Physical Property Understanding from Language-Embedded Feature Fields

Albert J. Zhai Yuan Shen Emily Y. Chen Gloria X. Wang Xinlei Wang
Sheng Wang Kaiyu Guan Shenlong Wang
University of Illinois at Urbana-Champaign

Abstract

Can computers perceive the physical properties of objects solely through vision? Research in cognitive science and vision science has shown that humans excel at identifying materials and estimating their physical properties based purely on visual appearance. In this paper, we present a novel approach for dense prediction of the physical properties of objects using a collection of images. Inspired by how humans reason about physics through vision, we leverage large language models to propose candidate materials for each object. We then construct a language-embedded point cloud and estimate the physical properties of each 3D point using a zero-shot kernel regression approach. Our method is accurate, annotation-free, and applicable to any object in the open world. Experiments demonstrate the effectiveness of the proposed approach in various physical property reasoning tasks, such as estimating the mass of common objects, as well as other properties like friction and hardness. Code is available at https://ajzhai.github.io/NeRF2Physics.

1. Introduction

Imagine that you are shopping in a home improvement store. Even though there is a huge variety of tools and furniture items in the store, you can most likely make a reasonable estimate of how heavy most of the objects are in a glance. Now imagine that you are hiking in a forest and trying to cross a stream by stepping on some stones in the water. Simply by looking at the stones, you are probably able to identify which stones have enough friction to walk on and which stones would cause you to slip and fall into the water.

Humans are remarkably adept at predicting physical properties of objects based on visual information. Research in cognitive science and human vision has shown that humans make such predictions by associating visual appearances with materials that we have encountered before and have rich, grounded knowledge about [9, 10].

It is highly desirable for computers to be equipped with similar or even better capabilities of material perception. Having computational models for perceiving physics from visual data is crucial for various applications, including robotics, agriculture, urban planning, graphics, and many other domains. Nevertheless, challenges remain. One is the difficulty of acquiring labeled ground-truth data. For example, consider the endeavor of measuring the mass of a tree, or measuring the thermal conductivity densely throughout a coffee machine. Another is the highly uncertain nature of the prediction task due to having limited observations – there is simply no way to know with certainty what is in the interior of an object without additional information.

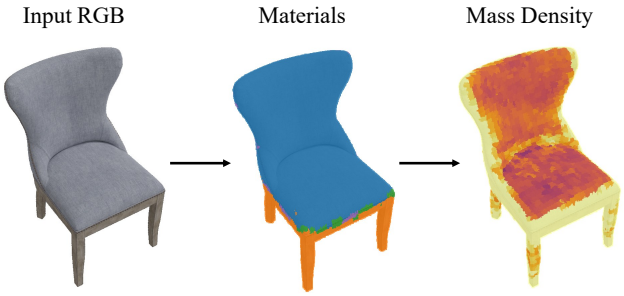

Figure 1. **Estimating physical properties from images.** Humans can predict physical properties of objects by associating visual appearances with grounded knowledge about materials. We propose to equip computers with this capability by combining language-embedded feature fields with LLM-based material reasoning.

This paper presents a training-free approach towards uncertainty-aware dense prediction of physical properties from a collection of images. Our method, named NeRF2Physics, integrates object-level semantic reasoning with point-level appearance reasoning. First, we use a neural radiance field to extract a set of 3D points on the object's surface and fuse 2D vision-language features into each point. Then, inspired by how humans reason about physics through vision, we draw upon the semantic knowledge contained within large language models to obtain a set of candidate materials for each object. Finally, we estimate the physical properties of each point using a zero-shot retrieval-based approach and propagate the estimates across the entire object via spatial interpolation. Our method is accurate, annotation-free, and applicable to any object in the open world.

We evaluate NeRF2Physics on the task of mass estimation using the ABO dataset, as well as our own dataset of real-world objects with manually measured friction and hardness

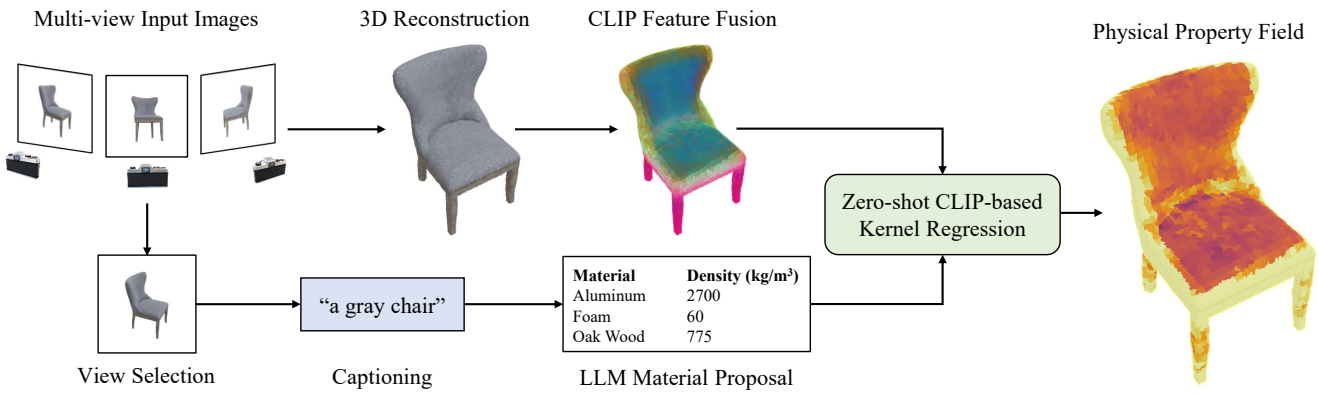

Figure 2. **Overview of NeRF2Physics.** Given a collection of posed images, we first train a neural radiance field to capture the 3D geometry of the scene. Then, we fuse vision-language features into a point cloud extracted from the field. Next, we use a captioning model to provide a text description of the scene and prompt an LLM to produce a dictionary of possible materials in the scene, along with their physical properties. From here, physical properties can be estimated at any query point using zero-shot CLIP-based kernel regression within the dictionary. The kernel regression process is illustrated in more detail in Fig. 3.

values. The results demonstrate that our method outperforms other zero-shot baselines and even supervised baselines for mass estimation. Visualizations of the predicted fields show that our approach can produce reasonable predictions of a variety of physical properties without supervision. In our ablation study, we compare our method with alternative approaches for language-driven physical property estimation.

2. Related Work

Visual physics reasoning. Reasoning about physical properties from visual data is a longstanding problem [2, 11, 38]. Studies have shown that deep learning models can potentially carry similar physical reasoning capabilities as humans, including estimation of object mass, friction, electric conductivity, and other material properties [37–39, 44]. Most of the prior work focuses on dynamic reasoning of object properties by either observing target dynamics [21, 38, 44] or directly interacting with the target in a 3D physical engine [29, 42]. A few studies have also tackled estimating material properties from static images directly [3, 4, 34, 35]. Although promising, existing work mostly tackles specific types of material properties, e.g., mass or tenderness, by collecting corresponding task-dependent data. In contrast, our method can generate diverse physical properties like mass density, friction, and hardness in a zero-shot manner from a single language-embedded feature field. Our work is greatly inspired by pioneering work from vision and cognitive science. Studies have found that humans are good at recognizing many material properties, e.g., thermal conductivity and hardness, from only visual inputs, even with only a brief demonstration [9, 10, 33]. Our work seeks to empower computational models with such perception capabilities of recognizing a diverse range of material properties.

Language grounding. CLIP is a large pre-trained vision-language model that can efficiently learn visual concepts from natural language supervision [30]. For training, CLIP jointly trains its image and text encoders to predict the correct pairings of image and text examples in a self-supervised fashion. Due to its success in capturing diverse visual concepts, CLIP has been widely used for zero-shot and few-shot tasks, spanning many applications from scene understanding [6, 16, 17, 28, 32] to texture generation [24] to language-grounded reasoning [12, 14, 19, 31, 40]. The models are trained via contrastive learning with a large amount of data and are thus capable of associating meaningful language-driven semantics with image patches. Our work leverages CLIP embeddings of small patches to provide a solid foundation for physical property reasoning in a zero-shot manner.

3. Neural Physical Property Fields

3.1. Overview

Our method takes as input a collection of posed images $\mathcal{I}$ and produces a physical property field $\rho(\mathbf{x})$ that can be queried to obtain physical property estimates at any occupied point within the scene. In this work, we focus on single-object scenes, but our approach can be extended to multiple-object scenes as long as segmentation masks are available. Inspired by how humans perceive and reason about physical properties of the objects they encounter, we propose to leverage language-vision embeddings as well as large language models (LLMs) to achieve this goal. Fig. 2 depicts an overview of our approach. First, we build a language-embedded point cloud from which per-point semantic features can be queried (Sec. 3.2). We then prompt an LLM to propose a dictionary $\mathcal{M}$ of candidate materials based on object semantics and apply zero-shot CLIP-based retrieval for reasoning about physical properties based on $\mathcal{M}$ (Sec. 3.3). Finally, for phys-

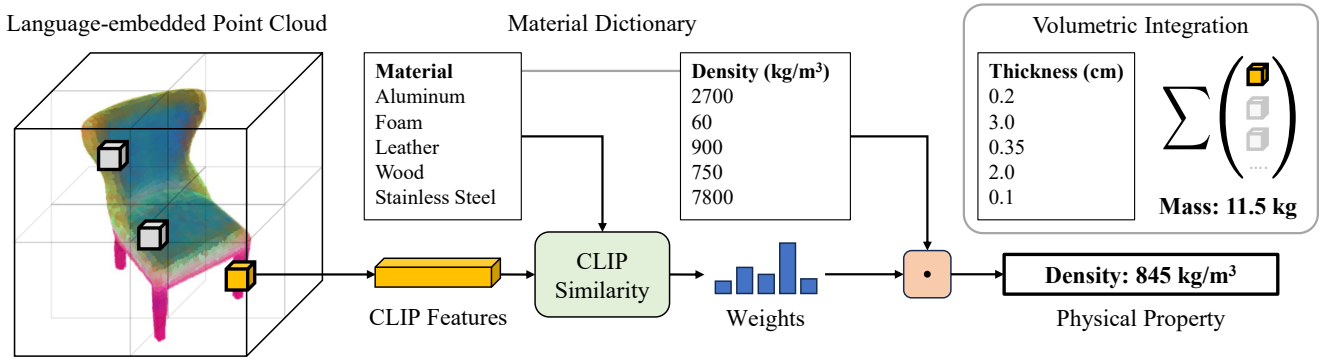

Figure 3. **Overview of zero-shot physical property prediction.** To predict physical property values from the language-embedded point cloud, we extract CLIP features and perform kernel regression using the predicted dictionary of materials and their properties. To predict the total mass of an object, we then integrate the predicted mass density across cuboids on the surface of the object. The thickness of each cuboid is estimated in the same way as the other physical properties.

ical properties that require volumetric integration, such as mass, we propose an integration method using LLM-based estimates of surface thickness (Sec. 3.4).

3.2. Language-embedded point cloud

Accurate estimation of physical properties, such as object mass, requires both accurate geometric and material understanding of the scene. To capture the 3D geometry of a scene, we train a neural radiance field [25] and extract a set of 3D "source" points via depth rendering. We then fuse per-patch CLIP [30] features into the source points using a simple visibility-aware averaging scheme.

Neural radiance field. For each scene, we first train a neural radiance field (NeRF) [25] on the given images and camera poses. We choose to use NeRF because it tends to give higher-quality, more robust depth maps compared to other 3D reconstruction methods, especially for reflective surfaces. We directly use the Nerfacto method from Nerfstudio [36], which combines several state-of-the-art techniques for improving performance. Once the NeRF is trained, we randomly sample a total of N rays from the input views and estimate the depth along each ray via median depth rendering [36]. Finally, we convert the depths into 3D points and perform voxel down-sampling to remove redundant points. The result is a point cloud $\mathcal{S} \subset \mathbb{R}^3$, where $|\mathcal{S}| \leq N$. We will refer to these points, which should cover all of the visible surfaces in the scene, as the *source points* for the scene.

3D language feature fusion. CLIP features have been shown to perform well in several zero-shot image classification tasks [27, 30], giving reason to believe that they can be successfully applied for material recognition. In order to enable 3D reasoning based on CLIP, one must aggregate CLIP features within a 3D representation of the scene. A number of works [6, 16, 28, 32] have proposed methods for 3D fusion of CLIP features that are conducive to object-level segmentation due to the use of object-level region proposals. However, these methods are not well-suited for discriminating between different materials *within the same object*, which is required for our use case.

In this work, we use simple averaging to fuse CLIP embeddings of small patches in the input images, which usually contain enough appearance information to discriminate between different materials. For each source point $\mathbf{s} \in \mathcal{S}$ and input image $\mathbf{I} \in \mathcal{I}$, we determine the pixel coordinates (u, v) of the point projected using the camera parameters of the image. We then test for occlusion using the NeRF-estimated depth to determine if the point is visible in the image. If it is visible, we extract a patch of size $P \times P$ centered at (u, v), and apply a CLIP image encoder to obtain a 512-dimensional feature vector for that patch. If segmentation masks are available, they can be applied here to focus on an object of interest. Once this is done for all of the input images and source points, the patch features are average-pooled to create a fused feature vector $\mathbf{z}$ for each source point $\mathbf{s}$.

3.3. Physical property reasoning

Our language-embedded point cloud contains rich semantic features for every source point in the 3D scene. Such features are usually tightly related to the physical properties of the object(s) in the scene (see Fig. 4). In this section, we propose a two-stage approach for estimating physical property values from the semantic features of the source points, which can then be propagated to any point in the continuous space by spatial interpolation. In the first stage, we prompt a VQA model to propose a dictionary of candidate materials along with their physical properties based on the input images. In the second stage, we perform a kernel regression over the materials in the dictionary for each source point using CLIP similarity in a zero-shot manner. Formally, for a given point $\mathbf{s}$ with semantic embedding $\mathbf{z}$, we formulate its physical

property as

$$\rho(\mathbf{s}) = F(\mathbf{z}, \mathcal{M}) = F(\mathbf{z}, G(\mathcal{I})), \tag{1}$$

where G is the mapping from the input images $\mathcal{I}$ to candidate materials $\mathcal{M}$, and F is the mapping from semantic features and candidate materials to our desired physical property.

LLM-based material proposal. The set of different materials that exist in the world is extremely large and difficult to define. Furthermore, many materials look identical and thus cannot be distinguished by local appearance alone. Despite this, humans are able to guess the material composition of objects through high-level reasoning about object semantics on top of low-level appearance cues. Inspired by this, our method calls upon LLMs for open-vocabulary semantic reasoning about materials and their physical properties.

Given a set of input images $\mathcal{I}$, we first select a canonical view $\mathbf{I}_0 \in \mathcal{I}$. If segmentation masks are available, we calculate the area of the mask in each frame and select the frame at the 75th percentile (as a heuristic method to obtain an informative view). If masks are not available, we select a view uniformly at random. We then use a VQA model (BLIP-2 [20]) to produce a text description of $\mathbf{I}_0$. Finally, we pass this description to an LLM (GPT 3.5) and prompt it to return a dictionary of K candidate materials $\mathcal{M} = \{(\texttt{key}_k, y_k)\}$, where $\texttt{key}_k$ is the material name text and y_k is the value of the physical property, e.g. $\{(\texttt{Aluminum}, 2700\texttt{kg/m}^3), (\texttt{Oak Wood}, 650 - 900\texttt{kg/m}^3)\}$. Note that y_k may be a range of values due to the inherent uncertainty in the task.

Although it is theoretically possible for a VQA model such as BLIP-2 [20], LLaVA [22], or GPT-4V [41] to propose the materials directly from the image, we find that decomposing the task into two parts produces more reliable results in our experiments. In the future, as VQA models become more powerful, one model may be sufficient.

Zero-shot CLIP-based kernel regression. Once the material dictionary $\mathcal{M}$ has been obtained, we use CLIP features to perform material retrieval for each source point. We predict the physical property values of each point by taking its fused CLIP features and conducting a CLIP-based kernel regression using the material dictionary. Formally speaking, the physical property field is equal to:

$$\rho(\mathbf{s}) = F(\mathbf{z}, \mathcal{M}) = \frac{\sum_{k=1}^{K} \exp(w_k[\mathbf{s}]/T) y_k}{\sum_{k=1}^{K} \exp(w_k[\mathbf{s}]/T)}, \tag{2}$$

where $w_k[\mathbf{s}] = \psi_{\text{CLIP}}(\mathbf{z}, \texttt{key}_k)$ is the cosine similarity between semantic feature $\mathbf{z}$ and the language CLIP feature of the material name $\texttt{key}_k$, and T is a temperature parameter chosen through validation. The physical property values can be propagated from the source points to any 3D query point via nearest-neighbor interpolation:

$$\rho(\mathbf{x}) = \rho(\text{argmin}_{\mathbf{s} \in \mathcal{S}} ||\mathbf{s} - \mathbf{x}||). \tag{3}$$

3.4. Object-level physical property aggregation

So far, we have discussed dense prediction of physical properties in a per-point manner. In practice, one may also be interested in object-level physical properties (e.g. mass) that require integration over volumes. Although NeRF gives us an estimate of the geometry of the visible surfaces in the scene, most objects contain a large amount of empty space in their interior, which heavily affects volumetric integration but cannot be captured by NeRF due to occlusion. To circumvent this, we prompt the large language model again to estimate the thickness t_k of each material in $\mathcal{M}$, and then use the estimated thickness to define a set of cuboids sampled on the object's surface. Similar to the source point sampling, the cuboid locations are sampled by voxel-downsampling surface points, and we define their size to be $d \times d \times \tau$, where d is the voxel size and τ is the predicted thickness at that point. Formally, the prediction $\hat{m}$ for the integral over the cuboids is given by

$$\hat{m} = \sum_{\mathbf{x} \in \mathcal{V}} \frac{\sum_{k=1}^{K} \exp(w_k[\mathbf{x}]/T) y_k t_k}{\sum_{k=1}^{K} \exp(w_k[\mathbf{x}]/T)} d^2, \tag{4}$$

where $\mathcal{V}$ is the set of voxel-downsampled points. Since this volume estimation can be biased depending on the geometry of the object, we introduce a scalar multiplication factor c determined by validation. We also clamp the total volume to an upper bound estimated by depth carving to avoid wildly inaccurate thickness predictions.

4. Experiments

We evaluate NeRF2Physics across two datasets. First, we compare NeRF2Physics with existing methods for per-object mass estimation on a set of 500 objects from the Amazon Berkeley Objects (ABO) dataset [7], which we refer to as ABO-500. To evaluate the dense prediction capabilities of NeRF2Physics, we collect our own dataset of real-world objects with per-point friction and hardness measurements.

4.1. Implementation Details

Our NeRF directly uses the Nerfacto method from Nerfstudio [36] with default settings except with the near-plane for sampling set to 0.4, the far-plane set to 6.0, and the background color set to random. For our own dataset, we set the scene scale to 2.0. The camera poses are scaled per scene to fit in a ± 1 box. We train each scene for 20K iterations, which takes around 8 minutes on an NVIDIA A40 GPU.

For source point extraction, we sample $N = 100\,000$ rays, voxel-downsample with a grid size of 0.01 (0.02 for

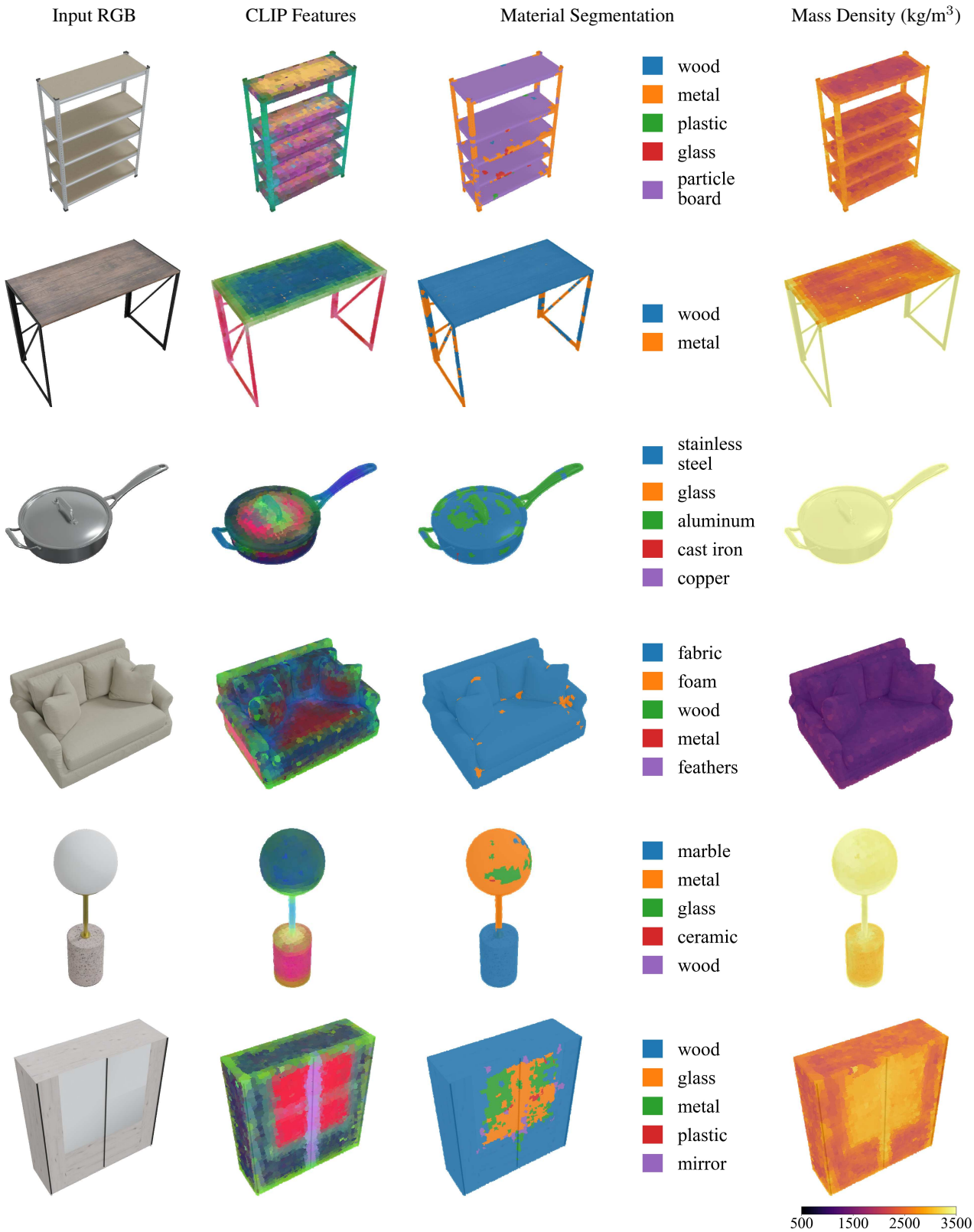

Figure 4. **Example visualizations.** We visualize input images from ABO-500 along with our model's CLIP feature PCA components, zero-shot material segmentation, and predicted mass density. Our model makes reasonable predictions of materials across different parts of objects in 3D, allowing for grounded predictions of physical properties.

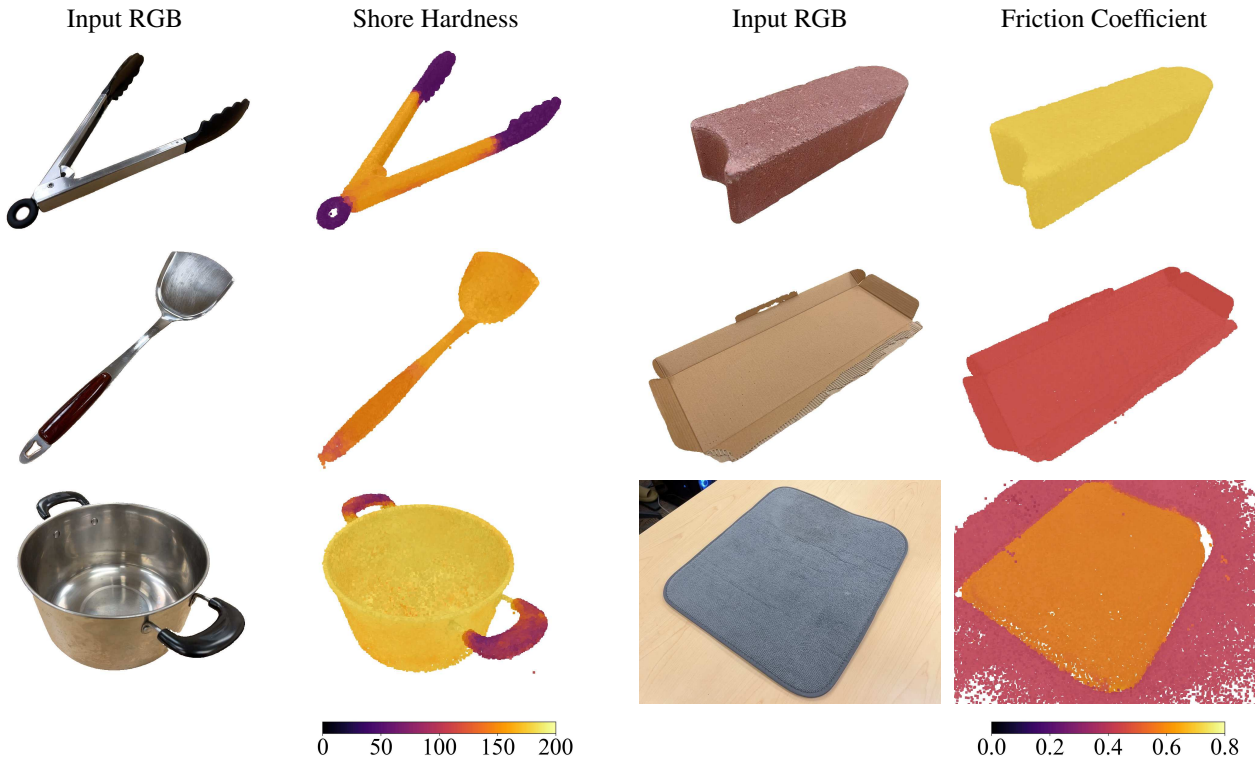

Figure 5. **Example predictions of different physical properties.** We visualize predictions of hardness and friction on objects from our own collected dataset. For evaluation purposes, Shore A and Shore D hardness was combined into the same scale. The friction coefficient represents the coefficient of kinetic friction against a fabric surface. We quantitatively evaluate these predictions using a set of sparse per-point measurements (see Sec. 4.3).

our own dataset), and remove outliers (see supplementary for details). For language feature fusion, we use the OpenCLIP [15] ViT B-16 model trained on DataComp-1B and set the patch size to $P = 56$ and the occlusion threshold to 0.01. For captioning, we use BLIP-2-Flan-T5-XL [20]. For mass density (and thickness), we use GPT-3.5 Turbo [43] and set $K = 5, T = 0.1$. For friction and hardness, we use GPT-4 [5] and set $K = 3, T = 0.01$. The exact prompts we used can be found in the supplementary. For mass integration, we voxel-downsample with a grid size of 0.005, carve with a grid size of 0.002, and scale the final mass by $c = 0.6$. Since our model usually returns a range of values, we take the center of the range as the final prediction.

4.2. Mass Estimation

Dataset. The ABO dataset [7] contains thousands of products sold on Amazon together with multi-view posed images, segmentation masks, mass measurements, and other product metadata. In order to create a diverse evaluation set, we created a stratified sample of 500 objects in which each "product_type" (e.g. "chair", "lamp") appeared no more than 10 times. Each object/scene has 30 views facing the object with camera centers randomly distributed over a hemisphere around the object. We call this dataset ABO-500 and split the scenes randomly into 300 train / 100 val / 100 test.

Metrics. We follow pioneering work on visual mass estimation [35] and report the following metrics, where m is the ground-truth mass and $\hat{m}$ is the estimated mass:

- Absolute difference error (ADE) : $|m - \hat{m}|$,
- Absolute log difference error (ALDE) : $|\ln m - \ln \hat{m}|$,
- Absolute percentage error (APE) : $|\frac{m-\hat{m}}{m}|$, and
- Min ratio error (MnRE) : $\min(\frac{m}{\hat{m}}, \frac{\hat{m}}{m})$.

We agree with the authors of [35] that MnRE is the preferred metric, because it is not biased towards models that systematically over- or under-estimate and also does not over-emphasize performance on heavier instances.

Baselines. We compare NeRF2Physics with the following baselines on the ABO-500 dataset:

- **Image2mass** [35] uses a CNN to predict mass directly from a single image and 3D bounding box dimensions. We evaluate the official model pretrained on Amazon products and use bounding boxes extracted from our source points.
- **2D CNN** takes a frozen ResNet50 [13] pretrained on ImageNet [8] and trains three addtional layers to predict mass on our dataset. We apply a negative LogSigmoid layer to ensure that the predictions are positive.
- **LLaVA** [22] is a large vision-language model that is designed to follow arbitrary instructions given an image. We

Table 1. Mass estimation on ABO-500 test set (100 objects). ADE is measured in kilograms. **Bold**: best model.

Method	ADE (↓)	ALDE (↓)	APE (↓)	MnRE (↑)
Image2mass [35]	12.496	1.792	**0.976**	0.341
2D CNN	15.431	1.609	14.459	0.362
LLaVA [22]	17.328	1.893	1.837	0.306
Ours	**8.730**	**0.771**	1.061	**0.552**

Table 2. Ablation study for mass estimation on ABO-500 val set (100 objects). **Bold**: best model.

Method	ADE (↓)	ALDE (↓)	APE (↓)	MnRE (↑)
No thickness	18.587	0.749	1.364	0.552
Retrieval ($T \to 0$)	12.266	0.780	**0.801**	0.536
Uniform CLIP	10.396	0.637	1.102	0.597
Ours	**9.786**	**0.610**	0.931	**0.609**

prompt LLaVA to estimate the mass of the object in the image (see supplementary for exact prompt). [1]

For all of the baselines, we provide the same canonical view as in our method, in which the background is set to white.

Qualitative Results. We show example visualizations of our language-embedded point cloud and material predictions in Fig. 4. CLIP features were converted to RGB values according to the top 3 PCA components per object. The PCA visualization suggests that the CLIP features give enough information to perform material segmentation. The material visualization shows that our method can propose reasonable candidate materials and use the CLIP features to identify the primary material in different parts of an object, such as metal in the legs of a table and wood on the tabletop. However, the boundaries of each part are not localized perfectly, and the model will often mix similar materials together (e.g."stainless steel" and "aluminum"). The last column of visualizations show that sensible mass density estimates follow from the material predictions.

Quantitative Results. We report test-set mass estimation metrics on ABO-500 in Tab. 1. Mass predictions for all models were clipped to be between 0.01 and 100 kilograms. The image2mass [35] pretrained model performs poorly since it does not generalize well to objects larger than those found in its training data. The 2D CNN baseline also did not perform well – it failed to learn meaningful patterns and tended to predict towards the mean of the dataset. The LLaVA [22] model usually gives answers that are not metrically precise (e.g. 1 kg or 10 kg), despite extensive prompt engineering. Our zero-shot method's predictions outperform these baselines by a large margin in all metrics except APE. We note that APE is heavily biased towards models that underestimate, as it is dominated by overestimates on small objects.

Ablation Studies. We perform ablations on various aspects of our method in Tab. 2. First, we remove the thickness estimation step and integrate over occupied voxel produced by depth-based carving. This results in consistent overestimation since it ignores the fact that many objects have empty space in their interior. Next, we examine the effect of performing kernel regression instead of just retrieving the most likely material per point (effectively setting the temperature T to zero). Here, the retrieval performs worse because there is inherent uncertainty in predicting materials based on visual appearance. Lastly, we evaluate the use of a single global CLIP embedding of the canonical view instead of fused patch embeddings, which gives a uniform prediction across the whole object. We find that the performance is only slightly worse, suggesting that the total mass for most objects is dominated by a single material.

[1] We also tried to apply GPT-4V for this task but had difficulty preventing it from producing complaints about not having enough information.

4.3. Friction and Hardness Estimation

Dataset. The task of mass estimation does not directly evaluate the ability of our model to perform dense prediction of different physical property values within the same object. To the best of our knowledge, there does not exist a realistic dataset with images and paired measurements suitable for this purpose. Thus, we collect our own dataset containing 15 household objects across 13 scenes with real-world multi-view images and paired measurements of per-point kinetic friction coefficient and Shore hardness. The images and poses were captured using Polycam on an iPhone 13 Pro, with a median of 82 per scene. Coefficient of kinetic friction was collected on 6 surfaces using an iOLab with fabric pads attached, averaging over 10 trials. Shore hardness was collected at 31 points using Gain Express A/D durometers, averaging over 3 trials with Shore D being used when the Shore A reading was above 90. Each point's location is annotated as pixel coordinates in an image. Grounding SAM [18, 23] was used to obtain object masks.

Note that Shore A and Shore D durometers use different indenters and thus do not measure exactly the same physical property [26]. However, for evaluation purposes, we combine the measurements into a single scale from 0-200, where the Shore A measurements lie in the 0-100 range and Shore D measurements lie in the 100-200 range.

Metrics. We report the same metrics as before, along with an additional metric of Pairwise Relationship Accuracy (PRA), defined as the classification accuracy of the predicted relationships (greater than or less than) between every pair of points. This metric focuses on relative compar-

Table 3. Estimation of per-point Shore hardness on the real-world in-house collected dataset (31 points, 11 objects). **Bold**: best model.

Method	ADE (↓)	ALDE (↓)	APE (↓)	MnRE (↑)	PRA (↑)
GPT-4V	**32.752**	0.330	0.304	0.758	0.609
CLIP	32.857	**0.294**	**0.266**	**0.774**	0.647
Ours	34.295	0.315	0.276	0.765	**0.710**

Table 4. Estimation of per-point kinetic friction coefficient on the in-house collected dataset (6 points, 6 objects). **Bold**: best model.

Method	ADE (↓)	ALDE (↓)	APE (↓)	MnRE (↑)	PRA (↑)
GPT-4V	0.209	0.430	0.549	0.692	0.467
CLIP	0.222	0.455	0.602	0.654	0.533
Ours	**0.155**	**0.321**	**0.360**	**0.736**	**0.800**

isons and is thus more robust to measurement noise, which is especially significant for the hardness measurements due to local deformations in the object surface around each point.

Baselines. There are no existing methods for predictions of arbitrary physical properties from images, so we design the following baselines for comparison:

- **GPT-4V** [41] is a large vision-language model that can accept masks in its prompt. For each point, we provide GPT-4V with the associated image and a mask highlighting its pixel location, and ask it to estimate the physical property at that point.
- **CLIP** refers to using a global CLIP embedding of the canonical view instead of fused patch features in our method. This was also considered in our ablations above.

We instruct each LLM to choose Shore A/D hardness based on which is more appropriate for the material in question.

Qualitative Results. We show example predictions of hardness and friction from our model in Fig. 5. Again, the model is able to distinguish different materials and derive reasonable physical property estimates from them, even for unusual objects such as the ripped piece of cardboard. In addition, the model is fairly robust to errors in the geometry from NeRF, thanks to our feature fusion strategy. The example with the bath mat demonstrates that our method can be applied with or without object segmentation masks.

Quantitative Results. We report quantitative evaluation metrics for hardness prediction in Tab. 3 and for friction prediction in Tab. 4. For hardness, we observe that all three models perform similarly across most of the metrics, but ours achieves the highest PRA, indicating that it localizes different materials more precisely than the other models. For friction, we find that our model outperforms the others by a wide margin in all of the metrics. GPT-4V performs similarly with the uniform CLIP model, suggesting that it has trouble distinguishing between different surfaces within the same scene. Also note that GPT-4V must run on each individual query point, which is extremely computationally expensive. In contrast, once the feature field and candidate materials for our model have been prepared, thousands of points can be queried with little computational cost.

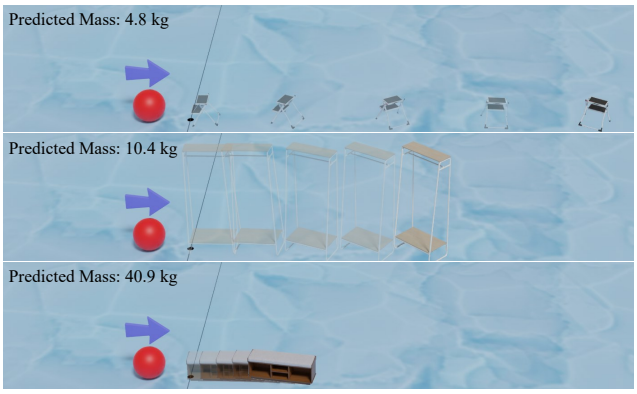

Figure 6. **Digital twins with realistic physical properties.** We show that realistic physical interactions can be simulated using mass-aware digital twins created by NeRF2Physics. In each example trajectory visualization here, the ball hits the object with the same initial momentum, and friction is zero.

4.4. Applications

NeRF2Physics can be applied to create physically realistic digital twins for immersive computing and content creation (Fig. 6). Improved physical property understanding is also crucial for advancing embodied AI and robot simulation. Another application is estimating crop biomass [1], which is important for agriculture but labor-intensive and destructive to measure manually.

5. Conclusion

We presented NeRF2Physics, a novel method for dense prediction of physical properties from a collection of images. Our method fuses vision-language embeddings into a 3D point cloud and leverages LLMs to provide material information, enabling zero-shot estimation of any physical property for any object in the open world. Experimental results demonstrate that our method outperforms baselines on estimation of mass, hardness, and friction coefficients across a variety of objects. In the future, our approach may be improved by incorporating prior knowledge to reason about materials in internal object parts that cannot be seen.

Acknowledgments. This project is supported by the Intel AI SRS gift, the IBM IIDAI Grant, the Insper-Illinois Innovation Grant, the NCSA Faculty Fellowship, the Agroecosystem Sustainability Center at UIUC, and NSF Awards #2331878, #2340254, and #2312102. We greatly appreciate NCSA for providing computing resources.

References

[1] Estimating cover crop biomass. https://www.nrcs.usda.gov/sites/default/files/2022-09/EstBiomassCoverCrops_Sept2018.pdf. Accessed: 2023-11-17. 8

[2] Edward H Adelson. On seeing stuff: the perception of materials by humans and machines. In *Human vision and electronic imaging VI*. SPIE, 2001. 2

[3] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *SIGGRAPH*, 2014. 2

[4] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *CVPR*, 2015. 2

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 6

[6] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *ICRA*, 2023. 2, 3

[7] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022. 4, 6

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[9] Roland W. Fleming. Visual perception of materials and their properties. *Vision Research*, 2014. 1, 2

[10] Roland W Fleming, Christiane Wiebel, and Karl Gegenfurtner. Perceptual qualities and material classes. *Journal of vision*, 2013. 1, 2

[11] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *ICLR*, 2016. 2

[12] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. Clip on wheels: Zero-shot object navigation as object localization and exploration. *arXiv preprint arXiv:2203.10421*, 2022. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[14] Yining Hong, Chunru Lin, Yilun Du, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. 3d concept learning and reasoning from multi-view images. In *CVPR*, 2023. 2

[15] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. https://github.com/mlfoundations/open_clip. 6

[16] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023. 2, 3

[17] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language embedded radiance fields. In *ICCV*, 2023. 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 7

[19] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2

[20] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 4, 6

[21] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-NeRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *ICLR*, 2022. 2

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 4, 6, 7

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 7

[24] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022. 2

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3

[26] Magdy I Mohamed and Gamal A Aggag. Uncertainty evaluation of shore hardness testers. *Measurement*, 33(3):251–257, 2003. 7

[27] Zachary Novack, Julian McAuley, Zachary Lipton, and Saurabh Garg. Chils: Zero-shot image classification with hierarchical label sets. In *ICML*, 2023. 3

[28] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. *arXiv preprint arXiv:2211.15654*, 2022. 2, 3

[29] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *ECCV*, pages 3–18. Springer, 2016. 2

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3

[31] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *CoRL*, 2023. 2

[32] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022. 2, 3

[33] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 2009. 2

[34] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *IJCV*, 2013. 2

[35] Trevor Standley, Ozan Sener, Dawn Chen, and Silvio Savarese. image2mass: Estimating the mass of an object from its image. In *CoRL*, 2017. 2, 6, 7

[36] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *SIGGRAPH*, 2023. 3, 4

[37] Manik Varma and Andrew Zisserman. A statistical approach to material classification using image patch exemplars. *TPAMI*, 2008. 2

[38] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *NeurIPS*, 2015. 2

[39] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*, 2016. 2

[40] Yuefan Wu, Zeyuan Chen, Shaowei Liu, Zhongzheng Ren, and Shenlong Wang. Casa: Category-agnostic skeletal animal reconstruction. *NeurIPS*, 2022. 2

[41] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9, 2023. 4, 8

[42] Shaoxiong Yao and Kris Hauser. Estimating tactile models of heterogeneous deformable objects in real time. In *ICRA*, 2023. 2

[43] Junjie Ye, Xuanting Chen, Nuo Xu, Can Zu, Zekai Shao, Shichun Liu, Yuhan Cui, Zeyang Zhou, Chao Gong, Yang Shen, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *arXiv preprint arXiv:2303.10420*, 2023. 6

[44] Ilker Yildirim, Jiajun Wu, Yilun Du, and Joshua B Tenenbaum. Interpreting dynamic scenes by a physics engine and bottom-up visual cues. 2