Rigid Body Adversarial Attacks

Aravind Ramakrishnan
University of Toronto
aravind@cs.toronto.edu

David I.W. Levin
University of Toronto
NVIDIA
diwlevin@cs.toronto.edu

Alec Jacobson
University of Toronto
Adobe Research
jacobson@cs.toronto.edu

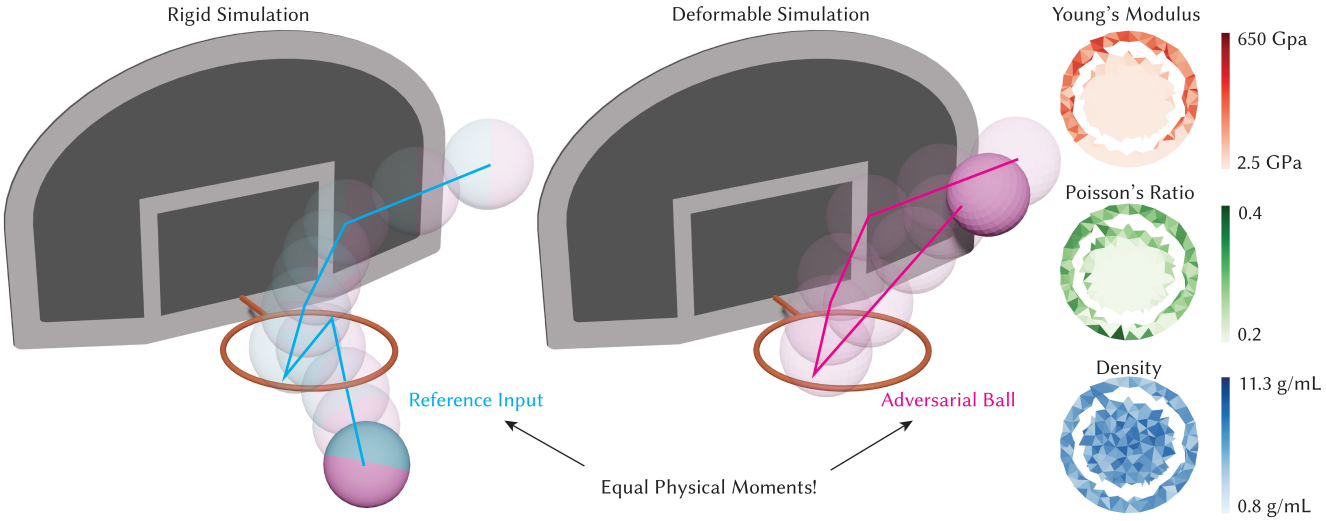

Figure 1. We construct an adversarial ball (right) out of perceptually stiff materials (with a minimum Young's modulus of 2.5 GPa), such that it results in a maximally different deformable simulation trajectory (middle) compared to a reference ball while having identical physical moments and thus trajectories in a rigid body simulation (left).

Abstract

Due to their performance and simplicity, rigid body simulators are often used in applications where the objects of interest can considered very stiff. However, no material has infinite stiffness, which means there are potentially cases where the non-zero compliance of the seemingly rigid object can cause a significant difference between its trajectories when simulated in a rigid body or deformable simulator.

Similarly to how adversarial attacks are developed against image classifiers, we propose an adversarial attack against rigid body simulators. In this adversarial attack, we solve an optimization problem to construct perceptually rigid adversarial objects that have the same collision geometry and moments of mass to a reference object, so that they behave identically in rigid body simulations but maximally different in more accurate deformable simulations. We demonstrate the validity of our method by comparing simulations of several examples in commercially available simulators.

1. Introduction

When simulating stiff objects, rigid body simulators are extremely popular and are often used in lieu of more accurate deformable simulators due to their performance and simplicity. Of course, even very stiff materials are deformable at some scale. These deformations change how objects respond to contacts and how internal stresses propagate through an object. Moreover, spatial variations of material properties in an object are not captured by the object-level parameters of a rigid body simulator: mass, center of mass, and moment of inertia.

As a result, it is possible that simulating certain objects with rigid body methods can lead to less accurate results. This can be dangerous in contexts where safety of the system is determined by the robustness of the underlying model. For example, in machine learning, there is increasing interest in learning physical models (e.g. [16, 60, 73, 82]), including rigid body dynamics (e.g. [13, 15, 79]), for use in various downstream applications where inaccuracies can cause problems. In robotics, Sim2Real training (see

[37, 81, 109]) based on vulnerable physics simulation could be dangerous in safety critical tasks.

In deep learning, such vulnerabilities are studied in the form of *adversarial attacks*, where seemingly normal malicious inputs to the network are generated (typically as imperceptible perturbations to a reference input) in order to cause the model to make mistakes. For instance, images with some underlying change such as small modifications to pixel values or lighting can be used to trick image classifiers into misclassifying them, which can have grave consequences if the classifier was used in a safety critical application [17, 103]. These attacks exist in physical real-world applications - for example, using adversarial background music to disrupt the functionality of the voice assistant tool Amazon Alexa [55], makeup to cause face recognition software to fail [106], even placing small markings on the road to trick Tesla vehicles' autopilot lane detection software [92]. In machine learning, researching these adversarial attacks was the first step towards improving safety, eventually leading to adversarial training which turned attacks into strategies to make models more robust.

In this paper, we propose using optimization techniques to construct adversarial objects using physically reasonable materials, which will behave identically to a reference object in rigid body simulation, but maximally different in more physically accurate deformable simulation. As rigid body simulators use only the collision geometry and the mass moments of the object, the adversarial objects require identical external geometry and first three moments of mass to the reference object so that they are indistinguishable in the rigid body setting. To achieve this objective, we first define a cost function that encodes the difference in the simulation result of the reference and adversarial object. We choose the degrees of freedom of the optimization to be the object's material distribution and internal geometry. The adjoint method allows efficient computation of gradients, enabling the use of descent methods to determine the degrees of freedom subject to the constraints.

We demonstrate the efficacy of our method by constructing several adversarial objects and comparing the results of their simulations with their reference in POLYFEM, a commercially available simulator. Thus, we show that robotics planning and control tools make a potentially dangerous model assumption when using rigid body simulators.

2. Related Works

To construct adversarial objects, our work draws on ideas from inverse problems, machine learning, and simulation.

2.1. Inverse Problems

Consider a system whose behavior is governed by some set of parameters. In a *forward problem*, the goal is to compute the evolution of that system given its parameters. In the corresponding *inverse problem*, the goal is to determine the parameters of the system from observations of the evolution of the system - either from physical measurements or via simulation (see e.g. [91, 95]). In machine learning and vision, common inverse problems include inverse kinematics (e.g. [56, 105]) and imaging (e.g. [47, 87]), though there have been works studying inverse problems in physics using invertible neural networks [3] and approximate inverse simulation with a learned correction function [38].

In the context of this work, the system we consider is the elastodynamics simulation of an object, and our inverse problem to determine its material parameters and internal geometry. In mechanics, Xu et al. [102] use physics informed neural networks to solve inverse problems with linear elastic and hyperelastic materials where they determine loads and internal pressures. In computer animation, there has been some work in solving inverse problems to optimize the trajectories of objects. Approaches include determining control forces via sequential quadratic programming [100], determining physical parameters (i.e. initial positions and velocities, surface normal variations, etc.) that minimize an energy subject to position constraints of an object at chosen times [74], and simulating backwards using time-reversed simulators to determine initial conditions [94]. Unlike these prior works, we are optimizing for material properties rather than loads, control forces, or initial conditions.

Optimizing the internal geometry of our adversarial object leads us to the topology optimization inverse problem, where one finds the optimal allocation of a fixed amount of material over a design domain such that the compliance of the design is minimized when subjected to a load. There are a few broad approaches to topology optimization including nodal/element based design variables [12, 101], level sets [1, 97], and implicit neural representations [108]. In our work, we follow the solid isotropic material with penalization parametrization (SIMP) approach of defining a per element occupancy and using a penalizing power-law to scale stiffness (see [11]).

Unlike the standard topology optimization problem, our constraint is matching moments of mass rather than having a fixed quantity of material. Additionally, our objective is not minimum compliance, we have a continuous spread of allowed materials rather than one or a few, and furthermore, we are interested in optimizing over dynamic loads. The mass moments constraint is a similar problem to Matejka and Fitzmaurice [65], where the authors construct datasets of 2D points such that they have the same mean (first moment), standard deviation (second moment about the mean), and Pearson's R value to those of some reference dataset. They use a simulated annealing optimization scheme to the new dataset along a target shape. We instead leave our material density distribution as an untargeted optimization, taking any values such that the moments match (see Fig. 2).

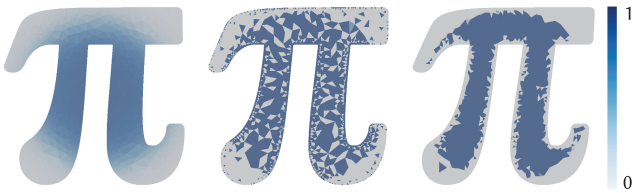

Figure 2. For a given geometry, there can be many mass density distributions that have identical moments of mass. Above are three visually different density distributions on a mesh that have the same total mass, center of mass, and moment of inertia.

2.2. Adversarial Attacks

Our method is inspired by adversarial attacks in machine learning, where an adversarial input is constructed by causing small perturbations to a reference. There has been considerable interest for adversarial attacks in various contexts and applications, including images [33, 52, 63, 69, 90], videos [46, 49, 58, 59, 99], audio [19, 39, 64, 107], geometric representations of 3D shapes [40, 54, 88], 3D printed objects that fool classifiers [4, 41], machine learning components of industrial control systems [2, 25, 29], and even control signals in robots [71].

Most relevant to our work is the intersection of physical inverse design problems and adversarial attacks. To our knowledge, the only study combining the two fields is Azakami et al. [5], who perform an attack over the shape of legged robots that are simulated in MuJoCo. These robots have body components parameterized by their lengths and thicknesses. The authors use a differential evolution optimization scheme to find the parameters that are close to those of the original robot but induce a failure in the control, causing them to fall over. However, the parameter changes may cause the new design to have different centers of mass and broken symmetries.

In machine learning, adversarial attacks have led to adversarial training, where model robustness is improved by training on adversarial examples [6, 76]. We think our adversarial rigid objects can similarly be used for improved validation and training of learned physics simulation and robotics planning and control methods.

2.3. Simulation

Both rigid body (RBD) and deformable simulation techniques are popular in robotics [62].

2.3.1 Rigid Body Simulation

Dynamic simulation of highly stiff objects is a common task across several fields. A rigid body is an idealized model, where the simulated object is treated as infinitely stiff - external stimulus like forces or impulses are propagated almost instantaneously across the object eliminating relative deformations [53]. As a result, rigid body simulators require only per object rotational and translational degrees of freedom, rather than requiring degrees of freedom for each material point in the object to capture deformations. This multiple orders of magnitude reduction in degrees of freedom per object makes the rigid body model extremely performant and useful in a variety of interactive applications. Its speed also means that rigid body simulators are often used for simulating highly stiff objects in robotics and are used in robotics control [32, 48, 75].

Rigid body simulation has been used for modelling multi object scenes with collision since the late 1980's [7, 35, 68], and a comprehensive survey on the topic can be found in Bender et al. [9]. The most involved process in rigid body simulation is in the handling of collision and contact. Two main classes of rigid body simulators have emerged, some handling contact using penalties or barriers (e.g. [23, 28]), and others using constraints encoding non-penetration formulated as linear complementarity or velocity-level approximate linear complementarity problems (e.g. [26, 93]).

2.3.2 Deformable Simulation

The simulation of deformable bodies has been extensively studied both in robotics and in the broader academic community. We choose to use the finite element method (FEM) for our deformable simulation, which is used to solve boundary value problems over potentially intricate domains by using a discretization into elements (in our case tetrahedra). For a detailed introduction, see [50, 80, 85]. In FEM, one can choose between different constitutive models for materials depending on the application - we choose the stable Neohookean energy [50, 86]. While FEM can be very accurate and is constantly being improved, it may still be too slow to be used in certain applications.

2.3.3 Differentiable Simulation

In order to solve the inverse problem using a descent method, we need to be able to calculate gradients through the simulation. Differentiable simulation is a powerful tool for physics based learning and control problems, and has picked up in prevalence both in the context of rigid bodies [21, 36, 89, 104] and deformable objects [24, 31, 42, 45, 61, 77]. We follow the approach taken by Jatavallabhula et al. [45], where we implement our physics simulator with an autodifferentiation framework to which we provide manual derivatives at each timestep.

3. Adversarial Objects

Given a solid reference object $\Omega \subset \mathbb{R}^3$ and its moments of mass $m_0 \in \mathbb{R}_{>0}$, $m_1 \in \mathbb{R}^3$, and $m_2 \in \mathbb{R}^{3x3}$, we optimize

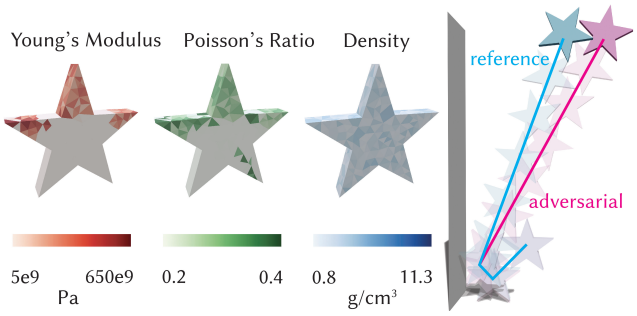

Figure 3. We simulate a star colliding off of the ground and a wall in a rigid body simulator (blue). From it, we construct an adversarial object (left) that has identical surface geometry and first three moments of mass, and simulate it in a deformable simulator (pink). Note that the angular difference between the post-contact trajectories means separation will grow over time and the effect can cascade through subsequent contacts.

a material distribution (consisting of Young's modulus Y, Poisson's ratio ν, mass density ρ, and material occupancy α) to maximally change the trajectories after simulation:

$$\underset{Y,\,\nu,\,\rho,\,\alpha}{\text{argmax}} \quad \int_\Omega ||q_{\text{adv}}(t_{\text{end}}) - q_{\text{ref}}(t_{\text{end}})||^2 dV \tag{1}$$

$$\text{s.t.} \quad Y(x) \in [Y_{\min}, Y_{\max}] \quad \forall x \in \Omega \tag{1a}$$

$$\nu(x) \in [\nu_{\min}, \nu_{\max}] \quad \forall x \in \Omega \tag{1b}$$

$$\rho(x) \in [\rho_{\min}, \rho_{\max}] \quad \forall x \in \Omega \tag{1c}$$

$$\alpha(x) \in \{0, 1\} \quad \forall x \in \Omega^{\text{o}} \tag{1d}$$

$$\alpha(x) = 1 \quad \forall x \in \partial\Omega \tag{1e}$$

$$\int_\Omega \rho(x)\alpha(x) dV = m_0 \tag{1f}$$

$$\int_\Omega x\rho(x)\alpha(x) dV = m_1 \tag{1g}$$

$$\int_\Omega ((x \cdot x)I - xx^T)\rho(x)\alpha(x) dV = m_2, \tag{1h}$$

where q_{ref} is the simulation result of the reference object, q_{adv} is the simulation result of the adversarial object, $q(t_{\text{end}})$ represents the final state of the simulation. We delegate detailed discussion on the physical meaning of material properties (e.g. [22]). Eq. 1(a-c) ensure that material parameters stay within a specified range of perceptually stiff materials. Eq. 1(d-e) allow changing the internal geometry of the object while ensuring that the boundary of the domain remains the same. And Eq. 1(f-h) ensure that the moments of the object are the same. Together, the constraints enforce our notion of "imperceptible perturbation" to the rigid body simulator - the object is made of only highly stiff materials (such that one may reasonably choose to simulate with rigid body techniques - see Supplemental D), the object has the same external geometry to the reference (so that it experiences identical contacts), and it has identical moments of mass (meaning identical response to the reference in rigid body simulation due to the reduced degrees of freedom).

Typically, a rigid body simulator takes as input the boundary of the domain $\partial\Omega$ as, e.g., a triangle mesh and the moments as parameters. We choose to discretize our problem by tetrahedralizing the interior of the input mesh and treating Y, ν, ρ, and α as per-tetrahedron quantities:

$$\underset{Y,\,\nu,\,\rho,\,\alpha}{\text{argmax}} \quad \sum_i V_i ||q_i^{\text{adv}}(t_{\text{end}}) - q_i^{\text{ref}}(t_{\text{end}})||^2 \tag{2}$$

$$\text{s.t.} \quad Y_i \in [Y_{\min}, Y_{\max}] \quad \forall i \tag{2a}$$

$$\nu_i \in [\nu_{\min}, \nu_{\max}] \quad \forall i \tag{2b}$$

$$\rho_i \in [\rho_{\min}, \rho_{\max}] \quad \forall i \tag{2c}$$

$$\alpha_i \in \{0, 1\} \quad \forall i \in \Omega^{\text{o}} \tag{2d}$$

$$\alpha_i = 1 \quad \forall i \in \partial\Omega \tag{2e}$$

$$\sum_i V_i \rho_i \alpha_i = m_0 \tag{2f}$$

$$\sum_i \rho_i \alpha_i \int_i x dV = m_1 \tag{2g}$$

$$\sum_i \rho_i \alpha_i \int_i ((x \cdot x)I - xx^T) dV = m_2, \tag{2h}$$

where i refers to an individual tetrahedron in Ω (now represented with a tetrahedral mesh: V, T).

3.1. Dealing with the Constraints

While the objective function in Eq. 2 is very simple, dealing with its constraints requires some care.

3.1.1 Material Constraints

The box constraints from Eq. 2(a-c) are used to ensure that the adversarial object is made up of physically reasonable, perceptually rigid materials by setting an allowable range of Young's moduli, Poisson's ratio, and density.

While the material occupancy constraint from Eq 2(d) is a discrete value of either 0 or 1, we follow the SIMP approach of treating it as a continuous variable between $[\alpha_{\min}, 1]$ where the optimization drives it towards those endpoints, and then snapping it to the discrete values as a post-processing step. This allows elements ranging from fully occupied to unoccupied, while not running into the simulation errors that would arise from 0 occupancy elements [67]. To satisfy the boundary occupancy constraint from Eq. 2(e), we simply set the values of α corresponding to boundary elements to 1, and reduce the α degrees of freedom to the interior elements.

We satisfy these constraints by parameterizing the materials with new per element variables $\theta_{\{Y,\nu,\rho,\alpha\}}$:

$$\begin{aligned} x &= f_{\text{param}}(\theta, x_{\min}, x_{\max}) \\ &= \frac{1}{2}\tanh(\theta)(x_{\max} - x_{\min}) + \frac{1}{2}(x_{\max} + x_{\min}), \end{aligned} \tag{3}$$

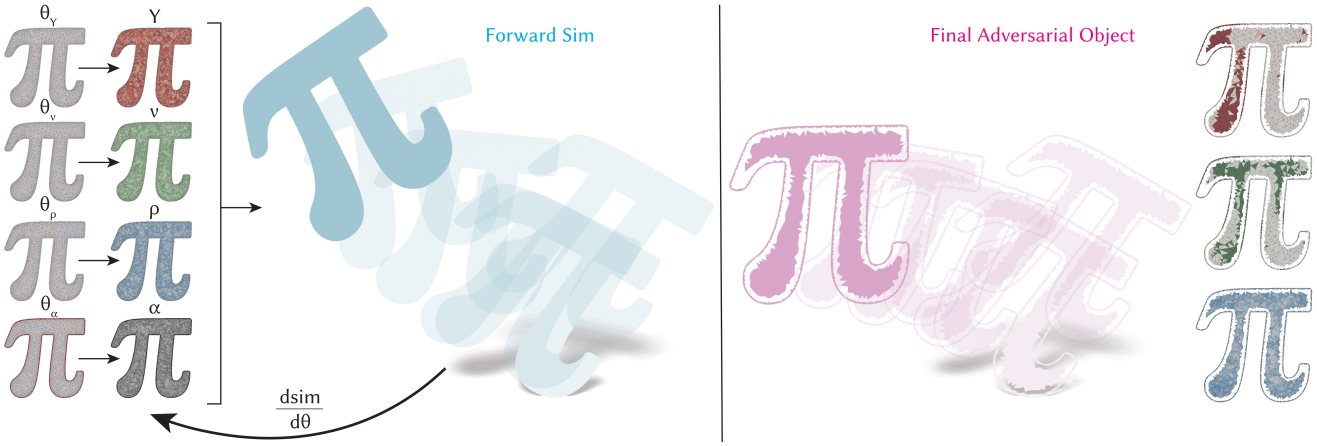

Figure 4. Our differentiable physics simulator allows us to construct adversarial objects using a first order optimization method.

where x is a stand-in for a material parameter, θ is its corresponding parameterized variable, and x_{min} and x_{max} are its minimum and maximum value. Now any value of θ will correspond to a material parameter in the allowable range, and thus implicitly satisfying the constraints while using an unconstrained optimization over θ (see Fig 4).

3.1.2 Moments of Mass Constraints

Finding a physically plausible occupancy and mass density distribution over the object that satisfies the mass moments constraints from Eq. 2(f-h) is straightforward, and there are likely many such distributions for given target moments given the very large number of degrees of freedom compared to the number of constraints.

We can compute the mass moments exactly via quadrature rule, using the element volumes, mass densities $\rho = f_{\text{param}}(\theta_\rho)$, and material occupancies $\alpha = f_{\text{param}}(\theta_\alpha)$. Since the material parameters are constant per element, this ultimately means that we can construct a moments of mass operator $S(V,T) \in \mathbb{R}^{10 \times |T|}$ that acts on the per element (effective) density to compute the mass moments with the quadrature rule. Each row in S corresponds to a different moment. Then, the adversarial object's moments are found as a matrix-vector multiplication:

$$M_{adv} = S(\alpha \odot \rho), \tag{4}$$

where $\odot$ refers to the Hadamard product (and $\alpha \odot \rho$ is the effective density). We choose to enforce Eq. 2(f-h) via soft constraint.

3.2. Final Optimization Problem

Using the parameterization scheme and soft constraint formulation for the moments of mass, we simplify the optimization problem in Eq. 2:

$$\begin{aligned} \underset{\theta_\Upsilon,\theta_\nu,\theta_\rho,\theta_\alpha}{\text{argmin}} \quad & -||q_{\text{adv}}(t_{\text{end}}) - q_{\text{ref}}(t_{\text{end}})||_M^2 + \\ & \beta\, ||M_{\text{ref}} - M_{\text{adv}}||^2, \end{aligned} \tag{5}$$

which we can be solved using a standard first-order descent method - we choose ADAM [51] (see Fig 4). We determine β experimentally. While the use of soft constraint means the mass moments of the adversarial object are not guaranteed to exactly match those of the reference, appropriate choice of the coefficient β results in very low error (see Supplemental C, Table 1) such that rigid body trajectory differences will be negligible on the timescales of interest.

3.3. Forward Simulation

Due to the increased stability and the ability to take larger timesteps, implicit integrators are commonly used for elastodynamics simulation. We use BDF-2 due to it's superior damping performance in comparison to implicit Euler [27].

Using one of the standard techniques in simulation, we pose each dynamic step as the solution to a non-linear optimization problem where forces (including contact) are described as gradients of continuous potential energy functions (see e.g. [30, 57]). Our hyperelastic constitutive model is the stable Neohookean energy from Kim and Eberle [50], and our contact forces are from the smoothly clamped barrier energy in Li et al. [57]. The "energy" in each time step is fully differentiable, but to compute the solution we may require many Newton iterations with line-search. For an explicit formulation, see Supplemental (A).

It is now clear how the material parameters are incorporated; the mass matrix M must account for the densities and occupancies, and the hyperelastic strain energy E_Ψ must account for the Young's modulus, Poisson's ratio, and occupancies. Following the topology optimization literature, we raise α to the power of 3 when it is used to scale the stiffnesses in E_Ψ in order to promote occupancy sparsity [10].

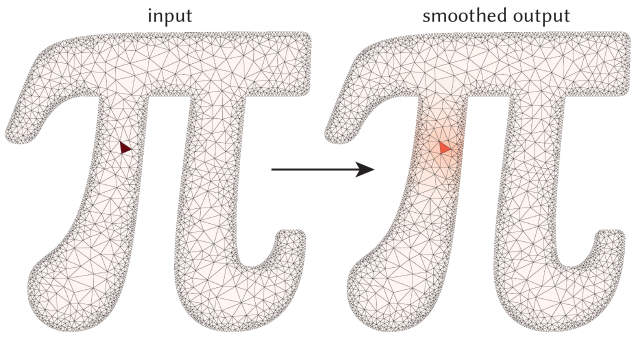

Figure 5. Our geometrically motivated smoothing operator is used to smooth an element defined function on an irregular mesh.

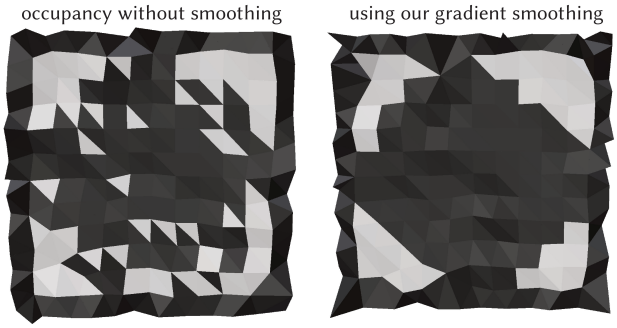

Figure 6. Using our geometrically motivated smoothing operator on the material occupancy gradients of this adversarial cube, we can avoid the checkerboarding artifacts that otherwise appear.

From the total energy of the system, we can can analytically calculate its gradient and Hessian (which we project to be positive definite), and run a Newton solve to convergence at each timestep until the end of the simulation.

3.4. Simulation Gradients

Our optimization problem is now well posed, but simply backpropagating through the nested loops over timestep and Newton iterations is untenable. Like much of the inverse design community, we instead leverage the Adjoint Method (summarized in [66]), which is used to differentiate through constrained optimization problems of the form:

$$\begin{aligned} &\underset{\theta}{\text{argmin}}\ f(q,\theta) \\ &\text{s.t. } g(q,\theta) = 0, \end{aligned} \tag{6}$$

where q is the state, θ is the design parameters to optimize, $f(q,\theta)$ is the cost function, $g(q,\theta)$ is the constraint. Then, the gradient can be calculated as:

$$\frac{df}{d\theta} = \frac{\partial f}{\partial \theta} - \left(\frac{\partial f}{\partial q}\left(\frac{\partial g}{\partial q}\right)^{-1}\right)\frac{\partial g}{\partial \theta}. \tag{7}$$

By using the Adjoint method to provide the derivatives for each timestep, we can avoid the massive performance hit of using auto-differentiation through Newton solves while still taking advantage of the convenience afforded by auto-differentiation frameworks [45]. The optimization problem from the forward simulation is implicitly in the form of the adjoint problem:

$$q_{t+1} = argmin_q\ E(q,\theta)\ \text{ s.t. }\ G(q,\theta) = 0,$$

where E is the minimization objective and G its gradient ($\frac{\partial E}{\partial q_{t+1}}$). See Supplemental (B) for implementation details.

3.4.1 Gradient Smoothing

One limitation of nodal/element based topology optimization is the existence of the so-called non-physical "checkerboard patterns" where the optimized material occupancies are allocated such that patches are connected only at the element corners due to mesh connectivity. The standard approaches to avoid this is to use gradient smoothing, generally referred to in topology optimization as applying a "filter function" [14]. In topology optimization, the design domain is usually discretized as a regular grid, and the filter functions are typically hat functions centered at each element.

As we encounter irregular meshes, we instead use a more geometrically motivated mesh Laplacian based smoothing proposed by Jacobson [43]. The standard cotangent Laplacian L acts on vertex valued functions while our occupancy is per element. So, we instead first construct the false barycentric subdivision of the mesh so that the original elements have corresponding vertices. We can then apply Laplacian blurring using this subdivided mesh, and use a matrix $N = [0\ I]$, to transfer per element values to the barycenter vertices. Thus, we blur a per element input δ as:

$$\hat{\delta} = N^T(\tilde{M} - \gamma\tilde{L})^{-1}NA\,\delta, \tag{8}$$

where $\tilde{L}$ is the subdivided mesh Laplacian, and $\tilde{M}$ a modified mass matrix for the subdivided mesh, with the barycenter vertices having an area of the original mesh elements and all other vertices having area of 0 (see Fig. 5). This smoothing operator is also conservative - it will preserve the volume weighted integral of the input function.

We apply this smoothing method on the gradient of the material occupancies (θ_α) to avoid the aforementioned checkerboarding artifacts (see Fig. 6). The smoothing matrix from Eq. 8 is of size $|T| \times |T|$, while the θ_α is of size $|T|_{\text{interior}}$, so we use selection matrices to apply it.

3.5. Postprocessing

In SIMP, raising α to the power of 3 will drive the occupancies towards extremes by penalizing intermediary values - the effective volume is proportional to α but the effective stiffness is less than proportional (see Bendsøe and Sigmund [10] for a derivation and in depth discussion). However, this is not enough to guarantee no intermediate occupancy values. Therefore, as a post-processing step after the

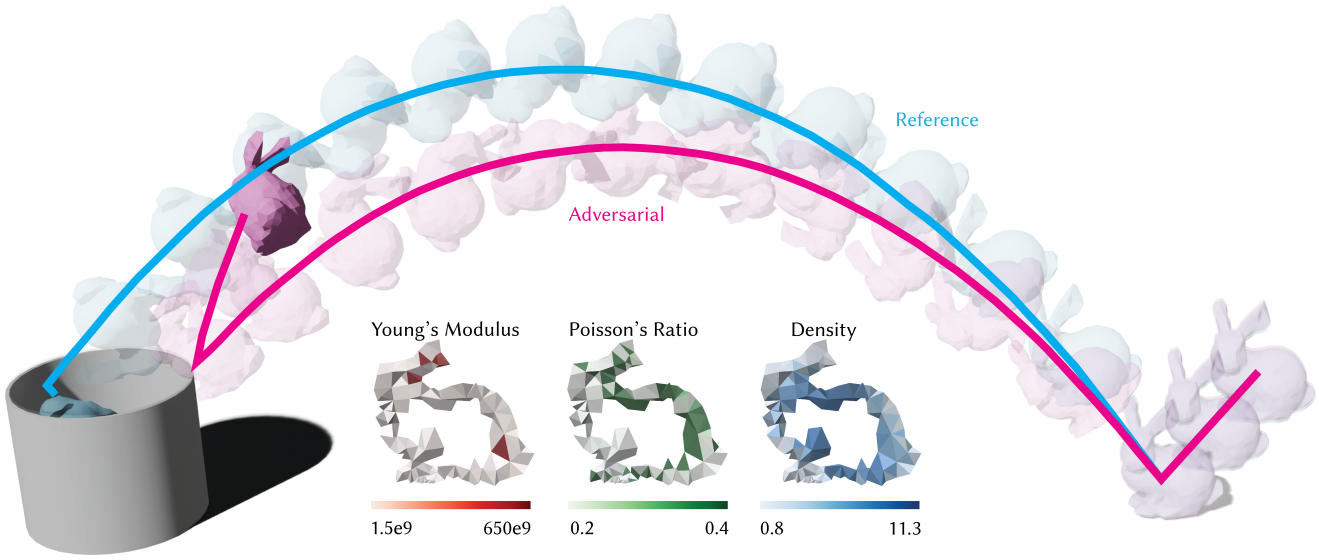

Figure 7. Starting with a reference bunny (blue) which has a trajectory going into a bin, we construct an adversarial bunny (pink). A greater portion of the kinetic energy in the adversarial case is rotational, causing it to travel less distance and instead bounce off the rim of the bin.

materials are determined, we sharpen the occupancies by rounding α so that its elements are either 0 or 1. We then have an optimization to ensure the mass moments constraint is still satisfied:

$$\underset{\theta_\rho}{\text{argmin}} \; ||M_{\text{ref}} - M_{\text{adv}}||^2, \tag{9}$$

which we solve by running gradient descent starting with our previously found θ_ρ.

4. Implementation and Experiments

We implement our simulation in PYTHON, using PYTORCH [72] as our autodifferentiation framework. We use the sparse direct solver CHOLESPY [70] for all of our linear system solves. We use LIBIGL [44] for all necessary geometry processing, and BLENDER and POLYSCOPE [84] for visualization. Our simulations are run with a timestep of 0.01 seconds, and an IPC barrier distance of 10^{-3} unless stated otherwise. We run all of our experiments on CPU on a 2021 M1 Macbook Pro. We demonstrate the power of our method by conducting black box attacks, where our rigid reference objects are simulated in BULLET [20], and adversarial objects are constructed using our custom simulator and demonstrated on POLYFEM [83].

Unless noted otherwise, we use following bounds on material parameters for all examples. The Young's modulus is allowed to range from 2.5 GPa to 650 GPa which corresponds respectively to ABS plastic and tungsten carbide. The Poisson's ratio is allowed to range from 0.2 to 0.4 which corresponds to most metals and plastics. The mass density is allowed to range from 0.8 g/cc to 11.3 g/cc, which corresponds respectively to PMP plastic and lead.

We find that due to the chaotic nature of contact, it is possible to successfully create adversarial results even with very simple object and contact geometries. In Fig. 1, we construct an adversarial ball from just the first contact off the backboard, and the resulting trajectory difference is greatly amplified by the subsequent contacts off the rim. Similarly, in Fig. 7, we construct an adversarial bunny from the first collision off the ground, such that it has a different center of mass trajectory than its reference, with more energy being stored in rotational kinetic energy. This trajectory difference is sufficient for the bunny to bounce off the rim of a bin rather than fall into it. In Fig. 3, we have a more complex geometry, with a star bouncing off of two perpendicular planes. In this case, our optimized optimization includes contact off of both planes. The star example is run with a timestep of 1/30 seconds, and the minimum Young's modulus is raised to 5GPa. Due to the thin shape of the star, we only optimize the Young's modulus, Poisson's ratio, and density, leaving the material occupancy fixed.

In Fig. 8, we show a more complex scenario with a cube striking a stack of three other cubes with a timestep of 1e-3 seconds. Using reference blocks, the stack remains upright post collision, while with adversarial blocks, it falls over. The adversarial cube is created from planar impacts from above and below. Even though our deformable simulation is frictionless, these adversarial cubes are successful at attacking a simulator with a friction coefficient of 0.4.

Thus far, all examples have been undirected attacks. In Fig. 9, we change our cost function to demonstrate a directed attack, where an adversarial bat is constructed specifically so that the trajectory of the struck ball is as close to

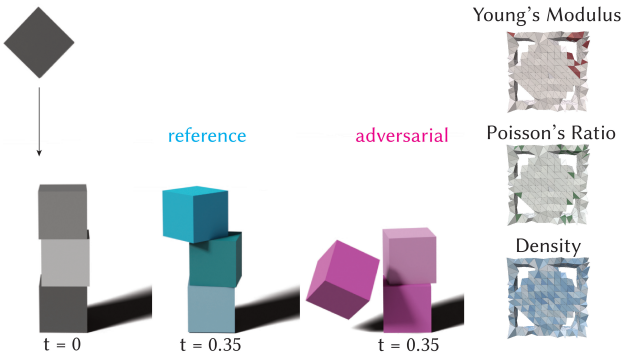

Figure 8. Here, we simulate a collision between a block and a stack of blocks. In the reference blocks (left), the stack is left intact after the collision. Using adversarial blocks (right), the stack falls down.

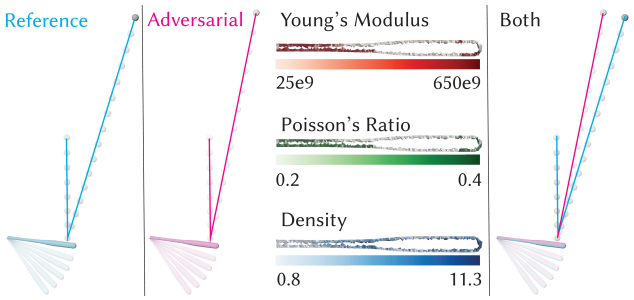

Figure 9. We construct an adversarial bat that aims to get the trajectory of the struck ball as close as possible towards the center (i.e. $x = 0$).

the center as possible causing a significant angular deviation from the reference.

For all of these examples, the comparison of their moments of mass and that of the corresponding reference object is given in Table 1 of Supplemental C. For all reference objects, we determine moments by using a constant mass density of 2.5g/cc. We provide further experimental details in the Supplemental, as well as baseline comparisons with simulations of stiff uniform material objects.

5. Conclusion and Future Work

In this paper we introduce the adversarial setting of rigid body simulators, and present a method of constructing perceptually rigid adversarial objects that will behave identically to a reference object in a rigid body simulation but maximally different in deformable simulations. We do so by implementing a differentiable deformable simulator, and defining a cost function that encodes both the difference in the trajectories of the adversarial and reference object as well as the first three moments of mass. We then minimize this cost using the ADAM algorithm. We have shown over several examples that there can be significant deviations in the trajectories even through simple contact. While it is unknown how common these adversarial objects can accidentally occur or how hard physical adversarial objects would be to manufacture, knowing that they can happen at all is interesting and important in consideration of rigid body simulation tools being used in safety critical applications (e.g. robotics control in industrial settings).

The primary limitation of our method is that the materials parameters are allowed to continuously span some range. This means that even though the material parameters individually are in reasonable ranges, there is no guarantee that a constructed material will actually exist. In the future, it would be interesting to incorporate a materials library that can be populated with real world materials. By our choice of hyperelastic constitutive model, we are limited to isotropic materials. One direction of future work could be to investigate if we can cause even greater trajectory differences by allowing anisotropic materials or multiscale modelling, where even microstructures are considered in the optimization. Another limitation is that our method can occasionally result in disconnected pieces after rounding occupancies, which we currently fix as a postprocess (via `steiner_tree` [34]). We leave more sophisticated topological postprocesses (e.g., [18]) or reparameterizations (e.g., [98]) as future work. Our method currently considers a fully deterministic setting and a single trajectory. It would be interesting to study the construction and performance of adversarial objects in a stochastic setting.

Having identified a deficiency in current simulation techniques, it is interesting to consider how they can be improved. An immediate first approach could be using our adversarial objects to improve robustness of learned physics models through adversarial training. More broadly, stiff dynamics simulation is typically seen as a dichotomy between fast rigid body simulation, and very slow deformable simulation. In computer graphics and animation, there has been research in between those regimes, for example, elastodynamics modelling via adding visual effects to augment rigid body simulation (e.g. [8]). Our work suggests there could be value in studying more accurate methods for stiff materials. Recent work in this area includes Wang et al. [96], who incorporate spatially varying coefficient of restitution to rigid body simulation. Interesting future work in simulation could be to incorporate our adversarial objects to evaluate methods in this area of research with the ultimate goal of sufficiently performant and accurate simulation techniques for stiff objects that can be used in the current robotics and machine learning pipelines.

We hope that our work encourages members of the machine learning and robotics community to investigate whether their rigid body simulations are sufficiently accurate for their needs and further consider what steps can be taken to improve simulator robustness.

References

[1] Grégoire Allaire, François Jouve, and Anca-Maria Toader. A level-set method for shape optimization. *Comptes Rendus Mathematique*, 334(12):1125–1130, 2002. 2

[2] Eirini Anthi, Lowri Williams, Matilda Rhode, Pete Burnap, and Adam Wedgbury. Adversarial attacks on machine learning cybersecurity defences in industrial control systems. *Journal of Information Security and Applications*, 58:102717, 2021. 3

[3] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2019. 2

[4] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 3

[5] Takaaki Azakami, Hiroshi Kera, and Kazuhiko Kawamoto. Adversarial body shape search for legged robots. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 682–687. IEEE, 2022. 3

[6] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4312–4321. International Joint Conferences on Artificial Intelligence Organization, 2021. Survey Track. 3

[7] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. *SIGGRAPH Comput. Graph.*, 23(3):223–232, 1989. 3

[8] Otman Benchekroun, Jiayi Eris Zhang, Siddartha Chaudhuri, Eitan Grinspun, Yi Zhou, and Alec Jacobson. Fast complementary dynamics via skinning eigenmodes. *ACM Trans. Graph.*, 42(4), 2023. 8

[9] Jan Bender, Kenny Erleben, Jeff Trinkle, and Erwin Coumans. Interactive Simulation of Rigid Body Dynamics in Computer Graphics. In *Eurographics 2012 - State of the Art Reports*. The Eurographics Association, 2012. 3

[10] Martin Bendsøe and Ole Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69:635–654, 1999. 5, 6

[11] Martin P. Bendsøe and Ole Sigmund. *Topology Optimization - Theory, Methods, and Applications*. Springer Verlag, Germany, 2003. 2

[12] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural optimization*, 1(4):193–202, 1989. 2

[13] Ravinder Bhattoo, Sayan Ranu, and N M Anoop Krishnan. Learning articulated rigid body dynamics with lagrangian graph neural network. In *Advances in Neural Information Processing Systems*, 2022. 1

[14] Blaise Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50 (9):2143–2158, 2001. 6

[15] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 173–180, 2017. 1

[16] Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *International Conference on Machine Learning*, 2022. 1

[17] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45, 2021. 2

[18] Chao Chen, Daniel Freedman, and Christoph H. Lampert. Enforcing topological constraints in random field image segmentation. In *CVPR 2011*, pages 2089–2096, 2011. 8

[19] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. Who is real bob? adversarial attacks on speaker recognition systems. *2021 IEEE Symposium on Security and Privacy (SP)*, pages 694–711, 2019. 3

[20] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021. 7

[21] Filipe de Avila Belbute-Peres, Kevin A. Smith, Kelsey R. Allen, Joshua B. Tenenbaum, and Zico Kolter. End-to-end differentiable physics for learning and control. In *Neural Information Processing Systems*, 2018. 3

[22] J.P. Den Hartog. *Strength of Materials*. Dover Publications, 2012. 4

[23] Evan Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):231–240, 2008. 3

[24] 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. 3

[25] Alessandro Erba, Riccardo Taormina, Stefano Galelli, Marcello Pogliani, Michele Carminati, Stefano Zanero, and Nils Ole Tippenhauer. Constrained concealment attacks against reconstruction-based anomaly detectors in industrial control systems. In *Proceedings of the 36th Annual Computer Security Applications Conference*, page 480–495, New York, NY, USA, 2020. Association for Computing Machinery. 3

[26] Kenny Erleben. Velocity-based shock propagation for multibody dynamics animation. *ACM Transactions on Graphics (TOG)*, 26(2):12–es, 2007. 3

[27] A. Ern and J.L. Guermond. *Finite Elements III: First-Order and Time-Dependent PDEs*. Springer International Publishing, 2021. 5

[28] Zachary Ferguson, Minchen Li, Teseo Schneider, Francisca Gil-Ureta, Timothy Langlois, Chenfanfu Jiang, Denis Zorin, Danny M. Kaufman, and Daniele Panozzo. Intersection-free rigid body dynamics. *ACM Transactions on Graphics (SIGGRAPH)*, 40(4), 2021. 3

[29] Henry Figueroa, Yi Wang, and George C. Giakos. Adversarial attacks in industrial control cyber physical systems. In *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6, 2022. 3

[30] T. F. Gast and C. Schroeder. Optimization integrator for large time steps. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, page 31–40, Goslar, DEU, 2015. Eurographics Association. 5

[31] Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. Add: analytically differentiable dynamics for multi-body systems with frictional contact. *ACM Trans. Graph.*, 39(6), 2020. 3

[32] Markus Giftthaler, Michael Neunert, Markus Stäuble, Marco Frigerio, Claudio Semini, and Jonas Buchli. Automatic differentiation of rigid body dynamics for optimal control and estimation. *Advanced Robotics*, 31(22):1225–1237, 2017. 3

[33] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 3

[34] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008. 8

[35] James K Hahn. Realistic animation of rigid bodies. *ACM Siggraph computer graphics*, 22(4):299–308, 1988. 3

[36] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S. Sukhatme. Neuralsim: Augmenting differentiable simulators with neural networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9474–9481. IEEE Press, 2021. 3

[37] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, C. Karen Liu, Jan Peters, Shuran Song, Peter Welinder, and Martha White. Sim2real in robotics and automation: Applications and challenges. *IEEE Transactions on Automation Science and Engineering*, 18(2):398–400, 2021. 2

[38] Benjamin Holzschuh, Simona Vegetti, and Nils Thuerey. Solving inverse physics problems with score matching. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 2

[39] I. Hossen and X. Hei. aaecaptcha: The design and implementation of audio adversarial captcha. In *2022 IEEE 7th European Symposium on Security and Privacy*, pages 430–447, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 3

[40] Shengshan Hu, Junwei Zhang, Wei Liu, Junhui Hou, Minghui Li, Leo Yu Zhang, Hai Jin, and Lichao Sun. Pointca: evaluating the robustness of 3d point cloud completion models against adversarial examples. AAAI Press, 2023. 3

[41] Yao Huang, Yinpeng Dong, Shouwei Ruan, Xiao Yang, Hang Su, and Xingxing Wei. Towards transferable targeted 3d adversarial attack in the physical world. *ArXiv*, abs/2312.09558, 2023. 3

[42] Zizhou Huang, Davi Colli Tozoni, Arvi Gjoka, Zachary Ferguson, Teseo Schneider, Daniele Panozzo, and Denis Zorin. Differentiable solver for time-dependent deformation problems with contact. *ACM Trans. Graph.*, 2024. 3

[43] Alec Jacobson. Laplacian smoothing per-triangle values, 2024. 6

[44] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 7

[45] Krishna Murthy Jatavallabhula, Miles Macklin, Florian Golemo, Vikram Voleti, Linda Petrini, Martin Weiss, Breandan Considine, Jerome Parent-Levesque, Kevin Xie, Kenny Erleben, Liam Paull, Florian Shkurti, Derek Nowrouzezahrai, and Sanja Fidler. gradsim: Differentiable simulation for system identification and visuomotor control. *International Conference on Learning Representations (ICLR)*, 2021. 3, 6

[46] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. page 864–872, New York, NY, USA, 2019. Association for Computing Machinery. 3

[47] Zahra Kadkhodaie and Eero P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Advances in Neural Information Processing Systems*, 2021. 2

[48] Robert K. Katzschmann, Cosimo Della Santina, Yasunori Toshimitsu, Antonio Bicchi, and Daniela Rus. Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 454–461, 2019. 3

[49] H. Kim, M. Son, M. Kim, M. Kwon, and C. Kim. Breaking temporal consistency: Generating video universal adversarial perturbations using image models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4302–4311, Los Alamitos, CA, USA, 2023. IEEE Computer Society. 3

[50] Theodore Kim and David Eberle. Dynamic deformables: Implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses*, New York, NY, USA, 2022. Association for Computing Machinery. 3, 5, 1

[51] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[52] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. 3

[53] Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. Affine body dynamics: Fast, stable and intersection-free simulation of stiff materials. *ACM Trans. Graph.*, 41(4), 2022. 3

[54] Itai Lang, Uriel Kotlicki, and Shai Avidan. Geometric Adversarial Attacks and Defenses on 3D Point Clouds. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 1196–1205, 2021. 3

[55] Juncheng Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J. Zico Kolter, and Florian Metze. Adversarial music: Real world audio adversary against wake-word detection system. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 2

[56] Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, and Cewu Lu. Niki: Neural inverse kinematics with invertible neural networks for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12933–12942, 2023. 2

[57] Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics. *ACM Trans. Graph.*, 39(4), 2020. 5, 1

[58] Shasha Li, Ajaya Neupane, S. Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury, and Ananthram Swami. Adversarial perturbations against real-time video classification systems. *ArXiv*, abs/1807.00458, 2018. 3

[59] Shasha Li, Abhishek Aich, Shitong Zhu, M. Salman Asif, Chengyu Song, Amit K. Roy-Chowdhury, and Srikanth V. Krishnamurthy. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. In *Neural Information Processing Systems*, 2021. 3

[60] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019. 1

[61] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM Transactions on Graphics*, 42(1), 2022. 3

[62] C Karen Liu and Dan Negrut. The role of physics-based simulators in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:35–58, 2021. 3

[63] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 3

[64] Xiaolei Liu, Kun Wan, Yufei Ding, Xiaosong Zhang, and Qingxin Zhu. Weighted-sampling audio adversarial example attack. In *AAAI Conference on Artificial Intelligence*, 2019. 3

[65] Justin Matejka and George Fitzmaurice. Same stats, different graphs: Generating datasets with varied appearance and identical statistics through simulated annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1290–1294, New York, NY, USA, 2017. Association for Computing Machinery. 2

[66] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint method. *ACM Transactions On Graphics (TOG)*, 23(3):449–456, 2004. 6

[67] H. P. Mlejnek. Some aspects of the genesis of structures. *Structural optimization*, 5:64–69, 1992. 4

[68] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, 1988. 3

[69] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 3

[70] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (TOG)*, 40(6):1–13, 2021. 7

[71] Takuto Otomo, Hiroshi Kera, and Kazuhiko Kawamoto. Adversarial joint attacks on legged robots. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 676–681. IEEE, 2022. 3

[72] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019. 7

[73] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. *ArXiv*, abs/2010.03409, 2020. 1

[74] Jovan Popović, Steven M Seitz, Michael Erdmann, Zoran Popović, and Andrew Witkin. Interactive manipulation of rigid body simulations. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 209–217, 2000. 2

[75] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Algorithmic Foundations of Robotics X*, pages 527–542, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 3

[76] Zhuang Qian, Kaizhu Huang, Qiu-Feng Wang, and Xu-Yao Zhang. A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies. *Pattern Recognition*, 131:108889, 2022. 3

[77] Yiling Qiao, Junbang Liang, Vladlen Koltun, and Ming Lin. Differentiable simulation of soft multi-body systems. *Advances in Neural Information Processing Systems*, 34: 17123–17135, 2021. 3

[78] Singiresu S. Rao. *The Finite Element Method in Engineering*. Butterworth-Heinemann, sixth edition edition, 2018. 1

[79] Lucas Rath, Andreas René Geist, and Sebastian Trimpe. Using physics knowledge for learning rigid-body forward dynamics with gaussian process force priors. In *Proceedings of the 5th Conference on Robot Learning*, pages 101–111. PMLR, 2022. 1

[80] Junuthula Narasimha Reddy. *Introduction to the finite element method*. McGraw-Hill Education, 2019. 3

[81] Fereshteh Sadeghi, Alexander Toshev, Eric Jang, and Sergey Levine. Sim2real viewpoint invariant visual servoing by recurrent control. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[82] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning

to simulate complex physics with graph networks. *ArXiv*, abs/2002.09405, 2020. 1

[83] Teseo Schneider, Jérémie Dumas, Xifeng Gao, Denis Zorin, and Daniele Panozzo. PolyFEM. https://polyfem.github.io/, 2019. 7

[84] Nicholas Sharp et al. Polyscope, 2019. www.polyscope.run. 7

[85] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *Acm siggraph 2012 courses*, pages 1–50. 2012. 3

[86] Breannan Smith, Fernando De Goes, and Theodore Kim. Stable neo-hookean flesh simulation. *ACM Trans. Graph.*, 37(2), 2018. 3

[87] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[88] Tomer Stolik, Itai Lang, and Shai Avidan. Saga: Spectral adversarial geometric attack on 3d meshes. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4261–4271, 2022. 3

[89] Michael Strecke and Joerg Stueckler. Diffsdfsim: Differentiable rigid-body dynamics with implicit shapes. In *2021 International Conference on 3D Vision (3DV)*, pages 96–105, 2021. 3

[90] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2014. 2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014. 3

[91] M. Tanaka and G.S. Dulikravich. *Inverse Problems in Engineering Mechanics*. Elsevier Science, 1998. 2

[92] Tencent. Experimental security research of tesla autopilot, 2019. 2

[93] J. C. Trinkle, J. A. Tzitzouris, and J. S. Pang. Dynamic multi-rigid-body systems with concurrent distributed contacts. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, 359(1789):2575–2593, 2001. 3

[94] Christopher D Twigg and Doug L James. Backward steps in rigid body simulation. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008. 2, 3

[95] Curtis R. Vogel. *Computational Methods for Inverse Problems*. Society for Industrial and Applied Mathematics, 2002. 2

[96] Jui-Hsien Wang, Rajsekhar Setaluri, Doug L. James, and Dinesh K. Pai. Bounce maps: an improved restitution model for real-time rigid-body impact. *ACM Trans. Graph.*, 36(4), 2017. 8, 4

[97] Michael Yu Wang, Xiaoming Wang, and Dongming Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192 (1):227–246, 2003. 2

[98] Yu Wang and Justin Solomon. Fast quasi-harmonic weights for geometric data interpolation. *ACM Trans. Graph.*, 40 (4), 2021. 8

[99] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. AAAI Press, 2019. 3

[100] Andrew Witkin and Michael Kass. Spacetime constraints. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, pages 159–168, New York, NY, USA, 1988. Association for Computing Machinery. 2

[101] Y.M. Xie and G.P. Steven. A simple evolutionary procedure for structural optimization. *Computers & Structures*, 49(5): 885–896, 1993. 2

[102] Chen Xu, Ba Trung Cao, Yong Yuan, and Günther Meschke. Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios. *Computer Methods in Applied Mechanics and Engineering*, 405:115852, 2023. 2

[103] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020. 2

[104] Gang Yang, Siyuan Luo, Yunhai Feng, Zhixin Sun, Chenrui Tie, and Lin Shao. Jade: A differentiable physics engine for articulated rigid bodies with intersection-free frictional contact. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 16915–16922, 2024. 3

[105] Tarun Yenamandra, Florian Bernard, Jiayi Wang, Franziska Mueller, and Christian Theobalt. Convex optimisation for inverse kinematics. In *2019 International Conference on 3D Vision (3DV)*, pages 318–327, 2019. 2

[106] Bangjie Yin, Wenxuan Wang, Taiping Yao, Junfeng Guo, Zelun Kong, Shouhong Ding, Jilin Li, and Cong Liu. Adv-makeup: A new imperceptible and transferable attack on face recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1252–1258. International Joint Conferences on Artificial Intelligence Organization, 2021. Main Track. 2

[107] Zhiyuan Yu, Yuanhaur Chang, Ning Zhang, and Chaowei Xiao. SMACK: Semantically meaningful adversarial audio attack. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3799–3816, Anaheim, CA, 2023. USENIX Association. 3

[108] Jonas Zehnder, Yue Li, Stelian Coros, and Bernhard Thomaszewski. Ntopo: Mesh-free topology optimization using implicit neural representations. In *Advances in Neural Information Processing Systems*, pages 10368–10381. Curran Associates, Inc., 2021. 2

[109] John Z. Zhang, Yu Zhang, Pingchuan Ma, Elvis Nava, Tao Du, Philip Arm, Wojciech Matusik, and Robert K. Katzschmann. Sim2real for soft robotic fish via differentiable simulation. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12598–12605, 2021. 2