Universiteit Leiden
The Netherlands

Bachelor Computer Science & Economics

Does it burn:
Using AI to predict physical properties and affordances for AI-generated objects

Max de Vlieger

First supervisor, Second supervisor:
Bas Haring, Peter van der Putten

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl
22/12/2024

Abstract

In this paper, we propose ways of how one could assign attributes, such as materials and affordances, to 3d generated objects during their generation based on a text prompt. We first discuss how and when in the generation process one could predict what material the object will be made of or its affordance. This can be approached by either basing the prediction on the text prompt or by using image/material recognition on a 2d rendering of the object. We will also demonstrate these concepts through small-scale experiments. Secondly, we go over the different ways of determining the correct/relevant attributes based on the material. These can either be taken directly form a dataset, or obtained using a large language model. Lastly, we discuss how we would store these attributes in the same file as the object by either changing the way to store the attributes or choosing a more flexible filetype.

Contents

1 Introduction

In a three-dimensional virtual landscape, objects play an important role. Although in the past these objects would be manually created and placed in the environment, recent advances in generative models have created great interest in artificially generated objects. This has led to advances in the field of simple 3d object generation based on textual descriptions such as DreamFusion(Poole, B., 2022 [PJBM22])'s usage of a 2d image generation model and Lorraine, J., 2023 [LXZ+23] approach of training their model over multiple prompts simultaneously. Furthermore, developments in 3d room generation (Fang, C., 2023 [FHLT23]) show that it is possible to create both an editable environment and objects based on textual descriptions. We even see a much more recent example with Genie 2 (Parker-Holder, J., 2024 [JPH24]; This model goes even further in also including the affordance of some of the generated objects, making the objects behave the same way a user would expect the object in real life. However, we still see that the generated objects lack the intrinsic physical properties that similar real-world objects adhere to.

When you interact with an object, there will often be a response such as making a sound, moving, or deforming. These differ from object to object on the basis of their material, structure, temperature, and other factors. Simulating these real-world behaviors in the virtual realm is a daunting task. Classical studies on object deformation modeling (James, D. L., 1999 [JP99]) show progress toward translating the real-world behavior of objects to virtual models. Through extensive testing on a real object (Pai, D. K., 2001 [PDJ+01]), realistic behavior can be simulated on a virtual model of the same object. However, there are more ways objects interact with the environment. For example, if the object floats on water or is able to catch fire.

By adding these attributes to an object on generation, it is possible to make the object not only visually represent the prompt or picture, but also physically. This would imply that when the object is imported into a physics engine, all the relevant attributes of the object should be in the object file and can be directly used to accurately simulate the expected behavior of the object. This will allow for a more realistic representation of generated objects in virtual environments.

In this paper, we will propose ways of how one could assign attributes to 3d generated objects generated from a text prompt. This would mean that when the user inputs a description of the object they want to create, the model assigns attributes, such as whether the object can burn or float, to the created object. The selected attributes must comply with the expected behavior of real-life objects and materials. For example, if we assume that the object is made out of wood, it should have the attributes of wood and not of stone. We test different methods of assigning attributes by comparing the attributes assigned against the 'expected attributes' and testing the accuracy. After this, we will explore the different ways of extracting information about related attributes and how to store the attributes as object data.

While this is our initial approach, we will also explore other possibilities for assigning attributes to the generated object, such as defining the material based on the generated object or during the generation process.

In this paper, we will limit ourselves to the material of the object, which is a lower-level attribute, as well as the higher-level attributes related to the material such as density, flammability, flotation or conduction. Although objects themselves have other attributes that define them, like being able to be opened or turned on, due to the wide variety in features and added generational challenges that accompany these attributes, we decided to focus on the materials. However, we do explore the possibility of using affordance to assign other attributes. Lastly, we assume that the objects consist

of only a single material.

The remainder of this thesis is structured as follows. Section 2 discusses works related to the topic and how they can provide insight into our research. Section 3 includes the different ways one predict and assign the attributes. Section 4 explains the different experimental setups that produce the results found in Section 5. Lastly, we discuss our findings, as well as our research and future work in Section 6, ending with the conclusion in Section 7.

2 Related Work

We have three research fields that are of great relevance to this paper. These are text-based 3d object generation, material recognition and affordance. We go more in-depth in these fields to provide more background information as well as related context to create further insight into this paper.

2.1 Text-based 3d object generation

Because this research will be about generating 3d objects, research done in this field is of high relevance to this project. For a generative model to generate a 3d object in response to the user input, it needs to be trained on a large amount of data. This training allows the model to make predictions about what the best response to the given input should be. For example, language models answering a question or image generative models constructing an image as response.

The most straightforward way of training a generative model is to train it using a representation of what you want it to create combined with the corresponding input. For example, if you want a model to generate an image from a text input, you can train it on images with descriptions. For 3d object this would mean training the model on 3d object data. One of such ways is by training the model on voxels, which are three-dimensional pixels representing the surface of the object (Dzik, S., 1992 [DE92]). Another representation of 3d objects are pointclouds, which can also be used (Yang_2019_ICCV [YHH+19]). The main problem with this approach is that while there are 3d CAD (computer-aided design) datasets like ShapeNet (Wu_2015_CVPR [WSK+15]) for these voxels or pointclouds, compared to text-to-image data, the amount of available training data is very small. Because of this, it can be difficult for a model trained on this data to produce a wide variety of objects. The main upside of this method is that because the training data is 3d, if there is enough training data available, the structural accuracy will be high.

A way to circumvent this problem is by utilizing a 2d diffusion model as a base. Generation methods like DreamFusion (Poole, B., 2022 [PJBM22]) show that it is possible for a model to learn 3d structures only based on text-to-image data. By utilizing neural radiance fields (NeRF's) the 3d geometry of the object can be determined, and 2d renderings can be made of other angles. By minimizing a loss function so that the 2d renderings are as realistic as possible, a proper 3d model can be created. These 2d renderings allow for an interesting way to determine the material of the object, which we can utilize in our approach. However, the use of 2d training data for 3d models does come with an inherent problem. Due to perspective, multiple different 3d scenes can result in the same 2d image. This can lead to irregular geometry and texturing.

While the texture of objects can be generated as well by using these diffusion approaches, there is also the possibility of just creating the structure of the object, without texturing. Richardson, E.,

2023 [RMA+23] and Siddiqui, Y., 2022 [STM+22] propose methods for texturing a 3d object. The latter method focuses on text-guided texturing, which can be useful if we would first generate the material before texturing the object.

These methods of generating the object are important because not only does it provide the backbone of the object generation process, the different methods of generating the object and its texturing allows for multiple approaches in assigning attribute data.

2.2 Material recognition

Because we limit ourselves to the attributes related to the material of which the object is composed, material recognition is essential to our research. If we are able to recognize the material the generated object consists of, we can more accurately determine the correct attributes to assign and make the objects look and attributes more coherent. Most material recognition is done through image recognition. Images supply a large sum of data about the visual properties of an material, may it be through color, lighting or structure. However, because so many factors affect the visual appearance of the material, it can be difficult to correctly classify it. Approaches such as Bello-Cerezo, R., 2019 [BCBDM+19] and Liu, C., 2010 [LSAR10] show the ability to recognize material based on visual aspects, while other articles such as Lin, G., 2017 [LSVDHR17] and Bell, s., 2015 [BUSB15] show the importance of context. We can use these methods of material recognition to classify what material the generated object composes of and assign the relevant attributes.

If we look at material recognition for text data, the visual aspects are no longer present. This would leave only the contextual description to make a prediction of. A model that shows great promise is MatScitBert (Gupta, T., 2022 [GZKM22]). MatSciBert is a text mining and information extraction model for material science. The model is trained on scientific literature in this field to handle scientific notations and jargon. This can be useful if the material has to be very specific. Another option is to use a regular language model if we use the right training data and utilize prompt engineering. One last thing to note is that the training data for the generative model and the material recognition model are different. This means that if the two models do not work together, a disparity can occur between the material predicted based on the prompt and the material used for the texture of the object.

2.3 Affordance

While you can use the texture of the object to predict the material and then using that material to predict other related attributes, it is also possible to predict these related attributes directly if they are part of the affordance. Affordance is tied to the psychology of perception and is described by The Interaction Design Foundation as: “Affordances are the characteristics or properties of an object that suggest how it can be used.” [IxD16]. This means that affordance is not really an exact property of an object, but it is more like the actions an actor would assume they would be able to perform with the object. They are basically the “opportunities for actions” an actor assumes when looking at the object. A good example for this would be a (door) handle. If we see a handle on an object, like a door, we intuitively assume that pushing down and pulling/pushing will open the object, which could be a door or a drawer. The same is true in virtual environments. The affordance of the objects we perceive in these environments are linked to what we expect from interacting with them, making them not unlike our “expected attributes”.

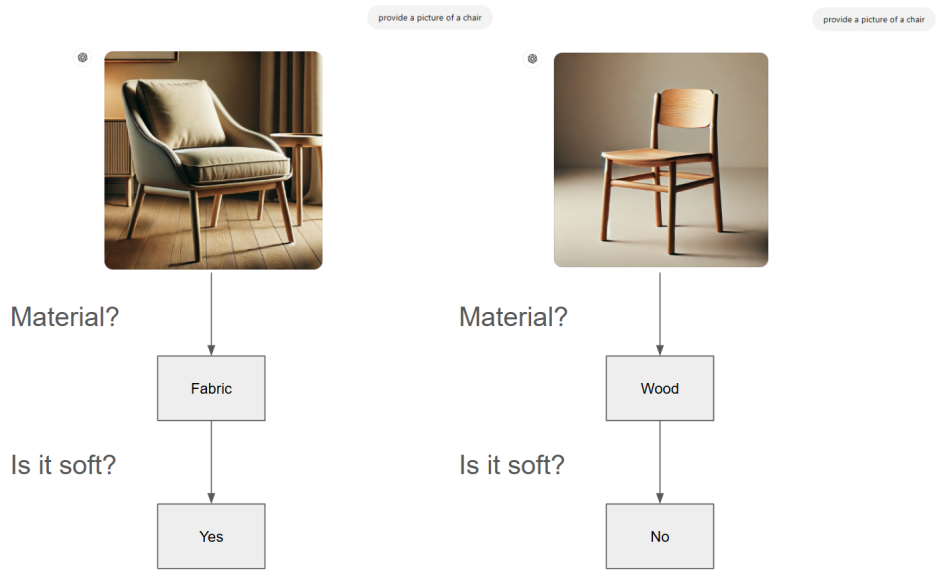

Figure 1: We can see that, while using the same prompt, we still get different objects with different materials and affordances.

There are different levels of information that one could perceive about an object. One way of describing these levels is that higher levels of information are constructed from the lower levels. We can take a balloon for example. A lower-level observation would be that it is a hollow object made of rubber, which is under tension. We can then infer the higher-level information that the balloon would pop if we poked it with a sharp object. This makes the higher-level information an indirect perception. We only constructed it from the lower-level observations, not perceived it directly. However, the psychologist James Gibson opposed this view. He argues in his paper "The Theory of Affordance" [Gib77], that animals can pick up information about affordances directly from ambient light. This would allow them to get to the higher-level perceptions without constructing them from lower-level perceptions. The same can be said for humans and their perception of something. This would mean that in addition to how we can assign attributes (like combustion or flotation) based on the material of the object we perceive, we could assign certain attributes directly from the sensory input, allowing us to assign attributes which are a big part of the "expected attributes" without first determining other lower-level observations which might be less relevant.

3 Attribute Assignment Methods

When generating an object from text, the user fills in a text prompt from which the model generates the object. Based on the information the model is trained on, it can make predictions about what the user wants to achieve with their input and generate an object accordingly. The same as for the object the model is generating, so can the attributes be generated purely of the text input alone. However, this is where we run into some problems. If the provided prompt is too ambiguous, the same prompt could produce multiple different results with different affordances or attributes. An example would be Figure 1. Some attributes are heavily dependent on certain factors. In our case, because we are interested in the attributes related to the material, the material determines if an

attribute is correct or not. For example, if we create a cup, it could be made of wood or ceramic. While the wooden cup does burn and float, the ceramic cup might not. If we only have the context of "a cup" we would still want the attributes to be consistent to one another. Furthermore, the model that generates the object also visualizes the texture of the object. If the texture looks like wood, we do not want the attributes that are assigned to match ceramic.

To test if this is an issue, we designed a test in which we ask a large-language model to predict the material of an object based on a text prompt. We then use this text prompt to generate a 3d object, which we show to a test subject who also has to make a prediction about what material they expect the object to be. The entire experimental setup can be found in Section 4.

We can see in Figure 5 that there exist cases in which the material predicted based on the prompt differs from the material predicted by the test subject. This can indicate that the texture of the created object and the predicted material that is partially based on it do not align with the prediction based on the prompt. Although this is not strange because differences in training data produce different assumptions, the issue is something that needs to be addressed if we want our assigned attributes to match the "expected attributes" as well as possible.

A large part of our research is assigning attributes that are related to the material. To ensure that these attributes are more accurately aligned with each other, we propose first to determine the material, after which further attributes can be assigned using the material as a baseline.

While we focus on materials, for some objects, their affordance can speak volumes about what an object should be able to do and, by extension, what attributes should be assigned. This allows for elaborate attributes, which are a product of multiple lower level attributes combined. Because the affordance of an object can play a big part in the expected attributes of an observer, being able to detect and use these affordances might lead to more realistic object interaction and thus a higher accuracy between the assigned and expected attributes. Although our main approach determines the material before assigning the attributes, to explore the effectiveness of using affordance for attribute predictions, we conducted two small tests which can be found in Sections 4 and 5.

3.1 Material prediction

If you generate a textured 3d object, it might look like the material is already discovered and utilized. Because if the object has a wooden texture, why would not the model know that the material the object is made of is wood? However, the visual material and color of the object are based on the training data about how the related imagery of such an object looks. The actual material is not something directly determined or defined. If there is no material in its prompt, it will not go out of its way to determine the material before assigning how the object actually looks. Even if the material is in the prompt, the material still needs to be recognized as something that needs to be determined and saved as an attribute. This is why we need the model to recognize the attributes, in this case the material, so that further attributes can be assigned based on that information. To ensure the material and its attributes line up with the texture of the object, we have two options. The first approach is to directly define the material based on the text prompt inputted by the user. This allows us to use the material not only for assigning the correct attributes but also for determining the texture of the object, increasing the coherency between the attributes and the texture. The other approach is to define the material based on the texture of the generated object. If we utilize material recognition on the texture of the object, it allows us to predict what material it consists of, and by extension what attributes to assign. Because we use the texture of

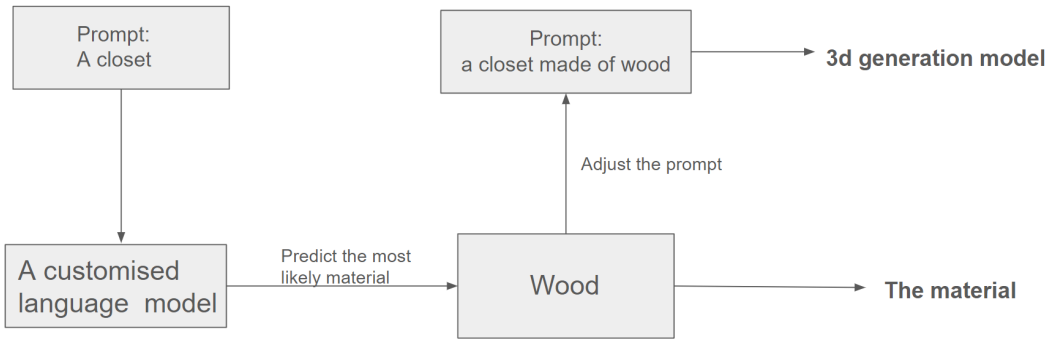

Figure 2: By using a customised language model on the initial prompt, we can predict the material the object likely consists of before the object is created. We can then adjust the prompt accordingly to increase the coherency between the predicted material and the texture of the object

the object to define its material, it has a similar effect on the coherency.

3.1.1 Predicting the material based on the text prompt

If we want to directly define the material based on the text prompt, there are some things to consider. We are working under the assumption that there is no set format for the text prompt. This means that the text prompt can vary a lot in the amount of information provided. While sometimes the prompt can include the material, color, object and size, it can also just include a single word. Due to this variation, it is difficult to train a model to determine the material on very specific information. To be able to make a prediction about the material with whatever information is provided, we can use large language models (LLM's).

Large language models have shown their predictive capabilities in the past. By having access to an enormous amount of text data, they are able to respond to inputs of high variety. For example, what the most likely material is for a cup to be made out of. We could fine-tune a language model like BERT as well as utilizing prompt engineering to make a prediction of the material only based on the text prompt. Prompt engineering is useful because it allows us to format the initial input in a way that increases the accuracy of the response. This approach is shown in Figure 2. Although there can be a large difference between the material predicted from the text prompt and the visualization of the 3d object, there is a way to circumvent this risk. If we determine the material prior to starting the generation of the 3d object, we can use the material as an additional context, which in turn would help to reduce disparity. Alternatively, if we generate a textureless object, we can use models such as Richardson, E., 2023 [RMA+23] and Siddiqui, Y., 2022 [STM+22] to texture the model with the material related to the attributes we assigned.

3.1.2 Predicting the material based on a 2d image

The other option we have to predict the material of the object is to base it on a 2d image. While it might seem like we are not able to use this, due to our paper focusing on a text input instead of an image input, there are still ways in which we can apply it. As discussed in Section 2.1, the amount of image text-image data is much higher than text-3d object data, which led to text to 3d

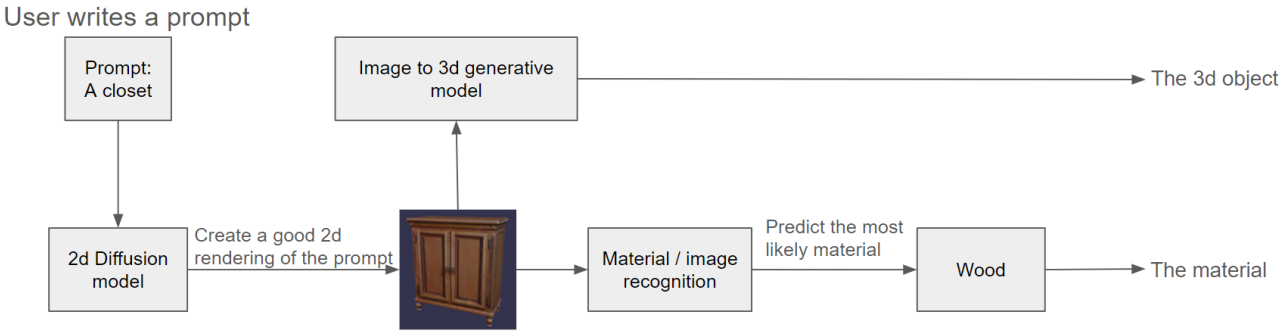

Figure 3: By using Material / Image recognition on the 2d rendering of the prompt created by the diffusion model, we can predict the most likely material of the object.

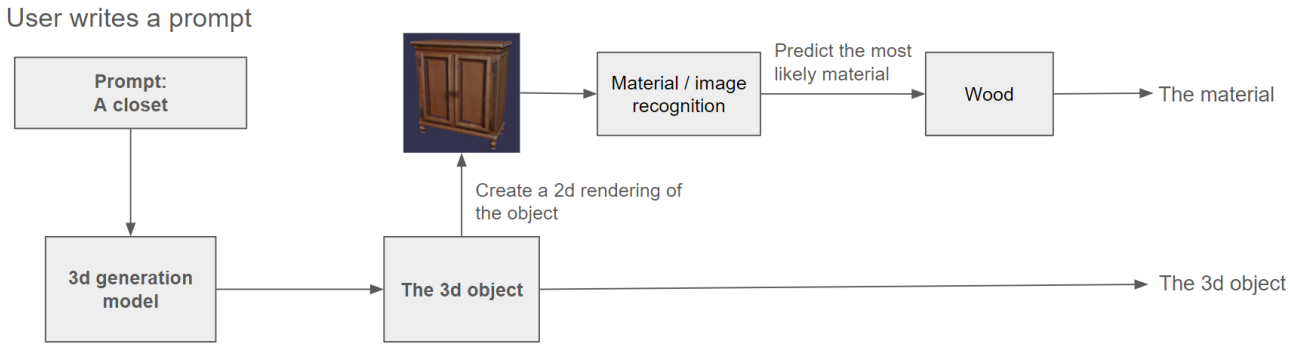

Figure 4: The alternative approach where the 3d object is generated first and then a 2d rendering of the object is made.

generative models such as DreamFusion [PJBM22] and Magic3D (Lin, C., 2023 [LGT+23]) to use 2d image diffusion models in their approach. These diffusion models generate 2d image data, which then is used to create the 3d object, allowing these approaches to utilize the vast amount of text to image data. A big benefit for us, because it also provides us with the 2d input we need to use image recognition on, leading to the approach shown in Figure 3.

However, this approach does not have to be limited to just models that use images in its generation process. A 2d image representation of an 3d object can be made after its generation, just like a picture. We can then use image recognition on the 2d image to determine the material. This approach is shown in Figure 4 Because the object is created prior to the material being assigned, this is another way of improving the coherency between the material, the visual representation of the object, and, by extension, the related attributes. It is important to note that, while He, R., 2023 [HSY+23] discuss the use of artificial imagery for training image recognition models, the accuracy of material recognition on artificial generated imagery compared to real imagery can vary.

3.2 Assigning the attributes

After determining the material of the object, we can continue to assign properties and attributes based on the predicted material. One way to do this is to step away from predictive models and use a data set that contains all available materials with their relevant attributes. By using the direct data from the dataset, any possibilities of miss-assignment can be avoided and the assignment can be done quickly. This makes it the ideal approach for when there is a set amount of materials and features that can be generated. However, this approach comes with a limiting factor. Because a data set has a set number of materials and features, it will not work if the object is made from a material that is not yet in the data set. This means that you have to limit your material recognition/prediction model to only include materials in the data set or group materials together. Although the amount of material data available is huge, it is not feasible to include every material in a single dataset. Not only because not all data are available for every material, new compounds are continually being discovered. While in many use-cases, having the ability to assign hyperspecific materials and attributes is not necessary, it is still something to keep in mind. This method is also limited to the attributes available in the dataset, so if there are attributes that you would like to assign that are not available, either the dataset needs to be adjusted or they need to be assigned in another way. However, the big upside is that it avoids the need for a language model, making it less prone to inaccuracies. This makes it a preferable option when you only need basic attributes and common materials.

Other than assigning the attributes directly, we could also use language models for this task. The use of language models for information extraction in scientific fields has increased in recent years because of the large amount of information available in books, papers, and other texts. Finding the information you need from such a large number of possible sources can take a lot of time, so being able to automate that step is of great interest. One such usage of language models for information extraction is in the workflow introduced by Liu, S.,2024. [LWPS24]. Their workflow is structured as followed as stated in their paper: "(i) define the material classification problem to be addressed; (ii) design prompts (via prompt engineering) to distill knowledge from LLMs and store the information as textual data; (iii) fine-tune a bidirectional encoder representations from transformers (BERT) model to train on the stored textual data-label pairs; (iv) apply the model to explore new materials or study composition-structure-property relationships." (Liu, S.,2024. [LWPS24]) As shown in their paper with metallic glass classification in Figure 3, this approach would allow us to infer specific attributes if we provide it with a material. This makes the workflow a good display of how language models can be used for our goal and is a valid option to use. Language models can also be a lot more versatile, due to being able to make predictions about attributes where the data is not known or not in the training data.

To fully utilize this approach, we would need a good model to replace MgBert (the model that they use) with. This is because we want attributes of a large variety of materials, not just metallic glasses. One such option is MatScitBert (Gupta, T., 2022 [GZKM22]) which is a pretrained language model specifically designed for the material science domain. Because the model is trained on a large number of scientific papers related to material science, it can be used to gain a lot of material-specific information. Furthermore, it is a continuation of the SciBert model, so while MatScibert's corpus is from the material science families of inorganic glasses, metallic glasses, alloys, and cement and concrete, it also encompasses thermoelectric, nanomaterials, polymers, and biomaterials. This means that it has access to a wide variety of materials, making it a valuable

option for our task. However, this also comes with a caveat. By widening the amount of materials and possibilities, we also run the risk of reducing the accuracy. If we want higher accuracy and only need access to specific materials, it can be beneficial to train a custom model. There is another problem. Because the data available is very specific and exact about the compounds and materials as well as their attributes, this might not be the best option if we want a more generalized approach. Furthermore, over-specification of a material might hamper the collaboration between the material recognition model and the attribute prediction model.

Although we focus only on the material attributes, an easy attribute that can be assigned additionally is weight. After generating the object, we should have access to the dimensions, so we can calculate the volume. Combining this with the weight of the material allows us to calculate the weight of the object. This does come with a problem, however, and that is the scale of the object. If we assign a set weight to the object, if the object is either up- or down-scaled, the weight would not change with the size, leading to inconsistencies.

3.3 Where do we store our object attributes

If you generate a 3d object, you also need a place to store it. A file type that is often used and is often the default is an object(OBJ.) file. These files typically only consist of the geometry, color and texture of the object while also referencing to another file, the material library file. In the material library file, the information about the material and its lighting is stored. Because we want to assign other attributes to the object, it would be beneficial if these attributes could be stored directly inside of the object file. However, an object file does not support additional variables that we would like to use to store our attributes. There are two ways to circumvent this issue. Firstly, we could add our attributes as comments to the files. This can be done both in the material library file as well as the object file. Because comments are normally ignored, they do not result in an unsupported file format. To extract our attribute data from the file, we would need a custom parse tool that can extract the comments from our file, after which we can utilize the data as if they were variables.

The second option is to use a more flexible file format. There are a multitude of different 3d file types with a wide variety of properties. While most file types can store our 3d object data, not all of them can hold the variable data for our attributes. This means that the file formats we can use are limited to formats that support the input of custom variables. If the file format allows for custom variables, we can make as many variables as we want for as many attributes we want to assign, making it a lot easier to store and use our attributes.

One file type that we can use is glTF. glTF (Graphics Library Transmission Format) is an file format that is primarily in JSon but is still able to refer to external data. This can be compared to how object files reference a material library file to obtain its textures or lighting. The main difference glTF has over the object file format comes from it being in JSON. Because JSON is easily parsable, it allows for a much smoother integration with possible physic engines. Continuing on, glTF allows the file to be extended using JSON, allowing us to add our own variables and, by extension, our attributes to the file. Lastly, glTF can be used in many cases. While there are a multitude of content creation tools that provide glTF import and export, other file formats, like object files, can be converted to the glTF format trough the use of a conversion tool. This makes glTF a viable option which is not dependent on the 3d object generation tool that is used. It is important to note that gltf is just an option and that there are multiple file formats that can be

used for this.

4 Experimental Set Up

To test the different methods we propose in Section 3.1, we designed multiple exploratory experiments. These tests aim to put the predictions of the model, which we could use for automatically assigning the attributes, against the expected attributes a potential observer would assume. For our experiments, we use the Tencent/Hunyuan3D-1 [YSZ+24] model on Hugginface [Hug24] to create a 3d object based on the provided prompt. This model has the added benefit of creating a 2d image on which the 3d object is based. We can use this image to test the first approach we propose in Section 3.1.2. Furthermore, we use GPT-4o to make predictions in each of the different tests.

In our first experiment, we test if there is a difference between a model predicting the material based on a text prompt and a test subject predicting the material based on the 3d object. We first ask a large language model the question: "Answer in only a single material with no additional information, what material is a spoon". We then asked a 3d generative model to create the same object, in this case "a spoon". This object is shown to an test subject without explaining what the object is or how it is created with the question: "What material is this object?" from which the results make up the Expected attributes.

The goal of our second experiment is to use prompt engineering on the prompt provided to the generative model such that the created object more closely aligns with the material predicted on the initial text prompt. We ask GPT-4o the question: "Answer in only a single material with no additional information, what material is a cabinet" for ten different objects (cabinet being one of them). We then store this response as our predicted attribute, as well as adjusting the prompt with which we would provide the 3d generative model to include: "... made of 'prediction'" with the prediction being the material predicted by our language model. Example: "a spoon made of metal". After generating a 3d object with our new prompt we show it to a test subject with only the question: "What material is this object made of?". We do not give additional information, such as what the prompt was or what the object is supposed to be, the same as for our first experiment.

Our third experiment includes the image generated during object generation. We provide the object generation model with only the object. Example: "a spoon". The image generated by the model is then provided to GPT-4o with the following prompt: "Answer in only a single material with no additional information, what material is this object". These results will then be tested against the predictions made by our test subject in the same way as in our other experiments.

Finally, our fourth experiment is by creating 2d renderings of the 3d generated object. Just as in our third experiment, we provide our object generation model with the object prompt. Instead of using the 2d image created, we take 2 pictures of the created object from opposing sides, which we provide to our language model with the same question as in our second experiment. This too will then be tested against the predictions made by our test subject, in the same way as done in our other experiments.

We used these experimental setups on a small scale to provide the results shown in our Results section. These experiments can systematically be scaled up to obtain results that could prove which method is the most effective. For this, a larger set of objects has to be used, as well as a large group of test subjects to provide expected attributes which are on average more in line with a potential observer/user of the object.

	Based on prompt	Expected attributes
Spoon	Metal	Metal
Table	Wood	Marble
Plane	Aluminum	Metal
Ball	Rubber	Cloth
Cabinet	Wood	Wood
Cup	Ceramic	Ceramic
Car	Steel	Metal
Vase	Glass	Porcelain
Chair	Wood	Leather
Closet	Wood	Wood

Figure 5: Results of 10 different objects using GPT-4o for the prediction and Tencent/Hunyuan3D-1 for the object generation. For our expected materials we used a test subject which only gets to see the 3d object and gets asked the question: "What material is this object"

Although we limit ourselves to the attributes of materials, we also wanted to use these experimental setups to test the prediction of the affordance for some of these objects. As we can see in the recent Genie 2 model, by using affordance the objects become even more realistic to interact with. To see if the language model could also predict the affordance, we asked GPT-4o the question: "answer in only a single word, what happens to this object when i push against it" as well as the question: "answer in only a single word, what happens to this object when i press a sharp object against it". We asked this while providing a text prompt ("a balloon"), a 2d image, a 2d rendering of the 3d generated object, as well as asking a test subject the same question while providing the 3d objects.

5 Results

Figure 5 shows the result of our initial experiment to test if there is a difference between the material predicted on the text prompt and the expected material of a test subject based on the 3d generated object. The accuracy is 40%, which could indicate that there is a high chance that differences can occur between the two predictions. The data of the experiments can be found on our github https://github.com/lucario121212/Attributes-to-3d-objects

Figure 6 shows the result of our proof-of-concept experiment in which we rewrite the input for the 3d generative model to include the material predicted based on the prompt. The accuracy is 70%, while without adjusting the prompt it is 40% as shown in Figure 5. This indicates that there is a high possibility that this method is a meaningful improvement in coherency between the assigned and expected attributes, which could be proven by conducting this experiment on a larger scale.

Next we have the two methods discussed in Section 3.1.2, which is to predict the material based on a 2d image created during the generation process and based on 2d renderings of the created object itself. We can see in Figure 7 that for the materials predicted based on the image, the accuracy is 80% while it is 70% for the predictions based on the 2d renderings of the 3d object. This is, just like the experiment shown in Figure 5, a large increase in accuracy, making both these

	Based on prompt	Based on object
Spoon	Metal	Metal
Table	Wood	Wood
Plane	Aluminum	Metal
Ball	Rubber	leather
Cabinet	Wood	Wood
Cup	Ceramic	Ceramic
Car	Steel	Metal
Vase	Glass	Glass
Chair	Wood	Wood
Closet	Wood	Wood

Figure 6: Results of 10 different objects where the material based on the prompt was used to edit the input for the 3d generative model by adding made of "material" at the end of the prompt. example: a spoon made of metal. We used GPT-4o to predict the material based on the prompt and use a test subject which only gets to see the 3d object to determine our expected material.

approaches possibly viable and in need of further research.

Lastly, we also wanted to do a small test on the affordance of objects instead of the material. These results are shown in Figure 8 and 9. We can see that, for all three of the different methods, the results align with the expected reaction. This could mean that all of these methods can be utilized to assign attributes based on affordance.

6 Discussion

The scope of attributes to assign and how its done can be very wide, thus we need to reflect on the results of our research, as well as discuss our known limitations and possible future work that can be done based on this research.

6.1 Reflection on the results

While the approach of making the prediction based on the 2d image has the highest accuracy, at 80%, due to the tests we conducted being only a proof of concept, we cannot say with definitive proof that this result is the best. Furthermore, different methods might prove beneficial depending on the use case. If we do not need specific material information but just want to generalize materials, for example, aluminum and steel are both metals, we can see that the accuracy of the prediction based on the text prompt increases to 90%. However, this works in both ways. If the test subject that provides the expected attributes is not knowledgeable about a certain object, their predicted material might be a lot more generalized than an expert who knows the most likely type of metal or wood for this specific object. This is why researching these methods with different use cases could also prove worthwhile.

If we look at our experiment for the affordance, we can see that the predictions of the model for all different methods align. This could mean that all three methods could be used to make

	Based on Image	Based on object	Expected attributes
Spoon	Metal	Metal	Metal
Table	Glass	Marble	Marble
Plane	Metal	Metal	Metal
Ball	Plastic	Plastic	Cloth
Cabinet	Wood	Metal	Wood
Cup	Ceramic	Ceramic	Ceramic
Car	Metal	Metal	Metal
Vase	Porcelain	Ceramic	Porcelain
Chair	Leather	Leather	Leather
Closet	Wood	Wood	Wood

Figure 7: Results of 10 different objects. The results based on the picture are based on the picture created by Tencent/Hunyuan3D-1 during the generation of the 3d object. The results based on object are from the 2d renderings of the object.

	Based on text	Based on Image	Based on object	Expected reaction
Balloon	Deforms	Deforms	Deform	Deforms
Door	Opens	Opens	Open	Opens

Figure 8: Results of two different objects provided with the question: "answer in only a single word, what happens to this object when i push against it"

	Based on text	Based on Image	Based on object	Expected reaction
Balloon	Pops	Burst	Pops	Pops
Door	Scratches	Pierce	Scratches	Scratches

Figure 9: Results of two different objects provided with the question: "answer in only a single word, what happens to this object when i press a sharp object against it"

assumptions about the affordance directly. However, just as with the other tests, to create certainty we would need to conduct this test on a larger scale.

6.2 Limitations

There are a couple of limitations with this research. Firstly, we confine our approach to materialistic attributes. Because of this, we can approach the problem by first defining the material, as discussed in Section 3.1. If we were to extend this research to include other attributes, for example, if an object can be opened or turned on, this approach can not fully be used because the way the object is generated has to accompany the attributes. If a closet that is generated does not have an inside or well-defined doors, defining that the object cannot be opened is technically true. However, we stated in the start of our paper that the generated objects must comply with their expected behavior. While the closet might be closed, you still expect there to be an inside.

While we are on the topic of expected behavior, it is not something that is factual and set in stone. The goal of assigning attributes with their expected behavior in mind is such that the user experience for interacting with the object is as realistic as possible, which can be different between people. Because the generated objects only exist digitally, the object does not have a ground truth to compare against. While the accuracy of the assigned attributes and material in relation to the 3d generated object can not be checked, the accuracy of the assigned attributes in relation to the predicted material can. Because we base all of our attributes on the predicted material, when using a language model for this task, its accuracy can be checked against factual data about said material.

Another limitation to keep in mind is that we use only a single language model and an object generation model in our tests, these being GPT-4o and Tencent/Hunyuan3D-1. In Section 3.1 we talk about different models and how they could be used in our particular case. These models could prove to be more effective at the different tasks provided, and as such will also need to be subjected to more research, ultimately comparing the best models for the different approaches to see which approach comes out on top.

Lastly, while we add all the relevant attributes to the file, this does not mean that the file is immediately usable. In many cases, for the assigned attributes to be relevant, one needs to import the object into a physics engine. Physics engines would still need to know how to handle the additional variables added to the file.

6.3 Future work

Generating objects with inherent attribute data allows for more complete objects. One way to expand on this research is by adding this way of object generation to world generation. Because these objects already have their attributes generated, this would allow for players to interact with these objects directly after they are generated. If this can be combined with room generation approaches such as (Fang, C., 2023 [FHLT23]) it could lead to even more complete generation.

Another way of continuing this research is to assign not only material-based attributes to the object on generation but also object attributes. We explored this in our small part about affordance. For example, if you generate a lamp, that you can interact with it to turn it on. This would be not that different from our approach because it is also an attribute that can either be true or false; however, instead of basing the attribute of the material, we base it on the object.

We can take this even further by combining this affordance with the structural generation. While a closet can be opened, the object needs to know what parts of the object are the actual doors. Not only that, but the inside has to be hollow and textured. To make the doors properly responsive, either the exact coordinates of the "doors" of the object could be defined and the side of the hinge, or the doors could be their own objects with their own attributes.

Another possible continuation is to make distinctions between the different parts an object consists of. A closet not only consists of wood, but can also have nails, screws, or iron hinges. If those parts can be generated separately, or just be defined with their own features, even more realistic object generation and interaction can be achieved.

Lastly, an easy addition would be to combine the different methods discussed. For example not only giving the model 2d renderings of the object, but also the prompt which was used to create the object as well as the 2d image. Combining these methods gives the model more information to base their predictions of, possibly resulting in a higher accuracy.

7 Conclusion

Virtual environments are becoming larger and more complex. While 3d object generation could prove to relieve some of the time-intensive task of manually creating objects, the generated objects are not very realistic. They often cannot be interacted with and do not interact with the environment as you would expect. This is why we proposed a method of adding attributes to the objects on generation, as well as going over alternative ways of approaching this problem. By utilizing a fine tuned language model we can define what material the object will be made of from the text prompt. By using both this material for predicting our attributes and for the generation process, we increase the coherency between each other. Alternatively, image recognition can be used during or after object generation to determine the material based on the texture of the created object, ensuring coherence in that way.

Even if we have determined the material of the object, the use-cases for attributes can still vastly differ. Where some people might want to use the objects for simple interactive environments, others could want to extend the accuracy of the attributes to material science levels. This is why we discussed two different approaches. Either using a database with the relevant attributes and materials, or using a fine-tuned large-language model, for the very in-dept questions, where relevant data is only found in scientific papers.

Finally we need a place to store the 3d object and its attributes. While the OBJ (object file) is an often used filetype, we would need to store our attributes in comments because it does not support additional variables. Alternatively, there are other file types that we could utilize that are much more flexible, allowing easy storage and retrieval of our attributes.

References

[BCBDM+19] Raquel Bello-Cerezo, Francesco Bianconi, Francesco Di Maria, Paolo Napoletano, and Fabrizio Smeraldi. Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions. *Applied Sciences*, 9(4), 2019.

[BUSB15] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3479–3487, 2015.

[DE92] Steven Dzik and Jay Ezrielev. Representing surfaces with voxels. *Computers Graphics*, 16(3):295–301, 1992.

[FHLT23] Chuan Fang, Xiaotao Hu, Kunming Luo, and Ping Tan. Ctrl-Room: Controllable text-to-3d room meshes generation with layout constraints. *arXiv preprint arXiv:2310.03602*, 2023.

[Gib77] JJ Gibson. The Theory of Affordances. *Perceiving, acting and knowing: Towards an ecological psychology/Erlbaum*, 1977.

[GZKM22] Tanishq Gupta, Mohd Zaki, NM Anoop Krishnan, and Mausam. MatSciBERT: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, 2022.

[HSY+23] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition?, 2023.

[Hug24] Hugginface. Tencent Hunyuan3D-1.0: A Unified Framework for Text-to-3D and Image-to-3D Generation, 2024. https://huggingface.co/spaces/tencent/Hunyuan3D-1 [Accessed: 6 Dec 2024].

[IxD16] Interaction Design Foundation IxDF. What are Affordances?, 2016. https://www.interaction-design.org/literature/topics/affordances#:~:text=affordances%20are%20the%20characteristics%20or%20properties%20of%20an%20object%20that%20suggest%20how%20it%20can%20be%20used. [Accessed: 22 Dec 2024].

[JP99] Doug L James and Dinesh K Pai. ArtDefo: accurate real time deformable objects. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 65–72, 1999.

[JPH24] Jake Bruce Vibhavari Dasagi Kristian Holsheimer Christos Kaplanis Alexandre Moufarek Guy Scully Jeremy Shar Jimmy Shi Stephen Spencer Jessica Yung Michael Dennis Sultan Kenjeyev Shangbang Long Vlad Mnih Harris Chan Maxime Gazeau Bonnie Li Fabio Pardo Luyu Wang Lei Zhang Frederic Besse Tim Harley Anna Mitenkova Jane Wang Jeff Clune Demis Hassabis Raia Hadsell Adrian Bolton Satinder Singh Tim Rocktäschel Jack Parker-Holder, Philip Ball. Genie 2: A large-scale foundation world model, 2024. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/ [Accessed: 8 Dec 2024].

[LGT+23] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023.

[LSAR10] Ce Liu, Lavanya Sharan, Edward H Adelson, and Ruth Rosenholtz. Exploring features in a bayesian framework for material recognition. In *2010 ieee computer society conference on computer vision and pattern recognition*, pages 239–246. IEEE, 2010.

[LSVDHR17] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Exploring context with deep structured models for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1352–1366, 2017.

[LWPS24] Siyu Liu, Tongqi Wen, A. S. L. Subrahmanyam Pattamatta, and David J. Srolovitz. A prompt-engineered large language model, deep learning workflow for materials classification, 2024.

[LXZ+23] Jonathan Lorraine, Kevin Xie, Xiaohui Zeng, Chen-Hsuan Lin, Towaki Takikawa, Nicholas Sharp, Tsung-Yi Lin, Ming-Yu Liu, Sanja Fidler, and James Lucas. Att3d: Amortized text-to-3d object synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17946–17956, 2023.

[PDJ+01] Dinesh K Pai, Kees van den Doel, Doug L James, Jochen Lang, John E Lloyd, Joshua L Richmond, and Som H Yau. Scanning physical interaction behavior of 3D objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 87–96, 2001.

[PJBM22] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion, 2022.

[RMA+23] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[STM+22] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*, pages 72–88. Springer, 2022.

[WSK+15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[YHH+19] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[YSZ+24] Xianghui Yang, Huiwen Shi, Bowen Zhang, Fan Yang, Jiacheng Wang, Hongxu Zhao, Xinhai Liu, Xinzhou Wang, Qingxiang Lin, Jiaao Yu, Lifu Wang, Zhuo Chen, Sicong Liu, Yuhong Liu, Yong Yang, Di Wang, Jie Jiang, and Chunchao Guo. Tencent Hunyuan3D-1.0: A unified framework for text-to-3d and image-to-3d generation, 2024.