
SupPhysField: Fast and Generalizable Supervised Learning of 3D Physics from Visual Features

Anonymous Author(s)

Affiliation

Address

email

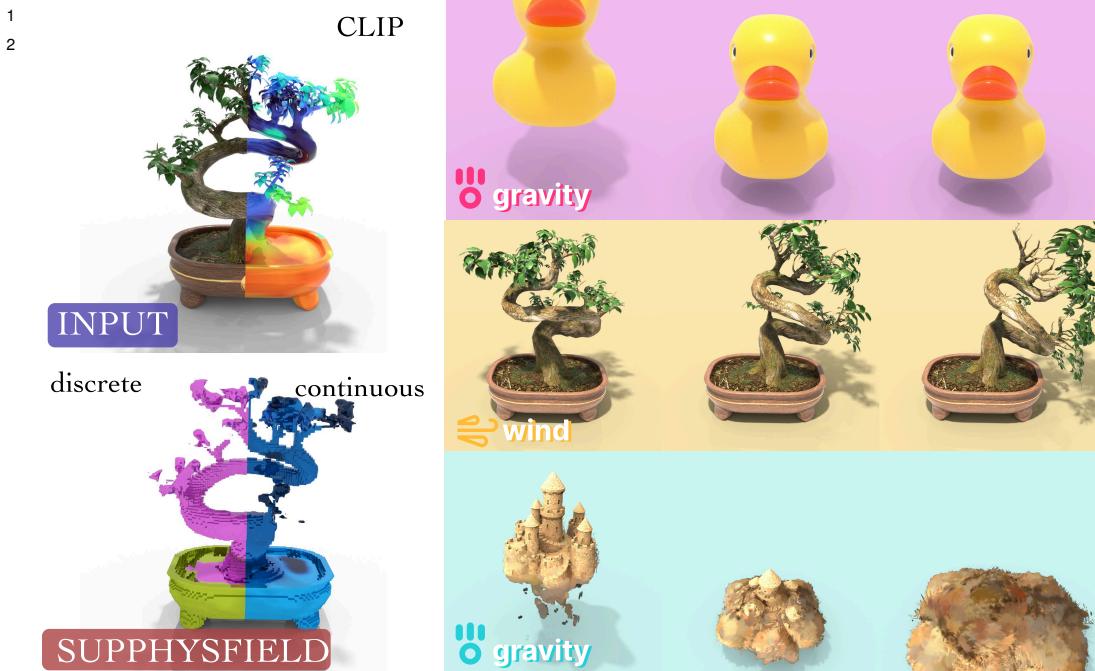


Figure 1: We introduce SUPPHYSFIELD, a novel method for learning simulatable physics of 3D scenes from visual features. Trained on a curated dataset of paired 3D objects and physical material annotations, SUPPHYSFIELD can predict both the discrete material types (e.g., rubber) and continuous values including Young’s modulus, Poisson’s ratio, and density for a variety of materials, including elastic, plastic, and granular. The predicted material parameters can then be coupled with a learned static 3D model such as Gaussian splats and a physics solver such as the Material Point Method (MPM) to produce realistic 3D simulation under physical forces such as gravity and wind.

Abstract

3 Inferring the physical properties of 3D scenes from visual information is a critical
4 yet challenging task for creating interactive and realistic virtual worlds. While
5 humans intuitively grasp material characteristics such as elasticity or stiffness,
6 existing methods often rely on slow, per-scene optimization, limiting their gen-
7 eralizability and application. To address this problem, we introduce SUPPHYS-
8 FIELD, a novel method that trains a generalizable neural network to predict phys-
9 ical properties across multiple scenes from 3D visual features purely using su-
10 pervised losses. Once trained, our feed-forward network can perform fast in-

11 ference of plausible material fields, which coupled with a learned static scene
12 representation like Gaussian Splatting enables realistic physics simulation under
13 external forces. To facilitate this research, we also collected SUPPHYSVERSE,
14 one of the largest known datasets of paired 3D assets and physic material anno-
15 tations. Extensive evaluations demonstrate that SUPPHYSFIELD is about 2.21-
16 4.58x better and orders of magnitude faster than test-time optimization methods.
17 By leveraging pretrained visual features like CLIP, our method can also zero-shot
18 generalize to real-world scenes despite only ever been trained on synthetic data.
19 <https://neurips-2025-20627.github.io/>

20 1 Introduction

21 Advances in learning-based scene reconstruction with Neural Radiance Fields [23] and Gaussian
22 Splatting [15] have made it possible to recreate photorealistic 3D geometry and appearance from
23 sparse camera views, with broad applications from immersive content creation to robotics and simu-
24 lation. However, these approaches focus exclusively on visual appearance—capturing the geometry
25 and colors of a scene while remaining blind to its underlying physical properties.

26 Yet the world is not merely a static collection of shapes and textures. Objects bend, fold, bounce,
27 and deform according to their material composition and the forces acting upon them. Consequently,
28 there has been a growing body of work that aims to integrate physics into 3D scene modeling
29 [25, 22, 19, 10, 9, 34, 26, 11, 21, 35, 5]. Current approaches for acquiring the material proper-
30 ties of the scene generally fall into two categories, each with significant limitations. Some works
31 such as [34, 11] require users to manually specify material parameters for the entire scene based
32 on domain knowledge. This manual approach is limited in its application as it places a heavy bur-
33 den on the user and lacks fine-grained detail. Another line of work aims to automate the material
34 discovery process via test-time optimization. Works including [14, 19, 37, 13, 21, 36] leverage dif-
35 ferentiable physics solvers, iteratively optimizing material fields by comparing simulated outcomes
36 against ground-truth observations or realism scores from video generative models. However, pre-
37 dicting physical parameters for hundreds of thousands of particles from sparse signals (i.e., a single
38 rendering or distillation scalar loss) is an extremely slow and difficult optimization process, often
39 taking hours on a single scene. Furthermore, this heavy per-scene memorization does not generalize:
40 for each new scene, the incredibly slow optimization has to be run from scratch again.

41 In this paper, we propose a new framework, SUPPHYSFIELD, which unifies geometry, appearance,
42 and physics learning via direct supervised learning. Our approach is inspired by how humans intui-
43 tively understand physics: when we see a tree swaying in the wind, we do not memorize the
44 stiffness values for each specific coordinate (x, y, z) – instead, we learn that objects with tree-like
45 visual features behave in certain ways when forces are applied. This physical understanding from
46 visual cues allows us to anticipate the motion of a different tree or even other vegetation like grass,
47 in an entirely new context. Thus, our insight is to leverage rich 3D visual features such as those
48 distilled from CLIP [27] to predict physical materials in a direct supervised and feed-forward way.
49 Once trained, our model can associate visual patterns (e.g., "if it looks like vegetation") with phys-
50 ical behaviors (e.g., "it should have material properties similar to a tree"), enabling fast inference
51 and generalization across scenes. To facilitate this research, we have curated and labeled SUPPHYS-
52 VERSE, a dataset of 1624 paired 3D objects and annotated materials spanning 10 semantic classes.
53 To our knowledge, this is the largest open-source dataset of paired 3D assets and physical material
54 labels. Trained on SUPPHYSVERSE, our feed-forward network can predict material fields that are
55 2.21-4.58x better and orders of magnitude faster than test-time optimization methods. By leverag-
56 ing pretrained visual features, SUPPHYSFIELD can also zero-shot generalize to real-world scenes
57 despite only ever being trained on synthetic data.

58 Our contributions include:

- 59 1. **Novel Framework for 3D Physics Prediction:** We introduce SUPPHYSFIELD, a unified frame-
60 work that predicts discrete material types and continuous physical parameters (Youngs modulus,
61 Poissons ratio, density) directly from visual features using supervised learning.
- 62 2. **SUPPHYSVERSE Dataset:** We curate and release SUPPHYSVERSE, the largest open-source
63 dataset of 3D objects with physical material annotations (1624 objects, 10 semantic classes).

- 64 3. **Fast and Generalizable Inference:** By leveraging pretrained visual features from CLIP and
 65 a feed-forward 3D U-Net, SUPPHYSFIELD performs inference orders of magnitude faster than
 66 prior test-time optimization approaches, achieving a 2.21-4.58x improvement in realism scores
 67 as evaluated by a state-of-the-art vision-language model.
- 68 4. **Zero-Shot Generalization to Real Scenes:** Despite being trained solely on synthetic data, SUP-
 69 PHYSFIELD generalizes to real-world scenes, showing how visual feature distillation can effec-
 70 tively bridge the sim-to-real gap.
- 71 5. **Seamless Integration with MPM Solvers:** The predicted material fields can be directly coupled
 72 with Gaussian splatting models for realistic physics simulations under applied forces such as
 73 wind and gravity, enabling interactive and visually plausible 3D scene animations.

74 2 Related Work

75 **2D World Models** Some early works [3, 2] learn to predict material labels on 2D images. Recently,
 76 learning forward dynamics from 2D video frames has also been explored extensively. For instance,
 77 Google’s Genie [24] trains a next-frame prediction model conditioned on latent actions derived from
 78 user inputs, capturing intuitive 2D physics in an unsupervised manner. While these methods achieve
 79 impressive 2D generation and control, they do not explicitly model 3D geometry or a physically
 80 grounded world. Other works such as [6, 20] also explore generating or editing images based on
 81 learned real-world dynamics. While these methods achieve impressive results in 2D visual synthe-
 82 sis and can imply motion dynamics, they typically do not explicitly model 3D geometry, and only
 83 encode physics implicitly via next-frame prediction rather than through explicit material parameters,
 84 nor do they infer physically grounded material properties decoupled from appearances. These can
 85 lead to problems such as a lack of object permanence or implausible interactions. In contrast, SUP-
 86 PHYSFIELD directly operates in 3D, predicting explicit physical parameters (e.g., Young’s modulus,
 87 density) for 3D objects, enabling their integration into 3D physics simulators or neural networks [31]
 88 for realistic interaction.

89 **Manual Assignment or Assignment of Physics using LLMs** A number of recent methods
 90 have explored combining learned 3D scene representations (e.g., Gaussian splatting) with a physics
 91 solver where material parameters are assigned manually or through high-level heuristics. This often
 92 involves users specifying material types for the scene [34, 1] or using scripted object-to-material
 93 dictionaries [26] or large language and vision-language models [12, 4, 35, 18, 33] to guide the
 94 assignment.

95 **Test-time material optimization using videos** Other works explore more automatic and prin-
 96 cipled ways to infer material properties using rendered videos. Some techniques [14, 19, 37] optimize
 97 material parameters by comparing simulated deformations against ground-truth observations, often
 98 requiring ground-truth multi-view videos of objects or ground-truth particle positions under known
 99 forces. More recent approaches [13, 21, 36] use video diffusion models as priors to optimize physics
 100 via a motion distillation loss. Notably, these approaches suffer from extremely slow per-scene opti-
 101 mization, often taking hours on a single scene, and do not generalize to new scenes. In stark contrast,
 102 SUPPHYSFIELD employs a feed-forward neural network that, once trained, predicts physical param-
 103 eters in seconds, and can generalize to unseen scenes. A recent work Vid2Sim [5] also aims to learn
 104 a generalizable material prediction network across scenes. This was done by encoding a front-view
 105 video of the object in motion with a foundation video transformer [30] and learning to regress these
 106 motion priors into physical parameters. Unlike Vid2Sim, SUPPHYSFIELD does not require videos,
 107 relying instead on visual features from static images.

108 3 Method

109 Our central thesis is that 3D visual appearance provides sufficient information to recover an object’s
 110 physical parameters. Texture, shading, and shape features captured from multiple calibrated images
 111 correlate with physical quantities such as Young’s modulus and Poisson’s ratio. By learning a map-
 112 ping from these visual features to material properties, we can augment a volumetric reconstruction
 113 model (e.g., Gaussian splatting) with a point-wise material estimate, without requiring force re-
 114 sponse observations. In Sec. 3.1, we detail our framework, leveraging rich visual priors from CLIP
 115 to predict a material field, which can be used by a physics solver to animate objects responding to

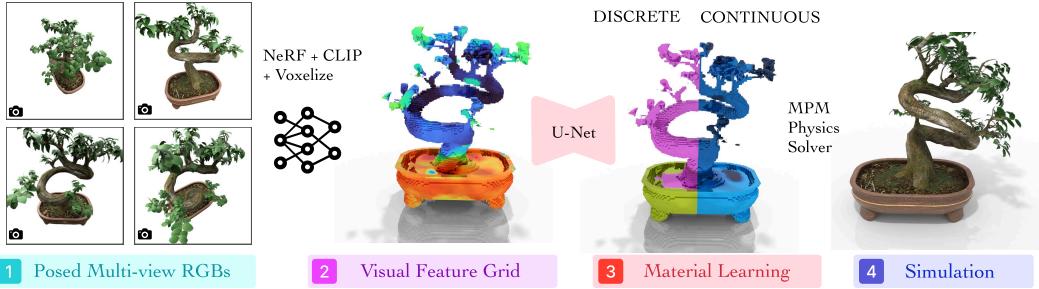


Figure 2: Method Overview. From posed multi-view RGB images of a static scene, SUPPHYSFIELD first reconstructs a 3D model with NeRF and distilled CLIP features [28]. Then, we voxelize the features into a regular $N \times N \times N \times D$ grid where N is the grid size and D is the CLIP feature dimension. A U-Net neural network [8] is trained to map the feature grid to the material field $\hat{\mathcal{M}}_G$ which consists of a discrete material model ID and continuous Young’s modulus, Poisson’s ratio, and density value for each voxel. Coupled with a separately trained Gaussian splatting model, $\hat{\mathcal{M}}_G$ can be used to simulate physics with a physics solver such as MPM.

116 external forces. To train this model, we curated SUPPHYSVERSE, a large dataset of paired 3D assets
 117 and material annotations, as detailed in Sec. 3.2. Figure 2 gives an overview of our method.

118 3.1 SUPPHYSFIELD Physics Learning

119 **Problem Formulation** Formally, the goal is to learn a mapping:

$$f_\theta : (\mathcal{I}, \Pi) \longrightarrow \hat{\mathcal{M}} \quad (1)$$

120 that turns some calibrated RGB images of the static scene $\mathcal{I} = \{I_k\}_{k=1}^K$ and their joint camera
 121 specification Π into a continuous three-dimensional *material field*. For every point $\mathbf{p} \in \mathbb{R}^3$ within
 122 the scene bounds, the field returns

$$\hat{\mathcal{M}}(\mathbf{p}) = (\hat{\ell}(\mathbf{p}), \hat{E}(\mathbf{p}), \hat{\nu}(\mathbf{p}), \hat{d}(\mathbf{p})) ,$$

123 where $\hat{\ell} : \mathbb{R}^3 \rightarrow \{1, \dots, L\}$ is the discrete material class and $\hat{E}, \hat{\nu}, \hat{d} : \mathbb{R}^3 \rightarrow \mathbb{R}$ are the continuous
 124 Young’s modulus, Poisson’s ratio, and density value respectively. Recall that the discrete material
 125 class, also known as the constitutive law, in Material Point Method is a combination of the choices of
 126 an expert-defined hyperelastic energy function \mathcal{E} and return mapping \mathcal{P} (Sec. A.1). Learning a point-
 127 mapping like this provides a fine-grained material segmentation where for every spatial location we
 128 assign both a semantic material label and the physical parameters that characterise that material.
 129 Learning the mapping in Eqn. (1) directly from 2D images to 3D materials is clearly not simple
 130 neither sample efficient. Instead, we leverage a distilled feature field which has rich visual priors to
 131 represent the intermediate mapping between 2D images and 3D visual features, and then a separate
 132 U-Net architecture to compute the mapping between 3D visual features and physical materials. We
 133 describe these components below.

134 **3D Visual Feature Distillation** Recent work on distilled feature fields has shown that dense
 135 2D visual feature embeddings extracted from foundation models, such as CLIP, based on images
 136 can be lifted into 3D, yielding a volumetric representation that is both geometrically accurate and
 137 rich in terms of visual and semantic priors [28]. These works have used distilled features to better
 138 understand 3D scenes for robotics manipulation tasks. To our knowledge, this idea has not been
 139 applied to material prediction, despite the promise in using semantically rich 3D feature volumes
 140 to encode cues about an objects composition and stiffness. Here we augment the classical NeRF
 141 representation [23] to predict a view-independent feature vector in addition to color and density, i.e.,

$$F_\theta : (\mathbf{x}, \mathbf{d}) \longmapsto (f(\mathbf{x}), c(\mathbf{x}, \mathbf{d}), \sigma(\mathbf{x})) ,$$

142 where $c \in \mathbb{R}^3$, and $\sigma \in \mathbb{R}_{\geq 0}$ are the standard color and radiance from NeRF and the extra output
 143 $f \in \mathbb{R}^d$ is a high-dimensional descriptor capturing visual semantics (e.g., object identity or other
 144 attributes), which we assume to be view-independent. We can render both the color and feature
 145 channels into any camera view via the standard volume rendering procedure. Concretely, for a
 146 camera ray $r(t) = \mathbf{o} + t\mathbf{d}$ passing through a pixel p , the accumulated color $C(p)$ and feature vector
 147 $F(p)$ are given by integrals along the ray:

$$C(p) = \int_{t_n}^{t_f} T(t, \sigma(r(t)), c(r(t), \mathbf{d})) dt \quad F(p) = \int_{t_n}^{t_f} T(t, \sigma(r(t)), f(r(t))) dt , \quad (2)$$

148 where $T(t) = \exp\left(-\int_{t_n}^t \sigma(r(s)) ds\right)$ is the accumulated transmittance from the ray origin to depth
 149 t . At each training iteration, a batch of rays is sampled from the input views. For each ray r (pixel
 150 p), we enforce that the rendered color $C(p)$ matches the ground-truth pixel RGB $C^*(p)$, while the
 151 rendered feature $F(p)$ matches the corresponding CLIP-based feature vector $F^*(p)$ extracted from
 152 the image. The loss of the network is:

$$\mathcal{L} = \sum_p \|C(p) - C^*(p)\|_2^2 + \lambda_{\text{feat}} \sum_p \|F(p) - F^*(p)\|_2^2 ;$$

153 the first term enforces color fidelity, while the second aligns the rendered volumetric CLIP features
 154 with the dense 2D features extracted from the training images.

155 From a trained distilled feature field F_θ , we obtain a regular feature grid F_G of dimension $N \times N \times$
 156 $N \times D$ grid, where $N = 64$ is the grid size and $D = 768$ is the CLIP feature dimension. This is
 157 done via voxelization using known scene bounds. For our synthetic dataset, we center and normalize
 158 all objects within a unit cube.

159 **Material Grid Learning** Our material learning network f_M consists of a feature projector f_P
 160 and a U-Net f_U . As the CLIP features are very high-dimensional which can cause memory issues
 161 on GPUs, we learn a feature projector network f_P , which consists of three layers of 3D convolution
 162 mapping CLIP features \mathbb{R}^{768} to a low-dimensional manifold \mathbb{R}^{64} . We then use the U-Net architecture
 163 f_U from OpenAI’s Guided Diffusion codebase [8] with 2D convolution replaced by 3D kernels to
 164 learn the mapping from the projected feature grid F_G to a material grid $\hat{\mathcal{M}}_G(\mathbf{p})$, which is a voxelized
 165 version of the material field $\hat{\mathcal{M}}(\mathbf{p})$. The feature projector f_P and U-Net f_U are jointly trained
 166 end-to-end via a cross entropy and mean-squared error loss to both predict the discrete material
 167 classification and the continuous values including Young’s modulus, Poisson’s ratio and density.

168 We found that our voxel grids are very sparse with around 98% of the voxels being background.
 169 Naively trained, the material network f_M would learn to always predict background. Thus, we
 170 also separately compute an occupancy mask grid $\mathbb{M} \in \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N$, constructed by filtering
 171 out all voxels whose NeRF densities fall below a threshold $\alpha = 0.01$. The supervised losses—
 172 cross entropy and mean squared errors—are only enforced on the occupied voxels. Concretely, the
 173 masked supervised loss consists of a discrete cross entropy and continuous mean-squared error loss:

$$\begin{aligned} \mathcal{L}_{\text{sup}} = \frac{1}{N_{\text{occ}}} \sum_{\mathbf{p} \in \mathcal{G}} \mathbb{M}(\mathbf{p}) & \left[\lambda \cdot \text{CE}(\hat{\ell}(\mathbf{p}), \ell^{GT}(\mathbf{p})) + (\hat{E}(\mathbf{p}) - E^{GT}(\mathbf{p}))^2 \right. \\ & \left. + (\hat{\nu}(\mathbf{p}) - \nu^{GT}(\mathbf{p}))^2 + (\hat{d}(\mathbf{p}) - d^{GT}(\mathbf{p}))^2 \right] , \end{aligned} \quad (3)$$

174 where $N_{\text{occ}} = \sum_{\mathbf{p} \in \mathcal{G}} \mathbb{M}(\mathbf{p})$ is the total number of occupied voxels in the grid, $\hat{\ell}(\mathbf{p})$ and $\ell^{GT}(\mathbf{p})$ are
 175 the predicted material class logits and the ground-truth, CE is the cross entropy loss, λ is a loss bal-
 176 aancing factor, and E, ν, d are the Young’s modulus, Poisson’s ratio and density values, respectively.
 177 The material network f_G is trained on 12 NVIDIA RTX A6000 GPUs, each with a batch size of 4,
 178 in one day using the Adam optimizer [17].

179 **Physics Simulation** We use the Material Point Method (MPM) to simulate physics. The MPM
 180 solver (Sec. A.1.2) takes a point cloud of initial particle poses along with predicted material prop-
 181 erties, and the external force specification, and simulates the particles’ transformations and defor-
 182 mations. Although it is possible to sample particles from a NeRF model (e.g., via Poisson disk
 183 sampling [9]), we have found that it is easier to use a Gaussian Splatting model (Sec. A.1.1) as each
 184 Gaussian can naturally be thought of as a MPM particle [34]. Thus, we separately learn a Gaussian
 185 splatting model from posed multi-view RGB images. We then transfer the material properties from
 186 our predicted material grid into the Gaussian splatting model via nearest neighbor interpolation.

188 3.2 SUPPHYSVERSE Dataset

189 We collect one of the largest and highest quality known datasets of diverse objects with annotated
 190 physical materials. Our dataset (Fig. 3) covers 10 semantic classes, ranging from organic matter
 191 (trees, shrubs, grass, flowers) and granular media (sand, snow and mud) to hollow containers (soda-
 192 cans, metal crates), and toys (rubber ducks, sport balls). The dataset is sourced from Objaverse
 193 [7], the largest open-source dataset of 3D assets. Since Objaverse objects do not have physical
 194 parameter annotations, we develop an automatic multi-stage labeling pipeline leveraging foundation
 195 vision-language models i.e., Gemini-2.5-Pro [29]. More details is given in Appendix A.2.



Figure 3: **SupPhysVerse Dataset Overview.** We collect 1624 high-quality single-object assets, spanning 10 semantic classes (a), and 6 constitutive material types (b). The dataset is annotated with detailed physical properties including spatially varying discrete material types (b), Young’s modulus (c), Poisson’s ratio (d), and mass density (e). The left figure shows representative examples from the dataset: organic matter (*tree, shrubs, grass, flowers*), deformable toys (*rubber ducks*), sports equipment (*sport balls*), granular media (*sand, snow & mud*), and hollow containers (*soda cans, metal crates*).

196 4 Experiments

197 **Dataset** We train SUPPHYSFIELD on a random 90% split of the SUPPHYSVERSE dataset. We
 198 evaluate on 38 synthetic scenes from the test set of SUPPHYSVERSE, and three real-world scene
 199 from the NeRF [23] and LERF [16] datasets.

200 **Simulation Details** We use the material point method (MPM) implementation from PhysGaussian [34] as the physics solver. The solver takes a gaussian splatting model augmented with physics
 201 where each Gaussian particle also has a discrete material model ID, and continuous Young’s modulus,
 202 Poisson’s ratio, and density values. Each simulation is run for around 50 to 125 frames on a
 203 single Nvidia RTX A6000 GPU. External forces such as gravity and wind are applied to the static
 204 scenes as boundary conditions to create physics animations.

205 **Baselines** We evaluate SUPPHYSFIELD against two recent test-time optimization methods:
 206 DreamPhysics [13] and OmniPhysGS [21], and a LLM method – NeRF2Physics [35]. Dream-
 207 Physics optimizes a Young’s modulus field, requiring users to specify other values including ma-
 208 terial ID, Poisson’s ratio, and density. OmniPhysGS, on the other hand, selects a hyperelastic energy
 209 density function and a return mapping model, which, in combination, specifies a material ID for
 210 each point in the field, requiring other physics parameters to be manually specified. Both methods
 211 rely on a user prompt such as “a tree swing in the wind” and a generative video diffusion model to
 212 optimize a motion distillation loss. SUPPHYSFIELD, in contrast, infers all discrete and continuous
 213 parameters jointly (Fig. 5). NeRF2Physics first captions the scene and query a LLM for all plausi-
 214 ble material types (e.g., “metal”) along with the associated continuous values. Then, the material
 215 semantic names are associated with 3D points in the CLIP feature field, and physical properties are
 216 thus assigned via weighted similarities. This method is similar to our dataset labeling in principle
 217 with some notable difference as detailed in Appendix A.2, allowing SUPPHYSVERSE to have much
 218 more high-quality labels. SUPPHYSFIELD thus produces much less noisy predictions (Fig. 6).

219 **Evaluation Metrics** We utilize a state-of-the-art vision-language model, Gemini-2.5-Pro [29],
 220 from Google as a judge. The model is prompted to compare the rendered candidate animations
 221 generated using physics parameters predicted by different baselines, and score those videos on a
 222 scale from 0 to 5, where a higher score is better. We also measure the reconstruction quality using

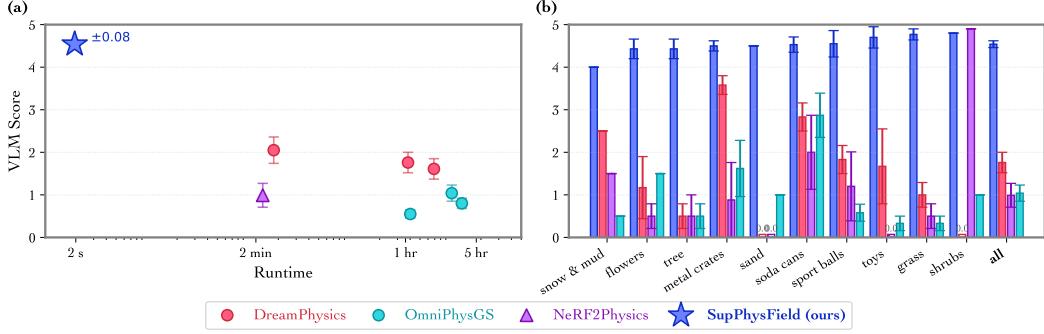


Figure 4: **Main VLM Results.** (a) **VLM score versus wall-clock time:** SUPPHYSFIELD is three orders of magnitude faster than previous works while achieving 2.21-4.58x improvement in realism. Test-time optimization methods are run with varying numbers of epochs i.e., 1, 25, 50 for DreamPhysics and 1, 2, 5 for OmniPhysGS while inference methods are only run once. (b) **Per-class VLM score:** Our method leads on every object class. Standard errors are also included.

Table 1: **Main Quantitative Results.** We report the average reconstruction quality (PSNR, SSIM) against the reference videos in SUPPHYSVERSE, the Gemini VLM scores, and five other metrics our method optimizes including discrete material accuracy and continuous errors over E, ν, ρ . Standard errors are also included, and best values are **bolded**. SUPPHYSFIELD-CLIP is by far the best method across all metrics, achieving 2.21-4.58x improvement in VLM score and 3.6-30.3% gains in PSNR and SSIM. Our CLIP variant is also notably more accurate than RGB and occupancy features as measured by material class accuracy and average continuous MSE on the test set. While our method simultaneously recovers all physical properties, some prior works only predict a subset, hence “-”.

Method	PSNR \uparrow	SSIM \uparrow	VLM \uparrow	Mat. Acc. \uparrow	Avg. Cont. MSE \downarrow	E err \downarrow	ν err \downarrow	ρ err \downarrow
DreamPhysics [13]								
1 epoch	19.398 ± 1.090	0.880 ± 0.020	2.05 ± 0.31	-	-	2.393 ± 0.123	-	-
25 epochs	19.078 ± 0.939	0.881 ± 0.019	1.76 ± 0.24	-	-	1.419 ± 0.097	-	-
50 epochs	19.189 ± 0.980	0.880 ± 0.020	1.61 ± 0.24	-	-	1.387 ± 0.097	-	-
OmniPhysGS [21]								
1 epoch	17.907 ± 0.359	0.882 ± 0.007	0.55 ± 0.10	0.072 ± 0.0511	-	-	-	-
2 epochs	17.889 ± 0.372	0.882 ± 0.007	1.04 ± 0.19	0.109 ± 0.0704	-	-	-	-
5 epochs	17.842 ± 0.354	0.883 ± 0.007	0.80 ± 0.12	0.104 ± 0.0681	-	-	-	-
NeRF2Physics [35]								
SUPPHYSFIELD								
Occupancy	17.887 ± 1.524	0.866 ± 0.027	1.76 ± 0.41	0.686 ± 0.054	0.175 ± 0.021	0.138 ± 0.027	0.177 ± 0.027	0.209 ± 0.032
RGB	18.652 ± 2.031	0.861 ± 0.035	2.53 ± 0.46	0.641 ± 0.066	0.197 ± 0.023	0.144 ± 0.026	0.191 ± 0.028	0.256 ± 0.035
CLIP (ours)	23.256 ± 2.456	0.918 ± 0.023	4.54 ± 0.08	0.809 ± 0.043	0.105 ± 0.013	0.072 ± 0.016	0.118 ± 0.015	0.125 ± 0.020

224 PSNR and SSIM metric against the reference videos in the SUPPHYSVERSE dataset. Other metrics
225 our method optimizes including class accuracy and continuous errors over E, ν, ρ are also computed.

226 4.1 Synthetic Scene Experiments

227 Figure 4 (a) plots Gemini score versus runtime. SUPPHYSFIELD achieves a VLM score of **4.54 ± 0.08** – a **2.21-4.58x** improvement over all baselines – while reducing inference time from minutes
228 or hours to **2 s**. A per-class breakdown in Fig. 4 (b) shows our lead in all classes. In Table 1, our
229 model improves perceptual metrics such as PSNR and SSIM by 3.6 – 30.3% and VLM scores by
230 2.21 – 4.58x over prior works. Figure 5 qualitatively visualizes the physical properties predicted
231 by our network, showing SUPPHYSFIELD’s ability to cleanly and accurately recover discrete and
232 continuous parameters across a diverse sets of objects and continuous value spectrum. Figure 6 vi-
233 sualises four representative scenes, comparing SUPPHYSFIELD against prior works. DreamPhysics
234 leaves stiff artifacts due to missegmentation or overly high predicted E values, OmniPhysGS col-
235 lapses under force, and NeRF2Physics introduces high-frequency noise, whereas SUPPHYSFIELD
236 generates smooth, class-consistent motion and segment boundaries.

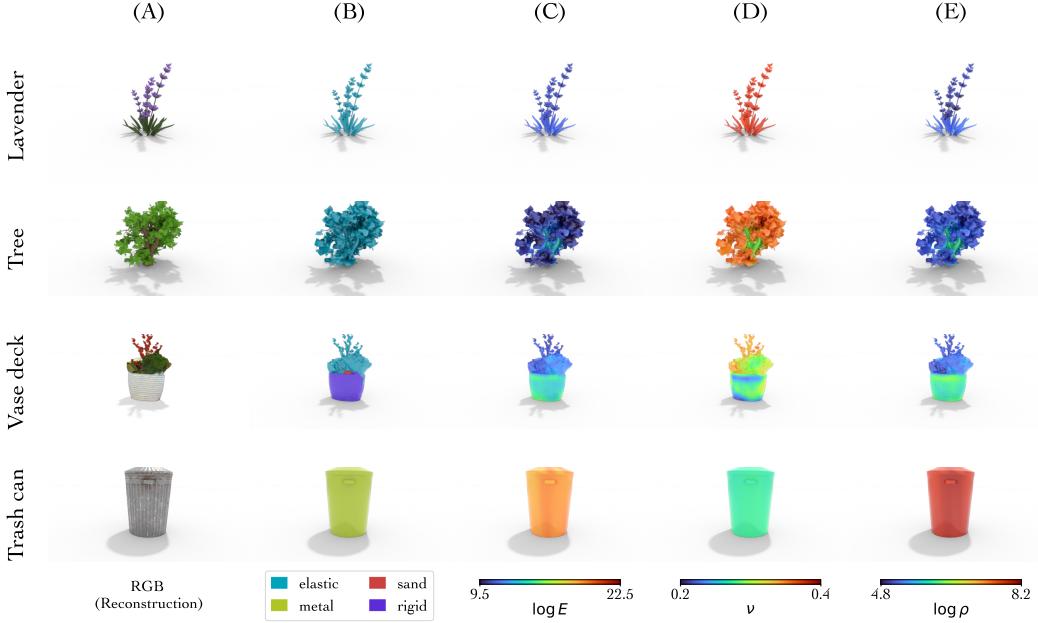


Figure 5: **SUPPHYSFIELD Prediction Visualization.** SUPPHYSFIELD simultaneously recovers discrete material class (B), continuous Young’s modulus (C), Poisson’s ratio (D), and mass density (E) with a high degree of accuracy. For example, the model correctly labels foliage as elastic and the metal can as rigid, while recovering realistic stiffness and density gradients within each object.

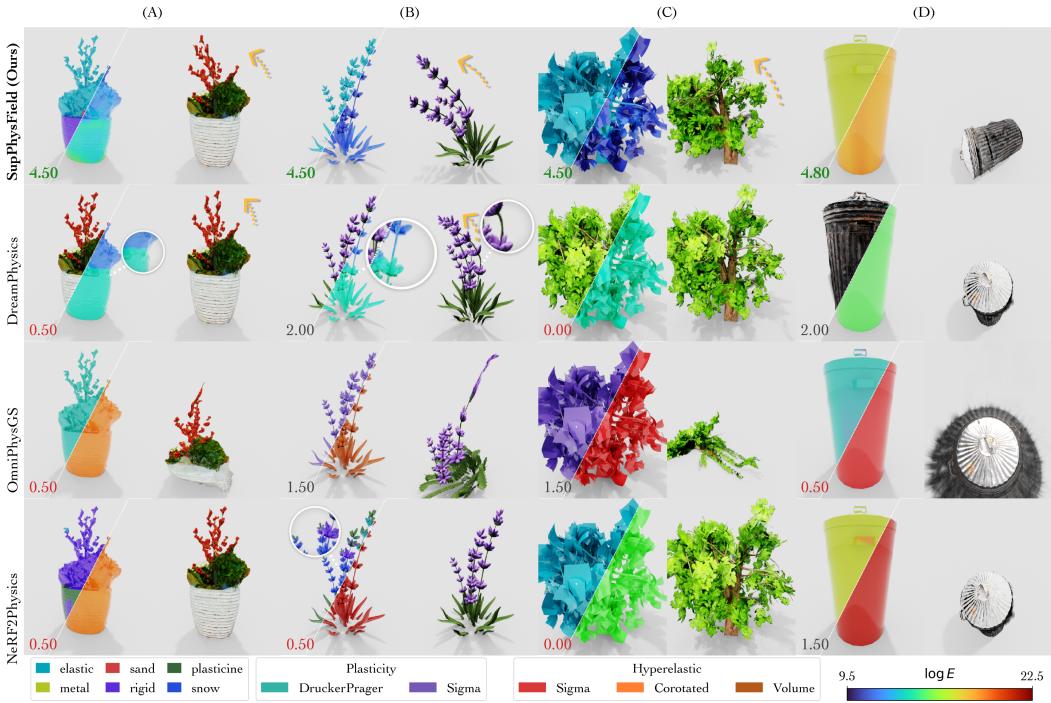


Figure 6: **Qualitative comparison on synthetic scenes.** Best Gemini score per scene is highlighted in **Green** while low scores are in **Red**. We visualized the predicted material class and E predictions (left, right respectively) for SUPPHYSFIELD and Nerf2Physics, E for DreamPhysics (right), and the plasticity and hyperelastic function classes predicted by OmniPhysGS. SUPPHYSFIELD produces stable, physically plausible motion while DreamPhysics remains overly stiff due to inaccurate fine-grained E prediction or too high E (e.g., see tree (C)), OmniPhysGS collapses under load due to unrealistic combination of plasticity and hyperelastic functions, and NeRF2Physics exhibits noisy artifacts. Please <https://neurips-2025-20627.github.io/> for the videos.



Figure 7: **SUPPHYSFIELD’s Zero-shot Real-scene Generalization.** Trained only on synthetic SUPPHYSVERSE, SUPPHYSFIELD can predict plausible physic properties, enabling realistic MPM simulation of real scenes. Here, we visualize the material types (left) and Young’s modulus (right) prediction in the first frame, and subsequent frames impacted by a wind force. Please see the videos in our website <https://neurips-2025-20627.github.io/>.

238 4.2 Zero-shot Generalization to Real-World Scenes

239 Without any real-scene supervision, SUPPHYSFIELD can zero-shot generalize as shown in Fig. 7.
 240 Our method correctly assigns rigid vase bases and flexible leaves, yielding realistic motion that
 241 closely matches human expectation. No other baseline generalises under this setting.

242 4.3 SUPPHYSFIELD’s Feature Type Ablation

243 Replacing CLIP with RGB or occupancy features drops VLM score by 40-60 % and nearly doubles
 244 parameter MSE (Table 1, rows Occupancy and RGB). The material class prediction also dramatically
 245 drops across most classes as shown in Fig. 9. Figure 8 shows the failure modes for real scenes,
 246 highlighting RGB and occupancy’s struggle to generalize to unseen data as compared to CLIP.

247 5 Conclusion and Limitations

248 We presented SUPPHYSFIELD, a framework that jointly reconstructs geometry, appearance, and ex-
 249 plicit physical material fields from posed RGB images. By distilling rich CLIP features into 3D and
 250 training a feed-forward 3D U-Net with per-voxel material supervision on our new SUPPHYSVERSE
 251 dataset, SUPPHYSFIELD avoids the expensive test-time optimization required by prior work. Once
 252 trained, it produces full material fields in a few seconds, improving Gemini realism scores by 14.5%
 253 to 51.8% over DreamPhysics and OmniPhysGS while reducing inference time by three orders of
 254 magnitude. SUPPHYSFIELD leverages CLIP’s strong visual priors, which enables zero-shot trans-
 255 fer to real scenes, even though it is only trained on synthetic data. The method enables realistic,
 256 physically plausible 3D scene animation with off-the-shelf MPM solvers.

257 **Limitations** We take the first step towards learning a supervised model for physical material pre-
 258 diction. Like prior art, our work focuses on single object interaction leaving multi-object scenes

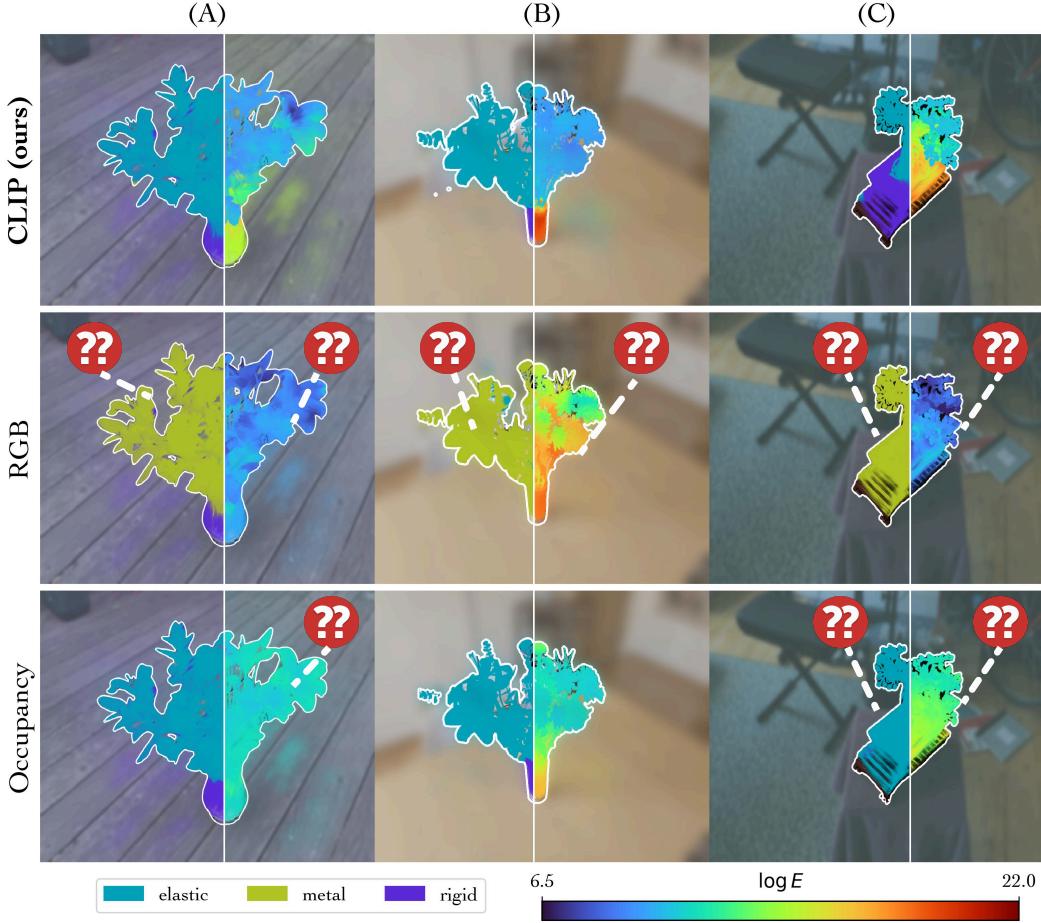


Figure 8: **SUPPHYSFIELD’s Feature Type Ablation on Real Scenes.** Replacing CLIP features with RGB or occupancy severely degrades the material prediction. Incorrect predictions such as leave mislabelled as metal or Young’s modulus being uniform within an object are marked with question marks. This highlights the power of pretrained visual features in bridging the sim2real gap.

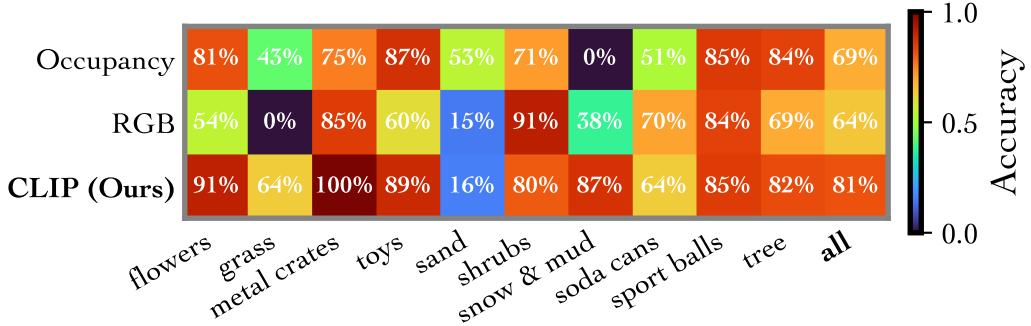


Figure 9: **SUPPHYSFIELD Ablation’s Per-class Accuracy on synthetic scenes.** CLIP features generalizes in synthetic scenes, outperforming RGB and occupancy on 9/10 classes.

for future investigation. Another limitation is that while our UNet predict a point estimate for each voxel, materials in the real-world contain uncertainty that visual information alone cannot resolve (e.g., a tree can be stiff or flexible). A promising extension is to learn a distribution of materials (e.g., using diffusion) instead.

263 **References**

- 264 [1] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Suenderhauf. Physically embodied
265 gaussian splatting: A realtime correctable world model for robotics. In *8th Annual Conference
266 on Robot Learning*, 2024. URL <https://openreview.net/forum?id=AEq0onGrN2>.
- 267 [2] Daniel M Bear, Elias Wang, Damian Mrowca, Felix J Binder, Hsiao-Yu Fish Tung, RT Pramod,
268 Cameron Holdaway, Sirui Tao, Kevin Smith, Fan-Yun Sun, et al. Physion: Evaluating physical
269 prediction from vision in humans and machines. *arXiv preprint arXiv:2106.08261*, 2021.
- 270 [3] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. Material recognition in the wild
271 with the materials in context database. In *Proceedings of the IEEE conference on computer
272 vision and pattern recognition*, pages 3479–3487, 2015.
- 273 [4] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shen-
274 long Wang. Physgen3d: Crafting a miniature interactive world from a single image. *arXiv
275 preprint arXiv:2503.20746*, 2025.
- 276 [5] Chuahao Chen, Zhiyang Dou, Chen Wang, Yiming Huang, Anjun Chen, Qiao Feng, Jiatao Gu,
277 and Lingjie Liu. Vid2sim: Generalizable, video-based reconstruction of appearance, geome-
278 try and physics for mesh-free simulation. *IEEE Conference on Computer Vision and Pattern
279 Recognition (CVPR)*, 2025.
- 280 [6] Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming
281 Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via
282 learning real-world dynamics. *arXiv preprint arXiv:2412.07774*, 2024.
- 283 [7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Lud-
284 wig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of
285 annotated 3d objects, 2022. URL <https://arxiv.org/abs/2212.08051>.
- 286 [8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis.
287 *Advances in neural information processing systems*, 34:8780–8794, 2021.
- 288 [9] Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. Pie-nerf:
289 Physics-based interactive elastodynamics with nerf, 2023.
- 290 [10] Michael Fischer, Iliyan Georgiev, Thibault Groueix, Vladimir G Kim, Tobias Ritschel, and
291 Valentin Deschaintre. Sama: Material-aware 3d selection and segmentation. *arXiv preprint
292 arXiv:2411.19322*, 2024.
- 293 [11] Minghao Guo, Bohan Wang, Pingchuan Ma, Tianyuan Zhang, Crystal Elaine Owens, Chuang
294 Gan, Joshua B. Tenenbaum, Kaiming He, and Wojciech Matusik. Physically compatible 3d
295 object modeling from a single image. *arXiv preprint arXiv:2405.20510*, 2024.
- 296 [12] Hao-Yu Hsu, Zhi-Hao Lin, Albert Zhai, Hongchi Xia, and Shenlong Wang. Autovfx:
297 Physically realistic video editing from natural language instructions. *arXiv preprint
298 arXiv:2411.02394*, 2024.
- 299 [13] Tianyu Huang, Yihan Zeng, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics:
300 Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv
301 preprint arXiv:2406.01476*, 2024.
- 302 [14] Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini,
303 Martin Weiss, Breandan Considine, Jerome Parent-Levesque, Kevin Xie, Kenny Erleben, Liam
304 Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradsim: Differentiable simu-
305 lation for system identification and visuomotor control. *International Conference on Learning
306 Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=c_E8kFWfhp0.
- 307 [15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian
308 splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
- 309 [16] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf:
310 Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Confer-
311 ence on Computer Vision*, pages 19729–19739, 2023.

- 312 [17] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint*
 313 *arXiv:1412.6980*, 2014.
- 314 [18] Long Le, Jason Xie, William Liang, Hung-Ju Wang, Yue Yang, Yecheng Jason Ma, Kyle
 315 Vedder, Arjun Krishna, Dinesh Jayaraman, and Eric Eaton. Articulate-anything: Automatic
 316 modeling of articulated objects via a vision-language foundation model. *arXiv preprint*
 317 *arXiv:2410.13882*, 2024.
- 318 [19] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chen-
 319 fanfu Jiang, and Chuang Gan. PAC-neRF: Physics augmented continuum neural radiance
 320 fields for geometry-agnostic system identification. In *The Eleventh International Conference*
 321 *on Learning Representations*, 2023. URL <https://openreview.net/forum?id=tVkrbkz42vc>.
- 323 [20] Zhengqi Li, Richard Tucker, Noah Snavely, and Aleksander Holynski. Generative image dy-
 324 namics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-*
 325 *nition*, pages 24142–24153, 2024.
- 326 [21] Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong MU. OmniphysGS: 3d constitutive gaus-
 327 sians for general physics-based dynamics generation. In *The Thirteenth International Con-*
 328 *ference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=9HZtP6I51v>.
- 330 [22] Pingchuan Ma, Peter Yichen Chen, Bolei Deng, Joshua B Tenenbaum, Tao Du, Chuang Gan,
 331 and Wojciech Matusik. Learning neural constitutive laws from motion observations for general-
 332 izable pde dynamics. In *International Conference on Machine Learning*, pages 23279–23300.
 333 PMLR, 2023.
- 334 [23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi,
 335 and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communi-*
 336 *cations of the ACM*, 65(1):99–106, 2021.
- 337 [24] Jack Parker-Holder, Philip Ball, Jake Bruce, Vibhavari Dasagi, Kristian Holsheimer, Chris-
 338 tos Kaplani, Alexandre Moufarek, Guy Scully, Jeremy Shar, Jimmy Shi, Stephen Spencer,
 339 Jessica Yung, Michael Dennis, Sultan Kenjeyev, Shangbang Long, Vlad Mnih, Har-
 340 ris Chan, Maxime Gazeau, Bonnie Li, Fabio Pardo, Luyu Wang, Lei Zhang, Frederic
 341 Besse, Tim Harley, Anna Mitenkova, Jane Wang, Jeff Clune, Demis Hassabis, Raia Had-
 342 sell, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 2: A large-scale
 343 foundation world model. 2024. URL <https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/>.
- 345 [25] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF:
 346 Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on*
 347 *Computer Vision and Pattern Recognition*, 2020.
- 348 [26] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature splatting: Language-driven
 349 physics-based scene synthesis and editing. *arXiv preprint arXiv:2404.01223*, 2024.
- 350 [27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agar-
 351 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable
 352 visual models from natural language supervision. In *International conference on machine*
 353 *learning*, pages 8748–8763. PmLR, 2021.
- 354 [28] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola.
 355 Distilled feature fields enable few-shot language-guided manipulation, 2023. URL <https://arxiv.org/abs/2308.07931>.
- 356 [29] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Sori-
 357 cut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of
 358 highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- 359 [30] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are
 360 data-efficient learners for self-supervised video pre-training. *Advances in neural information*
 361 *processing systems*, 35:10078–10093, 2022.

- 363 [31] Chen Wang, Chuhao Chen, Yiming Huang, Zhiyang Dou, Yuan Liu, Jiatao Gu, and Lingjie
364 Liu. Physctrl: Generative physics for controllable and physics-grounded video generation. In
365 *arXiv preprint*, 2025.
- 366 [32] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep
367 self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances
368 in neural information processing systems*, 33:5776–5788, 2020.
- 369 [33] Hongchi Xia, Zhi-Hao Lin, Wei-Chiu Ma, and Shenlong Wang. Video2game: Real-time, inter-
370 active, realistic and browser-compatible environment from a single video, 2024.
- 371 [34] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu
372 Jiang. Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint
373 arXiv:2311.12198*, 2023.
- 374 [35] Albert J Zhai, Yuan Shen, Emily Y Chen, Gloria X Wang, Xinlei Wang, Sheng Wang, Kaiyu
375 Guan, and Shenlong Wang. Physical property understanding from language-embedded feature
376 fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-
377 nition*, pages 28296–28305, 2024.
- 378 [36] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely,
379 Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-based interaction with 3d objects
380 via video generation. In *European Conference on Computer Vision*. Springer, 2024.
- 381 [37] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation
382 of elastic objects with spring-mass 3d gaussians. *European Conference on Computer Vision
(ECCV)*, 2024.

384 **A Appendix**

385 **A.1 Preliminaries**

386 This section briefly reviews foundational concepts in 3D scene representation and physics modeling
 387 relevant to our work.

388 **A.1.1 Learned Scene Representation**

389 Reconstructing 3D scenes from 2D images is commonly achieved by learning a parameterized repre-
 390 sentation, F_θ , optimized to render novel views that match observed images $\{I^{(i)}\}_{i=1}^M$ given camera
 391 parameters $\{\pi^{(i)}\}_{i=1}^M$. This typically involves minimizing a photometric loss:

$$\min_{\theta} \sum_{i=1}^M \left\| \hat{I}^{(i)}(\theta) - I^{(i)} \right\|_2^2 ,$$

392 where $\hat{I}^{(i)}(\theta)$ is the image rendered from viewpoint i . Two prominent representations are Neural
 393 Radiance Fields (NeRF) and Gaussian Splatting (GS) models.

394 **Neural Radiance Fields (NeRF)** [23] model a scene as a continuous function $F_\theta : (\mathbf{x}, \mathbf{d}) \mapsto (c, \sigma)$,
 395 mapping a 3D location \mathbf{x} and viewing direction \mathbf{d} to an emitted color c and volume density σ .
 396 Images are synthesized using volume rendering, integrating color and density along camera rays.
 397 This process' differentiability allows for end-to-end optimization from images.

398 **Gaussian Splatting (GS)** [15] represents scenes as a collection of 3D Gaussian primitives, each
 399 defined by a center μ_i , covariance Σ_i , color \mathbf{c}_i , and opacity α_i . These Gaussians are projected onto
 400 the image plane and blended using alpha compositing to render views.

401 In our work, the principles of neural scene representation, particularly NeRF-like architectures, are
 402 leveraged not only for visual reconstruction but also for creating dense 3D visual feature fields. As
 403 detailed in Sec. 3.1, we utilize a NeRF-based model to distill 2D image features (e.g., from CLIP)
 404 into a volumetric 3D feature grid. This 3D feature representation, F_G , then serves as the primary
 405 input to our physics prediction network. For subsequent physics simulation, GS offers a convenient
 406 particle-based representation.

407 **A.1.2 Material Point Method (MPM) for Physics Simulation**

408 To simulate how objects move and deform under applied forces, a physics engine requires knowl-
 409 edge of their material properties. These properties are typically defined within the framework of
 410 continuum mechanics, which describes the behavior of materials at a macroscopic level. The funda-
 411 mental equations of motion (conservation of mass and momentum) are:

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \mathbf{f}^{\text{ext}} \quad \nabla \cdot \mathbf{v} = 0 , \quad (4)$$

412 where ρ is mass density, \mathbf{v} the velocity field, $\boldsymbol{\sigma}$ the Cauchy stress tensor, and \mathbf{f}^{ext} any external force
 413 (e.g. gravity or user interactions). The material-specific *constitutive laws* define how $\boldsymbol{\sigma}$ depends on
 414 the local deformation gradient \mathbf{F} . For elastic materials, stress depends purely on the recoverable
 415 strain; for plastic materials, a yield condition enforces partial flow once strain exceeds a threshold.

416 **Constitutive Laws and Parameters** Most continuum simulations separate the constitutive model
 417 into two core components:

$$\begin{aligned} \mathcal{E}_\mu : \mathbf{F}^e &\mapsto \mathbf{P}, \\ \mathcal{P}_\mu : \mathbf{F}^{e,\text{trial}} &\mapsto \mathbf{F}^{e,\text{new}}, \end{aligned} \quad (5)$$

418 where \mathbf{F}^e is the *elastic* portion of the deformation gradient, \mathbf{P} is the (First) Piola–Kirchhoff stress,
 419 and μ represents the set of material parameters (e.g. Young's modulus E , Poisson's ratio ν , yield
 420 stress). The *elastic law* \mathcal{E}_μ computes stress from the current elastic deformation, while the *return-*
 421 *mapping* \mathcal{P}_μ projects any trial elastic update $\mathbf{F}^{e,\text{trial}}$ onto the feasible yield surface if plastic flow
 422 is triggered. Typically, the constitutive laws i.e., \mathcal{E}_μ and \mathcal{P}_μ are hand-designed by domain experts.
 423 The choice of \mathcal{E} and \mathcal{P} jointly define a class of material (e.g., rubber). Within a material class,
 424 additional continuous parameters μ including Young's modulus, Poisson's ratio and density can be
 425 specified for a more granular control of the material properties (e.g., stiffness of rubber). In our work,
 426 SUPPHYSFIELD jointly predicts the discrete material model and the continuous material parameters.

427 **A.2 SUPPHYSVERSE Dataset Details**

428 We heavily curate the dataset to a set of 1624 objects after a multi-stage filter that removes multi-
429 object scenes, missing textures, duplicated assets, and objects whose material labeling is either am-
430 biguous or physically implausible.

431 First, we define some object class (e.g., “tree”) and some alternative query terms (e.g., “ficus, fern,
432 evergreen etc”). We then use a sentence transformer model [32] to compute the cosine similarity
433 between the search terms and the name of each Objaverse object. We select $k = 500$ objects
434 with the highest similarity score for each class, creating an initial candidate pool. However, since
435 Objaverse objects vary greatly in asset quality, lighting conditions, and some scenes contain multiple
436 objects which are not suitable for our material learning, an additional filtering step is needed. The
437 Gemini VLM is prompted to filter out low-quality or unsuitable scenes. A distilled NeRF model
438 is fitted to each object. Then, the VLM is provided five multi-view RGB images of an object, and
439 prompted to provide a list of the object’s semantic parts along with associated material class and
440 ranges for continuous values (e.g., see Fig. 10). The ranges such as $E \in \{1e4, 1e5\}$ allow us to
441 simulate a wider range of dynamics from flexible to more rigid trees. The VLM is also prompted to
442 specify a list of constraints such as to ensure that the leaf’s density is lower than the trunk’s. We then
443 sample the continuous values from the VLM’s specified ranges subject to the constraint via rejection
444 sampling. The semantic parts (e.g., “pot”) are used with the CLIP distilled feature field to compute
445 a 3D semantic segmentation of the object into parts, and the sampled material properties are applied
446 uniformly to all points within a part. This ground-truth material and feature fields are then voxelized
447 into regular grids for use in supervised learning by the SUPPHYSFIELD framework.

```
{ "pot": {"density": [400, 600], "E": [1e8, 2e8], "nu": [0.2, 0.4], "material_id": 6},  
  "trunk": {"density": [300, 500], "E": [5e5, 1e7], "nu": [0.3, 0.45], "material_id": 0},  
  "leaf": {"density": [100, 300], "E": [1e4, 1e5], "nu": [0.35, 0.48], "material_id": 0},  
  "constraints" : "assert leaf_{density} < trunk_{density}, ..."}}
```

Figure 10: An example of a material annotation by Gemini VLM for the SUPPHYSVERSE dataset.

448 **NeurIPS Paper Checklist**

449 **1. Claims**

450 Question: Do the main claims made in the abstract and introduction accurately reflect the paper's
451 contributions and scope?

452 Answer: [Yes]

453 Justification: The abstract and introduction state the claims and contributions of this paper.

454 Guidelines:

- 455 • The answer NA means that the abstract and introduction do not include the claims made in the
456 paper.
- 457 • The abstract and/or introduction should clearly state the claims made, including the contribu-
458 tions made in the paper and important assumptions and limitations. A No or NA answer to this
459 question will not be perceived well by the reviewers.
- 460 • The claims made should match theoretical and experimental results, and reflect how much the
461 results can be expected to generalize to other settings.
- 462 • It is fine to include aspirational goals as motivation as long as it is clear that these goals are not
463 attained by the paper.

464 **2. Limitations**

465 Question: Does the paper discuss the limitations of the work performed by the authors?

466 Answer: [Yes]

467 Justification: We discuss the limitations in the last section.

468 Guidelines:

- 469 • The answer NA means that the paper has no limitation while the answer No means that the
470 paper has limitations, but those are not discussed in the paper.
- 471 • The authors are encouraged to create a separate "Limitations" section in their paper.
- 472 • The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-
473 specification, asymptotic approximations only holding locally). The authors should reflect on
474 how these assumptions might be violated in practice and what the implications would be.
- 475 • The authors should reflect on the scope of the claims made, e.g., if the approach was only
476 tested on a few datasets or with a few runs. In general, empirical results often depend on
477 implicit assumptions, which should be articulated.
- 478 • The authors should reflect on the factors that influence the performance of the approach. For
479 example, a facial recognition algorithm may perform poorly when image resolution is low or
480 images are taken in low lighting. Or a speech-to-text system might not be used reliably to
481 provide closed captions for online lectures because it fails to handle technical jargon.
- 482 • The authors should discuss the computational efficiency of the proposed algorithms and how
483 they scale with dataset size.
- 484 • If applicable, the authors should discuss possible limitations of their approach to address prob-
485 lems of privacy and fairness.
- 486 • While the authors might fear that complete honesty about limitations might be used by review-
487 ers as grounds for rejection, a worse outcome might be that reviewers discover limitations that
488 aren't acknowledged in the paper. The authors should use their best judgment and recognize
489 that individual actions in favor of transparency play an important role in developing norms
490 that preserve the integrity of the community. Reviewers will be specifically instructed to not
491 penalize honesty concerning limitations.

493 **3. Theory assumptions and proofs**

494 Question: For each theoretical result, does the paper provide the full set of assumptions and a
495 complete (and correct) proof?

496 Answer: [NA]

497 Justification: N/A

498 Guidelines:

- 499 • The answer NA means that the paper does not include theoretical results.
- 500 • All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- 501 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 502 • The proofs can either appear in the main paper or the supplemental material, but if they appear
503 in the supplemental material, the authors are encouraged to provide a short proof sketch to
504 provide intuition.

- 505 • Inversely, any informal proof provided in the core of the paper should be complemented by
 506 formal proofs provided in appendix or supplemental material.
 507 • Theorems and Lemmas that the proof relies upon should be properly referenced.

508 4. **Experimental result reproducibility**

509 Question: Does the paper fully disclose all the information needed to reproduce the main experi-
 510 mental results of the paper to the extent that it affects the main claims and/or conclusions of the
 511 paper (regardless of whether the code and data are provided or not)?

512 Answer: [Yes]

513 Justification: The paper discusses all implementation details necessary for reproduction. We will
 514 also release the training data, code, and checkpoints.

515 Guidelines:

- 516 • The answer NA means that the paper does not include experiments.
- 517 • If the paper includes experiments, a No answer to this question will not be perceived well by
 518 the reviewers: Making the paper reproducible is important, regardless of whether the code and
 519 data are provided or not.
- 520 • If the contribution is a dataset and/or model, the authors should describe the steps taken to
 521 make their results reproducible or verifiable.
- 522 • Depending on the contribution, reproducibility can be accomplished in various ways. For
 523 example, if the contribution is a novel architecture, describing the architecture fully might
 524 suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary
 525 to either make it possible for others to replicate the model with the same dataset, or provide
 526 access to the model. In general, releasing code and data is often one good way to accomplish
 527 this, but reproducibility can also be provided via detailed instructions for how to replicate the
 528 results, access to a hosted model (e.g., in the case of a large language model), releasing of a
 529 model checkpoint, or other means that are appropriate to the research performed.
- 530 • While NeurIPS does not require releasing code, the conference does require all submissions
 531 to provide some reasonable avenue for reproducibility, which may depend on the nature of the
 532 contribution. For example
 - 533 (a) If the contribution is primarily a new algorithm, the paper should make it clear how to
 534 reproduce that algorithm.
 - 535 (b) If the contribution is primarily a new model architecture, the paper should describe the
 536 architecture clearly and fully.
 - 537 (c) If the contribution is a new model (e.g., a large language model), then there should either
 538 be a way to access this model for reproducing the results or a way to reproduce the model
 539 (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - 540 (d) We recognize that reproducibility may be tricky in some cases, in which case authors are
 541 welcome to describe the particular way they provide for reproducibility. In the case of
 542 closed-source models, it may be that access to the model is limited in some way (e.g.,
 543 to registered users), but it should be possible for other researchers to have some path to
 544 reproducing or verifying the results.

545 5. **Open access to data and code**

546 Question: Does the paper provide open access to the data and code, with sufficient instructions
 547 to faithfully reproduce the main experimental results, as described in supplemental material?

548 Answer: [Yes]

549 Justification: We will release the training data, code, and checkpoints.

550 Guidelines:

- 551 • The answer NA means that paper does not include experiments requiring code.
- 552 • Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 553 • While we encourage the release of code and data, we understand that this might not be possible,
 554 so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless
 555 this is central to the contribution (e.g., for a new open-source benchmark).
- 556 • The instructions should contain the exact command and environment needed to run to repro-
 557 duce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- 558 • The authors should provide instructions on data access and preparation, including how to access
 559 the raw data, preprocessed data, intermediate data, and generated data, etc.

- 562 • The authors should provide scripts to reproduce all experimental results for the new proposed
 563 method and baselines. If only a subset of experiments are reproducible, they should state which
 564 ones are omitted from the script and why.
 565 • At submission time, to preserve anonymity, the authors should release anonymized versions (if
 566 applicable).
 567 • Providing as much information as possible in supplemental material (appended to the paper) is
 568 recommended, but including URLs to data and code is permitted.

569 **6. Experimental setting/details**

570 Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters,
 571 how they were chosen, type of optimizer, etc.) necessary to understand the results?

572 Answer: [Yes]

573 Justification: We provide implementation details.

574 Guidelines:

- 575 • The answer NA means that the paper does not include experiments.
- 576 • The experimental setting should be presented in the core of the paper to a level of detail that is
 577 necessary to appreciate the results and make sense of them.
- 578 • The full details can be provided either with the code, in appendix, or as supplemental material.

579 **7. Experiment statistical significance**

580 Question: Does the paper report error bars suitably and correctly defined or other appropriate
 581 information about the statistical significance of the experiments?

582 Answer: [Yes]

583 Justification: We include standard error bars along with the mean scores.

584 Guidelines:

- 585 • The answer NA means that the paper does not include experiments.
- 586 • The authors should answer "Yes" if the results are accompanied by error bars, confidence inter-
 587 vals, or statistical significance tests, at least for the experiments that support the main claims of
 588 the paper.
- 589 • The factors of variability that the error bars are capturing should be clearly stated (for example,
 590 train/test split, initialization, random drawing of some parameter, or overall run with given
 591 experimental conditions).
- 592 • The method for calculating the error bars should be explained (closed form formula, call to a
 593 library function, bootstrap, etc.)
- 594 • The assumptions made should be given (e.g., Normally distributed errors).
- 595 • It should be clear whether the error bar is the standard deviation or the standard error of the
 596 mean.
- 597 • It is OK to report 1-sigma error bars, but one should state it. The authors should preferably
 598 report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of
 599 errors is not verified.
- 600 • For asymmetric distributions, the authors should be careful not to show in tables or figures
 601 symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- 602 • If error bars are reported in tables or plots, The authors should explain in the text how they
 603 were calculated and reference the corresponding figures or tables in the text.

604 **8. Experiments compute resources**

605 Question: For each experiment, does the paper provide sufficient information on the computer
 606 resources (type of compute workers, memory, time of execution) needed to reproduce the experi-
 607 ments?

608 Answer: [Yes]

609 Justification: We provide details on our hardware setup and training duration.

610 Guidelines:

- 611 • The answer NA means that the paper does not include experiments.
- 612 • The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud
 613 provider, including relevant memory and storage.
- 614 • The paper should provide the amount of compute required for each of the individual experi-
 615 mental runs as well as estimate the total compute.
- 616 • The paper should disclose whether the full research project required more compute than the
 617 experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it
 618 into the paper).

619 **9. Code of ethics**

620 Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS
621 Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

622 Answer: [Yes]

623 Justification: We conform to the NeurIPS Code of Ethics.

624 Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

630 **10. Broader impacts**

631 Question: Does the paper discuss both potential positive societal impacts and negative societal
632 impacts of the work performed?

633 Answer: [NA]

634 Justification: The authors have not ascertained a path towards misuse using this technology.

635 Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

657 **11. Safeguards**

658 Question: Does the paper describe safeguards that have been put in place for responsible release
659 of data or models that have a high risk for misuse (e.g., pretrained language models, image
660 generators, or scraped datasets)?

661 Answer: [NA]

662 Justification: The authors have not ascertained a path towards misuse using this technology.

663 Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

672 **12. Licenses for existing assets**

673 Question: Are the creators or original owners of assets (e.g., code, data, models), used in the
674 paper, properly credited and are the license and terms of use explicitly mentioned and properly
675 respected?

676 Answer: [Yes]

677 Justification: The creators of the original dataset and models are properly cited.

678 Guidelines:

- 679 • The answer NA means that the paper does not use existing assets.
680 • The authors should cite the original paper that produced the code package or dataset.
681 • The authors should state which version of the asset is used and, if possible, include a URL.
682 • The name of the license (e.g., CC-BY 4.0) should be included for each asset.
683 • For scraped data from a particular source (e.g., website), the copyright and terms of service of
684 that source should be provided.
685 • If assets are released, the license, copyright information, and terms of use in the package should
686 be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for
687 some datasets. Their licensing guide can help determine the license of a dataset.
688 • For existing datasets that are re-packaged, both the original license and the license of the de-
689 rived asset (if it has changed) should be provided.
690 • If this information is not available online, the authors are encouraged to reach out to the asset's
691 creators.

692 **13. New assets**

693 Question: Are new assets introduced in the paper well documented and is the documentation
694 provided alongside the assets?

695 Answer: [Yes]

696 Justification: We discuss our dataset at length in Sec. 3.2.

697 Guidelines:

- 698 • The answer NA means that the paper does not release new assets.
- 699 • Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- 700 • The paper should discuss whether and how consent was obtained from people whose asset is used.
- 701 • At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

702 **14. Crowdsourcing and research with human subjects**

703 Question: For crowdsourcing experiments and research with human subjects, does the paper
704 include the full text of instructions given to participants and screenshots, if applicable, as well as
705 details about compensation (if any)?

706 Answer: [NA]

707 Justification: The paper does not involve human subjects.

708 Guidelines:

- 709 • The answer NA means that the paper does not involve crowdsourcing nor research with human
710 subjects.
- 711 • Including this information in the supplemental material is fine, but if the main contribution of
712 the paper involves human subjects, then as much detail as possible should be included in the
713 main paper.
- 714 • According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or
715 other labor should be paid at least the minimum wage in the country of the data collector.

716 **15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

717 Question: Does the paper describe potential risks incurred by study participants, whether such
718 risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals
719 (or an equivalent approval/review based on the requirements of your country or institution) were
720 obtained?

721 Answer: [NA]

722 Justification: The paper does not involve human subjects.

723 Guidelines:

- 724 • The answer NA means that the paper does not involve crowdsourcing nor research with human
725 subjects.
- 726 • Depending on the country in which research is conducted, IRB approval (or equivalent) may
727 be required for any human subjects research. If you obtained IRB approval, you should clearly
728 state this in the paper.
- 729 • We recognize that the procedures for this may vary significantly between institutions and loca-
730 tions, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for
731 their institution.
- 732 • For initial submissions, do not include any information that would break anonymity (if appli-
733 cable), such as the institution conducting the review.

734 **16. Declaration of LLM usage**

738 Question: Does the paper describe the usage of LLMs if it is an important, original, or non-
739 standard component of the core methods in this research? Note that if the LLM is used only
740 for writing, editing, or formatting purposes and does not impact the core methodology, scientific
741 rigorousness, or originality of the research, declaration is not required.

742 Answer: [Yes]

743 Justification: The paper discuss the use of LLMs as it is critical to the paper's approach.

744 Guidelines:

- 745 • The answer NA means that the core method development in this research does not involve
746 LLMs as any important, original, or non-standard components.
- 747 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what
748 should or should not be described.