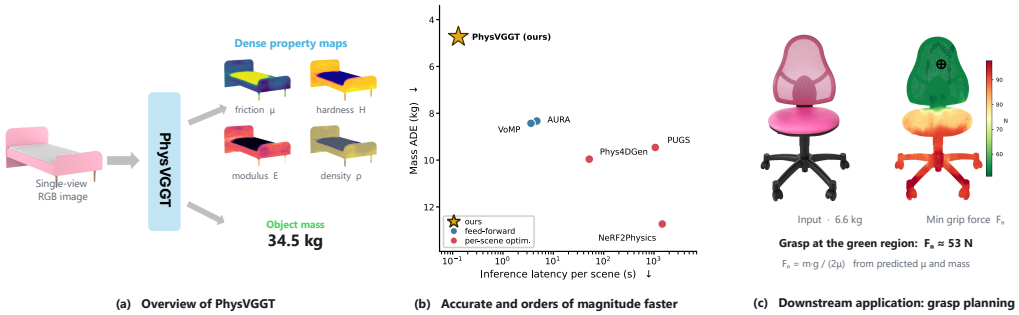
PHYSVGGT: FEED-FORWARD DENSE PHYSICAL PROPERTY ESTIMATION FROM A SINGLE IMAGE

Anonymous authors
Paper under double-blind review



(a) Overview of PhysVGGT (b) Accurate and orders of magnitude faster (c) Downstream application: grasp planning

Figure 1: PhysVGGT overview. (a) From a single RGB image, PhysVGGT predicts dense friction μ, hardness H, Young's modulus E, and density ρ, together with object-level mass. (b) PhysVGGT achieves accurate physical-property prediction while being $27\times$ faster than the previous state of the art. (c) The predicted friction and mass directly enable downstream grasp planning by estimating the minimum grip force $F_n = mg/(2\mu)$ and identifying a low-force contact region.

ABSTRACT

Physical properties, such as friction, hardness, stiffness, and density, govern how robots should grasp, manipulate and interact with objects, yet estimating these properties from RGB images remains challenging. Existing methods typically employ per-object reconstruction augmented with physical properties or directly query vision-language models at test time, which results in substantial computational overhead that limits their applicability. In this work, we present PhysVGGT, a feed-forward model that predicts dense maps of friction coefficient, Shore hardness, Young's modulus, and density, together with object-level mass, from a single RGB image in one forward pass. The key idea of PhysVGGT is to formulate physical property estimation as a dense per-pixel prediction problem and employ a visual geometry transformer to extract geometry-aware tokens from the input image followed by a dense prediction branch for estimating local physical properties and a global prediction branch for estimating object-level mass. In addition, we introduce a scalable pseudo-label generation pipeline that enables large-scale weakly supervised training for dense physical property prediction, substantially reducing the need for expensive direct physical measurements. Extensive experiments show that PhysVGGT achieves state-of-the-art performance on the ABO-500 dataset and generalizes effectively to the out-of-distribution NeRF2Physics dataset. Moreover, PhysVGGT eliminates the need for per-object reconstruction and test-time optimization, achieving an inference latency of only $0.13\,\mathrm{s}$ per image, making it $27\times$ faster than the previous state of the art.

1 INTRODUCTION

Physical properties are latent attributes of the visual world that, unlike geometry or semantics, are difficult to observe directly from images. Yet humans routinely infer them from visual cues, estimating whether a surface is rough or slippery, whether an object is soft or rigid, or whether it is likely to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

be light or heavy. Such reasoning extends visual perception beyond object recognition and geometric understanding toward the physical characteristics that govern how objects and surfaces behave and interact. Endowing vision systems with this capability is an important step toward richer understanding of physical scenes, with potential benefits for downstream tasks in robotics, autonomous driving, and physics-based applications (Xie et al., 2024; Gao et al., 2023).

Recent work has demonstrated that physical properties can nevertheless be estimated from visual observations. One of the most popular paradigms is to leverage Neural Radiance Field (NeRF) (Mildenhall et al., 2021) or 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) to reconstruct per-object 3D representations from multiple images and then perform semantic or material reasoning to infer physical properties (Zhai et al., 2024; Xu et al., 2025; Shuai et al., 2025; Chopra et al., 2026). While effective, this paradigm has significant computational overhead, as it requires expensive reconstruction and inference for each test object. Some recent works attempt to mitigate this overhead by using more efficient feed-forward approaches that bypass iterative per-object reconstruction. However, they still face limitations in terms of reliance on prepared 3D assets, test-time Vision-Language Models (VLMs), material retrieval, property lookup, and further post-processing (Dagli et al., 2026; Le et al., 2025; Lan et al., 2025). In addition, existing methods typically require *multiple views* of the scene to accurately infer physical properties, which further limits their applicability in real-world scenarios where only a single view may be available.

In this paper, we propose **PhysVGGT**, a feed-forward model that directly predicts dense physical-property maps of friction coefficient, Shore hardness, Young's modulus, and density, together with object-level mass, from a *single* RGB image in one forward pass. The key idea of PhysVGGT is to formulate physical property estimation as a dense per-pixel prediction problem and construct a feed-forward framework that directly maps geometry-aware visual features to physical properties. Specifically, PhysVGGT employs a visual geometry transformer backbone for extracting geometry-aware tokens from the input image, followed by a dense prediction branch for estimating local physical properties and a global prediction branch for estimating object-level mass. In this framework, PhysVGGT leverages three key components to achieve accurate and efficient physical property estimation. First, since different physical properties are inherently correlated through the underlying material and geometric characteristics, PhysVGGT introduces a cross-property coupling module into the dense prediction branch, which couples different physical properties at the feature level by attending across property-specific channels at each spatial location. This design exploits the strong correlations among friction, hardness, stiffness, and density, allowing the prediction of each property to be informed by the others rather than decoded independently. Second, PhysVGGT introduces a shared decoder with lightweight per-property heads, which replaces independent decoders with a single shared decoding trunk followed by small property-specific output heads. This design reduces the number of decoder parameters by approximately $4\times$, while the parameter sharing across the correlated prediction tasks provides an effective inductive bias and further improves prediction accuracy. Third, PhysVGGT introduces an object-level mass head that directly regresses object mass from pooled geometry-aware tokens using a lightweight MLP. This global prediction pathway is decoupled from the dense decoder, complementing the per-pixel physical-property maps with an object-level scalar estimate. Together, these designs enable PhysVGGT to train only $\sim 32\,\mathrm{M}$ parameters on top of a frozen $1.26\,\mathrm{B}$-parameter backbone, while achieving $27\times$ speedup compared to the current state of the art (Dagli et al., 2026) and yielding higher prediction accuracy (Figure 1).

In addition, we introduce a scalable pseudo-labeling pipeline that generates dense physical-property annotations directly from images, bypassing the need for curated 3D assets and manual physical measurements. Existing datasets do not provide dense per-pixel annotations of friction, hardness, stiffness, and density, posing a significant challenge for training feed-forward models at scale. Although there have been some physical property annotation pipelines, they either assign physical attributes to 3D assets, limiting their scalability to the size and diversity of curated asset collections, or infer physical properties through per-object reconstruction and optimization, which is computationally expensive and does not produce a reusable large-scale corpus of annotated images. In contrast, our pipeline combines part segmentation, material and hardness prediction, and cross-view refinement to generate high-quality pseudo-labels for dense physical-property maps. This allows us to create a large-scale dataset of densely annotated physical-property maps without the need for direct physical measurements, enabling the training of feed-forward models like PhysVGGT at scale.

We evaluate PhysVGGT on five datasets spanning in-distribution and out-of-distribution settings. On ABO (Collins et al., 2022), we evaluate mass on ABO-500 and qualitatively assess the dense

predictions for properties without measured per-pixel ground truth. In addition, we evaluate zero-shot generalization under a distribution shift to real measurements on NeRF2Physics (Zhai et al., 2024) and image2mass (Standley et al., 2017), and qualitatively study terrain generalization on RUGD (Wigness et al., 2019) and RELLIS-3D (Jiang et al., 2021). Our experiments show that PhysVGGT achieves state-of-the-art performance across multiple benchmarks while being faster than existing methods, specifically achieving a $27\times$ inference speedup compared with the previous state of the art. Our **contributions** are summarized as follows:

- We propose **PhysVGGT**, a feed-forward framework that directly predicts dense physical properties together with object-level mass from a single RGB image in one forward pass, eliminating per-object reconstruction and test-time optimization.
- We conduct extensive experiments and demonstrate that PhysVGGT achieves state-of-the-art performance while running $27\times$ faster than the previous state of the art.
- We propose a scalable pseudo-label generation pipeline that enables large-scale weakly supervised training of dense physical property prediction models, eliminating the need for expensive direct physical measurements.

2 Related Work

Physical Properties from Visual Observations. Early methods typically build per-object 3D representations and then reason about materials on top of them. NeRF2Physics (Zhai et al., 2024) augments a Neural Radiance Field with language-aligned features to transfer material properties to reconstructed geometry, while GaussianProperty (Xu et al., 2025), PUGS (Shuai et al., 2025) and PhysGS (Chopra et al., 2026) attach dense physical attributes to 3D Gaussians through multimodal reasoning, feature propagation and probabilistic fusion. These methods require per-object reconstruction before any physical reasoning can begin. Recent methods bypass the need for iterative reconstruction using feed-forward approaches. AURA (Lan et al., 2025) replaces iterative reconstruction with a single geometry foundation model pass, yet still queries CLIP, a VLM and a material database at test time, whereas VoMP (Dagli et al., 2026) and PIXIE (Le et al., 2025) predict volumetric properties from voxel or latent representations produced by a pretrained 3D model. Phys4DGen (Lin et al., 2025) embeds physical properties into generative 4D modelling, and Lee et al. (2026) regresses global mass from a monocular image under physically guided supervision. In contrast, PhysVGGT predicts dense friction, Shore hardness, Young's modulus, and density maps, together with object-level mass, from a single RGB image in one forward pass, with neither reconstruction nor any auxiliary semantic model at inference.

Physical Property Data Annotation. Since dense physical measurements are labor-intensive to acquire at scale, recent work derives supervision from pretrained models using SAM 2 (Ravi et al., 2025) for part localization and VLMs combined with material databases for property assignment. VoMP (Dagli et al., 2026) labels the parts of pre-segmented 3D objects with a VLM and a material database and propagates the result to voxels, while PIXIE (Le et al., 2025) assembles PixieVerse semi-automatically by distilling a proprietary VLM, CLIP and human priors into per-object material classes. Both methods are bounded by the scale of curated 3D asset collections and neither covers friction or Shore hardness. On the other hand, per-scene pipelines such as NeRF2Physics (Zhai et al., 2024) and GaussianProperty (Xu et al., 2025) instead query GPT-4/GPT-4V for one reconstructed object at a time and therefore never yield a reusable training corpus. We follow the weak-supervision direction, but with an offline pipeline that operates purely on multiview product imagery, needs no 3D assets, and produces dense per-pixel supervision at an order of magnitude more objects.

3 Method

Problem Statement. In this work, PhysVGGT focuses on directly predicting dense physical properties and object-level mass from a single RGB image in a feed-forward manner, rather than relying on 3D reconstruction or iterative optimization. Let $I \in \mathbb{R}^{3\times H\times W}$ denote a single RGB image of an object. We consider four dense physical properties $\mathcal{C} = \{\mu, H, E, \rho\}$, corresponding to the coefficient of friction, Shore hardness, Young's modulus and density. For each property $c \in \mathcal{C}$ the goal is to predict a dense property map $\widehat{Y}^c \in [0,1]^{H\times W}$, together with an object-level mass estimate

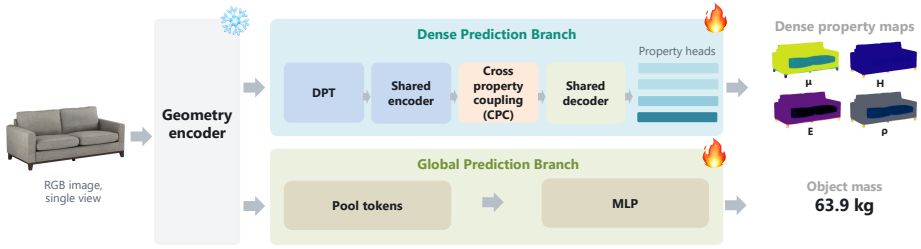


Figure 2: The overall framework of PhysVGGT.

$\widehat{M} \in \mathbb{R}_{>0}$. PhysVGGT therefore formulates dense physical property estimation as a feed-forward prediction problem $(\widehat{\mathbf{Y}}, \widehat{M}) = f_\Theta(I)$, where $\widehat{\mathbf{Y}} = \{\widehat{Y}^c \mid c \in \mathcal{C}\}$ is the set of predicted dense property maps and f_Θ the proposed network.

3.1 Feed-Forward Physical Property Estimation: PhysVGGT

Geometry Encoder. Visual geometry transformers (Wang et al., 2025; 2026; Zhuo et al., 2026) have shown strong capabilities in capturing 3D spatial structure from 2D images and been demonstrated well-suited for serving as backbones in scene understanding tasks (Wu et al., 2025). Inspired by these advances, we adopt a visual geometry transformer as the backbone (geometry encoder Φ_{geo}) to extract geometry-aware token representations T from the input RGB image I, which serve as the foundation for subsequent physical property prediction:

$$T = \Phi_{\text{geo}}(I). \tag{1}$$

Then, as shown in Figure 2, PhysVGGT predicts the dense physical property maps $\widehat{\mathbf{Y}}$ and the object-level mass $\widehat{M}$ through two parallel prediction branches.

Dense Prediction Branch. We first convert the geometry tokens into a dense spatial representation using a DPT-style feature extractor (Ranftl et al., 2021). Multi-scale token representations from multiple transformer stages are projected into a common feature space and hierarchically fused through a top-down refinement pathway to produce a high-resolution dense feature representation:

$$F = \mathcal{H}_{\text{DPT}}(T), \tag{2}$$

where $F \in \mathbb{R}^{d_f \times H \times W}$. The dense feature map is subsequently processed by a shared encoder that progressively aggregates local and global context while reducing the spatial resolution:

$$(S_1, \ldots, S_L, B) = \mathcal{E}_\theta(F), \tag{3}$$

where S_l denotes the skip feature at the lth resolution and $B \in \mathbb{R}^{d_b \times H_b \times W_b}$ is the bottleneck feature map containing the highest-level semantic representation. Although each physical property is predicted independently by a separate head, these properties are governed by shared underlying material characteristics and exhibit strong physical correlations. To explicitly model these dependencies, we introduce a *Cross-Property Coupling (CPC)* module that operates on the shared bottleneck representation and performs self-attention across property-specific latent representations. For each property $c \in \mathcal{C}$, the shared bottleneck feature is projected into a property-specific embedding:

$$z_{c,x}^{(0)} = A_c(B(x)) + e_c, \tag{4}$$

where A_c is a learnable projection, e_c is a learnable property embedding, and x denotes a spatial location. The embeddings are grouped as $Z_x^{(0)} = [z_{c,x}^{(0)}]_{c \in \mathcal{C}}$ and refined by a lightweight transformer, $Z_x = \mathcal{T}(Z_x^{(0)})$, which enables information exchange among correlated physical properties at each spatial location. The coupled representations are projected back to the shared feature space through residual connections:

$$B_c(x) = B(x) + O_c(Z_{x,c}), \tag{5}$$

where O_c denotes a learnable output projection. The output projections are initialized to zero so that the CPC module initially behaves as an identity mapping and gradually learns residual cross-property interactions during training. Each property-specific bottleneck representation is independently processed by a shared decoder that progressively restores the spatial resolution using the

corresponding encoder skip features:

$$U_c = \mathcal{D}_\phi(B_c, S_1, \ldots, S_L), \tag{6}$$

and followed by a lightweight property-specific prediction head:

$$\widehat{Y}^c = \sigma(h_c(U_c)), \tag{7}$$

where h_c is the property-specific prediction layer and $\sigma(\cdot)$ denotes the sigmoid activation. Sharing the feature extractor, encoder, and decoder enables efficient multi-task learning while allowing the CPC module to capture dependencies among correlated physical properties and the prediction heads to preserve property-specific calibration.

Global Prediction Branch. In parallel, object-level mass is predicted directly from the extracted geometry tokens. Since object mass depends primarily on global object geometry rather than local material characteristics, the geometry tokens are globally pooled, batch-normalized per feature to preserve the magnitude cue encoding object scale and processed by an MLP:

$$\bar{t} = \mathrm{Pool}(T), \qquad \widehat{\ell}_M = g_\psi(\bar{t}), \qquad \widehat{M} = \exp(\widehat{\ell}_M), \tag{8}$$

where $\widehat{\ell}_M$ is the predicted logarithmic mass. Unlike previous approaches that estimate mass from predicted density and reconstructed volume, we directly regress mass from the global geometry-aware representation. This avoids error accumulation from intermediate density and volume estimation while allowing the mass branch to specialize independently from the dense prediction branch.

3.2 Training Objective

PhysVGGT is jointly optimized for dense physical-property prediction and object-level mass estimation. Since the dense supervision is automatically generated (Section 3.3) and may contain local noise or calibration errors, we combine a robust reconstruction objective with a pairwise ranking loss that emphasizes reliable relative relationships. Let Y^c and $\widehat{Y}^c$ denote the pseudo ground-truth and predicted maps for property $c \in \mathcal{C}$, respectively. Given the foreground mask M_{fg}, we define the valid pixels as $\Omega_{\mathrm{fg}} = \{x \mid M_{\mathrm{fg}}(x) = 1\}$ and apply the robust Charbonnier penalty (Charbonnier et al., 1994):

$$\mathcal{L}_c^{\mathrm{rec}} = \frac{1}{|\Omega_{\mathrm{fg}}|} \sum_{x \in \Omega_{\mathrm{fg}}} \rho\Big(\widehat{Y}^c(x) - Y^c(x)\Big). \tag{9}$$

Absolute pseudo-label values can be noisy even when their relative ordering is reliable. We therefore complement reconstruction with a ranking objective that preserves the ordering of physical properties across regions. To avoid ambiguous supervision, we retain only pixel pairs whose pseudo-label difference exceeds a threshold:

$$\mathcal{Q}_c = \big\{(i,j) \in \Omega_{\mathrm{fg}}^2 : |Y^c(i) - Y^c(j)| > \varepsilon_{\mathrm{rank}}\big\}. \tag{10}$$

For each retained pair, we define $s_{ij}^c = \mathrm{sign}(Y^c(i) - Y^c(j))$ and minimize

$$\mathcal{L}_c^{\mathrm{rank}} = \frac{1}{|\mathcal{Q}_c|} \sum_{(i,j) \in \mathcal{Q}_c} \mathrm{softplus}\left(-\frac{s_{ij}^c\big(\widehat{Y}^c(i) - \widehat{Y}^c(j)\big) - \gamma}{\tau}\right), \tag{11}$$

where τ is the temperature and γ is the ranking margin. We sample 2048 foreground pairs per image and property and use $\tau = 0.1, \gamma = 0$, and $\varepsilon_{\mathrm{rank}} = 0.02$. Since the properties are normalized to $[0, 1]$, the threshold removes pairs differing by less than 2% of the prediction range. The dense objective combines reconstruction and ranking across all properties:

$$\mathcal{L}_{\mathrm{dense}} = \sum_{c \in \mathcal{C}} \lambda_c \left(\mathcal{L}_c^{\mathrm{rec}} + \lambda_{\mathrm{rank}} \mathcal{L}_c^{\mathrm{rank}}\right), \tag{12}$$

where λ_c balances individual properties and λ_{rank} controls the ranking contribution. For object-level mass, we regress log-mass using an ℓ_1 loss:

$$\mathcal{L}_{\mathrm{mass}} = \Big|\widehat{\ell}_M - \log M\Big|, \tag{13}$$

where M is the ground-truth object mass. Samples without valid mass annotations contribute only to the dense objective. The final training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{dense}} + \lambda_{\mathrm{mass}} \mathcal{L}_{\mathrm{mass}}. \tag{14}$$

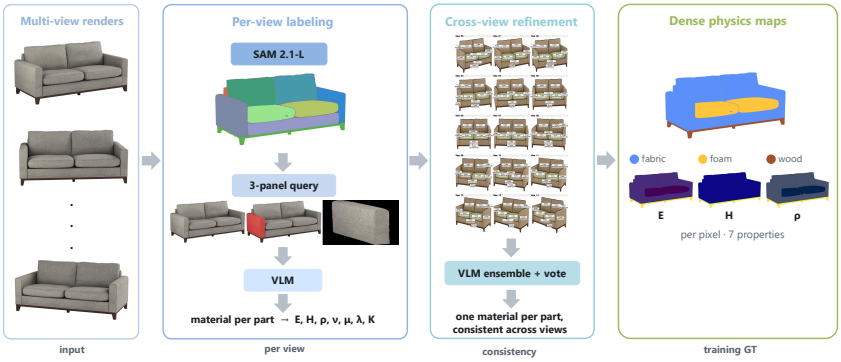


Figure 3: The overview of the proposed pseudo-label generation pipeline.

3.3 Pseudo-Label Generation Pipeline

Since large-scale datasets with dense physical-property annotations are unavailable, we construct an offline pseudo-label generation pipeline (Figure 3) to produce supervision for PhysVGGT. Given multi-view images $\mathcal{I}$ of an object, the set of generated dense physical property maps $\mathbf{Y} = \{Y^c \mid c \in \mathcal{C}\}$ is obtained by:

$$\mathbf{Y} = \mathcal{G}(\mathcal{I}), \tag{15}$$

where $\mathcal{G}$ denotes the pseudo-label generation pipeline that works in three stages: part segmentation, material and hardness prediction, and cross-view refinement. **(I) Part segmentation.** Each view is decomposed into part-level regions with SAM 2 (Ravi et al., 2025) to divide image into parts. This is important since different parts may contain different material and have different physical properties. **(II) Material and hardness prediction.** For every part we construct a three-panel query image, including the full object, the object with the part highlighted, and a tight crop of the part, and prompt a VLM to return a material category from a fixed 12-material vocabulary together with a Shore hardness range and its scale (Shore A or Shore D). The three-panel format supplies global object context and local appearance detail simultaneously, and predictions below a confidence threshold are discarded. **(III) Cross-view refinement.** Since the views are labelled independently, the same physical part may be assigned different materials in different views. All views are therefore tiled into a single composite image and presented to the VLM with a prompt that asks it to locate and correct the inconsistent parts; the corrections are combined by majority vote over $(\text{view}, \text{part})$ pairs, yielding a single material per part that is consistent across all views.

The physical properties are then derived from the refined labels. Young's modulus is computed from the predicted Shore hardness through an empirical hardness-modulus relation and clipped to the literature range of the assigned material, while density, Poisson's ratio and the coefficient of friction are read from a material-property table compiled from standard engineering references. The Lamé parameters and the bulk modulus follow in closed form from E and ν. Finally, the per-part values are rasterized onto the image plane to produce the dense supervision $\mathbf{Y}$. Object-level mass annotations are obtained independently from the ABO product metadata. The resulting training dataset is

$$\mathcal{D} = \{(\mathcal{I}, \mathbf{Y}, M)\}, \tag{16}$$

where M denotes the object-level mass annotation.

4 Experiments

Datasets. We evaluate PhysVGGT on two benchmarks used by prior work on physical property estimation. *(i) Mass:* the ABO-500 test split (Zhai et al., 2024) (100 held-out objects with catalogue-measured ground-truth weight). *(ii) Friction & hardness:* the NeRF2Physics 13-scene real dataset (Zhai et al., 2024), which provides sparse in-the-wild measurements of the kinetic friction coefficient (μ) and Shore hardness on real objects; this is an out-of-distribution (OOD) test, disjoint from our ABO training data. For qualitative out-of-distribution generalization we additionally use the off-road semantic-segmentation datasets RUGD (Wigness et al., 2019) and RELLIS-3D (Jiang et al.,

Table 1: Mass estimation on ABO-500 (100 test objects). Baselines quoted from the VoMP protocol (Dagli et al., 2026) and the respective papers. ↓/↑ denote lower/higher is better. Best in **bold**. ⋆Taken from existing work for reference purpose.

Method	ADE (kg)↓	ALDE↓	MnRE↑	Paradigm
NeRF2Physics (Zhai et al., 2024)	12.73	0.736	0.564	per-scene NeRF + VLM
PUGS (Shuai et al., 2025)	9.46	0.661	0.576	per-scene 3DGS
Phys4DGen⋆ (Lin et al., 2025)	9.96	0.664	0.566	3DGS + MLLM
VoMP (Dagli et al., 2026)	8.43	0.631	0.576	feed-forward
PGVME (Lee et al., 2026)	8.32	0.998	–	feed-forward
AURA (Lan et al., 2025)	8.33	0.582	0.634	feed-forward + VLM
PhysGS⋆ (Chopra et al., 2026)	8.25	0.999	0.474	per-scene 3DGS + VLM
GaussianProperty (Xu et al., 2025)	5.96	0.744	0.559	3DGS + MLLM
PhysVGGT (ours)	**4.73**	**0.574**	**0.655**	feed-forward

Table 2: Friction and Shore hardness on the NeRF2Physics 13-scene real benchmark. Reproduced baselines under the protocol of Zhai et al. (2024).

	Friction μ				**Shore hardness**			
Method	ADE↓	ALDE↓	APE↓	MnRE↑	ADE↓	ALDE↓	APE↓	MnRE↑
NeRF2Physics (Zhai et al., 2024)	0.217	0.459	0.446	0.661	66.21	0.672	1.105	0.546
PhysGS (Chopra et al., 2026)	0.218	0.462	0.438	0.647	34.82	0.456	0.532	0.671
GPT-4V (Achiam et al., 2023)	0.209	0.430	0.549	0.692	32.75	0.330	0.304	0.758
CLIP (Radford et al., 2021)	0.222	0.455	0.602	0.654	32.86	0.294	0.266	0.774
PhysVGGT (ours)	**0.102**	**0.206**	**0.225**	**0.831**	**7.05**	**0.103**	**0.111**	**0.908**

Table 3: Inference efficiency. Per-item inference latency. Baseline latencies are reported from (Dagli et al., 2026; Lan et al., 2025) and the corresponding papers. Lower is better.

Method	Latency (s)↓	Test-time dependency
NeRF2Physics (Zhai et al., 2024)	1454.6	NeRF optimization + VLM
PUGS (Shuai et al., 2025)	1058.3	3DGS optimization
Pixie (Le et al., 2025)	201.6	Scene optimization
Phys4DGen (Lin et al., 2025)	51.7	3DGS + MLLM
AURA (Lan et al., 2025)	4.7	CLIP + VLM
VoMP (Dagli et al., 2026)	3.6	DINOv2 + voxelization
PhysVGGT (ours)	**0.13**	**None**

2021). We train PhysVGGT on ABO-8k, a set of 8,213 household objects drawn from the Amazon Berkeley Objects catalogue (Collins et al., 2022), each rendered from 15 calibrated views (∼123k images), with per-view dense property maps. Object-level *mass* uses the catalogue item weight, available for 7,326 of the objects. The model is trained directly on these maps without any per-scene optimization. To prevent leakage, the *100 ABO-500 test objects are held out* from all training.

Metrics. Following (Zhai et al., 2024; Dagli et al., 2026), we report the Absolute Distance Error (ADE, in physical units), the Absolute Log-Distance Error $\mathrm{ALDE} = |\log \hat{y} - \log y|$, the Absolute Percentage Error $\mathrm{APE} = |\hat{y} - y|/y$, and the Min-of-Ratio Error $\mathrm{MnRE} = \min(\hat{y}/y,\, y/\hat{y}) \in (0, 1]$.

4.1 Comparison with State-of-the-Arts

Mass. Table 1 reports object-level mass estimation on ABO-500. PhysVGGT improves on the best previously reported value of every metric: it lowers ADE by 21% relative to GaussianProperty, the strongest prior result on that metric, and by 43% relative to AURA, the strongest prior method on ALDE and MnRE, while also improving ranking consistency, and it outperforms NeRF- and 3DGS-based pipelines despite using a single RGB image at inference. Directly regressing mass from

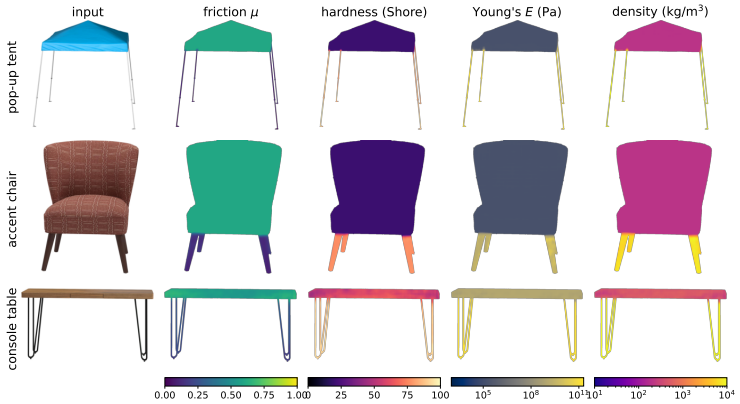


Figure 4: Dense physical property predictions. From the single input view (left), PhysVGGT predicts per-pixel friction μ, Shore hardness, Young's modulus E and density in one forward pass.

Table 4: Component-wise ablation. N2P MnRE on a fixed 2000-object subset using identical training settings, averaged over four epochs.

Configuration	Friction MnRE↑	Hardness MnRE↑
Full (trunk + coupling + shared dec. + combined loss)	**0.831**	**0.908**
− w/o cross-property coupling	0.805	0.869
− w/o shared decoder (→ 4 independent)	0.810	0.860
− w/o combined loss (→ plain L1)	0.750	0.784

geometry-aware representations is thus a more accurate and robust alternative to reconstruction- or density-based estimation.

Friction and Hardness. Table 2 reports zero-shot transfer to the out-of-distribution NeRF2Physics real benchmark. PhysVGGT is best on every metric for both properties, reducing friction ADE by 51.2% (MnRE +20.1%) and Shore hardness ADE by 78.5% (MnRE +17.3%), outperforming both reconstruction-based and vision-language baselines on unseen real objects.

Efficiency. PhysVGGT infers in a single feed-forward pass with no per-scene optimization and no test-time VLM, requiring only 0.13 s per image, a 27× reduction over VoMP (Dagli et al., 2026), the fastest existing feed-forward method (Table 3). Figure 1 plots this accuracy-latency trade-off against prior work. The proposed physics head contains only 32.4M trainable parameters on top of a frozen StreamVGGT backbone and requires just 4.1 TFLOPs per image.

4.2 Qualitative Analysis

Figure 4 shows dense predictions for all four properties on multi-material objects. No benchmark provides *measured-E* or density ground truth, so a head-to-head comparison is not possible, but the maps are physically consistent and material-aware: the model separates the tent's fabric canopy from its metal poles, the chair's upholstery from its wooden legs, and the table's wooden top from its steel hairpin legs, assigning the metal parts high stiffness and density but low friction.

4.3 Ablation Study and Discussion

Component-Wise Ablation Study. We compare the *Full* model against leave-one-out variants on the same 2000-object subset with identical hyperparameters, reporting N2P MnRE averaged over four epochs. Table 4 shows that the combined loss is the main contributor: replacing it with L1 reduces friction and hardness by 9.7% and 13.7%, respectively. In contrast, removing cross-property coupling costs 3.1% friction and 4.3% hardness, and removing the shared decoder 2.5% and 5.3%, both far smaller than the loss. Overall, the Full model provides the best performance.

Table 5: Cross-property coupling ablation. Comparison of different feature interaction mechanisms for modeling dependencies among physical properties.

Coupling	Friction MnRE↑	Hardness ADE↓	Mass MnRE↑
w/o CPC	0.802	9.09	0.474
Attention, 1-layer	0.822	10.79	0.468
Attention, 2-layer (default)	**0.831**	**7.05**	**0.655**
Cross-stitch	0.801	10.16	0.488
FiLM	0.790	13.50	0.469

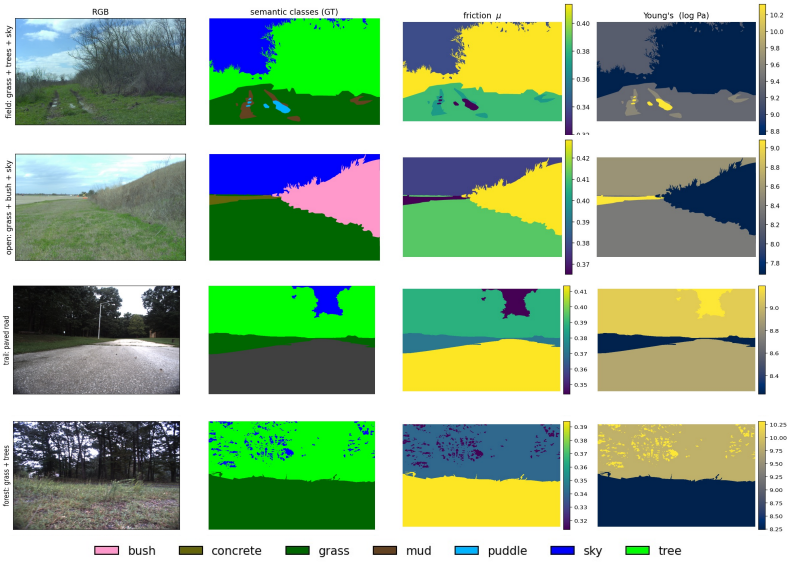


Figure 5: Out-of-distribution scenes on RUGD (upper two examples) and RELLIS-3D (lower two examples). Ground-truth semantic regions coloured by our predicted friction and Young's modulus. The model differentiates unseen materials.

Cross-Property Coupling. Table 5 compares strategies for modeling interactions among physical properties. Against the no-CPC variant, the proposed two-layer attention module improves friction MnRE by 3.6%, reduces hardness ADE by 22.4% and improves mass MnRE by 38.2%. Cross-stitch networks (Misra et al., 2016) and FiLM (Perez et al., 2018) capture only limited interactions, confirming that attending across friction, hardness, Young's modulus and density is what matters.

Domain Generalization Capability. Generalization of PhysVGGT holds beyond object-centric imagery. Figure 5 shows dense predictions on the off-road RUGD (Wigness et al., 2019) and RELLIS-3D (Jiang et al., 2021) scenes: despite the shift from household objects to natural terrain, the predicted friction and Young's modulus still differentiate materials never seen during training.

Please refer to the Appendix for more ablation study and hyperparameter sensitivity analysis.

5 Conclusion

This work presents PhysVGGT, a feed-forward framework that estimates dense physical properties, including friction, Shore hardness, Young's modulus, and density, together with object-level mass, from a single RGB image. PhysVGGT consists of a geometry encoder that extracts appearance-aware 3D representations from a single RGB image, followed by a dense prediction branch that estimates dense physical properties and a global prediction branch that estimate object-level mass. Extensive experiments show that PhysVGGT outperforms state-of-the-art methods in terms of both prediction accuracy and inference efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language fluency and grammar checking of the manuscript. We also used a vision-language model (VLM) as part of our pseudo-label generation pipeline, as described in the paper. We did not use generative AI tools to develop the core research ideas or to interpret the experimental results. All AI-assisted content and outputs were reviewed by the authors. We take full responsibility for the final content of this work, including the text, claims, results, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work uses only existing publicly available datasets and does not involve human subjects or personally identifiable information. All datasets are used in accordance with their respective licenses and terms of use. While PhysVGGT can be integrated into various downstream applications, we encourage responsible use and deployment in real-world systems.

REPRODUCIBILITY STATEMENT

The main paper and appendix provide detailed descriptions of the model architecture, training and pseudo-labeling procedures, evaluation protocols, hyperparameters, and ablation settings. All experiments are conducted using the cited publicly available datasets and pretrained backbone.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st International Conference on Image Processing*, volume 2, pp. 168–172. IEEE, 1994.

Samarth Chopra, Jing Liang, Gershom Seneviratne, and Dinesh Manocha. PhysGS: Bayesian-inferred gaussian splatting for physical property estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18980–18990, 2026.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. ABO: Dataset and benchmarks for real-world 3D object understanding. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21094–21104. IEEE, 2022.

Rishit Dagli, Donglai Xiang, Vismay Modi, Charles Loop, Clement Fuji Tsang, Anka He Chen, Anita Hu, Gavriel State, David I. W. Levin, and Maria Shugrina. VoMP: Predicting volumetric mechanical property fields. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ruohan Gao, Yiming Dou, Hao Li, Tanmay Agarwal, Jeannette Bohg, Yunzhu Li, Li Fei-Fei, and Jiajun Wu. The ObjectFolder benchmark: Multisensory learning with neural and real objects. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17276–17286. IEEE, 2023.

Peng Jiang, Philip Osteen, Maggie Wigness, and Srikanth Saripalli. RELLIS-3D dataset: Data, benchmarks and analysis. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1110–1116. IEEE, 2021.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

Hongbo Lan, Zhenlin An, Haoyu Li, Vaibhav Singh, and Longfei Shangguan. Accelerating physical property reasoning for augmented visual cognition. *arXiv preprint arXiv:2511.03126*, 2025.

Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. PIXIE: Fast and generalizable supervised learning of 3D physics from pixels. *arXiv preprint arXiv:2508.17437*, 2025.

Sungjae Lee, Junhan Jeong, Yeonjoo Hong, and Kwang In Kim. Physically guided visual mass estimation from a single RGB image. *arXiv preprint arXiv:2601.20303*, 2026.

Jiajing Lin, Zhenzhong Wang, Dejun Xu, Shu Jiang, Yunpeng Gong, and Min Jiang. Phys4DGen: Physics-compliant 4D generation with multi-material composition perception. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10398–10407, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168. IEEE, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

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. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4478–4485. IEEE, 2025.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026.

Maggie Wigness, Sungmin Eum, John G Rogers, David Han, and Heesung Kwon. A RUGD dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5000–5007. IEEE, 2019.

Xianfeng Wu, Yajing Bai, Minghan Li, Xianzu Wu, Xueqi Zhao, Zhongyuan Lai, Wenyu Liu, and Xinggang Wang. 4DLangVGGT: 4D language-visual geometry grounded transformer. *arXiv preprint arXiv:2512.05060*, 2025.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D gaussians for generative dynamics. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389–4398. IEEE, 2024.

Xinli Xu, Wenhang Ge, Dicong Qiu, Zhifei Chen, Dongyu Yan, Zhuoyun Liu, Haoyu Zhao, Hanfeng Zhao, Shunsi Zhang, Junwei Liang, and Ying-Cong Chen. GaussianProperty: Integrating physical properties to 3D gaussians with LMMs. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7231–7240. IEEE, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28296–28305. IEEE, 2024.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

A Appendix

In this section, we present more details on the implementation and discuss additional experimental results. The following is an overview of the organization of the appendix.

A.1 Implementation Details
A.2 Limitation and Future Work
A.3 Architectural Design
A.4 More Ablation Study
A.5 Hyperparameter Sensitivity
A.6 Further Discussion and Analysis

A.1 Implementation Details

We implement our approach with Python and PyTorch. We initialize the geometry encoder backbone with pretrained weights from Zhuo et al. (2026), freeze all backbone parameters ($\sim$1.26B), and train only the proposed shared encoder-decoder, CPC module and physics heads ($\sim$32.4M). Models are optimized using AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0.9, 0.95)$, a learning rate of 4×10^{-5}, weight decay 10^{-4}, batch size 4, and gradient clipping with a maximum norm of 1.0. During training, multiple calibrated RGB views are jointly processed by the frozen backbone, whereas inference requires only a single RGB image. All experiments are conducted on the same single data-center GPU system with approximately 140 GB of high-bandwidth memory under identical hardware setting. Table 6 lists the complete configuration of the deployed model.

A.2 Limitation and Future Work

Although PhysVGGT demonstrates strong performance in estimating physical properties from a single image, an inherent limitation is that it relies solely on visual information, which can be misleading in cases where the appearance does not accurately reflect the underlying physical properties. Incorporating additional sensory modalities could complement visual cues, potentially improving the robustness and accuracy of physical property estimation. Future work will focus on addressing this limitation and integrating PhysVGGT into robotic systems to evaluate its impact on robotic perception and manipulation.

Table 6: Hyperparameters of the deployed PhysVGGT. The frozen StreamVGGT backbone (1.26 B parameters) receives no gradient; all listed settings apply to the 32.4 M-parameter physics heads.

Architecture	
backbone (frozen)	StreamVGGT, 1.26 B params
backbone token dim C	2048
DPT feature width	128
UNet head width h	64 $(h, 2h, 4h, 8h$ stages)
dense properties	4 (μ, H, E, ρ)
coupling	attention, 2 layers
coupling dim / heads	128 / 4
decoder	shared trunk + per-property heads
decoder dropout	0.1
trainable parameters	32.4 M
Optimization	
optimizer	AdamW, $\beta = (0.9, 0.95)$
learning rate	4×10^{-5}, constant
weight decay	1×10^{-4}
warm-start LR multiplier	0.1
batch size	4 per GPU $\times$ 4 GPUs
epochs	8
gradient clipping	1.0
seed	42
input mode / resolution	centre crop, 518
Losses	
reconstruction	Charbonnier, foreground-masked
λ_μ / λ_H / λ_E / λ_ρ	3 / 1 / 1 / 1
λ_{rank}, τ	0.25, 0.1
λ_{mass}	1.0 (log-space)

A.3 Architectural Design

PhysVGGT attaches lightweight trainable heads to the frozen geometry encoder backbone. The trainable modules include the DPT feature extractor, UNet encoder, CPC module, shared decoder, and mass head, which have 22.8M, 5.1M, 0.9M, 3.0M, and 0.55M parameters, respectively. In total, these trainable modules account for approximately 32.4M parameters. The detailed architectural design of cross-property coupling (CPC) module, shared decoder, and property specific head is illustrated in Figure 6.

Cross-Property Coupling. Physical properties are correlated, yet independent output heads cannot explicitly exchange information. CPC introduces this interaction at the encoder bottleneck, where features are most semantic and spatial resolution is lowest. The $8h = 512$-channel bottleneck is projected into four 128-D property tokens at each spatial location. Learned property embeddings identify the tokens, and two pre-norm transformer blocks (4 attention heads, MLP ratio 4) model cross-property dependencies. Property-specific projections then map the tokens back to 512 channels and are added residually to the bottleneck. Attention operates across the four *properties*, rather than spatial locations, making its cost independent of image resolution. CPC therefore adds only 0.9M parameters (2.8% of the head). We zero-initialize the output projections, making CPC an identity mapping at initialization and enabling stable initialization from an uncoupled checkpoint.

Shared Decoder. Frozen backbone tokens are converted to dense features by a DPT head and passed through a four-stage UNet encoder with widths $(h, 2h, 4h, 8h)$, where $h = 64$. Rather than using an independent decoder for each property, we employ a single shared upsampling trunk with encoder skip connections, followed by lightweight property-specific 3×3 output heads. This design exploits the common spatial structure of physical properties, which typically share material boundaries while differing in their values. Bilinear interpolation before skip fusion makes the decoder resolution-agnostic. Sharing improves both efficiency and accuracy: compared with four indepen-

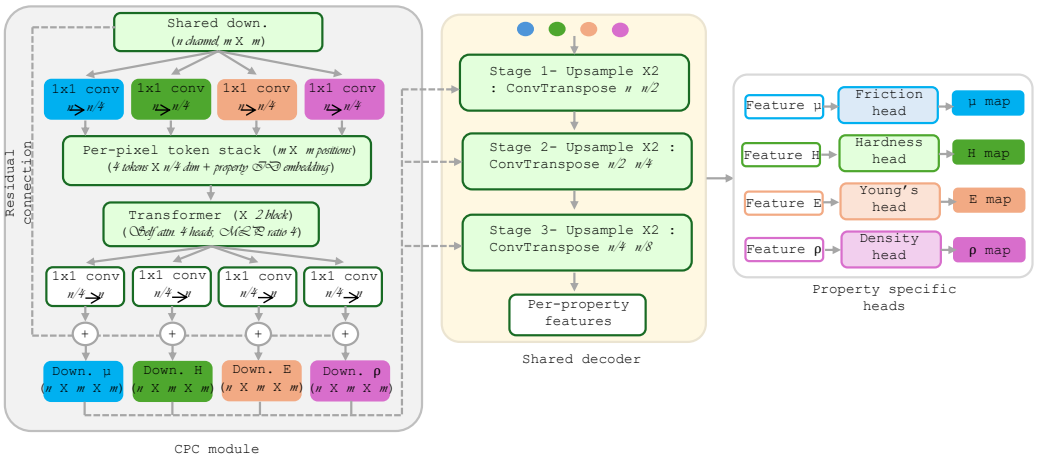


Figure 6: Detailed architectural design of PhysVGGT's cross-property coupling (CPC) module, shared decoder, and property specific head.

Table 7: Spatial dense trunk vs. pooled-token regression. The per-pixel trunk is essential for the spatial properties.

	Friction μ		Shore hardness	
Head	MnRE↑	ADE↓	MnRE↑	ADE↓
Pooled-token	0.742	0.148	0.844	12.52
Dense trunk (ours)	**0.831**	**0.102**	**0.908**	**7.05**

dent decoders, it reduces the head from 40.9M to 31.9M parameters while improving friction MnRE from 0.808 to 0.831 and hardness ADE from 8.62 to 7.05. Thus, sharing provides a useful inductive bias in addition to parameter savings.

Physics Property and Mass Heads. The shared decoder predicts four dense properties: friction, Shore hardness, Young's modulus, and density. Friction is predicted directly, hardness is normalized to $[0, 1]$, and Young's modulus and density are predicted in $\log_{10}$ space to accommodate their large dynamic ranges. We optimize these maps using a foreground-masked Charbonnier reconstruction loss together with a pairwise ranking loss. Mass is treated separately because it is an object-level rather than per-pixel quantity. Instead of deriving mass from the dense bottleneck, the mass branch operates directly on pooled frozen backbone tokens, which empirically improves MnRE from 0.55 to 0.61. Specifically, we average the patch tokens from the last aggregator layer and apply an MLP $2048 \to 256 \to 64 \to 1$ with BatchNorm, GELU, and dropout, predicting $\log m$. This branch runs in parallel with the dense pathway, so mass prediction does not alter the dense features. BatchNorm is preferred over LayerNorm because it preserves sample-level magnitude information useful for estimating object scale, improving MnRE from 0.547 to 0.605. Finally, predicting mass in log space better accommodates its large dynamic range and aligns training with the ratio-based evaluation metrics.

A.4 Ablation and Sensitivity Study

Dense Prediction vs. Pooled Tokens. Table 7 compares direct regression from pooled backbone tokens against the proposed shared dense trunk. The dense design improves friction MnRE by 12.0% (ADE −31.1%) and hardness MnRE by 7.6% (ADE −43.6%): friction and hardness are spatial properties that depend on local material variation and benefit from pixel-wise prediction. This motivates a shared dense trunk for the four properties, with object-level mass predicted separately from globally pooled geometry features.

Table 8: Choice of geometry backbone under identical head and training protocol.

	Friction μ		Shore hardness	
Backbone	MnRE↑	ALDE↓	MnRE↑	ALDE↓
StreamVGGT (ours)	**0.831**	**0.206**	**0.908**	**0.103**
VGGT-1B (Wang et al., 2025)	0.819	0.292	0.875	0.159
DINOv2-L (Oquab et al., 2024)	0.802	0.252	0.834	0.173

Table 9: Backbone freezing strategy. Freezing the entire StreamVGGT backbone vs. fine-tuning its global-attention blocks.

StreamVGGT freezing	Train. params	Friction MnRE↑	Hardness MnRE↑	Mass MnRE↑
Fully frozen (ours)	**32 M**	**0.831**	**0.908**	**0.655**
Unfreeze global-attn	334 M	0.695	0.814	0.517

Table 10: Decoder ablation. Decoder architectures for multi-task physical property prediction.

Decoder	Dec. params	Total train.	Fric. MnRE↑	Hard. ADE↓	Hard. MnRE↑
original (4 decoders)	12.1 M	40.9 M	0.808	8.62	0.887
slim (4 half-width)	4.3 M	33.2 M	0.808	7.48	0.899
shared + heads (ours)	**3.0 M**	**31.9 M**	**0.831**	**7.05**	**0.908**

Table 11: Loss-component ablation.

Loss	Friction MnRE↑	Hardness MnRE↑	Mass ADE↓
L1 baseline	0.684	0.826	12.39
+ foreground mask	0.759	0.856	10.72
+ pairwise ranking	0.811	0.851	11.49
+ all (mask, Charb., rank)	**0.831**	**0.908**	**4.73**

Choice of Geometry Backbone. Table 8 compares different frozen backbone representations while keeping the prediction head and training protocol unchanged. StreamVGGT and VGGT perform comparably, however, the streaming formulation preserves the geometry representation needed for single-image estimation. DINOv2 trails on both friction and hardness, while StreamVGGT is best on both — the geometry-aware VGGT-family features benefit both properties.

Backbone Freezing Strategy. Table 9 evaluates whether fine-tuning the backbone helps. Unfreezing the global-attention blocks raises the trainable parameter count from 32 M to 334 M yet degrades every task, and the fully frozen model improves friction, hardness and mass MnRE by 19.6%, 11.5% and 26.7% respectively. Fine-tuning under weak pseudo supervision likely overfits the pretrained representation, so optimizing only the lightweight physics head gives both higher accuracy and far better parameter efficiency.

Shared Decoder. Table 10 compares decoder designs. Replacing four independent decoders with a single shared decoder plus lightweight property-specific heads cuts decoder parameters by 75.2% and total trainable parameters by 22.0%, yet still improves friction MnRE by 2.8% and reduces hardness ADE by 18.2%. The four properties evidently share a common spatial representation, so parameter sharing aids both efficiency and generalization.

Loss Components. We ablate foreground masking, Charbonnier reconstruction, and pairwise ranking losses under controlled 3-epoch runs (Table 11). Ranking primarily improves friction, increasing MnRE by 18.6%, while foreground masking improves hardness by 3.6%. Combining

Table 12: Data scaling. Mass prediction on ABO-500 and OOD friction/hardness on N2P. Performance improves consistently with training-set size.

	Mass (ABO-500)			OOD (N2P)	
Train. obj.	ADE (kg) ↓	ALDE ↓	MnRE ↑	μ MnRE ↑	H MnRE ↑
500	5.62	0.633	0.651	0.787	0.824
2000	5.26	0.658	0.641	0.794	0.860
7225	**4.73**	**0.574**	**0.655**	**0.831**	**0.908**

Table 13: Mass branch: dense ($\rho \times V$) versus global (direct head) on ABO-500. Both geometry tokens come from the same frozen geometry backbone and the same trained model; only the path from features to mass differs.

Mass branch	ADE (kg) ↓	ALDE ↓	APE ↓	MnRE ↑
dense, $\bar{\rho} \times V_{\text{shell}} \times \kappa$	10.3	0.896	1.043	0.517
global, direct mass head	**4.73**	**0.574**	**0.787**	**0.655**

Table 14: Refined vs. raw supervision comparison.

Supervision	Friction	Hardness	Mass (dense)
Raw	0.813	0.856	0.450
Refined	**0.831**	**0.908**	**0.655**

all components achieves the best overall performance, improving friction and hardness MnRE by 21.5% and 9.9% and reducing mass ADE by 61.8% over the L1 baseline, and is used by default.

Data Scaling. We study the effect of training-set size using 500, 2000, and 7225 objects (Table 12). Going from 500 to 7225 training objects lowers mass ADE on ABO-500 by 15.8%, although mass MnRE is essentially flat ($+0.6\%$) and dips at 2000 objects. The out-of-distribution N2P properties benefit more clearly and monotonically: friction MnRE improves by 5.6% and hardness MnRE by 10.2%.

Dense vs. Global Mass Prediction. We compare two strategies for mass prediction: deriving mass from dense density and volume following the existing SOTA methods $m = \bar{\rho} V \kappa$, and directly regressing $\log m$ from pooled backbone tokens. As shown in Table 13, direct regression performs substantially better, improving MnRE from 0.517 to 0.655 and reducing ADE from 10.3 to 4.73 kg. The dense route is limited by coarse material-density labels, volume-estimation errors, and error propagation through ρV. In contrast, the global head directly optimizes the target quantity and leverages all 7326 ABO product weights without requiring volume annotations. We therefore use the global branch for mass prediction and retain dense heads for spatial physical properties.

Refine vs. Raw Pseudo Label Supervision. Table 14 compares training on the raw per-view pseudo labels with training on the cross-view refined labels. Refinement improves every property: friction MnRE rises from 0.813 to 0.831 ($+2.2\%$), hardness from 0.856 to 0.908 ($+6.1\%$), and dense mass from 0.450 to 0.655 ($+45.6\%$). The effect is smallest on friction and by far the largest on dense mass, which aggregates predictions over the whole object and is therefore the most exposed to a part being assigned different materials in different views. This confirms the value of the cross-view refinement stage in the pseudo-label pipeline.

A.5 Hyperparameter Sensitivity

We assess sensitivity to head width, friction-loss weight, learning rate, and ranking-loss weight by varying one hyperparameter at a time from the final configuration ($\star$). All runs use the same 2000-object subset for 2 epochs; we additionally evaluate 4 random seeds to estimate training variance. Thus, Table 15 is intended to compare relative trends rather than absolute performance.

Table 15: Hyperparameter sensitivity. N2P MnRE (↑) under one-at-a-time sweeps from the final configuration (⋆).

Hyperparameter	Setting	μ MnRE↑	H MnRE↑
head width h	32	0.792	0.855
	48	**0.839**	**0.877**
	64 ⋆	0.829	**0.877**
	96	0.822	0.859
$\lambda_{\text{friction}}$	1	0.807	0.864
	2	0.770	0.868
	3 ⋆	0.829	**0.877**
	5	0.799	0.859
learning rate	2×10^{-5}	0.837	0.864
	4×10^{-5} ⋆	0.829	**0.877**
	8×10^{-5}	0.770	0.846
	1.6×10^{-4}	*diverged (NaN)*	
λ_{rank}	0	0.741	0.845
	0.1	0.781	0.863
	0.25 ⋆	0.829	**0.877**
	0.5	**0.878**	0.865
seed	42 ⋆	0.829	0.877
	1	0.847	0.865
	2	0.798	0.854
	3	0.767	0.853

Table 16: Zero-shot mass prediction on image2mass. All baselines are trained on image2mass (Standley et al., 2017), whereas PhysVGGT is trained only on ABO and evaluated without fine-tuning. †: human estimates provided with the dataset.

Method	Zero-shot	ALDE↓	MnRE↑
Xception k-NN	No	1.454	0.341
Xception k-NN (SIM)	No	0.781	0.534
Pure CNN	No	0.680	0.575
CNN with ALDE	No	0.666	0.573
No Geometry	No	0.520	0.638
Shape-aware	No	0.465	0.675
Shape-aware, multi-view	No	**0.437**	**0.693**
PhysVGGT (ours)	Yes	0.585	0.653
Human median†	—	0.459	0.675
Human ensemble†	—	0.323	0.754

Table 15 shows that PhysVGGT is generally robust to hyperparameter choices, with most variations comparable to the observed seed variance. The ranking loss has the clearest effect: removing it decreases friction MnRE from 0.829 to 0.741 (-10.6%), while increasing its weight to 0.5 reaches 0.878. Head capacity largely saturates at $h = 48$–64, and moderate changes to $\lambda_{\text{friction}}$ have limited impact. The learning rate remains stable up to 4×10^{-5}, degrades at 8×10^{-5}, and diverges at 1.6×10^{-4}. We retain $\lambda_{\text{rank}} = 0.25$ despite the higher friction score at 0.5 because the final configuration was fixed before this sweep, avoiding tuning on the evaluation benchmark.

A.6 Further Analysis and Discussion

PhysVGGT Supports Grounded, Interpretable Physical Reasoning. Figure 7 demonstrates a language-guided interface built on PhysVGGT predictions. Given a single RGB image, PhysVGGT predicts dense physical properties and object-level mass, while SAM2 defines spatial regions. The original image, segmented image, and region-level predictions are provided to Qwen3-VL for multi-round question answering and physical-composition captioning. Thus, visual semantics and lan-



Figure 7: Language-guided physical reasoning. *Top:* multi-round question answering about object- and region-level physical properties. *Bottom:* single-shot physical-composition captioning. PhysVGGT supplies dense friction, Shore hardness, Young's modulus, density, and object-level mass predictions; SAM2 supplies spatial regions; and Qwen3-VL interprets the original and segmented images and generates the responses. Material names are estimated by matching each predicted property tuple (μ, H, E, ρ) to a reference material database. All numerical physical claims are derived from PhysVGGT predictions.

guage generation come from the VLM, whereas all reported physical quantities are grounded in PhysVGGT predictions.

Domain Generalization Capability. We further evaluate the ABO-trained mass head on the real-world image2mass household benchmark (Standley et al., 2017) without fine-tuning. This represents a substantial domain shift from synthetic ABO renders to real images of everyday household objects. As shown in Table 16, PhysVGGT achieves 0.653 MnRE, outperforming five of seven baselines despite all baselines being trained in-domain on approximately 150k image2mass products. It trails only the shape-aware models (0.675/0.693), which additionally use explicit object dimensions, and is within 0.022 MnRE of the median human estimate (0.675).

Downstream Application For an anti-slip two-finger pinch, the Coulomb condition $2\mu F_n \geq mg$ gives the minimum normal force $F_n = mg/(2\mu)$. Since PhysVGGT predicts per-pixel friction μ and object-level mass m from a single image, we can directly derive a dense grip-force map without depth sensing, force measurements, or physical interaction. As shown in Figure 8, the resulting maps consistently favor high-friction surfaces while assigning substantially larger forces to slippery regions. Thus, the predicted physical properties provide a spatially grounded signal for both *where* to grasp and *how hard* to squeeze, demonstrating their direct utility for downstream robotic manipulation.

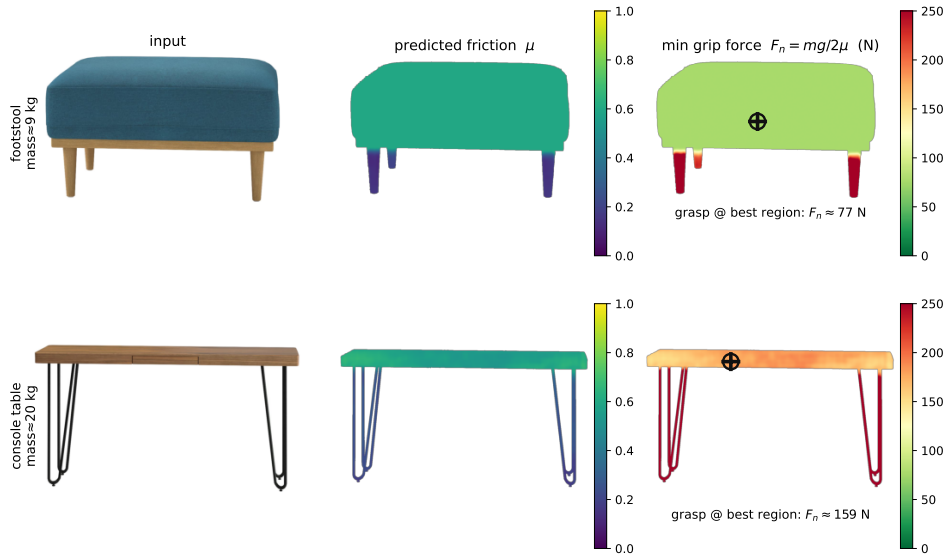


Figure 8: Grasp-force planning from predicted physics. From one image PhysVGGT predicts per-pixel friction μ (middle) and object mass, yielding a dense minimum grip-force map $F_n = mg/(2\mu)$ (right; green=secure/low force, red=slippery/high force) and a best contact region (marker). On both objects, the model marks the grippy top surface as the grasp region while the slick legs (wooden; thin metal) turn red (> 250 N).