Structure from Collision

Takuhiro Kaneko
NTT Corporation

Abstract

Recent advancements in neural 3D representations, such as neural radiance fields (NeRF) and 3D Gaussian splatting (3DGS), have enabled the accurate estimation of 3D structures from multiview images. However, this capability is limited to estimating the visible external structure, and identifying the invisible internal structure hidden behind the surface is difficult. To overcome this limitation, we address a new task called Structure from Collision (SfC), which aims to estimate the structure (including the invisible internal structure) of an object from appearance changes during collision. To solve this problem, we propose a novel model called SfC-NeRF that optimizes the invisible internal structure of an object through a video sequence under physical, appearance (i.e., visible external structure)-preserving, and keyframe constraints. In particular, to avoid falling into undesirable local optima owing to its ill-posed nature, we propose volume annealing; that is, searching for global optima by repeatedly reducing and expanding the volume. Extensive experiments on 115 objects involving diverse structures (i.e., various cavity shapes, locations, and sizes) and material properties revealed the properties of SfC and demonstrated the effectiveness of the proposed SfC-NeRF.[1]

1. Introduction

Learning 3D representations from multiview images is a fundamental problem in computer vision and graphics, with applications across various domains, including augmented and virtual reality, gaming, robotics, and autonomous driving. Recent advancements in neural 3D representations, such as neural radiance fields (NeRF) [47] and 3D Gaussian splatting (3DGS) [32], have enabled the accurate estimation of 3D structures from multiview images and yielded impressive results in novel view synthesis.

However, this benefit is limited to the estimation of the *visible external* structure, and it remains difficult to estimate the *invisible internal* structure hidden behind the surface.[2]

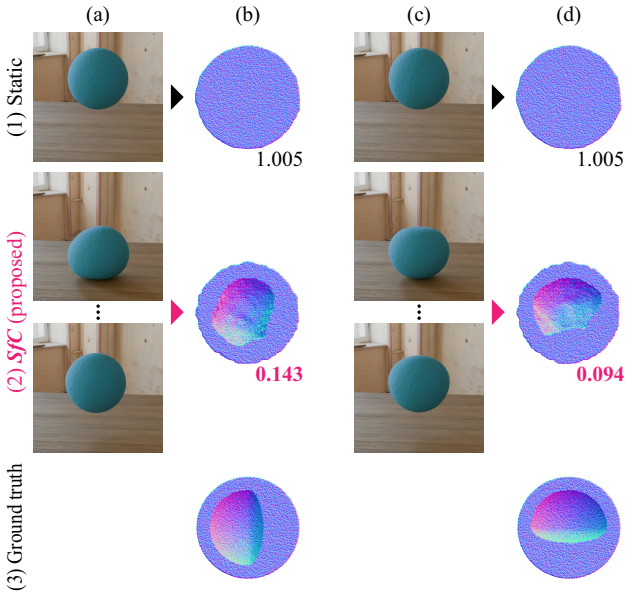

Figure 1. Concept of *Structure from Collision (SfC)*. (a) and (c) Examples of training images taken from a certain viewpoint. (b) and (d) Cross-sectional views of the internal structures cut perpendicular to the viewpoint. The score indicates the chamfer distance ($\times 10^3\downarrow$) between the ground-truth and estimated particles (the smaller, the better). Here, two objects appear to be identical in static images (1) but actually have different internal structures (3). (1) A static 3D representation learning model cannot distinguish the difference in internal structures (b)(d) because there is no difference in appearance in static images (a)(c). (2) To overcome this limitation, we address *SfC*. As shown in (a) and (c), changes in shape and appearance during collision are influenced by the internal structure. We utilize this property to identify the internal structure of the object. Although it is still difficult to identify perfectly owing to its ill-posed nature, the proposed method has succeeded in capturing the bias in the location of the holes (b)(d).

For example, in Figure 1, the two objects have different internal structures, as shown in Figure 1(3)(b) and (3)(d). However, they are identical in the static images, as shown in Figure 1(1)(a) and (1)(c). Consequently, a standard static neural 3D representation learning model (e.g., the voxel-based NeRF [63] used in this example) learns the same in-

[1]The project page is available at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/sfc/.

[2]More strictly, when an object is transparent or translucent, it is possible to estimate the internal structure hidden behind the surface using a volume rendering-based 3D representation learning model (e.g., NeRF [47]) because it represents appearance on the basis of cumulative volume densities. However, this effect is limited when an object is not transparent. This study aims to identify the internal structure even in the latter case.

ternal structures (Figure 1(1)(b) and (1)(d)) and ignores the differences in the internal structures. This misestimation of the internal structure can cause issues in practical applications, such as reproducing and simulating objects in virtual and augmented reality and controlling forces during interactions with objects in robotics.

To overcome this limitation, we address a novel task called *Structure from Collision (SfC)*, the objective of which is to identify the structure (including the invisible internal structures) of an object based on observations at collision. This is motivated by the observation that changes in appearance and shape during collisions are influenced by the internal structures. For example, as shown in Figure 1(2)(a) and (2)(c), when a hole exists inside the sphere on the left side (Figure 1(3)(b)) or on the upper side (Figure 1(3)(d)), the sphere crumples when it hits the ground. We use this property to identify the internal structure of the object.

We formulated *SfC* to optimize the invisible internal structures of an object under *physical*, *appearance (i.e., visible external structure)-preserving*, and *keyframe constraints*. Specifically, we implemented this approach using *SfC-NeRF*, which consists of four components.

(1) Physical constraints. *SfC* is ill-posed because the observable data represent just one of many possible solutions. To address this issue, we narrow the solution space by incorporating *physical constraints*, specifically by using physics-augmented continuum NeRF (PAC-NeRF) [36].

(2) Appearance-preserving constraints. Owing to the recent advancements in neural 3D representations, learning *visible external* structures is easier than learning *invisible internal structures*. Accordingly, we first learn the external structures using a standard static neural 3D representation learning model (voxel-based NeRF [63] in practice) using the first frame (Figure 1(1)). We then optimize the internal structures using a video sequence (Figure 1(2)). In the second step, to avoid damaging the external structures learned in the first step when fitting the entire video, we introduce *appearance-preserving constraints* that optimize the internal structures while preserving the external structures.

(3) Keyframe constraints. In a collision video, a specific frame (e.g., immediately after a collision) is effective in explaining the shape change caused by the collision. Accordingly, we incorporate *keyframe constraints* to strengthen shape learning in the keyframe.

(4) Volume annealing. To avoid becoming stuck in undesirable local optima owing to the existence of multiple solutions, we developed *volume annealing*, in which the global optimum is searched for through an annealing process that repeatedly reduces and expands the volume.

We comprehensively evaluated the proposed method using a dataset containing 115 objects with diverse structures (i.e., various cavity shapes, locations, and sizes) and material properties. Our results reveal the properties of *SfC* and demonstrate the effectiveness of *SfC-NeRF*. Figure 1(2)(b) and (d) show examples of the results obtained using *SfC-NeRF*. Although it is challenging to perfectly match the internal structures to the ground truth owing to the high degrees of freedom in the solution, *SfC-NeRF* successfully identified the deviation of the hole inside the sphere.

The contributions of this study are threefold:

- We address a novel task called *SfC*, whose aim is to identify structures (including the internal structures) from the appearance changes at collision.
- To solve *SfC*, we propose *SfC-NeRF*, which consists of four components: *physical*, *appearance-preserving*, and *keyframe constraints*, and *volume annealing*.
- Through extensive experiments on 115 objects, we demonstrate the effectiveness of *SfC-NeRF* while clarifying the properties of *SfC*. We also provide detailed results and implementation details in the Appendices. Video samples are available at the project page.

2. Related work

Neural 3D representations. Learning 3D representations is a fundamental problem in computer vision and graphics. Recent advancements in neural 3D representations, such as NeRF [47] and 3DGS [32], have lead to significant breakthroughs, and various derivative models have been proposed. These models can be roughly divided into three categories, based on their objectives. *(1) Improvement of quality* of rendered images or reconstructed 3D data [4–6, 24, 27, 37, 39, 43, 48, 67, 74, 79, 80], *(2) improvement of efficiency*, i.e., speeding up and reducing memory usage in training or inference [3, 10, 12, 16, 19, 22, 23, 30, 34, 35, 42, 44, 49–51, 57, 58, 60, 62, 63, 68, 71, 78], and *(3) incorporation of other modules or functionalities*, such as generative models [7–9, 11, 14, 18, 20, 29, 40, 52, 54, 59, 61, 64, 65, 70, 73, 77, 81] and physics/dynamics [1, 2, 13, 15, 17, 21, 28, 31, 36, 38, 45, 46, 53, 55, 56, 66, 72, 75, 76]. This study focuses on the third category, aiming to discover internal structures based on dynamic observations under physical constraints. Because these models are mutually developed, applying the proposed approach to other models presents an interesting direction for future research.

Dynamic neural 3D representations. Dynamic neural 3D representations can be classified into two categories, based on whether they incorporate physics. *(1) Non- (or weak) physics-informed models* [17, 38, 45, 46, 53, 55, 66, 75, 76] and *(2) physics-informed models* [1, 2, 13, 15, 21, 28, 31, 36, 56, 72]. The first category offers flexibility, and can be applied to scenes or objects that are difficult to describe physically. However, it requires a large amount of training data and lacks interpretability because of its fully data-driven black-box nature. By introducing physics, the second category provides a better interpretability and narrows the solution space. However, they lose flexibility and are difficult to apply to scenes or objects that cannot be explained by physics. This study adopts a physics-informed model (the second-category strategy) because *SfC* is an ill-posed problem, and physics plays an important role in nar-

rowing the solution space. However, in the future, it would be interesting to explore how the first-category strategy can be used by expanding data and developing new theories.

Physics-informed neural 3D representations. Physics-informed neural 3D representations can be divided into two categories based on the problem setting. *(1) Forward engineering* [15, 28, 56, 72], where a physics-informed model is optimized to fit *static* scenes or objects, and then physics-informed dynamic simulations or interactive manipulations are performed. In most cases, the inside of the object is assumed to be *filled*, and internal factors, such as physical properties, are *manually adjusted* to achieve visually plausible results. *(2) Reverse engineering* [1, 2, 13, 21, 31, 36], which focuses on system identification—identifying internal factors (e.g., physical properties) from *dynamic* observations (i.e., video sequences). This study falls into the second category because it aims to reverse engineer the *internal structure*, which is hidden but essential for describing the system, from collision videos.

Reverse engineering is generally ill-posed because the observable data represent only one of the many possible solutions. To address this issue, the methods in this category typically impose assumptions on internal factors that are not optimized. Previous studies have made various assumptions regarding the internal structure, which is the main focus of this study. For example, [13] assumes that an object, such as smoke, is *translucent*, allowing *part of the internal structure to be visible*. Other studies [1, 2, 21, 31, 36] considered *non-transparent* objects but assumeed that the interior is *filled*. Consequently, *non-transparent and unfilled objects* have not been sufficiently explored. Therefore, this study focused on such objects. It is important to note that, as with conventional problems, solving *SfC* is challenging without making any assumptions. In this study, we assumed that certain internal factors, such as physical properties, are known in advance. Even with this assumption, as shown in Figure 1 (where physical properties, such as mass, Young's modulus, and density, are identical), multiple solutions still exist, making *SfC* a challenging problem. Details of the problem settings are discussed in Section 3.1.

3. Method

3.1. Problem statement

First, we define the *SfC* problem. Given a set of multi-view videos in which objects collide (e.g., Figure 1(2)(a) and (2)(c)), the objective of *SfC* is to identify the structure of the object, including its *invisible internal* structure, based on the appearance changes before and after the collision. Formally, the training data, i.e., a set of multiview videos, are defined as a collection of ground-truth color observations $\hat{\mathbf{C}}(\mathbf{r}, t)$. Here, $\mathbf{r} \in \mathbb{R}^3$ is a camera ray defined as $\mathbf{r}(s) = \mathbf{o} + s\mathbf{d}$, where $\mathbf{o} \in \mathbb{R}^3$ is the camera origin, $\mathbf{d} \in \mathbb{S}^2$ is the view direction, and $s \in [s_n, s_f]$ is the distance from $\mathbf{o}$. During training, $\mathbf{r}$ is sampled from $\hat{\mathcal{R}}$, which is a collection of camera rays in the training dataset. $t \in \{t_0, \ldots, t_{N-1}\}$ represents the time, where N is the total number of frames. Given these data, we aim to estimate the 3D structure (both external and internal ones) of the object $\mathcal{P}^P(t_0)$, which corresponds to the ground truth $\hat{\mathcal{P}}^P(t_0)$. Here, we represent the 3D structures as particle sets, $\mathcal{P}^P(t_0)$ and $\hat{\mathcal{P}}^P(t_0)$, as shown in Figure 1(b) and (d). During training, only the external appearance $\hat{\mathbf{C}}(\mathbf{r}, t)$ is observed; $\hat{\mathcal{P}}^P(t_0)$, which includes the internal structure, is not observable.

As discussed in Sections 1 and 2, *SfC* is an ill-posed problem with multiple solutions. Internal structures and physical properties, such as Young's modulus, have a mutually dependent relationship because both can explain the relationship between strain and stress. For example, a highly elastic object can be created either by making it hollow or by using soft materials. To address this issue, PAC-NeRF [36] optimizes *physical properties* by assuming that *the inside of the object is filled*. In contrast, we address a complementary problem, namely optimizing the *internal structure* based on the assumption that the *physical properties are known*. Specifically, we assume that the physical properties related to the material (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) and mass $\hat{m}$ are known. Even with this assumption, *SfC* remains a challenging problem because multiple internal structures can satisfy the same set of physical properties, as shown in Figure 1.

3.2. Preliminary: PAC-NeRF

As explained in the previous subsection, the problem settings differ between the PAC-NeRF study [36] and this study. However, because the proposed model uses PAC-NeRF to describe the physics, we briefly review PAC-NeRF here. PAC-NeRF is a variant of NeRF that bridges the Eulerian grid-based scene representation [63] with a Lagrangian particle-based differentiable physical simulation [26] for continuum materials, such as elastic materials, plasticine, sand, and fluids. PAC-NeRF obtains this functionality using three components: a continuum NeRF, a particle–grid interconverter, and a Lagrangian field.

Continuum NeRF. Continuum NeRF is built on dynamic NeRF (NeRF for a dynamic scene) [55]. In the dynamic NeRF, the volume density and color fields for position $\mathbf{x}$, view direction $\mathbf{d}$, and time t are defined as $\sigma(\mathbf{x}, t)$ and $\mathbf{c}(\mathbf{x}, \mathbf{d}, t)$, respectively. On this basis, the color of each pixel $\mathbf{C}(\mathbf{r}, t)$ is rendered using volume rendering [47]:

$$\mathbf{C}(\mathbf{r}, t) = \int_{s_n}^{s_f} T_{\mathbf{r}}(s, t)\sigma(\mathbf{r}(s), t)\mathbf{c}(\mathbf{r}(s), \mathbf{d}, t)ds, \quad (1)$$

$$T_{\mathbf{r}}(s, t) = \exp\left(-\int_{s_n}^{s} \sigma(\mathbf{r}(u), t)du\right). \quad (2)$$

This model can be trained using a pixel loss.

$$\mathcal{L}_{\text{pixel}} = \frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{|\hat{\mathcal{R}}|}\sum_{\mathbf{r}\in\hat{\mathcal{R}}}\|\mathbf{C}(\mathbf{r}, t_i) - \hat{\mathbf{C}}(\mathbf{r}, t_i)\|_2^2. \quad (3)$$

Dynamic NeRF is extended to continuum NeRF to describe the dynamics of continuum materials. This is achieved by applying the conservation laws to $\sigma(\mathbf{x}, t)$ and $\mathbf{c}(\mathbf{x}, \mathbf{d}, t)$:

$$\frac{D\sigma}{Dt} = 0, \quad \frac{D\mathbf{c}}{Dt} = \mathbf{0}, \tag{4}$$

where $\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi$ for an arbitrary time-dependent field $\phi(\mathbf{x}, t)$. Here, $\mathbf{v}$ is the velocity field and obeys momentum conservation for continuum materials:

$$\rho \frac{D\mathbf{v}}{Dt} = \nabla \cdot \mathbf{T} + \rho \mathbf{g}, \tag{5}$$

where ρ is the physical density field, $\mathbf{T}$ is the internal Cauchy stress tensor, and $\mathbf{g}$ is the gravitational acceleration. This equation can be solved differentially using the differentiable material point method (DiffMPM) [26].

Particle–grid interconverter. DiffMPM is a particle-based method that conducts simulations in a Lagrangian space. However, these particles do not necessarily lie on the ray, which makes rendering difficult. Considering this, PAC-NeRF renders in an Eulerian grid space with voxel-based NeRF [63] and bridges these two spaces using grid-to-particle (G2P) and particle-to-grid (P2G) conversions:

$$\mathcal{F}_p^P \approx \sum_i w_{ip} \mathcal{F}_i^G, \quad \mathcal{F}_i^G \approx \frac{\sum_p w_{ip} \mathcal{F}_p^P}{\sum_p w_{ip}}, \tag{6}$$

where $\mathcal{F}^X = \{\sigma^X(\mathbf{x}, t), \mathbf{c}^X(\mathbf{x}, \mathbf{d}, t)\}$ for $X \in \{G, P\}$. Here, G and P represent the Eulerian and Lagrangian views, respectively. When $\mathcal{F}^X$ is used with a subscript, that is, $\mathcal{F}_x^X$ ($x \in \{i, p\}$), the subscripts i and p indicate the grid node and particle index, respectively. w_{ip} denotes the weight of the trilinear shape function defined at i and evaluated at p.

Lagrangian field. The physical simulation and rendering pipeline in PAC-NeRF proceeds as follows: (1) Volume densities and colors are initialized over the first frame of the video sequence in an Eulerian grid field, $\mathcal{F}^{G'}(t_0)$. Here, we use the superscript G' to distinguish $\mathcal{F}^{G'}$ from $\mathcal{F}^G$ used in Step (4). (2) Using the G2P process, $\mathcal{F}^{G'}(t_0)$ is converted into a Lagrangian particle field, $\mathcal{F}^P(t_0)$. In this step, particles $\mathcal{P}^P(t_0)$ are sampled at intervals of half the grid, that is, $\frac{\Delta x}{2}$ (where Δx is the grid size), with random fluctuations. The alpha value (or amount of opacity) α_p^P is calculated for each particle using $\alpha_p^P = 1 - \exp(-\text{softplus}(\sigma_p^P))$, and a particle is removed if $\alpha_p^P < \epsilon$ ($\epsilon = 10^{-3}$ in practice). (3) The particle field in the next step, $\mathcal{F}^P(t_1)$, is calculated from $\mathcal{F}^P(t_0)$ using DiffMPM [26], where $t_1 = t_0 + \delta t$, and δt is the duration of the time step. Similarly, the particle field at t, $\mathcal{F}^P(t)$, is calculated for $t \in \{t_0, \ldots, t_{N-1}\}$. (4) Using the P2G process, $\mathcal{F}^P(t)$ is converted into an Eulerian grid field, $\mathcal{F}^G(t)$. (5) $\mathbf{C}(\mathbf{r}, t)$ is rendered based on $\mathcal{F}^G(t)$ by using voxel-based volume rendering [63].

During training, two-step optimization is conducted. (i) $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence by conducting processes (1)–(5) for $t = t_0$. (ii) Physical properties, such as the Young's modulus E and Poisson's ratio ν, are optimized for the entire video sequence by conducting processes (1)–(5) for $t \in \{t_0, \ldots, t_{N-1}\}$. In both optimizations, $\mathcal{L}_{\text{pixel}}$ (Equation 3) is used as the objective function.

3.3. Proposal: SfC-NeRF

Similar to PAC-NeRF, *SfC-NeRF* performs two-step optimization, as shown in Figure 2. The first-step optimization (Figure 2(i)) is the same as that in PAC-NeRF, that is, $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence. In this step, the filled object is learned, as shown in Figure 1(1). In contrast, the second step of the optimization (Figure 2(ii)) differs because of the difference in the optimization target. In the PAC-NeRF, the *physical properties* are optimized in this step, whereas in the *SfC-NeRF*, the *internal structure* is optimized. Specifically, as explained in the previous section, we obtain particles $\mathcal{P}^P(t_0)$ based on $\sigma^P(t_0)$, which is calculated from $\sigma^{G'}(t_0)$ (Steps (1) and (2)). Therefore, we select $\sigma^{G'}(t_0)$ as the optimization target.[3] In particular, we formulate *SfC* as a problem of optimizing $\sigma^{G'}(t_0)$ under *physical*, *appearance (i.e., external structure)-preserving*, and *keyframe constraints*, along with *volume annealing*.

Physical constraints. As discussed in Section 3.1, we assume that the physical properties related to the material (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) and mass $\hat{m}$ are known. We utilize them to narrow the solution space of *SfC*.

Physical constraints on material properties. We can reflect material-specific physical properties (e.g., Young's modulus $\hat{E}$, Poisson's ratio $\hat{\nu}$, and density $\hat{\rho}$) explicitly when constructing DiffMPM [26]. Motivated by this fact, we optimize $\sigma^{G'}(t_0)$ under the *explicit material-specific physical constraints imposed by DiffMPM*.

Physical constraints on mass. Unlike physical material properties, mass is not determined only by the material and varies depending on the individual objects. Therefore, instead of explicitly representing the mass in DiffMPM, we constrain the mass using a *mass loss*.

$$\mathcal{L}_{\text{mass}} = \| \log_{10}(m) - \log_{10}(\hat{m}) \|_2^2, \tag{7}$$

$$m = \sum_{p \in \mathcal{P}^P(t_0)} \hat{\rho} \cdot \left(\frac{\Delta x}{2} \right)^3 \cdot \alpha_p^P, \tag{8}$$

[3] Note that Lagrangian particle optimization (LPO) [31] also considers a similar optimization (i.e., optimizing $\mathcal{F}^P(t_0)$ or $\mathcal{F}^{G'}(t_0)$ through a video sequence) for few-shot (sparse view) learning. However, it aims to compensate for the *external* structure where the viewpoint is missing and *has not* sufficiently considered the components necessary for estimating the internal structures, which are discussed in the following paragraphs. We demonstrate the limitations of LPO in our experiments (Section 4).

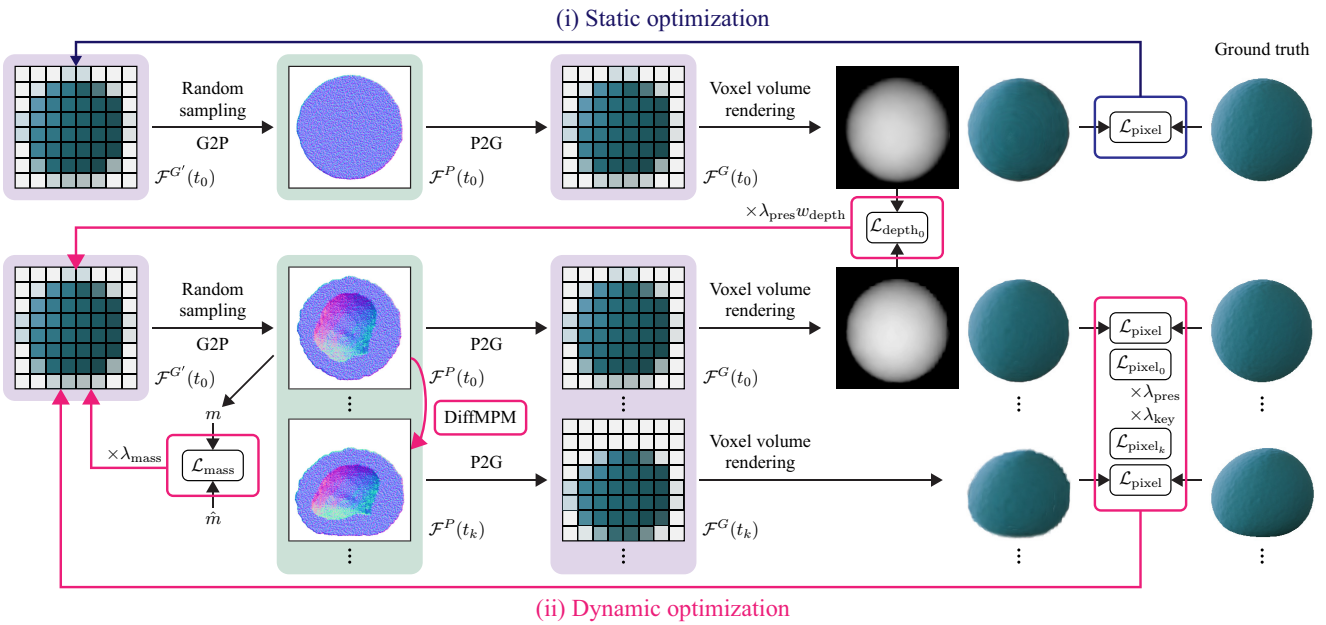

Figure 2. Optimization pipelines of SfC-NeRF. (i) The grid field $\mathcal{F}^{G'}(t_0)$ is initially optimized using the first frame of the video sequence. (ii) Subsequently, the structure (i.e., volume density $\sigma^{G'}(t_0) \in \mathcal{F}^{G'}(t_0)$) of the object is optimized through the entire video sequence with physical constraints ($\mathcal{L}_{\text{mass}}$ and DiffMPM), appearance-preserving constraints (i.e., $\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$), and keyframe constraints ($\mathcal{L}_{\text{pixel}_k}$) along with a standard pixel loss ($\mathcal{L}_{\text{pixel}}$).

where m and $\hat{m}$ are the estimated and ground-truth masses, respectively. In Equation 8, m is computed by summarizing the mass of each particle indexed by $p \in \mathcal{P}^P(t_0)$, where the mass of each particle is given by the product of the physical density $\hat{\rho}$, the unit volume of a particle $\left(\frac{\Delta x}{2}\right)^3$, and the alpha value α_p^P. In Equation 7, we employ a logarithmic scale to prioritize scale matching.

Appearance-preserving constraints. As mentioned above, we use two-step optimization: (i) $\mathcal{F}^{G'}$ is initially optimized using the first frame of the video sequence (Figure 2(i)). (ii) $\sigma^{G'}$ is optimized through a video sequence (Figure 2(ii)). In Step (ii), the external structure (or surface) learned in Step (i) does not need to be changed, considering that learning the external structure is easier than learning the internal structure. However, the physical constraints discussed above are not sufficient to satisfy this requirement. Hence, we introduced appearance-preserving constraints at both the loss and training scheme levels.

Appearance-preserving loss. The standard pixel loss (Equation 3) treats the loss for each frame equally. This is insufficient to prevent the external structure, which is well-learned in Step (i), from changing as a result of the fitting of the entire video sequence. Hence, we employ a *pixel-preserving loss* that preserves the appearance of the initial frame.

$$\mathcal{L}_{\text{pixel}_0} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} \|\mathbf{C}(\mathbf{r}, t_0) - \hat{\mathbf{C}}(\mathbf{r}, t_0)\|_2^2. \tag{9}$$

This is a variant of pixel loss (Equation 3) when $N = 1$. Because the constraints on the 2D projection plane alone are insufficient for preserving the 3D structure (e.g., objects with reversed concavity may be learned), we also incorporate a *depth-preserving loss* to encourage the preservation of the depth of the initial frame.

$$\mathcal{L}_{\text{depth}_0} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} (\|\Delta_h Z(\mathbf{r}, t_0) - \Delta_h \tilde{Z}(\mathbf{r}, t_0)\|_2^2, + \|\Delta_v Z(\mathbf{r}, t_0) - \Delta_v \tilde{Z}(\mathbf{r}, t_0)\|_2^2), \tag{10}$$

where $Z(\mathbf{r}, t_0)$ and $\tilde{Z}(\mathbf{r}, t_0)$ are the depths predicted by the current model and the model before performing Step (ii), respectively. We use $\tilde{Z}(\mathbf{r}, t_0)$ because the ground-truth depth is not observable. $Z(\mathbf{r}, t_0)$ is calculated by $Z(\mathbf{r}, t_0) = \int_{s_n}^{s_f} T_{\mathbf{r}}(s, t)\sigma(\mathbf{r}(s), t) s ds$, and $\tilde{Z}(\mathbf{r}, t_0)$ is calculated in a similar manner. Δ_h and Δ_v are operations that calculate the horizontal and vertical differences between adjacent pixels, respectively. We compare the differences rather than the raw data to mitigate the negative effects of depth estimation errors.

Appearance-preserving training. Ideally, when an object is non-transparent, its appearance is not expected to change, even if the internal volume density is changed. However, in preliminary experiments, we found that it is difficult to retain the appearance learned in Step (i) through a simple adaptation of the appearance-preserving losses. This motivated us to employ *appearance-preserving training*, that is, reoptimizing $\mathcal{F}^{G'}(t_0)$ using the first frames of the video sequence every time after optimizing $\sigma_{G'}(t_0)$ for the entire video sequence.

Keyframe constraints. As mentioned in the explanation of appearance-preserving loss, the standard pixel loss treats the loss for each frame equally. However, in preliminary experiments, we found that certain frames, particularly the frame immediately after the collision, were useful for explaining shape changes due to the internal structures. Based on this observation, we impose a *keyframe pixel loss* defined as follows:

$$\mathcal{L}_{\text{pixel}_k} = \frac{1}{|\hat{\mathcal{R}}|} \sum_{\mathbf{r} \in \hat{\mathcal{R}}} \|\mathbf{C}(\mathbf{r}, t_k) - \hat{\mathbf{C}}(\mathbf{r}, t_k)\|_2^2, \tag{11}$$

where k is the keyframe index (the frame immediately after the collision is used in practice).

Volume annealing. As discussed previously, we begin the optimization from the state in which the interior of the object is filled (Figure 2(i)). The internal structure is then optimized by reducing the volume using the aforementioned techniques (Figure 2(ii)). Owing to these learning dynamics, if the volume reduction goes in the wrong direction and leads to a local optimum, it becomes challenging to determine the global optimum. To address this issue, we introduce *volume annealing*, which involves alternating between the volume reduction and expansion. This strategy facilitates the search for a global optimum. Specifically, we implement the volume expansion by successively performing the G2P and P2G processes and replacing the obtained $\mathcal{F}^G(t_0)$ with $\mathcal{F}^{G'}(t_0)$.

Full objective. The full objective used in Step (ii) is expressed as follows:

$$\begin{aligned}\mathcal{L}_{\text{full}} &= \mathcal{L}_{\text{pixel}} + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}} \\ &+ \lambda_{\text{pres}}(\mathcal{L}_{\text{pixel}_0} + w_{\text{depth}}\mathcal{L}_{\text{depth}_0}) + \lambda_{\text{key}}\mathcal{L}_{\text{pixel}_k}\end{aligned} \tag{12}$$

where λ_{mass}, λ_{pres}, w_{depth}, and λ_{key} are the weighting hyperparameters. The effect of each loss is analyzed using the ablation study presented in Section 4.

4. Experiments

4.1. Experimental setup

We conducted three experiments to evaluate *SfC-NeRF* and explore the properties of *SfC*. First, we examined the impact of changes in the internal structure, focusing on the *cavity sizes* (Experiment I in Section 4.2) and *locations* (Experiment II in Section 4.3). We then explored the effect of the *material properties* in Experiment III (Section 4.4). The main results are summarized here, with the detailed results and implementation details provided in the Appendices. Video samples are available at the project page.

Dataset. Because *SfC* is a new task and there is no established dataset, we created a new dataset called the *SfC dataset* based on the protocol of the PAC-NeRF study [36]. We prepared 115 objects by changing their external shapes, internal structures, and materials. Figure 3 shows examples

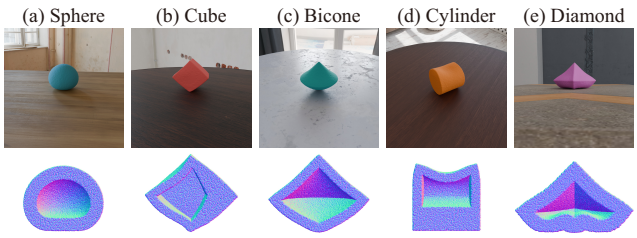

Figure 3. Examples of the data in the SfC dataset.

of the data in this dataset. First, we prepared five external shapes: *sphere*, *cube*, *bicone*, *cylinder*, and *diamond*. Regarding the internal structure and material, we set the default values as follows: the cavity size rate for filled object, s_c, was set to $(\frac{2}{3})^3$, the cavity location, l_c, was set at the center, and the material was defined as an elastic material with Young's modulus $\hat{E} = 10^6$ and Poisson's ratio $\hat{\nu} = 0.3$. Under these default properties, one of them was changed as follows: (a) Three differently sized cavities: $s_c \in \{0, (\frac{1}{2})^3, (\frac{3}{4})^3\}$. (b) Four different cavity locations: center l_c is moved $\{$up, down, left, right$\}$. (c) Eight different elastic materials: those with four different Young's moduli $\hat{E} \in \{2.5\times10^5, 5\times10^5, 2\times10^6, 4\times10^6\}$ and four different Poisson's ratios $\hat{\nu} \in \{0.2, 0.25, 0.35, 0.4\}$. Seven different materials: two Newtonian fluids, two non-Newtonian fluids, two plasticines, and one sand. Their physical properties were derived from the PAC-NeRF dataset [36]. Thus, we created 5 external shapes $\times$ (1 default + 3 sizes + 4 locations + $(8+7)$ materials) $= 115$ objects.

Following the PAC-NeRF study [36], ground-truth data were generated using the MLS-MPM simulator [25], where each object fell freely under the influence of gravity and collided with the ground plane. Images were rendered under various environmental lighting conditions and ground textures using a photorealistic renderer. Each scene was captured from 11 viewpoints, including an object, using cameras spaced in the upper hemisphere.

Preprocessing. Following the PAC-NeRF study [36], we made two assumptions and performed preprocessing to focus on solving *SfC*. (1) The intrinsic and extrinsic parameters of the cameras are known. (2) Collision objects, such as the ground plane, are known. As mentioned in [36], the latter can be easily estimated from observed images. For preprocessing, we applied video matting [41] to exclude static background objects, and concentrated the computation on the object of interest. This process provides a background segmentation mask $\hat{B}(\mathbf{r}, t)$. NeRF can estimate a background segmentation mask $B(\mathbf{r}, t)$ using $B(\mathbf{r}, t) = 1 - T_{\mathbf{r}}(s_f, t)$. Taking advantage of this property, we also used a background loss $\mathcal{L}_{\text{bg}} = \|B(\mathbf{r}, t) - \hat{B}(\mathbf{r}, t)\|_2^2$ when calculating the pixel-related losses ($\mathcal{L}_{\text{pixel}}$, $\mathcal{L}_{\text{pixel}_0}$, and $\mathcal{L}_{\text{pixel}_k}$) with a weighting hyperparameter of w_{bg}. In the experiments, this technique was applied to all the models.

Comparison models. Because there is no established method for *SfC*, we adapted previous methods to make them

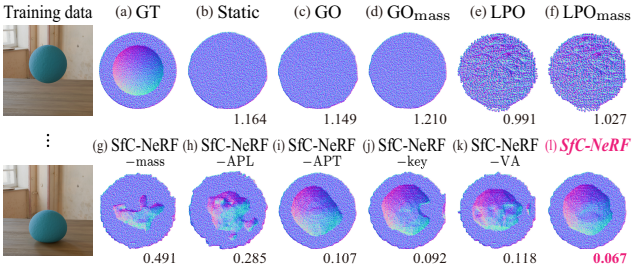

Figure 4. Comparison of learned structures for sphere objects with $s_c = (\frac{2}{3})^3$. The score under particles indicates the CD ($\times 10^3 \downarrow$). (c)–(f) GO/LPO failed to determine optimal learning directions. (g)–(k) The ablated models failed to avoid improper solutions. (l) The full model overcomes these issues and achieves the best CD.

suitable for *SfC*. Specifically, we used grid optimization (*GO*) and Lagrangian particle optimization (*LPO*) [31] as baselines. GO and LPO are improved variants of PAC-NeRF that optimize $\mathcal{F}^{G'}(t_0)$ and $\mathcal{F}^{P}(t_0)$, respectively, using $\mathcal{L}_{\text{pixel}}$ across a video sequence for few-shot learning. For a fair comparison with *SfC-NeRF*, GO and LPO were trained using the ground-truth physical properties. Although the original GO and LPO do not use the mass information for training, it may not be fair to apply it solely to the proposed method. Therefore, we also examined *GO$_{mass}$* and *LPO$_{mass}$*, extensions of GO and LPO that incorporate $\mathcal{L}_{\text{mass}}$. Furthermore, as an ablation study, we compared *SfC-NeRF* with various variants: *SfC-NeRF$_{-mass}$*, *SfC-NeRF$_{-APL}$*, *SfC-NeRF$_{-APT}$*, *SfC-NeRF$_{-key}$*, and *SfC-NeRF$_{-VA}$*, in which the mass loss ($\mathcal{L}_{\text{mass}}$),[4] appearance-preserving losses ($\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$), appearance-preserving training, keyframe loss ($\mathcal{L}_{\text{pixel}_k}$), and volume annealing were ablated, respectively. We also examined *Stacic*, a model trained using only the first frame of a video sequence, to assess the effect of optimization across videos.

Evaluation metric. As mentioned in Section 3.1, we use particles $\mathcal{P}^P(t_0)$ to represent the structure (including the internal structure) of an object and estimate $\mathcal{P}^P(t_0)$ that matches the ground truth $\hat{\mathcal{P}}^P(t_0)$. Therefore, we evaluate the model by measuring the distance between $\mathcal{P}^P(t_0)$ and $\hat{\mathcal{P}}^P(t_0)$ using the *chamfer distance (CD)*. The smaller the value, the higher is the degree of matching.

4.2. Experiment I: Influence of cavity size

First, we investigated the influence of the *cavity size* inside the object. Table 1 summarizes the quantitative results, and the qualitative results are presented in Figure 4, Appendix B.1, and the project page. Our findings are threefold. *(1) Limitations of GO and LPO [31].* GO, a simple voxel grid optimization using $\mathcal{L}_{\text{pixel}}$, failed to determine an appropriate optimization direction, which led to the deterioration of $\mathcal{P}^P(t_0)$ as it fits the video. LPO showed a slight improvement by moving particles within physical constraints via DiffMPM. However, its effectiveness was limited because significant particle movement could alter the unit volume density, making it difficult to find the optimal internal structure. Furthermore, in both GO and LPO, using mass knowledge with $\mathcal{L}_{\text{mass}}$ did not improve the performance, possibly because they lacked appearance-preserving mechanisms, and forcing m close to $\hat{m}$ can damage the overall structure. *(2) Effectiveness of each component.* The ablation study confirms the importance of each model component. *(3) Increased difficulty with increased cavity size.* Because optimization begins in the filled state, large cavity sizes require significant volume changes. We believe that this is the key reason for the deterioration in performance as the cavity size increases.

[4] As explained in Appendix C.3, the mass information is not only used in the loss but also in adjusting the learning rate. In this experiment, we ablated both to simulate a case in which the mass is unknown.

s_c	0	$(\frac{1}{2})^3$	$(\frac{2}{3})^3$	$(\frac{3}{4})^3$	Avg.
Static	0.093	0.294	0.920	1.574	0.720
GO	0.091	0.301	0.941	1.586	0.730
GO$_{\text{mass}}$	0.081	0.319	1.244	2.291	0.984
LPO	0.092	0.284	0.841	1.406	0.656
LPO$_{\text{mass}}$	0.087	0.284	0.876	1.477	0.681
SfC-NeRF$_{-\text{mass}}$	0.089	0.226	0.550	1.148	0.503
SfC-NeRF$_{-\text{APL}}$	0.106	0.423	0.898	1.326	0.688
SfC-NeRF$_{-\text{APT}}$	0.085	0.261	0.332	0.661	0.335
SfC-NeRF$_{-\text{key}}$	0.082	0.127	0.211	0.325	0.186
SfC-NeRF$_{-\text{VA}}$	0.146	0.293	0.370	0.456	0.316
SfC-NeRF	0.081	0.122	0.195	0.262	0.165

Table 1. Comparison of CD ($\times 10^3 \downarrow$) when varying the cavity size s_c. The scores were averaged over five external shapes.

l_c	left	right	up	down	Avg.
Static	0.841	0.842	0.815	0.813	0.828
GO	0.874	0.853	0.878	0.870	0.869
GO$_{\text{mass}}$	1.349	1.334	1.104	1.001	1.197
LPO	0.791	0.787	0.796	0.743	0.779
LPO$_{\text{mass}}$	0.824	0.817	0.828	0.775	0.811
SfC-NeRF$_{-\text{mass}}$	0.513	0.485	0.705	0.479	0.545
SfC-NeRF$_{-\text{APL}}$	0.845	0.783	0.805	0.583	0.754
SfC-NeRF$_{-\text{APT}}$	0.624	0.428	0.384	0.464	0.475
SfC-NeRF$_{-\text{key}}$	0.308	0.296	0.307	0.313	0.306
SfC-NeRF$_{-\text{VA}}$	0.542	0.596	0.333	0.385	0.464
SfC-NeRF	0.303 (0.367)	0.258 (0.431)	0.274 (0.448)	0.291 (0.417)	0.281 (0.416)

Table 2. Comparison of CD ($\times 10^3 \downarrow$) when varying the cavity location l_c. The gray score in parentheses indicates ACD ($\times 10^3$).

4.3. Experiment II: Influence of cavity location

Next, we examined the influence of the *cavity location*. Table 2 summarizes the quantitative results, and the qualitative results are presented in Appendix B.1 and project page. Similar to Experiment I, we observed two main findings: *(1) Limitations of GO and LPO. (2) Effectiveness of each component*. In addition, we discuss *(3) how well SfC-NeRF captured the cavity location.* A simple CD is insufficient for this evaluation because it does not account for the deviations. Therefore, we calculated the *anti-chamfer distance*

$\hat{E}$	2.5×10^5	5.0×10^5	1.0×10^6	2.0×10^6	4.0×10^6
Static	0.920	0.921	0.920	0.920	0.920
SfC-NeRF	0.289	0.254	0.195	0.314	0.374
$\hat{\nu}$	0.2	0.25	0.3	0.35	0.4
Static	0.920	0.919	0.920	0.920	0.921
SfC-NeRF	0.196	0.198	0.195	0.207	0.224

Table 3. Comparison of CD ($\times10^3\downarrow$) when varying Young's moduls $\hat{E}$ and Poisson's ratio $\hat{\nu}$.

	Newtonian	Non-Newtonian	Plasticine	Sand
Static	0.921	0.919	0.920	0.920
SfC-NeRF	0.196	0.218	0.230	0.222

Table 4. Comparison of CD ($\times10^3\downarrow$) for various materials.

(ACD), which measures the chamfer distance between the predicted particles $\mathcal{P}^P(t_0)$ and the ground-truth particles $\tilde{\mathcal{P}}^P(t_0)$, where the cavity is placed on the opposite side. This distance is expected to be longer than the original CD. The results confirm that the original CD is smaller than the ACD. These findings suggests that *SfC-NeRF* can capture the positional deviation of a cavity.

4.4. Experiment III: Influence of material

Finally, we investigated the influence of the *material properties*. Table 3 summarizes the quantitative results for elastic materials when $\hat{E}$ and $\hat{\nu}$ were varied. Table 4 summarizes the quantitative results for other materials. Appendix B.2 and project page present the qualitative results. These results demonstrate that *SfC-NeRF* improves the structure estimation compared with the initial state, regardless of the material. However, the rate of improvement depends on the material used. For example, when an object is soft, its shape changes significantly, making it difficult to capture dynamic changes. In contrast, when the object is hard, there are fewer shape changes that provide limited cues for estimating the internal structure, making learning more difficult. Thus, the proposed method is most effective when the object is moderately soft or hard. As an initial approach to address *SfC*, we proposed a general-purpose method in this study. However, in future studies, it would be interesting to develop methods that are specifically tailored to individual materials.

4.5. Application to future prediction

To demonstrate the practical importance of *SfC*, we investigated the effectiveness of *SfC-NeRF* for future prediction. Specifically, the first 14 frames were used for training and the subsequent 14 frames were used for evaluation. We compared *SfC-NeRF*, which *optimizes the internal structures* with *fixed physical properties*, with PAC-NeRF [36], which *optimizes physical properties* with *fixed (filled) internal structures*. Table 5 summarizes the results. SfC-NeRF outperformed PAC-NeRF in terms of the peak-to-signal ratio (PSNR) and structural similarity index measure (SSIM) [69]. These results indicate that the optimization of the internal structure is crucial in practical scenarios.

	Internal structure	PSNR↑	SSIM↑
PAC-NeRF	Fixed (filled)	23.44	0.975
SfC-NeRF	Optimized	26.60	0.981

Table 5. Results of future prediction. The scores were averaged over all cavity sizes and locations for the 40 objects examined in Experiments I and II.

Error rate	-30%	-20%	-10%	0%	10%	20%	30%
Young's modulus $\hat{E}$	0.363	0.242	0.216	0.195	0.213	0.231	0.244
Poisson's ratio $\hat{\nu}$	0.240	0.231	0.208	0.195	0.200	0.214	0.236
Density $\hat{\rho}$	0.798	0.533	0.289	0.195	0.207	0.259	0.308

Table 6. Comparison of CD ($\times10^3\downarrow$) for inaccurate physical properties. In the 0% case, an elastic material with default settings ($s_c = \left(\frac{2}{3}\right)^3$, $l_c =$ center, $\hat{E} = 10^6$, and $\hat{\nu} = 0.3$) was used.

5. Discussion

Based on the above experiments, we obtained promising results for *SfC*. However, the proposed method has some limitations. (1) Our approach assumes that the objects deform during collisions. Therefore, its performance depends on the type of material used. For example, it may be difficult to apply this method to metallic objects that do not deform. However, detecting small changes may help to overcome this issue. (2) Since *SfC* is a novel task, this study focused on evaluating its fundamental performance using simulation data, leaving the validation with real data as a challenge for future research. To explore its potential use with real data, we examined its robustness against inaccurate physical properties. Table 6 presents the results when errors exist in the physical properties. A significant error (e.g., -30%) in $\hat{\rho}$ causes a notable degradation owing to its negative impact on volume estimation in $\mathcal{L}_{\text{mass}}$. However, in other cases, the degradation is moderate. All the scores exceed those of the baselines listed in Table 1 (e.g., 0.841 by LPO). These results indicate that the proposed method is robust against inaccurate physical properties. Additional challenges associated with real data are discussed in Appendix A.4.

6. Conclusion

We introduced *SfC* to identify the *invisible internal* structure of an object—a task that remains challenging even with the latest neural 3D representations. We proposed *SfC-NeRF* as an initial model to address this challenge. *SfC-NeRF* solves *SfC* by optimizing the internal structures under *physical*, *appearance-preserving*, and *keyframe constraints*, along with *volume annealing*. As discussed in Section 5, the proposed method has certain limitations. Nonetheless, this study suggests a new direction for the development of neural 3D representations, and we believe that future developments in this field will overcome these limitations.

References

[1] Jad Abou-Chakra, Feras Dayoub, and Niko Sünderhauf. ParticleNeRF: A particle-based encoding for online neural radiance fields. In *WACV*, 2024. 2, 3

[2] Jad Abou-Chakra, Krishan Rana, Feras Dayoub, and Niko Sünderhauf. Physically embodied Gaussian splatting: Embedding physical priors into a visual 3D world model for robotics. In *CoRL*, 2024. 2, 3

[3] Benjamin Attal, Jia-Bin Huang, Michael Zollhoefer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding networks. In *CVPR*, 2022. 2

[4] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 2

[5] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[6] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-aliased grid-based neural radiance fields. In *ICCV*, 2023. 2

[7] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *CVPR*, 2021. 2

[8] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

[9] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3D-aware diffusion models. In *ICCV*, 2023. 2

[10] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*, 2022. 2

[11] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3D content creation. In *ICCV*, 2023. 2

[12] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. HAC: Hash-grid assisted context for 3D Gaussian splatting compression. In *ECCV*, 2024. 2

[13] Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Trans. Graph.*, 41(4), 2022. 2, 3

[14] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. GRAM: Generative radiance manifolds for 3D-aware image generation. In *CVPR*, 2022. 2

[15] Yutao Feng, Yintong Shang, Xuan Li, Tianjia Shao, Chenfanfu Jiang, and Yin Yang. PIE-NeRF: Physics-based interactive elastodynamics with NeRF. In *CVPR*, 2023. 2, 3

[16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2

[17] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *CVPR*, 2021. 2

[18] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create anything in 3D with multi-view diffusion models. In *NeurIPS*, 2024. 2

[19] Stephan J. Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. FastNeRF: High-fidelity neural rendering at 200FPS. In *ICCV*, 2021. 2

[20] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2

[21] Shanyan Guan, Huayu Deng, Yunbo Wang, and Xiaokang Yang. NeuroFluid: Fluid dynamics grounding with particle-driven neural radiance fields. In *ICML*, 2022. 2, 3

[22] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *ICCV*, 2021. 2

[23] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. EfficientNeRF: Efficient neural radiance fields. In *CVPR*, 2022. 2

[24] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-MipRF: Tri-Mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 2

[25] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), 2018. 6, 27

[26] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. In *ICLR*, 2020. 3, 4, 16, 27

[27] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. GaussianShader: 3D Gaussian splatting with shading functions for reflective surfaces. In *CVPR*, 2024. 2

[28] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. VR-GS: A physical dynamics-aware interactive Gaussian splatting system in virtual reality. *ACM Trans. Graph.*, 78, 2024. 2, 3

[29] Takuhiro Kaneko. AR-NeRF: Unsupervised learning of depth and defocus effects from natural images with aperture rendering neural radiance fields. In *CVPR*, 2022. 2

[30] Takuhiro Kaneko. MIMO-NeRF: Fast neural rendering with multi-input multi-output neural radiance fields. In *ICCV*, 2023. 2

[31] Takuhiro Kaneko. Improving physics-augmented continuum neural radiance field-based geometry-agnostic system identification with Lagrangian particle optimization. In *CVPR*, 2024. 2, 3, 4, 7, 17

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4), 2023. 1, 2

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 27

[34] Andreas Kurz, Thomas Neff, Zhaoyang Lv, Michael Zollhöfer, and Markus Steinberger. AdaNeRF: Adaptive sampling for real-time rendering of neural radiance fields. In *ECCV*, 2022. 2

[35] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3D Gaussian representation for radiance field. In *CVPR*, 2024. 2

[36] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-NeRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *ICLR*, 2023. 2, 3, 6, 8, 17, 25, 27

[37] Yanyan Li, Chenyu Lyu, Yan Di, Guangyao Zhai, Gim Hee Lee, and Federico Tombari. GeoGaussian: Geometry-aware Gaussian splatting for scene rendering. In *ECCV*, 2024. 2

[38] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021. 2

[39] Zhihao Liang, Qi Zhang, Wenbo Hu, Ying Feng, Lei Zhu, and Kui Jia. Analytic-Splatting: Anti-aliased 3D Gaussian splatting via analytic integration. In *ECCV*, 2024. 2

[40] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. In *CVPR*, 2023. 2

[41] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L. Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, 2021. 6, 13

[42] David B. Lindell, Julien N. P. Martel, and Gordon Wetzstein. AutoInt: Automatic integration for fast neural volume rendering. In *CVPR*, 2021. 2

[43] Jiayue Liu, Xiao Tang, Freeman Cheng, Roy Yang, Zhihao Li, Jianzhuang Liu, Yi Huang, Jiaqi Lin, Shiyong Liu, Xiaofei Wu, Songcen Xu, and Chun Yuan. MirrorGaussian: Reflecting 3D Gaussians for reconstructing mirror reflections. In *ECCV*, 2024. 2

[44] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 2

[45] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3D geometry-aware deformable Gaussian splatting for dynamic view synthesis. In *CVPR*, 2024. 2

[46] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 2

[47] 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. In *ECCV*, 2020. 1, 2, 3

[48] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *CVPR*, 2022. 2

[49] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4), 2022. 2

[50] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton Kaplanyan, and Markus Steinberger. DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. *Comput. Graph. Forum*, 40(4), 2021.

[51] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3D Gaussian splatting for accelerated novel view synthesis. In *CVPR*, 2024. 2

[52] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[53] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021. 2

[54] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *ICLR*, 2023. 2

[55] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, 2021. 2, 3

[56] Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Feature Splatting: Language-driven physics-based scene synthesis and editing. In *ECCV*, 2024. 2, 3

[57] Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. DeRF: Decomposed radiance fields. In *CVPR*, 2021. 2

[58] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. KiloNeRF: Speeding up neural radiance fields with thousands of tiny MLPs. In *ICCV*, 2021. 2

[59] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *NeurIPS*, 2020. 2

[60] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *NeurIPS*, 2021. 2

[61] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. EpiGRAF: Rethinking training of 3D GANs. In *NeurIPS*, 2022. 2

[62] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *CVPR*, 2022. 2

[63] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 1, 2, 3, 4, 27

[64] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view Gaussian model for high-resolution 3D content creation. In *ECCV*, 2024. 2

[65] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3D content creation. In *ICLR*, 2024. 2

[66] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *ICCV*, 2021. 2

[67] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *CVPR*, 2022. 2

[68] Huan Wang, Jian Ren, Zeng Huang, Kyle Olszewski, Menglei Chai, Yun Fu, and Sergey Tulyakov. R2L: Distilling neural radiance field to neural light field for efficient novel view synthesis. In *ECCV*, 2022. 2

[69] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4), 2004. 8

[70] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. In *NeurIPS*, 2023. 2

[71] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. NeX: Real-time view synthesis with neural basis expansion. In *CVPR*, 2021. 2

[72] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D Gaussians for generative dynamics. In *CVPR*, 2024. 2, 3

[73] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. GIRAFFE HD: A high-resolution 3D-aware generative model. In *CVPR*, 2022. 2

[74] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3D Gaussian splatting for anti-aliased rendering. In *CVPR*, 2024. 2

[75] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D Gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. 2

[76] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D Gaussian splatting. In *ICLR*, 2024. 2

[77] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. GaussianDreamer: Fast generation from text to 3D Gaussians by bridging 2D and 3D diffusion models. In *CVPR*, 2024. 2

[78] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021. 2

[79] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-Splatting: Alias-free 3D Gaussian splatting. In *CVPR*, 2024. 2

[80] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[81] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. DreamScene360: Unconstrained text-to-3D scene generation with panoramic Gaussian splatting. In *ECCV*, 2024. 2

Contents

A. Detailed analyses and discussions

A.1. Detailed ablation studies

Owing to space limitations in the main text, we conducted an ablation study that focused only on the selected key components. In this appendix, we present detailed ablation studies that further assess the effectiveness of the proposed method from multiple perspectives. Specifically, we examine the effects of *each appearance-preserving loss* (Appendix A.1.1), *keyframe selection* (Appendix A.1.2), and *background loss* (Appendix A.1.3).

$\mathcal{L}_{\text{pixel}_0}$	$\mathcal{L}_{\text{depth}_0}$	0	$(\frac{1}{2})^3$	$(\frac{2}{3})^3$	$(\frac{3}{4})^3$	Avg.
		0.106	0.423	0.898	1.326	0.688
✓		0.105	0.142	0.334	0.342	0.231
	✓	0.079	0.313	0.314	0.287	0.248
✓	✓	0.081	0.122	0.195	0.262	0.165

Table 7. Results of the detailed ablation study of APLs when the cavity size s_c is varied. The score indicates CD ($\times 10^3\downarrow$). A checkmark ✓ indicates that the corresponding loss was used.

$\mathcal{L}_{\text{pixel}_0}$	$\mathcal{L}_{\text{depth}_0}$	left	right	up	down	Avg.
		0.845	0.783	0.805	0.583	0.754
✓		0.295	0.451	0.325	0.311	0.345
	✓	0.362	0.299	0.348	0.389	0.349
✓	✓	0.303	0.258	0.274	0.291	0.281

Table 8. Results of the detailed ablation study of APLs when the cavity location l_c is varied. The score indicates CD ($\times 10^3\downarrow$). A checkmark ✓ indicates that the corresponding loss was used.

A.1.1. Effect of each appearance-preserving loss

As explained in Section 3.3, regarding appearance-preserving constraints, we adopted two appearance-preserving losses (APLs): pixel-preserving loss $\mathcal{L}_{\text{pixel}_0}$ (Equation 9) and depth-preserving loss $\mathcal{L}_{\text{depth}_0}$ (Equation 10). These losses help prevent the degradation of the external structure, which is effectively learned from the first frame of the video sequence during the fitting process across the entire video sequence. In the ablation study presented in Sections 4.2 and 4.3, we ablated both losses simultaneously to examine the overall effect of APLs. For a more detailed ablation study, we assessed the performance when each APL was individually ablated.

Results. Table 7 summarizes the results when the cavity size s_c is varied, and Table 8 summarizes the results when the cavity location l_c is varied. Our findings are threefold:

(1) No APL vs. either $\mathcal{L}_{pixel_0}$ or $\mathcal{L}_{depth_0}$. Both SfC-NeRF with only $\mathcal{L}_{\text{pixel}_0}$ and SFC-NeRF with only $\mathcal{L}_{\text{depth}_0}$ outperformed SfC-NeRF without APL in all cases. These results indicate that both $\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$ effectively enhance the performance of *SfC*.

(2) Full APLs vs. either $\mathcal{L}_{pixel_0}$ or $\mathcal{L}_{depth_0}$. SfC-NeRF with both $\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$ outperformed SfC-NeRF with only $\mathcal{L}_{\text{pixel}_0}$ and SFC-NeRF with only $\mathcal{L}_{\text{depth}_0}$ in most cases. These results indicate that $\mathcal{L}_{\text{pixel}_0}$ and $\mathcal{L}_{\text{depth}_0}$ contribute to improving the performance of *SfC* from different perspectives and are most effective when used together.

(3) $\mathcal{L}_{pixel_0}$ vs $\mathcal{L}_{depth_0}$. The superiority or inferiority of each loss depends on the cavity settings. This is related to the

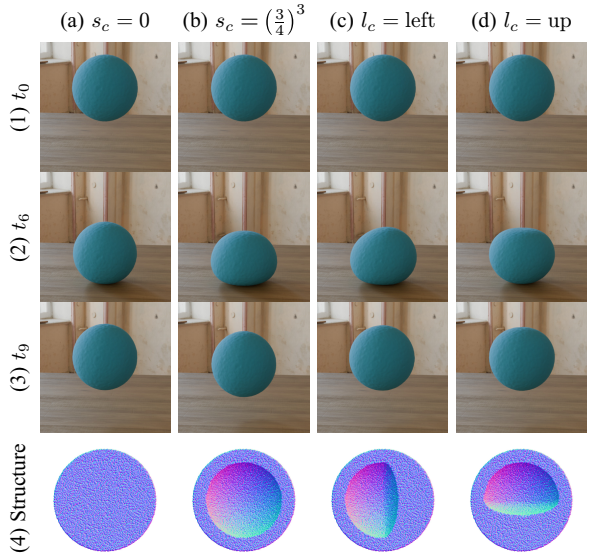

Figure 5. Comparison of appearances for objects with different internal structures when t is varied within $\{t_0, t_6, t_9\}$.

learnability of 3D appearance, and further detailed analyses will be an interesting direction for future research.

A.1.2. Effect of keyframe selection

As discussed in Section 3.3, regarding keyframe constraints, we employed a keyframe pixel loss $\mathcal{L}_{\text{pixel}_k}$ (Equation 11) to effectively capture shape changes caused by internal structures. Specifically, we selected the frame immediately after the collision as the keyframe ($k = 6$, where k is the keyframe index) for the experiments described in the main text. An important question is whether the choice of k is optimal. To investigate this, we evaluated the change in performance by varying the value of k, specifically within $\{6, 9\}$. Figure 5 compares the appearances of objects with different internal structures in these keyframes. For reference, we also provide scores for the model without keyframe pixel loss (denoted by $k =$ *None*).

Results. Table 9 summarizes the results when the cavity size s_c is varied, and Table 10 summarizes the results when the cavity location l_c is varied. Our findings are twofold:

(1) $\mathcal{L}_{pixel_6}$ *vs.* $\mathcal{L}_{pixel_9}$. SfC-NeRF with $\mathcal{L}_{\text{pixel}_6}$ outperformed that with $\mathcal{L}_{\text{pixel}_9}$ in most cases. As shown in Figure 5, immediately after the collision (at t_6 (2)), the difference in the shapes of the objects is noticeable. However, as time progressed after the collision (at t_9 (3)), the difference in the shapes of the objects decreased, whereas the difference in their positions became more pronounced. We consider this to be the main reason why SfC-NeRF with $\mathcal{L}_{\text{pixel}_6}$ performed better than that with $\mathcal{L}_{\text{pixel}_9}$.

(2) $\mathcal{L}_{pixel_6}$/$\mathcal{L}_{pixel_9}$ *vs. None.* We found that SfC-NeRF with $\mathcal{L}_{\text{pixel}_6}$ or $\mathcal{L}_{\text{pixel}_9}$ outperformed SfC-NeRF without keyframe pixel loss in most cases. These results indicate that strategically weighing frames is more effective than treating all frames equally.

k	0	$(\frac{1}{2})^3$	$(\frac{2}{3})^3$	$(\frac{3}{4})^3$	Avg.
None	0.082	0.127	0.211	0.325	0.186
6	0.081	0.122	0.195	0.262	0.165
9	0.082	0.120	0.208	0.290	0.175

Table 9. Analysis of the effect of keyframe selection when the cavity size s_c is varied. The score indicates CD ($\times 10^3 \downarrow$). When $k =$ None, a keyframe pixel loss $\mathcal{L}_{\text{pixel}_k}$ was not used. In contrast, when $k \in \{6, 9\}$, $\mathcal{L}_{\text{pixel}_k}$ was used.

k	left	right	up	down	Avg.
None	0.308	0.296	0.307	0.313	0.306
6	0.303	0.258	0.274	0.291	0.281
9	0.296	0.296	0.313	0.303	0.302

Table 10. Analysis of the effect of keyframe selection when the cavity location l_c is varied. The score indicates CD ($\times 10^3 \downarrow$). When $k =$ None, a keyframe pixel loss $\mathcal{L}_{\text{pixel}_k}$ was not used. In contrast, when $k \in \{6, 9\}$, $\mathcal{L}_{\text{pixel}_k}$ was used.

A.1.3. Effect of background loss

As mentioned in the explanation of preprocessing in Section 4.1, we use a background loss $\mathcal{L}_{\text{bg}}$ by leveraging the fact that the background segmentation has been obtained. For example, when an image with a white background is given, this background loss is useful for distinguishing whether the white part belongs to the background or a foreground object. We used a background segmentation that is not created manually but is predicted from a given image using a DNN-based image matting model [41]. Therefore, this setting is not unrealistic. However, it is important to investigate the effectiveness of the background loss. To this end, we investigated the performance of *SfC-NeRF*$_{-bg}$, where the background loss ($\mathcal{L}_{\text{bg}}$) was ablated. In this setting, the performance of a model trained using only the first frame of the video sequence (Step (i) in Figure 2(a)) also changes because the background loss is ablated in this step. We refer to this model as *Static*$_{-bg}$. We compared the scores of these models with those of the original models (i.e., *SfC-NeRF* and *Static*).

Results. Table 11 summarizes the results when the cavity size s_c is varied, and Table 12 summarizes the results when the cavity location l_c is varied. Our findings are twofold:

(1) SfC-NeRF vs. SfC-NeRF$_{-bg}$. SfC-NeRF outperformed SfC-NeRF$_{-\text{bg}}$ in most cases. As mentioned above, the background loss is useful for distinguishing between background and foreground objects, allowing for a more accurate capture of external structures. The movements of an object are affected by its external and internal structures. Therefore, if the external structure can be estimated more

	0	$(\frac{1}{2})^3$	$(\frac{2}{3})^3$	$(\frac{3}{4})^3$	Avg.
Static	0.093	0.294	0.920	1.574	0.720
SfC-NeRF	0.081	0.122	0.195	0.262	0.165
Static$_{-\text{bg}}$	0.093	0.290	0.906	1.545	0.708
SfC-NeRF$_{-bg}$	0.101	0.149	0.222	0.279	0.188

Table 11. Results of the ablation study of background loss when the cavity size s_c is varied. The score indicates CD ($\times 10^3 \downarrow$).

	left	right	up	down	Avg.
Static	0.841	0.842	0.815	0.813	0.828
SfC-NeRF	0.303	0.258	0.274	0.291	0.281
Static$_{-\text{bg}}$	0.831	0.830	0.799	0.800	0.815
SfC-NeRF$_{-bg}$	0.324	0.210	0.361	0.277	0.293

Table 12. Results of the ablation study of background loss when the cavity location l_c is varied. The score indicates CD ($\times 10^3 \downarrow$).

accurately, the internal structure can also be estimated more accurately.

(2) SfC-NeRF$_{-bg}$ *vs Static*$_{-bg}$. SfC-NeRF$_{-\text{bg}}$ outperformed Static $_{-\text{bg}}$ except when dealing with filled objects ($s_c = 0$ in Table 11).[5] These results indicate that the proposed method is effective for improving the performance of *SfC*, even without the use of advanced techniques, such as background loss.

A.2. Extended experiments

A.2.1. Experiment IV: Influence of collision angle

In the above experiments, the collision angle was fixed, as shown in Figures 6–13, regardless of the internal structure and physical properties, to focus on comparisons related to the internal structures and physical properties. For completeness, we investigated the influence of *collision angle* θ_c on the performance of *SfC*. Specifically, we selected objects with default settings ($s_c = (\frac{2}{3})^3$, l_c = center, and elastic material defined by $\hat{E} = 1.0 \times 10^6$ and $\hat{\nu} = 0.3$) as the objects of investigation and examined their performance when only the collision angles were altered. The objects were rotated in the depth direction, as shown in Figure 14. The collision angle θ_c was chosen from $\{0°, 22.5°, 45°, 67.5°, 90°\}$. We compared the performance of *Static* and *SfC-NeRF*.

Results. Table 13 summarizes the quantitative results. Figure 14 shows the qualitative results. Our findings are twofold:

(1) SfC-NeRF vs. Static. SfC-NeRF outperformed Static in all cases. These results indicate that optimizing the internal structure through a video sequence using the proposed method is beneficial, regardless of the collision angle.

(2) Effect of collision angle. We found that the collision angle influenced the performance of *SfC*. The strength of this effect depends on the object shape. There are three possible reasons for this performance variation: *(i) Changes in estimation accuracy of external structures.* The internal structure was optimized under the constraint that the external structure, learned from the first frame, should be maintained. Therefore, when the accuracy of the external structure estimation changed, the accuracy of the internal structure estimation also changed. *(ii) Difference in amount of deformation.* The amount of deformation varied depending on the collision angle. This factor also affected the ease of estimating the internal structure. *(iii) Asymmetry.* When an object was not symmetrical relative to the collision angle, its behavior after the collision became asymmetrical. Consequently, the ease of estimating the internal structure also became asymmetrical.

Sphere	0°	22.5°	45°	67.5°	90°
Static	1.164	1.163	1.163	1.162	1.160
SfC-NeRF	0.067	0.068	0.066	0.067	0.066
Cube	0°	22.5°	45°	67.5°	90°
Static	0.775	0.776	0.848	0.768	0.776
SfC-NeRF	0.201	0.173	0.627	0.201	0.201
Bicone	0°	22.5°	45°	67.5°	90°
Static	0.933	0.925	0.918	0.921	0.926
SfC-NeRF	0.144	0.194	0.187	0.146	0.154
Cylinder	0°	22.5°	45°	67.5°	90°
Static	0.891	0.905	0.915	0.905	0.964
SfC-NeRF	0.342	0.288	0.311	0.209	0.639
Diamond	0°	22.5°	45°	67.5°	90°
Static	0.837	0.830	0.833	0.819	0.838
SfC-NeRF	0.220	0.300	0.222	0.163	0.209

Table 13. Comparison of CD ($\times 10^3 \downarrow$) when collision angle θ_c is varied.

A.3. Evaluation from multiple perspectives

A.3.1. Evaluation through video sequences

In the main experiments, we evaluated the models using the chamfer distance between the ground-truth particles $\hat{\mathcal{P}}^P(t_0)$ and estimated particles $\mathcal{P}^P(t_0)$ in the *first frame* of the video sequence, i.e., at $t = t_0$. For the multidimensional analysis, we investigated the chamfer distance between the ground-truth particles $\hat{\mathcal{P}}^P(t)$ and estimated particles $\mathcal{P}^P(t)$, averaged over the *entire video sequence*, i.e., $t \in \{t_0, \ldots, t_{N-1}\}$. For clarity, we refer to the former

[5] When handling a filled object, inaccurate estimation of external structure is problematic because it causes a difference between the actual and estimated masses. In this situation, if the estimated mass is encouraged to approach the ground-truth mass using a mass loss while maintaining the external appearance using APLs, the internal structure must be changed unnecessarily. Consequently, SfC-NeRF$_{-\text{bg}}$ degrades the performance of *SfC* when handling filled objects. An accurate estimation of the external structure using a background loss is effective for addressing this issue.

	0		$(\frac{1}{2})^3$		$(\frac{2}{3})^3$		$(\frac{3}{4})^3$		Avg.	
Static	0.093	0.104	0.294	0.309	0.920	1.057	1.574	1.964	0.720	0.859
GO	0.091	0.092	0.301	0.301	0.941	0.944	1.586	1.612	0.730	0.737
GO_{mass}	0.081	0.083	0.319	0.325	1.244	1.266	2.291	2.367	0.984	1.010
LPO	0.092	0.091	0.284	0.282	0.841	0.833	1.406	1.380	0.656	0.646
LPO_{mass}	0.087	0.087	0.284	0.283	0.876	0.868	1.477	1.451	0.681	0.672
SfC-NeRF$_{-mass}$	0.089	0.090	0.226	0.225	0.550	0.544	1.148	1.112	0.503	0.493
SfC-NeRF$_{-APL}$	0.106	0.108	0.423	0.421	0.898	0.886	1.326	1.307	0.688	0.680
SfC-NeRF$_{-APT}$	0.085	0.101	0.261	0.279	0.332	0.337	0.661	0.680	0.335	0.349
SfC-NeRF$_{-key}$	0.082	0.086	0.127	0.131	0.211	0.213	0.325	0.325	0.186	0.189
SfC-NeRF$_{-VA}$	0.146	0.269	0.293	0.338	0.370	0.407	0.456	0.485	0.316	0.375
SfC-NeRF	0.081	0.085	0.122	0.126	0.195	0.196	0.262	0.258	0.165	0.166

Table 14. Comparison of CD ($\times 10^3 \downarrow$) when the cavity size s_c is varied. This is an extended version of Table 1. For each condition, the left score indicates CD_{static}, the chamfer distance between $\mathcal{P}(t_0)$ and $\hat{\mathcal{P}}(t_0)$ at the first frame, i.e., $t = t_0$, and the right score indicates CD_{video}, the chamfer distance between $\mathcal{P}(t)$ and $\hat{\mathcal{P}}(t)$ averaged over the entire video sequence, i.e., $t \in \{t_0, \ldots, t_{N-1}\}$.

	left		right		up		down		Avg.	
Static	0.841 (0.841)	1.159 (1.294)	0.842 (0.843)	1.306 (1.154)	0.815 (0.814)	1.731 (1.246)	0.813 (0.813)	1.241 (1.727)	0.828 (0.828)	1.359 (1.355)
GO	0.874 (0.872)	0.879 (2.606)	0.853 (0.856)	0.870 (2.549)	0.878 (0.881)	0.875 (1.471)	0.870 (0.870)	1.035 (1.673)	0.869 (0.870)	0.915 (2.075)
GO_{mass}	1.349 (1.340)	1.386 (3.134)	1.334 (1.344)	1.375 (3.126)	1.104 (1.127)	1.141 (1.866)	1.001 (1.004)	1.370 (1.805)	1.197 (1.204)	1.318 (2.483)
LPO	0.791 (0.802)	0.789 (2.493)	0.787 (0.800)	0.787 (2.507)	0.796 (0.819)	0.776 (1.468)	0.743 (0.737)	0.721 (1.471)	0.779 (0.790)	0.768 (1.985)
LPO_{mass}	0.824 (0.833)	0.822 (2.529)	0.817 (0.832)	0.818 (2.556)	0.828 (0.847)	0.806 (1.497)	0.775 (0.771)	0.753 (1.538)	0.811 (0.821)	0.800 (2.030)
SfC-NeRF$_{-mass}$	0.513 (0.858)	0.520 (2.502)	0.485 (0.878)	0.491 (2.661)	0.705 (0.747)	0.689 (1.506)	0.479 (0.956)	0.457 (1.762)	0.545 (0.860)	0.539 (2.108)
SfC-NeRF$_{-APL}$	0.845 (1.069)	0.840 (2.885)	0.783 (1.083)	0.788 (2.943)	0.805 (0.934)	0.786 (1.764)	0.583 (0.883)	0.580 (1.750)	0.754 (0.992)	0.749 (2.335)
SfC-NeRF$_{-APT}$	0.624 (0.588)	0.631 (1.920)	0.428 (0.586)	0.604 (1.486)	0.384 (0.579)	0.461 (1.196)	0.464 (0.646)	0.514 (1.305)	0.475 (0.600)	0.553 (1.477)
SfC-NeRF$_{-key}$	0.308 (0.372)	0.307 (1.854)	0.296 (0.396)	0.326 (1.746)	0.307 (0.387)	0.306 (1.291)	0.313 (0.389)	0.343 (1.105)	0.306 (0.386)	0.321 (1.499)
SfC-NeRF$_{-VA}$	0.542 (0.639)	0.611 (2.304)	0.596 (0.757)	0.767 (2.265)	0.333 (0.445)	0.389 (1.338)	0.385 (0.549)	0.421 (1.339)	0.464 (0.597)	0.547 (1.811)
SfC-NeRF	0.303 (0.367)	0.308 (1.821)	0.258 (0.431)	0.313 (1.647)	0.274 (0.448)	0.273 (1.262)	0.291 (0.417)	0.307 (1.204)	0.281 (0.416)	0.300 (1.483)

Table 15. Comparison of CD and ACD ($\times 10^3 \downarrow$) when the cavity location l_c is varied. This is an extended version of Table 2. For each condition, the left score indicates CD_{static}, the chamfer distance between $\mathcal{P}(t_0)$ and $\hat{\mathcal{P}}(t_0)$ at the first frame, i.e., $t = t_0$, and the right score indicates CD_{video}, the chamfer distance between $\mathcal{P}(t)$ and $\hat{\mathcal{P}}(t)$ averaged over the entire video sequence, i.e., $t \in \{t_0, \ldots, t_{N-1}\}$. The gray score in parentheses indicates the ACD. For each condition, the left score indicates ACD_{static}, the anti-chamfer distance at the first frame, and the right score indicates ACD_{video}, the anti-chamfer distance averaged over the entire video sequence. It is expected that each original CD is smaller than the corresponding ACD.

(chamfer distance for the first static frame) as CD_{static} and the latter (chamfer distance for the entire video sequence) as CD_{video}. In the evaluation of the influence of cavity location (Section 4.3), we introduce anti-chamfer distance, which is the chamfer distance between the predicted particles $\mathcal{P}^P(t_0)$ and ground-truth particles $\tilde{\mathcal{P}}^P(t_0)$, where the cavity is placed on the opposite side, in the *first frame* of the video sequence to evaluate how well the cavity location is captured. For further analysis, we calculated and averaged similar scores for the *entire video sequence*. For clarity, we refer to the former (anti-chamfer distance for the first static frame) as ACD_{static} and the latter (anti-chamfer distance for the entire video sequence) as ACD_{video}.

Results. Table 14 summarizes the results when the cavity size s_c is varied, and Table 15 summarizes the results when the cavity location l_c is varied. Our findings are fourfold:

(1) CD_{static} vs. CD_{video}. The relative values of CD_{static} and CD_{video} vary across different cases. When calculating CD_{static} in the first frame, the locations of the ground-truth and synthesized objects were well aligned, allowing us to

focus on the differences in shapes. In contrast, when calculating CD_{video} for the entire video sequence, we must consider not only the differences in shapes but also the differences in absolute locations. Misalignments accumulate over time because the locations must vary within the allowance of the physical constraints via DiffMPM [26]. Because the objective of this study was to correctly predict the shape rather than the location, $\text{CD}_{\text{static}}$ is a more valid evaluation than CD_{video} for this purpose.

(2) Comparison of $\text{CD}_{\text{static}}$ and CD_{video} among models. Although there was some variation in the superiority of the models depending on the metric used, the general trend remained consistent: SfC-NeRF achieved the best score in most cases. The two exceptions are CD_{video} for $s_c = 0$ in Table 14 and CD_{video} for $l_c =$ left in Table 15. However, the difference from the best score is small (less than 0.002). These results validate the effectiveness of the proposed method compared to the baseline and ablated models according to both metrics.

(3) $\text{ACD}_{\text{static}}$ vs. $\text{ACD}_{\text{video}}$. Comparing $\text{ACD}_{\text{static}}$ with $\text{ACD}_{\text{video}}$, $\text{ACD}_{\text{static}}$ is smaller than $\text{ACD}_{\text{video}}$. This is because the difference in location gradually increased after the collision when the cavity was located on the opposite side. As the objective of this study was to correctly predict the shape rather than the location, $\text{ACD}_{\text{static}}$ is a more valid evaluation than $\text{ACD}_{\text{video}}$ for this purpose.

(4) Comparison of $\text{CD}_{\text{static}}$ and $\text{ACD}_{\text{static}}$ among models. When comparing the models, the baselines (i.e., the GO- and LPO-based models) tended to obtain similar $\text{CD}_{\text{static}}$ and $\text{ACD}_{\text{static}}$ values because they struggled to determine the optimization direction, as shown in Figures 6–10. In contrast, the proposed models (i.e., the SfC-NeRF-based models, including the ablated models) tended to obtain a smaller $\text{CD}_{\text{static}}$ than $\text{ACD}_{\text{static}}$. These results indicate that the proposed models effectively capture the positional bias of the cavity. Notably, a larger $\text{ACD}_{\text{static}}$ does not indicate better performance unless $\text{CD}_{\text{static}}$ is adequately small because it is possible to increase $\text{ACD}_{\text{static}}$ while sacrificing $\text{CD}_{\text{static}}$.

A.3.2. Evaluation per external shape

In Experiments I (Section 4.2) and II (Section 4.3), we reported the scores averaged over external shapes (i.e., sphere, cube, bicone, cylinder, and diamond objects). For a different evaluation perspective, this appendix presents the scores for each external shape, averaged over other conditions, i.e., either $s_c \in \{0, (\frac{1}{2})^3, (\frac{2}{3})^3, (\frac{3}{4})^3\}$ or $l_c \in \{\text{left}, \text{right}, \text{up}, \text{down}\}$.

Results. Table 16 summarizes the results when the cavity size s_c is varied (related to the results in Table 1), and Table 17 summarizes the results when the cavity location l_c is varied (related to the results in Table 2). Although the scores were affected by the external shape, the same trends observed previously regarding the superiority or inferiority of the models were maintained. In particular, SfC-NeRF outperformed both the baseline and ablated models in most cases.

	Sphere	Cube	Bicone	Cylinder	Diamond
Static	0.897	0.612	0.724	0.697	0.671
GO	0.889	0.637	0.704	0.756	0.663
GO_{mass}	0.934	1.345	0.760	1.218	0.663
LPO	0.774	0.564	0.639	0.678	0.622
LPO_{mass}	0.796	0.605	0.656	0.726	0.622
$\text{SfC-NeRF}_{-\text{mass}}$	0.561	0.500	0.455	0.447	0.553
$\text{SfC-NeRF}_{-\text{APL}}$	0.303	1.082	0.579	0.885	0.591
$\text{SfC-NeRF}_{-\text{APT}}$	0.178	0.375	0.286	0.502	0.331
$\text{SfC-NeRF}_{-\text{key}}$	0.081	0.173	0.159	0.288	0.230
$\text{SfC-NeRF}_{-\text{VA}}$	0.113	0.279	0.363	0.558	0.268
SfC-NeRF	0.067	0.163	0.138	0.264	0.193

Table 16. Comparison of CD ($\times 10^3 \downarrow$) when the cavity size s_c is varied. The scores were averaged over cavity sizes.

	Sphere	Cube	Bicone	Cylinder	Diamond
Static	1.006	0.719	0.824	0.818	0.772
GO	0.991	0.809	0.847	0.898	0.799
GO_{mass}	1.065	1.528	0.934	1.332	1.125
LPO	0.954	0.673	0.764	0.804	0.701
LPO_{mass}	0.980	0.723	0.796	0.845	0.711
$\text{SfC-NeRF}_{-\text{mass}}$	0.695	0.480	0.424	0.595	0.533
$\text{SfC-NeRF}_{-\text{APL}}$	0.548	1.064	0.373	1.194	0.592
$\text{SfC-NeRF}_{-\text{APT}}$	0.318	0.502	0.374	0.730	0.451
$\text{SfC-NeRF}_{-\text{key}}$	0.189	0.371	0.235	0.448	0.286
$\text{SfC-NeRF}_{-\text{VA}}$	0.240	0.418	0.790	0.534	0.338
SfC-NeRF	0.152 (0.417)	0.342 (0.386)	0.231 (0.365)	0.393 (0.491)	0.289 (0.420)

Table 17. Comparison of CD ($\times 10^3 \downarrow$) when the cavity location l_c is varied. The scores were averaged over cavity locations. The gray score in parentheses indicates ACD ($\times 10^3$). It is expected that the original CD is smaller than this.

A.4. Possible challenges with real data

As discussed in Section 5, because *SfC* is a novel task, this study focused on evaluating its fundamental performance using simulation data, leaving validation with real data a challenge for future research. However, it is both feasible and important to discuss the potential challenges associated with real data, which we address in this appendix. Three potential challenges are outlined below:

(1) Difficulty in accurately estimating external structures. Although significant progress has been made in the estimation of 3D external structures in recent years, it is not yet possible to accurately estimate them for all objects in all situations. The proposed method assumes that the external structure learned in the first frame of the video sequence is accurate. Therefore, if this estimation fails, the overall performance is degraded. We believe that incorporating the concept of a physics-informed model, particularly in chal-

lenging scenarios (e.g., sparse views), such as Lagrangian particle optimization [31], could provide a solution to this issue.

(2) Gap between real physics and the physics used in the simulation. Despite recent advancements in physical simulation models, discrepancies between real-world physics and the physics underlying the simulation still persist. We believe that refining the proposed method alongside physics-informed models (e.g., those discussed in Section 2) could help alleviate this problem.

(3) Difficulty in accurately estimating physical properties. As mentioned in Section 3.1, we address *SfC* under the assumption that the ground-truth physical properties are available in advance to mitigate the chicken-and-egg problem between the physical properties and internal structures. This assumption is reasonable if the material can be identified; however, obtaining perfectly accurate values for physical properties in real-world scenarios is challenging. Although the issue of solving the chicken-and-egg problem remains, an appearance-based physical property estimation method has already been proposed (e.g., PAC-NeRF [36]). Combining the proposed approach with previous methods for the simultaneous optimization of physical properties and internal structures is an exciting direction for future research.

B. Qualitative results

This appendix presents the qualitative results. The corresponding demonstration videos are available at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/sfc/.

B.1. Qualitative results of Experiments I and II

We provide the qualitative results of Experiments I (Section 4.2) and II (Section 4.3) in Figures 6–10.

B.2. Qualitative results of Experiment III

We provide the qualitative results of Experiment III (Section 4.4) in Figures 11–13.

B.3. Qualitative results of Experiment IV

We provide the qualitative results of Experiment IV (Appendix A.2.1) in Figure 14.

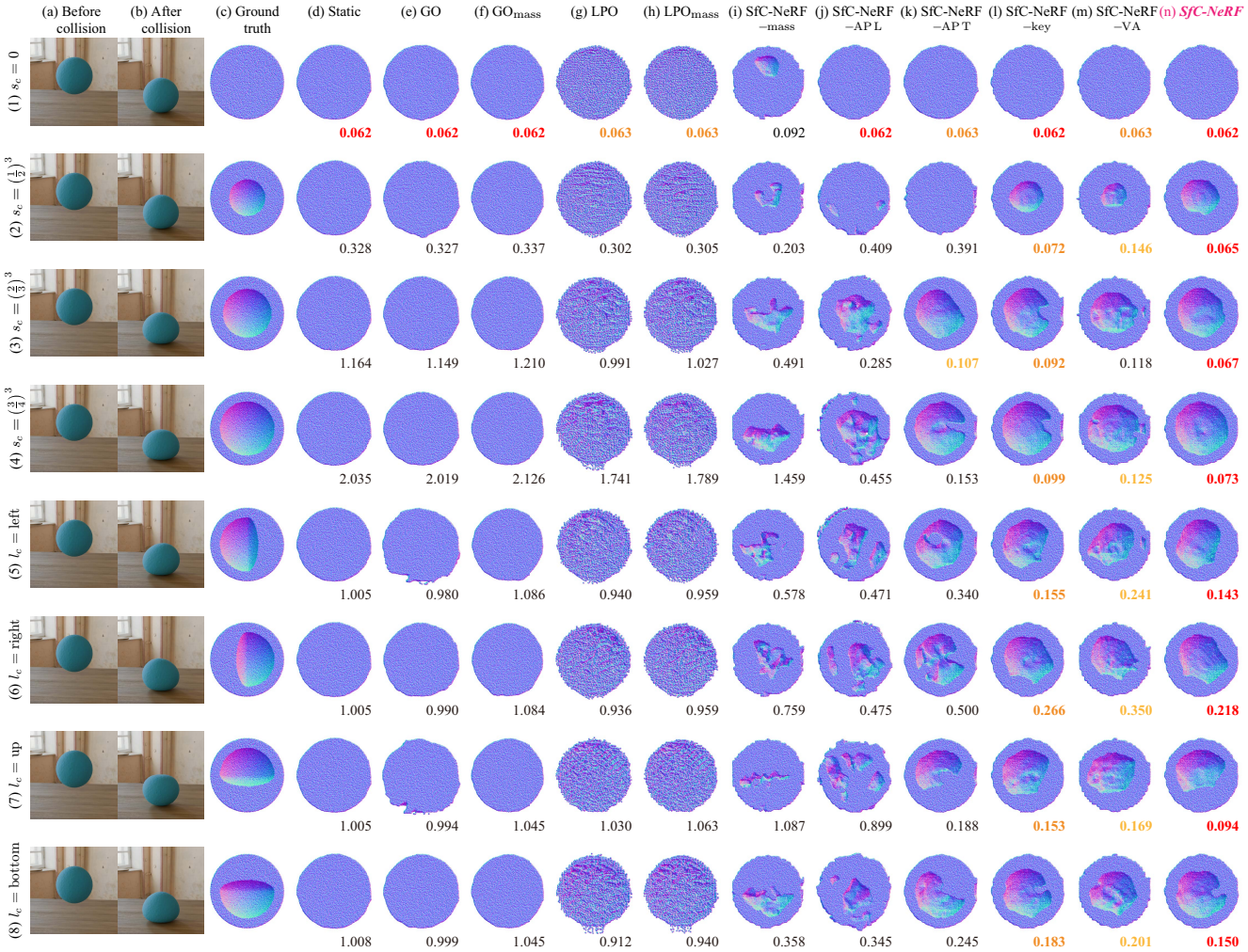

Figure 6. Comparison of learned internal structures for *sphere* objects. *(a) and (b) Examples of training images.* The images are zoomed in for easy viewing. *(a) Examples of training images **before** collision.* As shown in this column, the appearances of the objects are the same across all scenes (1)–(8). Consequently, it is difficult to distinguish the internal structures based solely on these appearances. *(b) Examples of training images **after** collision.* To overcome the difficulty mentioned above, we address *SfC*, in which we aim to identify the internal structures based on appearance changes before and after collision, as shown in (a) and (b). *(c)–(n) Internal structures visualized through cross-sectional views perpendicular to the ground.* In (d)–(n), the score below each image indicates CD ($\times 10^3 \downarrow$). *(c) Ground-truth internal structures.* As shown in this column, although the external appearances are the same in (a), the internal structures are different. *(d) Internal structures learned from the first frames of the video sequences.* The same internal structures (i.e., filled objects) were learned because the appearances were the same before the collision (a). *(e)–(h) Internal structures learned using the baselines (GO- and LPO-based models).* These models struggled to determine optimal learning directions. *(i)–(m) Internal structures learned using the ablated models.* The ablated models are insufficient to prevent convergence to improper solutions. *(n) Internal structures learned using SfC-NeRF (full model).* The full model overcomes the above drawbacks and achieved the best CD.

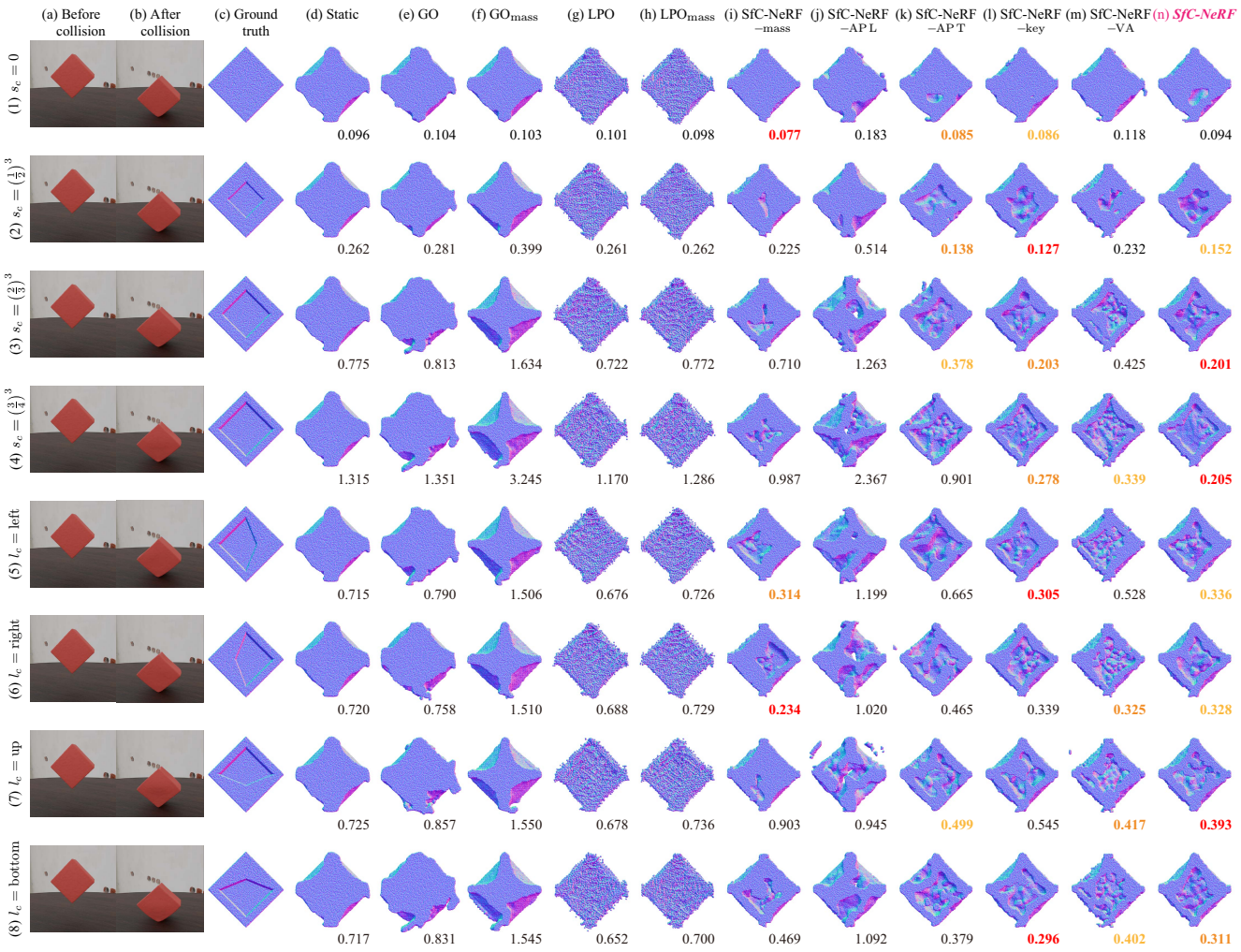

Figure 7. Comparison of learned internal structures for *cube* objects. The view in the figure is the same as that of Figure 6.

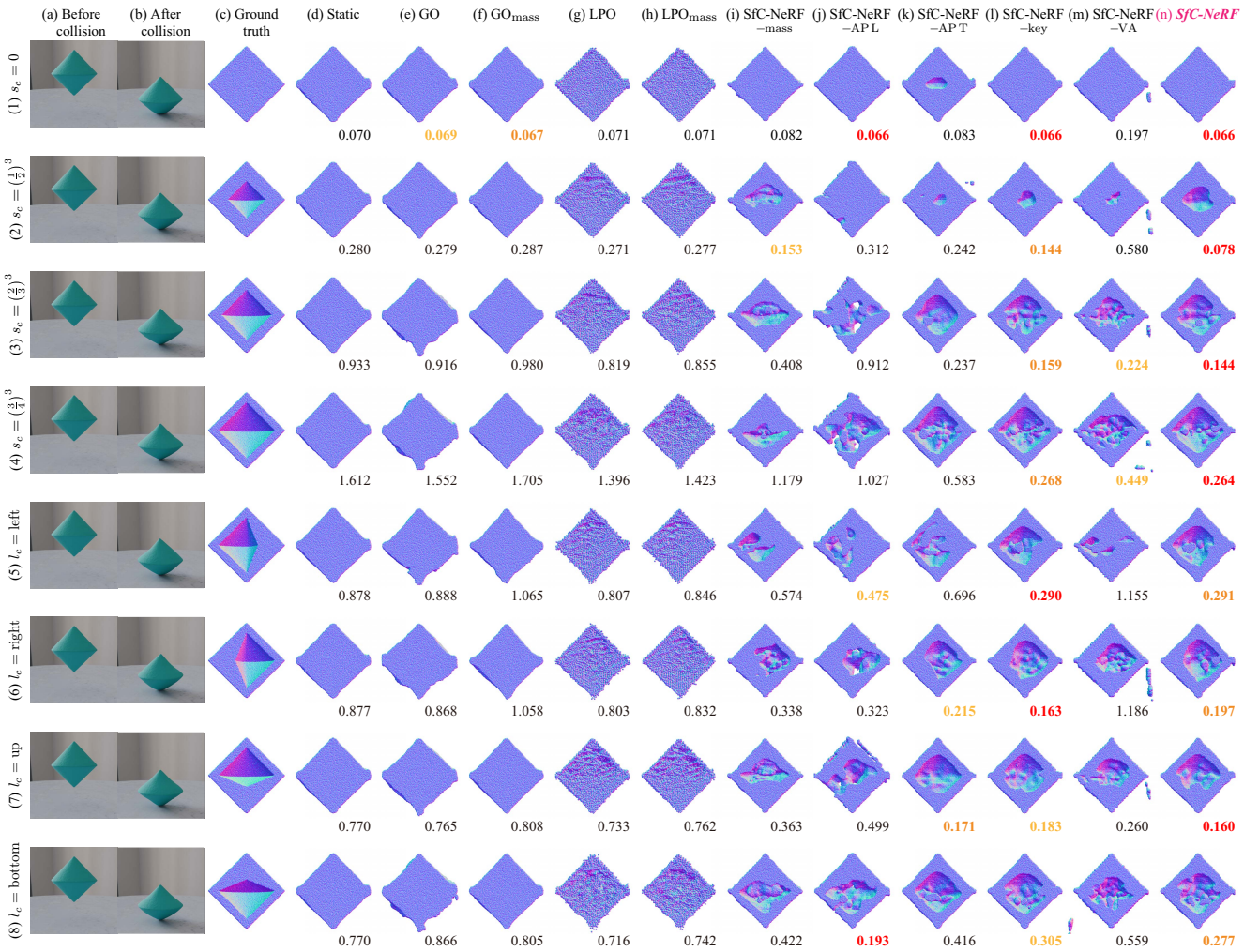

Figure 8. Comparison of learned internal structures for *bicone* objects. The view in the figure is the same as that of Figure 6.

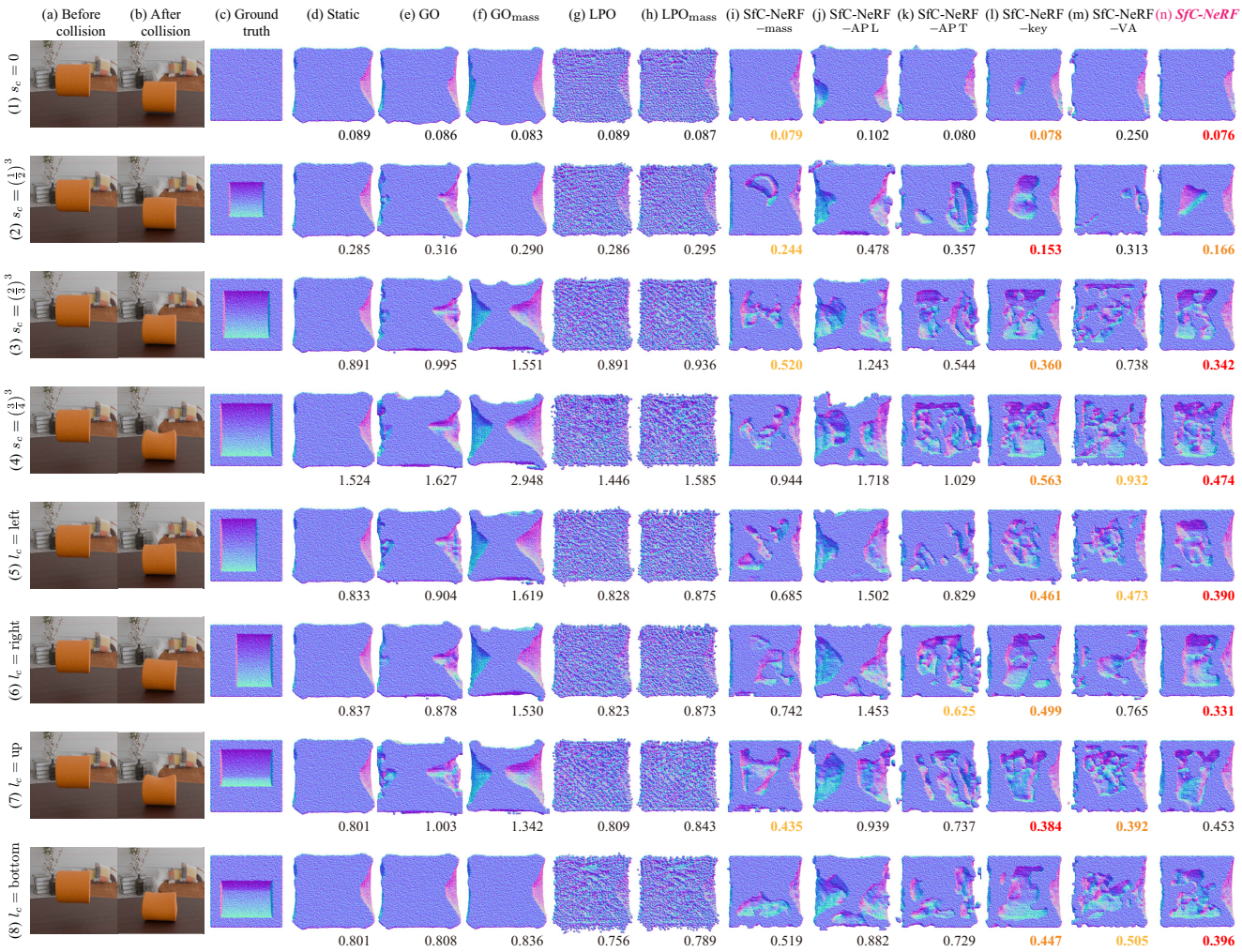

Figure 9. Comparison of learned internal structures for *cylinder* objects. The view in the figure is the same as that of Figure 6.

Figure 10. Comparison of learned internal structures for *diamond* objects. The view in the figure is the same as that of Figure 6.

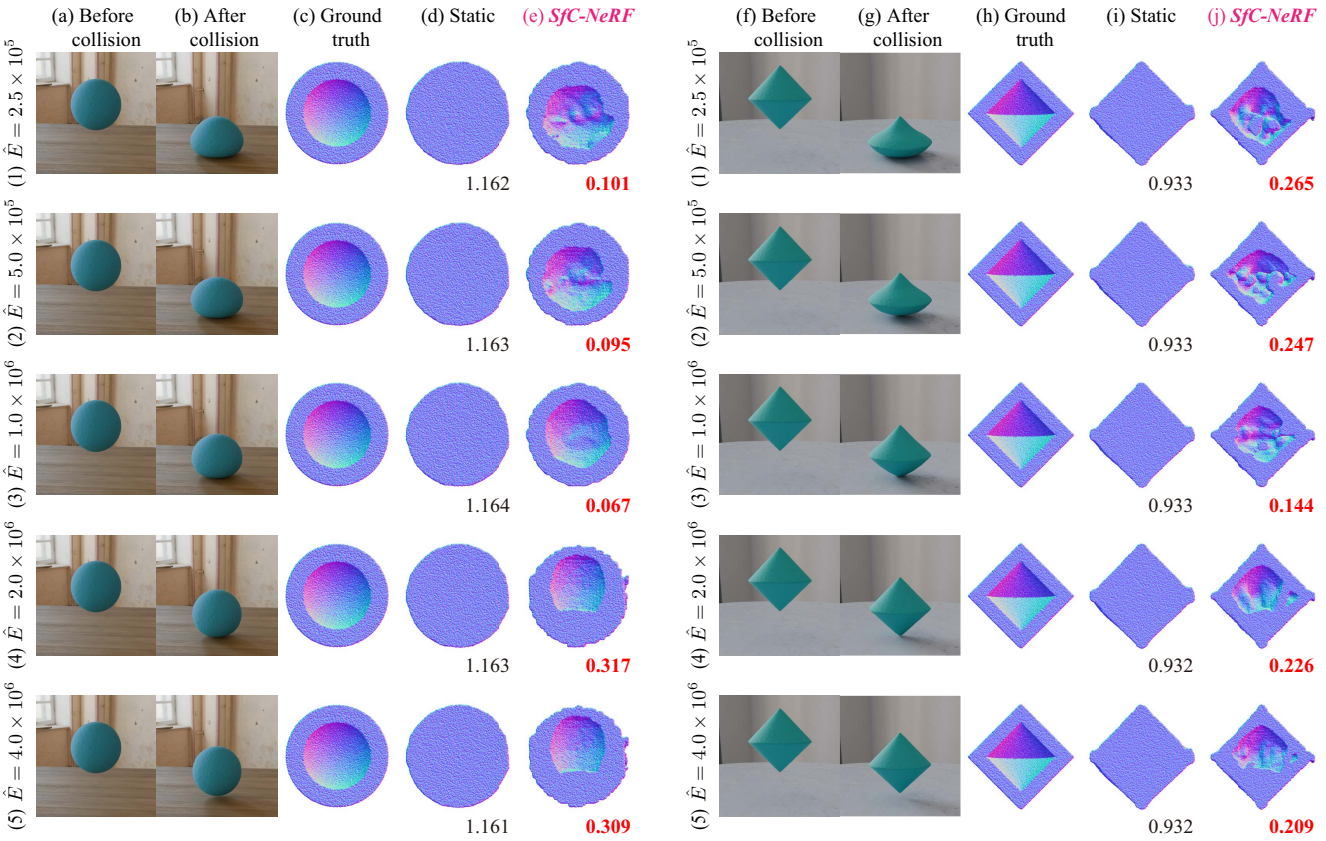

Figure 11. Comparison of learned internal structures for *sphere* objects (left) and *bicone* objects (right) when Young's modulus $\hat{E}$ is varied. Young's modulus is a measure of elasticity and quantifies tensile or compressive stiffness when force is applied. Here, we discuss the results for the sphere objects because the same tendencies were observed for the bicone objects. As shown in (a) and (c), the external appearances before collision (a) and internal structures (c) are the same in all cases (1)–(5). However, as shown in (b), the shapes after collision differ because of variations in Young's modulus $\hat{E} \in \{2.5 \times 10^5, 5.0 \times 10^5, 1.0 \times 10^6, 2.0 \times 10^6, 4.0 \times 10^6\}$. In particular, as Young's modulus increases from top to bottom, the object becomes stiffer, and the amount of shape change decreases. In the Static model (b), the internal structure was learned from the first frame, which looks the same in all cases. As a result, the same internal structure was learned across all variations. In contrast, in SfC-NeRF (e), the internal structure was learned using video sequences with different appearances. In this example, the same internal structure is expected to be learned in all cases. However, the varying appearances after collision (b), which provide a clue for solving the problem, lead to different outcomes. As shown in (1)(b) and (2)(b), when the object is soft, it deforms significantly after collision. This makes it difficult to capture the internal structure consistently, as shown in (1)(e) and (2)(e). In contrast, as shown in (4)(b) and (5)(b), when the object is stiffer, the shape change is limited. This narrows the range within which internal structures can be estimated, as shown in (4)(e) and (5)(e). Because *SfC* is an ill-posed problem with multiple possible solutions, the obtained results are considered reasonable. However, further improvement remains a topic for future work.

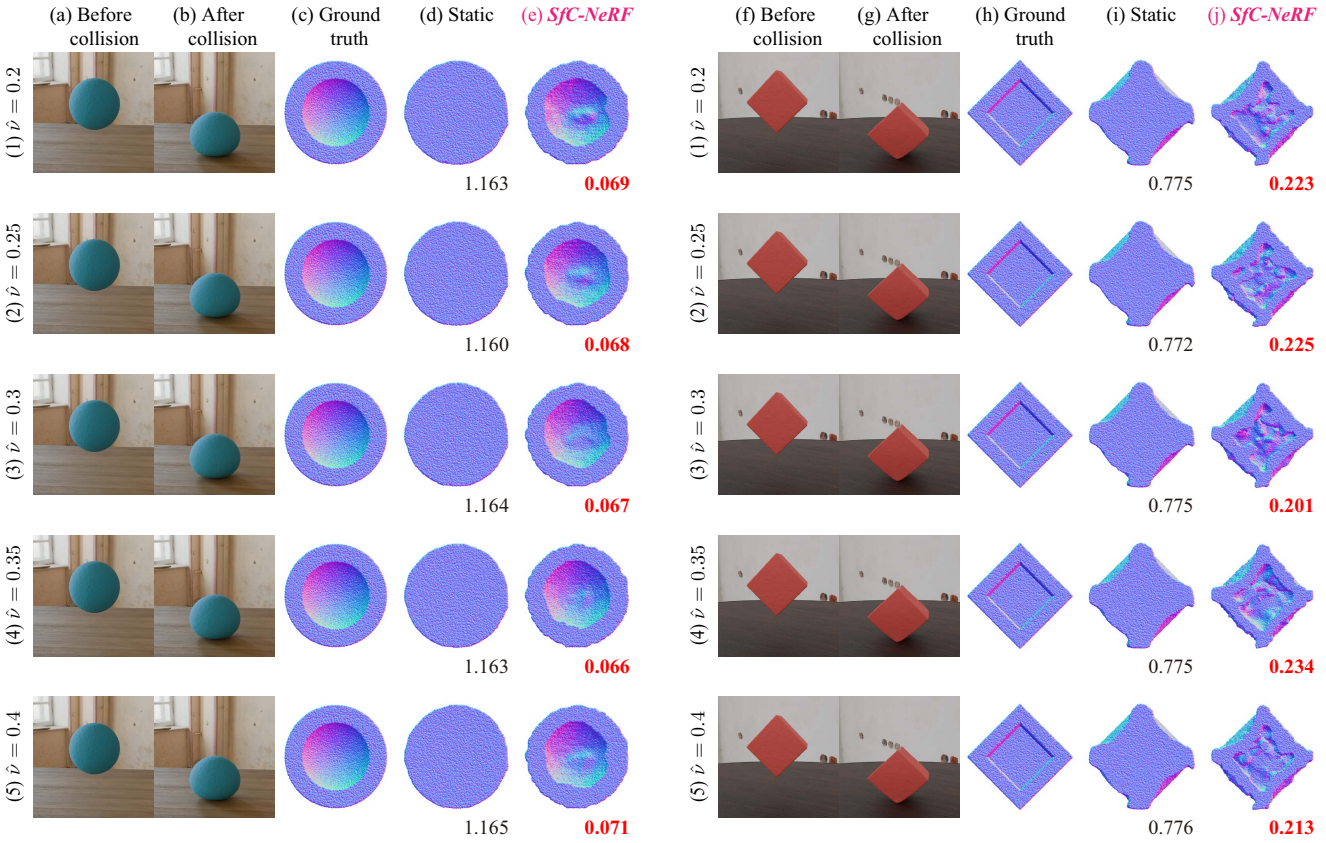

Figure 12. Comparison of learned internal structures for *sphere* objects (left) and *cube* objects (right) when Poisson's ratio $\hat{\nu}$ is varied. Poisson's ratio is a measure of the Poisson effect and quantifies how much a material deforms in a direction perpendicular to the direction in which force is applied. We varied Poisson's ratio $\hat{\nu}$ within the range of values commonly observed in real materials, i.e., $\hat{\nu} \in \{0.2, 0.25, 0.3, 0.35, 0.4\}$. As shown in (b) and (g), this physical property does not significantly affect the appearance after the collision compared to the results when Young's modulus is varied (Figure 11). As a result, the learned internal structures are almost identical, as shown in (e) and (j).

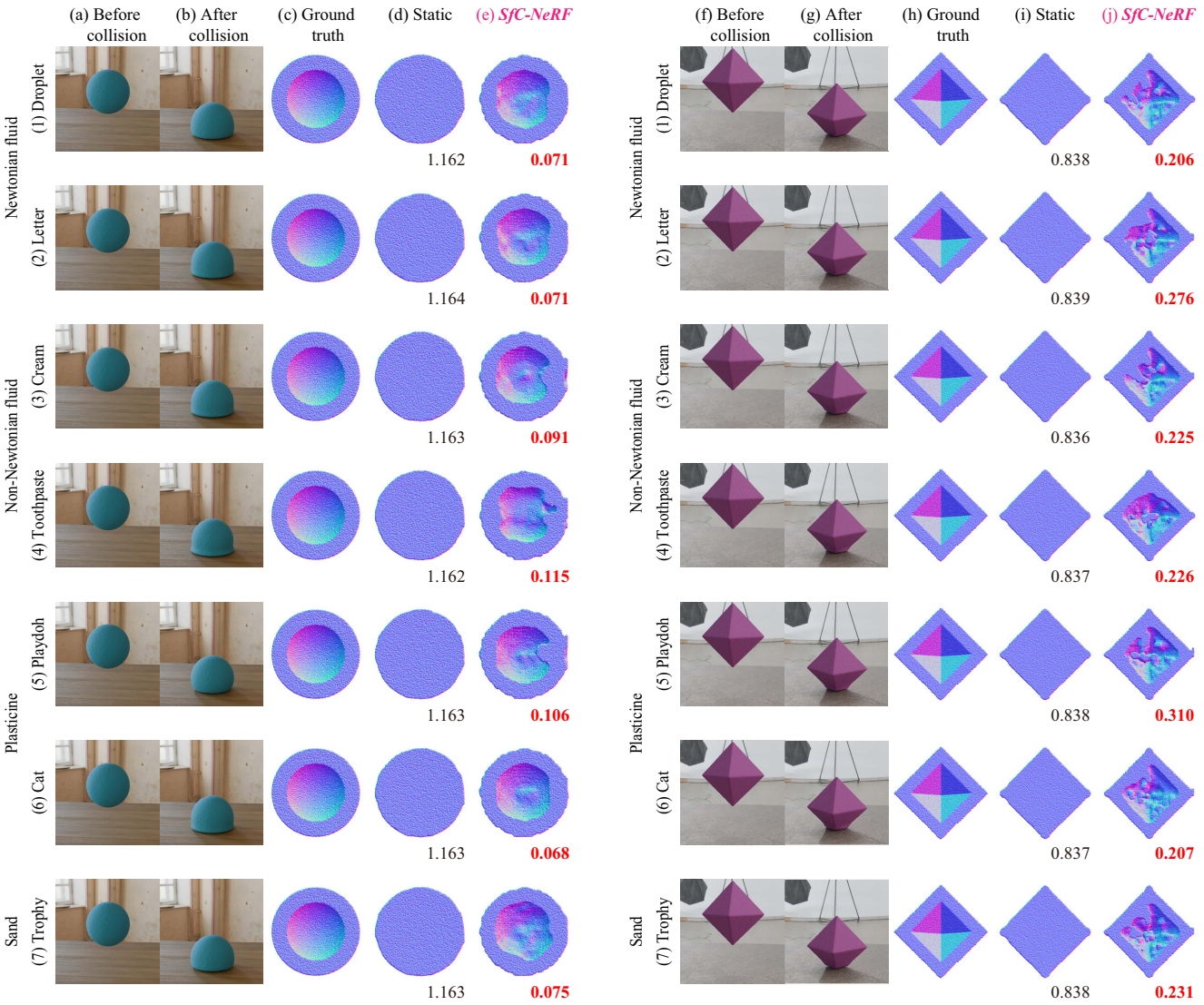

Figure 13. Comparison of learned internal structures for *sphere* objects (left) and *diamond* objects (right) with varying materials. The physical properties were based on the PAC-NeRF dataset [36]. Specifically: (1) Newtonian fluid with the "Droplet" setting (fluid viscosity $\hat{\mu} = 200$ and bulk modulus $\hat{\kappa} = 10^5$). (2) Newtonian fluid with the "Letter" setting ($\hat{\mu} = 100$ and $\hat{\kappa} = 10^5$). (3) Non-Newtonian fluid with the "Cream" setting (shear modulus $\hat{\mu} = 10^4$, bulk modulus $\hat{\kappa} = 10^6$, yield stress $\hat{\tau}_Y = 3 \times 10^3$, and plasticity viscosity $\hat{\eta} = 10$). (4) Non-Newtonian fluid with the "Toothpaste" setting ($\hat{\mu} = 5 \times 10^3$, $\hat{\kappa} = 10^5$, $\hat{\tau}_Y = 200$, and $\hat{\eta} = 10$). (5) Plasticine with the "Playdoh" setting (Young's modulus $\hat{E} = 2 \times 10^6$, Poisson's ratio $\hat{\nu} = 0.3$, and yield stress $\hat{\tau}_Y = 1.54 \times 10^4$). (6) Plasticine with the "Cat" setting ($\hat{E} = 10^6$, $\hat{\nu} = 0.3$, and $\hat{\tau}_Y = 3.85 \times 10^3$). (7) Sand with the "Trophy" setting ($\hat{\theta}_{fric} = 40°$). These results demonstrate that *SfC-NeRF* ((e) and (j)) improves structure estimation compared to Static ((d) and (i)), regardless of the material. However, the improvement rate depends on the material. As an initial approach to address *SfC*, we proposed a general-purpose method. However, it would be interesting to develop methods specifically tailored to individual materials in future work.

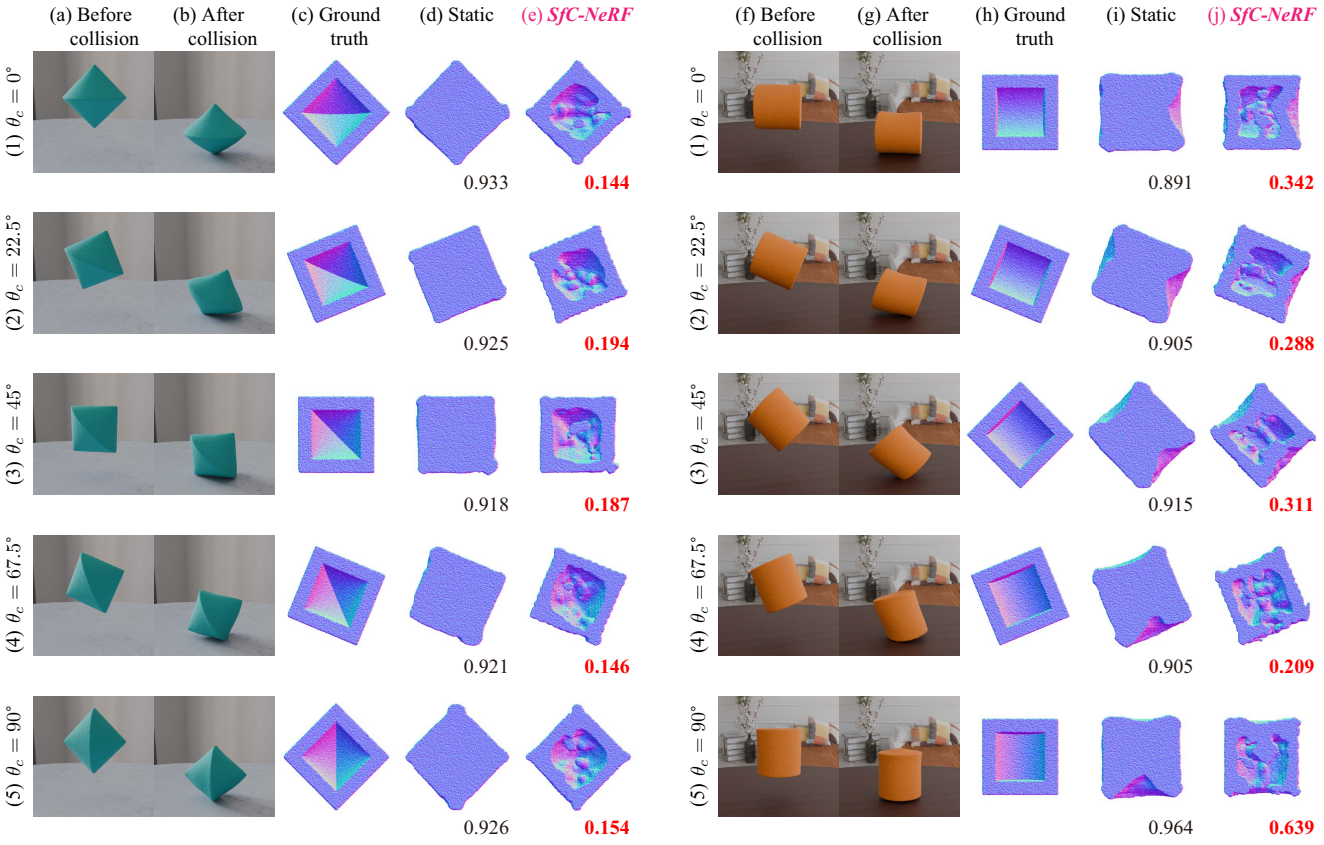

Figure 14. Comparison of learned internal structures for *bicone* objects (left) and *cylinder* objects (right) when collision angle θ_c is varied. We varied collision angle $\theta_c \in \{0°, 22.5°, 45°, 67.5°, 90°\}$. We found that the effect of collision angle on the estimation of the internal structure depends on the object shape. (a)–(e) In the case of an object such as *bicone*, where the object is entirely visible regardless of the collision angle, the estimation performance remains relatively stable across different collision angles. (f)–(j) In contrast, in the case of an object, such as *cylinder*, where the visible area varies greatly depending on the collision angle, the estimation performance also changes with the collision angle. For example, in (5)(g), the bottom of the object is not visible when it collides with the ground. As a result, a hole is generated at the bottom of the object in (5)(j). This issue may be alleviated by improving camera placement. Other possible factors that affect estimation performance are discussed in Appendix A.2.1.

C. Implementation details

C.1. Dataset

Because *SfC* is a new task and no established dataset is available, we created a new dataset called the *SfC dataset* based on the protocol of PAC-NeRF [36], which is a pioneering study on geometry-agnostic system identification. In the main experiments presented in Section 4, we prepared 115 objects by changing their external shapes, internal structures, and materials. Figure 3 shows examples of the data in this dataset. First, we prepared five external shapes: *sphere*, *cube*, *bicone*, *cylinder*, and *diamond*. Regarding the internal structure and material, we set the default values as follows: the cavity size rate for the filled object, s_c, was set to $(\frac{2}{3})^3$, the cavity location, l_c, was set to the center, and the material was defined as an elastic material with Young's modulus $\hat{E} = 10^6$ and Poisson's ratio $\hat{\nu} = 0.3$. Under these default properties, one of them was changed as follows:

(a) Three different sized cavities: $s_c \in \{0, (\frac{1}{2})^3, (\frac{3}{4})^3\}$.

(b) Four different locations of cavities: the center l_c is moved $\{\text{up}, \text{down}, \text{left}, \text{right}\}$.

(c-1) Eight different elastic materials: those with four different Young's moduli $\hat{E} \in \{2.5 \times 10^5, 5 \times 10^5, 2 \times 10^6, 4 \times 10^6\}$ and four different Poisson's ratios $\hat{\nu} \in \{0.2, 0.25, 0.35, 0.4\}$.

(c-2) Seven different materials: two Newtonian fluids, two non-Newtonian fluids, two plasticines, and one sand. Their physical properties were derived from the PAC-NeRF dataset [36]. Specifically, the two Newtonian fluids included one with the "Droplet" setting (fluid viscosity $\hat{\mu} = 200$ and bulk modulus $\hat{\kappa} = 10^5$) and one with the "Letter" setting ($\hat{\mu} = 100$ and $\hat{\kappa} = 10^5$). The two non-Newtonian fluids included one with the "Cream" setting (shear modulus $\hat{\mu} = 10^4$, bulk modulus $\hat{\kappa} = 10^6$, yield stress $\hat{\tau}_Y = 3 \times 10^3$, and plasticity viscosity $\hat{\eta} = 10$) and one with the "Toothpaste" setting ($\hat{\mu} = 5 \times 10^3$, $\hat{\kappa} = 10^5$, $\hat{\tau}_Y = 200$, and $\hat{\eta} = 10$). The two plasticines included one with the "Playdoh" setting (Young's modulus $\hat{E} = 2 \times 10^6$, Poisson's ratio $\hat{\nu} = 0.3$, and yield stress $\hat{\tau}_Y = 1.54 \times 10^4$) and one with the "Cat" setting ($\hat{E} = 10^6$, $\hat{\nu} = 0.3$, and $\hat{\tau}_Y = 3.85 \times 10^3$). The sand had the "Trophy" setting ($\hat{\theta}_{fric} = 40°$).

Thus, we created 5 external shapes $\times$ (1 default + 3 sizes + 4 locations + $(8 + 7)$ materials) = 115 objects.

We also prepared 20 objects for the extended experiments described in Appendix A.2. Specifically, we considered four collision angles: $\theta_c \in \{22.5°, 45°, 67.5°, 90°\}$. Thus, in this appendix, we created 5 external shapes $\times$ 4 collision angles = 20 objects. The total number of objects created in the main text and this appendix is 115+20 = 135.

Following the PAC-NeRF study [36], the ground-truth data were generated using the MLS-MPM simulator [25], where each object fell freely under the influence of gravity and collided with the ground plane. Images were rendered under various environmental lighting conditions and ground textures using a photorealistic renderer. Each scene was captured from 11 viewpoints using cameras spaced in the upper hemisphere including an object.

C.2. Model

We implemented the models based on the official PAC-NeRF code [36].[6] PAC-NeRF represents an Eulerian grid-based scene representation using voxel-based NeRF (specifically, direct voxel grid optimization (DVGO) [63]) and conducts a Lagrangian particle-based differentiable physical simulation using a differentiable MPM simulator (specifically, DiffTaichi [26]). More specifically, DVGO represents a volume density field $\sigma^{G'}$ using a 3D dense voxel grid and represents a color field $\mathbf{c}^{G'}$ using a combination of a 4D dense voxel grid and a two-layer multi-layer perceptron (MLP) with a hidden dimension of 128. When the MLP is employed, positional embedding in the viewing direction $\mathbf{d}$ is used as an additional input. We set the resolutions of $\sigma^{G'}$ and $\mathbf{c}^{G'}$ to match those in PAC-NeRF [36].

C.3. Training settings

We performed static optimization (Figure 2(i)) using the same settings as those used for PAC-NeRF. Specifically, we trained the model for 6000 iterations using the Adam optimizer [33] with learning rates of 0.1 for the volume density and color grids and a learning rate of 0.001 for the MLP. The momentum terms β_1 and β_2 were set to 0.9 and 0.999, respectively. In the dynamic optimization (Figure 2(ii)), we trained the model for 1000 iterations using the Adam optimizer [33] with a default learning rate of 6.4 for the volume density grid. The momentum terms β_1 and β_2 were set to 0.9 and 0.999, respectively. We found that a high learning rate is useful for efficiently reducing the volume density; however, this is not necessary when the estimated mass m sufficiently approaches the ground-truth mass $\hat{m}$. Therefore, we divided the learning rate by 2 (with a minimum of 0.1) as long as the estimated mass m was below the ground-truth mass $\hat{m}$. Conversely, we multiplied the learning rate by 2 (with a maximum of 6.4) as long as the estimated mass m exceeded the ground-truth mass $\hat{m}$.

We conducted volume annealing every 100 iteration during the dynamic optimization. When the estimated mass m was significantly larger than the ground-truth mass $\hat{m}$ (specifically, when the difference exceeded 10 in practice), the expansion process was skipped to prevent m from deviating further from $\hat{m}$.

In appearance-preserving training, static optimization was performed using settings similar to those mentioned

[6] https://github.com/xuan-li/PAC-NeRF

above (i.e., static optimization in Step (i) (Figure 2(i))), but the number of iterations was reduced to 10.

We empirically set the hyperparameters for the full objective $\mathcal{L}_{\text{full}}$ (Equation 12) to $\lambda_{\text{mass}} = 1$, $\lambda_{\text{pres}} = 100$, $w_{\text{depth}} = 0.01$, and $\lambda_{\text{key}} = 10$. The hyperparameter for background loss $\mathcal{L}_{\text{bg}}$ was set to $w_{\text{bg}} = 0.2$.

C.4. Evaluation metrics

As mentioned in Section 3.1, we use particles $\mathcal{P}^P(t_0)$ to represent the structure (including the internal structure) of an object and estimate $\mathcal{P}^P(t_0)$ to match the ground-truth particles $\hat{\mathcal{P}}^P(t_0)$. Therefore, we evaluated the model by measuring the distance between $\mathcal{P}^P(t_0)$ and $\hat{\mathcal{P}}^P(t_0)$ using the *chamfer distance (CD)*. The smaller the value, the higher the degree of matching. As mentioned in Section 4.3, we also used the *anti-chamfer distance (ACD)*, which is the chamfer distance between the predicted particles $\mathcal{P}^P(t_0)$ and ground-truth particles $\tilde{\mathcal{P}}^P(t_0)$, where the cavity was placed on the opposite side, to evaluate the capture of the cavity location.