Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Vysics: Object Reconstruction Under Occlusion by Fusing Vision and Contact-Rich Physics

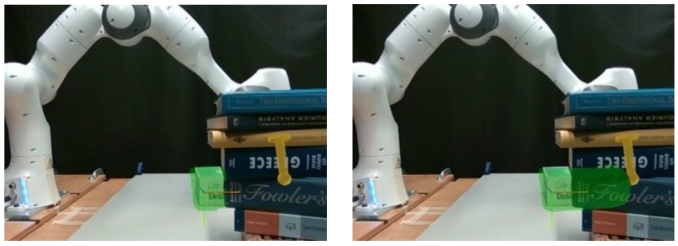
Bibit Bianchini*†, Minghan Zhu*†, Mengti Sun‡, Bowen Jiang†, Camillo J. Taylor†, Michael Posa†
*The first two authors contributed equally to this work.
†General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory
University of Pennsylvania, Philadelphia, PA 19104
Emails: {bibit, minghz, bwjiang, cjtaylor, posa}@seas.upenn.edu
‡Amazon, Seattle, WA 98004
Email: mengtis@amazon.com

***Abstract*—We introduce Vysics, a vision-and-physics framework for a robot to build an expressive geometry and dynamics model of a single rigid body, using a seconds-long RGBD video and the robot's proprioception. While the computer vision community has built powerful visual 3D perception algorithms, cluttered environments with heavy occlusions can limit the visibility of objects of interest. However, observed motion of partially occluded objects can imply physical interactions took place, such as contact with a robot or the environment. These inferred contacts can supplement the visible geometry with "physible geometry," which best explains the observed object motion through physics. Vysics uses a vision-based tracking and reconstruction method, BundleSDF, to estimate the trajectory and the visible geometry from an RGBD video, and an odometry-based model learning method, Physics Learning Library (PLL), to infer the "physible" geometry from the trajectory through implicit contact dynamics optimization. The visible and "physible" geometries jointly factor into optimizing a signed distance function (SDF) to represent the object shape. Vysics does not require pretraining, nor tactile or force sensors. Compared with vision-only methods, Vysics yields object models with higher geometric accuracy and better dynamics prediction in experiments where the object interacts with the robot and the environment under heavy occlusion. Project page: https://vysics-vision-and-physics.github.io/**

I. Introduction

In-the-wild manipulation will require robots to encounter a vast array of different objects. While some might be recognized from an existing database, others will require physical interaction to be newly understood on the spot. Dexterous manipulation of these objects will benefit from the ability to rapidly learn or identify object properties: geometry is most critical, but inertial properties are also valuable for predicting motion, particularly under forceful manipulation. Use of such models boasts the benefits of interpretability and expected generalizability, at the cost of requiring the model. This paper presents Vysics, which builds dynamics models of novel objects from vision and physical interaction, even in the face of substantial visual occlusions (Figure 1).

Rapid modeling requires combining all available information in a unified fashion. This work leverages recent results from visual tracking and object reconstruction [66] in combination with contact-implicit model learning [9, 53] via the shared connection of object geometry Depth camera measurements on an object's surface are direct observations of portions of its geometry, and observations of the object's state evolution can inject more geometric information when contact is inferred. Visual information is limited by occlusions, but contact, which typically occurs on visually-occluded faces of objects, provides a secondary source of information: "physible" geometry.



(a) BundleSDF [66] (b) Vysics (ours)

Fig. 1: Vision-based shape reconstruction (projection shown in green) can be limited by occlusion. Fusing vision and contact-rich physics, our method recovers the occluded geometry through object interactions with the robot and environment. The robot end effector in yellow shows the robot-object interaction.

Consider the example depicted in Figure 1 of a robot arm interacting with an object, which is significantly obscured by a stack of books in the foreground. A vision-only approach would only reconstruct the portions of the geometry that is visible but is also able to detect object motions. By reasoning about physics, we can infer more about the geometry by assuming object trajectories are explainable by its physical interactions with the environment and robot. In the Figure 1 example, the robot end effector is occluded as it pushes the object from its occluded end. While not explained by visible geometry, this guides estimation of the "physible" geometry, namely that the geometry should extend to the right to make contact with the robot.

Estimating geometry through contact-rich interactions is not a trivial problem. Object trajectories result from the sum of all forces on the object, including continuous and contact forces, whose interplay relies on more object properties than just geometry [9]. Additionally, making and breaking contact

introduces multi-modality into the dynamics, further complicating system identification [5, 52]. We take an approach that embraces the multi-modal nature of the dynamics [8, 53], starting by feeding it visually-estimated trajectories, then fusing visually-observed with physically-inferred geometry.

Our aim is to leverage vision and contact-rich physics to perform dynamics model building directly from RGBD data in extreme low data regimes – learning a dynamics model whose geometry matches or outperforms vision-based approaches while learning other critical simulation parameters like inertia, with only seconds of data.

A. Contributions and Outline

We make the following contributions in this work:

- Introduce Vysics, a novel method that builds expressive dynamics models of objects from RGBD videos featuring contact-rich trajectories. With seconds of data, Vysics can identify an object's geometry with no fundamental priors as well as its inertial properties, automatically generating an accompanying Unified Robotics Description Format (URDF) file and mesh.
- Present and make available a new RGBD video dataset of a Franka Panda robot arm interacting with objects on a table in the face of significant visual occlusions (code and data can be found on our project page).
- Demonstrate our method's efficacy on shape reconstruction and learning dynamics models. We compare against vision-only shape reconstruction and show the benefits of the additional contact physics information source with small amounts of data.

In §II, we review related work from computer vision and dynamics model learning. We discuss preliminaries for our work in §III before outlining Vysics in §IV. §V describes experiment details before §VI presents results, followed by a discussion of limitations in §VII and conclusion in §VIII.

II. Related Work

A. Vision-Based Geometry Reconstruction and Completion

Contact is at the heart of robotic manipulation, and contact is driven by geometry. Computer vision has made strides in geometry reconstruction from image inputs. Dense visual SLAM systems [10, 61, 41] recover dense geometry from multi-view RGB or depth streams, yet leave occluded regions unresolved. Neural implicit methods represent shapes as continuous fields: foundational works like [50, 43] encode object-level geometry, while [6, 47] extend these representations to full scenes.

When portions of the object remain unseen, learned completion methods infer missing geometry from partial data. Synthetic shape datasets [13] enable learning shape priors of common object categories, facilitating various tasks including shape completion [72, 19], point cloud completion [73, 16], and single-image shape reconstruction [45, 18]. However, most operate in an object-centered canonical frame, limiting their applicability to raw sensor data, with a few exceptions [62, 55] leveraging 2.5D representations to guide camera-frame shape learning. Robotics and VR applications have prompted camera-frame shape completion from partial point clouds [63, 46] or RGBD observations [69, 37], with recent extensions to multi-object scenes [68, 30]. Moreover, advances in image generation models [54], 3D scene representations [44, 32], and large-scale 3D object datasets [22, 21] have spurred 3D generative pipelines [39, 42, 27, 38, 71, 76], though these typically require unobstructed RGB inputs and do not integrate depth. Unlike these vision-only and data-driven alternatives, Vysics requires no pretraining or canonical pose assumptions: it relies solely on vision-based measurements and physics-based hypotheses extracted from a short RGBD video to regress a class-agnostic shape.

B. Vision-Based Object Pose Estimation

While there are many works that perform robotic manipulation without explicitly estimating object poses (e.g. [14]), access to object pose estimates remains a commonplace assumption in robotics. Classical vision-based pose estimators require the 3D shape model of the target object to facilitate the pose estimate [34, 33, 51], often impractical in realistic applications. Other new approaches do not require geometry models beforehand [67], but this has its own challenges. Model-free methods without this assumption rely on sparse keypoints, dense optical flow, or deep features with reference frames with known pose [12, 59], without building a coherent object model. Such approaches are susceptible to long-term drift, though even weak geometry-based supervision can help [28]. Limiting a pose estimator's scope to tracking in-category objects can boost pose accuracy [36, 23, 15], but limits generalization.

C. Simultaneous Tracking and Shape Reconstruction

Recent computer vision works combine the pose and shape estimation problems [75, 31, 66, 58]. Solving both problems simultaneously has its benefits, including that maintaining a geometry estimate can improve novel-object pose estimation and vice versa [57, 64]. Other applications include multiple sparse-view alignment [40, 59, 33] and single-view pose estimation [48, 74].

D. Trajectory-Based Dynamics Model Learning

System identification is an important robotics subfield that aims to build accurate system models, which can then be leveraged via model-based control techniques. Differentiable simulators [35, 29, 20] have pushed recent advancements, though they can struggle to find good solutions when applied in contact-rich settings [5, 8]. While high-stiffness contact dynamics generally are a challenge for system identification [52], more creative strategies can make finding inertial parameters [24], contact parameters [53], or both [9] possible and efficient. Many of these methods rely on access to state information, limiting their use to lab-based settings where fiducials are available, or necessitating integration with a generalizable vision-based solution.

E. Physics as a Prior for Shape Reconstruction

Some prior works have used physics as a prior for shape reconstruction. Unlike Vysics, these other works either assume objects of interest are statically stable [49, 3] or avoid learning other non-geometric dynamic properties like inertia [1, 56]. The most similar in spirit to Vysics is [56], as they reconstruct geometries with occlusions through physical robot and environment interactions. While [56] avoids the problematic gradients in contact-rich scenarios by using a gradient-free search over a discrete set of hypothesized geometries, Vysics leverages smooth, implicit-based losses and thus can directly regress the object geometry based on visual measurements. This enables Vysics to avoid the need for a high-dimensional graph search that limits geometries to a relatively small set of possibilities.

III. Background

Vysics leverages a vision-based pose and shape estimation tool, BundleSDF [66], and a physics-inspired dynamics learning tool, Physics Learning Library (PLL) [9, 53], which are described in §III-A and §III-B, respectively. Neither of these tools requires pretraining on an extensive dataset. BundleSDF and PLL both train on and generate results from only the measurements provided or inferred by their per-instance input data, making them suitable for immediate application when a robot encounters and needs to model an object.

A. Vision-Based Pose and Shape Estimation

BundleSDF [66] takes masked RGBD videos as input, and outputs both estimated poses of the object over time as well as a geometry estimate. It is built on top of the generalizable, model-free 6D pose tracker, BundleTrack [65]. Alongside the pose estimator, BundleSDF introduces a parallel Neural Object Field that simultaneously maintains an estimate of the object's geometry, which is used to further refine the estimated poses to encourage long-term consistency.

To represent geometry, BundleSDF uses a signed distance function (SDF), an expressive implicit shape representation with no imposed restrictions such as an inherent resolution limit or convexity prior. Due to their expressivity, SDFs have seen great success in representing shapes of objects and environments, even with generalization to unseen objects [50]. An SDF is a function of the form,

$$\text{SDF}(\mathbf{p}) = d, \tag{1}$$

where any 3D point $\mathbf{p}$ relative to a geometry's reference frame can be queried, and the scalar signed distance d away to the nearest surface point is returned. See Figure 2 for illustration.

From RGBD data, not only do depth returns imply points at which $d = 0$, but samples along the camera rays of valid object depth returns can additionally yield supervision for points with $d > 0$. The result is that BundleSDF trains its neural network SDF with supervision from sets of points and signed distances $\{(\mathbf{p}, d)_i\}$. BundleSDF implements a hybrid SDF model on this set of points, by only regressing points whose target signed distance is within a truncation distance, as in [60, 50]. We refer interested readers to [66] for more details on BundleSDF loss components. Upon termination, the marching cubes algorithm extracts a mesh from the underlying SDF.

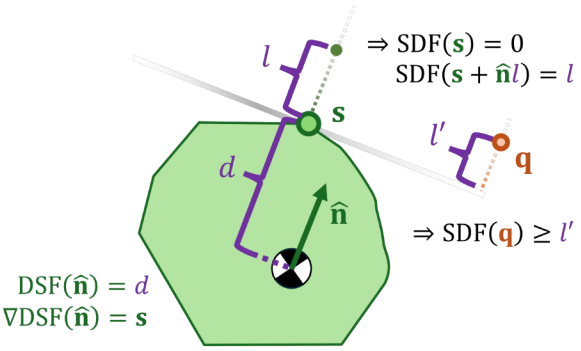


Fig. 2: A 2D depiction of the physical meaning of a DSF (3) and its implication on the SDF (1). Shades of green points have exact SDF values and are subject to (8), and the orange example point $\mathbf{q}$'s signed distance can be lower-bounded by the supporting hyperplane as in (11).

B. Physics-Inspired Dynamics Learning

PLL [9, 53] is an odometry-based method that jointly learns continuous and contact dynamics of objects undergoing contact-rich trajectories. Given only a state trajectory (no direct observation of contact events), PLL is able to identify the geometry, frictional properties, and inertia, all from hypothesizing contact forces that best explain observed state transitions. Its key insight is in its implicit loss formulation, which has provably better generalization than alternatives that require differentiable simulation [8]. The loss applies a cost to the current model belief's incompatibility with the hypothesized contact forces; see [9, 53] for more details.

The geometry of the object directly determines when it makes and breaks contact with the robot and environment, thus playing a critical role in its dynamics. PLL solves the inverse problem: estimate the contact geometry from observed dynamics. The estimated geometry is represented using PLL's deep support function (DSF) [25], an input-convex, homogeneous deep neural network [4]. Like an SDF, a DSF has no inherent resolution limits, however it can only represent the convex hull of any given shape. In our application where our robot interacts with convex and nonconvex objects, this assumption is satisfied as long as collisions occur only on the object's convex hull. A DSF has all the expressivity required for those scenarios while also being concise and fast to compute.

A DSF's input can be any 3D point, but for physically meaningful outputs, PLL only queries the DSF with unit vectors. A DSF takes as input a unit vector in the object's body frame and outputs the scalar distance the geometry extends in that direction [7]. The gradient of a DSF with respect to its input is (almost always [25]) the 3D point on the object geometry that extends furthest in the queried input direction. Mathematically, for an object whose surface (or volume) is represented by the set $\mathcal{S}$, a DSF yields the following output

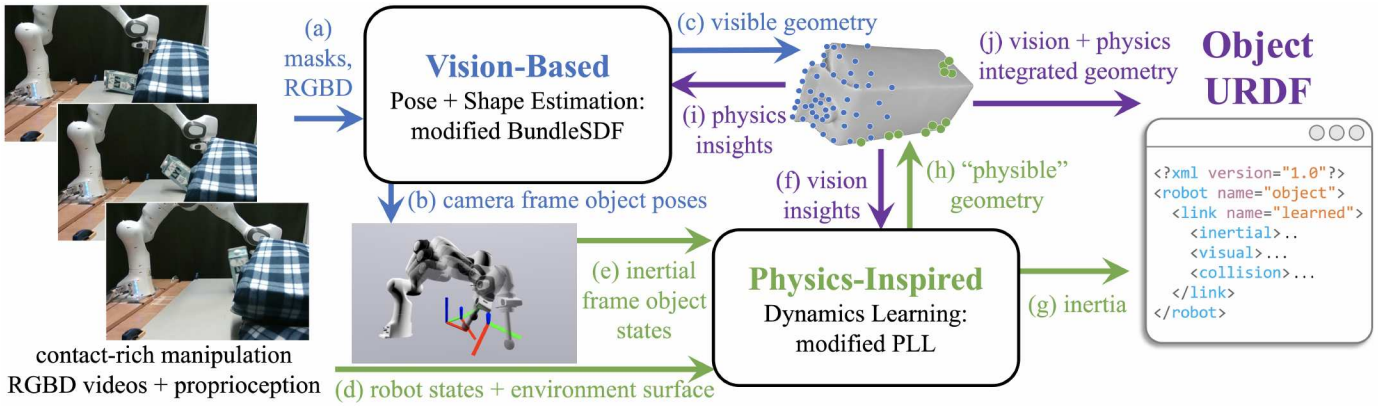


Fig. 3: Detailed Vysics diagram. Blue arrows denote the vision-based information flow through BundleSDF [66], and green arrows for PLL [9, 53]. Purple arrows indicate the unifying connections Vysics makes to factor both vision and contact-rich physics into the geometry learning problem.

and gradient,

$$\text{DSF}(\hat{\mathbf{n}}) = \max_{\mathbf{s}_i \in \mathcal{S}} \mathbf{s}_i \cdot \hat{\mathbf{n}}, \tag{2}$$

$$\nabla_{\hat{\mathbf{n}}}\text{DSF}\left(\hat{\mathbf{n}}\right) = \arg\max_{\mathbf{s}_i \in \mathcal{S}} \mathbf{s}_i \cdot \hat{\mathbf{n}} =: \mathbf{s}. \tag{3}$$

The queried normal direction $\hat{\mathbf{n}}$ and its associated support point $\mathbf{s}$ fully define a supporting hyperplane of the object geometry [11]. See Figure 2 for an example of a queried normal direction $\hat{\mathbf{n}}$, its corresponding support point $\mathbf{s}$, and the constraints they impose on the SDF (explained in detail in section IV-B). A mesh may be exported by PLL with vertices $\{\mathbf{s}_i\}$ as outputs of $\nabla_{\hat{\mathbf{n}}}$DSF on a uniformly sampled set of unit vectors $\{\hat{\mathbf{n}}_i\}$.

IV. Approach

Figure 3 illustrates Vysics from input RGBD videos and robot states (left) to URDF output (right). Its core components are BundleSDF [66] for vision-based tracking and shape reconstruction, and PLL [9, 53] for physics-inspired dynamics learning. Beyond the insights that led to this systems integration, our main contribution lies in how Vysics incorporates these two powerful tools together such that they supervise each other and output an object dynamics model, featuring geometry informed by both vision and contact.

Referring to the labeled arrows in Figure 3, we obtain the object trajectory (b) and the initial shape estimates (c) from masked input RGBD images (a) via BundleSDF. The object trajectories are converted to an inertial reference frame, where the table surface (d) identified in the depth images is on a known plane. From this plane (d) and the processed poses (e), PLL detects "physible" portions of the geometry by inferring contact events in the observed dynamics. PLL is also subject to supervision from BundleSDF (f) to encourage consistency with the visible geometry. BundleSDF runs again, fusing both the visible (a) and "physible" data (i) into a geometric model consistent with both information streams. The final output of Vysics inherits the physics-supervised inertial parameters (g) and jointly-supervised geometry (j). This is exported as a URDF which can be simulated to produce dynamics predictions.

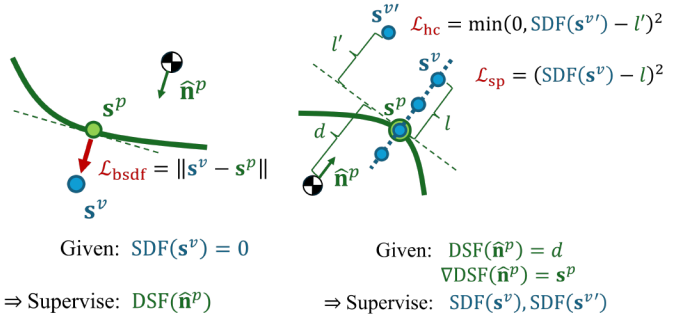
The basis of our contribution is in how we unify the visible and "physible" geometry measurements together. §IV-A discusses how vision helps in the contact learning problem (f), and §IV-B describes how we inject the geometric information estimated from the contact dynamics back into the vision-based shape estimation (i).

A. Supervising Contact-Based Geometry with Vision

Contact-based geometry reconstruction has the ability to learn the full convex hull of any shape as long as every point on its convex hull makes contact with the known environment plane or robot over its recorded trajectory. In extreme low-data regimes where many sides of the geometry never contact the robot or ground, the complete geometry is unobservable, and the solutions of the geometry that can accurately explain the trajectories are non-unique. Since PLL's violation-based implicit loss function has terms that scale with the magnitude of inferred contact forces [53], it is encouraged to find the DSF that minimizes the inferred contact force magnitudes while explaining the trajectory data. In finding a geometry-force combination that explains a net observed torque, PLL can be biased to learn large geometries to increase lever arms and thus reduce force magnitudes. Without intervention, this scenario is common and problematic in low-data, where many sides of the object may never contact the robot or ground. However, vision can provide an additional source of information (denoted by superscript ^{v}) to disambiguate the possible geometries that may explain the trajectory.

Given an RGBD video, BundleSDF yields an object mesh corresponding to the zero SDF level set. We first calculate the convex hull of the mesh since DSF in PLL can only represent convex shapes. Then, we apply a visibility check to preserve only the vertices visible in the RGBD video, and we call this set of points $\mathcal{V}$, the visible geometry.

The visible geometry $\mathcal{V}$ can supervise the surface predicted by PLL's DSF. For each $\mathbf{s}^v \in \mathcal{V}$, we wish to penalize the



(a) Supervising contact-based DSF network with visible geometry $\mathbf{s}^v \in \mathcal{V}$.

(b) Supervising vision-based SDF network with "physible" geometry $(\hat{\mathbf{n}}^p, \mathbf{s}^p) \in \mathcal{P}$.

Fig. 4: Visualization of the loss functions as the incorporation of vision and contact dynamics. Blue represents the geometry learned from vision, and green represents the geometry learned from contact dynamics.

distance from the nearest surface point $\mathbf{s}^p$ predicted by the DSF,

$$\frac{1}{|\mathcal{V}|} \sum_{\mathbf{s}^v \in \mathcal{V}} \|\mathbf{s}^p - \mathbf{s}^v\| . \tag{4}$$

However, it is not straightforward to locate the $\mathbf{s}^p$ surface point on the DSF closest to a given point, since the DSF is an implicit shape representation. To approximate, we sample the DSF to create a mesh and find the point $\mathbf{s}^{p'}$ on its surface that is closest to $\mathbf{s}^v$. The vector

$$\hat{\mathbf{n}}^{p'} = \frac{\mathbf{s}^v - \mathbf{s}^{p'}}{\|\mathbf{s}^v - \mathbf{s}^{p'}\|} \tag{5}$$

is an approximation of the querying vector $\hat{\mathbf{n}}^p$ such that $\nabla\text{DSF}(\hat{\mathbf{n}}^p) = \mathbf{s}^p$. The approximation is up to the angular resolution of the unit vector samples when converting the DSF to a mesh. The direction of the vector $\hat{\mathbf{n}}^{p'}$ is chosen to always point outwards from the shape. In the end, we use

$$\mathcal{L}_{\text{bsdf}} = \frac{1}{|\mathcal{V}|} \sum_{\mathbf{s}^v \in \mathcal{V}} \left\| \nabla\text{DSF}(\hat{\mathbf{n}}^{p'}) - \mathbf{s}^v \right\| \tag{6}$$

as the vision-based supervision of the PLL training. See Figure 4a for an illustration. See Appendix A for hyperparameters, including the relative loss term weights in PLL.

B. Supervising Vision-Based Geometry with Contact

As mentioned in section IV-A, the SDF estimated by BundleSDF may only capture the visible geometry, while PLL estimates a convex shape, in the form of DSF, that best explains the trajectory through contact dynamics. We combine the two streams of information into a single geometry estimation. We use SDF as the final geometry representation, given its capability to model non-convex shapes. The SDF training in the BundleSDF is run on the RGBD video for the second time but with new contact-induced losses from the DSF, as introduced below.

1) From PLL to the "Physible" Geometry: The DSF from PLL can be represented as a set of ∇DSF input/output pairs, $\{(\hat{\mathbf{n}}^p, \mathbf{s}^p)\}$. However, only the local area where contacts happened may be effectively supervised through contact dynamics optimization, while the geometry of other areas is ambiguous with unreliable DSF value. Therefore, we filter the set and only preserve an $(\hat{\mathbf{n}}^p, \mathbf{s}^p)$ pair if it is associated with contact force, estimated by PLL, above a certain threshold. The filtered set, $\mathcal{P}$, is the "physible" geometry output from PLL.

2) Support Point Loss: Given a "physible" geometry data point $(\hat{\mathbf{n}}^p, \mathbf{s}^p)$, not only is $\mathbf{s}^p$ on the object boundary and should have $\text{SDF}(\mathbf{s}^p) = 0$, but we can use $\hat{\mathbf{n}}^p$ to infer constraints on the local geometric region. Consider the ray $\vec{r}$ from $\mathbf{s}^p$ pointing in direction $\hat{\mathbf{n}}^\mathbf{p}$. Any point on this ray, lying a distance $l \in [0, \infty)$ on the ray from $\mathbf{s}$, has a signed distance of exactly l,

$$\text{SDF}\left(\mathbf{s}^p + l\hat{\mathbf{n}}^p\right) = l. \tag{7}$$

See Figure 2 for illustration. For $\mathbf{s}^v = \mathbf{s}^p + l\hat{\mathbf{n}}^p$, we denote $\overline{\text{SDF}}(\mathbf{s}^v) := l$ to represent the signed distance induced from DSF. If we extend the possible range for l values to go negative, i.e. $l \in [-\epsilon, \infty)$, then we can encourage zero-crossings in the SDF network, but the negative signed distance values may not be strictly correct for certain geometries and choices of ϵ. In practice, this can be accurate enough during supervision with small ϵ and enables more meaningful change at the geometry surface. Thus, we introduce a **support point loss** for $\mathbf{s}^v \in \mathcal{P}_{\text{sp}}(\hat{\mathbf{n}}^p, \mathbf{s}^p)$,

$$\mathcal{L}_{\text{sp}} = \left(\overline{\text{SDF}}(\mathbf{s}^v) - \text{SDF}(\mathbf{s}^v)\right)^2, \tag{8}$$

where

$$\mathcal{P}_{\text{sp}}(\hat{\mathbf{n}}^p, \mathbf{s}^p) = \{\mathbf{s}^v | \mathbf{s}^v = \mathbf{s}^p + l\hat{\mathbf{n}}^p, \quad l \in [-\epsilon, \infty)\} \tag{9}$$

is the set of *support points* induced by the "physible" point $(\hat{\mathbf{n}}^p, \mathbf{s}^p)$. $\mathcal{L}_{\text{sp}}$ encourages signed distance consistency at and around observed "physible" points. Figure 4b depicts four such sampled points $\mathbf{s}^v$ and their supervised signed distance values l induced from a single support point $\mathbf{s}^p$.

3) Hyperplane-Constrained Loss: On this ray, (8) imposes strong supervision on the SDF network, though only local to areas near PLL-inferred contacts. However, as depicted in Figure 2, *any* point $\mathbf{q}$ can have its signed distance minimum-bounded based on a support direction/point pair $(\hat{\mathbf{n}}, \mathbf{s})$. Consider a pair $(\hat{\mathbf{n}}^p, \mathbf{s}^p) \in \mathcal{P}$, its supporting hyperplane implies that the signed distance at $\mathbf{s}^{v'}$ can be lower bounded by the distance from $\mathbf{s}^v$ to the supporting hyperplane,

$$\text{SDF}(\mathbf{s}^{v'}) \geq \left(\mathbf{s}^{v'} - \mathbf{s}^p\right) \cdot \hat{\mathbf{n}}^p. \tag{10}$$

Thus, we introduce a **hyperplane-constrained loss** valid for any $\mathbf{s}^{v'} \in \mathbb{R}^3$,

$$\mathcal{L}_{\text{hc}} = \min\left(0, \text{SDF}(\mathbf{s}^{v'}) - (\mathbf{s}^{v'} - \mathbf{s}^p) \cdot \hat{\mathbf{n}}^p\right)^2. \tag{11}$$

Figure 4b gives one example of a point $\mathbf{s}^{v'}$ whose signed distance is lower bounded with respect to a support direction/point (8). While $\mathbf{s}^{v'}$ may be sampled arbitrarily, we

sample them around a cylindrical neighborhood of the support points in practice.

4) Convexity Loss: In comparison to the dense visible points, "physible" points are sparser and can be located far away from the set of visible points. With the assumption that the robot is interacting with a single object at a time, we add a bias loss term to encourage the estimated shape to be convex when no observed RGBD data signals otherwise. This helps the sparse contact points attach to the visible shape in the SDF regression. We randomly sample pairs of points from the near-surface points estimated by the SDF network, $\mathbf{S}^{v_0} = \{\mathbf{s}^v \in \mathbb{R}^3 \mid |\text{SDF}(\mathbf{s}^v)| \leq \epsilon\}$, and the collection of support points (9) induced by all "physible" points $\mathbf{S}^{p_0} = \{\mathbf{s}^v | \mathbf{s}^v \in \mathbf{S}_{\text{sp}}(\hat{\mathbf{n}}^p, \mathbf{s}^p), (\hat{\mathbf{n}}^p, \mathbf{s}^p) \in \mathcal{P}\}$. Points in $\mathbf{S}^{v_0}$ have SDF supervision from the RGBD video, while points in $\mathbf{S}^{p_0}$ have induced $\overline{\text{SDF}}$ supervision from the contact dynamics. Then we sample an interpolated point between each pair of points. The interpolated SDF serves as the upper bound of the SDF prediction at the interpolation point if we assume the shape is convex,

$$\text{SDF}(\mathbf{s}^v) \leq l\ \text{SDF}(\mathbf{s}_1) + (1-l)\ \overline{\text{SDF}}(\mathbf{s}_2), \tag{12}$$

for $\mathbf{s}^v = l\mathbf{s}_1 + (1-l)\mathbf{s}_2, 0 \leq l \leq 1$, where $\mathbf{s}_1 \in \mathbf{S}^{v_0}, \mathbf{s}_2 \in \mathbf{S}^{p_0}$ are samples from the near-surface points and support points. Correspondingly, the **convexity loss** for point $\mathbf{q}$ is,

$$\mathcal{L}_{\text{cvx}} = \max\left(0, \text{SDF}(\mathbf{s}^v) - \left(l\ \text{SDF}(\mathbf{s}_1) + (1-l)\ \overline{\text{SDF}}(\mathbf{s}_2)\right)\right). \tag{13}$$

See Appendix A for hyperparameters used in our experiments, including the relative loss term weights in BundleSDF.

V. Experimental Setup

A. Dataset

We consider a new dataset of RGBD videos of a Franka Emika Panda arm with a spherical end effector interacting with an object repeatedly on a flat table surface. These robot interactions were teleoperated via commanded end effector poses tracked with impedance control. The dataset includes the RGBD videos of the objects in interactions with object mask annotations, as well as the robot joint states. In PLL, the object can collide with the tip of the end effector (modeled as a sphere) and the table surface (modeled as a plane). The end-effector sphere location is known from the robot joint states in combination with the transform between the camera and the robot base, and the height of the table surface is detected automatically based on the depth readings from the camera. The ground truth shapes of the objects are provided in the form of meshes for evaluation. The end-effector pose commands by the teleoperator are also included for the dynamics prediction evaluation (see section V-B). There are substantial visual occlusions preventing the camera from directly seeing much of the object geometry.

The objects in our dataset are a mixture of everyday items depicted in Figure 5. We used a RealSense D455 collecting 640x480 pixel RGBD images at 30Hz. The object masks for a video were semi-automatically generated from manual masks

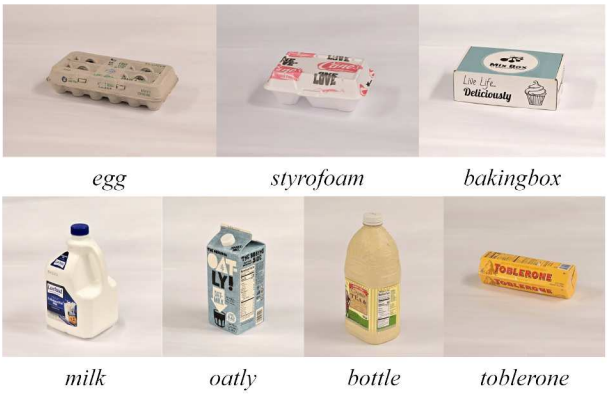


Fig. 5: The 7 objects and their names in our dataset.

on the first frame using XMem [17]. For every object, we collected multiple sessions of the robot arm interacting with the object with its the spherical end effector. Each session is around 10 seconds long. The dataset varies in starting configurations, foreground occlusions, and robot interactions, e.g. sliding vs. pivoting vs. toppling and in what directions. In the evaluation, we excluded the sessions in which BundleSDF lost track of the object and failed to yield the object trajectory. Therefore, the number of sessions for each object vary.

While PLL is capable of identifying friction coefficients essentially by observing acceleration during periods of sliding, sliding motions in our dataset largely occur during sustained object-robot contact, making it difficult to uniquely identify frictional ground contact forces from contact forces with the robot. Thus, we use PLL to learn only geometry and inertia. We set the pair-wise friction coefficients to reasonable values of 0.26 (object-table) and 0.15 (object-robot) for all experiments, higher for object-table because of a high-friction silicone mat on the table.

B. Metrics

We evaluate the performance of the identified geometry and dynamics in two ways. First, we evaluate the geometric error between the estimated shape and the ground truth shape. Specifically, we report the chamfer distance and the volumetric intersection-over-union (IoU) between the learned geometry and the ground truth geometry, which we align with manual annotation and ICP for refinement. This alignment step is only for geometric evaluation, not for training or deployment of the learned model. Second, we perform dynamics predictions to evaluate the geometry through the resulting trajectories. We implemented our impedance controller in simulation and ran it on the recorded end effector pose commands to reproduce the robot's actions. Using the estimated object geometry and its tracked pose at the first frame as the initial condition, we generate simulated trajectories of the object. The predicted trajectories are compared with the real-world trajectory tracked by BundleSDF to show how well the estimated geometry explains the dynamics.

To quantify the dynamics prediction error, we use three

Method	bakingbox	bottle	egg	milk	oatly	styrofoam	toblerone	all
3DSGrasp [46]	3.83	2.80	3.78	3.15	2.51	2.66	2.77	3.06
IPoD [70]	3.25	1.80	2.16	2.37	2.73	1.93	1.97	2.47
V-PRISM [68]	3.52	2.47	2.31	3.33	2.30	2.54	2.48	2.80
OctMAE [30]	3.11	2.22	1.52	2.93	2.13	2.00	2.36	2.45
BundleSDF [66]	3.84	2.65	3.70	3.17	2.45	2.55	2.44	2.98
Vysics (ours)	**1.83**	**1.36**	**1.05**	**1.53**	**1.25**	**1.45**	**1.02**	**1.45**

TABLE I: Average chamfer distance (unit: cm) of shape completion baselines compared with BundleSDF and our method.

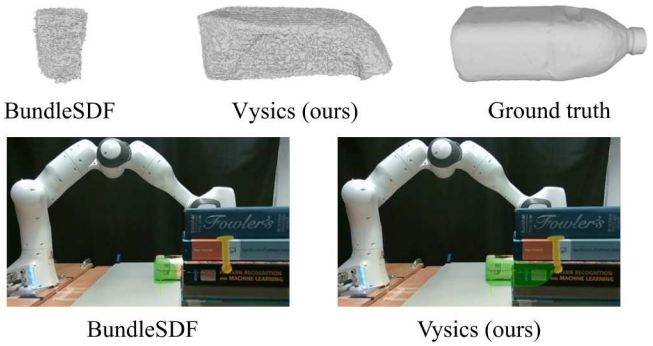
(a) RGB input

(b) OpenLRM [27, 26]

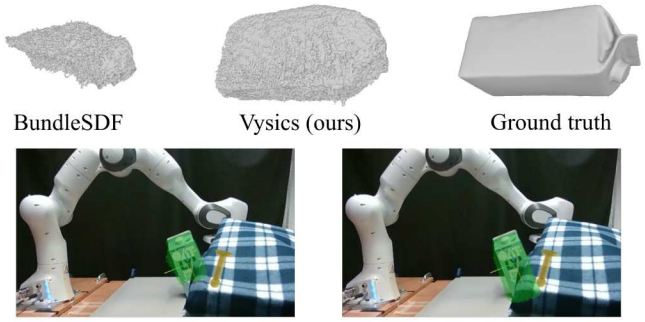
(c) One-2-3-45++ [38]

(d) TriplaneGaussian [76]

Fig. 6: A qualitative example of generative single-view reconstruction on an occluded RGB image of the *egg* object.



(a) An example reconstruction of *bottle* in a pushing interaction.



(b) An example reconstruction of *oatly* in a pivoting interaction.

Fig. 7: A qualitative comparison of the geometry reconstruction under heavy occlusion between our method and the vision-only baseline. In the image view, the mesh projection is shown in green, and the robot end effector is shown in yellow to illustrate the robot-object interaction.

metrics. The first is *average pose error* (split into position and rotation terms), which measures the difference between the tracked pose and the simulated pose throughout the open-loop simulation. As contact dynamics are naturally chaotic, any open-loop simulation is subject to substantial uncertainty. To provide an alternative assessment of the accuracy of the learned model, our second metric is *time-before-divergence*, which measures the time from the start of the simulation to the point when the simulated pose error is higher than a given threshold. We use a positional threshold of 10 cm and a rotational threshold of 45 degrees. The third is the *temporal IoU of contact activation*, which measures the overlap of when the robot-object contact happens in the simulated rollout and in the real experiment. The real contact duration is manually annotated by inspecting the RGBD videos. This metric highlights the compatibility between the physical contact and the estimated geometry. Both average pose error and time before divergence are with respect to the BundleSDF estimated poses.

C. Baselines

Using quantitative geometry reconstruction metrics, we compare against vision-only methods, including BundleSDF and a variety of shape completion methods: 3DSGrasp [46] for point cloud completion (we use points from visible mesh faces reconstructed by BundleSDF as input); IPoD [70] for single-object completion from an RGBD image; and V-PRISM [68] and OctMAE [30] for multi-object scene completion from an RGBD image. All of these methods, like Vysics, are category-agnostic. For shape completion methods based on an RGBD image, we use the last frame of each video clip, which, in our data, is typically less occluded. We provide object masks for IPoD and foreground masks for V-PRISM and OctMAE, then segment the object from their scene completion outputs. We use published pretrained models for evaluation. We additionally provide a qualitative comparison against methods of 3D reconstruction from a single RGB image: OpenLRM [26, 27], One-2-3-45++ [38], and Triplane Gaussian [76], to see if these models trained with large-scale 2D and 3D data can recover the complete shape from a partial view.

For dynamics predictions, we compare against fair variations of BundleSDF and PLL. The BundleSDF variation features the geometry from BundleSDF and inertia centered on its geometric centroid. We explored instead using the inertia learned by PLL in combination with the BundleSDF geometry, but this often results in unrealistic dynamics predictions due to the PLL-learned center of mass landing outside the BundleSDF geometry's convex hull. The PLL variation is the result of running PLL on the BundleSDF pose estimates plus vision supervision via (6); this is the same as Vysics without the second round of BundleSDF.

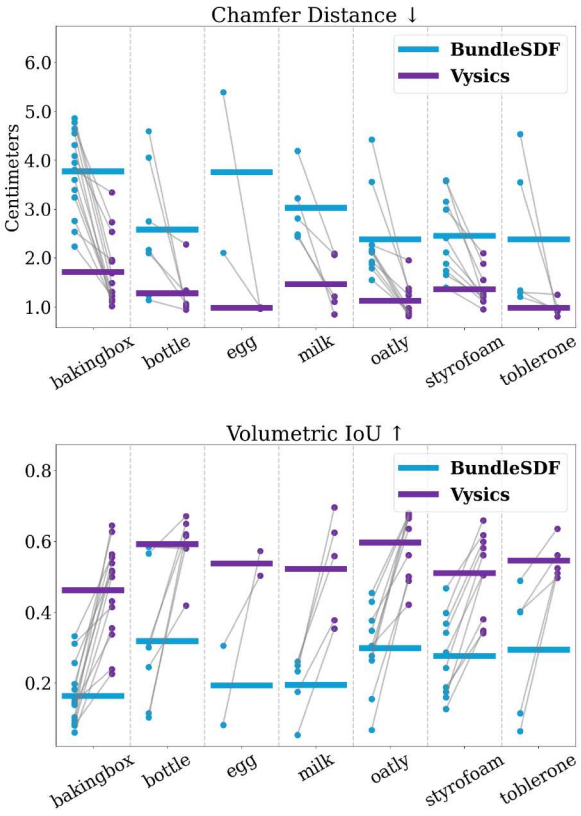


Fig. 8: The quantitative comparison of the geometric reconstruction accuracy. Each dot is one session. The results of the same session from different methods are connected by a gray line. ↑ means higher is better. ↓ means lower is better.

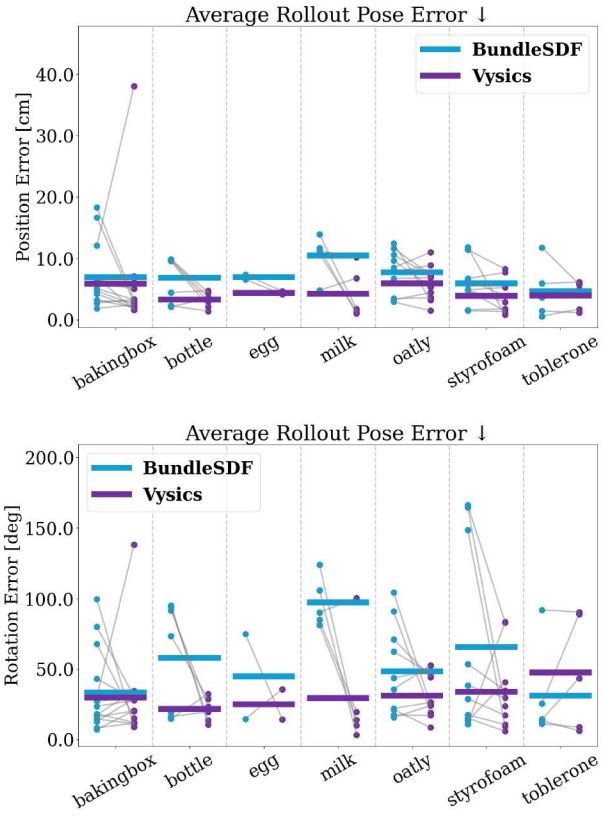


Fig. 9: The quantitative comparison of the dynamics prediction accuracy in pose error. Trajectories are predicted by replaying the robot interaction with the estimated geometry in simulation.

VI. Results

A. Geometry Reconstruction

We first compare the geometry reconstruction of our method with that of shape completion models and single-view 3D generation models. Table I presents the quantitative results of shape completion models in chamfer distance, averaged per object and over all objects, compared with BundleSDF and Vysics. Under severe occlusion, while the shape completion models can achieve similar or slightly lower chamfer distance than pure vision-based reconstruction, BundleSDF, they fall behind Vysics by a large margin, showing that the data-driven completion models are not as successful as Vysics at filling in the missing pieces. Qualitative results of the single-view 3D generation models are shown in Figure 6. We find that these generative models typically assume an unobstructed view of the object and do not generate a complete shape when given a partially occluded view. Therefore, we do not evaluate these models quantitatively.

In the following, we provide detailed comparisons between Vysics and BundleSDF, as they both do not require a pre-trained model on a large object dataset. Qualitative comparisons of Vysics and BundleSDF geometric reconstruction is shown in Figure 7. In these examples, the robot arm touches an occluded portion of the geometry and causes motion of the object. The view of the object is heavily occluded by the obstacles (books/blanket), resulting in BundleSDF missing a significant portion of the object in its geometry estimate. In contrast, our method recovers the occluded geometry so that the robot arm's interactions with the objects can explain the observed object trajectory. Figure 8 shows the quantitative results of geometric reconstruction for each individual session in terms of the surface-based metric, chamfer distance, and the volume-based metric, IoU. Our method recovers the occluded geometry through physics-based reasoning over the observed trajectories, substantially and consistently improving the geometric accuracy in both metrics.

B. Dynamics Predictions

We further use dynamics predictions to show that the geometry estimated by our method better explains the observed trajectory. Figure 9 shows the average position and rotation errors when predicting the entire length of the original trajectory as an open-loop rollout. The trajectories range in length from 3 to 18 seconds. Compared with the shape estimated by the vision-only method, the geometry estimated by our method enabled much smaller simulation pose errors across

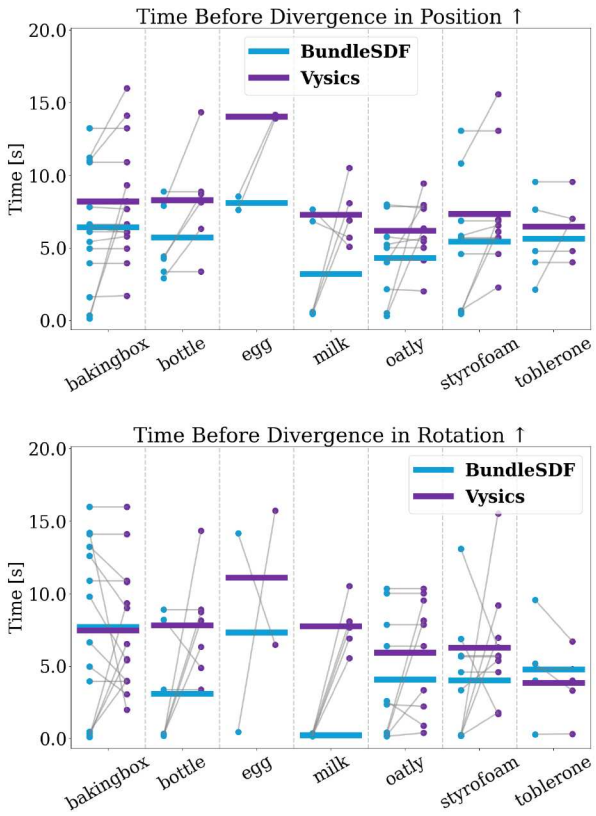


Fig. 10: The dynamics prediction accuracy evaluated by the duration of the simulated trajectory under small pose error.

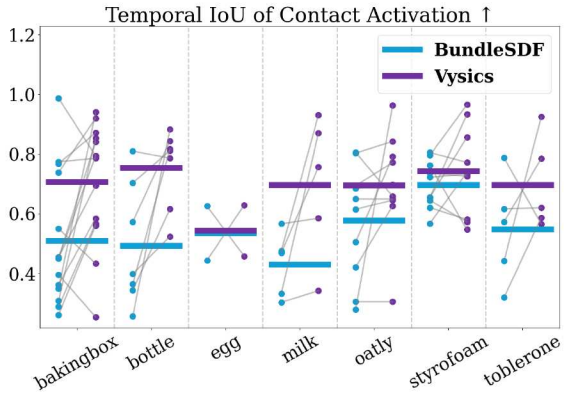


Fig. 11: The dynamics prediction accuracy evaluated by the temporal overlap of robot-object contact happening in simulation and in the real world. The real world ground truth was manually annotated.

the dataset.

A similar trend can be found in the time-before-divergence metric in Figures 10 and 12. Our method maintained a small pose error for a longer time than the vision-only baseline in simulations. Figure 12 compares Vysics against the vision-only baseline, physics-only baseline, and a baseline featuring

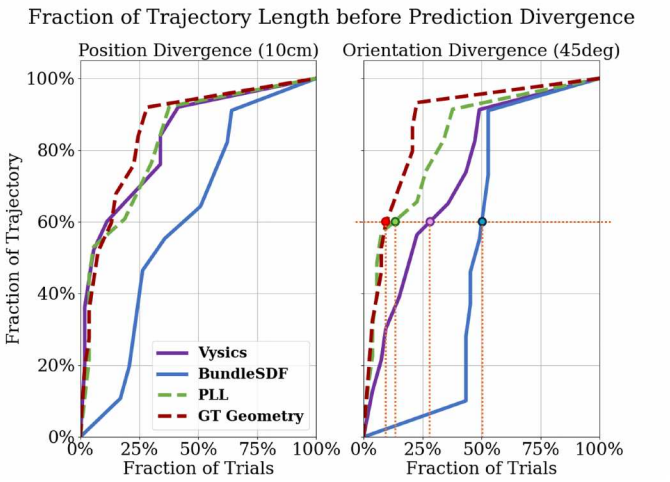


Fig. 12: For quantifying dynamics prediction performance, we compare how far into an open-loop rollout the predicted pose stays within 10cm of position error and within 45 degrees of rotational error from the BundleSDF tracked poses. We normalize the y-axis to the length of the trajectory. An example interpretation from the right plot (the orange dashed lines): 60% into the predicted trajectory, approximately half of the BundleSDF dynamics predictions diverged in orientation compared to 30% of the Vysics dynamics predictions and 10% of the ground truth geometry and PLL predictions.

a simulation with the ground truth geometry. As expected, the ground truth geometry simulations maintained more accurate dynamics predictions for the longest. We point out that even this baseline is imperfect, despite using essentially perfect geometry, due to inaccurate modeling assumptions such as object rigidity and the divergent nature of the dynamics in many of our robot interactions. Vysics and PLL perform closely to this baseline, though Vysics is moderately worse in orientation divergence. While most of the dynamics performance by PLL is retained in Vysics, it is unsurprising to see a slight performance drop, given PLL optimizes only for physics accuracy while Vysics balances with visual objectives. The vision-only baseline is the least performant in both position and orientation rollout accuracy.

The temporal IoU of contact activation is shown in Figure 11. Our method achieved better temporal consistency between the observed contact and the contact reproduced in the simulations than the vision-based geometry. Despite the chaotic nature of open-loop simulation, our method achieved consistent improvements in all three dynamics prediction metrics across the dataset, which shows the capability of our method in building better object models by exploiting the vision and physics information.

VII. Limitations and Future Work

A limitation of Vysics is that it does not incorporate notions of object elasticity or bounciness into the learning problem. This shortcoming means the dynamics predictions of our learned models could deviate drastically from the ground-truth

trajectories, which saw more chaotic bouncing and slower energy dissipation. While automated searches for simulator contact parameters exist [2], preliminary attempts to improve our dynamics prediction accuracy with these methods did not yield significant improvements over our inelastic contact model.

A considerable limitation of our method in its current form is that it often cannot recover from poor pose estimates. In our experience, this issue is accentuated by the challenges of our occlusion-rife, contact-rich dataset. We found shorter video lengths usually resulted in more consistent pose tracking, but this is at odds with the benefits of more data for dynamics parameter regression. It is possible that the geometry supervision from PLL during BundleSDF's second run can help it perform better pose estimation. In this case, we could cyclically repeat our BundleSDF-PLL process until both shape and trajectories converge. Future versions of Vysics may consider posing a single joint learning problem that performs pose estimation and dynamics model building simultaneously.

The scarcity of observation fundamentally limits the quality of the reconstructed shape from our method. The visual data is heavily occluded, while the contact signal is sparse by nature, leaving limited information for our method to optimize over. Therefore, readers may find the reconstructed shapes qualitatively unappealing, with noisy artifacts and limited details. A potential future direction is to leverage the data-based priors in foundation models to guide shape estimation while respecting visual and physics signals.

Lastly, our experiments featured robot interactions from teleoperation with diversity of interactions in mind. In a closed-loop system, it could be possible for explorative strategies to determine where the robot should initiate contact to lower its uncertainty about the object's properties. While this interactive approach is compatible with Vysics as designed, this remains future work.

VIII. Conclusion

Vysics equips robots with the ability to construct high-fidelity dynamics models of novel objects, identified from vision and proprioception, in the face of contact-rich interactions and extremely small amounts of data. This is the first step toward unifying vision-based geometry estimation with contact dynamics. Beyond improvements to the core method of Vysics, discussed in §VII, future work might replace teleoperated data collection with autonomous, active exploration or the integration of these learned models with planning and control to accomplish some desired task.

Acknowledgments

We thank our anonymous reviewers, who provided thorough and fair feedback that improved the quality of our paper. This work was supported by a National Defense Science and Engineering Graduate (NDSEG) Fellowship, an NSF CAREER Award under Grant No. FRR-2238480, and the RAI Institute.

References

[1] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Suenderhauf. Physically embodied gaussian splatting: A realtime correctable world model for robotics. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=AEq0onGrN2.

[2] Brian Acosta, William Yang, and Michael Posa. Validating robotics simulators on real-world impacts. *IEEE Robotics and Automation Letters*, 7(3):6471–6478, 2022.

[3] William Agnew, Christopher Xie, Aaron Walsman, Octavian Murad, Yubo Wang, Pedro Domingos, and Siddhartha Srinivasa. Amodal 3d reconstruction for robotic manipulation via stability and connectivity. In *Conference on Robot Learning*, pages 1498–1508. PMLR, 2021.

[4] Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

[5] Rika Antonova, Jingyun Yang, Krishna Murthy Jatavallabhula, and Jeannette Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on Robot Learning*, pages 276–286. PMLR, 2023.

[6] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022.

[7] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan M Shetty. *Nonlinear programming: theory and algorithms*. John wiley & sons, 2013.

[8] Bibit Bianchini, Mathew Halm, Nikolai Matni, and Michael Posa. Generalization bounded implicit learning of nearly discontinuous functions. In *Learning for Dynamics and Control Conference*, pages 1112–1124. PMLR, 2022.

[9] Bibit Bianchini, Mathew Halm, and Michael Posa. Simultaneous learning of contact and continuous dynamics. In *Conference on Robot Learning*, pages 3966–3978. PMLR, 2023.

[10] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2560–2568, 2018.

[11] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[12] Pedro Castro and Tae-Kyun Kim. Posematcher: One-shot 6d object pose estimation by deep feature matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2148–2157, 2023.

[13] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint*

arXiv:1512.03012, 2015.

[14] Tao Chen, Megha Tippur, Siyang Wu, Vikash Kumar, Edward Adelson, and Pulkit Agrawal. Visual dexterity: In-hand reorientation of novel and complex object shapes. *Science Robotics*, 8(84):eadc9244, 2023.

[15] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 139–156. Springer, 2020.

[16] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13581–13590, 2023.

[17] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022.

[18] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4456–4465, 2023.

[19] Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Nießner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion. *Advances in neural information processing systems*, 36: 75951–75966, 2023.

[20] Filipe de Avila Belbute-Peres, Kevin Smith, Kelsey Allen, Josh Tenenbaum, and J Zico Kolter. End-to-end differentiable physics for learning and control. *Advances in neural information processing systems*, 31:7178–7189, 2018.

[21] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023.

[22] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[23] Yan Di, Ruida Zhang, Zhiqiang Lou, Fabian Manhardt, Xiangyang Ji, Nassir Navab, and Federico Tombari. Gpv-pose: Category-level object pose estimation via geometry-guided point-wise voting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6781–6791, 2022.

[24] Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Learning data-efficient rigid-body contact models: Case study of planar impact. In *Conference on Robot Learning*, pages 388–397. PMLR, 2017.

[25] Mathew Halm. *Addressing Stiffness-Induced Challenges in Modeling and Identification for Rigid-Body Systems With Friction and Impacts*. PhD thesis, University of Pennsylvania, 2023.

[26] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023.

[27] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

[28] Tingbo Hou, Adel Ahmadyan, Liangkai Zhang, Jianing Wei, and Matthias Grundmann. Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision. *arXiv preprint arXiv:2003.03522*, 2020.

[29] Taylor A Howell, Simon Le Cleac'h, J Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 2022.

[30] Shun Iwase, Katherine Liu, Vitor Guizilini, Adrien Gaidon, Kris Kitani, Rareş Ambruş, and Sergey Zakharov. Zero-shot multi-object scene completion. In *European Conference on Computer Vision*, pages 96–113. Springer, 2024.

[31] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. *International Conference on 3D Vision (3DV)*, 2024.

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

[33] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020.

[34] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. *arXiv preprint arXiv:2212.06870*, 2022.

[35] Simon Le Cleac'h, Mac Schwager, Zachary Manchester, Vikas Sindhwani, Pete Florence, and Sumeet Singh. Single-level differentiable contact simulation. *IEEE Robotics and Automation Letters*, 2023.

[36] Yunzhi Lin, Jonathan Tremblay, Stephen Tyree, Patricio A Vela, and Stan Birchfield. Keypoint-based category-level object pose tracking from an rgb sequence with uncertainty estimation. In *2022 International*

Conference on Robotics and Automation (ICRA), pages 1258–1264. IEEE, 2022.

[37] Stefan Lionar, Xiangyu Xu, Min Lin, and Gim Hee Lee. Nu-mcc: Multiview compressive coding with neighborhood decoder and repulsive udf. *Advances in Neural Information Processing Systems*, 36:63011–63022, 2023.

[38] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10072–10083, 2024.

[39] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[40] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images. In *European Conference on Computer Vision*, pages 298–315. Springer, 2022.

[41] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 international conference on 3D vision (3DV)*, pages 32–41. IEEE, 2018.

[42] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2023.

[43] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[44] 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.

[45] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 306–315, 2022.

[46] Seyed S Mohammadi, Nuno F Duarte, Dimitrios Dimou, Yiming Wang, Matteo Taiana, Pietro Morerio, Atabak Dehban, Plinio Moreno, Alexandre Bernardino, Alessio Del Bue, et al. 3dsgrasp: 3d shape-completion for robotic grasp. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3815–3822. IEEE, 2023.

[47] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020.

[48] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Yinlin Hu, Renaud Marlet, Mathieu Salzmann, and Vincent Lepetit. NOPE: Novel Object Pose Estimation from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[49] Junfeng Ni, Yixin Chen, Bohan Jing, Nan Jiang, Bin Wang, Bo Dai, Puhao Li, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Phyrecon: Physically plausible neural scene reconstruction. 2024.

[50] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[51] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.

[52] Mihir Parmar, Mathew Halm, and Michael Posa. Fundamental challenges in deep learning for stiff contact dynamics. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5181–5188. IEEE, 2021.

[53] Samuel Pfrommer, Mathew Halm, and Michael Posa. ContactNets: Learning Discontinuous Contact Dynamics with Smooth, Implicit Representations. In *The Conference on Robot Learning (CoRL)*, 2020. URL https://proceedings.mlr.press/v155/pfrommer21a.html.

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[55] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3069, 2018.

[56] Changkyu Song and Abdeslam Boularias. Inferring 3d shapes of unknown rigid objects in clutter through inverse physics reasoning. *IEEE Robotics and Automation Letters*, 4(2):201–208, 2018.

[57] Manuel Stoiber, Martin Sundermeyer, and Rudolph Triebel. Iterative corresponding geometry: Fusing region and depth for highly efficient 3d tracking of textureless objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6855–6865, 2022.

[58] Edgar Sucar, Kentaro Wada, and Andrew Davison.

Nodeslam: Neural object descriptors for multi-view shape reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 949–958. IEEE, 2020.

[59] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou. Onepose: One-shot object pose estimation without cad models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6825–6834, 2022.

[60] Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, et al. Neural feels with neural fields: Visuo-tactile perception for in-hand manipulation. *arXiv preprint arXiv:2312.13469*, 2023.

[61] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[62] Anh Thai, Stefan Stojanov, Vijay Upadhya, and James M Rehg. 3d reconstruction of novel object shapes from single images. In *2021 International Conference on 3D Vision (3DV)*, pages 85–95. IEEE, 2021.

[63] Mark Van der Merwe, Qingkai Lu, Balakumar Sundaralingam, Martin Matak, and Tucker Hermans. Learning continuous 3d reconstructions for geometrically aware grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11516–11522. IEEE, 2020.

[64] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066. IEEE, 2020.

[65] Bowen Wen and Kostas Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models, 2021.

[66] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 606–617, 2023.

[67] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. *arXiv preprint arXiv:2312.08344*, 2023.

[68] Herbert Wright, Weiming Zhi, Matthew Johnson-Roberson, and Tucker Hermans. V-prism: Probabilistic mapping of unknown tabletop scenes. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1078–1085. IEEE, 2024.

[69] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9065–9075, 2023.

[70] Yushuang Wu, Luyue Shi, Junhao Cai, Weihao Yuan, Lingteng Qiu, Zilong Dong, Liefeng Bo, Shuguang Cui, and Xiaoguang Han. Ipod: Implicit field learning with point diffusion for generalizable 3d object reconstruction from single rgb-d images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20432–20442, 2024.

[71] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pages 1–20. Springer, 2024.

[72] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022.

[73] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.

[74] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision*, pages 592–611. Springer, 2022.

[75] Weiming Zhi, Haozhan Tang, Tianyi Zhang, and Matthew Johnson-Roberson. Simultaneous geometry and pose estimation of held objects via 3d foundation models. *IEEE Robotics and Automation Letters*, 2024.

[76] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10324–10335, 2024.

APPENDIX

A. BundleSDF Hyperparameters

Description	Symbol in [66]	Value
RGB loss weight	w_c	100
Uncertain free space loss weight	w_u	50
Uncertain free space SDF target	ϵ	0.1
Empty space loss weight	w_e	1
Near-surface space loss weight	w_{surf}	3,000
Truncation distance of the near-surface space	N/A	0.01 m
SDF target corresponding to the truncation distance	λ	1
Eikonal loss weight	w_{eik}	0
(Ours) Support point loss weight on (8)	N/A	2
(Ours) Hyperplane-constrained loss weight on (11)	N/A	1
(Ours) Convexity loss weight on (13)	N/A	1

TABLE II: Hyperparameters for modified BundleSDF [66].

BundleSDF [66] employed five loss terms to optimize the object SDF and poses:

- RGB loss: a penalty on the discrepancy between the color rendered from the SDF and object pose and the observed RGB value.
- Uncertain free space loss: encourages the SDF to predict a small constant in regions near observed object surfaces. This preserves the shape's contour while being able to quickly adapt to more reliable observations.
- Empty space loss: encourages the foreground empty space in front of the depth readings up to a truncation distance to predict the constant truncation SDF.
- Near-surface loss: for near-surface points with distance to the depth readings under the truncation distance, encourages the SDF to be predicted exactly.
- Eikonal loss: encourages the regressed SDF to have a gradient of magnitude 1 at any point. While described in [66], this loss term has a weight of 0 in BundleSDF's official implementation.

The weights for the loss terms introduced in Vysics are listed in Table II.

B. PLL Hyperparameters

Description	Symbol in [9]	Value
Prediction loss weight	w_{pred}	4
Complementarity loss weight	w_{comp}	1
Dissipation loss weight	w_{diss}	5000
Penetration loss weight	w_{pen}	0.5
(Ours) BundleSDF loss weight on (6)	N/A	0.04

TABLE III: Loss term weights for modified PLL [9, 53].

PLL [9, 53] features a violation-based implicit loss function with 4 main components:

- Prediction loss: a penalty on the difference between the dynamics model's predicted next state (using hypothesized contact forces) and the observed next state.
- Complementarity loss: a penalty on violation of contact complementarity, i.e. penalize any predicted contact forces if the geometry model does not predict contact.
- Dissipation loss: a penalty on violation of the Coulomb friction model.
- Penetration loss: a penalty on the penetration of the current geometric model into the known environment surface at the end of the time step.

PLL learns dynamics models as a bilevel optimization problem: 1) Per epoch, the hypothesized set of contact forces are selected to minimize the sum of the above terms. 2) Between epochs, the dynamics model parameters update via gradient descent. See [53] for more details. We add (6) as a term directly to the above from PLL with the loss weights according to Table III.

C. Other Vysics Hyperparameters

1) Hyperparameters for converting from PLL outputs to modified BundleSDF inputs: §IV-B describes querying the PLL geometry for supervising BundleSDF via (8) and (11). Since odometry-based contact learning may lack signal in many directions, data for supervising the next round of BundleSDF is extracted by running all of the input trajectory data again (after PLL has already been trained) through PLL's violation-based implicit loss. Vysics retains only the queried points with the top fraction of hypothesized contact normal forces – we used the top 30%. Thus, we yield hypothesized-force-filtered data points $\{(\hat{\mathbf{n}}^p, \mathbf{s}^p)_i\}$ that confidently saw contact during PLL training.

We go from one DSF input/output pair $(\hat{\mathbf{n}}^p, \mathbf{s}^p)$ to many SDF inputs/outputs $\{(\mathbf{p}, l)_j\}$ subject to (8) by sampling on the ray $\vec{r}$ defined by $(\hat{\mathbf{n}}^p, \mathbf{s}^p)$:

- 100 points randomly sampled up to 5mm from $\mathbf{s}^p$ in the $-\hat{\mathbf{n}}^p$ direction (into the convex hull),
- 100 points randomly sampled up to 5mm from $\mathbf{s}^p$ in the $+\hat{\mathbf{n}}^p$ direction, and
- 50 points randomly sampled up to 10cm from $\mathbf{s}^p$ in the $+\hat{\mathbf{n}}^p$ direction.

Per epoch, BundleSDF randomly picks up to 1K of these SDF input/output pairs.

For obtaining SDF inputs/lower bounds $\{(\mathbf{q}, l_{\text{min}})_k\}$ subject to (11), we first sample 25K points evenly-spaced on the PLL mesh. We filter out any points that are more than 1mm away from a supporting hyperplane defined by the hypothesized-force-filtered support points and directions $\{(\hat{\mathbf{n}}^p, \mathbf{s}^p)_i\}$. This yields a set of points sampled on the PLL learned mesh plus their associated face's outward normal, $\{(\hat{\mathbf{m}}, \mathbf{v})_n\}$. Sampling on the mesh instead of using the support points and directions directly yields supervision over a broader portion of the geometry even after filtering based on supporting hyperplane proximity, since the support points themselves are sparse and can be highly local. We go from one direction/point pair $\{(\hat{\mathbf{m}}, \mathbf{v})\}$ to many SDF inputs/lower bounds $\{(\mathbf{q}, l_{\text{min}})_k\}$ by sampling:

- 200 points randomly sampled within a cylinder of radius 5cm, starting at $\mathbf{v}$ with its axis extending 5mm along the $-\hat{\mathbf{m}}$ direction,
- 100 points randomly sampled within a cylinder of radius 5cm, starting at $\mathbf{v}$ with its axis extending 5mm along the $+\hat{\mathbf{m}}$ direction, and
- 100 points randomly sampled within a cylinder of radius 10cm, starting at $\mathbf{v}$ with its axis extending 10cm along the $+\hat{\mathbf{m}}$ direction.

Per epoch, BundleSDF randomly picks up to 5K of these input/SDF limit pairs.

2) Hyperparameters for converting from BundleSDF outputs to modified PLL inputs: §IV-A describes querying the BundleSDF geometry to obtain DSF input/output pairs $\{(\hat{\mathbf{n}}^v, \mathbf{s}^v)_i\}$ for supervising PLL via (6). We query 5,768 approximately evenly-spaced $\hat{\mathbf{n}}^v$ directions and obtain their corresponding support points $\mathbf{s}^v$ on the BundleSDF geometry. Per epoch, PLL randomly picks up to 1K of these points to incorporate into its loss via (6).