Efficient Structure-Guided 3D Physical Property Reasoning

Anonymous OpenSUN3D submission

Paper ID

Abstract

Inferring an object's physical properties such as material type and surface hardness from visual observations is essential for augmented reality, robotic perception, and embodied intelligence. However, existing solutions to physical property reasoning like NeRF2Physics are computationally expensive and error-prone because they interpolate sparse, noisy CLIP features across dense 3D scenes. This creates a fundamental conflict between the pursuit of high semantic resolution and high reasoning efficiency while making the system sensitive to oblique or low-quality viewpoints. We introduce a lightweight, structure-guided framework that achieves fine-grained semantic consistency for physical property reasoning with orders-of-magnitude lower computational cost. Our key insight is that the 3D structural priors offer a stronger cue for the object's semantic organization, which allows us to avoid the dense interpolation for physical property reasoning. We project 2D DINO embeddings into 3D for coarse component segmentation, perform adaptive sparse sampling of representative CLIP source points, and apply a view-quality-aware patch selection with probability-weighted aggregation. These designs successfully eliminate dense interpolation, suppress noisy viewpoints, and drastically cut the number of CLIP queries. Extensive experiments on ABO dataset demonstrate our method reduces end-to-end ***runtime from hundreds of seconds to mere seconds*** *per scene while improving semantic accuracy, spatial coherence, and downstream physical-property inference.*

1. Introduction

Reasoning an object's physical properties from its visual observations, such as material type, weights, surface roughness, and structural attributes, is a fundamental capability for visual intelligence and embodied perception [6, 24]. Recent advances that ground 2D Vision-Language Models (VLMs) in 3D representations have begun to unlock open-world physical reasoning by connecting an object's appearance to its underlying material semantics [13, 19, 23, 27, 30]. This progress opens new avenues for augmented reality interaction [1], more accurate physics-based simulation [10] and improved robotic manipulation and planning [5, 21].

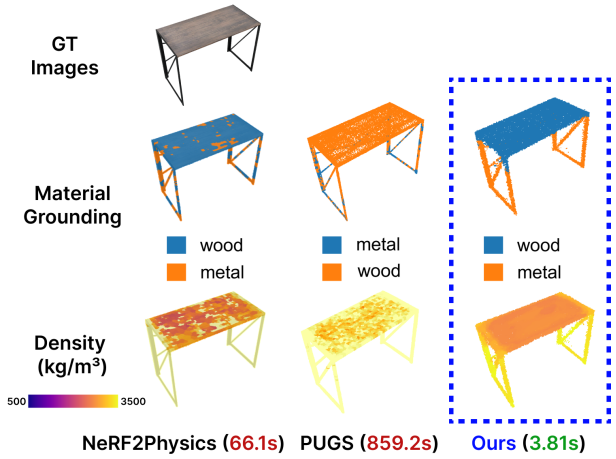

Figure 1. **Material grounding and physical property estimation comparison.** Compared to NeRF2Physics [29] (66.1s) and PUGS [23] (859.2s), our method produces more accurate material grounding and physical property estimation in only 3.81s, achieving over 17× speedup over NeRF2Physics and 225× over PUGS. Our method yields coherent, component-level material assignments (e.g., correctly distinguishing the wooden tabletop from the metal frame) and more uniform density distributions within homogeneous regions. In contrast, prior methods exhibit fragmented material boundaries and inconsistent physical property estimation.

While the detailed techniques may differ, existing vision-based physical property reasoning solutions [19, 23, 30] all follow a common pipeline: begin with a reconstructed 3D point cloud and, for each point, retrieve the corresponding multi-view image patches. Each patch is fed into the vision encoder [8, 20] to obtain sparse semantic features, which are then interpolated across the full 3D point cloud to predict a material label for every point. Finally, these per-point material estimates are integrated to infer physical properties such as hardness or density.

However, such approaches suffer from two limitations.

- *Excessively long processing time from exploding CLIP-*

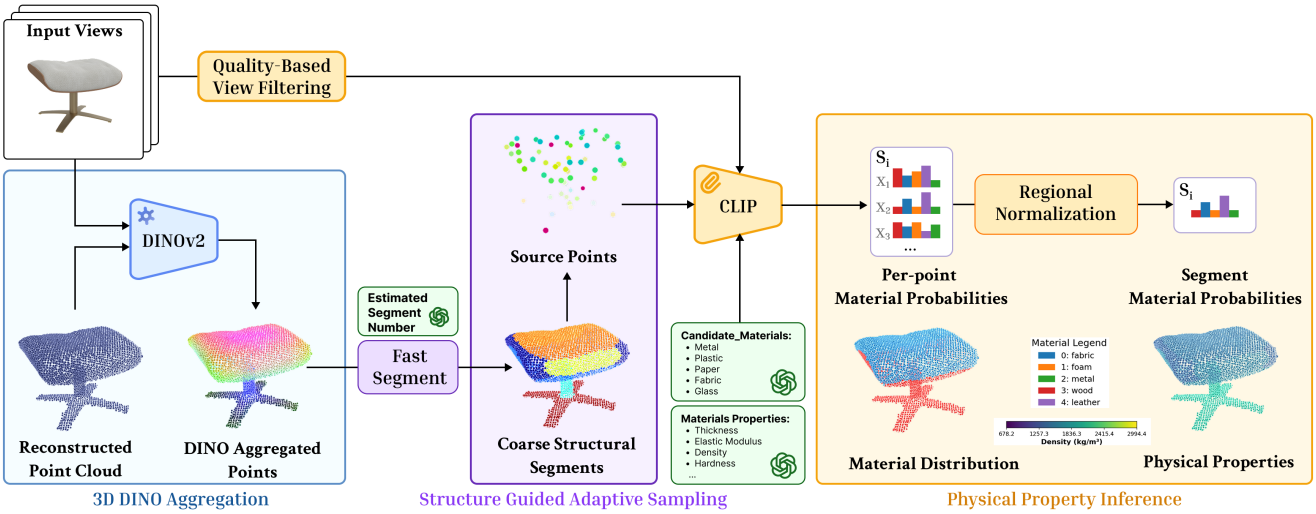

Figure 2. **Overview of S3-PHYS.** Given posed multi-view images, S3-PHYS lifts dense DINO features into 3D to form a structure-aligned feature field, segments the object into coarse components, and samples only a few representative points for CLIP encoding. After quality-based view filtering, we fuse CLIP features to estimate per-component material probabilities and predict downstream physical properties. Our design removes dense interpolation and enables fast, consistent 3D physical-property reasoning.

query complexity. Achieving fine-grained material resolution requires querying CLIP for a large number of source points, and each point must be processed from multiple visible views. Consequently, the number of CLIP calls scales multiplicatively with the number of points and views, quickly dominating runtime and making high-resolution inference computationally prohibitive.

• *Noisy vision features leading to unstable physical reasoning.* CLIP embeddings captured far away or from oblique viewpoints are frequently noisy (see Appendix), yet existing methods treat all visible views uniformly, allowing low-quality patches to pass through without filtering. These noisy features propagate through the dense interpolation stage, degrading material prediction accuracy and producing unstable estimates of physical properties. They further increase latency, as thousands of unreliable patches must still be processed to recover fine-grained detail.

In this paper, we propose S3-PHYS, an efficient yet more accurate framework for physical property reasoning that retains the semantic richness of CLIP-based methods while addressing the two core bottlenecks discussed above, as shown in Fig. 2. Our approach introduces three key ideas that together strike an effective balance between semantic fidelity and computational efficiency, as elaborated below.

(1) Structure-guided fast source-point selection. Instead of querying CLIP for every point–view pair, we use DINO features [4, 16] as efficient structural priors. DINO produces dense pixel-level features in a single forward pass per view, which we lift into 3D to obtain a dense feature field aligned with geometric and textural boundaries. This field provides a reliable signal for segmenting the scene into coarse, semantically coherent components. From each component, we then sample only a small set of representative points as CLIP source points. This significantly reduces the number of CLIP queries and improves computational efficiency while preserving the semantic detail needed for accurate material and physical property reasoning.

(2) Regional material normalization and property inference. Within each structural region, instead of relying on dense and potentially noisy interpolation, we normalize the per-point material probabilities of sparse source points to obtain a single, spatially consistent material probability distribution for the entire region. This regional normalization suppresses point-level semantic noise and enforces component-level coherence. The resulting distribution is then used to estimate the region-level physical property via kernel regression following [30] (Eq. 6), yielding smoother and more reliable material and physical property estimates.

(3) View-quality-aware patch filtering for robust CLIP embeddings. To further mitigate CLIP noise, we introduce a view-filtering module that evaluates each candidate view via geometric and photometric heuristics such as surface normal alignment, view angle, and illumination. Only high-quality patches are retained for CLIP encoding, which reduces noisy embeddings and decreases the number of CLIP calls by an order of magnitude. When combined with DINO-guided segmentation and adaptive sampling, this enables fast, stable, and high-resolution physical reasoning, lowering end-to-end runtime from hundreds of seconds to only a few seconds while improving both semantic coherence and physical accuracy.

We summarize the contributions of our work as follows:

- We provide, to our knowledge, the first analysis showing why existing CLIP-based 3D physical reasoning pipelines are slow and unstable, revealing two core bot-

tlenecks: (i) excessive CLIP queries caused by dense sampling, and (ii) noisy multi-view features that undermine material and physics inference.

- We introduce a structure-guided sampling strategy that projects 2D DINO features into 3D to obtain structural priors. This enables coarse component segmentation and reduces CLIP source-point queries by orders of magnitude while retaining semantic detail.
- We develop an efficient semantic fusion pipeline combining view-quality-aware patch filtering with probability-weighted component voting. This suppresses noisy viewpoints, stabilizes material estimation, and removes the need for dense point-wise interpolation.
- Extensive experiments on the ABO dataset show that our method improves semantic accuracy and spatial coherence, achieving **17.4 - 225.4× speedup** over prior CLIP-based baselines and enabling practical real-time 3D physical property reasoning.

2. Related Work

We divide related works into two categories.

Vision-based object's physical property reasoning. Inferring physical properties from visual observations has long been a core challenge in computer vision and cognitive science [5, 9, 26]. Prior work has explored dynamic reasoning by observing object motion [14, 26] or physical interactions in simulators [18, 28], and static reasoning from single images [2, 3, 22, 24]. While these methods can estimate quantities such as mass, friction, or elasticity, they rely on task-specific supervision or constrained experimental setups. Recent neural approaches extend visual physics reasoning into 3D. *NeRF2Physics* [30] embeds visual cues within NeRF fields but requires time-consuming volumetric optimization for each scene. 3D Gaussian–based physical reasoning algorithms [23, 27] accelerate this process via 3DGS reconstruction, yet they still suffer from instability under noisy multi-view inputs and high computational cost from redundant CLIP queries. Our method addresses these issues by introducing a DINO-guided 3D structural prior for robust component segmentation and adaptive sampling. This design not only suppresses noisy-view interference but also reduces CLIP invocations by orders of magnitude, enabling fast, consistent, and high-resolution physical reasoning within seconds rather than minutes.

3D language fields. A complementary line of work builds semantic 3D representations using vision–language models (VLMs). CLIP [20], trained on large-scale image–text pairs, has proven effective for zero-shot segmentation, localization, and open-world reasoning. Methods such as Distilled Feature Fields, LERF [13], and OpenNeRF [7] inject CLIP features into NeRFs for semantic 3D understanding, while approaches like Openscene [17, 31] map CLIP features onto point clouds, and feature splatting [19] propagates them into 3DGS representations. However, these pipelines typically rely on dense feature interpolation over millions of 3D points and exhaustive multi-view fusion, which greatly increases computation and exacerbates CLIP noise. In contrast, our method leverages CLIP more judiciously: we extract features only from sparse, high-quality patches selected via DINO-guided structural segmentation and view-quality heuristics, producing faster and more reliable 3D semantic fields.

3. Methods

In this section, we first present the system overview (§3.1). We then detail each design component, including structure-guided adaptive sampling (§3.2), efficient semantic fusion (§3.3), material proposal (§3.4), and regional-aware physical property inference (§3.5).

3.1. Overview

S3-PHYS takes as input a set of posed multi-view images and a reconstructed 3D space (e.g., point cloud from VGGT [25], NeRF [30], 3DGS [12, 23], or mesh reconstruction) and produces a spatially consistent physical-property map for any query point X. As illustrated in Fig. 2, the framework consists of three core stages:

Step One: Structure-guided adaptive sampling. We extract DINO features [16] from each input view and project them onto the 3D point cloud to obtain a dense per-point feature field that encodes both geometry and appearance. This lifted feature field serves as an efficient structural prior: it can be computed in a single multi-view pass and adds negligible overhead in practice (§5.4). Using these features, we segment the object into coarse, semantically coherent components and sample only a small number of representative points from each component as CLIP source points, substantially reducing the number of required CLIP queries.

Step Two: Efficient semantic fusion. For each representative point, we evaluate candidate views using geometric and photometric criteria and retain only the high-quality ones. We perform multi-scale CLIP feature fusion on these patches to obtain stable and reliable semantic embeddings.

Step Three: Material proposal and physical reasoning. In parallel, a vision-language model (VLM) analyzes a set of diverse input views to propose candidate materials and their physical properties. We fuse these proposals with the 3D semantic embeddings through *probability-weighted component voting*, yielding component-level material assignments and object-level physical property distributions.

These stages yield fast, noise-resistant, and semantically coherent 3D physical-property maps without relying on dense interpolation, reducing the end-to-end running latency from hundreds of seconds to 3.81s (§5).

3.2. Structure-Guided Adaptive Sampling

We consider the task of inferring per-point material semantics and downstream physical properties from a reconstructed 3D object. The input includes a point cloud $\{x_i\}$, and a set of posed multi-view images $\{I_v\}_{v\in V}$. For each 3D point x_i, the goal is to infer a material probability distribution and convert it into physical quantities. Most existing pipelines, such as NeRF2Physics, all rely on querying numerous source points from the object's point cloud to obtain sufficient semantic resolution. This design however creates a fundamental trade-off: sparse sampling triggers less CLIP encoding but produces inconsistent semantics, whereas dense sampling dramatically improves coverage at the cost of substantial runtime and increased noise.

To overcome this challenge, we first use DINO features as a geometric-semantic prior for adaptive sampling. For each input view $v \in V$, DINO produces dense pixel-level features $F_v(\cdot)$ in a single forward pass. We then lift these 2D features into 3D by projecting every point x_i onto each view through the camera projection function $\pi(x_i, v) \in \mathbb{R}^2$. Only views in which the point is actually visible contribute to its aggregated feature.

Aggregated DINO feature field. Let $M_{\text{vis}}(x_i, v)$ denote the visibility of point x_i from view v. The aggregated DINO feature for point x_i is defined as:

$$\Psi_{\text{DINO}}(x_i) = \frac{\sum_{v\in V} M_{\text{vis}}(x_i, v) \cdot F_v(\pi(x_i, v))}{\sum_{v_j\in V} M_{\text{vis}}(x_i, v_j)}. \quad (1)$$

Visibility function. A point is considered visible in a view if its projected depth is consistent with the depth map:

$$M_{\text{vis}}(x_i, v_j) = \begin{cases} 1, & \text{if } \ \text{dis}(x_i, v) \le \text{Depth}(\pi(x_i, v)) \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where $\text{dis}(x_i, v)$ is the depth of x_i along the camera ray of view v, and $\text{Depth}_v(\cdot)$ is the depth map rendered from that view.

Coarse structural segmentation. With the aggregated DINO field $\Psi_{\text{DINO}}(x_i)$, our next step is to partition the object into coarse, semantically coherent components that serve as regions for adaptive sampling. DINO features naturally align with geometric and textural boundaries, allowing structurally similar regions to be clustered together at negligible cost. However, the raw DINO embeddings are very high-dimensional, and K-means clustering is known to degrade in both accuracy and efficiency in such spaces. To address this, we apply PCA to obtain a compact representation of each aggregated feature (8D in our case), greatly reducing computational cost while preserving the relevant structure. Then we concatenate each PCA-reduced DINO feature with its 3D coordinate x_i, enabling the clustering to jointly consider appearance similarity and spatial proximity. Formally, the segment results are represented as follows:

$$\mathcal{S} = \text{KMeans}\left([f_{\text{PCA}}(\Psi_{\text{DINO}}(x_i)), x_i], N_S\right), \quad (3)$$

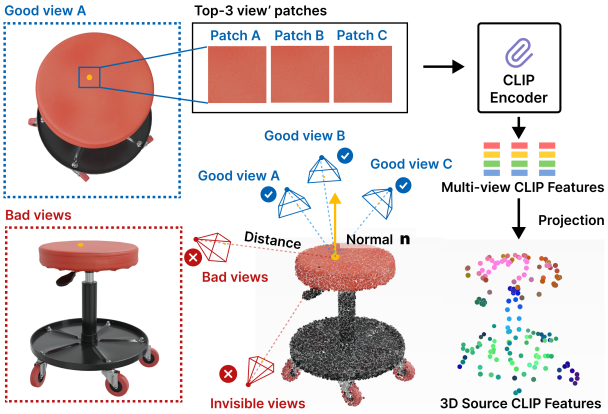

Figure 3. **Quality-based View Filtering** S3-PHYS excludes those bad views that contribute less to semantic fusion, thereby reducing the runtime latency.

where the number of clusters N_S is provided by the VLM during the material-proposal stage (§3.4). This effectively groups geometrically and visually coherent regions, establishing a structural prior for sampling. To ensure coverage of all structures regardless of size, we sample a fixed number of k=10 points from each segment S_i. This strategy drastically reduces computational overhead without sacrificing coverage. As detailed in Table 1, our approach reduces the average CLIP source points from 3,505 to 82 (42× reduction) while maintaining semantic diversity.

3.3. Efficient Semantic Fusion

Our semantic fusion module contains two components: quality-based view filtering and multi-scale feature fusion.

3.3.1. Quality-based View Filtering

After selecting representative 3D source points, we fuse CLIP features from multiple views to obtain stable semantic embeddings. However, multi-view images vary widely in quality: some views provide sharp, front-facing observations, while others suffer from blur, occlusion, extreme angles, or specular reflection, as shown in Fig. 3. Directly fusing CLIP features from all visible views amplifies noise and increases computation, a key weakness of prior CLIP-based pipelines. To mitigate this issue, we assess the quality of each visible view before extracting and fusing its CLIP features using two geometric factors:

- **Distance-induced resolution loss.** Faraway views contain fewer texture details and produce weak CLIP features, so they should be excluded.
- **View-surface angle mismatch.** Highly oblique viewing angles introduce foreshortening and reflection artifacts, which degrade semantic consistency.

We compute an importance score that combines distance and normal alignment for every point-view pair. Each sampled point has an estimated surface normal. For a point at

depth z from a view, the distance score is defined as follows:

$$S_{\text{dist}}(z) = \frac{1}{1+z},$$

The angle score is defined below:

$$S_{\text{angle}}(\mathbf{v}, \mathbf{n}) = \max\left(0,\ \mathbf{v} \cdot (-\mathbf{n})\right),$$

where $\mathbf{v}$ is the unit view direction and $\mathbf{n}$ is the outward normal. The importance score of an view is computed as:

$$S_{\text{total}} = S_{\text{dist}} \cdot S_{\text{angle}}.$$

We rank views by S_{total} and retain only the top-k (where k=3) for CLIP feature extraction. According to our experiments, this filtering reduces the number of embedded patches by approximately $2.7\times$ while improving robustness by discarding poorly aligned or low-quality observations, as shown in Table 1.

3.3.2. Multi-Scale Feature Fusion

After selecting the top-k views, we extract CLIP features from patches centered at the projected location of each sampled point. Using a single patch size often provides limited context: small patches fail to capture larger structures, while large patches smooth out fine texture. Following LeRF [13], we therefore aggregate features across multiple spatial scales to improve robustness. For each retained view, patches of different scales are encoded by the CLIP image encoder and averaged to form a multi-scale representation. For source point x, features from the top-k views are then averaged to obtain the final semantic embedding:

$$\mathbf{f}(x) = \frac{1}{k}\sum_{v=1}^{k}\left(\frac{1}{|\mathcal{L}|}\sum_{l\in\mathcal{L}}\phi\left(\mathcal{P}_v^{(l)}(x)\right)\right), \tag{4}$$

where $\phi(\cdot)$ is the CLIP encoder, $\mathcal{P}_v^{(l)}(s)$ is a patch of scale l extracted from view v, and $\mathcal{L}$ is the set of scales. This strategy preserves both fine-grained detail and coarse context while keeping computation manageable. As shown in Table 1, even with multi-scale processing, the total number of embedded patches remains $40\times$ smaller than the baseline.

Parallelize multi-scale feature fusion. To accelerate semantic feature fusion, we project all sampled points onto the selected views in one vectorized operation. Angle scores of all selected views are then batch-evaluated using pre-estimated normals, and all valid image patches are consolidated into a single global batch for CLIP encoding. This transforms the standard nested loop-based computation into a fully parallelized pass, substantially reducing latency.

With these optimizations, the entire semantic fusion stage, including CLIP feature extraction and fusion, completes within 1.2 s (Tab. 6).

3.4. Material Proposal

To ensure coverage of all visible materials, we select 2-4 views I_m with the largest pose differences from the input set and concatenate them into a single composite image. This composite image, together with a task-specific text prompt, is provided to the VLM to generate a detailed semantic description and a list of material candidates. For each material, the model predicts its approximate thickness within the object as well as corresponding physical property values. Formally, this process produces a material–property dictionary:

$$\mathcal{M} = \{(k_i, y_i, \theta_i)\}_{i=1}^{K},$$

where k_i is the material name, y_i is the associated set of physical properties or value ranges, and θ_i is the estimated thickness ratio. In addition, the VLM provides an estimated segment number N_S, which is used as the target cluster count in §3.2. The complete prompt and query design are provided in the Appendix.

3.5. Regional-aware Physical Inference

Given the spatially aggregated semantic features (§3.3.2) and the VLM-generated material proposals (§3.4), we next infer a consistent material distribution and its corresponding physical properties for each structural segment and subsequently estimate its corresponding physical properties to obtain the final object-level attributes. Because DINO features offer much finer granularity than CLIP embeddings, we assume that each DINO-derived segment S_i is dominated by a single material class.

Regional material normalization. To get robust material estimation, we first compute the material affinity score for each point and then apply regional material normalization. Specifically, for any point x and material candidate k, we define: $\omega_x(k) = \phi_{\text{CLIP}}\big(\mathbf{f}(x),\ \mathbf{t}(k)\big)$, where $\phi_{\text{CLIP}}(.)$ denotes cosine similarity, $\mathbf{f}(x)$ is the fused semantic feature at point x, $\mathbf{t}(k)$ is the CLIP text embedding of the k-th material label. Thus, $\omega_x(k)$ measures how well point x semantically matches material k in CLIP space.

Rather than relying on noisy point-wise predictions, we aggregate affinities over each structural segment to improve robustness. We compute the spatially averaged material score $\bar{\omega}_{S_i}(k)$ for the segment S_i using the equation:

$$\bar{\omega}_{S_i}(k) = \frac{1}{|S_i|}\sum_{x\in S_i}\omega_x(k). \tag{5}$$

We then assign the material with the maximum aggregated score to segment S_i.

Physical property inference. Once each segment S_i has been assigned a dominant material, we estimate its physical properties using the material–property dictionary provided by the VLM as shown in Fig. 2. Let $\{y_k\}_{k=1}^{K}$ denote the intrinsic physical values (e.g., density) associated with the K material candidates. To make the prediction robust, rather

Table 1. Workload reduction from each module in semantic fusion. SAS: Structure-Guided Adaptive Sampling; QVF: Quality-based View Filtering; MFF: Multi-scale Feature Fusion. Results from 5 scenes. Our full pipeline reduces processed patches from 28,040 to 702 (40×).

	Baseline	SAS (§3.2)	SAS + QVF (§3.3.1)	SAS + QVF + MFF (§3.3.2)
Avg. embedded source points	3,505	78	78	78
Avg. patches per point	8	8	3	9
Total embedded patches	28,040	624	234	702

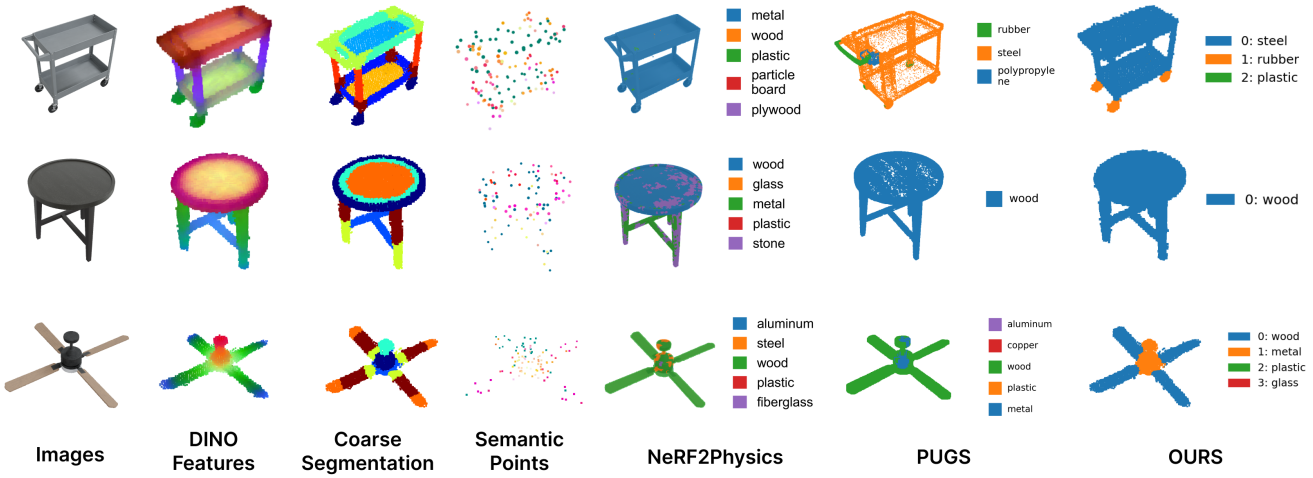

Figure 4. **Qualitative comparison on representative ABO objects.** For each object, we show the input RGB view, DINO features, coarse segementations, semantic points and material grounding. Compared with NeRF2Physics and PUGS, ours system produces cleaner, more coherent component boundaries and more accurate material assignments, leading to more reliable physical-property predictions.

than taking a hard material assignment, we compute a soft, probability-weighted estimate by applying a temperature-controlled softmax over the aggregated material affinities $\bar{\omega}_{S_i}(k)$. Here T is a temperature parameter that controls confidence sharpness (*lower T makes the distribution more peaked*, while higher T encourages smoother mixing across materials). Formally, the predicted physical property for segment S_i is represented as:

$$\rho(S_i) = \frac{\sum_{k=1}^{K} \exp\left(\bar{\omega}_{S_i}(k)/T\right) y_k}{\sum_{k=1}^{K} \exp\left(\bar{\omega}_{S_i}(k)/T\right)}. \tag{6}$$

This yields a single, stable physical-property estimate for the segment S_i, which is assigned uniformly to all points in S_i. To derive global object-level physical quantities such as total mass $\hat{\zeta}$, we accumulate material contributions from all segments as follows:

$$\hat{\zeta} = \sum_i \rho(S_i)\, \theta_i\, b^2\, \lambda, \tag{7}$$

where θ_i is the VLM-predicted thickness for segment S_i, b is the adaptive voxel size, and λ is the geometric correction factor from [30].

4. Implementation Details

We run all experiments on a workstation equipped with an NVIDIA A6000Ada GPU and an AMD Ryzen Threadripper PRO 5975WX CPU Processor, using `PyTorch`. We use DINOv2-B/14 [16] to extract 2D features, reduced to 8 dimensions by PCA before projection into 3D, and adopt OpenCLIP ViT-B/16 [11] as the image encoder for multi-scale patch embedding. The number of clusters N_S is estimated by the VLM during the material proposal stage, typically ranging from 6–14 depending on object complexity. For each segment, we sample $k_{sample} = 10$ representative source points and retain the top-3 views based on the view-quality score. Feature fusion and CLIP encoding are fully vectorized and executed in parallel, allowing the semantic fusion stage to complete within 1.2 s. The material proposal stage, based on GPT-4o [15] with 2 composite views, runs asynchronously. The full pipeline, including feature extraction, sampling, fusion, and voting, processes a complete scene in approximately 3.81 s on average, achieving 17.4 - 225.4× speedup over previous CLIP-based methods such as NeRF2Physics and PUGS.

5. Experiments

5.1. Metrics

We employ four complementary metrics to evaluate both physical accuracy and computational efficiency. (1) **Mass estimation error.** Following NeRF2Physics [30] and PUGS [23], we evaluate four quantitative measures between the predicted and measured mass on the ABO dataset (500 objects): (i) Absolute Deviation Error (ADE), (ii) Absolute

Table 2. **Accuracy results on ABO dataset (500 objects).** Lower is better for ADE/ALDE/APE; higher is better for MnRE.

Method	ADE↓	ALDE↓	APE(%)↓	MnRE(%)↑
NeRF2Physics [30]	12.725	0.736	1.040	0.564
PUGS [23]	9.461	0.661	0.767	**0.576**
Ours	**8.485**	**0.657**	**0.751**	0.571

Log-Deviation Error (ALDE), (iii) Absolute Percentage Error (APE), and (iv) Mean Relative Error (MnRE). (2) **Material segmentation IoU**, assessing spatial consistency across 100 ABO objects. (3) **Efficiency**, measured as the average scene processing time excluding point cloud reconstruction, for fair comparison with PUGS and NeRF2Physics.

For the segmentation metric, we unify all materials into 10 categories and manually annotate material masks on selected reference views for both datasets. The mean Intersection-over-Union (mIoU) is then computed between our predicted material distributions and the annotated ground truth.

5.2. Evaluation

We compare S3-PHYS against two recent representative CLIP-based baselines—NeRF2Physics [30] and PUGS [23]—which also perform material-aware physical property reasoning from visual inputs. Table 2 and Table 3 summarize the quantitative results for mass estimation and material segmentation.

Across all four mass-related metrics (ADE, ALDE, APE, and MnRE), our method consistently achieves competitive results, reducing ADE by 33% relative to NeRF2Physics and 10% relative to PUGS. The improvement mainly arises from our component-aware sampling and Regional aware physical inference, which enhance the semantic consistency of CLIP embeddings and eliminate interpolation artifacts commonly observed in previous methods.

Per-class results reveal complementary strengths: our method achieves clear gains on high-coverage, structurally coherent materials including **Plastic** (0.132 vs. 0.059/0.033), **Fabric** (0.752 vs. 0.720/0.634), and **Wood** (0.736 vs. 0.640/0.638), and remains competitive on **Ceramic** (0.850 vs. 0.865 for PUGS), while NeRF2Physics leads on **Metal** (0.297 vs. 0.202 ours) and PUGS on **Rubber** (0.367 vs. 0.135 ours). **Glass** yields near-zero IoU across all methods due to sparse pixel presence, and the **Other** category shows large inter-method variance, with NeRF2Physics scoring 0.362 while both PUGS and ours yield zero, likely reflecting its heterogeneous composition. Overall, the mIoU gains reflect stronger dataset-wide robustness on semantically coherent materials, with challenges on visually ambiguous or rare materials shared across methods.

In addition, our framework produces smoother and more spatially coherent material predictions. Qualitative comparisons in Fig. 4 show that NeRF2Physics and PUGS often exhibit fragmented or noisy material boundaries, while S3-PHYS yields uniform and physically plausible segmentation across diverse object categories. This demonstrates that leveraging DINO features as a structural prior not only improves quantitative performance but also enhances cross-component semantic coherence—an essential property for reliable physical reasoning in downstream applications. Fig. 5 exhibits S3-PHYS broad adaptability to multiple downstream physical properties.

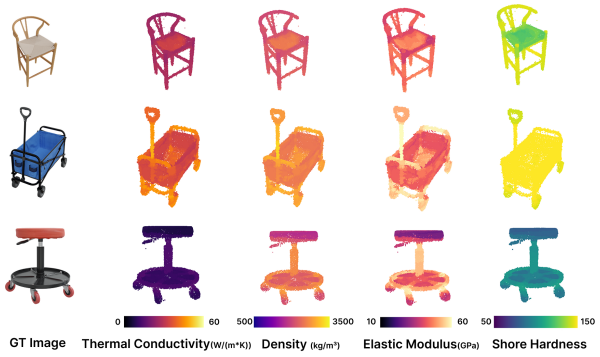

Figure 5. **Qualitative results in different physical properties.** Our method predicts four distinct physical properties—thermal conductivity, density, elastic modulus, and Shore hardness—across diverse object categories. The predicted distributions reflect material-level consistency: *e.g.*, the wooden chair exhibits uniformly low thermal conductivity and moderate density, while the metal components of the mechanic stool show distinctly higher hardness and elastic modulus values compared to its rubber wheels.

5.3. Ablation Studies

We conduct ablations on a 100-object subset of ABO to evaluate the contribution of each module in our pipeline. Starting from a naive CLIP-only baseline with kNN-based physical inference [30], we progressively add Structure-guided Adaptive Sampling (SAS), Regional-aware Physical Inference (RPI), Quality-based View Filtering (QVF), and Multi-Scale Feature Fusion (MFF). For a fair comparison, the naive baseline uses voxel downsampling to obtain approximately 80 source points, which matches the average number of points selected by SAS.

Table 4 summarizes the results. The naive baseline is the fastest, as it does not require DINO feature extraction or fusion. However, its accuracy is the lowest: with only a small number of source points, kNN-based inference struggles to preserve spatial granularity. Adding SAS alone does not resolve this issue.Although the sampling becomes more structured, the underlying kNN inference still collapses when the source set is sparse.

Introducing the RPI module leads to a substantial improvement: MnRE increases by over 12% without any run-

Table 3. **Per-class IoU and scene average mIoU comparison across methods.** Higher is better. The *Stone* category does not appear in the 100-object annotated subset and is therefore marked as N/A.

Method	Plastic	Rubber	Fabric	Metal	Wood	Ceramic	Glass	Stone	Other	Overall mIoU
NeRF2Physics	0.059	0.222	0.720	**0.297**	0.640	0.475	**0.015**	–	**0.362**	0.304
PUGS	0.033	**0.367**	0.634	0.235	0.638	**0.865**	0.000	–	0.000	0.428
Ours	**0.132**	0.135	**0.752**	0.202	**0.736**	0.850	0.000	–	0.000	**0.461**

Table 4. **Ablation on 100-object subset of ABO.** We progressively add SAS (Structure-guided adaptive Sampling), RPI (Regional-aware Physical Inference), QVF (Quality-based View Filtering), and MFF (Multi-Scale Feature Fusion) modules. Each delta percentage for MnRE relates to naive pipeline, and each delta runtime relates to its previous row. Naive denotes the baseline using sparse voxel sampling without further optimization.

Ablation Settings	MnRE↑	Time(s)↓
Naive	0.550	**1.73**
+ SAS	0.551 (+0.18%)	4.01 (+2.28s)
+ SAS + RPI	0.595 (+8.18%)	3.97 (-0.04s)
+ SAS + RPI + QVF	0.598 (+8.73%)	3.26 (-0.71s)
+ SAS + RPI + QVF + MFF	**0.602** (+9.45%)	3.44 (+0.18s)

time overhead. This demonstrates that region-level physical inference is significantly more robust than point-level voting when dealing with sparsely sampled observations.

Adding QVF further reduces runtime by 0.84s by discarding low-quality views before aggregation, while slightly improving MnRE. Finally, integrating MFF yields the best overall accuracy with only a modest increase in runtime, still lower than the unfiltered variants. Overall, each module contributes complementary benefits, jointly improving both accuracy and efficiency.

5.4. Efficiency Analysis

We further evaluate the runtime efficiency of our framework on the ABO dataset, measuring the average latency of each processing stage. As summarized in Table 5, our method achieves an average end-to-end processing time of only 3.81 s per scene (excluding reconstruction), corresponding to a 17.4× speedup over NeRF2Physics and a 225.5× speedup over PUGS. The improvement mainly stems from our structure-guided adaptive sampling and efficient semantic fusion, which substantially reduce the number of CLIP embeddings and patch queries required for material reasoning. The runtime scales linearly with the number of 3D points, demonstrating the method's suitability for large-scale deployment and real-time perception scenarios.

A detailed stage-wise breakdown is presented in Table 6. Among all modules, DINO feature extraction and semantic fusion account for the majority of processing time, contributing 42.2% and 30.7% of total runtime, respectively. DINO feature aggregation and adaptive sampling are relatively lightweight, accounting for 22.6% and 3.9%, while

Table 5. **Efficiency results on ABO dataset (500 objects).** Lower is better for runtime; higher is better for speedup.

Method	Time(s)↓	Speedup↑
NeRF2Physics [30]	66.1	x1.0
PUGS [23]	859.2	x0.08
Ours	**3.81**	**x17.4**

Table 6. Stage-wise latency breakdown per scene on ABO (excluding reconstruction).

Stage	Time(s)↓	Share(%)
DINO Feature Extraction	1.61	42.2
DINO Feature Aggregation	0.86	22.6
Structure Guided Adaptive Sampling	0.15	3.9
Efficient Semantic Fusion	1.17	30.7
Property inference	0.02	0.5
Total (Ours)	**3.81**	100

property inference is negligible (0.5%). This balanced runtime distribution indicates that the computational bottleneck has been largely mitigated, with no single component dominating the overall latency.

6. Conclusion

We have presented the design and evaluation of S3-PHYS, a lightweight, structure-guided framework for fast and reliable visual physical property reasoning. By combining DINO-derived structural priors with structure-guided adaptive sparse sampling, quality-based view filtering, and regional-aware physical inference, S3-PHYS achieves accurate and spatially coherent material understanding while reducing inference time from hundreds of seconds to only a few seconds per scene (achieving up to 225× speedup), which is an order-of-magnitude improvement over prior pipelines. Experiments on ABO dataset confirm strong gains in material grounding quality and physical property estimation. Current limitations include sensitivity to homogeneous surfaces in DINO-based segmentation and CLIP's difficulty distinguishing visually similar materials; future work will explore physically grounded cues and end-to-end differentiable formulations to address these challenges. Looking ahead, our approach establishes a practical foundation for real-time material perception in AR/VR, robotics, and multimodal digital twins.

References

[1] Syed Shah Alam, Samiha Susmit, Chieh-Yu Lin, Mohammad Masukujjaman, and Yi-Hui Ho. Factors affecting augmented reality adoption in the retail industry. *Journal of Open Innovation: Technology, Market, and Complexity*, 7(2):142, 2021. 1

[2] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM TOG*, 33(4):1–12, 2014. 3

[3] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild with the materials in context database. In *CVPR*, pages 3479–3487, 2015. 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pages 9650–9660, 2021. 2

[5] 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. *arXiv preprint arXiv:2209.09874*, 2022. 1, 3

[6] Wei Chow, Jiageng Mao, Boyi Li, Daniel Seita, Vitor Guizilini, and Yue Wang. Physbench: Benchmarking and enhancing vision-language models for physical world understanding. *arXiv preprint arXiv:2501.16411*, 2025. 1

[7] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. Opennerf: open set 3d neural scene segmentation with pixel-wise features and rendered novel views. *arXiv preprint arXiv:2404.03650*, 2024. 3

[8] Michael Fischer, Iliyan Georgiev, Thibault Groueix, Vladimir G Kim, Tobias Ritschel, and Valentin Deschaintre. Sama: Material-aware 3d selection and segmentation. *arXiv preprint arXiv:2411.19322*, 2024. 1

[9] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015. 3

[10] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935*, 2019. 1

[11] 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, July 2021. 6

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):139–1, 2023. 3

[13] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, pages 19729–19739, 2023. 1, 3, 5

[14] 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. *arXiv preprint arXiv:2303.05512*, 2023. 3

[15] OpenAI. Gpt-4o ("o" for omni) system card. Technical report, OpenAI, 2024. arXiv:2410.21276, https://doi.org/10.48550/arXiv.2410.21276. 6

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 6

[17] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023. 3

[18] 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. 3

[19] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature Splatting: Language-Driven Physics-Based Scene Synthesis and Editing, Apr. 2024. arXiv:2404.01223 [cs]. 1, 3

[20] 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. 1, 3

[21] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008. 1

[22] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *IJCV*, 103(3):348–371, 2013. 3

[23] Yinghao Shuai, Ran Yu, Yuantao Chen, Zijian Jiang, Xiaowei Song, Nan Wang, Jv Zheng, Jianzhu Ma, Meng Yang, Zhicheng Wang, et al. Pugs: Zero-shot physical understanding with gaussian splatting. *arXiv preprint arXiv:2502.12231*, 2025. 1, 3, 6, 7, 8

[24] Trevor Standley, Ozan Sener, Dawn Chen, and Silvio Savarese. image2mass: Estimating the mass of an object from its image. In *Conference on Robot Learning*, pages 324–333. PMLR, 2017. 1, 3

[25] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 3

[26] 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. *NeurIPS*, 28, 2015. 3

[27] Xinli Xu, Wenhang Ge, Dicong Qiu, ZhiFei Chen, Dongyu Yan, Zhuoyun Liu, Haoyu Zhao, Hanfeng Zhao, Shunsi Zhang, Junwei Liang, et al. Gaussianproperty: Integrating physical properties to 3d gaussians with lmms. In *CVPR*, pages 7231–7240, 2025. 1, 3

[28] Shaoxiong Yao and Kris Hauser. Estimating tactile models of heterogeneous deformable objects in real time. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12583–12589. IEEE, 2023. 3

[29] Albert J. Zhai, Yuan Shen, Emily Y. Chen, Gloria X. Wang,

Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical Property Understanding from Language-Embedded Feature Fields. pages 28296–28305, 2024. 1

[30] Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu Guan, and Shenlong Wang. Physical property understanding from language-embedded feature fields. In *CVPR*, pages 28296–28305, 2024. 1, 2, 3, 6, 7, 8

[31] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *CVPR*, pages 2048–2059, 2023. 3