Aladdin: Zero-Shot Hallucination of Stylized 3D Assets from Abstract Scene Descriptions

IAN HUANG, Stanford University, USA
VRISHAB KRISHNA, Stanford University, USA
OMORUYI ATEKHA, Stanford University, USA
LEONIDAS GUIBAS, Stanford University, USA

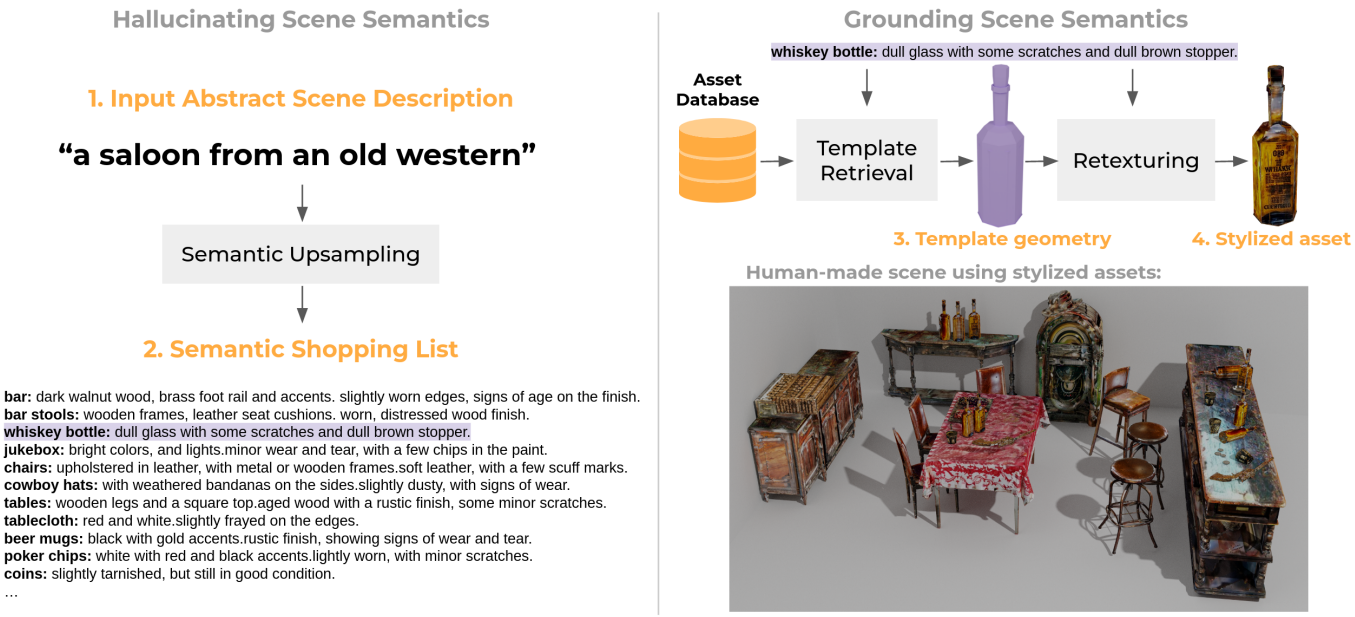

Fig. 1. Our system produces stylized assets to fit a scene description. Given an abstract scene description that does not provide details on what objects should be found within that scene, our system (1) infers a *semantic shopping list*, a human-readable and editable list of object categories and appearance attributes, and then uses this to (2) retrieve template shapes from a 3D asset database before (3) re-texturing them to fit the desired appearance attributes. The output of our system is a collection of textured meshes, which can be directly imported into 3D design software and used for other downstream tasks. Note the correspondences in the assets on the right with many of the desired object categories and appearance attributes generated by our system on the left!

What constitutes the “vibe” of a particular scene? What should one find in “a busy, dirty city street”, “an idyllic countryside”, or “a crime scene in an abandoned living room”? The translation from abstract scene descriptions to stylized scene elements cannot be done with any generality by extant systems trained on rigid and limited indoor datasets. In this paper, we propose to leverage the knowledge captured by foundation models to accomplish this translation. We present a system that can serve as a tool to generate stylized assets for 3D scenes described by a short phrase, without the need to enumerate the objects to be found within the scene or give instructions on their appearance. Additionally, it is robust to open-world concepts in a way that traditional methods trained on limited data are not, affording more creative freedom to the 3D artist. Our system demonstrates this using a foundation model “team” composed of a large language model, a vision-language model and several image diffusion models, which communicate using an interpretable and user-editable intermediate representation, thus allowing for more versatile and controllable stylized asset generation for 3D artists. We introduce novel metrics for this task, and show through human evaluations that in 91% of the cases, our system outputs are judged more faithful to the semantics of the input scene description than the baseline, thus highlighting the potential of this approach to radically accelerate the 3D content creation process for 3D artists.

CCS Concepts: • **Applied computing** → *Media arts.*

Additional Key Words and Phrases: Large Language Models, Foundation Models, Texture Generation, Scene Descriptions, Asset Retrieval

Authors’ addresses: Ian Huang, Stanford University, Stanford, CA, USA, ianhuang@cs.stanford.edu; Vrishab Krishna, Stanford University, Stanford, CA, USA, vrishab@stanford.edu; Omoruyi Atekha, Stanford University, Stanford, CA, USA, oatekha@stanford.edu; Leonidas Guibas, Stanford University, Stanford, CA, USA, guibas@cs.stanford.edu.

0730-0301/2023/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Ian Huang, Vrishab Krishna, Omoruyi Atekha, and Leonidas Guibas. 2023. Aladdin: Zero-Shot Hallucination of Stylized 3D Assets from Abstract Scene Descriptions. *ACM Trans. Graph.* 1, 1 (May 2023), 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

While language-to-shape generation has taken the world by storm, scene generation has been less accessible by language control partially because the language content needed to express details of a scene becomes prohibitively cumbersome for average human creators. Additionally, manually searching, selecting, and retexturing/restylizing assets from online 3D repositories is, in aggregate, a very time-consuming task for a single object, not to mention the task of doing so for 30-50 objects that may make up a 3D scene.

From a user's perspective, wouldn't it be convenient to say "generate me a scene of the financial district of New York" and have the system infer *what* should be in the scene and *how* every item should look? In other words, we would like to build a system that hallucinates both semantic and visual detail from an *abstract* high-level scene description – something that human users can provide much more conveniently than fully enumerative language found in [Achlioptas et al. 2020; Chang et al. 2015b; Ilinykh et al. 2019].

While valuable, this is not a problem that can be solved using traditional machine learning approaches, primarily due to limitations in data – the indoor scene datasets that dominate the domain of scene generation are largely limited in scene and object diversity [Chang et al. 2017; Fu et al. 2021a; Roberts et al. 2021; Song et al. 2015, 2017], far from open-vocabulary. The same can be said for language-scene multimodal datasets, where language labels are either limited or prohibitively enumerative to train for our primary task of interest [Achlioptas et al. 2020; Chang et al. 2015b].

In this paper, we ask, how far can zero-shot inference using foundation models go, with their common sense understanding [Brown et al. 2020; Radford et al. 2021; Rombach et al. 2022], in facilitating the 3D scene creation process for 3D artists? We introduce a system that allows 3D content creators to synthesize entire asset collections from abstract scene descriptions (e.g. "a busy city street"), by leveraging the immense amount of progress in Large Language Models and Vision Language Models.

To go from abstract description to stylized asset collections, we break the process into 3 stages. In the first stage, we "semantically upsample" the input abstract description into a plausible list of objects, attributes and appearances (which we call the "semantic shopping list") that may compose the described scene. For this, we use in-context prompting of LLM's [Brown et al. 2020], exploiting common sense knowledge of scene composition embedded within LLM's. The second stage requires a retrieval from an existing 3D asset database, given the attributes and appearances hallucinated in the semantic shopping list. We use visual and textual similarity given by large vision-language models like CLIP to retrieve top candidates. Finally, we use diffusion models to texture the surface of the objects given their hallucinated appearance attributes.

Our system uses natural language as an intermediary representation between these stages, for 3 reasons: **(1) Interpretability and Editability**: This means that users can visualize, interpret and edit the intermediary outputs. This is important, since this work employs a "team" of foundation models for the first time, where the output of one may not necessarily – in a zero-shot sense – be optimal as an input into another to accomplish the user's artistic intent. **(2) Varying Abtraction Levels**: Given language's ability to represent information at a variety of abstraction levels, it as a medium that allows both the large language model (as well as user edits) to specify semantic constraints at a wide range of specificity. **(3) Moore's law, but for foundation models**: Given recent trends, we're anticipating that the foundation models used in this paper will have more powerful replacements soon. We expect that users of our system will be able to "upgrade" different modules with the latest models.

Our system integrates with existing 3D asset databases and treats its assets as templates for both the appearance and the geometry. The benefits of this are two-fold: (1) while large amounts of work that does scene generation is reliant and restricted on indoor scene datasets [Chang et al. 2015b; Ma et al. 2018; Paschalidou et al. 2021], our method can generate outdoor scenes and radically out-of-distribution scenes as well, by leveraging diverse and larger-scale shape databases [Chang et al. 2015a; Deitke et al. 2022; Selvaraju et al. 2021] and (2) building ontop of a 3D asset store allows usage of such a system to be specialized, depending on the asset store provided, not to mention that it allows for nice priors, important for both geometric and textural manipulation [Hui et al. 2022; Michel et al. 2022,?; Xu et al. 2022].

The main contribution of this paper is three-fold: (1) we present the task of *stylized asset curation* given *abstract scene descriptions*, which, to the knowledge of the authors, has not been considered in isolation. (2) we present a system that tackles this task using the zero-shot capabilities of foundation models, and contribute a method that does this using semantic upsampling through in-context learning. (3) We introduce a new metric, CLIP-D/S, which can be used to measure both the diversity of the asset collection and the semantic alignment with respect to a target scene description. In addition to quantitative and qualitative evaluations, our human evaluation experiments conducted using 72 evaluators showcases the efficacy of our system, and the value of semantic upsampling as a key powerhorse in the quality of the generated assets and assembled scenes.

2 RELATED WORKS

Works like [Achlioptas et al. 2022; Fu et al. 2022; Gao et al. 2022; Huang et al. 2022; Jain et al. 2022; Jun and Nichol 2023; Lin et al. 2022; Michel et al. 2022; Nichol et al. 2022; Poole et al. 2022; Sanghi et al. 2022; Xu et al. 2022] focus on generating shapes from natural language. However, as many of them use non-mesh-based 3D representations like implicit representations [Jain et al. 2022; Lin et al. 2022; Poole et al. 2022; Xu et al. 2022], extracting meshes from them gives rise to disruptive artifacts in both texture and geometry, limiting the usability of the asset in almost all 3D design applications. As such their outputs are not optimized for usage by human users, since composing and editing scenes using implicit representations of assets remains non-trivial. Additionally, such systems are not optimized to read between the lines – the desired output is oftentimes what is described verbatim, given its object-centric

focus. However, for abstract scene descriptions, compositional understanding beyond what is typically captured by vision-language models is needed.

On the other hand, works on mesh generation and texturing using text prompts [Michel et al. 2022; Sanghi et al. 2022; Xu et al. 2022] make use of vision-language models like CLIP [Radford et al. 2021] coupled with differentiable rendering to optimize the mesh to correspond to a certain text embedding. These methods manage to edit the mesh to become semantically similar to the text prompt, but since CLIP was not directly optimized to guide differentiable rendering, the resulting optimization often leads to improbable or unrealistic outputs, as can be seen by disruptive artifacts that often give a distorted and blocky feel to the outputs. Meanwhile, through newer generative models, the world knowledge obtained from large-scale image-text datasets is easily accessible. [Lin et al. 2022] takes a step in this direction, using image diffusion models to generate high resolution textures of a mesh. However, this requires a detailed description specific to the individual objects to be generated, which is not provided a priori in our problem setting.

Along this line, works like [Fridman et al. 2023], [Höllein et al. 2023], and [Zhang et al. 2023] introduce pipelines that incorporate image diffusion models [Ho et al. 2020; Rombach et al. 2022; Saharia et al. 2022] to create *scenes*, but not in a way that allows assets that compose the scene to be easily and effectively extracted. Scenescape [Fridman et al. 2023] and Text2Room [Höllein et al. 2023] are two similar methods that make use of depth prediction models to craft a mesh using iterative predictions from a image diffusion model. SceneScape [Fridman et al. 2023], making use of generated super-resolution videos, is biased towards producing scenes that are long and tunnel-like, thereby restricting the set of producible scenes. Similarly, Text2Room [Höllein et al. 2023] can only generate closed, star-convex meshes due to the depth projection approach. The fundamental drawback of these methods is that the end result is a single connected mesh with limited flexibility to extract and edit assets. Meanwhile, [Po and Wetzstein 2023] recently introduced a model that uses locally conditioned diffusion to generate the scene compositionally by using different language instructions to generate different patches of the scene (e.g. "a firepit" in one part of the scene, "a tent" in another). However, not only do the assets generated suffer from the same aforementioned weaknesses in regards to mesh extraction, but the generative pipeline also requires fully enumerative language input, in contrast with the focus of this work.

Although considerate effort has been made towards collections of 3D scene datasets [Chang et al. 2017; Fu et al. 2021a; Roberts et al. 2021; Song et al. 2015, 2017] as well as training models to generate and position elements within indoor scenes [Chang et al. 2015b; Ma et al. 2018; Paschalidou et al. 2021; Ritchie et al. 2019; Wang et al. 2019, 2021], these were not designed to handle open vocabularies of objects, which makes them limited for creative applications. Moreover, the latter works do not have the ability to re-texture scene elements to better match the input language description, which is a prime focus of our system.

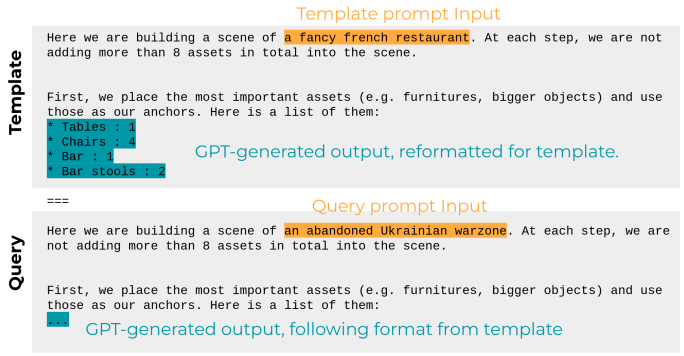

Fig. 2. The template and query segments of the GPT-3 input share the same structure up to the part where GPT-3 is prompted to do next-token prediction. We use this template for all generations of the anchor objects within the scene.

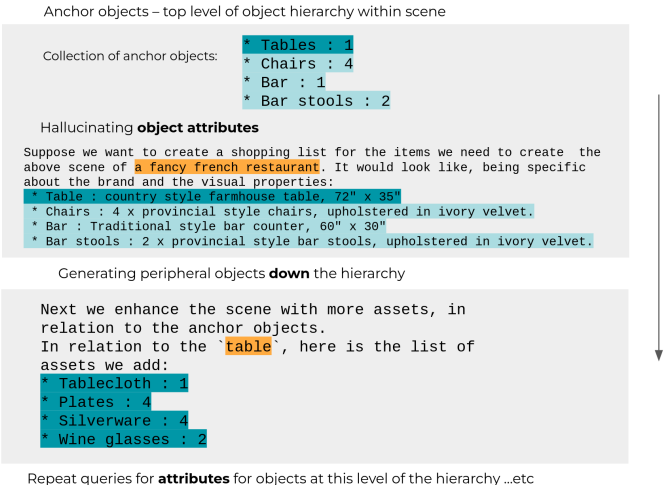

Fig. 3. We move down the scene hierarchy by asking GPT-3 to generate peripheral objects around each of the objects in the current level. Usually, this results in smaller and more peripheral objects that add to the realism of the scene. We use in-context learning again to generate their attributes.

3 OUR METHOD

3.1 Semantic Upsampling

Given an abstract scene description, our system "upsamples" the semantics of the scene description to the level of object categories, properties and appearance. To do this, we use few-shot prompting of GPT-3 [Brown et al. 2020], which has shown to be very useful in other settings [Chen et al. 2022; Dong et al. 2022; Min et al. 2021; Rubin et al. 2021; Shin et al. 2022; Wang et al. 2022; Wei et al. 2021, 2022; Zhang et al. 2022; Zhao et al. 2021; Zhou et al. 2022].

To do this, we create templates that cover a variety of different aspects of objects that may be found within the scene; object category, style, material properties, and condition (e.g. scratched, unused, rusted, ...). These templates can be found in the Appendix. Templates are used for two main reasons: (1) they effectively enforce a prior over the kind of attributes that one would like to use to describe objects within the scene and (2) they dictate a textual

format that can be very easily parseable by our system (e.g. comma separated attributes, colon separation between object category and attributes.).

In practice, we found that querying for *all* the objects within a scene at once can lead to degenerate results – generating details for way too many objects at once may cause the objects chosen to "drift" semantically away from the prompt. As such, we adopt a more hierarchical approach, where we first use in-context learning to ask GPT-3 [Brown et al. 2020] to generate a set of "anchor" objects (typically, this is a small set of 6-8 objects) and their attributes (Figure 2). For each of these anchor objects, we ask it to hallucinate objects (and their attributes) found "around" the anchor object (Figure 3), and repeating this recursively down the hierarchy. This works fairly well to elucidate the hierarchy of objects, and can be useful for object placement (e.g. for a "fancy french restaurant", an anchor object generated is a table, and objects generated *around* this anchor object are objects typically found *on* the table). Additionally, doing this hierarchically means that for abstract descriptions that involve a large set of objects, we need only call this procedure a few times before we arrive at the "leaf" objects within the implicit object hierarchy. A traversal through this hierarchy allows a full list of objects and appearance attributes of objects likely to be found within the described scene. We will refer to this list as the *semantic shopping list*. An example of this is shown on the left in Figure 1.

3.2 Object retrieval & retexturing

Given the semantic shopping list from semantic upsampling, the system use CLIP [Radford et al. 2021] embeddings of both visual renderings and textual annotations of objects within asset databases to retrieve the template geometries for each object.

This is, however, a nuanced objective; since all objects selected during retrieval will go through diffusion-based *re*-texturing, it's tempting to disregard the original texturing altogether, and retrieve only using a query composed of the object category information from the semantic upsampling (ignoring object attributes, which will be "painted" on in a later stage). In practice, this leads to suboptimal retrieval results. Some object attributes (e.g. "old" in "old car") are less solely based in texture, affecting both the visual appearance and the geometry. Moreover, the pretrained model of CLIP was trained on natural images, which relies on color properties for accurate similarity evaluation (similar observations have also been reported in [Michel et al. 2022]). As such, using a textureless rendering of the candidate asset can actually *hurt* the retrieval performance.

To match the open-world vocabulary found in semantic shopping lists, it is essential to have a large and diverse asset database to choose from. For this paper, we've chosen to use a combination of Future3D [Fu et al. 2021b] and a 300K-subset of Objaverse [Deitke et al. 2022]. Future3D specializes in objects commonly found in indoor environments, and is a useful dataset for the majority of "base" object found within indoor scenes, which we anticipate would make up of the majority of user scene queries. Objaverse is a lot more diverse in object category, and serves for the "personality" pieces of indoor scenes (e.g. the sword along the wall in Figure 4), which allows the scene to be more faithful to the "vibe" communicated in the input description. Additionally, it contains object categories typically found outdoors, which allows our system to construct outdoor scenes in ways that previous scene generation pipelines cannot (see Figure 10).

In our current implementation, we use the thumbnails of different assets to derive the CLIP image embeddings, since (1) these are readily available in most datasets and (2) human artists already use them to judge the appropriateness of a particular asset for their scene. Future works extending our pipeline can use more complex rendering techniques for different objects, and the question of how renderings should be done to encourage high accuracy 3D asset retrieval is an important direction for future work.

Enforcing stylistic consistency from the retrieval stage is hard. Empirically, we notice that using just the semantic shopping list alone often leads to retrieval of objects that are stylistically inconsistent in their template geometry, and thus do not aesthetically combine well once put in the same scene. This is because though semantic upsampling hallucinates visual details, it has no context of what would be important for stylistic consistency in the retrieved results *downstream*. Therefore, we merge the abstract scene description into all retrieval and texturing queries, for all objects, as a fail-safe for when the semantic shopping list provides inadequate stylistic information.

Given that many 3D assets have language annotations, we incorporate that information when determining the K-nearest neighbors through a simple linear weighting of the language- and image-based cosine similarities. Doing brings some more robustness to the retrieval process in the case when the asset thumbnail does not reflect the geometric content as well as its textual annotations do.

Once we have the template objects, we make use of pre-existing image generation pipelines to texture each retrieved object. Using an available depth-guided and language-guided image diffusion models [Rombach et al. 2022], we can generate images corresponding to views of an object and use differentiable rendering to optimize our mesh texture to match the generated image, while encouraging 3D consistency between different views through depth and language conditioning. We use the implementation of a recent paper [Richardson et al. 2023] to achieve this.

4 EXPERIMENTS

The main output of our system is a set of textured assets. To demonstrate the usability of our system outputs, we source ideas for input scene descriptions from 8 people who do not have any prior 3D design experience or experience interacting with our system. 20 such prompts were collected, ranging in *plausibility* (from "a romantic french restaurant" to "a church for strawberries"), *emotional valence* (from "a marvel-themed bedroom for a five-year-old toddler" to "murder in an abandoned living room") and *complexity* (from "a rustic backyard in the countryside" to "a busy street in downtown new york"). A full collection of the abstract scene descriptions can be found in the Appendix, as well as their corresponding visualizations.

We provide the prompts to our system, and – for the purposes of this paper – run our system in a fully automated way, sidestepping the possible option of user edits of the semantic shopping list between different stages. To do this, we use the same query string (generated from the semantic upsampling stage) for the retrieval

and texturing stages, and automatically select the top-1 sample in CLIP-Similarity in the retrieval outputs for texturing.

Note that to demonstrate the robustness of this system and the benefit of basing it on foundation models, we *do not cherrypick between different runs for the same input prompt.* In other words, all visualizations of scenes are done based on assets generated in a single pass.

4.1 Composing assets into scenes

To construct the final scene, three of the authors of this paper import the generated assets into Blender [Community 2018] and create 3D scenes according to the following rules: (1) they are allowed to translate, rotate and scale any 3D asset in the generated collection along any axis, (2) they are allowed to add ground and wall planes to the scene, (3) they are allowed to omit subsets of the asset collection from visualization, (4) they are allowed to duplicate assets as many times as they wish, and (5) they are *not* allowed to change the material properties of the textured mesh, except emissive properties for assets that should emit light (rare). On average, the importing, arrangement and rendering of a single scene took 20 minutes.

4.2 Evaluating stylized asset collections

Within the literature, CLIP-Similarity (or CLIP-S) has been recently used to measure adherence of generative output to the semantics of the input text [Fu et al. 2022; Xu et al. 2022]. Given a vision encoder v and a language encoder g, rendered views of the object x_i and associated language description l, CLIP-S is defined as:

$$S(x, l) = \max_i v(x_i)^T g(l) \tag{1}$$

However, using CLIP-S directly on our task has serious drawbacks. First, it's an observed phenomenon that CLIP's language model oftentimes behaves like a Bag-Of-Words model [Michel et al. 2022; Yuksekgonul et al. 2022], where important relations between entities or concepts are often not reflected in its similarity evaluations. This motivates why it's inappropriate to use such a metric to evaluate the adherence of a stylized asset to *the set of objects that likely composes* a scene of a particular semantic – the relationships that are key to the idea of *scene membership* (i.e. that an asset belongs to a scene) can be overpowered by the description of the scene itself. Empirically, we've found that such a metric tends to slightly favor the outputs of the system when semantic upsampling is *not used* and assets are retrieved and textured according to the abstract scene description, though this comes as little surprise. The distinction is demonstrated in Figure 6, which shows that CLIP (and by extension, CLIP-S) cannot favor *assets that compose a scene* over assets that may resemble the abstract prompt but do not compose that scene. We would like a metric that exhibits this behavior.

To solve this problem, we introduce the idea of CLIP-Diversity (CLIP-D), a score that is high when the assets that is generated are semantically varied. This metric counteracts the favoring of systems that generate assets that are very narrowly aligned with the scene description. Assuming the same visual encoder v, and that x_i^j is the renderings of asset j from angle i, as well as a function m that averages on the surface of the unit sphere over a set of points on the unit hypersphere of the CLIP embedding space, we define CLIP-Diversity as the *negative* mean pairwise cossine similarity between assets within the collection:

Fig. 4. Asset arrangement from the **first** run of our system for the input "office of the King's Hand in Game of Thrones".

$$D(\{x^j\}_{j=1\ldots N}) = \\ -\frac{2}{N(N-1)} \sum_{i<j} m\left(\{v(x_k^j)\}_{k=1\ldots K}\right)^T m\left(\{v(x_k^i)\}_{k=1\ldots K}\right) \tag{2}$$

However, diversity alone does not provide adherence to a language instruction and can be satisfied without consideration for the language prompt. As such, we construct CLIP-D/S, a metric that additively combines CLIP-D and CLIP-S over an asset collection $\{x^j\}$ and a language instruction l from a collection L of augmented utterances based on the scene description (see Section 4.3), given equal weighting:

$$DS(\{x^j\}_{j=1\ldots N}, l) = D(\{x^j\}_{j=1\ldots N}) + \frac{1}{N|L|} \sum_j \sum_{l \in L} S(x^j, l) \tag{3}$$

CLIP-D/S is therefore a combination of diversity and similarity, and can be used heuristically to measure the fidelity and usefulness of the asset collections that our system generates.

4.3 System outputs

Figures 1, 4, 5, 8, 9, 10, and 11 show some outputs from our system, arranged into 3D scenes. For Figures 8, 9, 10 and 11, please go to the full-page figures after the References. A longer list of examples, along with their corresponding semantic shopping lists, can be found in the Appendix. Predominantly, the benefit of this system is its ability to add "character" to a scene through inferring a wider, more diverse set of object categories.

A practical property of this system is that due to the inherent randomness present in next-token generation of GPT-3, running the system twice will create differing semantic shopping lists. This is a useful property for 3D artists, since this allows them to semantically densify their scenes by rerunning the semantic upsampling process. As shown in Figures 4 and 5, multiple runs can come up with different but valid assets, where the union or intersection of them could create even richer and accurate scenes.

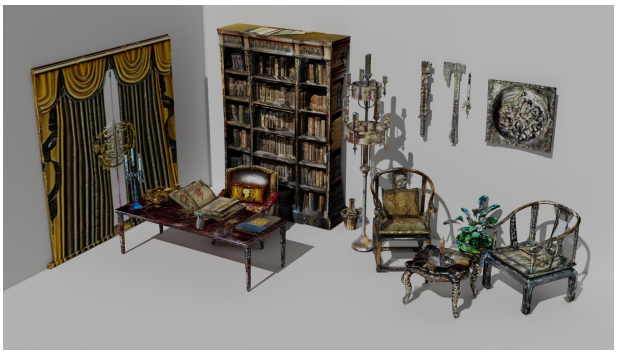

Fig. 5. Asset arrangement from the **second** run of our system for the input "office of the King's Hand in Game of Thrones".

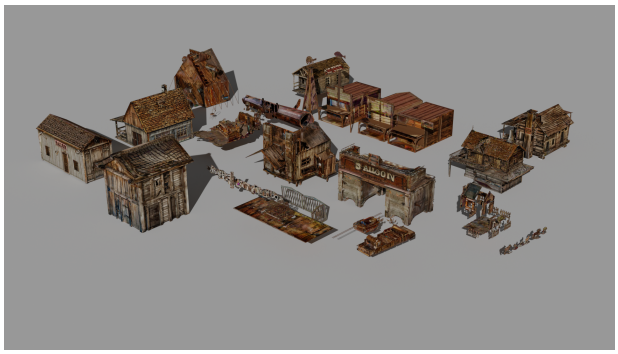

Fig. 6. Western saloon retrieval and retexturing *without* the use of semantic upsampling. What gets retrieved can be very narrowly aligned with the "western saloon" concept, but are not elements that can compose the scene described, unlike the assets in Figure 1.

As a measure of stylistic adherence, we alternatively ask, given an asset we've generated for each of the scenes, how well can one predict which scene they were generated for? We use the CLIP-S metric as a zero-shot classifier to classify each of our 572 stylized assets across the 20 scenes that they were generated for (full list can be seen in Table 1 and in Appendix), and find a classification accuracy of **32.69%**, substantially higher than the accuracy of guessing randomly (5%). We consider the predicted scene to be the scene that maximizes the average CLIP-S score across L, which is a set of language augmentations on the abstract scene description: (1) "an element in a scene of `[SCENE DESCRIPTION]`", (2) "an object from a scene of `[SCENE DESCRIPTION]`", (3) "a picture of an object form `[SCENE DESCRIPTION]`", (4) "a rendering of an asset from a 3D scene of `[SCENE DESCRIPTION]`" and (5) "`[SCENE DESCRIPTION]`".

4.4 The importance of semantic upsampling

The main contribution of our work is the use of in-context learning to generate semantically meaningful details as they pertain to assets. How important is this step? We compare against a baseline method that is exactly the same as the method proposed, *except* that it retrieve and retextures according to the input abstract scene description, instead of the semantic shopping list (composed of object category and attributes) given by semantic upsampling. For this, we retrieve and retexture the top-K assets that have the highest CLIP-similarity with the abstract scene description , where K is the number of assets generated by our method.

Table 1 shows the impact of removing semantic upsampling on the diversity (CLIP-D) and the CLIP-D/S score for the assets generated for each of the 20 scenes. This corroborates the observation that in the best case, as shown in Figure 6, assets that align very well with the scene itself might get retrieved, resulting a narrow selection that cannot be used to compose the scene. Or, as is often the case, an erroneous template shape is retrieved, and confuses the downstream retexturing to produce poorly textured 3D assets. Please see the Appendix for examples of this phenomenon.

4.5 The importance of retexturing

Given the ever-growing 3D asset collections, what is the benefit of replacing pre-fabricated textures using the last step of our system? Table 2 reflects what happens to the CLIP-S score (w.r.t. the abstract scene description) when we use the original texture, compared to that of our re-textured objects. This shows that in general, retexturing using the output from semantic upsampling allows an increase in visual similarity with respect to the abstract description of the *whole* scene. An example of this can be seen in Figure 7.

Scene Reference	D (b)	D (o) ↑	D/S (b)	D/S (o) ↑
rustic backyard	-0.84	**-0.80**	-0.61	**-0.60**
futuristic teahouse	-0.89	**-0.81**	-0.67	**-0.62**
confucius bedroom	-0.86	**-0.81**	-0.61	**-0.58**
alien teagarden	-0.84	**-0.80**	-0.63	**-0.60**
retro arcade	-0.84	**-0.79**	-0.59	**-0.56**
anne frank room	-0.87	**-0.80**	-0.64	**-0.57**
hades cave	-0.85	**-0.76**	-0.63	**-0.56**
shrek home	-0.86	**-0.78**	-0.59	**-0.53**
smurf house	-0.88	**-0.77**	-0.65	**-0.56**
mad scientist restaurant	-0.81	**-0.78**	-0.61	**-0.60**
western saloon	-0.82	**-0.79**	-0.60	**-0.59**
occult cult	-0.82	**-0.79**	-0.61	**-0.59**
marvel bedroom	-0.91	**-0.87**	-0.63	**-0.59**
murder room	-0.85	**-0.77**	-0.62	**-0.57**
strawberry church	-0.84	**-0.79**	-0.58	**-0.57**
poseidon living room	-0.83	**-0.77**	-0.61	**-0.55**
north korean classroom	-0.85	**-0.77**	-0.62	**-0.57**
antichrist vatican	-0.82	**-0.77**	-0.60	**-0.57**
romantic restaurant	-0.81	**-0.78**	-0.59	-0.59
busy new york street	-0.79	-0.79	**-0.62**	-0.63

Table 1. A comparison of CLIP-D (abbreviated D) and CLIP-D/S (abbreviated D/S) for all 20 scenes created using assets generated by our method (abbreviated **o**) and those generated by a baseline method (abbreviated **b**), which does not use semantic upsampling. Our method generally produces both higher performance in both CLIP-D (diversity) and CLIP-D/S.

4.6 User study

The metrics used to evaluate our method thus far are heuristical. To gauge the true value of our system, we conduct a user study composed of 72 human evaluators, across 11 randomly selected scenes in the full list of 20, for a total of 792 annotations. Each human

Fig. 7. The original Future3D asset retrieved and the same asset retextured by our system for the scene described by "a marvel-themed bedroom of a five-year old toddler". The effective texturing prompt created by semantic upsampling is "chair, in a scene of a marvel-themed bedroom for a five-year-old toddler, red and blue colors. no signs of wear and tear, firm supporting cushions." Note how the semantic adherence to the scene description increases after retexturing!

Scene Reference	Orig. ↑	Retextured ↑	% Improved
rustic backyard	0.17	**0.20**	80.00
futuristic teahouse	**0.21**	0.19	28.00
confucius bedroom	0.23	0.23	63.33
alien teagarden	0.18	**0.20**	70.59
retro arcade	0.21	**0.23**	66.67
anne frank room	0.18	**0.23**	85.19
hades cave	0.20	**0.20**	74.29
shrek home	0.18	**0.25**	93.55
smurf house	0.18	**0.21**	85.00
mad scientist restaurant	**0.19**	0.18	44.74
western saloon	0.16	**0.19**	81.82
occult cult	0.20	0.20	48.28
marvel bedroom	0.23	**0.27**	84.38
murder room	0.20	**0.21**	60.00
strawberry church	0.20	**0.22**	78.26
poseidon living room	0.20	**0.21**	56.00
north korean classroom	0.17	**0.21**	85.00
antichrist vatican	0.18	**0.20**	73.91
romantic restaurant	0.17	**0.20**	76.60
busy new york street	0.15	**0.16**	68.75

Table 2. The mean CLIP-S scores of generated asset collections w.r.t. their abstract scene description, with (Retextured) and without (Orig.) the retexturing using the semantic shopping lists. "% Improve" indicates the percentage of assets in the collection whose CLIP-S scores increased after retexturing. This shows that the retexturing is a valuable step of the pipeline to return assets that are more aligned with the scene semantics.

evaluators are first shown two options: (1) a scene rendering composed of assets generated by our system and (2) a scene rendering composed of assets generated by the baseline system (see Section 4.4). To decouple semantic alignment of the composed scene from the quality and diversity of the assets themselves, each evaluator is then asked two questions: (1) which arrangement of 3D assets is more accurate/faithful to the scene description? (2) If you were a 3D artist, which group of assets would you use to create a scene that matches the scene description? (Considering diversity, quality ...etc). Evaluators can only select one of the two options for each question. Please see the Appendix for the images shown to the human evaluators.

Scene reference	Q1(base)	Q1(our)	Q2(base)	Q2(our)
poseidon living room	25 %	**75%**	23.6%	**76.4%**
romantic restaurant	9.7%	**90.3%**	19.4%	**80.6%**
retro arcade	6.9 %	**93.1%**	5.6%	**94.4%**
anne frank room	31.9%	**68.1%**	22.2%	**77.8%**
smurf house	18.1%	**81.9%**	25%	**75%**
murder room	4.2%	**95.8%**	9.7%	**90.3%**
shrek home	9.7%	**90.3%**	12.5%	**87.5%**
confucius bedroom	**63.9%**	36.1%	**59.7%**	40.3%
marvel bedroom	16.7%	**83.3%**	25%	**75%**
futuristic teahouse	48.6%	**51.4%**	50%	50%
western saloon	12.5%	**87.5%**	19.4%	**80.6%**

Table 3. The percentage of human evaluators who selected each option for the two questions. **Q1** indicates the first question: "Which arrangement of 3D assets is more accurate/faithful to the scene description?" **Q2** indicates the second question: "If you were a 3D artist, which group of assets would you use to create a scene that matches the scene description? (Considering diversity, quality ...etc)". Our system (**our**) is consistently favored for both questions over the baseline (**base**), except for one scene. This will be expanded upon further in the Appendix.

A summary of user selections for these two questions for each of the 11 scenes is shown in Table 3. Note that in 10 out of the 11 scenes, the study showed that using the assets generated by our system allows the creator of the scene to better match the semantics of the abstract scene description, compared to assets generated by a version of our system without semantic upsampling. Additionally in 9 out of the 11 scenes, the assets were also considered a better selection for 3D artists for creating similar scenes. This demonstrates the efficacy of semantic upsampling in producing both more diverse and relevant assets, and their ability to constitute more semantically aligned 3D scenes.

5 DISCUSSION & CONCLUSION

In this paper, we present a system that leverages the common sense understanding of LLMs, Vision-Language models and Diffusion models to tackle the problem of 3D assets stylization given abstract scene descriptions. Our system uses the key insight that to generate higher quality elements that compose a scene, we can mine the common sense understanding of GPT-3 to semantically upsample the scene semantics based on the abstract scene description using in-context learning. The result is an intermediary representation that is human-readable, editable, and conducive towards higher quality retrieval and texturing of 3D assets.

As a framework for 3D asset generation, our system offers an easy method to transfer the world knowledge of foundational models extracted from modalities like image and text to identifying, texturing and composing meshes that can be used to construct a scene. As the reasoning, generation and texturing potential of these underlying foundation models improve, so would our system outputs.

We showcase our system in action using diverse language inputs, and show the importance of various aspects of our framework

through both quantitative metrics and user studies. In addition, we demonstrate the power and the robustness to our framework afforded by leveraging foundation models for this task in a zero-shot manner.

Although our work makes an important step towards scene synthesis, there are still many open questions to be addressed in future research. For instance, generating valid scene-layouts in an open-vocabulary and generalizable way remains a challenge. Furthermore, future efforts in inferring more 3D consistent texture maps as well as material properties from generative image models are also valuable. Finally, it would also be useful to develop methods that can adequately generate appropriate backgrounds for asset collections.

REFERENCES

Panos Achlioptas, Ahmed Abdelreheem, Fei Xia, Mohamed Elhoseiny, and Leonidas Guibas. 2020. Referit3d: Neural listeners for fine-grained 3d object identification in real-world scenes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 422–440.

Panos Achlioptas, Ian Huang, Minhyuk Sung, Sergey Tulyakov, and Leonidas Guibas. 2022. ChangeIt3D: Language-Assisted 3D Shape Edits and Deformations. *https://changeit3d.github.io/* (2022).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158* (2017).

Angel Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D Manning. 2015b. Text to 3d scene generation with rich lexical grounding. *arXiv preprint arXiv:1505.06289* (2015).

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. 2015a. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving In-Context Few-Shot Learning via Self-Supervised Training. *arXiv preprint arXiv:2205.01703* (2022).

Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2022. Objaverse: A Universe of Annotated 3D Objects. *arXiv preprint arXiv:2212.08051* (2022).

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A Survey for In-context Learning. *arXiv preprint arXiv:2301.00234* (2022).

Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. 2023. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133* (2023).

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 2021a. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10933–10942.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 2021b. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision* (2021), 1–25.

Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. 2022. Shapecrafter: A recursive text-conditioned 3d shape generation model. *arXiv preprint arXiv:2207.09446* (2022).

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems* 35 (2022), 31841–31854.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. 2023. Text2Room: Extracting Textured 3D Meshes from 2D Text-to-Image Models.

Ian Huang, Panos Achlioptas, Tianyi Zhang, Sergey Tulyakov, Minhyuk Sung, and Leonidas Guibas. 2022. LADIS: Language disentanglement for 3D shape editing. *arXiv preprint arXiv:2212.05011* (2022).

Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. 2022. Neural template: Topology-aware reconstruction and disentangled generation of 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18572–18582.

Nikolai Ilinykh, Sina Zarrieß, and David Schlangen. 2019. Tell me more: A dataset of visual scene description sequences. In *Proceedings of the 12th international conference on natural language generation*. 152–157.

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. 2022. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 867–876.

Heewoo Jun and Alex Nichol. 2023. Shap-E: Generating Conditional 3D Implicit Functions. *arXiv preprint arXiv:2305.02463* (2023).

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2022. Magic3D: High-Resolution Text-to-3D Content Creation. *arXiv preprint arXiv:2211.10440* (2022).

Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. 2018. Language-driven synthesis of 3D scenes from scene databases. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–16.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13492–13502.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943* (2021).

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *arXiv preprint arXiv:2212.08751* (2022).

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 12013–12026.

Ryan Po and Gordon Wetzstein. 2023. Compositional 3D Scene Generation using Locally Conditioned Diffusion. *arXiv preprint arXiv:2303.12218* (2023).

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

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

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. TEXTure: Text-Guided Texturing of 3D Shapes. *arXiv preprint arXiv:2302.01721* (2023).

Daniel Ritchie, Kai Wang, and Yu-an Lin. 2019. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6182–6190.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. 2021. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *International Conference on Computer Vision (ICCV) 2021*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633* (2021).

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* 35 (2022), 36479–36494.

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. 2022. Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18603–18613.

Pratheba Selvaraju, Mohamed Nabail, Marios Loizou, Maria Maslioukova, Melinos Averkiou, Andreas Andreou, Siddhartha Chaudhuri, and Evangelos Kalogerakis. 2021. BuildingNet: Learning to label 3D buildings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10397–10407.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509* (2022).

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 567–576.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1746–1754.

Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Proceedings of the 2022 Conference on Empirical*

Methods in Natural Language Processing. 2714–2730.

Kai Wang, Yu-An Lin, Ben Weissmann, Manolis Savva, Angel X Chang, and Daniel Ritchie. 2019. Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. 2021. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*. IEEE, 106–115.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903* (2022).

Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. 2022. Dream3D: Zero-Shot Text-to-3D Synthesis Using 3D Shape Prior and Text-to-Image Diffusion Models. *arXiv preprint arXiv:2212.14704* (2022).

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. 2022. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv e-prints* (2022), arXiv–2210.

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. 2023. Text2NeRF: Text-Driven 3D Scene Generation with Neural Radiance Fields. *arXiv preprint arXiv:2305.11588* (2023).

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493* (2022).

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*. PMLR, 12697–12706.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* (2022).

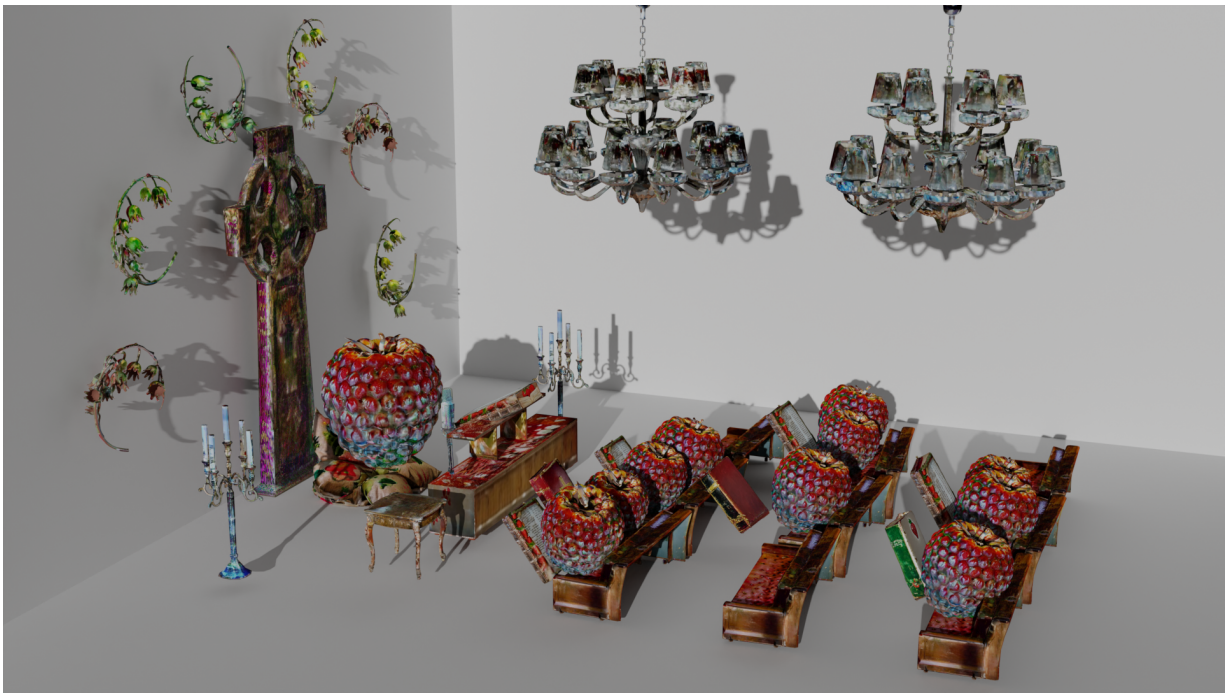

Fig. 8. A scene of "a Church for Strawberries" is one of the more out-of-distribution queries given to our system, but through a combination of human creativity and assets generated by our system, a rather funny scene emerges.

Fig. 9. A scene of "a murder in an abandoned living room". The hallucinations of semantic upsampling tells a gruesome and disturbing story, with the cleaver placed near the bloodied couch, the gun with empty shells around it, and crimson smears on the canvases. We acknowledge the graphic nature of this scene, include this example to show that our system is capable of producing scenes at the extremities of emotional valences, unlike more traditional scene generation systems.

Fig. 10. A scene of "a rustic backyard in the countryside". This example demonstrates the potential of our system to create outdoor-esque scenes by using the common-sense reasoning of foundation models. Notice the elements that suggests the outdoor environment – the gardening equipment on the table, the barbeque grill and bag of coal, the logs and rocks, match sticks and kerosene lamp, and the umbrella table.

Fig. 11. A scene of "a marvel-themed bedroom of a five-year old toddler". As foundation models are trained mostly by data on the internet, we observe that it's able to understand references to pop culture fairly well, resulting in this very prominently marvel-themed bedroom.