerns with most of these objects.

Photogrammetry hole detection. When scanning 3D objects, if the back or bottom of the object is not scanned, rendering from various viewpoints may contain holes, leading to a "bad" render image. For example, a non-trivial number of Polycam 3D objects lack the information from the "back side".

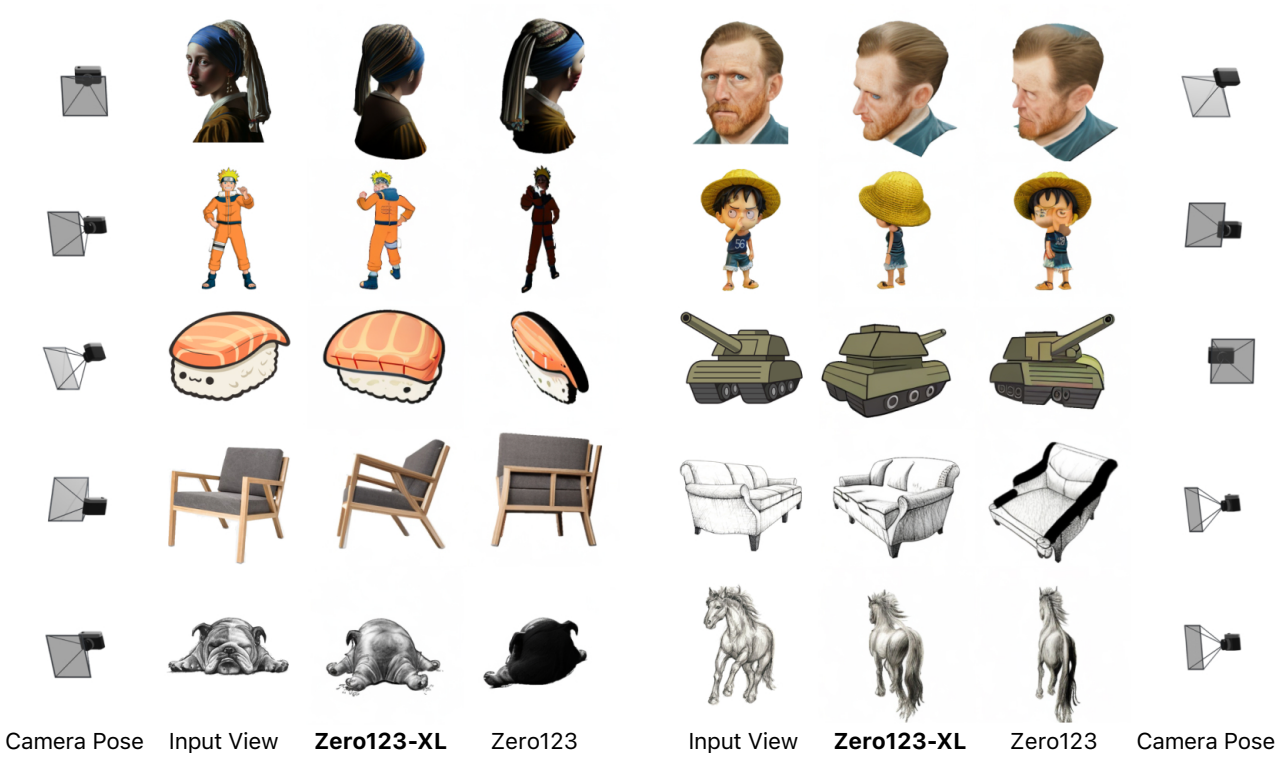


Figure 5: **Novel view synthesis on in-the-wild images.** Comparison between Zero123-XL trained on Objaverse-XL and Zero123 trained on Objaverse. Starting from the input view, the task is to generate an image of the object under a specific camera pose transformation. The camera poses are shown beside each example. Significant improvement can be found by training with more data, especially for categories including people (**1st row**), anime (**2nd row**), cartoon (**3rd row**), furniture (**4th row**), and sketches (**5th row**). Additionally, viewpoint control is significantly improved (see **2nd row**).

In most cases, images that are rendered from back-side viewpoints are noisy, low-fidelity, or contain holes. To analyze "bad rendering" at scale, we manually annotated 1.2K Polycam renders as "good" (label 1) or "bad" (label 0). We trained a "bad render" classifier (2-layer MLP) on top of the CLIP ViT-L/14 features of the rendered images; this classifier achieves a cross-validation accuracy of over 90% with a "render score" threshold of 0.5. Overall, out of 71K Polycam objects with 12 renders each, we found that 38.20% renders are "bad", with 58K objects having at least 2 bad renders.

4 Experiments

4.1 Novel View Synthesis with Zero123-XL

Generating 3D assets conditioned on in-the-wild 2D images has remained a challenging problem in computer vision. A crucial lesson learned from large language models is that pretraining on simple and easily scalable tasks, such as next word prediction, leads to consistently improved performance

Zero123-XL	PSNR (↑)	SSIM (↑)	LPIPS (↓)	FID (↓)
Base	18.225	0.877	0.088	0.070
w/ Alignment Finetuning	**19.876**	**0.888**	**0.075**	**0.056**

Table 2: **Effect of high-quality data finetuning on Zero123-XL.** When evaluated zero-shot on Google Scanned Objects [17], a model finetuned on a high-quality alignment subset of Objaverse-XL significantly outperforms the base model trained only on Objaverse-XL.

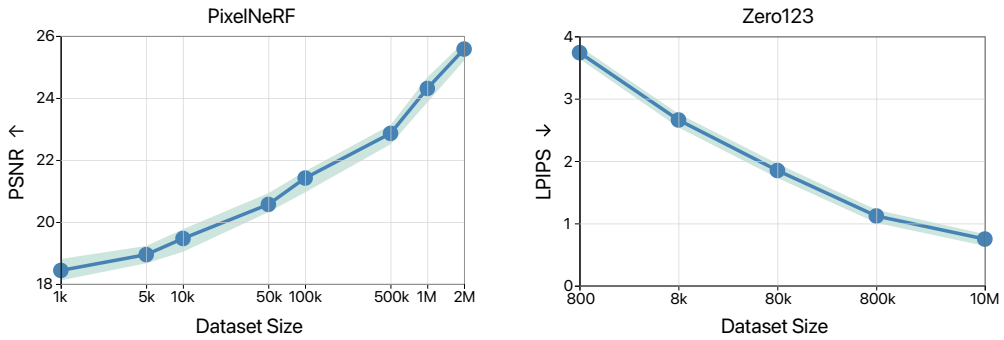


Figure 6: **Novel view synthesis at scale. Left**: PixelNeRF [64] trained on varying scales of data and evaluated on a held-out subset of Objavserse-XL. **Right**: Zero123 [36] trained on varying scales of data and evaluated on a zero-shot dataset. Note that the 800K datapoint is Zero123 and the 10M datapoint is Zero123-XL. The synthesis quality consistently improves with scale. LPIPS is scaled up 10 times for visualization.

and the emergence of zero-shot abilities. An analogous approach in 3D vision is to predict a novel view of an object from an input view. Zero123 [36] recently proposed a view-conditioned diffusion model to perform this task, where the weights of the diffusion model are initialized from Stable Diffusion to leverage its powerful zero-shot image generation abilities. Zero123 used objects in Objaverse 1.0 to render input and novel view pairs as the training dataset. We use this framework to create *Zero123-XL*, which is the same approach except trained on the much larger Objaverse-XL instead. As shown in [36], the pretrained view-conditioned diffusion model can also be plugged into a score distillation framework such as DreamFusion [48] or SJC [58] to obtain a 3D assets.

Zero-shot Generalization. We found that training Zero123 on Objaverse-XL achieves significantly better zero-shot generalization performance than using Objaverse 1.0. Figure 5 shows examples from categories of data commonly known to be challenging for baseline systems, including people, cartoons, paintings, and sketches. For example, in both of the examples shown in 2nd and 3rd rows of the first column, Zero123 interprets the input image as a 2D plane and performs a simple transformation similar to a homography transformation. In comparison, Zero123-XL is able to generate novel views that are more consistent with the input view. Additionally, Zero123-XL is able to generate novel views from sketches of objects while keeping the original style as well as object geometric details. These examples show the effectiveness of dataset scaling for zero-shot generalization in 3D.

Improvement with Scale. We further quantitatively evaluate the novel view synthesis performance on Google Scanned Objects dataset [17]. As shown in Figure 6, the rvisual similarity score [65] between the predicted novel view and the ground truth view continues to improve as the dataset size increases.

Alignment Finetuning. InstructGPT [44] shows that large-scale pretraining does not directly lead to a model aligned with human preferences. More recently, LIMA [66] shows that finetuning a pretrained model on a curated subset with high-quality data can achieve impressive alignment results. We adopted a similar approach here by selecting a high-quality subset of Objaverse-XL that contains 1.3 million objects. Selection is done by defining proxy estimation of human preference based on heuristics including vertex count, face count, popularity on the source website, and source of data, among other metrics. After pretraining the base model on the entire Objaverse-XL, we finetune Zero123-XL on the alignment subset with a reduced learning rate and performed an ablation study to evaluate the effect of alignment finetuning. Table 2 shows that alignment finetuning leads to significant improvement in zero-shot generalization performance. Please refer to Appendix A for more implementation details regarding our model and experiments.

4.2 Novel View Synthesis with PixelNeRF

Synthesizing objects and scenes from novel views is a long-standing challenge. Notably, neural radiance fields [39] have shown impressive capabilities in rendering specific scenes from novel views.

However, these methods require dozens of views of an individual scene, and can only synthesize views from the particular scene they were trained for. More recent methods [16, 27, 60, 64] have been proposed for constructing NeRF models that generalize across scenes with few input images. Due to the challenging nature of obtaining the necessary camera parameters for training, such methods have traditionally been trained on small scale data sets. With the Objaverse-XL data, we train a PixelNeRF model on over two million objects, magnitudes of more data than has previously been used. We find that PixelNeRF generalizes to novel scenes and objects significantly better and performance improves consistently with scale (Figure 6 and Table 3).

Improvement with Scale. We train PixelNeRF models conditioned on a single input image at varying scales of data (Figure 6) and evaluate on a held out set of Objaverse-XL objects. We find that novel view synthesis quality consistently improves with more objects even at the scale of 2 million objects and 24 million rendered images.

Generalization to Downstream Datasets. Similar to pretraining in 2D vision and language, we observe that pretraining on Objaverse-XL with PixelNeRF improves performance when fine-tuning to other datasets such as DTU [2] and ShapeNet [9] (Table 3). We pretrain and fine-tune the model conditioned on a single input view and report the peak signal-to-noise ratio (PSNR).

PixelNeRF	DTU [2]	ShapeNet [9]
Base	15.32	22.71
w/ Objaverse-XL	**17.53** $\pm$.37	**24.22** $\pm$.55

Table 3: **Comparison (PSNR (↑)) of PixelNeRF trained from scratch vs. fine-tuned from Objaverse-XL.** Performance significantly improves from pretraining on the large-scale corpus.

5 Limitations and Conclusion

Limitations. While Objaverse-XL is more than an order of magnitude larger than its predecessor, Objaverse 1.0, it is still orders of magnitude smaller than modern billion-scale image-text datasets. Future work may consider how to continue scaling 3D datasets and make 3D content easier to capture and create. Additionally, it may not be the case that all samples in Objaverse-XL are necessary to train high performance models. Future work may also consider how to choose datapoints to train on. Finally, while we focus on generative tasks, future work may consider how Objaverse-XL can benefit discriminative tasks such as 3D segmentation and detection.

Conclusion. We introduce Objaverse-XL, which is comprised of 10.2M 3D assets. In addition to documenting Objaverse-XL's unprecedented scale and sample diversity, we demonstrate the potential of Objaverse-XL for downstream applications. On the task of zero-shot novel view synthesis, we establish empirically promising trends of scaling dataset size, while keeping the model architecture constant. We hope Objaverse-XL will provide a foundation for future work in 3D.

Acknowledgements

We would like to thank Stability AI for compute used to train the experiments and LAION for their support. We would also like to thank Luca Weihs, Mitchell Wortsman, Romain Beaumont, and Vaishaal Shankar, Rose Hendrix, Adam Letts, Sami Kama, Andreas Blattmann, Kunal Pratap Singh, and Kuo-Hao Zeng for their helpful guidance and conversations with the project. Finally, we would like to thank the teams behind several open-source packages used throughout this project, including Blender [5], PyTorch [47], PyTorch Lightning [18], D3 [6], Matplotlib [26], NumPy [23], Pandas [45], Wandb [4], and Seaborn [61]. Following the NeurIPS guidelines, we would also like to acknowledge the use of LLMs for helping revise some text and general coding assistance. Finally, we would also like to thank and acknowledge the content creators who contributed to the dataset.

References

[1] URL https://commoncrawl.org/the-data/. 3

[2] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016. 9

[3] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[4] L. Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 10

[5] Blender Online Community. Blender - a 3d modelling and rendering package. https://www.blender.org, 2023. 10

[6] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 2011. 10

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[9] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3, 9

[10] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Advances in neural information processing systems*, 32, 2019. 3

[11] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 2

[12] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 628–644. Springer, 2016. 3

[13] J. Collins, S. Goel, K. Deng, A. Luthra, L. Xu, E. Gundogdu, X. Zhang, T. F. Y. Vicente, T. Dideriksen, H. Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21126–21136, 2022. 3

[14] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 2, 3

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[16] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 9

[17] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 3, 7, 8

[18] W. Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019. URL https://github.com/Lightning-AI/lightning. 10

[19] H. Fu, R. Jia, L. Gao, M. Gong, B. Zhao, S. Maybank, and D. Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021. 3

[20] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023. 6

[21] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. D. Iii, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021. 27

[22] G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019. 3

[23] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2. 10

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[25] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2, 3

[26] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9 (3):90–95, 2007. doi: 10.1109/MCSE.2007.55. 10

[27] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 9

[28] H. Jun and A. Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3

[29] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 3

[30] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018. 3

[31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 2

[32] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2992–2999, 2013. 3

[33] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. 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. 3

[34] R. Liu and C. Vondrick. Humans as light bulbs: 3d human reconstruction from thermal reflection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12531–12542, 2023. 3

[35] R. Liu, S. Menon, C. Mao, D. Park, S. Stent, and C. Vondrick. Shadows shed light on 3d objects. *arXiv preprint arXiv:2206.08990*, 2022. 3

[36] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2, 3, 8, 14

[37] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *ArXiv*, abs/2206.08916, 2022. 2

[38] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 3

[39] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 8

[40] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995. 3

[41] D. Morrison, P. Corke, and J. Leitner. Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation. *IEEE Robotics and Automation Letters*, 5(3): 4368–4375, 2020. 3

[42] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 3

[43] OpenAI. Gpt-4 technical report. *arXiv*, 2023. 2, 3

[44] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 8

[45] T. pandas development team. pandas-dev/pandas: Pandas, Feb. 2020. URL https://doi.org/10.5281/zenodo.3509134. 10

[46] K. Park, K. Rematas, A. Farhadi, and S. M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *arXiv preprint arXiv:1809.09761*, 2018. 3

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 10

[48] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 3, 8

[49] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[50] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 6

[51] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[52] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 3

[53] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[54] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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

[55] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 2, 6, 33

[56] J. Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 3

[57] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[58] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. 8

[59] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 3

[60] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 9

[61] M. L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60): 3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021. 10

[62] C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, and G. Gkioxari. Multiview compressive coding for 3d reconstruction. *arXiv preprint arXiv:2301.08247*, 2023. 3

[63] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[64] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3, 8, 9

[65] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[66] C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023. 8

[67] Q. Zhou and A. Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016. 3

A Implementation Details

A.1 Zero123-XL

A batch size of 2048 is used during training with a learning rate of 1e-4. Different from the original paper [36], we performed a second-stage finetuning with a smaller learning rate of 5e-5 on a high-quality subset of Objaverse-XL selected with dataset metadata. The first stage was trained for 375K iterations and the second stage is trained for 65K iterations. For dataset scaling experiment whose results are shown in 6, datasets with size below 800K are randomly sampled subsets from Objaverse 1.0. We keep the rest of the setting consistent with the original paper [36]. For calculating LPIPS metric in Figure 6, we multiple the score by 10 for better visualization.

B Additional Zero123-XL Comparisons

Figures 7-18 show additional comparisons between Zero123-XL and Zero123. Overall, Zero123-XL shows better generalization than Zero123 by both better following the camera transformation and generating more plausible outputs.

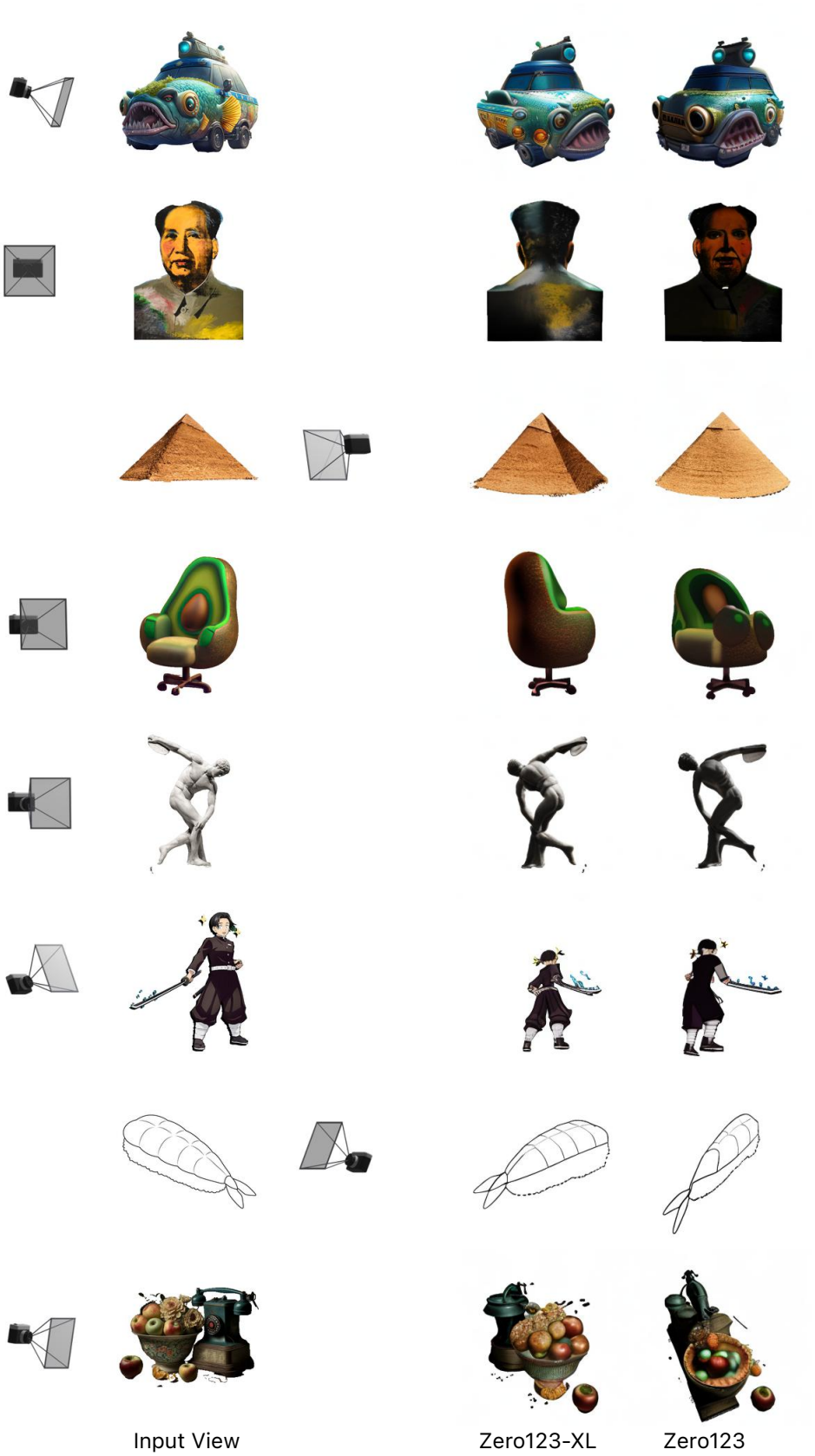


Figure 7: Additional examples comparing the outputs of Zero123-XL and Zero123 under different camera transformations.

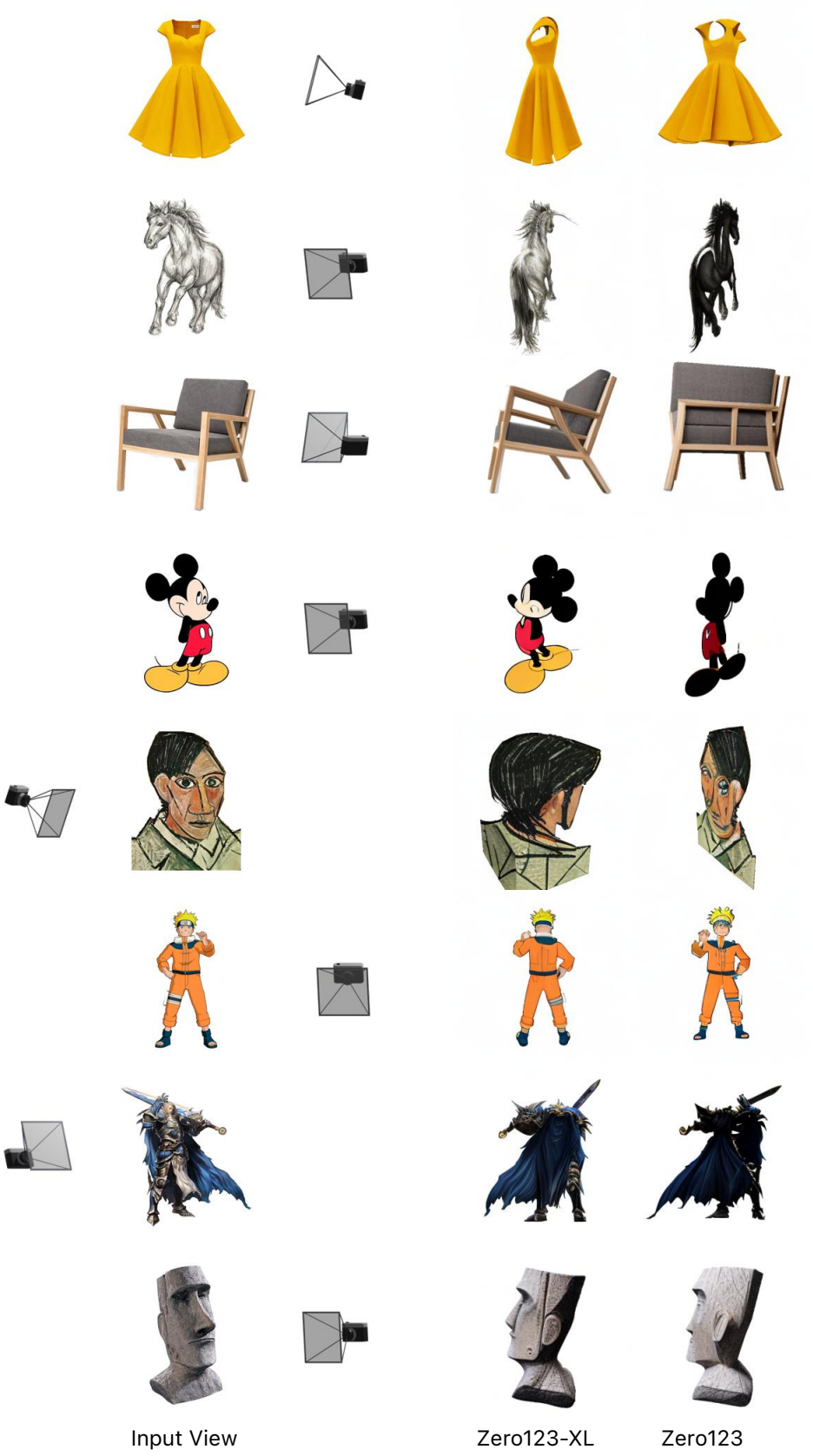


Figure 8: Continuation of additional examples comparing Zero123-XL and Zero123.

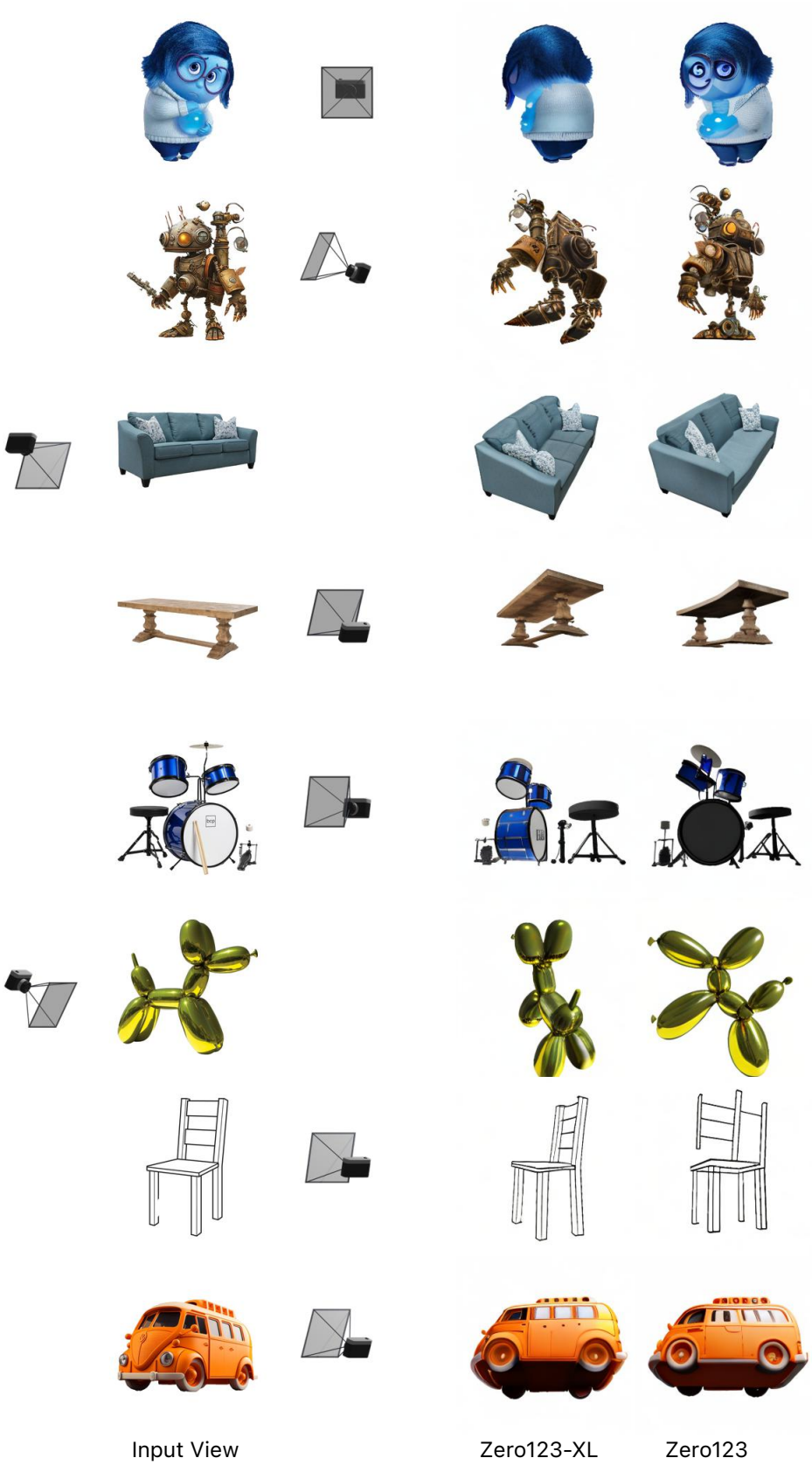


Figure 9: Continuation of additional examples comparing Zero123-XL and Zero123.

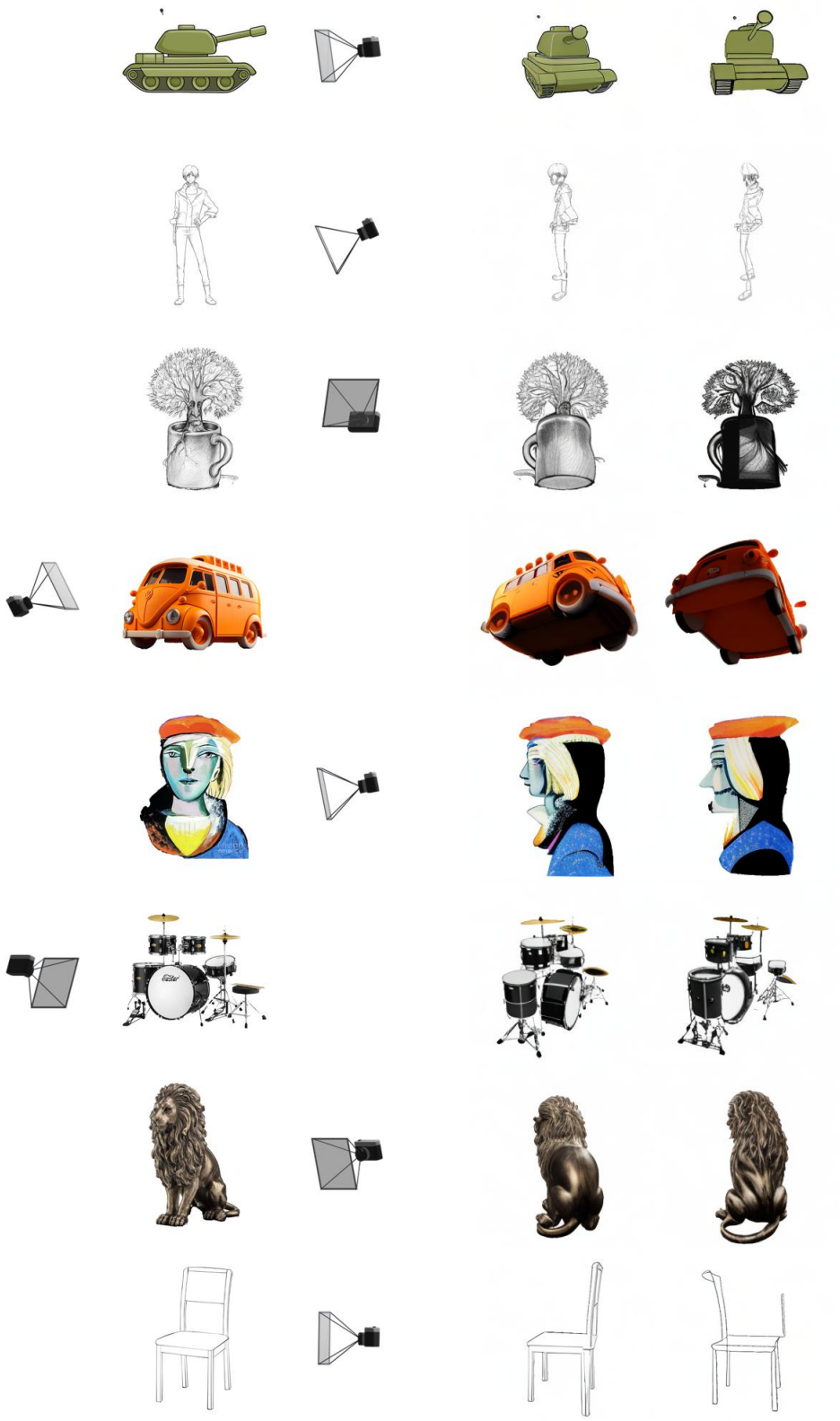


Figure 10: Continuation of additional examples comparing Zero123-XL and Zero123.

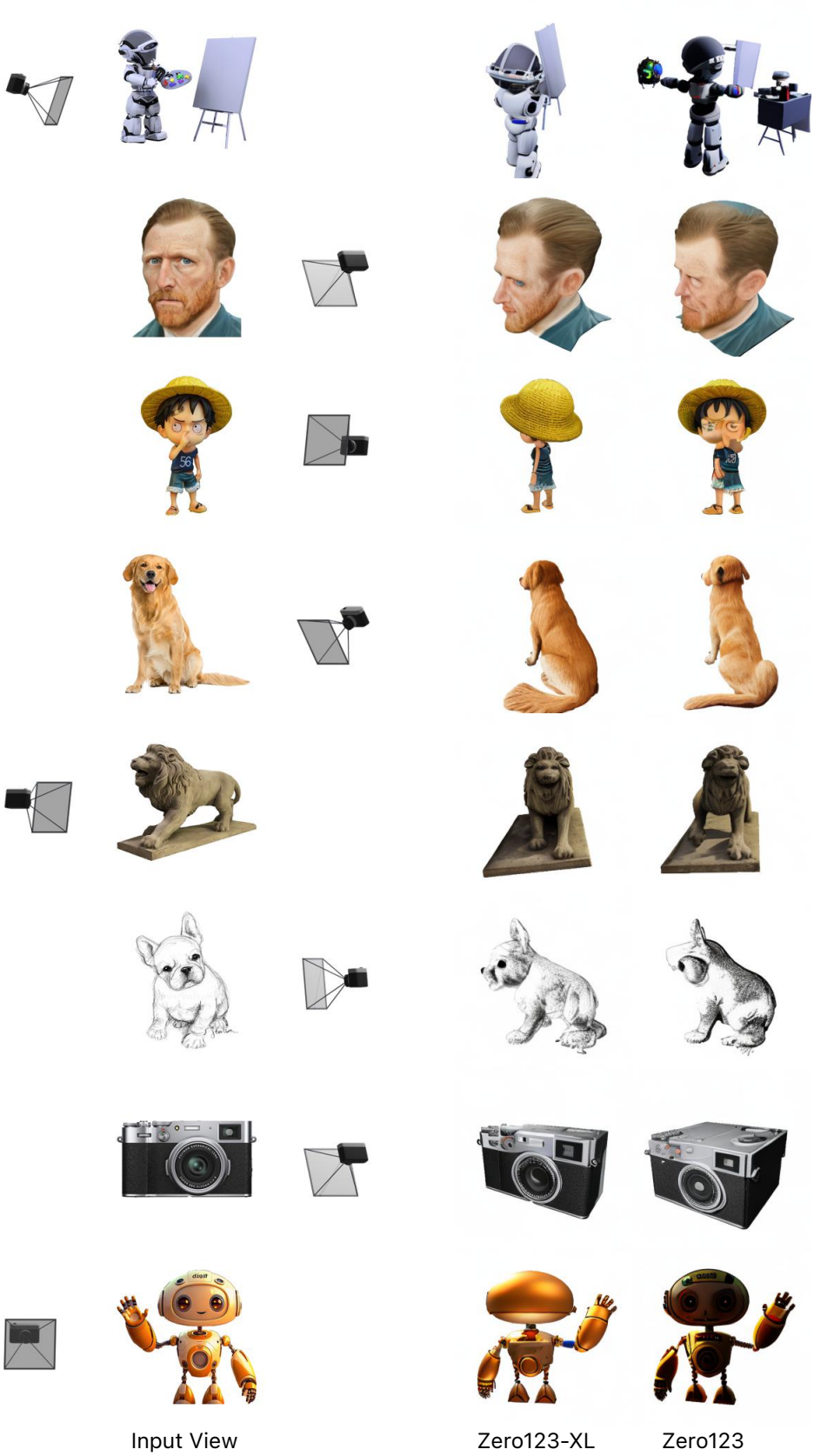


Figure 11: Continuation of additional examples comparing Zero123-XL and Zero123.

Figure 12: Continuation of additional examples comparing Zero123-XL and Zero123.

Figure 13: Continuation of additional examples comparing Zero123-XL and Zero123.

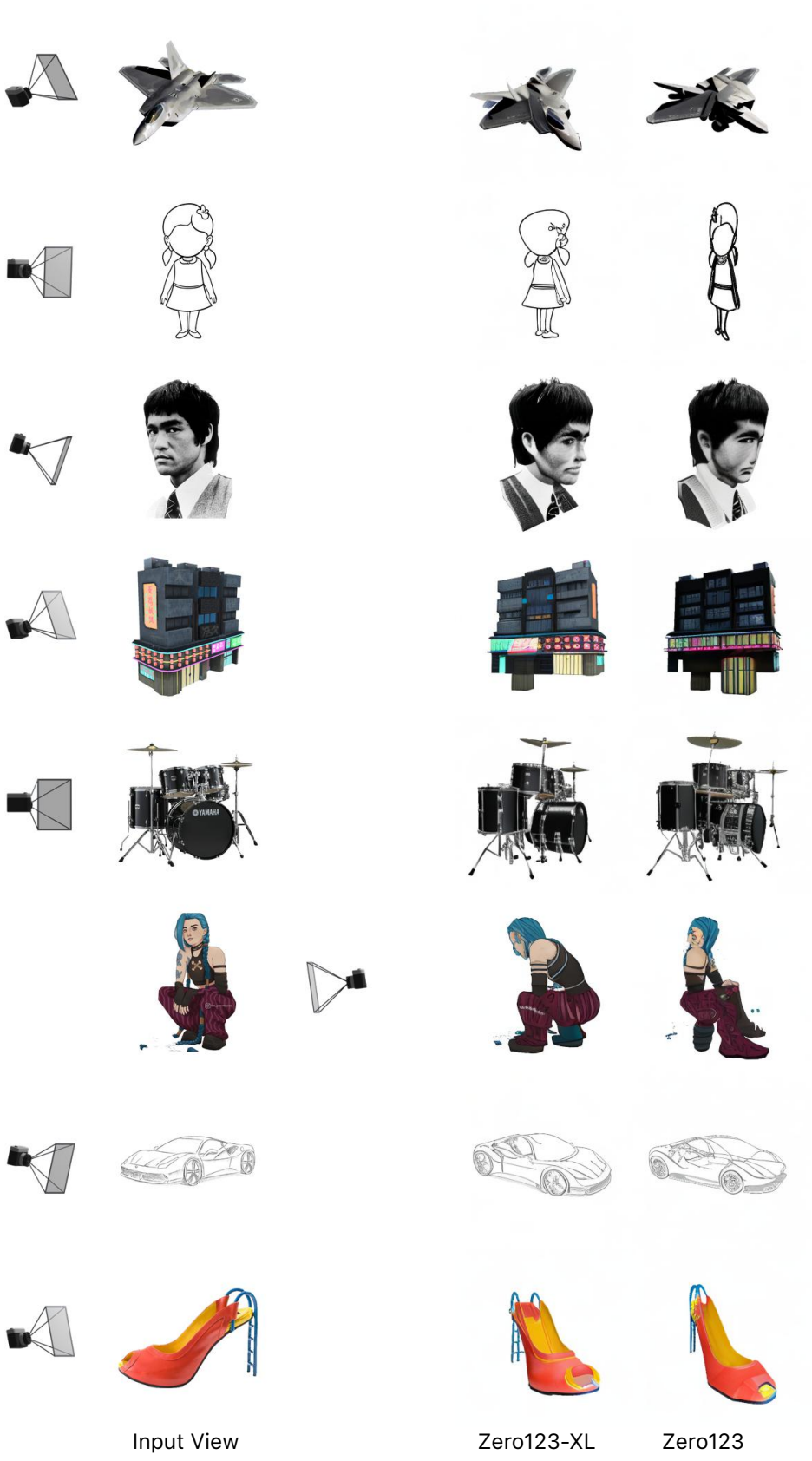


Figure 14: Continuation of additional examples comparing Zero123-XL and Zero123.

Figure 15: Continuation of additional examples comparing Zero123-XL and Zero123.

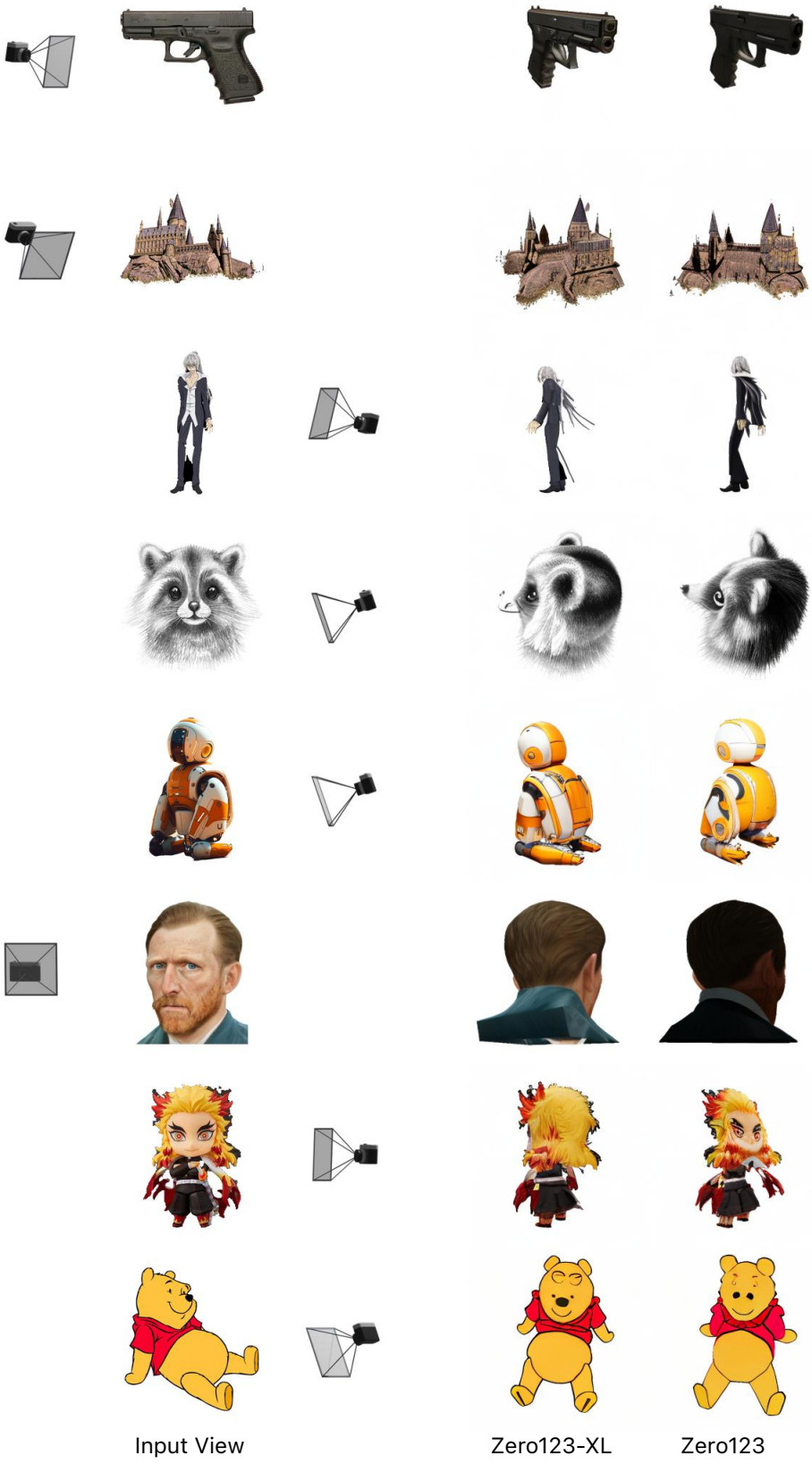


Figure 16: Continuation of additional examples comparing Zero123-XL and Zero123.

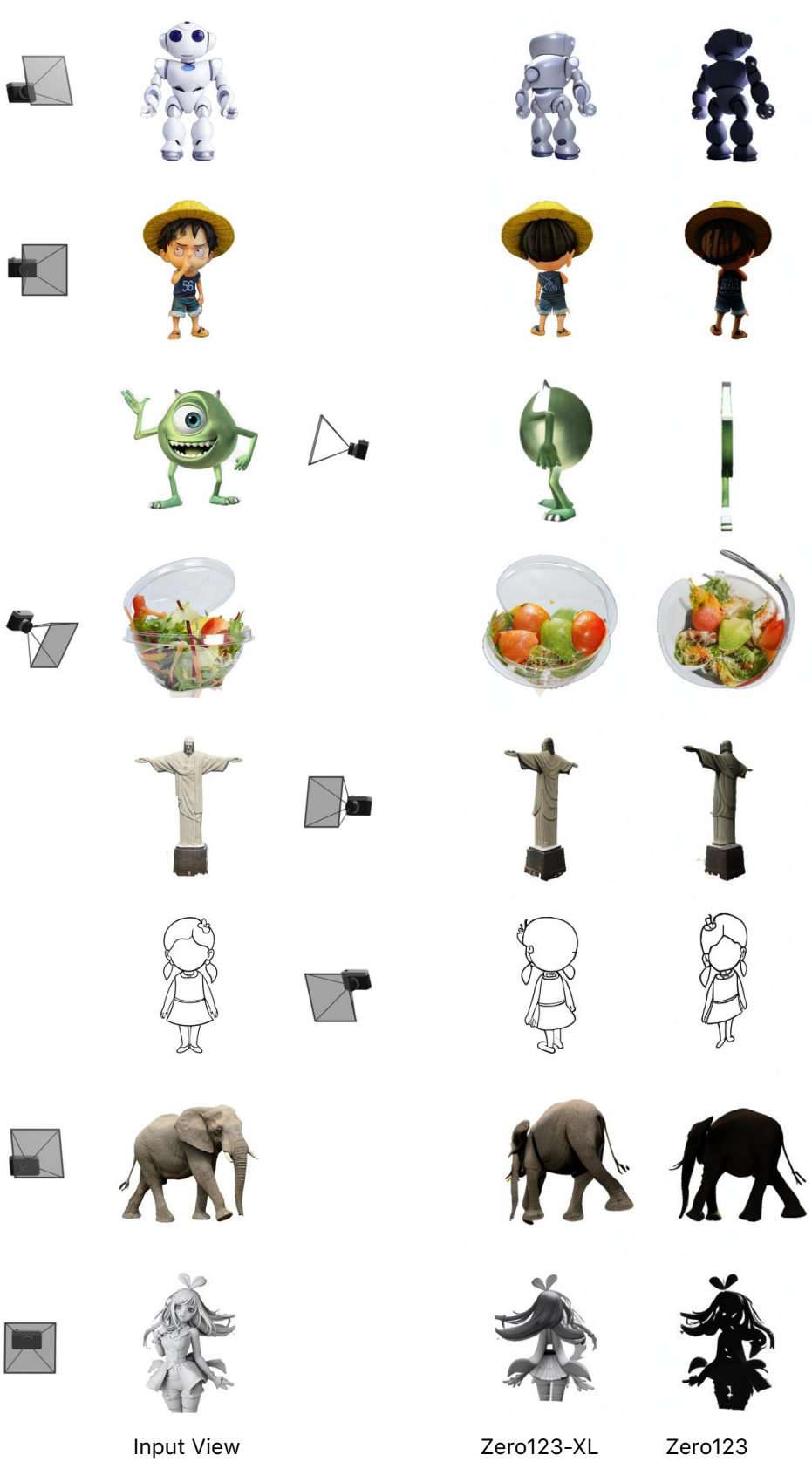


Figure 17: Continuation of additional examples comparing Zero123-XL and Zero123.

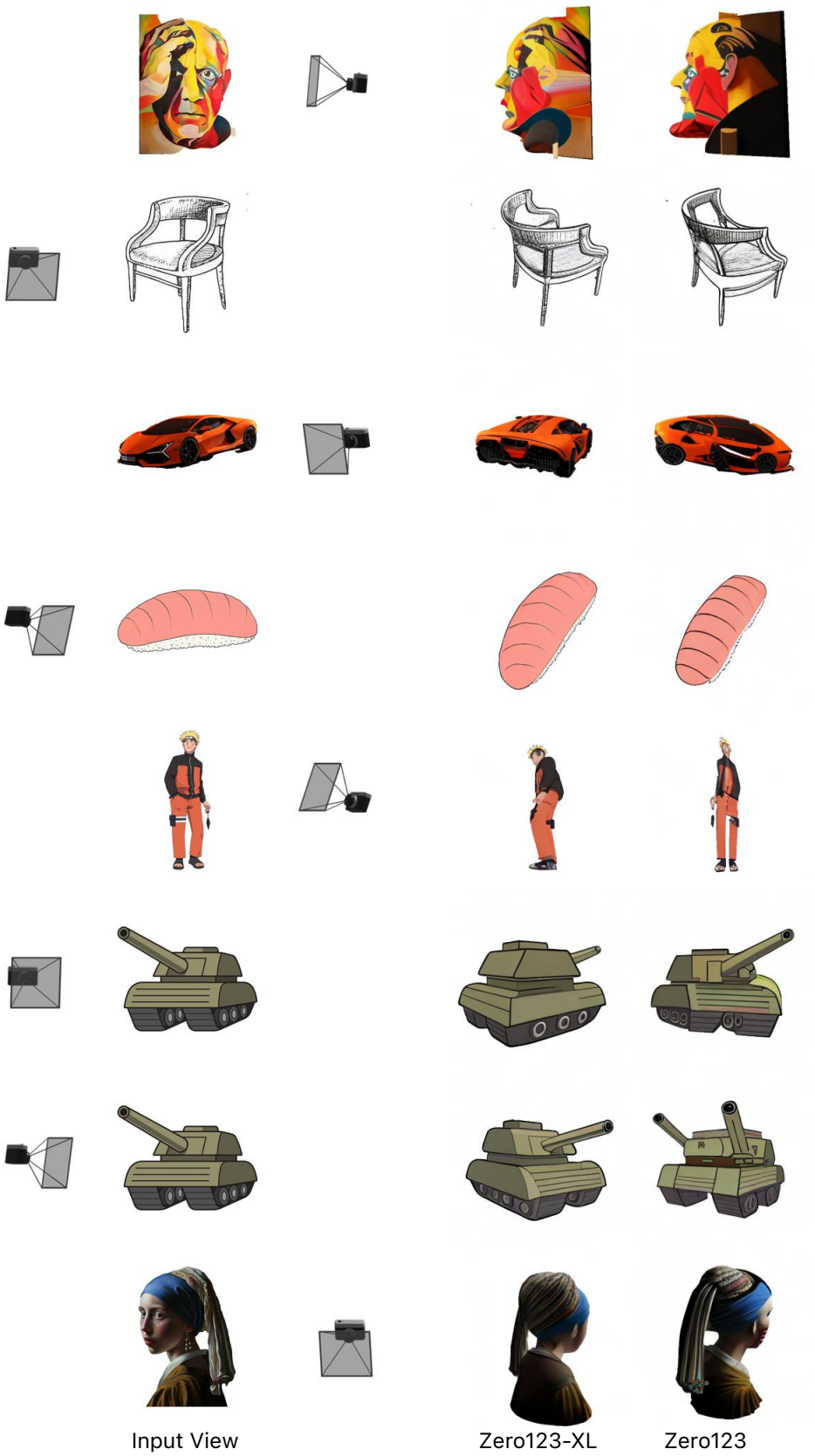


Figure 18: Continuation of additional examples comparing Zero123-XL and Zero123.

C Datasheet

This section provides a datasheet [21] for Objaverse-XL.

C.1 Motivation

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.
The Objaverse-XL dataset was created to address the lack of high-quality, large-scale datasets for 3D vision tasks. This was due to challenges in acquiring such data and the associated complexities of 3D object generation and reconstruction, which were largely reliant on smaller, handcrafted datasets. The dataset was designed with the aim of advancing the field of 3D vision, allowing for the development and generalization improvement of models like Zero123 which work on tasks like novel view synthesis. The creation of Objaverse-XL essentially fills the gap in data availability for 3D vision tasks, particularly in light of increasing demand and interest in simulation, AR and VR technologies, and generative AI.

Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?
The dataset was created by researchers at the Allen Institute for AI and at the University of Washington, Seattle.

What support was needed to make this dataset? (e.g., who funded the creation of the dataset? If there is an associated grant, provide the name of the grantor and the grant name and number, or if it was supported by a company or government agency, give those details.)
Stability AI provided compute support and guidance for the main experiments in the paper. The Allen Institute for AI also provided compute support for collecting the dataset and performing rendering.

Any other comments?
No

C.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description.
Instances of the dataset comprise of 3D objects and their associated metadata.

How many instances are there in total (of each type, if appropriate)?
There are approximately 10.2 million rendered 3D files. About 56% come from GitHub, 35% come from Thingiverse, 8% come from Sketchfab, and less than 1% come from Polycam and the Smithsonian Institute. We also release additional links to indexed GitHub files that are not included in the count due to being removed by deduplication or not being easily importable into Blender.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).
The dataset contains a sample of objects on GitHub, Sketchfab, Thingiverse, and Polycam, along with all the objects from the Smithsonian Institute.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.
Instances of the dataset vary based on source. For Polycam and Sketchfab objects, we release the full

3D objects along with associated metadata. For GitHub, Thingiverse, and Smithsonian objects, we release links to each of the files that can then be downloaded from the source, along with metadata such as license, poly count, vertex count, and other attributes discussed in Section 3.2.

Is there a label or target associated with each instance? If so, please provide a description.
No, just the 3D object and associated metadata. Labels and targets may be derived from the metadata, but vary based on the task.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.
No, information is not missing.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.
Individual instances are treated as independent.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them.
There are no recommended data splits across the entire dataset as splits vary based on task.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.
We deduplicated the objects by taking a sha256 of the file contents. There may still be near duplicates that exist, if the objects are slightly modified, which could potentially be filtered out, if desirable, using CLIP embeddings of the renders.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.
There are no fees. The Smithsonian data is hosted on a governmental website, which we believe will be well supported over time. The Sketchfab and Polycam data will be available to download easily on our platform. For Thingiverse and GitHub, the platforms are relatively stable and we expect most of the content to remain in place. For Thingiverse, an API key must be provided to download the content from their API. For GitHub, the data can be easily cloned. Users must follow the license of the content with which the original files were distributed and the terms of service for each platform.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description.
While rare, it is possible that confidential data exists as part of the 3D objects on the platforms.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why.
While rare, it is possible that data that is considered offensive, insulting, threatening, or might cause anxiety exists as part of the 3D objects on the platforms.

Does the dataset relate to people? If not, you may skip the remaining questions in this section.
People may be present in the dataset, but only make up a small part of it. Section discusses the results and analysis of running a face detector on renders of the objects. Most often, faces appear from dolls,

historic sculptures, and anthropomorphic animations. Moreover, even including such data, only about 2.5% captured faces.

Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.
We do not identify people by subpopulation.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how.
If a scanned person is included in the dataset, it may be possible to visually identify them or identify them if their name is included as part of the metadata.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description.
While rare, it is possible that sensitive data may exist as part of the 3D objects on the platforms.

Any other comments?
No.

C.3 Collection

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.
The data was directly observable and hosted on several platforms, including GitHub, Thingiverse, Sketchfab, Polycam, and the Smithsonian Institute.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created. Finally, list when the dataset was first published.
Sketchfab data was collected as part of Objaverse 1.0. The new data was collected in Q1 & Q2 of 2023.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? How were these mechanisms or procedures validated?
Python scripts were used to collect the data.

What was the resource cost of collecting the data? (e.g. what were the required computational resources, and the associated financial costs)
The cost of collecting the dataset was on the order of several thousand dollars, including costs to find, index, and download it. Resources included AWS CPU instances.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?
The dataset is filtered down based on licensing restrictions, duplicate content, and based on if the 3D object can successfully be imported into Blender.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?
The data collection process was primarily performed by employed researchers at the Allen Institute for AI.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.
Institutional review boards were not involved in the collection of the dataset.

Does the dataset relate to people? If not, you may skip the remainder of the questions in this section.
People may be present in the dataset, but only make up a small part of it. Section discusses the results and analysis of running a face detector on renders of the objects. Most often, faces appear from dolls, historic sculptures, and anthropomorphic animations. Moreover, even including such data, only about 2.5% captured faces.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?
Data was collected from public facing platforms. On each of the platforms, users opted to make their data public.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.
Individuals were not notified about the collection of the dataset.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.
Individuals were not notified about the collection of the dataset.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate)
Individuals were not notified about the collection of the dataset.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.
An analysis has not been conducted.

Any other comments?
No.

C.4 Processing / Cleaning / Labeling

Was any preprocessing/cleaning/labeling of the data done(e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remainder of the questions in this section.
Preprocessing the data was performed by computing renders of the objects and performing

deduplication.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data.
s The data that is downloaded contains the raw data and does not modify the individual files.

Is the software used to preprocess/clean/label the instances available? If so, please provide a link or other access point.
The software for cleaning the dataset and rendering will be made available.

Any other comments?
No.

C.5 Uses

Has the dataset been used for any tasks already? If so, please provide a description.
Yes, please see Section 4 of the paper.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.
We recommend checking the Semantic Scholar page for the Objaverse-XL and Objaverse 1.0 papers to find up to date papers that use the dataset.

What (other) tasks could the dataset be used for?
The dataset could be used for a large number of use cases. Examples include making 3D tools more accessible (e.g., 3D impainting, text to 3D, image to 3D), robotic simulation & embodied AI, training video models on animations, 2D vision tasks such as segmentation, and more.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) If so, please provide a description. Is there anything a future user could do to mitigate these undesirable harms?
Users should follow the license of the individual objects distributed as part of this dataset.

Are there tasks for which the dataset should not be used? If so, please provide a description.
New tasks must make sure to follow the license of the dataset and the license of the individual objects distributed as part of the dataset.

Any other comments?
No.

C.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description.
Yes. The dataset will be made public.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?

The dataset will be distributed through a Python API.

When will the dataset be distributed?
The dataset will be made publicly available towards the end of June, 2023.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.
The dataset as a whole will be distributed under the ODC-By 1.0 license. The individual objects are subject to the licenses that they are released under, and users need to assess license questions based on downstream use.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.
The individual objects are subject to the licenses that they are released under, and users need to assess license questions based on downstream use.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.
No.

Any other comments?
No.

C.7 Maintenance

Who is supporting/hosting/maintaining the dataset?
The dataset will be hosted on Hugging Face.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?
Please contact `mattd@allenai.org`.

Is there an erratum? If so, please provide a link or other access point.
No.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?
The dataset is currently self contained without immediate plans for updates.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced.
People may contact us to add specific samples to a blacklist.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users.

Objaverse 1.0 will continue to be supported.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users? If so, please provide a description.
We encourage others to build and extend the dataset for different use cases and may highlight some of those use cases if applicable.

Any other comments?
No

D Aesthetic Annotations

We run LAION-Aesthetics V2 [55] on renders of the objects, which can be used for filtering a higher quality subset of the objects. We group the objects into 3 tiers, which are depicted in Table 4. Figure 19 shows examples of renders of objects placed on the different tiers.

Category	Description	Aesthetic Score Cutoff	Percentage of Dataset
T1	Highest aesthetic ranked objects	Greater than 4.5	14.2%
T2	Medium aesthetic ranked objects	Between 4 and 4.5	69.2%
T3	Low aesthetic ranked objects	Less than 4	16.6%

Table 4: LAION-Aesthetics V2 categorization for renders of Objaverse-XL objects.

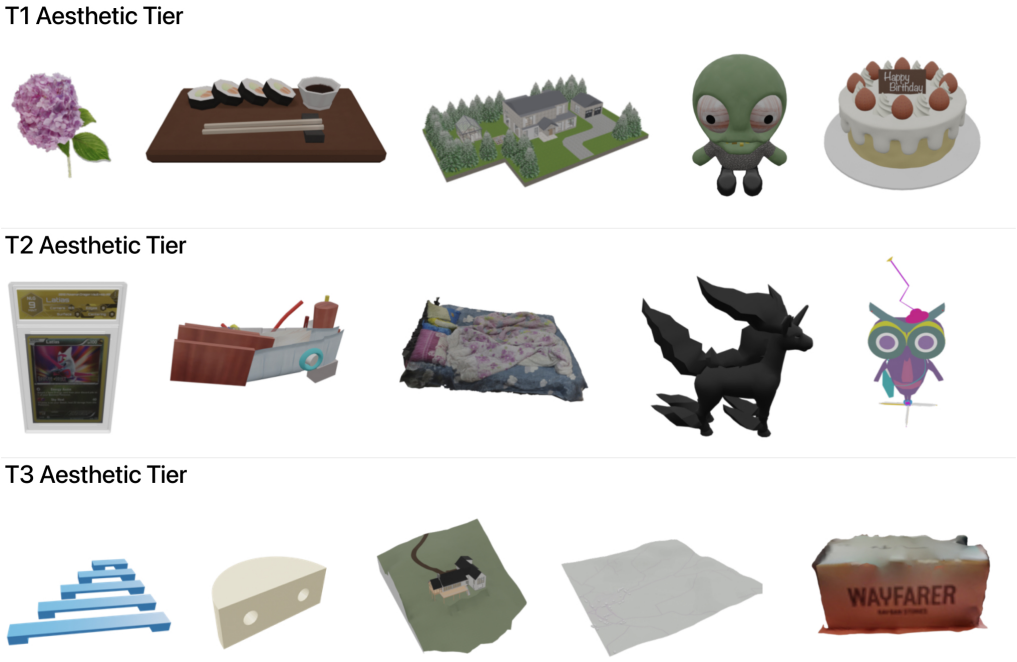


Figure 19: Random samples of renders showing LAION-Aesthetic V2 annotations across different tiers. Empirically, T1 tends to have the highest quality objects, followed by T2 and then T3.