

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 TOPOGAUSSIAN: INFERRING INTERNAL TOPOLOGY STRUCTURES FROM VISUAL CLUES

Anonymous authors

Paper under double-blind review

ABSTRACT

We present TopoGaussian,¹ a particle-based holistic pipeline for inferring the interior structure of an opaque object with easily accessible photos and videos as input. Traditional mesh-based approaches require tedious mesh filling and fixing process with high error rate, while output rough boundary surface. Our pipeline combines Gaussian splatting with a novel particle-based differentiable simulator for objects, without interference with mesh. Based on the gradients from this simulator, we provides flexible choice of topology representation for optimization, including particle, neural implicit surface, and quadratic surface. The resultant pipeline supports modeling, simulation, and optimization of the interior geometry in a heterogeneous object on a unified point-cloud representation. We present a synthetic dataset, and showcase the results of our pipeline on it. Moreover, we perform experiments on four real-world tasks, and further 3D print the output to validate the efficacy of our methods. Compared with existing mesh-based method, our pipeline is 5.26 times faster, with 2.33 times quality of reconstructed boundary in average. These results highlight the potential of our pipeline in 3D-vision, soft-robotics, and manufacturing applications.

1 INTRODUCTION

Seeing through the surfaces of an opaque object is an intriguing idea contemplated by scientists, engineers, and science-fiction writers. The capability to infer the internal structure of an object from its surface information unlocks many creative applications, e.g., 3D model design and industrial manufacturing. Traditional approaches from scientific or engineering disciplines either require intrusive probing sensors or operate on expensive computed tomography (CT) or magnetic resonance imaging (MRI) equipment, underscoring the challenge of this problem. Recent developments in 3D vision provide powerful tools for reconstructing 3D scenes with easily accessible, non-intrusive visual input only, e.g., NeRF Wang et al. (2021) or Gaussian Splatting Kerbl et al. (2023) with photos or videos as input. However, these methods only recover geometrically correct and visually plausible *surface* information of objects with no or hallucinated interior geometry.

This paper aims to develop a holistic pipeline for inferring the interior geometry of an opaque object in order to recover its major physical characteristics with easily accessible visual input (photos and videos) only. Our pipeline, TopoGaussian, begins with collecting multi-view images and uses Gaussian splatting to obtain a point cloud encoding the object’s exterior surface shape and appearance Kerbl et al. (2023). We construct a volumetric point cloud by filling interior points Xie et al. (2023), associated with an independent physical parameter each point, and provide two options to represent the interior geometry. One option depict the boundary directly based on the physical parameter of particles, inspired by standard topology optimization methods on voxels Sigmund (2001) and particles Li et al. (2021); the other option leverages neural implicit surface Park et al. (2019) to represent the interior boundary surface. The particle-based topology optimizer enables a flexible and expressive representation of the interior geometry, while neural implicit surface representation outputs more smooth surface. To accommodate the particle representation, we develop a novel particle-based differentiable simulator for rigid and soft objects with hollowed structures to evaluate their motions in simulation. Finally, we compare the simulation results and the object’s dynamic

¹[TODO: Add a github site]

054 motion in a video. With access to the simulation gradient, we run gradient-based methods to optimize each particle’s decision variable and obtain an interior geometry that leads to similar motions
 055 established in the input video. Our pipeline presents a mesh-free representation for flexible and
 056 expressive modeling, simulation, and optimization of heterogeneous objects gracefully unified in a
 057 holistic framework of particles only. Compared with traditional, intrusive methods for recovering
 058 internal geometries, our non-intrusive pipeline requires easily accessible images and videos as input,
 059 broadening the applications of modern scene-reconstruction methods in 3D vision.
 060

061 We design several synthetic tasks and four real-world tasks to demonstrate the efficacy of our
 062 pipeline. The rigid body tasks read static images or dynamic videos of their motions as input.
 063 Our pipeline can infer their internal geometries and give correct predictions of their physical prop-
 064 erties (e.g., hollowness or motion frequencies) that closely match the static or dynamic behaviors
 065 established in the input video. The soft body tasks consider different heterogeneous objects, in-
 066 cluding inserted hard part and pneumatically actuated hollow, with a video of its motion as input.
 067 Our pipeline infers a shape of the internal heterogeneous part that can reconstruct similar motions.
 068 Based on the result of the optimization, we perform 3D-print to manufacture a real object whose
 069 internal structure is predicted by our pipeline. These tasks showcase the efficacy of our pipeline in
 070 internal geometry reconstruction with non-intrusive visual inputs, enabling novel applications in 3D
 071 vision and soft robotics.

072 In summary, this work makes the following contributions:

- 073 • We develop a novel, non-intrusive pipeline for inferring the internal geometrical structures
 074 of solid objects from visual inputs only, i.e., multi-view images and videos of motion;
- 075 • We present particle-based differentiable simulation, combining with topology optimization
 076 and neural implicit surface methods for rigid and pneumatically actuated deformable ob-
 077 jects. Together with Gaussian splatting, our simulation and optimization methods lead to a
 078 completely meshless pipeline.
- 079 • We demonstrate the efficacy of our pipeline by showcasing several applications in 3D vision
 080 and soft robotics, including inferring internal geometries of various rigid and soft bodies,
 081 and manufacture the result using 3D-print to validate our result.

084 2 RELATED WORK

086 **3D scene reconstruction** Reconstructing 3D scenes from visual inputs is a fundamental problem
 087 extensively studied in computer vision for decades. Recent representative methods for this prob-
 088 lem include NeRF (Wang et al., 2021) and Gaussian splatting (Kerbl et al., 2023). NeRF bases
 089 its scene geometry representation on implicit surfaces learnable from multi-view images, with sev-
 090 eral improvements in its speed, quality, and applicability in dynamic and deformable objects (Bar-
 091 ron et al., 2021; Deng et al., 2022; Pumarola et al., 2021; Martin-Brualla et al., 2021). Gaussian
 092 splatting (Kerbl et al., 2023) is optimization-based and built upon a point cloud representation, and
 093 follow-up studies have proposed various improvements in its efficiency and reconstruction qual-
 094 ity (Chen et al., 2023; Matsuki et al., 2023; Tang et al., 2023; Wu et al., 2023; Yi et al., 2023). In
 095 particular, Xie et al. (2023) showcased novel synthesis of motions from editing a dynamic model
 096 recovered from Gaussian splatting, and Guo et al. (2024) reconstructs physically compatible objects
 097 using a quasi-static simulation. However, these works primarily focus on reconstructing visually
 098 and dynamically plausible *surface* information of an object without inferring its interior structure.

099 **Differentiable simulation** Differentiable simulation extends traditional simulation with gradient
 100 information and revives gradient-based optimization in graphics, learning, and robotics applica-
 101 tions (Yu et al., 2023; Werling et al., 2021; Xu et al., 2021; Lin et al., 2021; Qiao et al., 2021; Li
 102 et al., 2022). Previous studies have developed differentiable simulators for various physical systems
 103 including rigid objects (Geilinger et al., 2020; Wang et al., 2019), deformable solids (Liu et al.,
 104 2023; Gjoka et al., 2022), cloths (Yu et al., 2023), and fluids (Li et al., 2024). Many of these dif-
 105 ferentiable simulators are mesh- or finite-element-based due to their benefits in handling irregular
 106 shapes with an explicit sharp boundary representation. In contrast to these works, we focus on a
 107 purely particle-based differentiable simulator for hollowed shapes in accommodation with Gaussian
 108 splatting representation.

Topology optimization Topology optimization is an optimization-based structure design technique originating from structural and mechanical engineering (Sigmund, 2001; Eschenauer & Olhoff, 2001; Wu et al., 2021; Deaton & Grandhi, 2014). Traditionally, it considers a density-field representation of a structural geometry on a regular grid and optimizes independent decision variables at each voxel for optimal structural or mechanical performance, e.g., minimum compliance (Sigmund, 2001; Yang & Chen, 1996; Liu et al., 2018). Previous studies have also explored the idea with different geometric representations, e.g., level sets (Wang et al., 2003) and particle (Li et al., 2021). Recent years, neural implicit surface method has been proposed (Park et al., 2019), which encodes the shape with a deep neural network. Moreover, for rigid bodies, Hafner et al. (2024) has proved that quadratic surfaces are enough for optimization problems that only involves with the rigid physical characteristics. Existing topology optimization works primarily focus on applications in engineering disciplines, e.g., structural and mechanical analysis, aeronautics, and architecture (Jensen & Sigmund, 2011; Zhu et al., 2016; Dühring et al., 2008). We favor topology optimization over alternative shape optimization techniques due to its mesh-free property, and we explore the applicability of its core idea on point clouds from Gaussian splatting in 3D computer vision tasks.

3 METHOD

3.1 OVERVIEW

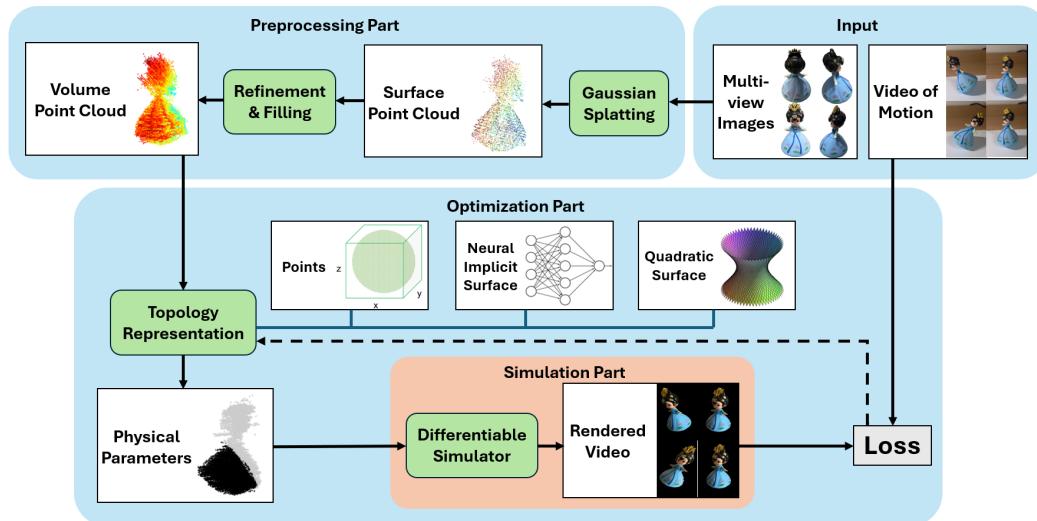


Figure 1: Pipeline overview (Sec. 3.1): our pipeline takes as input multi-view photos of the opaque object and its video of motion. We run Gaussian splatting on multi-view images to obtain a point cloud characterizing its surface geometry and appearance. Next, we refine and fill in internal points to obtain a volumetric point cloud, and add our topology representation (with 3 flexible choices) on it to attach physical parameters on it. We then simulate the volumetric point cloud with our particle-based differentiable simulator, which compares its motion with a reference video and backpropagates the gradient to the topology representation. Finally, we perform the optimization algorithm and obtain the result structure.

The overarching goal of our pipeline is to infer the internal topology structure of an opaque object to match its dynamic motion from a reference video. Our pipeline contains four steps: preprocessing (Sec. 3.2), which takes as input multi-view photos of a possibly hollowed opaque object and generates a densely sampled volumetric point cloud; topology representation (Sec. 3.3), which encodes the topology structure with flexible choices and attach the physical parameters on the point cloud; simulation (Sec. 3.4), which evaluates the dynamic motion of the point cloud for a given interior shape design configured by each point’s emptiness; optimization (Sec. 3.5), which runs gradient-based optimization to adjust each the parameters in the topology representation, leading to a shape that replicates the dynamic motion of the object in a reference video. Fig. 1 gives an overview of the pipeline.

162 3.2 PREPROCESSING
 163

164 We begin by collecting multi-view photos of the object and running Gaussian splatting due to its
 165 efficient and high-quality reconstruction of 3D objects from images. The output point cloud from
 166 Gaussian splatting gives visually and geometrically plausible surface information about the object.
 167 However, as we focus on inferring its interior structure, a surface geometrical representation is
 168 insufficient. Following PhysGaussian Xie et al. (2023), we refine and fill in points to the output
 169 point cloud from Gaussian splatting, leading to a clean and densely sampled volumetric point cloud
 170 $\{(\mathbf{X}_i, V_i)\}_{i=1}^N$ where N denotes the number of points, \mathbf{X}_i the their 3D locations, and V_i the volume
 171 associated with \mathbf{X}_i . The point cloud can be viewed as a particle-based discretization of the solid
 172 object.

173 3.3 TOPOLOGY REPRESENTATION
 174

175 We needs to encode the topology structure into a numerical representation which attaches the phys-
 176 ical parameters on the point cloud. We provides 3 flexible choices in our pipeline, and these choices
 177 can be divided into two parts, explained in Sec. 3.3.1 and Sec. 3.3.2 respectively.

179 3.3.1 POINT REPRESENTATION
 180

181 For each point i in the point cloud, we assign an optimizing variable t_i on it. Then we use a
 182 sigmoid function to map t_i to r_i , where r_i equals to either 0 or 1, representing two different physical
 183 parameters, while the boundary surface is the boundary of $r_i = 0$ and $r_i = 1$.

185 3.3.2 SURFACE REPRESENTATION
 186

187 We use a signed distance function (SDF) $s(\mathbf{x})$ to encode the boundary surface of the, where \mathbf{x} is any
 188 position inside the object, and $s(\mathbf{x})$ outputs the signed distance from this position to the boundary
 189 surface. Then the surface is the zero level set ($s(\mathbf{x}) = 0$) of the SDF, and we can assign two physical
 190 parameters to two sides of the surface respectively. In order to numerically represent the SDF, we
 191 provides two choices.

192 **Neural Implicit Surface** DeepSDF Park et al. (2019) proposed that we can use a neural network
 193 to represent the SDF. More concretely, we use a deep feed-forward network, which reads the position
 194 as intput and output the SDF value.

197 **Quadratic Surface (for rigid body)** Spin-it Faster Hafner et al. (2024) proved that for topology
 198 optimization of rigid body that only considers its rigid physical properties (mass, center of mass,
 199 and inertia), a simple quadratic function is enough to represent the optimal solution. To be more
 200 specific, if we write $\mathbf{x} = [x_1, x_2, x_3]$ the SDF can be written in

$$s(\mathbf{x}) = a_1 + a_2 x_1 + a_3 x_2 + a_4 x_3 + a_5 x_1 x_2 + a_6 x_2 x_3 + a_7 x_3 x_1 + a_8 x_1^2 + a_9 x_2^2 + a_{10} x_3^2 \quad (1)$$

203 with only 10 parameters (a_1 to a_{10}).

205 3.4 SIMULATION
 206

207 The next step in our pipeline involves simulating the dynamic motion of the opaque object. Our
 208 work focuses on rigid and soft solid objects, both of which have mature simulation solutions based
 209 on (finite-element) mesh representations Sifakis & Barbic (2012); Sin et al. (2013); Lee et al. (2018).
 210 However, as we pointed our, it is tedious to perform mesh extracting, fixing, and filling. Therefore,
 211 our pipeline proposes to consider a fully particle-based simulator instead and aims to build a com-
 212 pletely mesh-free pipeline with a unified particle-base representation shared by all steps: reconstruc-
 213 tion, simulation, and optimization. We note that previous works like PhysGaussian Xie et al. (2023)
 214 have discussed a similar motivation behind a fully particle-based pipeline, but we are faced with
 215 extra technical challenges from representing the internal geometric structure in the same particle
 system. We describe the technical details of our simulator below.

216 3.4.1 CONTINUOUS BODY SIMULATION
 217

218 **Rigid-body Simulation** When representing a rigid body with the particle system, the dynamic
 219 motion is governed by the Newton-Euler equation Lanczos (2012); Liu & Jain (2012) on six DoFs
 220 only. Specifically, the governing equation requires access to the mass m , and the center of mass \mathbf{c} ,
 221 and the moment of inertia \mathbf{I} , all of which can be computed from $\{(\mathbf{X}_i, m_i)\}$ in a straightforward
 222 manner. For example, the mass $m = \sum_i m_i$, and the center of mass \mathbf{c} can be computed as

$$\mathbf{c} = \frac{\sum_i m_i \mathbf{X}_i}{m}. \quad (2)$$

223 Similarly, the moment of inertia can be computed from
 224

$$\mathbf{I} = \sum_i -m_i [\mathbf{X}_i - \mathbf{c}] [\mathbf{X}_i - \mathbf{c}], \quad (3)$$

225 where $[\cdot]$ maps a vector to its cross-product (skew-symmetric) matrix.
 226

227 With these definitions at hand, we can simulate the rigid body with a standard time integration
 228 scheme (e.g., RK2) discretizing the Newton-Euler equation. Note that the hollowed shape affects
 229 the rigid-body dynamics via modulating s_i between 0 and 1 and therefore the mass m_i carried by
 230 each particle.
 231

232 **Soft-body Simulation** Following previous works on particle-based deformable-body simulation
 233 Müller et al. (2005), we consider simulating a soft body made of hyperelastic materials. The critical
 234 step is to estimate the deformation gradient \mathbf{F}_i , a 3×3 tensor characterizing the local deformation
 235 around each \mathbf{X}_i . Here, we follow the standard practice in particle-based simulation for deformable
 236 solids and fluids Müller et al. (2005; 2022) to construct \mathbf{F}_i . More concretely, given a *deformed* point
 237 cloud $\{\mathbf{x}_i\}$ during simulation and for any point \mathbf{x}_i , we estimate its \mathbf{F}_i with the method in Becker
 238 et al. (2009):
 239

$$\mathbf{F}_i = \mathbf{I} + \left(\sum_j V_j (\mathbf{R}_i^\top (\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{X}_j - \mathbf{X}_i)) \nabla W(\mathbf{X}_i - \mathbf{X}_j) \right)^T, \quad (4)$$

240 where j loops over particles in a neighborhood of \mathbf{X}_i and \mathbf{R}_i^\top estimated a rotation transformation
 241 based on the relative poses of these neighboring particles Becker et al. (2009).
 242

243 With the definition of \mathbf{F}_i at hand, we solve the following incremental potential Martin et al. (2011)
 244 problem to time-step the dynamic system implicitly at each time step k :
 245

$$\min_{\{\mathbf{x}_i^{k+1}\}} \frac{1}{2} \sum_i m_i (\mathbf{x}_i^{k+1} - \hat{\mathbf{x}}_i^k)^2 + \int \Psi(\mathbf{F}_i), \quad (5)$$

246 where $\hat{\mathbf{x}}_i^k$ denotes constant values computed at the beginning of the k -th time step, and Ψ the strain
 247 energy density function from a user-specified hyperelastic material. We apply the standard Newton's
 248 method to time-step the system with Eqn. (5).
 249

250 3.4.2 PNEUMATIC ACTUATION
 251

252 One important extension of our soft body simulator is the support for pneumatic actuation. soft
 253 finger in soft robotics Li et al. (2019); Miyashita et al. (2016). In particular, the pneumatic chamber
 254 embedded in the soft body is typically characterized by a customized chamber shape design invisible
 255 from the soft body's exterior surface. We note that the air pressure force induced by the pneumatic
 256 actuator is the conservative force from the volume energy Luo et al. (2022):
 257

$$E_a(\{\mathbf{x}_i\}) = \int_{\Omega\{\mathbf{x}_i\}} p dv, \quad (6)$$

258 where p denotes the air pressure. Note that the integral domain is the volumetric region of the
 259 deformed shape. With \mathbf{F}_i at hand, we can approximate E_p as
 260

$$E_a \approx \sum_i p v_i = \sum_i p |\mathbf{F}_i| V_i, \quad (7)$$

261 where $|\mathbf{F}_i|$ represents the determinant of \mathbf{F}_i . Therefore, to support pneumatically actuated soft-body
 262 simulation, we add the extra term E_a into the incremental potential energy (Eqn. (5)) and solve it
 263 with Newton's method as before.
 264

270 3.4.3 COLLISION
271

272 Based on existing study on collision simulation Kim & Eberle (2020), we handle collision using
273 a penalty-based method. To be more specific, for each particle of the soft body, we add a penalty
274 energy

$$275 \quad E_{pi} = \frac{1}{2}k(\min\{\delta - d_i, 0\})^2 \quad (8)$$

276 to force it away from the obstacles. In Eqn. (8), k is the penalty stiffness, δ is a small translation
277 parameter to avoid penetration, and d_i is the signed distance between particle i and the obstacle (eg.
278 ground). Then in order to simulate the collision of the whole object (either rigid or soft), we add the
279 sum of the penalty energy from all points into the incremental potential.

280 3.4.4 DIFFERENTIABILITY
281

282 We apply the chain rule and the standard adjoint equation Du et al. (2021); Geilinger et al. (2020);
283 Hahn et al. (2019); McNamara et al. (2004) to equip the proposed particle-based simulator with
284 gradient information. Detailed derivations of the gradients in rigid-body and deformable-body sim-
285 ulators can be found in previous works, e.g., ChainQueen Hu et al. (2019). For the newly introduced
286 volume energy and collision penalty energy, the critical steps in deriving its gradients involve gra-
287 dients with respect to the deformation gradient F_i and the signed distance d_i . The former has been
288 covered in the gradient computation in Ψ , while the latter is direct under Euclidean distance.

289 3.5 OPTIMIZATION
290

291 The final step of our pipeline is a gradient-based optimization process that infers the topology struc-
292 ture from the loss of the particles comparing the simulated motion with the reference video. We use
293 L-BFGS-B and Adam optimizer to optimize the parameters in our topology representation.

294 4 RESULTS
295296 4.1 SYNTHETIC EXPERIMENTS
297

300 **Experiment Setting** We perform experiment on our synthetic dataset to evaluate our method.
301 Rigid bodies with different shapes are hung by a string at a given position, and our goal is to optimize
302 the topology structure of their inside hole to keep it horizontal. As this is a static equilibrium
303 example, we first extract the center of mass c of the hawk from a shape parameter configuration
304 s . We stress that varying s changes the interior shape, which adjusts the weight distribution of the
305 hawk and, therefore, affects the center of mass. We define the loss as the distance between c and the
306 hanging position. **Pingchuan:** Maybe consider change the word “synthetic”. They are not a proof-
307 of-concept toy experiments. My suggested titles: 4.1 Experimental Setup; 4.2 Main result; 4.2.1
308 Rigid-body experiments; 4.2.2 Soft-body experiments; 4.3 Real-world validation. [Experiments
309 settings, baselines, metrics] all belong to experimental setup section

310 **Baseline** We choose two strong SOTA PGSR Chen et al. (2024) and Gaussian Surfels Dai et al.
311 (2024) as baseline. With the output triangle surface mesh from the model, we perform mesh fixing
312 and filling to obtain a watertight tetrahedron mesh. Then We replace the topology representation
313 with the this tetrahedron mesh, and chain it into our rest pipeline.

315 **Benchmark** We use two benchmarks to test the performance of the pipeline. The first benchmark
316 is the processing time of each method, which measures the speed on each object. The second
317 benchmark is the quality of the inferred topology structure based on the 3D printability index. This
318 index is defined as the Laplacian of surface normal to measure the smoothness of boundary, which
319 is the critical factor in 3D printing. **Pingchuan: Benchmark → Metrics/Criteria**

321 **Performance** Tab. 1 exhibits the processing time of our method and each baseline. We can observe
322 that among different objects, our methods is faster than any of the baseline. To be more specific, our
323 method is 5.26 times faster than PGSR (voxel size=0.05) in average, 4.81 times faster than PGSR
(voxel size=0.2) in average, and 1.28 times faster than Gaussian Surfels in average. This result

Table 1: Processing time (s). Expected small, with the optimal result bolded

Dataset id	frog	fish-1	swim-ring	cow	toy-1	toy-2	fandisk
Ours	238	157	202	227	184	244	255
PGSR (voxel size=0.05)	1778	1143	1065	1134	973	1198	1285
PGSR (voxel size=0.2)	1059	1146	1085	969	1197	1170	1029
Gaussian Surfels (depth=8)	285	Fail	289	Fail	265	298	310
Datset id	hippo	toy-3	bird	kangaroo	stone	pear	pig
Ours	223	216	217	210	197	206	232
PGSR (voxel size=0.05)	1213	1008	1050	1131	1035	1150	1095
PGSR (voxel size=0.2)	1093	1025	992	1038	956	1052	1017
Gaussian Surfels (depth=8)	294	268	272	281	267	280	276
Dataset id	spot	teapot	tumbler	fish-2	rabbit		
Ours	234	280	290	198	212		
PGSR (voxel size=0.05)	1257	1244	1141	1045	1040		
PGSR (voxel size=0.2)	1132	1123	1173	981	967		
Gaussian Surfels (depth=8)	316	288	307	276	274		

shows that our pipeline provides an highly efficient processing method. [Pingchuan: Performance → Main results](#)

For the quality of reconstruction, Tab. 2 shows that our method beats all of the baselines in most of the cases, except for toy-1 examples. More concretely, the quality of our method is 2.33 times better than PGSR (voxel size=0.05) in average, 2.50 times better than PGSR (voxel size=0.2) in average, and 2.55 times better than Gaussian Surfels in average, while we observe that in two tasks (fish-1 and cow), Gaussian Surfel fails to output valid result.

Fig. 2 and Fig. 3 visualize the optimization result of the experiment on six typical examples (the full gallery can be found in appendix Sec. C.1). Fig. 2 shows a slice of the optimized topology structure, where we can intuitively observe that the result from our method is more smooth and printable than the result from the baselines. Fig. 3 compares the initial and final pose of the objects, where the initial guess of fully solid object leads to obvious inclination, and the final result from our pipeline successfully make the objects horizontal. Based on this result, we further 3D print one object, validating the manufacturing ability of our method.

4.2 EXTENSION TO SOFT BODIES

We also extend our synthetic dataset to soft bodies with two experiments. The first experiment is a soft pear bouncing on the ground, with a hard part inserted into it. Different topology structures of this hard part lead to different bouncing behaviors, while our goal is to optimize this structure to reproduce the target video. The loss is defined as the distance of the motion of several pivot points on the pear to the target. The second experiment is a soft bird standing on its legs, with hard parts (bones) inside. Our goal is to optimize the topology structure of the bones in order to make the bird stable with least volume of bones. The loss is set as the combination of penalty from static pose and the volume of hard parts.

Fig. 4 visualize the result of these 2 experiments. In the pear experiment, the fully soft initial guess output a motion deviating large from the target, while our optimized result can closely reproducing the target motion. In the bird experiment, the fully soft initial guess quickly make the bird unstable, while our result with only 10.6% volume of bone can maintain the bird stable. We also show that with the same volume but random distribution of the bone, the bird still falls down easily, meaning that our delicately designed topology structure is necessary.

Table 2: Reconstruction quality. Expected small, with the optimal result bolded

Dataset id	frog	fish-1	swim-ring	cow	toy-1	toy-2	fandisk
Ours	0.306	0.358	0.242	0.300	0.294	0.274	0.335
PGSR (voxel size=0.05)	0.745	0.998	0.464	0.410	0.263	0.761	0.904
PGSR (voxel size=0.2)	0.781	0.907	0.894	0.801	0.595	0.845	0.878
Gaussian Surfels	0.851	Fail	0.882	Fail	0.552	0.939	0.865
Datset id	hippo	toy-3	bird	kangaroo	stone	pear	pig
Ours	0.663	0.285	0.268	0.388	0.290	0.245	0.343
PGSR (voxel size=0.05)	0.910	0.478	0.846	0.841	0.907	1.078	0.891
PGSR (voxel size=0.2)	0.869	0.720	0.850	0.480	0.904	1.027	0.867
Gaussian Surfels	0.939	0.459	0.906	0.976	0.890	1.148	0.939
Dataset id	spot	teapot	tumbler	fish-2	rabbit		
Ours	0.347	0.264	0.305	0.352	0.298		
PGSR (voxel size=0.05)	0.822	0.822	0.953	0.605	0.930		
PGSR (voxel size=0.2)	0.901	0.841	0.755	0.889	0.903		
Gaussian Surfels	0.724	0.850	0.963	0.472	0.940		

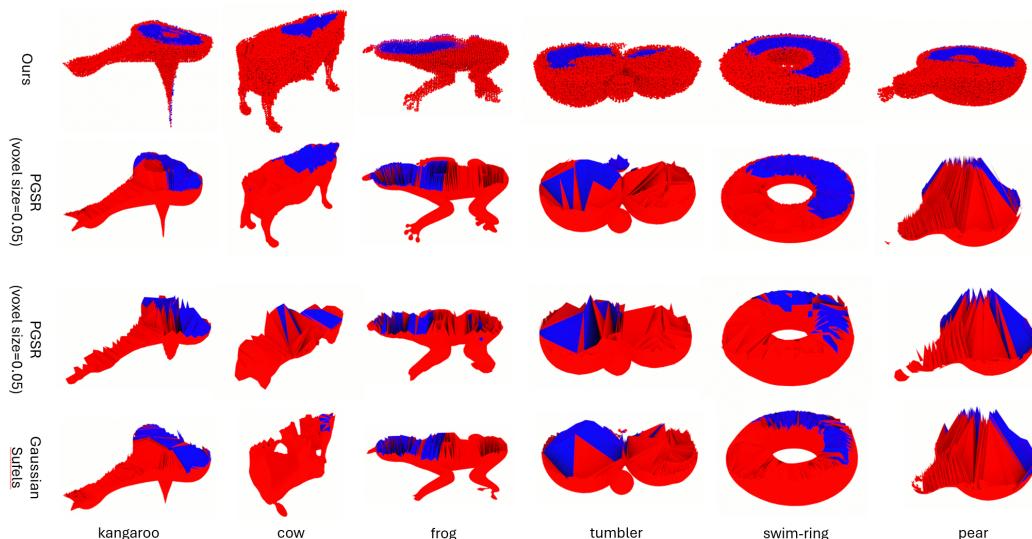
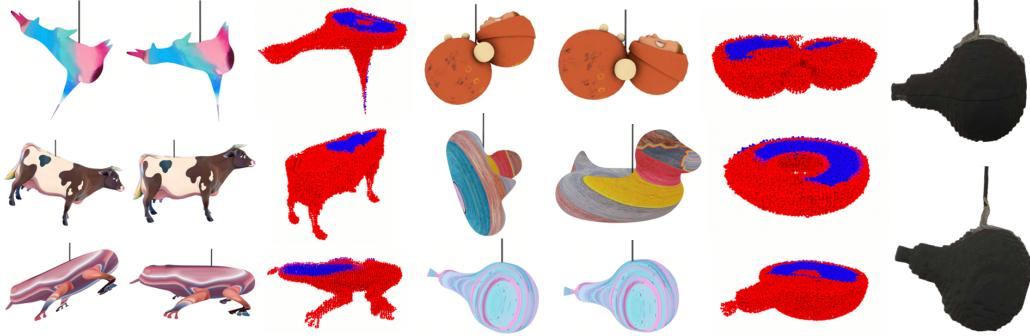


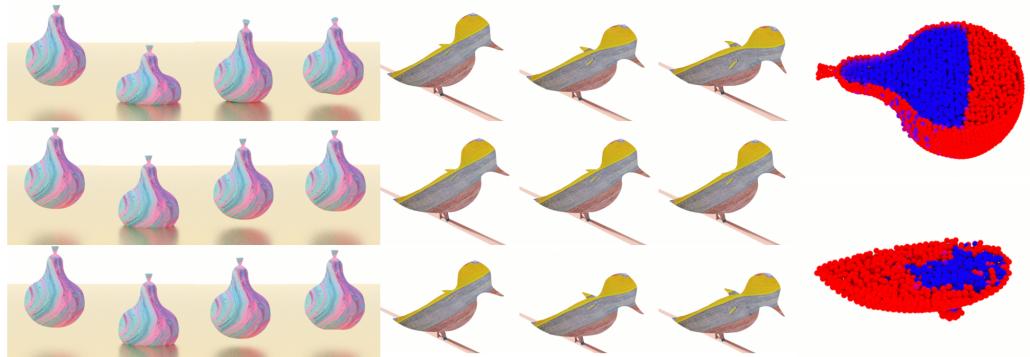
Figure 2: Comparison between our method and mesh baselines (Sec. 4.1). Red part represents solid part and blue part represents hollow part. Top row: our method; bottom three rows: mesh baselines.

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447



448
449 Figure 3: *Six rigid experiments* (Sec. 4.1). Left part: simulated result of each experiment. For each
450 experiment, left: initial balancing position of the object with fully solid topology structure; middle:
451 final balancing position of the object with the result from our pipeline; right: optimized topology
452 structure with red representing solid part and blue representing hollow part. Right: photos of the 3D
453 printed result. Top: Initial guess with inclining pose; bottom: final result with horizontal pose.
454
455
456
457
458
459
460
461
462

463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485



476 Figure 4: *Two soft body extensions* (Sec. 4.2). Left part: experiment on bouncing pear. Top row:
477 frames from simulated motion of a fully soft initial guess; middle row: simulated motion of our
478 optimized result; bottom row: target motion. Middle part: experiment on standing bird. Top row:
479 frames from simulated motion of a fully soft initial guess; middle row: simulated motion of our
480 optimized result; bottom row: simulated motion of a bird with the same bone volume as the result
481 but random distribution. Right part: slice of the optimized topology structure with red representing
482 soft part and blue representing hard part.
483
484
485



Figure 5: *Wobble doll* (Sec. 4.3). Left: a photo of the wobble doll. Middle: frames from a video recording of the harmonic oscillation of the doll; Middle: simulated motion with an interior design optimized from our pipeline; Bottom: simulated motion of a solid doll. Right: a visualization of the optimized interior geometry from our pipeline.

4.3 REAL-WORLD VALIDATIONS

Finally, We validate our pipeline on four real-world experiments. These experiments includes rigid and soft body with static and dynamic settings. One rigid and one soft body examples are shows in here, and the rest can be found in appendix Sec. C.2

Rigid The rigid body task involves inferring the interior structure of a wobble doll, a classic rigid-body toy with dynamical and oscillating motions (Fig. 5 left and top middle). The interior of a wobble doll is typically non-empty. The goal is to infer one possible interior structure of a wobble doll from a video recording of its harmonic oscillations (Fig. 5 top middle).

We define an objective that penalizes the discrepancy of the oscillation period between the reference video and the simulated motion. For the reference video, we use OpenCV to identify and extract the beginning and ending frames of one period of oscillation as well as identifying the maximal tilting angle θ from the middle frame in one period. For the simulated doll, we compute its center of mass c from the shape parameter s as described before and estimate R , the radius of curvature of the bottom surface by fitting a spherical surface to the bottom particles in the point cloud. We then compute the doll’s vibration frequency from these parameters.

Fig. 5 right shows our optimized interior structure of the doll, where most of the mass is clustered to one side of the bottom. The simulated motion of this optimized design (Fig. 5 middle), closely matches the maximal tilting angle and the period the recorded video (4.0Hz. See Fig. 5 top middle). In contrast, an initial guess of solid doll leads to a substantially different oscillation period (2.4Hz. See Fig. 5).

Soft The soft body task is to infer the chamber design in an opaque, pneumatically actuated soft finger, a widely used model in soft robotics (Fig. 6 left). This task is substantially more challenging than previous ones as it involves deformable bodies and air pressure in a pneumatic chamber. Given a video recording of the bending motion of the soft finger with increasing air pressure, our goal is to deduce the inflatable chamber design to reproduce the final bending angle (Fig. 6 middle left).

5 CONCLUSION AND LIMITATIONS

We presented a holistic pipeline for inferring interior geometries of opaque objects from visual inputs. Our pipeline extended Gaussian splatting with a particle-based interior shape representation, a novel differentiable point-cloud simulator, and flexible choices of topology optimization. Our whole pipeline was built upon a gracefully unified particle-based representation free from error-prone mesh



540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
Figure 6: *Soft finger* (Sec. 4.3). Left: a photo of the pneumatically actuated soft finger. Middle left: the bending finger with maximal air pressure. Middle: bending finger simulated with an optimized design. Middle right: the finger simulated with an initial guess of the chamber location; Right: visualization of the optimized chamber location.

extraction and processing in modeling, simulation, or optimization. Our pipeline enabled promising applications in reverse engineering tasks in 3D vision and soft robotics.

Our work has several limitations. First, our pipeline currently focuses on a single object and does not handle multiple objects. Second, our pipeline assumes the topology structure only contains two types of material. Finally, our current optimization method does not exploit the appearance information in Gaussian splatting.

REFERENCES

- 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.
- Markus Becker, Markus Ihmsen, and Matthias Teschner. Corotated sph for deformable solids. *NPH*, 9:27–34, 2009.
- Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgssr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024.
- Zilong Chen, Feng Wang, and Huaping Liu. Text-to-3d using gaussian splatting. *arXiv preprint arXiv:2309.16585*, 2023.
- Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.
- Joshua D Deaton and Ramana V Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49:1–38, 2014.
- Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12882–12891, 2022.
- Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Diffpd: Differentiable projective dynamics. *ACM Transactions on Graphics (TOG)*, 41(2):1–21, 2021.
- Maria B Dühring, Jakob S Jensen, and Ole Sigmund. Acoustic design by topology optimization. *Journal of sound and vibration*, 317(3-5):557–575, 2008.
- Hans A Eschenauer and Niels Olhoff. Topology optimization of continuum structures: A review*. *Applied Mechanics Reviews*, 54(4):331–390, 07 2001. ISSN 0003-6900. doi: 10.1115/1.1388075.

- 594 Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian
 595 Coros. Add: Analytically differentiable dynamics for multi-body systems with frictional contact.
 596 *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.
- 597
- 598 Arvi Gjoka, Zizhou Huang, Davi Colli Tozoni, Zachary Ferguson, Teseo Schneider, Daniele
 599 Panozzo, and Denis Zorin. Differentiable solver for time-dependent deformation problems with
 600 contact. *arXiv preprint arXiv:2205.13643*, 2022.
- 601 Minghao Guo, Bohan Wang, Pingchuan Ma, Tianyuan Zhang, Crystal Elaine Owens, Chuang Gan,
 602 Joshua B. Tenenbaum, Kaiming He, and Wojciech Matusik. Physically compatible 3d object
 603 modeling from a single image. *Advances in Neural Information Processing Systems*, 37, 2024.
- 604
- 605 Christian Hafner, Mickaël Ly, and Chris Wojtan. Spin-it faster: Quadrics solve all topology optimi-
 606 zation problems that depend only on mass moments. *ACM Transactions on Graphics (TOG)*,
 607 43(4):1–13, 2024.
- 608 David Hahn, Pol Banzet, James M Bern, and Stelian Coros. Real2sim: Visco-elastic parameter
 609 estimation from dynamic motion. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.
- 610
- 611 Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B Tenenbaum, William T Freeman, Jiajun
 612 Wu, Daniela Rus, and Wojciech Matusik. Chainqueen: A real-time differentiable physical simu-
 613 lator for soft robotics. In *2019 International conference on robotics and automation (ICRA)*, pp.
 614 6265–6271. IEEE, 2019.
- 615 Jakob Søndergaard Jensen and Ole Sigmund. Topology optimization for nano-photonics. *Laser &*
 616 *Photonics Reviews*, 5(2):308–321, 2011.
- 617
- 618 Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splat-
 619 ting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.
- 620
- 621 Theodore Kim and David Eberle. Dynamic deformables: implementation and production practical-
 622 ities. In *ACM SIGGRAPH 2020 Courses*, pp. 1–182. 2020.
- 623 Cornelius Lanczos. *The variational principles of mechanics*. Courier Corporation, 2012.
- 624
- 625 Jeongseok Lee, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha
 626 S. Srinivasa, Mike Stilman, and C Karen Liu. Dart: Dynamic animation and robotics toolkit. *The*
 627 *Journal of Open Source Software*, 3(22):500, 2018.
- 628
- 629 Shuguang Li, John J Stampfli, Helen J Xu, Elian Malkin, Evelin Villegas Diaz, Daniela Rus, and
 630 Robert J Wood. A vacuum-driven origami “magic-ball” soft gripper. In *2019 International Con-
 631 ference on Robotics and Automation (ICRA)*, pp. 7401–7408. IEEE, 2019.
- 632
- 633 Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu
 634 Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for
 635 geometry-agnostic system identification. In *The Eleventh International Conference on Learn-
 ing Representations*, 2022.
- 636
- 637 Yifei Li, Yuchen Sun, Pingchuan Ma, Eftychios Sifakis, Tao Du, Bo Zhu, and Wojciech Matusik.
 638 Neural fluidic system design and control with differentiable simulation. *Advances in Neural*
Information Processing Systems, 37, 2024.
- 639
- 640 Yue Li, Xuan Li, Minchen Li, Yixin Zhu, Bo Zhu, and Chenfanfu Jiang. Lagrangian-eulerian
 641 multi-density topology optimization with the material point method. *International Journal for*
Numerical Methods in Engineering, 122, 03 2021. doi: 10.1002/nme.6668.
- 642
- 643 Xingyu Lin, Zhiao Huang, Yunzhu Li, Joshua B Tenenbaum, David Held, and Chuang Gan. Diff-
 644 skill: Skill abstraction from differentiable physics for deformable object manipulations with tools.
 645 In *International Conference on Learning Representations*, 2021.
- 646
- 647 C Karen Liu and Sumit Jain. A quick tutorial on multibody dynamics. *Online tutorial, June*, pp. 7,
 2012.

- 648 Haixiang Liu, Yuanming Hu, Bo Zhu, Wojciech Matusik, and Eftychios Sifakis. Narrow-band
 649 topology optimization on a sparsely populated grid. *ACM Transactions on Graphics (TOG)*, 37
 650 (6):1–14, 2018.
- 651
- 652 Min Liu, Gang Yang, Siyuan Luo, Chen Yu, and Lin Shao. Softmac: Differentiable soft body
 653 simulation with forecast-based contact model and two-way coupling with articulated rigid bodies
 654 and clothes. *arXiv preprint arXiv:2312.03297*, 2023.
- 655 Yiyue Luo, Kui Wu, Andrew Spielberg, Michael Foshey, Daniela Rus, Tomás Palacios, and Wo-
 656 jciech Matusik. Digital fabrication of pneumatic actuators with integrated sensing by machine
 657 knitting. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*,
 658 pp. 1–13, 2022.
- 659
- 660 Sebastian Martin, Bernhard Thomaszewski, Eitan Grinspun, and Markus Gross. Example-based
 661 elastic materials. In *ACM SIGGRAPH 2011 papers*, pp. 1–8. 2011.
- 662
- 663 Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovit-
 664 skiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo
 665 collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-
 nition (CVPR)*, pp. 7210–7219, June 2021.
- 666
- 667 Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. Gaussian splatting slam.
 668 *arXiv preprint arXiv:2312.06741*, 2023.
- 669
- 670 Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint
 671 method. *ACM Transactions On Graphics (TOG)*, 23(3):449–456, 2004.
- 672
- 673 Shuhei Miyashita, Steven Guitron, Kazuhiro Yoshida, Shuguang Li, Dana D Damian, and Daniela
 674 Rus. Ingestible, controllable, and degradable origami robot for patching stomach wounds. In *2016
 675 IEEE international conference on robotics and automation (ICRA)*, pp. 909–916. IEEE, 2016.
- 676
- 677 Joseph J Monaghan and John C Lattanzio. A refined particle method for astrophysical problems.
 678 *Astronomy and Astrophysics (ISSN 0004-6361)*, vol. 149, no. 1, Aug. 1985, p. 135-143., 149:
 679 135–143, 1985.
- 680
- 681 Matthias Müller, Bruno Heidelberger, Matthias Teschner, and Markus Gross. Meshless deformations
 682 based on shape matching. *ACM transactions on graphics (TOG)*, 24(3):471–478, 2005.
- 683
- 684 Matthias Müller, Miles Macklin, Nuttapong Chentanez, and Stefan Jeschke. Physically based shape
 685 matching. In *Computer Graphics Forum*, volume 41, pp. 1–7. Wiley Online Library, 2022.
- 686
- 687 Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove.
 688 DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings
 689 of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.
- 690
- 691 Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural
 692 radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer
 693 Vision and Pattern Recognition (CVPR)*, pp. 10318–10327, June 2021.
- 694
- 695 Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Differentiable simulation of soft multi-
 696 body systems. *Advances in Neural Information Processing Systems*, 34:17123–17135, 2021.
- 697
- 698 Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner’s guide
 699 to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH
 700 ’12, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450316781.
 doi: 10.1145/2343483.2343501.
- 701
- Ole Sigmund. Sigmund, o.: A 99 line topology optimization code written in matlab. structural and
 multidisciplinary optimization 21, 120-127. *Structural and Multidisciplinary Optimization*, 21:
 120–127, 04 2001. doi: 10.1007/s001580050176.
- Fun Shing Sin, Daniel Schroeder, and Jernej Barbič. Vega: non-linear fem deformable object simu-
 lator. In *Computer Graphics Forum*, volume 32, pp. 36–48. Wiley Online Library, 2013.

- 702 Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative
 703 gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.
 704
- 705 Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology
 706 optimization. *Computer methods in applied mechanics and engineering*, 192(1-2):227–246, 2003.
 707
- 708 Ying Wang, Nicholas J Weidner, Margaret A Baxter, Yura Hwang, Danny M Kaufman, and Shinjiro
 709 Sueda. Redmax: Efficient & flexible approach for articulated dynamics. *ACM Transactions on
 710 Graphics (TOG)*, 38(4):1–10, 2019.
- 711 Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural
 712 radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.
 713
- 714 Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C Karen Liu. Fast and
 715 feature-complete differentiable physics engine for articulated rigid bodies with contact con-
 716 straints. In *Robotics: Science and Systems*, 2021.
- 717 Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian,
 718 and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. *arXiv preprint
 719 arXiv:2310.08528*, 2023.
- 720 Jun Wu, Ole Sigmund, and Jeroen P Groen. Topology optimization of multi-scale structures: a
 721 review. *Structural and Multidisciplinary Optimization*, 63:1455–1480, 2021.
 722
- 723 Tianyi Xie, Zeshun Zong, Yuxin Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang.
 724 Physgaussian: Physics-integrated 3d gaussians for generative dynamics. *arXiv preprint
 725 arXiv:2311.12198*, 2023.
 726
- 727 Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit
 728 Agrawal. An end-to-end differentiable framework for contact-aware robot design. In *Robotics:
 729 Science & Systems*, 2021.
- 730 RJ Yang and CJ Chen. Stress-based topology optimization. *Structural optimization*, 12:98–105,
 731 1996.
 732
- 733 Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and
 734 Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point
 735 cloud priors. *arXiv preprint arXiv:2310.08529*, 2023.
 736
- 737 Xinyuan Yu, Siheng Zhao, Siyuan Luo, Gang Yang, and Lin Shao. Diffclothai: Differentiable
 738 cloth simulation with intersection-free frictional contact and differentiable two-way coupling with
 739 articulated rigid bodies. In *2023 IEEE/RSJ International Conference on Intelligent Robots and
 740 Systems (IROS)*, pp. 400–407. IEEE, 2023.
- 741 Ji-Hong Zhu, Wei-Hong Zhang, and Liang Xia. Topology optimization in aircraft and aerospace
 742 structures design. *Archives of computational methods in engineering*, 23:595–622, 2016.
 743
- 744

745 A DETAILS OF ALGORITHMS

746 A.1 SOFT BODY DYNAMICS

747 **Discretization** We start from the discretization and interpolation of functions on the object space.
 748 For any function $f(\mathbf{x})$ on the object, we records its value on each particle f_i . Then for any position
 749 \mathbf{x} , the function value can be interpolated as

$$750 \langle f(\mathbf{x}) \rangle = \sum_i f(\mathbf{x}_i) W(\mathbf{x} - \mathbf{x}_i, h) \frac{m_i}{\rho_i} \quad (9)$$

751 where $W(r, h)$ is the kernel function used to smooth the interpolation. h is the interpolation range
 752 parameter and usually set to be 1 to 2 times average distance between particles. There are many
 753

forms of the kernel function, while we use the version based on spline functions Monaghan & Lattanzio (1985), with the following form

$$W(\mathbf{r}, h) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^3 & \text{if } 0 \leq q < 1 \\ \frac{1}{4}(2-q)^3 & \text{if } 1 \leq q < 2 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $q = |\mathbf{r}|/h$. m_i is the mass of particle i , and ρ_i is the density near particle i , which can be computed from

$$\rho_i = \sum_j m_j W(\mathbf{x}_{ij}^0, h) \quad (11)$$

where \mathbf{x}_{ij}^0 is the undeformed displacement from particle j to i .

Geometry When the soft body deforms, we can write the deformation map between the deformed position \mathbf{x} and undeformed position \mathbf{x}^0 in $\mathbf{x}^0 \mapsto \mathbf{x} = \mathbf{x}^0 + \mathbf{u}$, while the Jacobian of the deformation map can be written in $\mathbf{J} = \mathbf{I} + \nabla \mathbf{u}^T$. Note that $\nabla \mathbf{u}^T$ can be decomposed into a scaling part and a rotation part, while only the scaling part is effective for mechanical computations. Thus, we need to extract the rotation matrix \mathbf{R}_i for each particle based on its initial neighborhood. Define

$$\mathbf{A}_{pqi} = \sum_i m_i W(\mathbf{x}_{ij}^0, h) ((\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j^0 - \mathbf{x}_i^0)^T) \quad (12)$$

Then \mathbf{R}_i can be computed from the polar decomposition of \mathbf{A}_{pqi} , and

$$\nabla \mathbf{u}_i = \sum_j \frac{m_j}{\rho_j} \mathbf{u}_{ji} \nabla W(\mathbf{x}_{ji}^0, h)^T \quad (13)$$

where

$$\mathbf{u}_{ji} = \mathbf{R}_i^{-1}(\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad (14)$$

Mechanics Then we can use either non-linear Green-Saint-Venant strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2} (\mathbf{J}^T \mathbf{J} - \mathbf{I}) \quad (15)$$

or the linear Cauchy-Green strain tensor

$$\boldsymbol{\epsilon} = \frac{1}{2} (\mathbf{J} + \mathbf{J}^T) - \mathbf{I} \quad (16)$$

to calculate the stress energy of the object. Based on Saint Venant–Kirchhoff model, the stress energy can be written as

$$E = \sum_i \frac{m_i}{\rho_i} \left(\mu \boldsymbol{\epsilon} : \boldsymbol{\epsilon} + \frac{\lambda}{2} (\text{tr}(\boldsymbol{\epsilon}))^2 \right) \quad (17)$$

where μ and λ are the Lame parameters of the material. Then the stress force on each particle can be computed from the gradients of stress energy with respect to the position of each particle.

A.2 TIME INTEGRATION

We will briefly explain the math detail of the implicit time integration based on incremental potential in this section. Implicit time integration requires us to solve the following two equations

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}_{k+1}) \Delta t \quad (18)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1} \Delta t \quad (19)$$

where \mathbf{x}_k and \mathbf{x}_{k+1} is the position at time step k and $k+1$, \mathbf{v}_k and \mathbf{v}_{k+1} is the velocity at time step k and $k+1$, \mathbf{f} is the total force, Δt is the size of time step, and \mathbf{M} is the mass matrix. Notice that \mathbf{f} is usually in a complex form, meaning that there is usually no analytical solution to this equation. To numerically solve it, we write $\mathbf{f}(\mathbf{x}) = -\nabla E(\mathbf{x}) + \mathbf{f}^{ext}$, splitting it into internal and external part. Then we have

$$\nabla E(\mathbf{x}_{k+1}) + \frac{1}{\Delta t^2} \mathbf{M} (\mathbf{x}_{k+1} - (\mathbf{x}_k + \Delta t \mathbf{v}_k + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}^{ext})) = 0 \quad (20)$$

810 Table 3: Simulation parameters, N/A means that the experiment does not involve this parameter
811

Experiment	Time step (s)	Density (kg/m ³)	Young's modulus (Pa)	Poisson's ratio
Rigid	5e-3	1e3	N/A	N/A
Soft	5e-4	1e3	Soft: 1.5e5, hard: 3e7	0.4
Rigid (real)	5e-3	5e3	N/A	N/A
Soft (real)	5e-4	5e3	1e7	0.49

818
819 If we rewrite this into

820
821
$$\nabla \left(E(\mathbf{x}_{k+1}) + \frac{1}{2\Delta t^2} \text{tr}((\mathbf{x}_{k+1} - \mathbf{y}_k)^T \mathbf{M}(\mathbf{x}_{k+1} - \mathbf{y}_k)) \right) = 0 \quad (21)$$

822
823

824 where $\mathbf{y}_k = \mathbf{x}_k + \Delta t \mathbf{v}_k + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}^{ext}$. Notice that this can be converted into an optimization
825 problem, and then we can use Newton optimizer to solve \mathbf{x}_{k+1} .
826827

B DETAILS OF EXPERIMENTS

828829

B.1 IMPLEMENTATION

830831 We implemented the core steps in our pipeline in C++ and Python. We parallelized the differentiable
832 simulator with both CUDA on GPUs and OpenMP on CPUs, combining with Nvidia Warp. We used
833 Open3D for point cloud processing and OpenCV for reference video processing, and PyTorch for the
834 implementation of neural implicit surface. We evaluated our pipeline on the tasks below on a server
835 with an AMD EPYC 9754 128-Core CPU, 12× DDR5 4800 16GB (384GB in total) RAM, and 1 ×
836 NVIDIA RTX 4090 24G GPU. We used Open3D to perform point cloud processing, OpenCV for
837 video processing, and L-BFGS-B and Adam for optimization.
838839

B.2 PARAMETERS

840841 **Neural Implicit Surface** We implement this neural network with eight fully connected layers,
842 each of which is applied with dropouts. All internal layers have size 512×512 , and are connected
843 with ReLu non-linearities. The network is trained with weight-normalization upon the gradients
844 from the differentiable simulator. We train for 10000 epochs with learning rate 3e-6.
845846 **Simulation** The basic simulation parameters are listed in Tab. 3, while the actuation pressure of
847 the soft hand is set to be 5e5 Pa.
848849

C ADDITIONAL RESULTS OF EXPERIMENTS

850851

C.1 FULL GALLERY OF SYNTHETIC EXPERIMENTS

852853

C.2 ADDITIONAL REAL-WORLD EXAMPLES

854855

C.3 ABLATION STUDIES

856

857

858

859

860

861

862

863