Evaluating VLMs for Score-Based, Multi-Probe Annotation of 3D Objects

Rishabh Kabra[1 2]**, Loic Matthey**[1]**, Alexander Lerchner**[1]**, Niloy J. Mitra**[2]
[1]Google DeepMind, [2]University College London
{rkabra, lmatthey, lerchner}@google.com, n.mitra@ucl.ac.uk

Abstract

Unlabeled 3D objects present an opportunity to leverage pretrained vision language models (VLMs) on a range of annotation tasks—from describing object semantics to physical properties. An accurate response must take into account the full appearance of the object in 3D, various ways of phrasing the question/prompt, and changes in other factors that affect the response. We present a method, to marginalize over arbitrary factors varied across VLM queries, which relies on the VLM's scores for sampled responses. We first show that this probabilistic multi-probe aggregation can outperform a language model (e.g., GPT4) for summarization, for instance avoiding hallucinations when there are contrasting details between responses. Secondly, we show that aggregated annotations are useful for prompt-chaining; they help improve downstream VLM predictions (e.g., of object material when the object's type is specified as an auxiliary input in the prompt). Such auxiliary inputs allow ablating and measuring the contribution of visual reasoning over language-only reasoning. Using these evaluations, we show that VLMs approach the quality of human-verified annotations on both type and material inference on the large-scale Objaverse dataset.

1 Introduction

An abundance of text and visual paired data has powered the rise of powerful representation learning algorithms and generative models for images. In the 3D community, we are yet to see a comparable modeling revolution [1, 2] despite growing repositories of object models, character assets, and 3D scans. A key blocker is the lack of clean text annotations.

Synthetic annotation pipelines stand ready to fill the gap. 2D vision language models [3–6]–by virtue of being trained on a nontrivial fraction of human knowledge–contain rich information about the nature of all things. While we'd hope for their description prowess to translate to 3D, this comes with at least three challenges: (i) There can be discrepancies between multiple views of the same object (e.g. when viewed from the front or side). These need to be reconciled to produce an aggregate answer or description. (ii) It's unclear what to optimize to generate multipurpose annotations, and for that reason, how to tune the way we probe a VLM. This issue is compounded by the limited availability of ground truth or human annotations, which cannot be collected at the same rate at which VLMs can respond to arbitrary queries. (iii) While language is well suited to describing object semantics (i.e. type), human vocabulary is stretched when describing concepts like material or shape of particular interest in 3D. Unstructured captioning is unlikely to capture such properties consistently.

We make the case for property-specific annotations (PSAs) using structured visual question answering. In contrast to a descriptive blurb or caption, a bite-sized value can be easily compared or used for indexing/search. For that reason, such annotations are easier to aggregate across multiple occurrences (e.g. different views of an object or paraphrases of a question). With structured VQA, we have a better chance of collecting and evaluating VLM responses for less linguistic concepts. PSAs also

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

make it easier to intervene on the value of variables (e.g. by specifying them in the prompt) to probe VLMs in a causally driven manner.

To address the challenges of evaluating a VLM annotation pipeline, we explore two evaluation strategies: In Sec 3, we study the variation in VLM responses under changes in view or prompt, and how best to summarize responses reliably. In Sec 4, we assess the usefulness of an inferred property for the downstream task of inferring another property. We run the downstream inference in LLM and VLM modes. Besides their different evaluation focus, the two sections also focus on different object properties: semantic type and material respectively.

Our salient contributions are the following–we:

1. Run 55B-parameter variants of PaLI [7] to generate captions and property-specific annotations on the Objaverse [8] dataset.
2. Introduce a likelihood-based probabilistic aggregation of VLM responses across object views and multiple queries.
3. Compare our annotations and aggregation method with concurrent work based on GPT4 (CAP3D [9]) and baseline sources.
4. Show the value of aggregate, structured intermediate representations for downstream inference in VLMs, akin to chain-of-thought reasoning.
5. Plan to release our outputs at `https://github.com/google-deepmind/objaverse_annotations`.

2 Background

Dataset. The main target of our work is Objaverse 1.0 [8], a collection of 800K diverse but poorly annotated 3D models uploaded by 100K artists to the Sketchfab platform. While the tags and descriptions uploaded by artists are noisy and unreliable, a subset of 47K objects called Objaverse-LVIS is accompanied by human-verified categories. We rely on it to validate our semantic annotations.

Related work. A three-stage pipeline was proposed to generate captions for Objaverse concurrently to our work. Although our objective—to produce property-specific annotations—is meaningfully different, we rely on CAP3D [9] as the primary baseline for our work. The pipeline is as follows: a VLM (BLIP-2 [3]) first produces 5 candidate captions for 8 different object views; CLIP [6] filters all but one caption per view, and GPT4 [10] performs a final detail-preserving but hallucination-prone aggregation. Our procedure is similar up to CAP3D's first stage. We show major flaws with CAP3D's aggregation step in Sec 3.

Models. To generate our own captions or annotations, we rely on two variants of PaLI-X fine-tuned specifically for captioning or visual question answering. Both variants consist of a ViT-22B [11] vision model and 32B UL2 [12] language backbone. For the material prediction task, we also run BLIP-2 T5 XL as a baseline. All models are run zero-shot, one input image at a time, and output an autoregressive distribution over language tokens. The likelihood of any sampled text can be computed during the VLM sampling process (e.g. beam search) without any additional cost. None of our methods or results are specific to PaLI or BLIP.

3 Semantic Descriptions

To ask for the type of an object is the most language-amenable VLM query. Despite this, we illustrate the challenge of captioning a 3D object in Fig 1-A. Concurrent work (CAP3D [9]) to produce captions for Objaverse relies on GPT4 [10] to summarize annotations across multiple views of an object. This can produce deeply flawed summaries. The LLM propagates hallucinations or confusions when there's contrasting captions among views. Despite being instructed that it is given captions of the same object, the LLM tries to preserve details across views rather than reconcile them (see Fig 1-B).

To address this, we propose an alternative method of aggregating multi-view or multi-query annotations in Sec 3.1. We then compare semantic descriptions from baseline sources with annotations produced by our method in Sec 3.2. Finally, we unpack the performance of our aggregation relative to individual views or queries in 3.3.

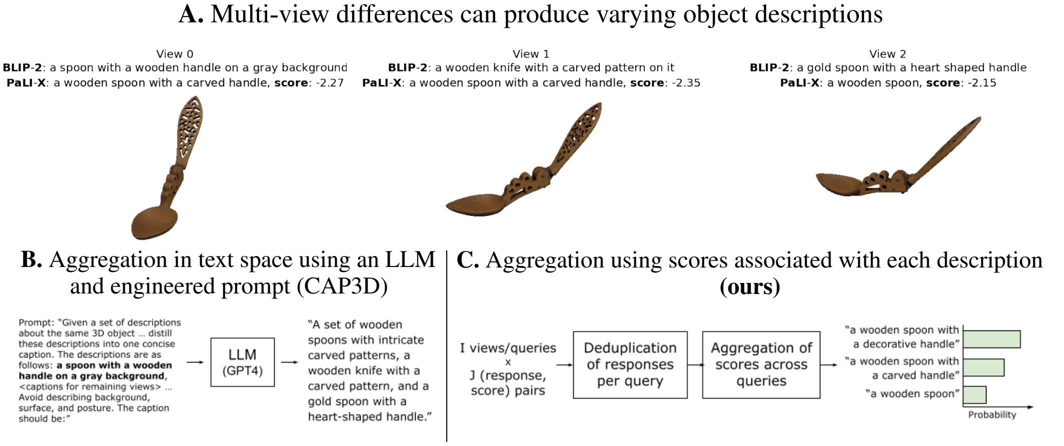

A. Multi-view differences can produce varying object descriptions
View 0
BLIP-2: a spoon with a wooden handle on a gray background
PaLI-X: a wooden spoon with a carved handle, score: -2.27
View 1
BLIP-2: a wooden knife with a carved pattern on it
PaLI-X: a wooden spoon with a carved handle, score: -2.35
View 2
BLIP-2: a gold spoon with a heart shaped handle
PaLI-X: a wooden spoon, score: -2.15
B. Aggregation in text space using an LLM and engineered prompt (CAP3D)
Prompt: "Given a set of descriptions about the same 3D object ... distill these descriptions into one concise caption. The descriptions are as follows: a spoon with a wooden handle on a gray background, <captions for remaining views> ... Avoid describing background, surface, and posture. The caption should be:"
LLM (GPT4)
"A set of wooden spoons with intricate carved patterns, a wooden knife with a carved pattern, and a gold spoon with a heart-shaped handle."
C. Aggregation using scores associated with each description (ours)
I views/queries x J (response, score) pairs
Deduplication of responses per query
Aggregation of scores across queries
"a wooden spoon with a decorative handle"
"a wooden spoon with a carved handle"
"a wooden spoon"
Probability

Figure 1: **A.** Three of eight regularly spaced views of a 3D object. Each view is accompanied by the top caption produced by two different models: BLIP-2 and PaLI-X. Captions from BLIP-2 were obtained from the CAP3D baseline, whereas captions from PaLI-X were generated with accompanying scores for this work. Both models show an expected variation in responses across views. **B.** To aggregate multi-view captions, CAP3D feeds them to GPT4 and prompts it for an object-level summary. The LLM is often unable to denoise captions from contrasting views, and simply "adds up" the contents. **C.** Our aggregation helps surface more reliable responses, weighting them based on their combined scores across views.

3.1 Aggregation of Responses

We introduce an aggregation for VLM outputs across multiple queries that relies on the log-likelihoods or scores of the sampled outputs. When VLM queries are correlated (e.g. views of the same object or paraphrased questions), we can expect recurring responses across queries. Say we run I queries to get J (response, score) pairs per query, for a total of IJ pairs $\{(r_{i,j}, s_{i,j})\}$. Let f be a map to post-process strings and reduce them to a canonical form. The following aggregation helps identify responses which occur frequently while accounting for the model's confidence in each occurrence:

$$\forall r \in \{r_{i,j}\}, \quad s_i(r) := \sup\{s_{i,j} \mid f(r_{i,j}) = r \text{ and } j = 1, 2, ..., J\} \tag{1}$$

$$s_{agg}(r) := \log \sum_i \exp(s_i(r)) \tag{2}$$

$$p(r|\{r_{i,j}, s_{i,j}\}) := \exp(s_{agg}(r)) / \sum_{r'} \exp(s_{agg}(r')) \tag{3}$$

Equation 1 deduplicates responses for a given VLM query i. The string processor f determines when $r_{i,j}$ is treated equivalent to r, and can be customized per VLM. This is useful when responses are identical up to punctuation, case, or uninformative tokens. Since these are undesirable duplicates, we want to avoid accumulating their scores, so we take the supremum instead. Note that $s_i(r)$ can be $-\infty$ if no r equivalent occurs in the J responses for query i.

Equation 2 then aggregates scores across occurrences of r in distinct queries. These are desirable duplicates (over distinct images or prompts) which merit reinforcing. Finally, equation 3 computes an aggregate probability distribution over responses by taking a softmax over the aggregate scores.

In contrast to model-based summarization (e.g., using an LLM), this aggregation requires a trivial numerical computation. There's no scoring cost in addition to generating the outputs; most VLM sampling methods can output the score simultaneously. Whereas an LLM needs a prompt specific to the aggregation task, our method can be used on arbitrary VLM responses that need aggregating. While an LLM produces a point estimate, our method outputs a distribution over all possible responses.

3.2 Comparative Evaluation

We collect four sets of semantic descriptions for Objaverse :

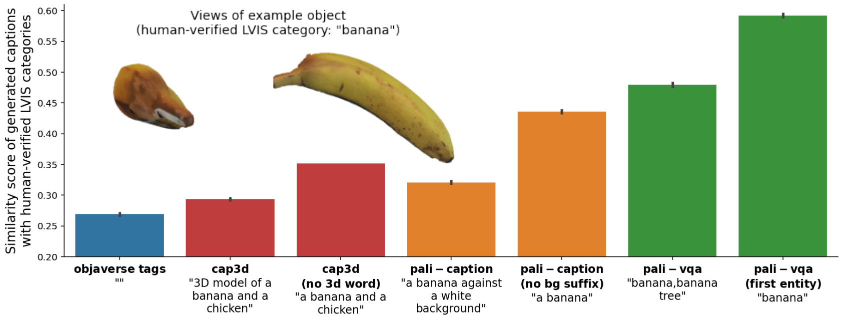


Figure 2: **Comparison of captions and type annotations generated from different sources/models.** The bars show cosine similarity scores (↑) computed in an independent text-embedding space and averaged over the Objaverse-LVIS subset. We show an example caption/type annotation beneath each bar; these correspond to a fixed object shown in the top-left corner from two different views.

1. **Objaverse tags**: these were uploaded by the creator of each 3D asset and are available as part of the original dataset. They are inherently noisy and inconsistent between objects. We comma-separate the tags to produce a concatenated string for each object.
2. **CAP3D captions**: these were generated and released by [9]. A post-processed version of the captions removes the frequent prefix, "3D model of." We compare both versions.
3. **PaLI captions**: using a captioning-specific variant of PaLI and simple prompt ("A picture of "), we generated descriptive captions similar to CAP3D's first stage. We then applied our aggregation to summarize $J = 5$ responses across $I = 8$ views per object. We compare results with and without a post-processing map f (Eq 1) to ignore suffixes of the form "on/against a white background."
4. **PaLI VQA annotations**: we used 4 VQA prompts to probe for the type of each object: (i) What is this? (ii) What type of object is this? (iii) What is in the image? (iv) Describe the object in the image. This produced 4 sets of top-5 responses per view. The responses are typically WordNet [13] entities that group synonyms or related terms in a comma-separated list. We deduplicate responses by taking the first such term per response. This post-processing map is also ablated.

We compare outputs from these sources to human-verified object categories from the Objaverse-LVIS subset. For sources that use our aggregation method, we take the likeliest output from each aggregate distribution. We proceed to embed all text using an independent language encoder, namely the Universal Sentence Encoder (v4) [14] from TensorFlow-Hub. Then, we compute cosine similarities between the embedded outputs and human-verified categories.

Fig 2 shows that all VLM pipelines outperform the tags from the original dataset. PaLI captions, with our likelihood-based aggregation, are slightly better than (the three-stage) CAP3D captions. Our PaLI VQA annotations perform significantly better. We will unpack the role of aggregating across multiple questions and object views in the following subsection.

3.3 Why Aggregate

To show why our aggregation works, we look at the individual views and queries that comprised our PaLI VQA annotations. Fig 3 shows the effect of aggregating across various slices of the views and questions presented to the VLM. We also compare our default log-sum-exp (LSE) aggregation (Eq 2) with the simpler choice of taking the maximum-score response across all views/questions.

There's a small but significant gap between the LSE and maximum-score aggregations. The latter performs worse than several views individually, because overconfident responses might dominate the aggregate. The LSE aggregation performs better than any individual view.

Comparing different questions, there is in fact a particular question which serves as the best VLM prompt for our current evaluation metric (cosine similarity with respect to LVIS categories). Including less optimal questions in our aggregation does not improve the score. Nevertheless it smoothens the aggregate response distribution and widens the support. We show this qualitatively in Fig 4. Aggregating across questions helps avoid mode collapse in bimodal cases (such as the bee on the

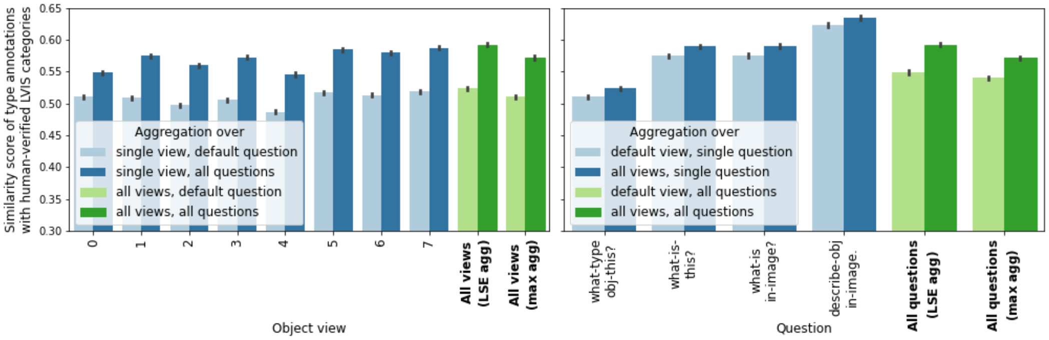


Figure 3: **Likelihood-based aggregation across views and across questions.** Each bar computes an aggregate output distribution from a different subset of responses. The mode of each aggregate output distribution is scored using cosine similarity on Objaverse-LVIS as before. The left plot scores 8 individual views versus the aggregate of all views, while highlighting the gap between asking a default question or multiple question variants. The right plot scores 4 individual questions versus the aggregate of all questions, while highlighting the gap between using a fixed object view or all views.

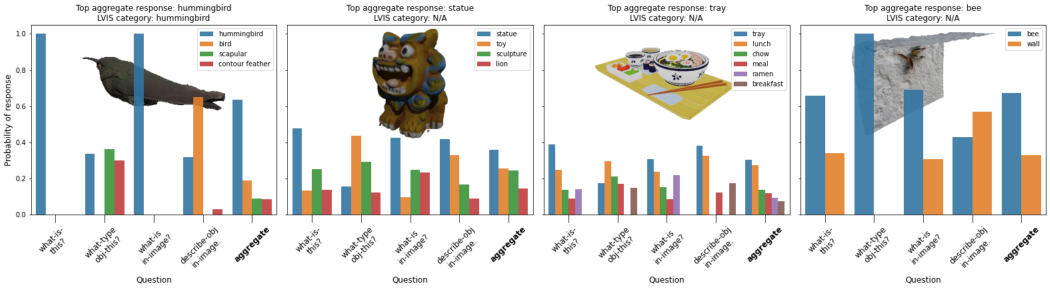


Figure 4: **Histograms showing PaLI VQA responses per question and after aggregation** for fixed views of a selection of objects. To reduce visual clutter, we filtered responses with scores below a fixed threshold (-1.2). Each subplot legend lists the possible responses sorted by aggregate probabilities. For comparison, we show the object's LVIS category where available.

wall), or smoothen over question-specific biases (e.g., questions that include the word "object" make the VLM likelier to say "toy," while remaining questions are likelier to elicit "statue" or "lion.")

With robustness in mind, we include all questions and object views when aggregating PaLI VQA annotations. Ultimately the goal is to produce an intermediate representation suitable for multiple tasks. To that end, we will test these annotations on downstream inference of properties in the next section. We will show that the performance advantage of PaLI VQA annotations is not limited to our type-specific metric, but extends to the inference of physical properties.

4 Physical Properties

What an object is made of has immediate implications for how it behaves physically. Whether it will sink, bounce, stretch, or crack is largely determined by its material composition. There is limited prior work to study whether VLMs can infer such properties of an object.

We expect material to be less amenable to description in language than type; this raises the question whether we should prompt a VLM to reason deeper about material. One way to do this is to equip the VLM with previous inferences about the object. Concretely, we can ask the VLM what material something is made of while including the object's type as part of the question/prompt. Thus, the VLM can make its prediction on the basis of two factors: object type and appearance.

We ablate the influence of each factor on the VLM as follows: we pose questions including or excluding the object type (e.g. "what material is the spoon made of" vs "what material is this made of"). We also pose the former question (mentioning the object's type) without a visual input and

Table 1: **Accuracy of material inference using two different VLMs**. The models are provided either an object type annotation or image as inputs or both. We report the top-3 accuracy (whether the correct material is in the top 3 predictions, ↑) as well as the soft accuracy (probability of the correct material in the output distribution, ↑) averaged over our curated material test set. Whenever we use appearance as an input (i.e., VLM mode), we aggregate responses across object views. Thus the predicted distributions contain up to J=5 alternatives in LLM mode or up to IJ=40 in VLM mode.

		Type only (LLM mode)		Appearance only (VLM mode)	Type and Appearance (VLM mode)	
		CAP3D captions	PaLI-VQA types	No caption/type information	CAP3D captions	PaLI-VQA types
PaLI-X 55B VQA	Top-3 acc.	0.84 ± 0.36	0.66 ± 0.47	0.84 ± 0.36	$\mathbf{0.90 \pm 0.30}$	$\mathbf{0.90 \pm 0.30}$
	Soft acc.	0.42 ± 0.27	0.31 ± 0.30	0.40 ± 0.28	0.46 ± 0.27	$\mathbf{0.49 \pm 0.31}$
BLIP-2 T5 XL	Top-3 acc.	0.28 ± 0.45	0.26 ± 0.44	0.66 ± 0.47	$\mathbf{0.70 \pm 0.46}$	$\mathbf{0.70 \pm 0.46}$
	Soft acc.	0.22 ± 0.37	0.20 ± 0.36	0.51 ± 0.43	0.48 ± 0.41	$\mathbf{0.53 \pm 0.42}$

definite article (e.g. "what material is a spoon made of"). This makes the model operate as an LLM, with the same model weights, and helps measure the accuracy of language-only reasoning.

When specifying the object's type as part of a question, we have the choice of using rich detailed captions like CAP3D's, or succinct type annotations as produced by our VQA pipeline (see Sec 3.2). We study which of these performs better (but refer to them as "type annotations" collectively).

To ensure our results are not specific to a model class or size, we run these evaluations on two VLMs: PaLI-X VQA as before, and the smaller BLIP-2 T5 XL (used in CAP3D).

4.1 Results

To measure the accuracy of material prediction, we curate a test set of objects spanning 13 material classes (see Appendix B.2 for details). We then compare different ways of probing the VLM on the test set. Unlike in Sec 3.2, we cannot rely on similarity in text embedding space because materials can be close even if they are not exactly the same (e.g. "wood" and "metal" have a cosine similarity score of 0.408). So we look for an exact string match in the VLM responses.

Table 1 reveals that using type annotations (CAP3D or PaLI-VQA) and object appearance simultaneously consistently outperforms using one or the other. This reasoning advantage is reminiscent of zero-shot chain-of-thought prompting [15] or iterative inference [16, 17]. Having access to previous computations can help the VLM avoid redundant processing. This holds regardless of whether the previous inference came from the same VLM, and even for a smaller VLM like BLIP-2 T5 XL.

Although CAP3D captions contain more material information than PaLI-VQA type annotations (see "Type only" sub-columns), they are less useful as an auxiliary input when also using the object's appearance (see soft accuracies under "Type and Appearance"). This could be explained by possible hallucinations or specious details in the captions which hinder VLM reasoning. It goes to suggest that property-specific annotations serve as more robust intermediate representations for downstream tasks.

5 Conclusions

We generated property-specific annotations for 3D objects using VLMs which take in a single image and text-based prompt. We attempted to probe for properties which are increasingly inaccessible to language-based reasoning, from semantic type to material composition.

Along the way, we evaluated what VLMs are sensitive to, including changes in object view, question wording, prior inferences specified in the prompt, and access to the object's appearance. We highlighted the value of marginalizing over some of these factors to produce an aggregate response, akin to how humans might arrive at an inference by examining an object from multiple angles.

We hope our outputs serve a variety of downstream 3D applications (from generation to retrieval, from physical simulation to neuro-symbolic processing); and that our evaluations and insights may help shape VLM annotation pipelines in other contexts.

References

[1] Zifan Shi, Sida Peng, Yinghao Xu, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663*, 2022.

[2] Qun-Ce Xu, Tai-Jiang Mu, and Yong-Liang Yang. A survey of deep learning-based 3d shape generation. *Computational Visual Media*, 9(3):407–442, 2023.

[3] 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] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 35:23716–23736, 2022.

[5] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *CVPR*, pages 19175–19186, 2023.

[6] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[7] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, pages 13142–13153, 2023.

[9] Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *arXiv preprint arXiv:2306.07279*, 2023.

[10] OpenAI. Gpt-4 technical report, 2023.

[11] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[12] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*, 2022.

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

[14] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *NeurIPS*, 35:22199–22213, 2022.

[16] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[17] Joe Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412. PMLR, 2018.

[18] Zhizhong Han, Chao Chen, Yu-Shen Liu, and Matthias Zwicker. Shapecaptioner: Generative caption network for 3d shapes by learning a mapping from parts detected in multiple views to sentences. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1018–1027, 2020.

[19] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J Guibas, and Minhyuk Sung. Partglot: Learning shape part segmentation from language reference games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16505–16514, 2022.

[20] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021.

[21] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, June 2023.

[22] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.

[23] Deyao Zhu, Jun Chen, Kilichbek Haydarov, Xiaoqian Shen, Wenxuan Zhang, and Mohamed Elhoseiny. Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. *arXiv preprint arXiv:2303.06594*, 2023.

[24] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *arXiv preprint arXiv:2309.02561*, 2023.

[25] Huy Ha and Shuran Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *6th Annual Conference on Robot Learning*, 2022.

[26] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *arXiv preprint arXiv:2307.12981*, 2023.

[27] Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning, 2023.

A Supplementary Material

A.1 The Case for Property-Specific Annotations

As a human would make inferences about an object—by examining it, making higher-order inferences before more specific ones–we propose an inference scheme for VLMs to annotate an arbitrary 3D object (Fig 5). Higher-level properties (such as type, material, shape, count, and composition) are ones that require access to the object's appearance, but might subsequently facilitate lower-level inferences using associative or symbolic reasoning. If the higher-level results are categorical, we could implement the lower-level inferences as lookups from a precomputed table.

Since language is our primary mode of probing for these properties, our inference scheme does not reflect a *causal* view of how the properties arose. Rather, it presents a suite of object-centric inference tasks to evaluate VLMs on.

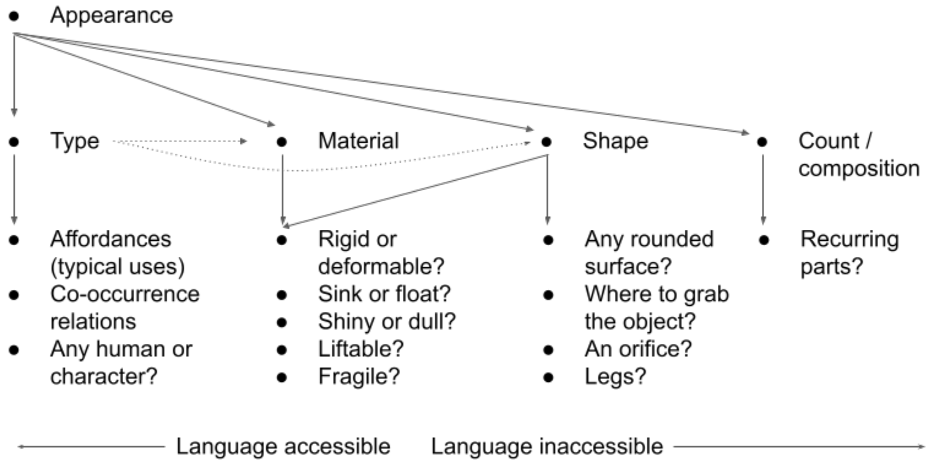


Figure 5: **A potential inference scheme to sequence property-specific annotations for 3D objects**. The arrows suggest where a VLM might benefit from having access to previous inferences. Properties toward the right require reasoning skills beyond language (e.g. spatial or counting). Lower-level inferences may not require access to the object's appearance.

A.2 Prior Work

Before VLMs, one approach [18] to caption 3D shapes focused on detecting parts of an object across multiple views, then translating a sequence of view-aggregated part features into a caption. Another work [19] showed that part segmentation was possible using human text-based annotations to discriminate between related shapes.

Foreshadowing the possibilities for semantic annotation of 2D images, [20] explored novel object detection using sparse bounding box annotations but extensive image-caption data. With the advent of VLMs, more image processing and reasoning tasks came within reach: VISPROG [21] used in-context VLM learning to produce Python code to invoke off-the-shelf computer vision models and image processing APIs. ViperGPT [22] also showed gains in reasoning spatially or at the level of object attributes by decomposing such queries into executable subroutines. Even closer to our work, [23] explored an interactive VQA approach using an LLM (ChatGPT) to ask questions about image contents, a VLM (BLIP-2) to answer them, and finally an LLM to produce a summary caption. [24] recently explored the inference of physical properties such as object material in images and collected a custom dataset to fine-tune VLMs.

Applying VLMs to 3D domains remains under-explored. [25] propose using object category labels to extract relevancy maps from 2D VLMs. These can be turned into 3D occupancies, and then utilized for scene completion or object localization tasks. [26] propose training 3D VLMs by projecting 3D feature maps into 2D and bootstrapping from a pretrained 2D VLM. The only method that contends with aggregating outputs from multiple VLM probes is ConceptGraphs [27], released when

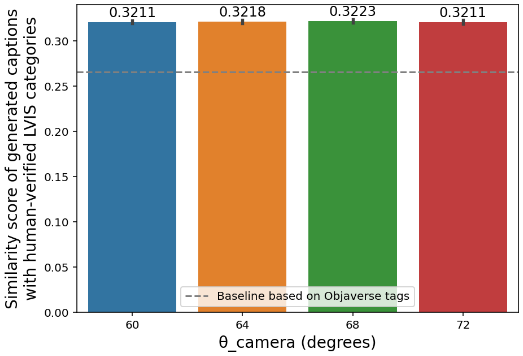


Figure 6: **Sweep over the rendering parameter θ_{camera} which determines the camera's height.** We render and caption all objects in Objaverse-LVIS for different values of θ_{camera}. All other camera parameters are held constant (we use a single viewpoint: view 7). Finally we score the captions using cosine similarity as before.

this work was submitted. Their focus is on building open-vocabulary scene graphs to help with navigation-related tasks in larger environments, which is different from our objective of generating object-centric, property-specific annotations.

B Dataset Details

B.1 Objaverse Rendering

We placed each object at the origin and scaled its maximum dimension to 1. We then rotated the camera at a fixed height and distance to the origin, rendering images at azimuthal intervals of 45 degrees. To determine the camera height, we swept over a few values of the angle θ w.r.t. the z-axis (Fig 6). We captioned each set of rendered images and used the cosine similarity metric on Objaverse-LVIS to choose the final camera height ($\theta = 68$).

B.2 Material Test Set

We search through tags in the original Objaverse dataset for 12 material keywords: 'glass', 'porcelain', 'leather', 'oak', 'metal', 'marble', 'wood', 'ceramic', 'gold', 'rubber', 'cardboard', and 'plastic'. The matches are noisy because the tags often contain spurious materials, likely to optimize search engine visibility. So we handpick a set of diverse objects for each material from the initial matches. The final test set contains 152 objects. See Appendix C.2 for examples along with our prediction results.

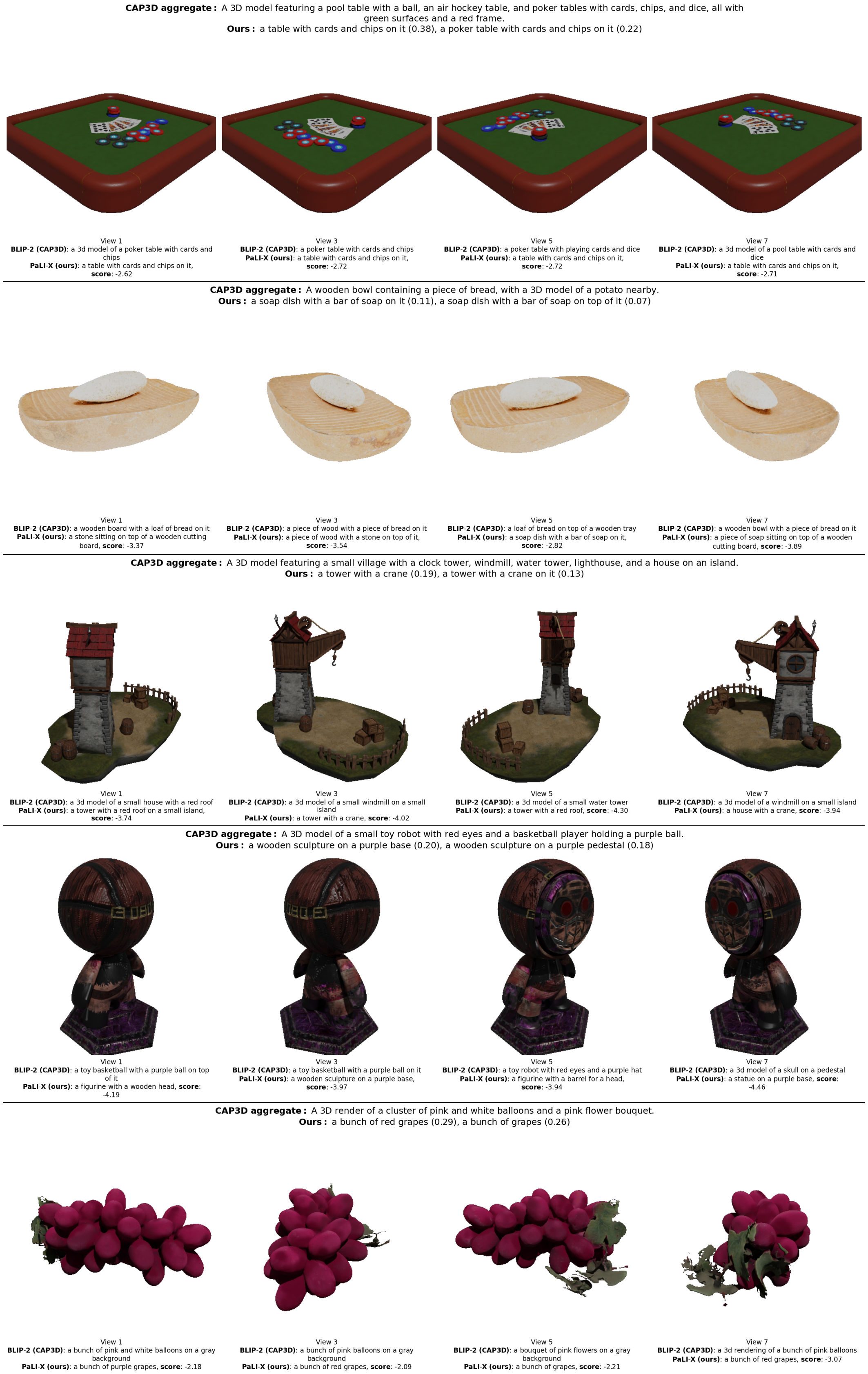
C Extended Results

C.1 Detailed Comparison with CAP3D

CAP3D uses a distinct image rendering pipeline, so this becomes an immediate point of difference with our work. While we render images at a fixed camera height, CAP3D images include top-down and bottom views of the object. Although we did not have access to their original images, we were able to compare rendered images from Fig 23 in the CAP3D paper. Only views 1, 3, 5, and 7 (zero-indexed) from our pipeline are comparable to 4, 3, 5, and 2 from the CAP3D pipeline.

We use 10 objects to illustrate the difference between our captioning results and those from CAP3D (Fig 7). We present images we rendered from the four views which are comparable between the two pipelines.

C.2 Detailed Comparison of Material Inference Results



CAP3D aggregate : A 3D model featuring a pool table with a ball, an air hockey table, and poker tables with cards, chips, and dice, all with green surfaces and a red frame.
Ours : a table with cards and chips on it (0.38), a poker table with cards and chips on it (0.22)

View 1
BLIP-2 (CAP3D): a 3d model of a poker table with cards and chips
PaLI-X (ours): a table with cards and chips on it, **score**: -2.62

View 3
BLIP-2 (CAP3D): a poker table with cards and chips
PaLI-X (ours): a table with cards and chips on it, **score**: -2.72

View 5
BLIP-2 (CAP3D): a poker table with playing cards and dice
PaLI-X (ours): a table with cards and chips on it, **score**: -2.72

View 7
BLIP-2 (CAP3D): a 3d model of a pool table with cards and dice
PaLI-X (ours): a table with cards and chips on it, **score**: -2.71

CAP3D aggregate : A wooden bowl containing a piece of bread, with a 3D model of a potato nearby.
Ours : a soap dish with a bar of soap on it (0.11), a soap dish with a bar of soap on top of it (0.07)

View 1
BLIP-2 (CAP3D): a wooden board with a loaf of bread on it
PaLI-X (ours): a stone sitting on top of a wooden cutting board, **score**: -3.37

View 3
BLIP-2 (CAP3D): a piece of wood with a piece of bread on it
PaLI-X (ours): a piece of wood with a stone on top of it, **score**: -3.54

View 5
BLIP-2 (CAP3D): a loaf of bread on top of a wooden tray
PaLI-X (ours): a soap dish with a bar of soap on it, **score**: -2.82

View 7
BLIP-2 (CAP3D): a wooden bowl with a piece of bread on it
PaLI-X (ours): a piece of soap sitting on top of a wooden cutting board, **score**: -3.89

CAP3D aggregate : A 3D model featuring a small village with a clock tower, windmill, water tower, lighthouse, and a house on an island.
Ours : a tower with a crane (0.19), a tower with a crane on it (0.13)

View 1
BLIP-2 (CAP3D): a 3d model of a small house with a red roof
PaLI-X (ours): a tower with a red roof on a small island, **score**: -3.74

View 3
BLIP-2 (CAP3D): a 3d model of a small windmill on a small island
PaLI-X (ours): a tower with a crane, **score**: -4.02

View 5
BLIP-2 (CAP3D): a 3d model of a small water tower
PaLI-X (ours): a tower with a red roof, **score**: -4.30

View 7
BLIP-2 (CAP3D): a 3d model of a windmill on a small island
PaLI-X (ours): a house with a crane, **score**: -3.94

CAP3D aggregate : A 3D model of a small toy robot with red eyes and a basketball player holding a purple ball.
Ours : a wooden sculpture on a purple base (0.20), a wooden sculpture on a purple pedestal (0.18)

View 1
BLIP-2 (CAP3D): a toy basketball with a purple ball on top of it
PaLI-X (ours): a figurine with a wooden head, **score**: -4.19

View 3
BLIP-2 (CAP3D): a toy basketball with a purple ball on it
PaLI-X (ours): a wooden sculpture on a purple base, **score**: -3.97

View 5
BLIP-2 (CAP3D): a toy robot with red eyes and a purple hat
PaLI-X (ours): a figurine with a barrel for a head, **score**: -3.94

View 7
BLIP-2 (CAP3D): a 3d model of a skull on a pedestal
PaLI-X (ours): a statue on a purple base, **score**: -4.46

CAP3D aggregate : A 3D render of a cluster of pink and white balloons and a pink flower bouquet.
Ours : a bunch of red grapes (0.29), a bunch of grapes (0.26)

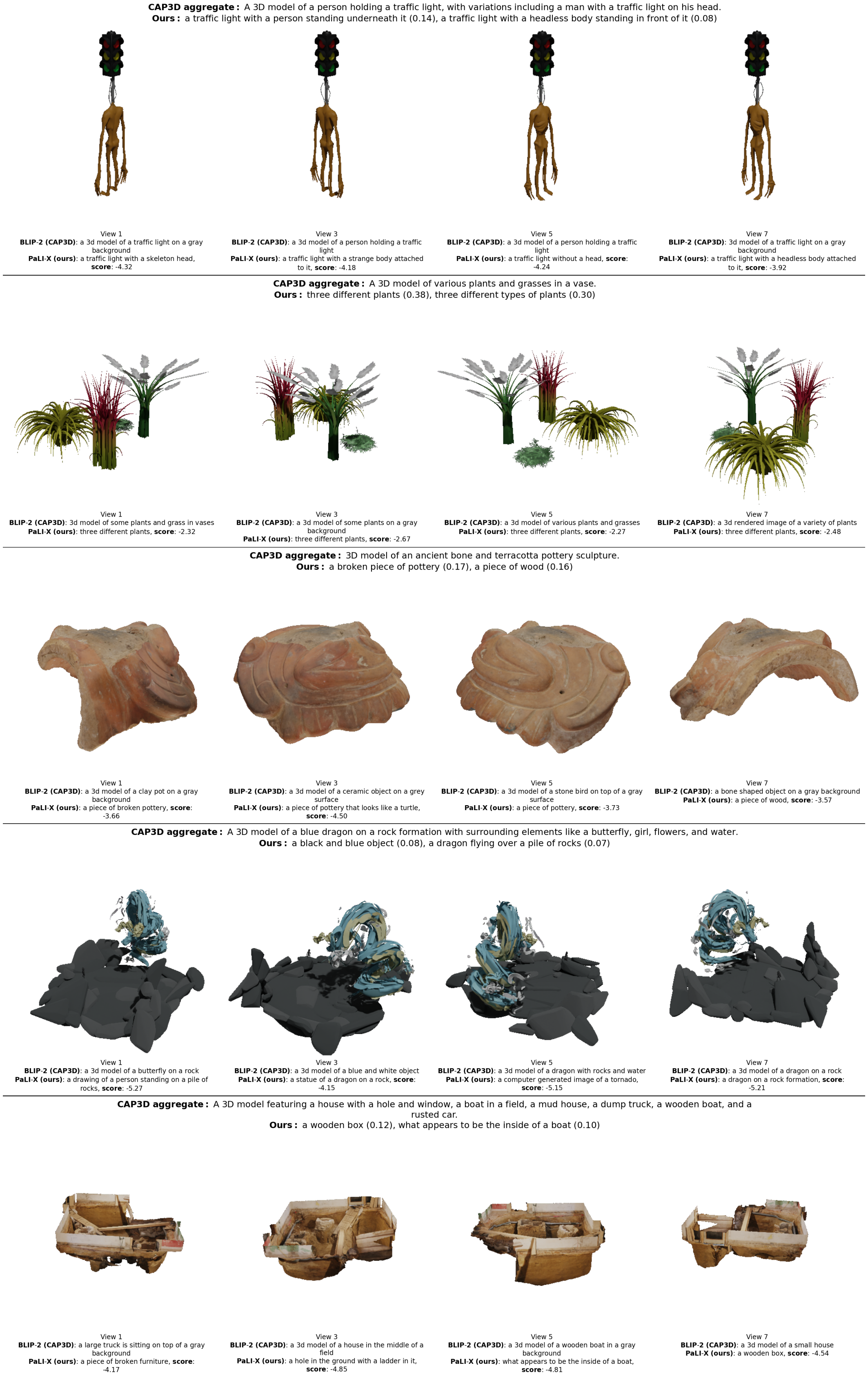
View 1
BLIP-2 (CAP3D): a bunch of pink and white balloons on a gray background
PaLI-X (ours): a bunch of purple grapes, **score**: -2.18

View 3
BLIP-2 (CAP3D): a bunch of pink balloons on a gray background
PaLI-X (ours): a bunch of red grapes, **score**: -2.09

View 5
BLIP-2 (CAP3D): a bouquet of pink flowers on a gray background
PaLI-X (ours): a bunch of grapes, **score**: -2.21

View 7
BLIP-2 (CAP3D): a 3d rendering of a bunch of pink balloons
PaLI-X (ours): a bunch of red grapes, **score**: -3.07

Figure 7: **Comparison of captions from our pipeline versus the baseline CAP3D.** We also show view-specific captions from the underlying VLMs (PaLI-X and BLIP-2).



CAP3D aggregate : A 3D model of a person holding a traffic light, with variations including a man with a traffic light on his head.
Ours : a traffic light with a person standing underneath it (0.14), a traffic light with a headless body standing in front of it (0.08)

View 1
BLIP-2 (CAP3D): a 3d model of a traffic light on a gray background
PaLI-X (ours): a traffic light with a skeleton head, **score**: -4.32

View 3
BLIP-2 (CAP3D): a 3d model of a person holding a traffic light
PaLI-X (ours): a traffic light with a strange body attached to it, **score**: -4.18

View 5
BLIP-2 (CAP3D): a 3d model of a person holding a traffic light
PaLI-X (ours): a traffic light without a head, **score**: -4.24

View 7
BLIP-2 (CAP3D): 3d model of a traffic light on a gray background
PaLI-X (ours): a traffic light with a headless body attached to it, **score**: -3.92

CAP3D aggregate : A 3D model of various plants and grasses in a vase.
Ours : three different plants (0.38), three different types of plants (0.30)

View 1
BLIP-2 (CAP3D): 3d model of some plants and grass in vases
PaLI-X (ours): three different plants, **score**: -2.32

View 3
BLIP-2 (CAP3D): a 3d model of some plants on a gray background
PaLI-X (ours): three different plants, **score**: -2.67

View 5
BLIP-2 (CAP3D): a 3d model of various plants and grasses
PaLI-X (ours): three different plants, **score**: -2.27

View 7
BLIP-2 (CAP3D): a 3d rendered image of a variety of plants
PaLI-X (ours): three different plants, **score**: -2.48

CAP3D aggregate : 3D model of an ancient bone and terracotta pottery sculpture.
Ours : a broken piece of pottery (0.17), a piece of wood (0.16)

View 1
BLIP-2 (CAP3D): a 3d model of a clay pot on a gray background
PaLI-X (ours): a piece of broken pottery, **score**: -3.66

View 3
BLIP-2 (CAP3D): a 3d model of a ceramic object on a grey surface
PaLI-X (ours): a piece of pottery that looks like a turtle, **score**: -4.50

View 5
BLIP-2 (CAP3D): a 3d model of a stone bird on top of a gray surface
PaLI-X (ours): a piece of pottery, **score**: -3.73

View 7
BLIP-2 (CAP3D): a bone shaped object on a gray background
PaLI-X (ours): a piece of wood, **score**: -3.57

CAP3D aggregate : A 3D model of a blue dragon on a rock formation with surrounding elements like a butterfly, girl, flowers, and water.
Ours : a black and blue object (0.08), a dragon flying over a pile of rocks (0.07)

View 1
BLIP-2 (CAP3D): a 3d model of a butterfly on a rock
PaLI-X (ours): a drawing of a person standing on a pile of rocks, **score**: -5.27

View 3
BLIP-2 (CAP3D): a 3d model of a blue and white object
PaLI-X (ours): a statue of a dragon on a rock, **score**: -4.15

View 5
BLIP-2 (CAP3D): a 3d model of a dragon with rocks and water
PaLI-X (ours): a computer generated image of a tornado, **score**: -5.15

View 7
BLIP-2 (CAP3D): a 3d model of a dragon on a rock
PaLI-X (ours): a dragon on a rock formation, **score**: -5.21

CAP3D aggregate : A 3D model featuring a house with a hole and window, a boat in a field, a mud house, a dump truck, a wooden boat, and a rusted car.
Ours : a wooden box (0.12), what appears to be the inside of a boat (0.10)

View 1
BLIP-2 (CAP3D): a large truck is sitting on top of a gray background
PaLI-X (ours): a piece of broken furniture, **score**: -4.17

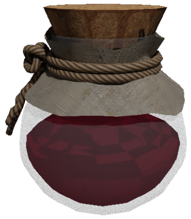
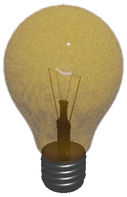
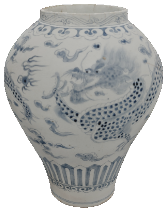
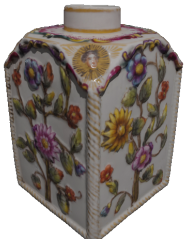
View 3
BLIP-2 (CAP3D): a 3d model of a house in the middle of a field
PaLI-X (ours): a hole in the ground with a ladder in it, **score**: -4.85

View 5
BLIP-2 (CAP3D): a 3d model of a wooden boat in a gray background
PaLI-X (ours): what appears to be the inside of a boat, **score**: -4.81

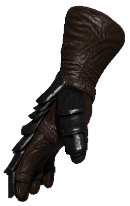
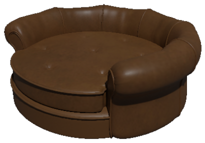
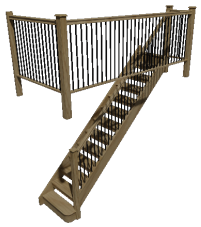
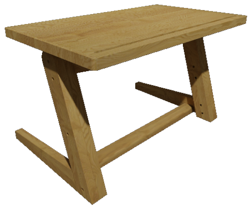
View 7
BLIP-2 (CAP3D): a 3d model of a small house
PaLI-X (ours): a wooden box, **score**: -4.54

Figure 7: **Comparison of captions from our pipeline versus the baseline CAP3D (contd).** The last two rows were described as failure cases for CAP3D in that paper.

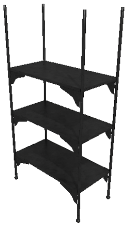
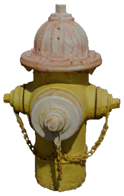
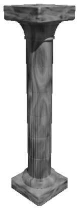
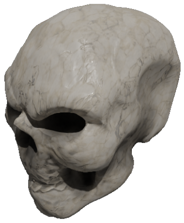
Table 2: **Material prediction examples on each category from our curated test set.** We show predicted distributions from both VLMs (PaLI-X and BLIP-2) and all five sets of inputs described in Sec 4. For brevity, each distribution is represented by the top two outputs along with their probabilities. We use t_{cap3d} or t_{pali} to denote the type annotations, A to denote all object views, $p_{vlm}(\hat{m}|.)$ to denote a predicted distribution, and m to denote the true material.

	m	"glass"
	t_{cap3d}	"hat and a jar, both with ropes tied around them"
	t_{pali}	"potion"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"cotton" (0.64), "can't tell" (0.36)
	$p_{pali}(\hat{m}\|t_{pali})$	"potion" (0.35), "glass" (0.27)
	$p_{pali}(\hat{m}\|A)$	"cork" (0.45), "glass" (0.19)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"burlap" (0.44), "canvas" (0.30)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"glass" (0.67), "cork" (0.17)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"straw" (0.49), "plastic" (0.33)
	$p_{blip}(\hat{m}\|t_{pali})$	"a tainted potion made of a tainted potion and a tainted potion" (0.77), "a tainted potion made of a tainted potion, and a tainted poti" (0.14)
	$p_{blip}(\hat{m}\|A)$	"wood" (0.83), "rope" (0.10)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"wood" (0.68), "leather" (0.13)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"wood" (0.95), "stone" (0.04)
	m	"glass"
	t_{cap3d}	"light bulb"
	t_{pali}	"light"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"glass" (0.77), "filament" (0.11)
	$p_{pali}(\hat{m}\|t_{pali})$	"glass" (0.58), "light-emitting diode,LED" (0.13)
	$p_{pali}(\hat{m}\|A)$	"glass" (0.41), "brass" (0.19)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"glass" (0.60), "porcelain" (0.13)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"glass" (0.51), "filament" (0.14)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"glass" (0.52), "filament" (0.29)
	$p_{blip}(\hat{m}\|t_{pali})$	"light-emitting diodes" (0.73), "light-emitting diodes (LEDs)" (0.20)
	$p_{blip}(\hat{m}\|A)$	"metal" (0.30), "3ds max" (0.22)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"metal" (0.84), "gold" (0.13)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"metal" (0.78), "gold" (0.14)
	m	"porcelain"
	t_{cap3d}	"blue and white vase featuring a dragon design"
	t_{pali}	"vase"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"ceramic" (0.38), "porcelain" (0.34)
	$p_{pali}(\hat{m}\|t_{pali})$	"ceramic" (0.35), "glass" (0.31)
	$p_{pali}(\hat{m}\|A)$	"faience" (0.62), "porcelain" (0.14)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"ceramic" (0.38), "porcelain" (0.32)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"faience" (0.44), "ceramic" (0.24)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"porcelain" (0.65), "Chinese celadon" (0.32)
	$p_{blip}(\hat{m}\|t_{pali})$	"Porcelain" (0.86), "terracotta" (0.09)
	$p_{blip}(\hat{m}\|A)$	"porcelain" (0.83), "ceramic" (0.08)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"porcelain" (0.88), "china" (0.12)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"porcelain" (0.80), "china" (0.07)
	m	"porcelain"
	t_{cap3d}	"small white porcelain vase with colorful floral designs on it"
	t_{pali}	"inkwell"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"porcelain" (0.29), "faience" (0.28)
	$p_{pali}(\hat{m}\|t_{pali})$	"glass" (0.28), "porcelain" (0.24)
	$p_{pali}(\hat{m}\|A)$	"faience" (0.88), "porcelain" (0.06)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"faience" (0.68), "porcelain" (0.15)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"faience" (0.71), "porcelain" (0.16)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"China" (0.58), "ceramic" (0.24)
	$p_{blip}(\hat{m}\|t_{pali})$	"metal" (0.93), "metal or plastic" (0.06)
	$p_{blip}(\hat{m}\|A)$	"porcelain" (0.99), "white porcelain" (0.01)

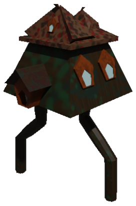
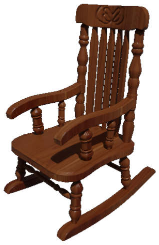
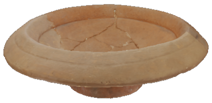
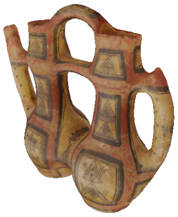
Material prediction examples on each category from our curated test set (contd).

	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"porcelain" (0.80), "china" (0.12)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"porcelain" (0.94), "china" (0.04)
	m	"leather"
	t_{cap3d}	"armored leather gloves and a brown leather boot"
	t_{pali}	"glove"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"leather" (0.83), "cowhide" (0.08)
	$p_{pali}(\hat{m}\|t_{pali})$	"leather" (0.34), "cotton" (0.21)
	$p_{pali}(\hat{m}\|A)$	"leather" (0.69), "armor plate,armour plate,armor plating,plate armor,plate armour" (0.08)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"leather" (0.80), "cowhide" (0.07)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"leather" (0.84), "nylon" (0.04)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"leather" (1.00)
	$p_{blip}(\hat{m}\|t_{pali})$	"leather" (0.98), "neoprene" (0.02)
	$p_{blip}(\hat{m}\|A)$	"leather" (1.00)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"leather" (1.00), "neoprene" (0.00)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"leather" (1.00), "neoprene" (0.00)
	m	"leather"
	t_{cap3d}	"round tan leather sofa-style dog bed with buttons"
	t_{pali}	"dog bed"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"leather" (0.70), "suede" (0.16)
	$p_{pali}(\hat{m}\|t_{pali})$	"foam" (0.39), "cotton" (0.37)
	$p_{pali}(\hat{m}\|A)$	"leather" (0.81), "upholstery" (0.08)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"leather" (0.87), "faux leather" (0.04)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"leather" (0.89), "faux leather" (0.04)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"faux leather" (0.87), "faux-leather" (0.13)
	$p_{blip}(\hat{m}\|t_{pali})$	"a soft fabric, such as cotton, wool, linen, or a combination of the two" (1.00), "a soft fabric, such as cotton, wool, linen, or a synthetic material, such as acetate or polypropylene" (0.00)
	$p_{blip}(\hat{m}\|A)$	"leather" (1.00)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"leather" (0.79), "3d model" (0.12)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"leather" (1.00)
	m	"oak"
	t_{cap3d}	"wooden staircase with metal railings"
	t_{pali}	"bannister"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"wood" (0.46), "steel" (0.19)
	$p_{pali}(\hat{m}\|t_{pali})$	"wood" (0.80), "marble" (0.07)
	$p_{pali}(\hat{m}\|A)$	"wood" (0.78), "timber" (0.06)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"wood" (0.46), "oak" (0.31)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"wood" (0.50), "metal" (0.26)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"a wooden staircase with metal railings" (1.00), "a wooden staircase with metal railings is called a balustrade" (0.00)
	$p_{blip}(\hat{m}\|t_{pali})$	"wood" (0.73), "wooden" (0.27)
	$p_{blip}(\hat{m}\|A)$	"wood" (0.99), "wooden railings" (0.00)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"wood" (0.98), "wooden staircase with metal railings" (0.02)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"wood" (0.97), "wooden" (0.03)
	m	"oak"
	t_{cap3d}	"small wooden table with two legs and a slanted top"
	t_{pali}	"trestle table"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"wood" (0.65), "oak" (0.24)
	$p_{pali}(\hat{m}\|t_{pali})$	"wood" (0.88), "timber" (0.05)
	$p_{pali}(\hat{m}\|A)$	"wood" (0.63), "oak" (0.15)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"wood" (0.43), "oak" (0.42)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"wood" (0.70), "oak" (0.20)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"trestle table" (0.80), "a trestle table" (0.20)
	$p_{blip}(\hat{m}\|t_{pali})$	"wood" (0.98), "wooden trestle" (0.02)

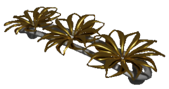
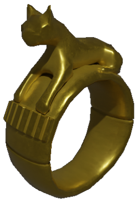
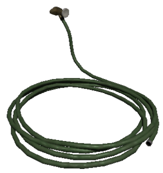
Material prediction examples on each category from our curated test set (contd).

	$p_{blip}(\hat{m}\|A)$	"wood" (0.97), "wooden" (0.03)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"wood" (0.90), "solid wood" (0.09)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"wood" (0.99), "solid wood" (0.01)
	m	"metal"
	t_{cap3d}	"three-tier metal shelving unit"
	t_{pali}	"bookshelf"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"steel" (0.41), "metal" (0.29)
	$p_{pali}(\hat{m}\|t_{pali})$	"wood" (0.91), "metal" (0.03)
	$p_{pali}(\hat{m}\|A)$	"metal" (0.42), "steel" (0.36)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"steel" (0.49), "metal" (0.29)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"metal" (0.59), "steel" (0.20)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"steel" (0.99), "steel or stainless steel" (0.01)
	$p_{blip}(\hat{m}\|t_{pali})$	"wood" (0.98), "reclaimed wood" (0.02)
	$p_{blip}(\hat{m}\|A)$	"metal" (0.72), "steel" (0.21)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"black metal" (0.43), "steel" (0.32)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"metal" (0.68), "steel" (0.20)
	m	"metal"
	t_{cap3d}	"yellow fire hydrant"
	t_{pali}	"fire hydrant"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"metal" (0.37), "steel" (0.24)
	$p_{pali}(\hat{m}\|t_{pali})$	"metal" (0.32), "steel" (0.25)
	$p_{pali}(\hat{m}\|A)$	"iron" (0.31), "metal" (0.17)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"metal" (0.37), "steel" (0.19)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"metal" (0.32), "iron" (0.21)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"cast iron" (0.91), "cast-aluminum" (0.09)
	$p_{blip}(\hat{m}\|t_{pali})$	"a fire hydrant is a device used to extinguish a fire." (0.98), "a fire hydrant is a device used to extinguish a fire by means of a pressurized stream of water" (0.01)
	$p_{blip}(\hat{m}\|A)$	"plastic" (0.35), "3ds max" (0.24)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"metal" (0.79), "plastic" (0.20)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"metal" (0.70), "plastic" (0.17)
	m	"marble"
	t_{cap3d}	"white marble column"
	t_{pali}	"pedestal"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"marble" (0.75), "limestone" (0.09)
	$p_{pali}(\hat{m}\|t_{pali})$	"marble" (0.44), "stone" (0.31)
	$p_{pali}(\hat{m}\|A)$	"marble" (0.69), "stone" (0.17)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"marble" (0.73), "carrara" (0.10)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"marble" (0.67), "stone" (0.21)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"marble" (1.00)
	$p_{blip}(\hat{m}\|t_{pali})$	"marble" (1.00)
	$p_{blip}(\hat{m}\|A)$	"marble" (0.96), "wood" (0.04)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"marble" (0.73), "white marble" (0.27)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"marble" (0.95), "wood" (0.04)
	m	"marble"
	t_{cap3d}	"white marble skull"
	t_{pali}	"skull"
	$p_{pali}(\hat{m}\|t_{cap3d})$	"marble" (0.79), "porcelain" (0.09)
	$p_{pali}(\hat{m}\|t_{pali})$	"bone" (0.75), "bones" (0.09)
	$p_{pali}(\hat{m}\|A)$	"clay" (0.35), "marble" (0.22)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	"marble" (0.55), "clay" (0.20)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	"clay" (0.33), "marble" (0.27)
	$p_{blip}(\hat{m}\|t_{cap3d})$	"limestone" (0.68), "marble" (0.32)
	$p_{blip}(\hat{m}\|t_{pali})$	"calcium phosphate" (0.83), "calcareous limestone" (0.08)
	$p_{blip}(\hat{m}\|A)$	"marble" (0.81), "white marble" (0.10)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	"white marble" (0.85), "marble" (0.07)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	"marble" (0.43), "limestone" (0.36)

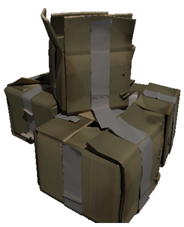
Material prediction examples on each category from our curated test set (contd).

	m	“wood”
	t_{cap3d}	“small metal house with a roof and legs”
	t_{pali}	“birdhouse”
	$p_{pali}(\hat{m} \mid t_{cap3d})$	“aluminum” (0.48), “steel” (0.34)
	$p_{pali}(\hat{m} \mid t_{pali})$	“wood” (0.75), “clay” (0.10)
	$p_{pali}(\hat{m} \mid A)$	“wood” (0.42), “copper” (0.23)
	$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“steel” (0.21), “iron” (0.19)
	$p_{pali}(\hat{m} \mid t_{pali}, A)$	“wood” (0.61), “metal” (0.17)
	$p_{blip}(\hat{m} \mid t_{cap3d})$	“a styrofoam styrofoam styrofoam sty” (0.36), “a styrofoam styrofoam styrofoam sandwich” (0.33)
	$p_{blip}(\hat{m} \mid t_{pali})$	“wood” (0.68), “Cedar” (0.31)
	$p_{blip}(\hat{m} \mid A)$	“metal” (0.86), “wood” (0.07)
	$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“3d model” (0.59), “rusty metal” (0.21)
	$p_{blip}(\hat{m} \mid t_{pali}, A)$	“metal” (0.61), “wood” (0.33)
	m	“wood”
	t_{cap3d}	“wooden rocking chair”
	t_{pali}	“rocking chair”
	$p_{pali}(\hat{m} \mid t_{cap3d})$	“wood” (0.58), “oak” (0.22)
	$p_{pali}(\hat{m} \mid t_{pali})$	“wood” (0.81), “wicker” (0.08)
	$p_{pali}(\hat{m} \mid A)$	“wood” (0.88), “rattan” (0.04)
	$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“oak” (0.40), “wood” (0.22)
	$p_{pali}(\hat{m} \mid t_{pali}, A)$	“wood” (0.93), “mahogany” (0.02)
	$p_{blip}(\hat{m} \mid t_{cap3d})$	“wood” (0.96), “rattan” (0.04)
	$p_{blip}(\hat{m} \mid t_{pali})$	“wood” (0.97), “wooden rocking chair” (0.03)
	$p_{blip}(\hat{m} \mid A)$	“wood” (0.98), “wooden” (0.01)
	$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“wood” (0.96), “wooden rocking chair” (0.04)
	$p_{blip}(\hat{m} \mid t_{pali}, A)$	“wood” (1.00), “wooden rocking chair” (0.00)
	m	“ceramic”
	t_{cap3d}	“terracotta bowl with a curved top, flat bottom”
	t_{pali}	“tray”
	$p_{pali}(\hat{m} \mid t_{cap3d})$	“ceramic” (0.40), “stoneware” (0.25)
	$p_{pali}(\hat{m} \mid t_{pali})$	“wood” (0.28), “ceramic” (0.24)
	$p_{pali}(\hat{m} \mid A)$	“clay” (0.33), “stoneware” (0.28)
	$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“clay” (0.47), “ceramic” (0.18)
	$p_{pali}(\hat{m} \mid t_{pali}, A)$	“clay” (0.41), “stoneware” (0.19)
	$p_{blip}(\hat{m} \mid t_{cap3d})$	“earthenware” (0.55), “terracotta” (0.31)
	$p_{blip}(\hat{m} \mid t_{pali})$	“stainless steel” (1.00), “stainless steel or stainless steel-alloys” (0.00)
	$p_{blip}(\hat{m} \mid A)$	“clay” (0.92), “terracotta” (0.05)
	$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“clay” (0.64), “terracotta” (0.35)
	$p_{blip}(\hat{m} \mid t_{pali}, A)$	“clay” (0.94), “terracotta” (0.05)
	m	“ceramic”
	t_{cap3d}	“vase with two handles and intricate designs”
	t_{pali}	“jug”
	$p_{pali}(\hat{m} \mid t_{cap3d})$	“ceramic” (0.38), “porcelain” (0.27)
	$p_{pali}(\hat{m} \mid t_{pali})$	“glass” (0.56), “porcelain” (0.17)
	$p_{pali}(\hat{m} \mid A)$	“stoneware” (0.29), “clay” (0.23)
	$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“clay” (0.36), “pottery” (0.27)
	$p_{pali}(\hat{m} \mid t_{pali}, A)$	“ceramic” (0.29), “clay” (0.27)
	$p_{blip}(\hat{m} \mid t_{cap3d})$	“Chinese celadon” (0.97), “Chinese lacquerware” (0.02)
	$p_{blip}(\hat{m} \mid t_{pali})$	“clay” (0.87), “tin” (0.13)
	$p_{blip}(\hat{m} \mid A)$	“clay” (0.73), “ceramic” (0.27)
	$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“clay” (0.76), “ceramic” (0.24)
	$p_{blip}(\hat{m} \mid t_{pali}, A)$	“clay” (0.87), “ceramic” (0.13)

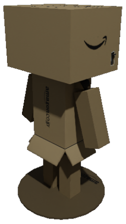
Material prediction examples on each category from our curated test set (contd).

	m	“gold”
	t_{cap3d}	“gold flower ring featuring a yellow and white flower design”
	t_{pali}	“hair slide”
	$p_{pali}(\hat{m}\|t_{cap3d})$	“gold” (0.74), “sterling silver” (0.09)
	$p_{pali}(\hat{m}\|t_{pali})$	“plastic” (0.44), “rubber” (0.24)
	$p_{pali}(\hat{m}\|A)$	“gold plate” (0.33), “brass” (0.31)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	“gold” (0.40), “brass” (0.24)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	“brass” (0.23), “metal” (0.23)
	$p_{blip}(\hat{m}\|t_{cap3d})$	“14K yellow gold” (0.35), “18k white gold” (0.34)
	$p_{blip}(\hat{m}\|t_{pali})$	“plastic” (0.88), “acetate” (0.12)
	$p_{blip}(\hat{m}\|A)$	“gold” (0.65), “metal” (0.32)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	“gold” (0.59), “3d model” (0.15)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	“gold” (0.72), “metal” (0.22)
	m	“gold”
	t_{cap3d}	“gold Egyptian cat ring”
	t_{pali}	“ring”
	$p_{pali}(\hat{m}\|t_{cap3d})$	“gold” (0.68), “gold plate” (0.11)
	$p_{pali}(\hat{m}\|t_{pali})$	“gold” (0.74), “brass” (0.10)
	$p_{pali}(\hat{m}\|A)$	“gold” (0.63), “gold plate” (0.20)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	“gold” (0.78), “brass” (0.10)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	“gold” (0.82), “brass” (0.09)
	$p_{blip}(\hat{m}\|t_{cap3d})$	“gold” (1.00), “gold-plated tibetan calfskin” (0.00)
	$p_{blip}(\hat{m}\|t_{pali})$	“precious metals, such as gold, silver, platinum, palladium, and rhodium” (0.93), “precious metals, such as gold, silver, platinum, palladium, rhodium, and tin” (0.03)
	$p_{blip}(\hat{m}\|A)$	“gold” (1.00), “gold 3d printed” (0.00)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	“gold” (0.90), “3d printed” (0.10)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	“gold” (1.00)
	m	“rubber”
	t_{cap3d}	“tire”
	t_{pali}	“tire”
	$p_{pali}(\hat{m}\|t_{cap3d})$	“rubber” (0.99), “rubber and steel” (0.00)
	$p_{pali}(\hat{m}\|t_{pali})$	“rubber” (0.99), “rubber and steel” (0.00)
	$p_{pali}(\hat{m}\|A)$	“rubber” (0.96), “blacktop,blacktopping” (0.02)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	“rubber” (0.97), “black rubber” (0.01)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	“rubber” (0.97), “black rubber” (0.01)
	$p_{blip}(\hat{m}\|t_{cap3d})$	“rubber” (0.90), “pneumatic tires” (0.09)
	$p_{blip}(\hat{m}\|t_{pali})$	“rubber” (0.73), “Rubber” (0.27)
	$p_{blip}(\hat{m}\|A)$	“rubber” (0.87), “black rubber” (0.10)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	“rubber” (0.93), “black rubber” (0.06)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	“rubber” (0.91), “black rubber” (0.09)
	m	“rubber”
	t_{cap3d}	“green coiled cable with a white plug and attached earbud”
	t_{pali}	“hose”
	$p_{pali}(\hat{m}\|t_{cap3d})$	“nylon” (0.44), “plastic” (0.36)
	$p_{pali}(\hat{m}\|t_{pali})$	“rubber” (0.86), “plastic” (0.05)
	$p_{pali}(\hat{m}\|A)$	“hose” (0.48), “rubber” (0.20)
	$p_{pali}(\hat{m}\|t_{cap3d}, A)$	“rubber” (0.38), “plastic” (0.19)
	$p_{pali}(\hat{m}\|t_{pali}, A)$	“rubber” (0.70), “plastic” (0.15)
	$p_{blip}(\hat{m}\|t_{cap3d})$	“tin-alloy” (0.79), “tin-plated copper” (0.20)
	$p_{blip}(\hat{m}\|t_{pali})$	“rubber” (0.95), “PTFE” (0.05)
	$p_{blip}(\hat{m}\|A)$	“wire” (0.33), “metal” (0.26)
	$p_{blip}(\hat{m}\|t_{cap3d}, A)$	“teflon” (0.92), “stranded copper” (0.05)
	$p_{blip}(\hat{m}\|t_{pali}, A)$	“plastic” (0.36), “pvc” (0.23)

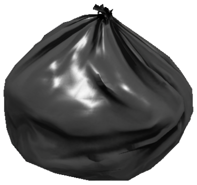
Material prediction examples on each category from our curated test set (contd).



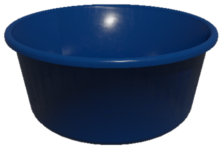
m	“cardboard”
t_{cap3d}	“stack of brown cardboard boxes with white tape on them”
t_{pali}	“packing box”
$p_{pali}(\hat{m} \mid t_{cap3d})$	“cardboard” (0.64), “paper” (0.30)
$p_{pali}(\hat{m} \mid t_{pali})$	“cardboard” (0.74), “paper” (0.13)
$p_{pali}(\hat{m} \mid A)$	“cardboard” (0.52), “cellulose tape,Scotch tape,Sellotape” (0.17)
$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“cardboard” (0.67), “paper” (0.13)
$p_{pali}(\hat{m} \mid t_{pali}, A)$	“cardboard” (0.82), “corrugated cardboard” (0.06)
$p_{blip}(\hat{m} \mid t_{cap3d})$	“shipping cartons” (1.00), “a receptacle for the shipment of goods” (0.00)
$p_{blip}(\hat{m} \mid t_{pali})$	“cardboard” (0.65), “paper” (0.27)
$p_{blip}(\hat{m} \mid A)$	“cardboard” (1.00), “styrofoam” (0.00)
$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“cardboard” (0.96), “3d model” (0.01)
$p_{blip}(\hat{m} \mid t_{pali}, A)$	“cardboard” (0.99), “paper” (0.01)



m	“cardboard”
t_{cap3d}	“cardboard Amazon robot toy with logo”
t_{pali}	“carton”
$p_{pali}(\hat{m} \mid t_{cap3d})$	“cardboard” (0.57), “paper” (0.32)
$p_{pali}(\hat{m} \mid t_{pali})$	“cardboard” (0.47), “paper” (0.45)
$p_{pali}(\hat{m} \mid A)$	“cardboard” (0.80), “carton” (0.13)
$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“cardboard” (0.73), “carton” (0.10)
$p_{pali}(\hat{m} \mid t_{pali}, A)$	“cardboard” (0.85), “corrugated cardboard” (0.06)
$p_{blip}(\hat{m} \mid t_{cap3d})$	“cardboard” (0.99), “acetate” (0.01)
$p_{blip}(\hat{m} \mid t_{pali})$	“paper” (0.75), “paperboard” (0.25)
$p_{blip}(\hat{m} \mid A)$	“cardboard” (1.00)
$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“cardboard” (1.00), “cardboard, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard boxes, cardboard” (0.00)
$p_{blip}(\hat{m} \mid t_{pali}, A)$	“cardboard” (0.99), “paper” (0.01)



m	“plastic”
t_{cap3d}	“large silver trash bag”
t_{pali}	“garbage bag”
$p_{pali}(\hat{m} \mid t_{cap3d})$	“plastic” (0.45), “aluminum” (0.35)
$p_{pali}(\hat{m} \mid t_{pali})$	“plastic” (0.80), “polythene” (0.07)
$p_{pali}(\hat{m} \mid A)$	“garbage” (0.45), “plastic” (0.42)
$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“plastic” (0.66), “cellophane” (0.13)
$p_{pali}(\hat{m} \mid t_{pali}, A)$	“plastic” (0.83), “polythene” (0.06)
$p_{blip}(\hat{m} \mid t_{cap3d})$	“plastic” (0.97), “woven polypropylene” (0.03)
$p_{blip}(\hat{m} \mid t_{pali})$	“plastic” (1.00), “a polyethylene terephthalate (PET) film” (0.00)
$p_{blip}(\hat{m} \mid A)$	“black plastic” (0.67), “3ds max” (0.15)
$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“plastic” (0.83), “black plastic” (0.11)
$p_{blip}(\hat{m} \mid t_{pali}, A)$	“plastic” (0.60), “black plastic” (0.38)



m	“plastic”
t_{cap3d}	“blue plastic bowl with a lid”
t_{pali}	“washtub”
$p_{pali}(\hat{m} \mid t_{cap3d})$	“polypropylene” (0.53), “plastic” (0.47)
$p_{pali}(\hat{m} \mid t_{pali})$	“porcelain” (0.56), “ceramic” (0.35)
$p_{pali}(\hat{m} \mid A)$	“plastic” (0.65), “polypropylene” (0.18)
$p_{pali}(\hat{m} \mid t_{cap3d}, A)$	“polypropylene” (0.58), “plastic” (0.24)
$p_{pali}(\hat{m} \mid t_{pali}, A)$	“plastic” (0.78), “polypropylene” (0.10)
$p_{blip}(\hat{m} \mid t_{cap3d})$	“borosilicate glass” (0.97), “PP (Polypropylene)” (0.02)
$p_{blip}(\hat{m} \mid t_{pali})$	“plastic” (0.90), “tin” (0.10)
$p_{blip}(\hat{m} \mid A)$	“plastic” (1.00), “polygons” (0.00)
$p_{blip}(\hat{m} \mid t_{cap3d}, A)$	“plastic” (0.99), “polypropylene” (0.01)

Material prediction examples on each category from our curated test set (contd).

$p_{blip}(\hat{m}\|t_{pali}, A)$	“plastic” (1.00)