



Make-it-Real:

Unleashing Large Multimodal Model’s Ability for Painting 3D Objects with Realistic Materials

Ye Fang^{1,4*}, Zeyi Sun^{2,4*}, Tong Wu^{3,6}, Jiaqi Wang⁴,
Ziwei Liu⁵, Gordon Wetzstein⁶, and Dahua Lin^{3,4}

¹ Fudan University

² Shanghai Jiao Tong University

³ The Chinese University of Hong Kong

⁴ Shanghai AI Laboratory

⁵ S-Lab, Nanyang Technological University

⁶ Stanford University



Fig. 1: **Make-it-Real** can refine any diffuse-map-only 3D object from both CAD design and generative models.

Abstract. Physically realistic materials are pivotal in augmenting the realism of 3D assets across various applications and lighting conditions. However, existing 3D assets and generative models often lack authentic material properties. Manual assignment of materials using graphic software is a tedious and time-consuming task. In this paper, we exploit advancements in Multimodal Large Language Models (MLLMs), particularly GPT-4V, to present a novel approach, **Make-it-Real**: **1)** We demonstrate that GPT-4V can effectively recognize and describe materials, allowing the construction of a detailed material library. **2)** Utilizing a combination of visual cues and hierarchical text prompts, GPT-4V precisely identifies and aligns materials with the corresponding components of 3D objects. **3)** The correctly matched materials are then meticulously applied as reference for the new SVBRDF material

*Equal Contribution

generation according to the original diffuse map, significantly enhancing their visual authenticity. Make-it-Real offers a streamlined integration into the 3D content creation workflow, showcasing its utility as an essential tool for developers of 3D assets. Our project website is at: <https://SunzeY.github.io/Make-it-Real/>.

Keywords: Texture Synthesis, MLLMs, Material Refinement, 3D Content Creation

1 Introduction

High-quality materials are important for the nuanced inclusion of view-dependent and lighting-dependent effects in 3D assets for traditional graphics pipelines, critical for achieving realism in gaming, online product showcasing, and virtual/augmented reality. However, many existing assets and generated objects often lack realistic material properties, limiting their application in downstream tasks. Furthermore, creating hand-designed realistic textures necessitates specialized graphic software and involves a laborious and time-consuming process, compounded by significant creative challenges [28].

Traditional computer graphics methods have been either manually creating materials or reconstructing them from physical measurements. Emerging text-to-3D generative models [6, 8, 10, 18, 30, 45, 50] are successful in creating complex geometries and detailed appearances, but they struggle to generate physically realistic materials, hampering their practical applicability. Recent studies also consider advanced aspects of appearance generation [9, 11, 57, 60], although they use simplified material models, limiting the diversity and realism of the generated materials. Moreover, they often require prohibitive training time.

The emergence of Multimodal Large Language Models (MLLMs) [2, 5, 12, 21, 41, 51] provides novel approaches to problem-solving. These models have powerful visual understanding and recognition capabilities, along with a vast repository of prior knowledge, which covers the task of material estimation, as shown in Fig. 2. Specifically, we are using GPT-4V(ision) for the identification and matching of materials. Specifically, we first create highly detailed descriptions for materials and build a comprehensive material library. Next, we use GPT-4V to identify and match materials for each part of the object, utilizing visual prompts [59] and hierarchical text prompts. Finally, we achieve precise transferring of materials from the material library to man-made meshes, text-to-3D models, and image-to-3D model generated meshes, endowing them with material properties.

Our work differs from the aforementioned studies by leveraging prior knowledge from foundation models like GPT-4V to help acquire materials, and leveraging existing libraries as references to create new materials. A direct application of our work is empowering mature 3D content generation models with enhanced realism, depth effects, and improved visual quality. Furthermore, this can greatly streamline the creation process for 3D asset developers, as they only need to paint the diffuse texture on the mesh, and then the material properties for each part can be automatically generated.

To achieve our objective of transforming rough, unrealistic 3D objects into ones with realistic materials, we address the problem in three steps: Firstly, we prepare a high-quality, selectable library of materials featuring detailed descriptions created by GPT-4V. This material library is constructed based on the MatSynth dataset, which comprises over 4,000 high-resolution, tileable materials available under the CC0 license. Secondly, to segment the 3D mesh into distinct components, we utilize segmentation and back-projection techniques on multi-view images to accurately segment texture maps according to different parts of the 3D object. Thirdly, suitable materials will be assigned to each part, which is the key part of this paper. It involves the accurate identification of the material for each component and matching it with the materials in the material library. Unlike previous material recognition methods [4, 35], the challenge lies in recognizing material from rendered images of 3D objects with only an diffuse map, which may be distorted, reflecting only the base color of the object (as shown in Fig. 1). Also it may be affected by shading or coupled with the color of illumination, both can influence judgment. Therefore, the model requires a strong recognition ability for materials and prior knowledge of object types and material information. Finally, the selected materials are meticulously applied to the 3D assets, endowing them with a realistic visual quality.

In summary, our contributions are as follows:

- We present the first exploration of leveraging multimodal large language models (MLLMs), *e.g.*, GPT-4V for material recognition and unleashing their potential in applying real-world materials to 3D objects.
- We create a material library containing thousands of materials with highly detailed descriptions readily for MLLMs to look up and assign.
- We develop an effective pipeline for texture segmentation, material identification and matching, enabling the high-quality application of materials to 3D assets.

2 Related Work

3D Object Generation. The generation of 3D models using deep learning methods has experienced rapid development in recent times. The mainstream research can be primarily divided into two categories. The first category relies on techniques that optimize a Neural Radiance Field (NeRF) [38] or 3D Gaussian [27] guided by 2D diffusion model through score-distillation-sampling (SDS) loss [9, 34, 37, 43, 46, 49, 54, 55] or Score Jacobian Chaining (SJC) [53]. The second category aims at obtaining 3D representation via direct inference model, *e.g.*, Point-E [39], Shap-E [26], and LRM [23], proven fast with high quality through large-scale pretraining [20, 22, 31, 48, 58]. Although the capabilities of these methods are continuously improving, they still lack a high degree of realism in textures. More importantly, since the textures of 3D objects obtained by these methods are shaded, they cannot directly respond to lighting changes under different lighting conditions, leading to less realism. Although some works attempt

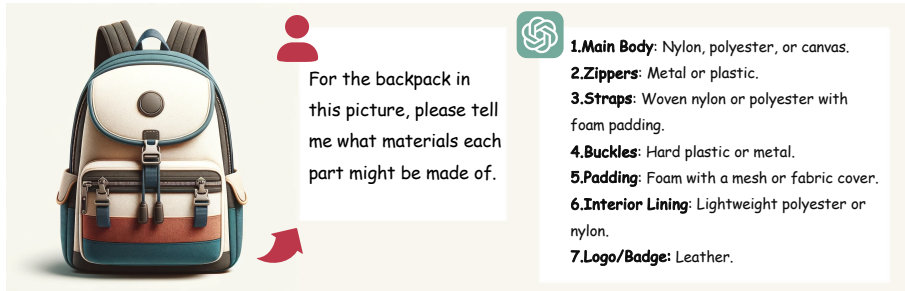


Fig. 2: Motivative observation. This example illustrates that GPT-4V can identify and appropriately conjecture materials to object components.

to generate PBR textures, their results are considerably limited to generate physically realistic materials due to the sparse generative guidance of SDS [9, 60]. Our work is the first to introduce the prior knowledge of MLLMs into the texture synthesis process. Through extensive experiments, we verify that our method can be seamlessly and effectively applied to generated 3D objects, facilitating their graphic downstream applications.

Material Capture and Generation. Recent studies such as MatPalette [35] and TwoshotBRDF [4] have made progress in the recognition and extraction of 3D object materials, allowing for the retrieval of SVBRDF information from images of real materials [35] and combining object shape and illumination [4]. On the other hand, works like Paint-it [60], Matlader [57], Fantasia3D [9], and TANGO [11] focus on generating text-controllable 3D meshes with physically-based rendering material properties. However, these methods either require extensive training beforehand or suffer from long computational times for generating complete PBR maps due to the calculation of SDS Loss. Additionally, they are limited to synthesizing only a subset of PBR textures and cannot generate the full range of material properties, such as height and displacement.

Multimodal Large Language Models. In the wake of the advancements achieved by large language models (LLMs) [2, 5, 12, 21, 41, 51], domain of research has increasingly turned its attention towards multimodal large language models (MLLMs). Recent advances in this field focus on the integration of vision understanding capabilities with LLMs [1, 3, 7, 17, 19, 24, 32, 33, 44, 47, 61]. The advent of GPT-4V [40] has marked a significant milestone in the evolution of MLLMs, demonstrating groundbreaking 2D comprehending capabilities and open-world knowledge. Although GPT-4V cannot directly process 3D data, a pioneering work, GPTEval3D [56], first exploited GPT-4V’s ability in evaluating the quality of generated 3D objects, and found that GPT-4V’s judgement was in line with human evaluation. In this work, we delve into a new application of GPT-4V for material assignment of 3D objects.

3 Methodology

3.1 Problem Definition

The extraction and restoration of materials corresponding to 3D meshes represent a challenging problem. This complexity is particularly pronounced when identifying and segregating different material regions within 3D objects with texture map limited in diffuse RGB domain, lacking in realism and stereoscopic effects. The process necessitates accurately identifying the material type of each region on the 3D mesh based solely on its intrinsic color. Following this identification, an appropriate material from our library is selected to enhance the visual fidelity of the original mesh. Moreover, the process involves accurately mapping the generated 2D material maps back onto the 3D mesh, ensuring the preservation of the material’s natural appearance and physical properties.

The specific problem definition is as follows: material maps primarily encompass seven attributes: **Diffuse**, **Albedo**, **Normal**, **Roughness**, **Metalness**, **Height**, and **Specular**, ensuring compatibility with both diffuse-based (e.g., [16]) and diffuse-based (e.g., [36]) workflows. We focus on the diffuse-based rendering for an easier application to the current 3D content generation model [9, 60]. Given the advancements in generating high-quality 3D shapes and diffuse maps, the restoration of realistic material properties remains a challenge. We highlight a novel problem: given a 3D mesh ($\tilde{O}$) and known diffuse ($\tilde{D}$) map, which reflect the object’s intrinsic appearance, the goal is to extract and restore all other SVBRDF attributes of the object.

$$\textit{Make-it-Real}(\tilde{O}, \tilde{D}) = \{N, R, M, H, S\} \quad (1)$$

Our goal is to utilize the Make-it-Real pipeline for the identification and synthesis of Spatially Varing Bidirectional Reflectance Distribution (SVBRDF) material maps, leveraging the existing diffuse map and prior knowledge to query the most appropriate material attribute maps $SV = \{N, R, M, H, S\}$, to use as references before further refinement.

3.2 Overview

In this section, we introduce our material synthesis pipeline Make-it-Real. As depicted in Fig. 4, the process begins with rendering images from three different perspectives of a 3D model. Subsequent steps involve using SemanticSAM [29] to semantically mask and annotate different parts of the model from varied view-points via Set-of-Mark (SoM) [59]. Following this, we employ the Multimodal Large Language Model (MLLM) to retrieve the best matched material to different parts. We partition the texture map with specific material type using back-projection. Then, we estimate the SVBRDF values by pixel, referenced by original diffuse and the queried material maps of different parts from our material library. ultimately yielding texture-aligned material maps of the object.

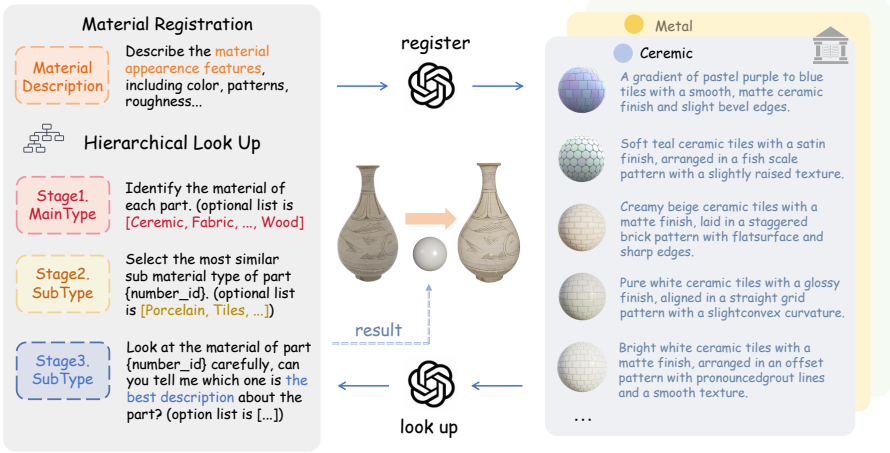


Fig. 3: Material Library with Fine-Grained Annotations. Utilizing GPT-4V model, we develop a material library, meticulously generating and cataloging comprehensive descriptions for each material. This structured repository facilitates hierarchical querying for material allocation in subsequent looking up processes.

3.3 Material Library

We aim to leverage both the material priors of the mesh as predicted by GPT-4V and the characteristics of similar materials from existing materials used by artists as references. As a result, we construct a fine-grained, annotated material library for Make, illustrated in Fig. 3. The library comprises 1,394 unique, tileable materials, spanning 13 major types ranging from ceramics to wood. Each material is equipped with detailed textual annotations generated by GPT-4V, which precisely describe the material’s appearance characteristics, providing rich semantic information for subsequent material assignment processes.

Data Source. Our data primarily originates from the MatSynth dataset [52], which offers comprehensive material maps with 4K resolution under the CC0 license. Each material is represented by seven maps: diffuse, diffuse normal, height, roughness, metallic and specular. We select 13 major types of materials from this dataset and crawl the corresponding material sphere images. These images are subsequently utilized to query GPT-4V for generating detailed descriptions of each material.

Prompt Design. Our prompt design aims to enable GPT-4V to generate detailed descriptions of each material based on realistic material sphere images, focusing on distinguishing appearance features such as color, pattern, roughness, metalness, embossing pattern, and material condition, while avoiding irrelevant information like the overall shape of the object. Our goal is to generate highly content-relevant descriptions that not only focus on appearance and attributes but also include knowledge-related information to differentiate subtle differences between materials. This approach is intended to precisely capture and describe the characteristics of various materials, facilitating fine-grained retrieval by GPT-4V, rather than mere categorization.

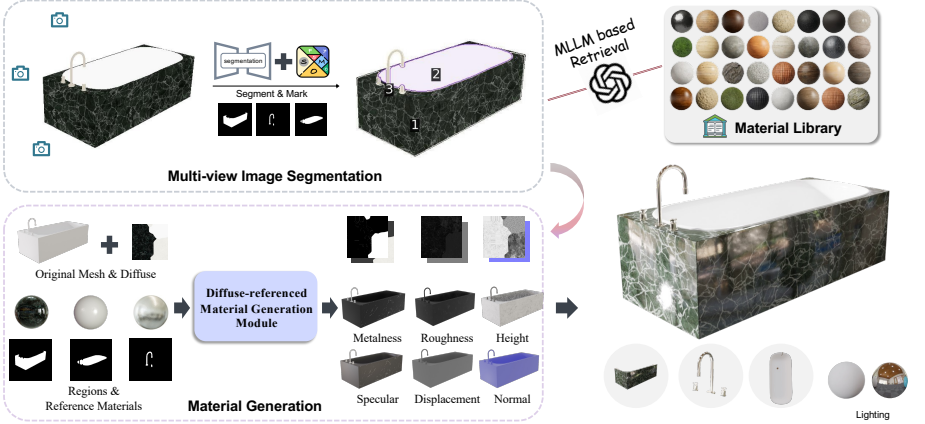


Fig. 4: Overall pipeline. This pipeline of Make-it-Real is composed of Multi-View Image Segmentation, MLLM-based Material Matching, and SVBRDF Maps Generation.

By integrating a fine-grained, annotated PBR (Physically Based Rendering) material library with detailed textual annotations generated by GPT-4V, our approach provides rich and precise material descriptions for 3D meshes. This not only enhances the search efficiency and practicality of the material library but also offers crucial semantic information for subsequent material assignment processes, significantly improving the ability to extract and restore materials from rendered images of 3D meshes.

3.4 Multi-View Image Segmentation

To accurately identify and segment different material regions on 3D meshes with diffuse RGB, we adopt an innovative segmentation strategy based on multi-view 2D image rendering. First, we render a series of 2D images of the 3D mesh from different perspectives. The goal of this step is to capture a comprehensive view of the 3D model, ensuring that we can identify material regions from various angles. Subsequently, we employ the Semantic-SAM [29] model for precise segmentation of the 2D rendered images. To enhance the accuracy of the segmentation and reduce noise interference, we introduce a mask filtering processing module. This module effectively handles and reduces the overlap and conflict between different masks, ensuring the clarity and accuracy of the segmentation results.

3.5 MLLM-based Material Matching

Following the successful segmentation of regions for multi-views of 3D meshes, our subsequent objective focuses on the allocation of appropriate materials to these regions. To achieve this aim, we introduce a material retrieval strategy powered by Multimodal Large Language Models (MLLMs), specifically leveraging the visual cues proposed by the Set-of-Mark (SoM) [59] model in conjunction

with the robust multimodal understanding capabilities of GPT-4V. To enhance the precision of recognition, we employ a meticulously designed combination of visual and hierarchical text prompts in tandem with GPT-4V to retrieve the most suitable materials, in Fig. 3. This approach signifies a novel integration of language model insights and visual perception for the advancement of material assignment in 3D modeling, underscoring the potential of MLLMs in facilitating complex, multimodal tasks in computer vision and graphics.

Visual Prompt Based on SoM. In our method, we draw inspiration from the Set-of-Mark (SoM) technique [59], which overlays an array of spatially coherent and semantically interpretable marks on images to stimulate the visual understanding capabilities of large multimodal models, such as GPT-4V. By incorporating SoM technology, we annotate each segmented region with a unique identifier as a visual cue, aiding the model in comprehending and analyzing each distinct material area. This approach enables a more nuanced interaction between the model and visual content, facilitating more effective and precise processing of material regions.

Hierarchical Text Prompt. To ensure efficiency and accuracy in assigning appropriate materials to segmented regions of 3D meshes, we adopt a hierarchical text prompting approach. This process initiates by identifying the primary material type corresponding to each region. Subsequently, hierarchical prompts guide GPT-4V in discerning specific subclasses within the primary material category. We retrieve all descriptions under this subclass material category to pinpoint the description that best matches the segmented block. Such hierarchical processing enables us to search our material library with greater granularity, identifying the most fitting description for each material segment. This approach not only facilitates the allocation of the most suitable materials to each area but also reduces memory and time expenditures.

3.6 SVBRDF Maps Generation

We conduct the generation of SVBRDF maps in two stages. Initially, preliminary texture map region segmentation is obtained through querying material regions. Subsequently, referenced by the original diffuse map of an object, an estimation of BRDF values in the pixel space is performed by querying the matched realistic material maps.

Region-level Texture Map Partitioning. After obtaining the partitioned masks for each view of the 3D object, combined with preliminary material assignments and the corresponding rendered images, we facilitate the extraction of local rendered images specific to masked regions. These localized images capture intricate details and textures associated with distinct material regions on the 3D model. Subsequently, by back-projecting these local rendered images onto the 3D mesh and employing UV mapping, we align masks on the 2D image with specific regions on the texture map, thereby segmenting the texture. This procedure delineates dedicated areas within the texture map for each material zone identified in the rendered images, meticulously preserving the uniqueness and characteristics of each material while ensuring precise correspondence and

seamless integration across the 3D model. This culminates in an initial segmentation of the texture map, where each section correlates with the attributes of a precisely queried material sphere.

Pixel-level Diffuse-referenced Estimation. To precisely estimate the spatially varying BRDFs per pixel, we draw on the common practice among artists who typically use the diffuse map as a guide to create maps for other material properties when generating various texture maps. This process involves using the original object’s diffuse map as a reference, and the matched material sphere’s diffuse map as a key. We find the nearest neighbor pixel index in the key diffuse for each pixel’s RGB value in the query diffuse pixel by pixel, which we employ the K-d tree algorithm to accelerate the querying process. We get SVBRDF values at the corresponding pixel location by querying the key material map. Notably, this procedure applies histogram equalization to both the query diffuse and key diffuse, normalizing them to a similar color space. This method enables the generation of a series of spatially variant BRDF maps, which maintain high consistency with the texture of the diffuse color.

Consequently, we successfully develop a pipeline for generating the material maps from 3D objects with constrained diffuse maps. This process finally projects the generated material textures back onto the 3D meshes, thereby achieving a natural and authentic appearance, along with the physical properties of the materials available for photo-realistic rendering in different environments.

4 Experiments

To verify the effectiveness of the proposed method, we conduct refinement experiments mainly on two types of objects. The first type is artificial 3D models, with the primary model from Objaverse [14]; The second type is objects generated by state-of-the-art 3D generation methods. We validate the powerful capabilities of our method through extensive qualitative and quantitative experiments on both types of 3D models.

4.1 Texture refinement for existing 3D assets

We observe that the realism of objects is significantly impacted by the existing publicly available 3D assets, which have only diffuse maps and lack material properties. This deficiency extends to artificial 3D models derived from Computer-Aided Design (CAD), which necessitate automated texture assignment. To address this issue, we conduct texture synthesis on 3D models from Objaverse [14], which are available only with diffuse maps. Our methodology, Make-it-Real, is employed to generate additional material texture maps. The qualitative outcomes are illustrated in Fig. 5. Make-it-Real demonstrates the capability to accurately segment objects, assign multiple appropriate materials, and synthesize textures with high fidelity and photorealism.

Assets enhanced by Make-it-Real exhibit the capability for relighting operations, with some materials displaying significant highlights, such as a bathtub

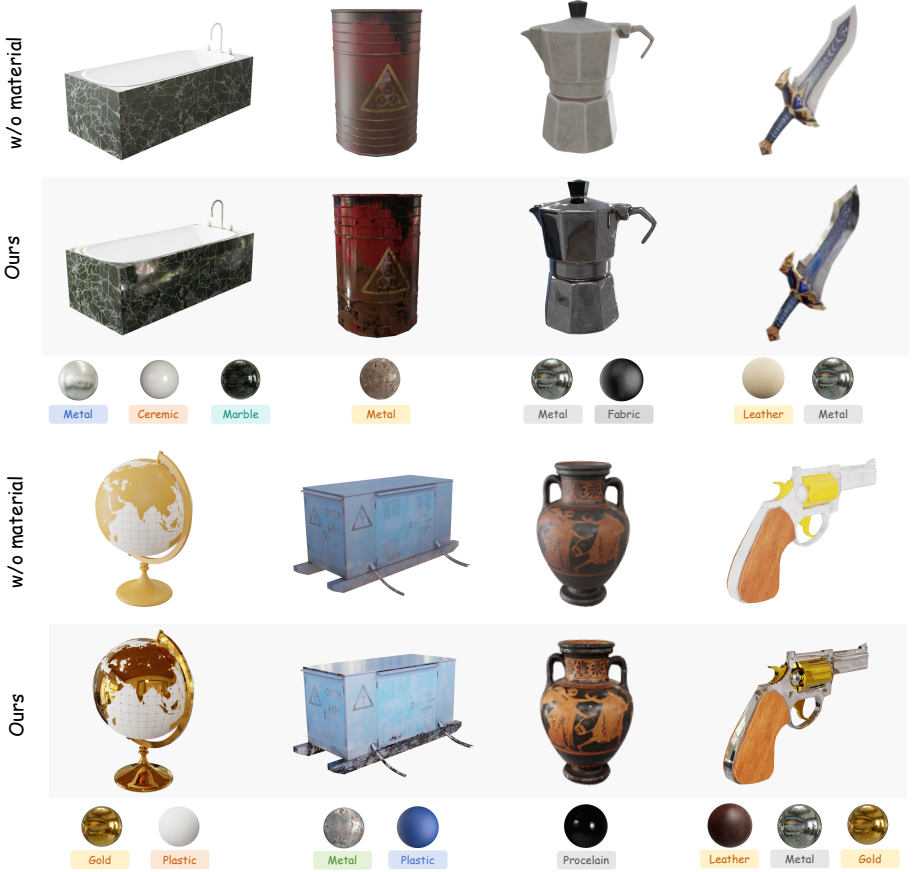


Fig. 5: Qualitative results of Make-it-Real refining 3D asserts without PBR map. Objects are selected from Objaverse [14] with diffuse only.

made of marble, a globe constructed from gold, and a vase crafted from ceramic. These different materials interact with light to varying degrees, resulting in distinct reflections of the same environment light and presenting varied textures. Moreover, the material characteristics change across regions; for example, the land and handle parts of the globe are imbued with the properties of gold, while other parts are identified as plastic. In addition, the material can alter in terms of roughness and appearance, such as the fine wrinkles that appear on the red box in the second column, and the blue box’s base showing more pronounced color contrast, enhancing realism and reflecting the object’s age of use.

4.2 Texture refinement for generated 3D object

We test the effectiveness of the proposed pipeline for texture refinement on generated 3D models. For image-to-3D, we use the state-of-the-art model TripoSR [50]. For text-to-3D, we use objects generated by typical models like In-

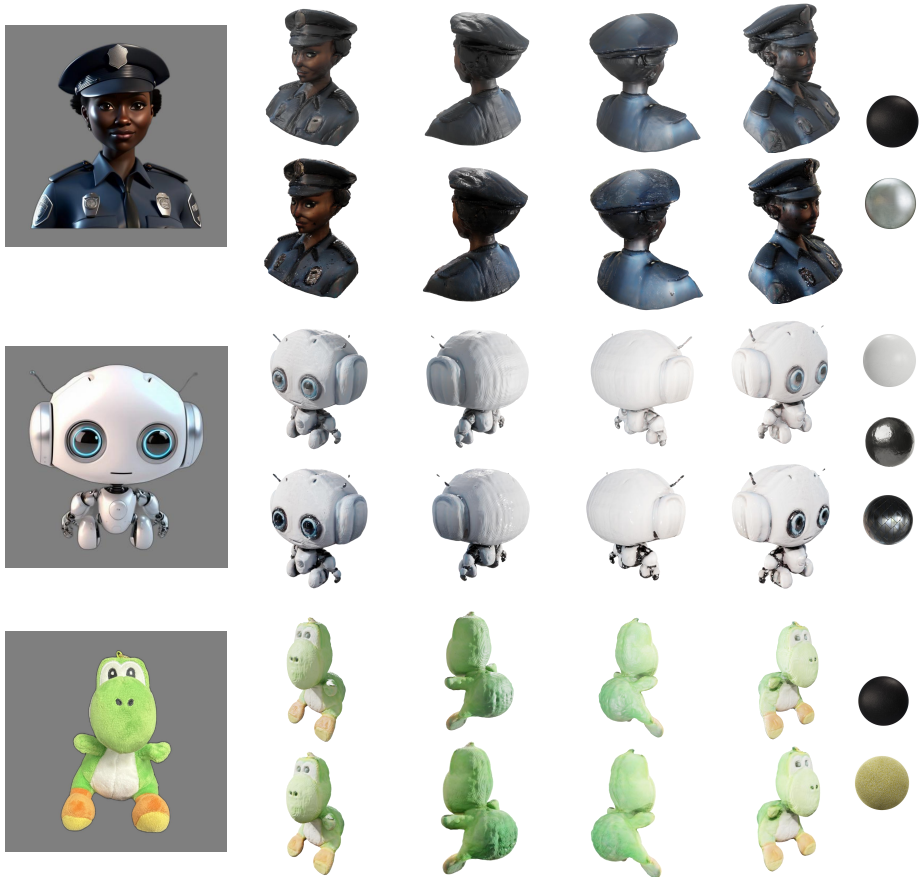


Fig. 6: Qualitative results of Make-it-Real refining 3D object generated by edge-cutting TripoSR [50] conditioned on single image. The first line per two are initial objects generated by TripoSR, while the second line are objects refined by Make-it-Real.

stant3D, MVDream and Fantasia3D [9, 30, 46]. Make-it-Real is used to refine these generated objects. Figure 6 displays the qualitative results of Tripo-SR, with the referenced image on the left. We note that the Tripo-SR model is capable of generating accurate shapes and consistent diffuse maps, yet it struggles to reconstruct material effects suitable for relighting in the images. By leveraging our model for enhancement, we observe that Make-it-Real successfully generates appropriate material maps for these models. In the first image, the badge exhibits a metallic sheen, and the clothing possesses a more fabric-like texture; in the second image, both the robot’s eyes and body parts exhibit brightness. The effects become more pronounced under different HDR conditions, demonstrated in the appendix. Figure 7 illustrates the outcomes of three different text-to-3D



Fig. 7: Qualitative results of Make-it-Real refining 3D object generated by edge-cutting text-to-3D models.

Domain	Method	Base object	+Make-it-Real
Image-to-3D	TripoSR [50]	36.4 %	63.6 %
Text-to-3D	Instant3D [31]	38.5 %	61.5 %
	MVDream [46]	44.1 %	55.9 %
	Fantasia3D [9]	46.2 %	53.8 %

Table 1: GPT evaluation. GPT’s preference comparison on Make-it-Real refined objects generated by state-of-the-art 3D generation methods with their original objects.

models. Similarly, our Make-it-Real successfully paint these models with materials.

We employ the same idea of GPTEval3D [56] for fair, human-aligned assessment, utilizing the powerful GPT4V model to compare models before and after refinement. Results are shown in Tab. 1. Compared to the original generated 3D models, our refined models are capable of enhancing the realism of 3D objects across all edge-cutting methods for 3D generation. Notably, as the 3D generation methods evolve and prior knowledge increases (from text prompts to single-view images), the realism of our refined results also improves. This validates that our approach can better endow materials with enhanced realism on the foundation of more advanced generation techniques.

Figures 6 and 7 display the qualitative experimental results. Figure 6 shows cases objects refined based on TripoSR [50]. Our Make-it-Real performs reasonable segmentation and material matching on object parts, refining them into more realistic models. Notably, as image-to-3D setting can utilize a high-

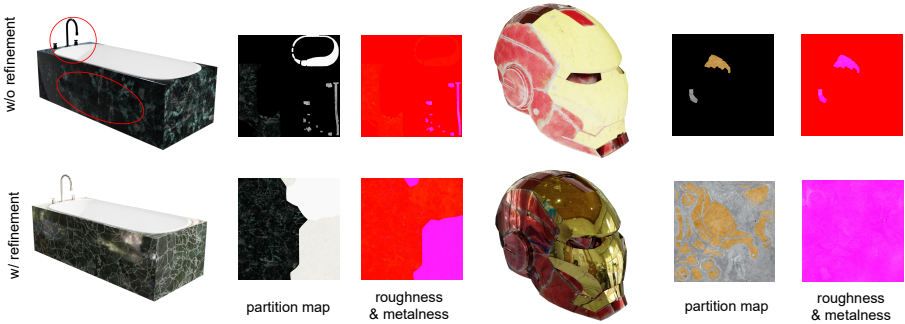


Fig. 8: Ablation study for texture refinement module.

resolution single-view image as prior, we first generate a detailed description of these images using MLLM. This can compensate for the lack of material information in images rendered from the initial 3D object (as seen in the first example of the police metal badge), thereby generating better results. Figure 7 presents the refinement results on different text-to-3D methods. Compared to image-to-3D, the original quality of these objects is relatively lower, yet our method still robustly matches the correct materials, thereby enhancing the quality of the objects.

4.3 Ablation Study

Missing Part Refinement. Upon completing the query retrieval steps, we precisely assign material types to the segmented regions within the view, thereby initially determining the material categories for each block on the texture map. However, due to limitations such as the number of views, there may exist regions on the texture map where materials are not assigned. To address this issue, we design a heuristic texture refinement post-processing operation. Specifically, we calculate the mean color values of the diffuse maps for regions where materials have been assigned and use this as a basis for clustering the remaining unassigned areas. This operation extends the material mapping of the texture map to full resolution, ensuring overall consistency in the texture of the 3D object while avoiding the absence of textures. This method not only improves the coverage of textures but also enhances the visual quality of the final 3D model. Results are shown in Fig. 8.

Effects of different material maps. To examine the individual contributions of various material maps to the rendering process, we incrementally introduced each material map—specifically, roughness, metalness, and displacement maps—into our model, shown in Fig. 9. The observed effects of these incremental additions were systematically analyzed. For instance, the inclusion of the roughness map was found to significantly enhance the depiction of surface textures, imparting a noticeable wrinkled appearance, while the introduction of the metalness map resulted in enhanced reflective properties, contributing to the realism of

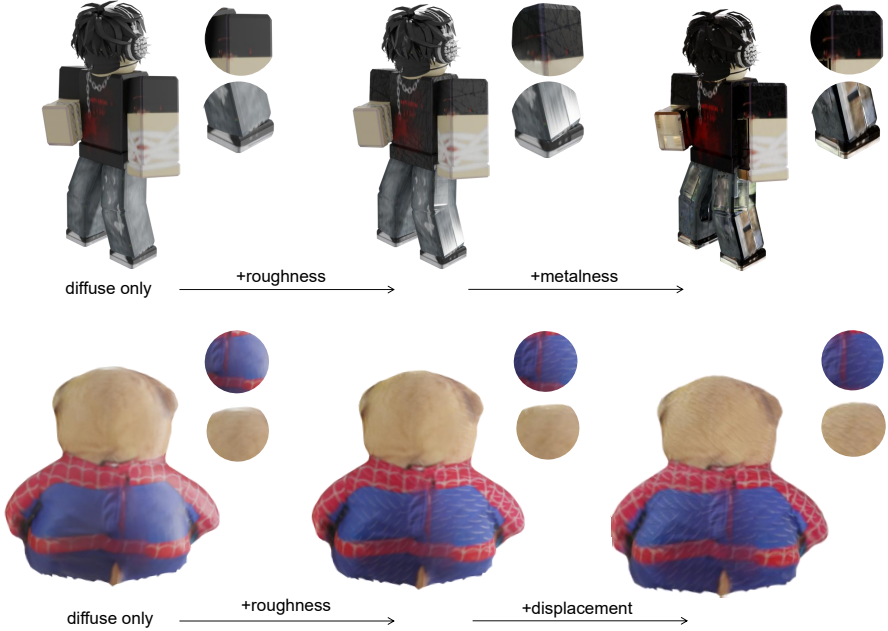


Fig. 9: Ablation study for effect of different queried maps.

metallic surfaces. Furthermore, the combination of both roughness and displacement maps was demonstrated to produce a pronounced three-dimensional effect, as illustrated in the second row of our results. These outcomes emphatically validate the utility of querying each specific material map within our framework. Our pipeline is designed to ensure the retrieval of a comprehensive set of material attributes, thereby facilitating their effective application in downstream 3D rendering pipelines.

5 Conclusion and Discussion

In this paper, we present a novel framework leveraging MLLMs prior of the world to build a material library and proposing an automatic pipeline to refine and synthesize new PBR maps for initial 3D models, achieving highly photo-realistic PBR textures maps synthesis. Experimental results confirm that our approach can automatically refine both generated and CAD models to achieve photo-realism under dynamic lighting conditions. We believe Make-it-Real is a new and promising solution in the last few procedures of AI based 3D content creation pipeline with the development of MLLMs like GPT-4V [40] as well as the roaring field of deep learning based 3D generation from scratch.

Limitations and future work. While promising results have been achieved by MLLMs for texture assignment through Make-it-Real. Our work still faces several unresolved challenges. First, although our method can achieve appropriate texture assignment for diffuse-only model, it does not support reverse transform

from shaded texture map to diffuse map for generated 3D objects. This causes problem of assigning different materials to the dark shadow and highlight area when generated object with mesh already shaded in different lighting conditions. Second, we find base model quality is essential for MLLM to assign correct materials. When base 3D object is of low quality, it is difficult for MLLMs to identify object properties when ground truth text prompt describing the object is not available. We consider methods mitigating these challenges as well as adding user friendly control into our fully-automatic pipeline as future work.

A Supplementary Material Overview

In this supplementary material, we provide additional details and results that are not included in the main paper due to the space limit. The attached video includes intuitive and interesting qualitative results of Make-it-Real.

B Details of Make-it-Real

In this section, we detail the pipeline that briefly outlined in the main paper. We commence by elaborating on the generation of SVBRDF Maps, incorporating illustrative figures to detail the process involving operations in computer graphics. Appendix B.1 is dedicated to explaining the acquisition of region-level texture map partition. Following that, in Appendix B.2 we discuss the method of pixel-level diffuse-referenced estimation. Then, in Appendix B.3 we declare some details of rendering procedure. Appendix B.4 details the prompt design for material captioning and matching. Appendix B.5 reports the details of the GPT-4V based evaluation.

B.1 Texture Partition Module Design

As illustrated in Fig. 10, our process initiates with the rendering of a 3D object incorporating the original diffuse (i.e. query diffuse) from multiple perspectives. Following this rendering phase, we employ GPT-4V alongside a segmentor to derive segmented masks for the materials associated with each viewpoint. The subsequent step involves the extraction of regions masked in these images and their back-projection onto the mesh of the object. By examining the object from all acquired viewpoints and applying UV unwrapping techniques, we achieve preliminary segmentation of all materials. Subsequently, each material’s segment is refined using a diffuse-based mask refinement operation. Ultimately, by combining the segments of all materials, we obtain a region-level texture partition map, which serves to guide our subsequent, more detailed operations.

B.2 Diffused-referenced Module Design

As illustrated in Fig. 11, we developed a pixel-level diffuse-referenced estimation module, building upon the foundations laid out in Appendix B.1. This module is inspired by a technique frequently employed by 3D artists, who often utilize diffuse maps as a reference to generate images of other material properties. Accordingly, we designate the known diffuse map as the query diffuse, and the diffuse corresponding to the region of interest in the material as the key diffuse.

The process to precisely obtain the final SVBRDF maps is divided into four steps: 1) Initially, to address potential gaps in color intensity between the two diffuse maps, histogram equalization is employed to achieve a more uniform color distribution across the image. 2) Subsequently, for each pixel on the query diffuse—termed a reference pixel—we seek the most similar neighboring color

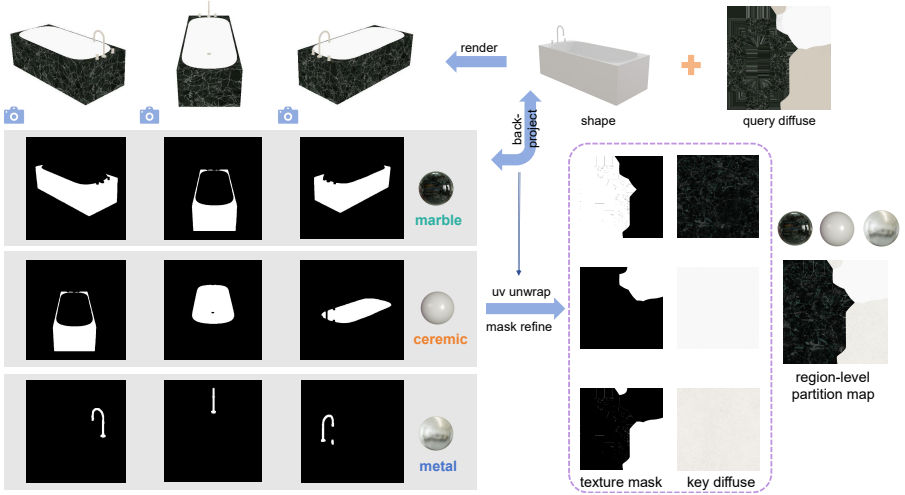


Fig. 10: Region-level texture partition module. This module extracts and back-projects localized rendered images on to a 3D mesh, using UV unwrapping for texture segmentation, thereby resulting in precise partition map of different materials.

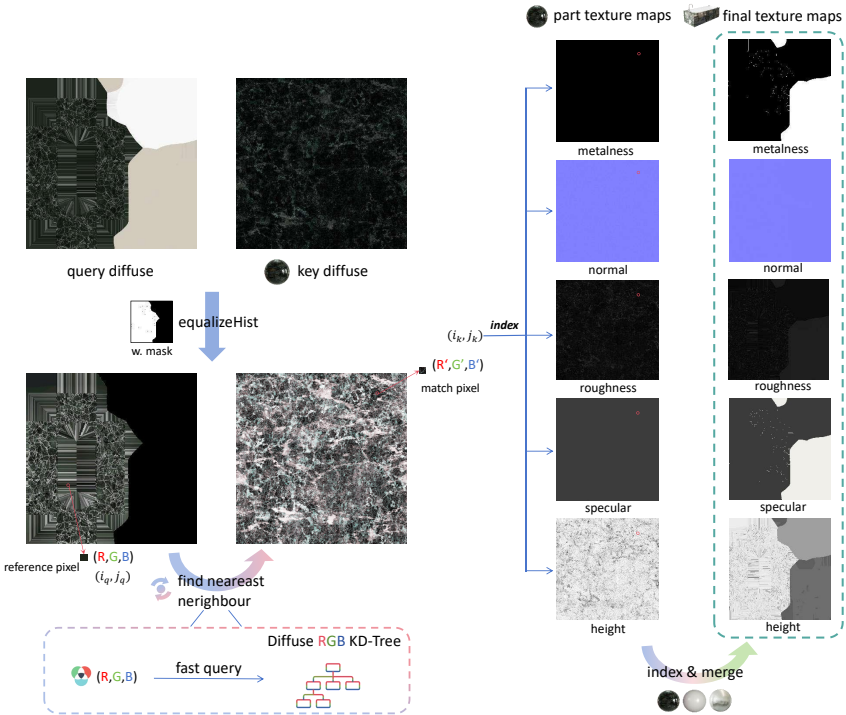


Fig. 11: Pixel-level diffuse-referenced estimation module. We generate spatially variant BRDF maps by referencing diffuse maps, employing KD-Tree algorithm for efficient nearest neighbor searches, and normalizing colors via histogram equalization.

index on the key diffuse. Given the high dimensionality of both maps (typically 1024x1024 pixels), a brute-force approach to this search would be computationally prohibitive. To this end, we accelerate the pixel query process using a KD-Tree algorithm, which organizes the RGB values of the key map’s diffuse into a KD-Tree for efficient nearest neighbor searches, reducing the computation time to under ten seconds. 3) The third step involves using the obtained indices to obtain corresponding values from the rest of the material maps. 4) Finally, by aggregating the query results for all material segments, we are able to generate the comprehensive spatially variant SVBRDF maps.

B.3 Rendering Procedure Details

In the context of computer graphics, “diffuse” typically denotes the primary color of a material, a concept analogous to “base color” in Physically-Based Rendering (PBR) paradigms, both representing the inherent color of the material under uniformly scattered illumination.

We observe that in the provided material collection [52] that a portion of the materials exhibit either completely black or low-quality diffuse maps. Therefore, for those materials with missing or subpar diffuse maps, we pragmatically substitute the base color map as the key diffuse map. In the majority of instances, our rendering adheres to the Cook-Torrance model [13], predominantly employing the original diffuse map, alongside our generated spatially variant normal map, metalness map, and roughness map to render objects. Depending on the workflow requirements, the inclusion of a height map, displacement map, specular map, and combinations of additional maps is optional. Regarding 2D-3D alignment techniques (such as back-projection and UV unwrapping), we rely on the Kaolin [25] package and back-projection with masks [45] techniques, with model parameters maintained consistent with [45]. This approach allows for the preservation of material visual quality in scenarios where the diffuse map is missing or of inferior quality, by leveraging the base color map as an effective substitute, thereby ensuring consistency and realism within the PBR workflow. Furthermore, the flexible application of various map types and 2D-3D alignment techniques enhances the detail and realism of rendered objects, meeting the demands of diverse rendering scenarios.

B.4 Prompt Details for Multi-modal Large Language Models

Prompts design has emerged as a pivotal factor for eliciting desired outcomes from MLLMs. This section delineates the intricacies of our prompt design for both material captioning and matching.

Material captioning. Due to limit context length for image tokens, it is currently impossible for MLLMs to directly memories thousands of images of material balls. To address this, we use GPT-4V to generate detailed captions for each material ball. As demonstrated in Fig. 12, we input concatenated images of material ball from same subcategory with prompt specifically tailored to highlight texture properties. This strategy guides GPT-4V to generate detailed caption



You are showed with a group of spherical PBR materials(all made of tiles, which is a kind of Ceramic, in total 5), can you generate a caption for each(about 20-30 words), try to distinguish their difference. Only describe the appearance features (must including color" and detailed material'(such as patterns, roughness, metalness, concave and convex patterns, condition))and don't give too much other information. Do not describe whether it is reflective or not. Do not describe the shape of overall object. Please use a dictionary to represent the output result {1:Description 1, 2: Description 2, 3: Description 3, 4: Description 4, 5: Description 5}



```
{1: "A gradient of pastel purple to blue tiles with a smooth, matte ceramic finish and slight bevel edges.",
2: "Soft teal ceramic tiles with a satin finish, arranged in a fish scale pattern with a slightly raised texture.",
3: "Creamy beige ceramic tiles with a matte finish, laid in a staggered brick pattern with flat surface and sharp edges.",
4: "Pure white ceramic tiles with a glossy finish, aligned in a straight grid pattern with a slight convex curvature.",
5: "Bright white ceramic tiles with a matte finish, arranged in an offset pattern with pronounced grout lines and a smooth texture."}
```

Fig. 12: Material captioning. We demonstrate how to construct detailed material descriptions, registering material information in textual form, thus providing a convenient bridge when querying with multi-modal large language models.

for each material ball, distinguishing the subtle differences between them. These detailed captions are then registered into material library.

Material matching. The MLLM-based material query process is exemplified through a simplified case, as illustrated in Fig. 13. Initially, GPT-4V is queried to identify the basic material for matching. Following this preliminary matching, GPT-4V is queried for more specific materials using the names of subcategories as filters, which narrows down the selection to a few candidates within the same subcategory. For final material selection, GPT-4V is prompted with detailed captions from the library, directing it to allocate the most suitable material. This is accomplished in conjunction with a meticulously engineered segmentation process in UV space, ultimately facilitating MLLM-based material matching. It is noteworthy that prompting GPT-4V to navigate through a three-level tree structure has been proven to enhance both the efficiency and accuracy of the matching process, as opposed to directly selecting from thousands of materials directly.

B.5 Evaluation Details of Quantitative Results

Data statistics. For TripoSR [50], we generate objects conditioned on 10 images from its official code-base and 100 images rendered from randomly sampled objaverse [15] objects. For [30, 46], we use text prompts proposed in GPTeval3D [56] to generate 130 objects. For Fantasia3D [9], we use 15 text prompts



Fig. 13: Detailed prompts of GPT-4V based material matching. Prompts in blue changes according to the current assigning part and GPT-4V's results.



Assuming you are an expert in determining whether the textures of 3D models meet Photorealistic standards (mainly focus on the quality of texture), we have generated some 3D objects using two different methods. These objects have the same shape but different textures. Above are the four-view images of a model generated by the two methods (the upper half is the four views of the object generated by method one, and the lower half is the four views of the object generated by method two). Please help me evaluate which of the two objects is more Photorealistic in terms of texture. Respond with 1 if you think the upper half object is more realistic. Respond with 2 if you think the lower half object is more realistic. Also tell me the reason in a new paragraph.



Based on the provided images, my evaluation would favor object 2 (the lower half object) as being more photorealistic in terms of texture.

The textures on object 2 have a more complex and nuanced appearance. The material of the uniform exhibits subtle variations in reflectivity and surface detail that suggest **a more realistic fabric texture**. Additionally, the **skin texture** appears more lifelike with variations in tone and reflectivity that are characteristic of human skin. The **metal police badges** on object 2 has a more nuanced reflection, which suggests **a more realistic interaction with environmental light**. In contrast, the textures on object 1, while still detailed, lack the same level of subtlety and variation. The uniform looks somewhat flat and less fabric-like, and the overall surface details are less convincing as natural variations.

Fig. 14: GPT-4V based evaluation prompts. We define a prompt for GPT-4V to generate human-aligned comparison over objects before and after our texture refinement.

from its original code-base and 15 text prompts designed by our human experts. Fantasia3D generate objects less than other methods due to the necessity for meticulous hyper-parameter design to generate meaningful objects.

Evaluation detail. For the purpose of evaluating objects pre- and post-refinement, we render four view images of each object and concatenate them vertically to facilitate a comprehensive assessment. We craft a specific prompt to guide GPT-4V in conducting an impartial comparison of the texture quality between two objects. As demonstrated in Fig. 14. GPT-4V’s advanced capabilities enable it to differentiate between the two objects, providing comparison results that closely align with assessments made by human experts. In the scenario evaluated, GPT-4V successfully identifies all enhancements (metal badges, human skin, and fabric textures).

C Additional Related Works

In the context of extracting real-world materials from single images, the methodologies most closely related to our work are Material Palette [35] and Photo-shape [42]. Material Palette enables the extraction of materials at the region

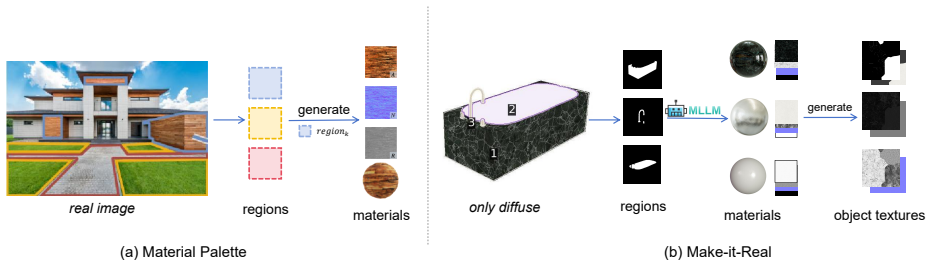


Fig. 15: Comparison between previous method and Make-it-Real. We demonstrate the distinctions between Material Palette [35] and our method in terms of material identification and extraction. Our overall pipeline presents a more challenging task, where the input is a rendered image with only diffuse information, and the output consists of textures for the entire object.

level from a single image, generating tilable texture maps for corresponding areas. Photoshape, on the other hand, automates the assignment of real materials to different parts of a 3D shape by training a material classification network. However, this approach needs the training of a material classifier, which is constrained to a limited set of object categories (e.g., chairs) and material types. Besides, both methods require rendered images of objects with real materials as input.

Our problem setting poses a more challenging task, as illustrated in Fig. 15, which involves identifying and recovering the original material properties from images of objects that only have a diffuse map, in addition to generating different material maps for the entire object. Humans have the capacity to intuitively infer the underlying material properties of 3D objects from single images containing only shape and basic colors. This capability is attributed to our robust material recognition abilities and comprehensive prior knowledge of object categories, colors, and material attributes. Building on this concept, our approach harnesses the potent image recognition capabilities and prior knowledge inherent in large-scale multimodal language models to efficiently execute region-level material extraction and identification, which is further enhanced by subsequent 2D-3D alignment and diffuse-reference techniques to generate and apply material texture maps for physically realistic rendering of 3D objects.

D Additional Experiments

D.1 User Preference Study

We randomly sample 100 objects generated from each methods (30 for Fantasia3D [9]) from Appendix B.5 for human volunteers. Their task is to express their preference for objects either before or after undergoing refinement by Make-it-Real. To facilitate this evaluation, we concatenate images rendered from four

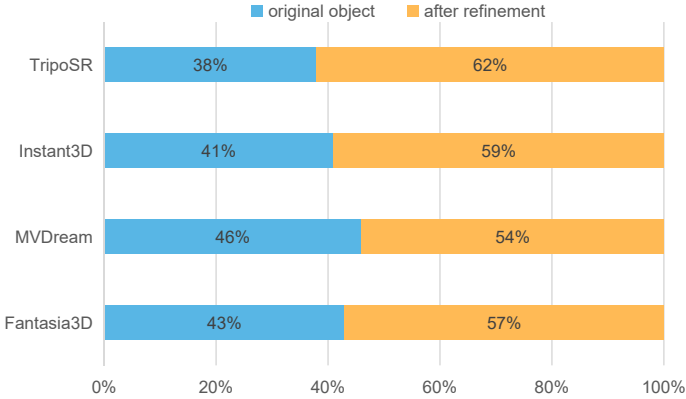


Fig. 16: User preference study of pre- and post-refinement of objects generated by edge-cutting 3D generation methods.

views of each object, both pre- and post-refinement, in a random sequence. Volunteers are instructed to select the version they preferred, based on texture quality and photo-realism. The statistical outcomes of this study are depicted in Fig. 16. It is observed that volunteers consistently favor the objects post-refinement across all evaluated 3D content creation methods. This preference aligns with the evaluations performed by GPT-4V, indicating a general consensus on the enhancement in quality achieved through Make-it-Real refinement process.

D.2 Effects of Different Texture Maps

We validate the impact of various texture maps generated by Make-it-Real on the appearance of 3D objects, as illustrated in Fig. 17. For each example, the first column lacks the corresponding map enhanced by Make-it-Real, while the second column includes the same conditions with the specific material map. Initially, we compare the effect of metalness on object appearance. We observe that the objects on the right, with a higher metalness map value, exhibit higher reflectivity, and the reflected light’s color is similar to the material’s own color, closely resembling real-world objects and appearing more aesthetically pleasing. In contrast, the objects on the left without the metalness map have lower reflectivity, and the reflected light tends to be white.

Next, we examine the impact of the roughness map, which controls the smoothness of the material’s surface. We observe that the silver chess piece on the right, with a low roughness texture map, becomes smooth, and together with metalness, produces a mirror-like reflection effect. On the other hand, the box on the left with a roughness map exhibits changes in light dispersion on its surface, with some areas showing highlights, while also adding and enriching scratch texture details on the surface.

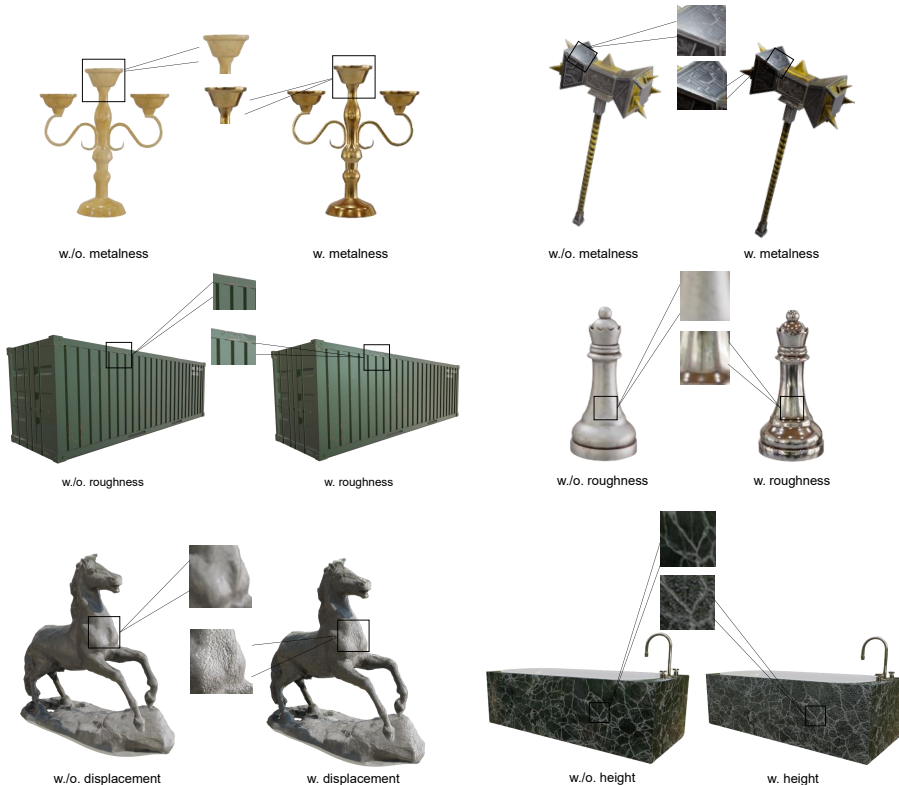


Fig. 17: Effects of different texture maps. We evaluate the effects of metalness, roughness, and displacement/height maps on the appearance of 3D objects.

Furthermore, we compare the effects of displacement and height on objects, both of which are usually optional and can also impact object appearance. Height maps typically use grayscale values to represent the surface’s relative height, simulating a relief effect through changes in lighting and shadows. As shown in the last row on the right, the object with a height map has an uneven surface, enhancing the sense of depth. Displacement mapping is a more powerful technique that changes the vertex positions of the geometry based on the map’s values, creating realistic relief effects. As shown on the left, the stone sculpture with displacement mapping exhibits very realistic details and a sense of relief.

E More Qualitive Results

E.1 Make-it-Real for Existing 3D Assets

In this section, we show more qualitative results of Make-it-Real for existing 3D assets from [15]. Results are shown in Fig. 18. The first row presents the original 3D object with only a diffuse map, while the second row showcases the object enhanced by our method with various material maps.

E.2 Make-it-Real for 3D Generation Models

In this section, we show more qualitative results of Make-it-Real for 3D generation models. We demonstrate the effectiveness of Make-it-Real for refining both image-to-3D and text-to-3D generated objects, as illustrated in Fig. 19 and Fig. 20, respectively.

E.3 Visualization of generated texture maps.

In this section, we show visualization of generated texture maps in Fig. 21. Our approach, guided by the query diffuse reference, produces material maps with distinct material partitions and maintains a distribution consistent with the diffuse map. As a result, the enhanced object exhibits both realism and texture consistency.

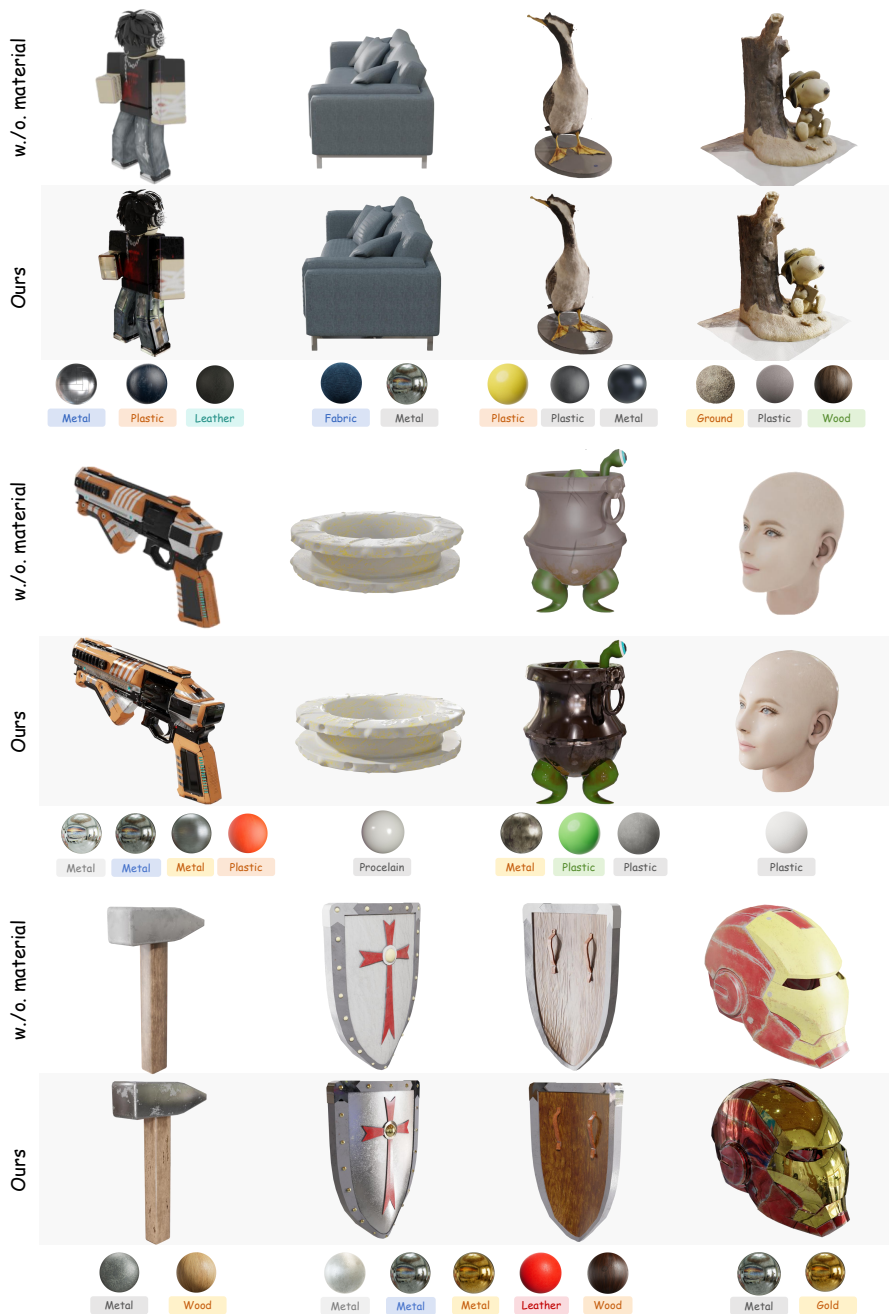


Fig. 18: More qualitative results of Make-it-Real refining existing 3D assets without material. Objects are selected from Objaverse [15] with diffuse only.

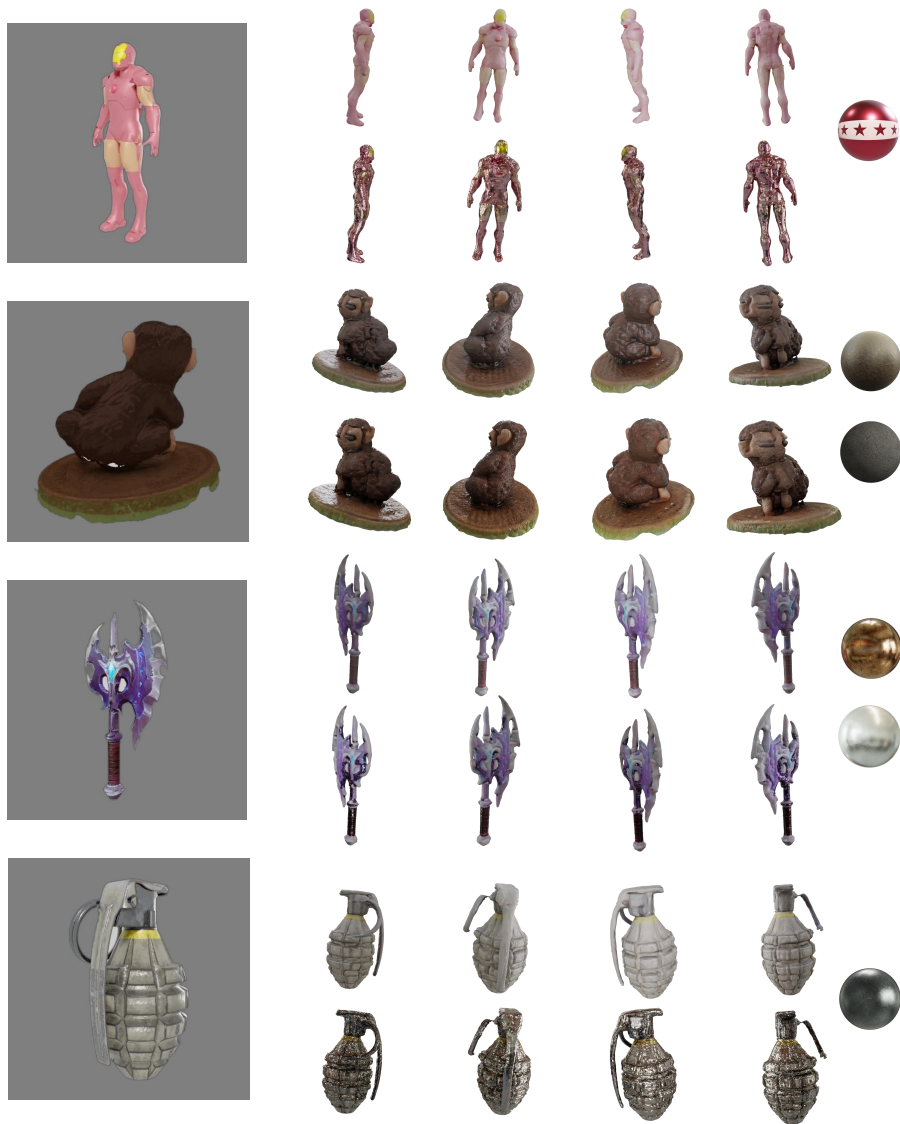


Fig. 19: More qualitative results of Make-it-Real refining objects generated by TripoSR [50]. Single View images are selected from Objaverse [15] rendered images. First line per two are original generated objects, Second lines are objects refined by Make-it-Real.

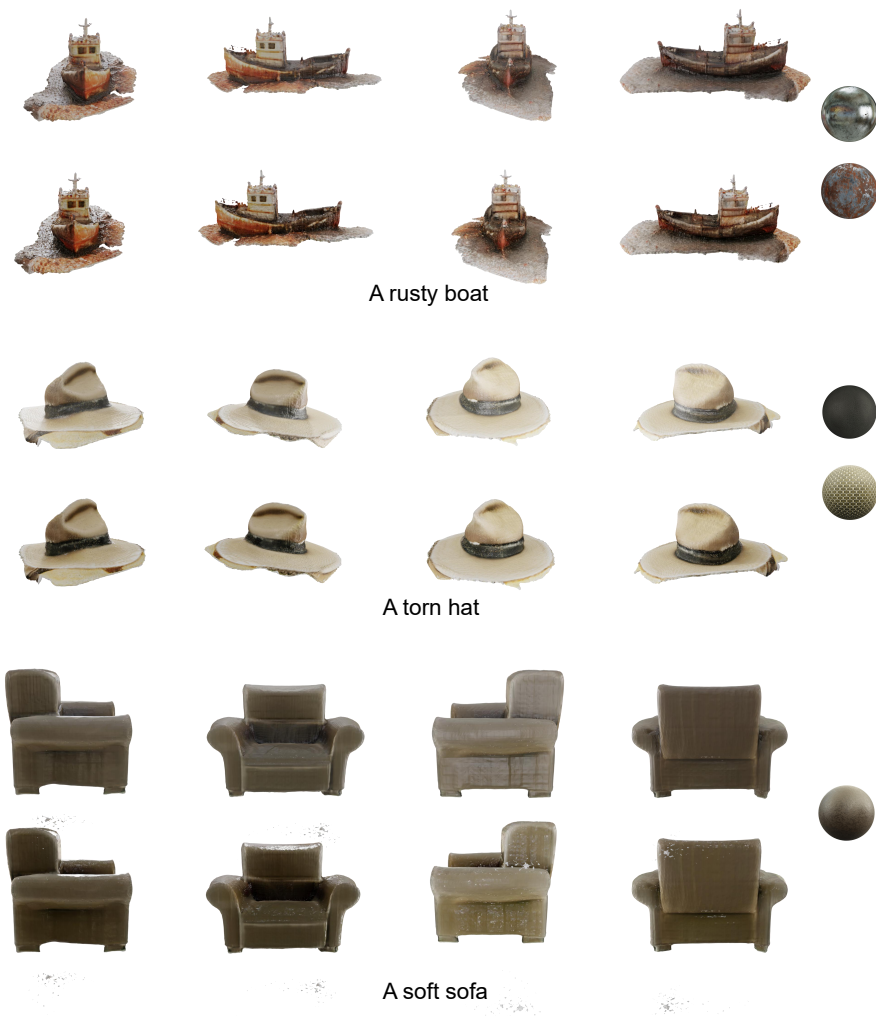


Fig. 20: More qualitative results of Make-it-Real refining objects generated by Instant3D [30]. Text prompts selected from GPTEval3D [56]. First line per two are original generated objects, Second lines are objects refined by Make-it-Real.

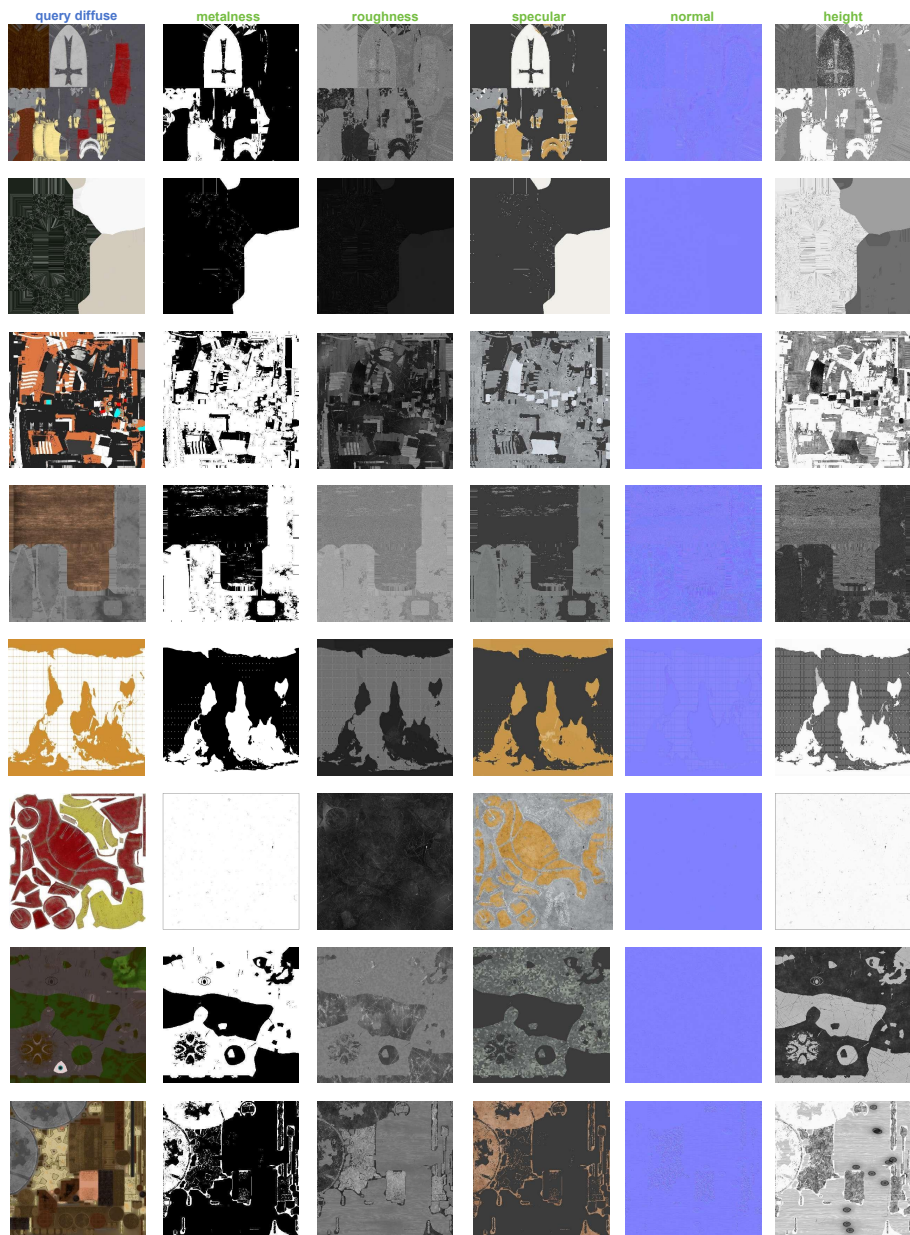


Fig. 21: Visualization of generated texture maps. The first column represents the original query diffuse map of 3D objects, while the subsequent columns showcase the corresponding material maps generated by Make-it-Real.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. ArXiv **abs/2204.14198** (2022), <https://api.semanticscholar.org/CorpusID:248476411> 4
2. Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A.T., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J., Shafey, L.E., Huang, Y., Meier-Hellstern, K.S., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G.H., Ahn, J., Austin, J., Barham, P., Botha, J.A., Bradbury, J., Brahma, S., Brooks, K.M., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C.A., Chowdhery, A., Crépý, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., D'iaz, M.C., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., García, X., Gehrmann, S., González, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A.R., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W.H., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M.L., Li, W., Li, Y., Li, J.Y., Lim, H., Lin, H., Liu, Z.Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A., Roy, A., Saeta, B., Samuel, R., Shelby, R.M., Slone, A., Smilkov, D., So, D.R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L.W., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., Wu, Y.: Palm 2 technical report. ArXiv **abs/2305.10403** (2023), <https://api.semanticscholar.org/CorpusID:258740735> 2, 4
3. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S.Y., Sagawa, S., Jitsev, J., Kornblith, S., Koh, P.W., Ilharco, G., Wortsman, M., Schmidt, L.: Openflamingo: An open-source framework for training large autoregressive vision-language models. ArXiv **abs/2308.01390** (2023), <https://api.semanticscholar.org/CorpusID:261043320> 4
4. Boss, M., Jampani, V., Kim, K., Lensch, H., Kautz, J.: Two-shot spatially-varying brdf and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3982–3991 (2020) 3, 4
5. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T.J., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. ArXiv **abs/2005.14165** (2020), <https://api.semanticscholar.org/CorpusID:218971783> 2, 4
6. Chan, E., Lin, C.Z., Chan, M., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 16102–16112 (2021), <https://api.semanticscholar.org/CorpusID:245144673> 2

7. Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., Lin, D.: Sharegpt4v: Improving large multi-modal models with better captions. arXiv preprint arXiv:2311.12793 (2023) [4](#)
8. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 22189–22199 (2023), <https://api.semanticscholar.org/CorpusID:257757213> [2](#)
9. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023) [2](#), [3](#), [4](#), [5](#), [11](#), [12](#), [19](#), [22](#)
10. Chen, X., Jiang, T., Song, J., Yang, J., Black, M.J., Geiger, A., Hilliges, O.: gdna: Towards generative detailed neural avatars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20427–20437 (2022) [2](#)
11. Chen, Y., Chen, R., Lei, J., Zhang, Y., Jia, K.: Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. Advances in Neural Information Processing Systems **35**, 30923–30936 (2022) [2](#), [4](#)
12. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N.M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B.C., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T.S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Díaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K.S., Eck, D., Dean, J., Petrov, S., Fiedel, N.: Palm: Scaling language modeling with pathways. J. Mach. Learn. Res. **24**, 240:1–240:113 (2022), <https://api.semanticscholar.org/CorpusID:247951931> [2](#), [4](#)
13. Cook, R.L., Torrance, K.E.: A reflectance model for computer graphics. ACM Transactions on Graphics (ToG) **1**(1), 7–24 (1982) [18](#)
14. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects (2022) [9](#), [10](#)
15. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. arXiv preprint arXiv:2212.08051 (2022) [19](#), [24](#), [26](#), [27](#)
16. Deschaintre, V., Aittala, M., Durand, F., Drettakis, G., Bousseau, A.: Single-image svbrdf capture with a rendering-aware deep network. ACM Transactions on Graphics (ToG) **37**(4), 1–15 (2018) [5](#)
17. Dong, X., Zhang, P., Zang, Y., Cao, Y., Wang, B., Ouyang, L., Wei, X., Zhang, S., Duan, H., Cao, M., Zhang, W., Li, Y., Yan, H., Gao, Y., Zhang, X., Li, W., Li, J., Chen, K., He, C., Zhang, X., Qiao, Y., Lin, D., Wang, J.: Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model (2024) [4](#)
18. Dong, Z., Chen, X., Yang, J., Black, M.J., Hilliges, O., Geiger, A.: Ag3d: Learning to generate 3d avatars from 2d image collections. arXiv preprint arXiv:2305.02312 (2023) [2](#)

19. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q.H., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., Florence, P.R.: Palm-e: An embodied multimodal language model. In: International Conference on Machine Learning (2023), <https://api.semanticscholar.org/CorpusID:257364842> 4
20. He, Z., Wang, T.: Openlm: Open-source large reconstruction models. <https://github.com/3DTopia/OpenLRM> (2023) 3
21. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L.A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J.W., Vinyals, O., Sifre, L.: Training compute-optimal large language models. ArXiv **abs/2203.15556** (2022), <https://api.semanticscholar.org/CorpusID:247778764> 2, 4
22. Hong, F., Tang, J., Cao, Z., Shi, M., Wu, T., Chen, Z., Wang, T., Pan, L., Lin, D., Liu, Z.: 3dtopia: Large text-to-3d generation model with hybrid diffusion priors (2024) 3
23. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d (2023) 3
24. Huang, S., Dong, L., Wang, W., Hao, Y., Singhal, S., Ma, S., Lv, T., Cui, L., Mohammed, O.K., Liu, Q., Aggarwal, K., Chi, Z., Bjorck, J., Chaudhary, V., Som, S., Song, X., Wei, F.: Language is not all you need: Aligning perception with language models. ArXiv **abs/2302.14045** (2023), <https://api.semanticscholar.org/CorpusID:257219775> 4
25. Jatavallabhula, K.M., Smith, E., Lafleche, J.F., Tsang, C.F., Rozantsev, A., Chen, W., Xiang, T., Lebedev, R., Fidler, S.: Kaolin: A pytorch library for accelerating 3d deep learning research (2019) 18
26. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023) 3
27. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/> 3
28. Labschütz, M., Krösl, K., Aquino, M., Grashöftl, F., Kohl, S.: Content creation for a 3d game with maya and unity 3d. Institute of Computer Graphics and Algorithms, Vienna University of Technology **6**, 124 (2011) 2
29. Li, F., Zhang, H., Sun, P., Zou, X., Liu, S., Yang, J., Li, C., Zhang, L., Gao, J.: Semantic-sam: Segment and recognize anything at any granularity (2023) 5, 7
30. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. <https://arxiv.org/abs/2311.06214> (2023) 2, 11, 19, 28
31. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model (2023) 3, 12
32. Li, J., Li, D., Savarese, S., Hoi, S.C.H.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. ArXiv **abs/2301.12597** (2023), <https://api.semanticscholar.org/CorpusID:256390509> 4
33. Li, J., Li, D., Xiong, C., Hoi, S.C.H.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International

- Conference on Machine Learning (2022), <https://api.semanticscholar.org/CorpusID:246411402> 4
34. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 300–309 (June 2023) 3
 35. Lopes, I., Pizzati, F., de Charette, R.: Material palette: Extraction of materials from a single image. arXiv preprint arXiv:2311.17060 (2023) 3, 4, 21, 22
 36. Martin, R., Roullier, A., Rouffet, R., Kaiser, A., Boubekeur, T.: Materia: Single image high-resolution material capture in the wild. In: Computer Graphics Forum. vol. 41, pp. 163–177. Wiley Online Library (2022) 5
 37. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. arXiv preprint arXiv:2211.07600 (2022) 3
 38. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM 65(1), 99–106 (2021) 3
 39. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022) 3
 40. OpenAI: Gpt-4v(ision) system card. OpenAI (2023), <https://api.semanticscholar.org/CorpusID:263218031> 4, 14
 41. OpenAI, R.: Gpt-4 technical report. arXiv pp. 2303–08774 (2023) 2, 4
 42. Park, K., Rematas, K., Farhadi, A., Seitz, S.M.: Photoshape: Photorealistic materials for large-scale shape collections. arXiv preprint arXiv:1809.09761 (2018) 21
 43. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022) 3
 44. Qi, Z., Fang, Y., Sun, Z., Wu, X., Wu, T., Wang, J., Lin, D., Zhao, H.: Gpt4point: A unified framework for point-language understanding and generation. arXiv preprint arXiv:2312.02980 (2023) 4
 45. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. arXiv preprint arXiv:2302.01721 (2023) 2, 18
 46. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv:2308.16512 (2023) 3, 11, 12, 19
 47. Sun, Z., Fang, Y., Wu, T., Zhang, P., Zang, Y., Kong, S., Xiong, Y., Lin, D., Wang, J.: Alpha-clip: A clip model focusing on wherever you want (2023) 4
 48. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024) 3
 49. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. ArXiv abs/2309.16653 (2023), <https://api.semanticscholar.org/CorpusID:263131552> 3
 50. Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.P.: Triposr: Fast 3d object reconstruction from a single image (2024) 2, 10, 11, 12, 19, 27
 51. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: Llama: Open and efficient foundation language models. ArXiv abs/2302.13971 (2023), <https://api.semanticscholar.org/CorpusID:257219404> 2, 4

52. Vecchio, G., Deschaintre, V.: Matsynth: A modern pbr materials dataset. arXiv preprint arXiv:2401.06056 (2024) [6](#), [18](#)
53. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. arXiv preprint arXiv:2212.00774 (2022) [3](#)
54. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. arXiv preprint arXiv:2305.16213 (2023) [3](#)
55. Wu, T., Li, Z., Yang, S., Zhang, P., Pan, X., Wang, J., Lin, D., Liu, Z.: Hyperdreamer: Hyper-realistic 3d content generation and editing from a single image (2023) [3](#)
56. Wu, T., Yang, G., Li, Z., Zhang, K., Liu, Z., Guibas, L., Lin, D., Wetzstein, G.: Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation. arXiv preprint arXiv:2401.04092 (2024) [4](#), [12](#), [19](#), [28](#)
57. Xu, X., Lyu, Z., Pan, X., Dai, B.: Matlabe: Material-aware text-to-3d via latent brdf auto-encoder. arXiv preprint arXiv:2308.09278 (2023) [2](#), [4](#)
58. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., Zhang, K.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model (2023) [3](#)
59. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v (2023) [2](#), [5](#), [7](#), [8](#)
60. Youwang, K., Oh, T.H., Pons-Moll, G.: Paint-it: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. arXiv preprint arXiv:2312.11360 (2023) [2](#), [4](#), [5](#)
61. Zhang, P., Dong, X., Wang, B., Cao, Y., Xu, C., Ouyang, L., Zhao, Z., Duan, H., Zhang, S., Ding, S., Zhang, W., Yan, H., Zhang, X., Li, W., Li, J., Chen, K., He, C., Zhang, X., Qiao, Y., Lin, D., Wang, J.: Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition (2023) [4](#)