SECOND-ORDER NEURAL CONTACT POTENTIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

While conventional contact models are often defined over separate discrete contact cases, continuous surface-integral potentials provide a unified alternative. However, approximating these potentials with neural networks places new demands on smoothness and derivative quality. We introduce Second-Order Neural Contact Potentials (SNCP), a smooth neural representation for sampling-free contact handling in physics-based simulation. Building on Neural Collision Fields (NCF), which replace explicit contact sampling with a learned integral over pairs of mesh primitives, we address two key limitations that prevent their use in second-order solvers: nonsmooth derivatives and difficulties representing energies near zero. Our formulation replaces the Gaussian kernel of NCF with a C^2-continuous Wendland polynomial and introduces a neural parameterization that enforces a smooth outer cutoff and supports consistent differentiation, yielding stable and well-behaved first- and second-order derivatives. The resulting contact potential generalizes to different meshes without retraining and supports both position-based dynamics and Newton-based finite-element methods. Across challenging simulations involving rigid bodies, volumetric deformable bodies, and thin shells, our method produces accurate and stable contact responses while enabling robust second-order optimization.

1 INTRODUCTION

Contact handling is essential to physics-based simulation of rigid bodies, deformable solids, and thin shells. Detecting an intersection alone is insufficient: a simulator also needs a measure of contact and a direction in which to resolve it. Contact-point methods construct this response from selected samples and normals, requiring special treatment of vertex–face and edge–edge configurations. Their choices of sample locations and density can change the simulated response, particularly near sharp features and extended contact regions (Erleben, 2018).

Surface integration provides an alternative geometric basis for this response. Integrating a proximity kernel over both surfaces aggregates interactions between nearby regions of the two surfaces, treating point, edge, and face contacts within one expression rather than choosing representative contacts (Zesch et al., 2023; Minarčík et al., 2026). Interpreted as a contact energy, this scalar defines contact forces through its gradient. Because the exact integral depends only on geometry, subdividing mesh elements merely repartitions the integration domain without altering the physical interaction. The construction also requires no enclosed volume, making it applicable to both solid boundaries and thin shells. These properties motivate defining contact through a surface integral, yet repeated numerical quadrature remains expensive. Neural Collision Fields (NCF) (Zesch et al., 2023) addresses this cost by learning the integral for triangle pairs, yielding a reusable contact primitive for new meshes without retraining. Its position-based dynamics results demonstrate the promise of this sampling-free formulation.

Extending NCF to Newton-based solvers requires consistent gradients and Hessians from the learned potential throughout each time-step solve. Two challenges arise. First, a smooth integral does not necessarily yield a smooth learned potential: NCF's ReLU network introduces derivative discontinuities even though its Gaussian kernel is smooth. Geometric preprocessing and support construction must also be differentiated, and closest-feature switches or unsigned distance at contact can introduce additional nonsmoothness. Second, contact energies approach zero as interactions disappear. Direct value fitting provides weak supervision near zero, whereas a logarithmic parameterization is singular as the energy vanishes. Small energy errors alone consequently provide no guarantee on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the forces and curvature that drive the solver. The key challenge is to represent contact onset and vanishing energy robustly for second-order optimization.

We introduce Second-Order Neural Contact Potentials (SNCP), a learned surface-integral potential for second-order contact handling. A compactly supported C^2 Wendland kernel (Wendland, 1995) defines the finite-range interaction. We factor its neural approximation into a prescribed geometry envelope and a coefficient learned in log space. This separates outer-cutoff behavior from learned geometric variation and accommodates vanishing energies. Analytical differentiation of the complete potential supplies the gradients and Hessians for Newton-based simulation. The same trained model supports position-based contact constraints and Newton-based finite-element simulation, and is reused across rigid bodies, volumetric deformable bodies, and thin shells without retraining.

Our contributions are summarized as follows:

- A finite-range, C^2-continuous triangle-pair contact energy based on a surface integral;
- A structured neural parameterization that prescribes outer support analytically while learning the remaining geometric variation in log space;
- A Newton-compatible contact formulation that improves convergence and robustness over prior neural approaches across challenging rigid, deformable, and thin-shell simulations.

2 Related Work

Contact methods. Classical penalty methods approximate non-penetration constraints by penalizing interpenetration with finite contact stiffness (Armero & Petőcz, 1998; Chen et al., 2024). Barrier methods use potentials that diverge as the contact gap vanishes. Incremental Potential Contact (IPC) (Li et al., 2020) combines these potentials with continuous collision detection for intersection-free simulation. Extensions address thin structures and rigid or nearly rigid bodies (Li et al., 2021; Ferguson et al., 2021; Lan et al., 2022), while GIPC and StiffGIPC improve curvature treatment and GPU solvers (Huang et al., 2024; 2025a). Integral formulations additionally describe how interactions are distributed over geometry. Convergent IPC and Geometric Contact Potential develop continuum surface formulations to address discretization-dependent response (Li et al., 2023; Huang et al., 2025b). Related approaches derive contact forces from overlap-volume energies (Wang et al., 2026) or use surface repulsion for mesh untangling (Minarčík et al., 2026). SNCP follows the surface-integral approach, approximating a finite-range triangle-pair interaction defined by a compactly supported Wendland kernel (Wendland, 1995).

Learned contact representations. Neural methods represent different aspects of contact, from geometry to deformation and local interaction energies. DeepSDF (Park et al., 2019) learns continuous signed-distance representations of shapes, while Cai et al. (2022) learn signed distances in reduced deformation spaces for self-collision handling. Contact deformation networks predict displacement fields conditioned on the collider (Romero et al., 2022; 2023), and energy-based physics-informed networks learn deformations by minimizing a prescribed contact functional (Bai et al., 2025). The closest prior work, Neural Collision Fields (NCF) (Zesch et al., 2023), learns a surface integral of a Gaussian kernel over a triangle pair that transfers across meshes without retraining and serves as a position-based dynamics constraint. SNCP retains this reusable local representation, introducing a compactly supported target energy and a smooth, factored potential whose gradients and Hessians support Newton-based simulation. The network supplies the contact energy, while the numerical solver computes the resulting deformation.

Structured neural potentials and derivative learning. Prescribed factors can encode geometric constraints in neural functions, including boundary conditions imposed through distance functions (Sukumar & Srivastava, 2022) and smooth local cutoffs in molecular models (Unke & Meuwly, 2019; Gasteiger et al., 2020). Derivative information is also central to Sobolev training (Czarnecki et al., 2017) and energy-consistent molecular force fields (Chmiela et al., 2017; Schütt et al., 2017). SIREN supplies smooth coordinate networks for representing signals and their derivatives (Sitzmann et al., 2020). SNCP combines these ideas in triangle-pair configuration space: a geometric envelope prescribes a conservative outer cutoff, while a positive neural coefficient captures the remaining integral variation. We train the coefficient in log space with first-derivative supervision and differentiate the complete potential for Newton optimization.

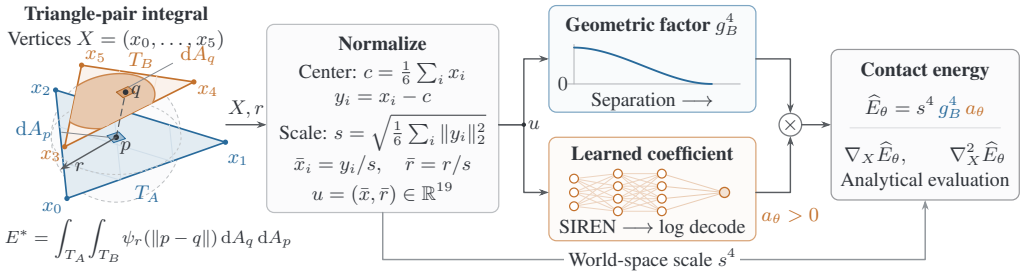

Figure 1: **SNCP overview.** The reference energy integrates a compactly supported kernel over triangles T_A and T_B; the shaded region on T_B lies within radius r of p. The six vertices $X = (x_0, \ldots, x_5)$ and radius r are normalized to form the input u. An analytic geometric factor g_B^4 controls the support, while a strictly positive learned coefficient a_θ captures the remaining variation. Multiplication by s^4 restores world-space scaling. Differentiating the complete reconstruction, including normalization and the geometric factor, yields consistent gradients and Hessians for simulation.

3 Preliminaries

Implicit time integration. Let X denote the physical degrees of freedom, v their velocities, M a constant mass matrix, and $V(X)$ the total potential energy. For conservative dynamics $M\dot{v} = -\nabla V(X)$, implicit Euler with time step Δt gives

$$X^{n+1} = X^n + \Delta t\, v^{n+1}, \qquad M\frac{v^{n+1} - v^n}{\Delta t} = -\nabla V(X^{n+1}). \tag{1}$$

Eliminating v^{n+1} yields the stationarity condition $\nabla\Phi(X^{n+1}) = 0$ for the incremental potential

$$\Phi(X) = \frac{1}{2}(X - \widehat{X})^\top M(X - \widehat{X}) + \Delta t^2 V(X), \qquad \widehat{X} = X^n + \Delta t\, v^n. \tag{2}$$

Here $\widehat{X}$ is the inertial prediction. We seek a local minimizer of Φ using Newton-type optimization, with iterations of the form

$$H_k p_k = -\nabla\Phi(X_k), \qquad X_{k+1} = X_k + \alpha_k p_k. \tag{3}$$

H_k is $\nabla^2\Phi(X_k)$ or a stabilized approximation. α_k is chosen by line search. Each term in V contributes its value to objective evaluation, gradient to the force, and Hessian to the search direction.

Neural Collision Fields. NCF learns a smoothed triangle-pair surface integral, replacing repeated numerical quadrature with a network reusable across meshes. Its target uses the exponential kernel of Zesch et al. (2023, Eq. (5)), written in our notation as

$$E_{\mathrm{NCF}}(T_A, T_B; k) = \int_{T_A}\int_{T_B} \exp\!\left(-k\|p - q\|_2^2\right)\,\mathrm{d}A_q\,\mathrm{d}A_p, \tag{4}$$

where $p \in T_A$ and $q \in T_B$ are surface points and $k > 0$ controls the kernel's sharpness. The learned field serves as a contact constraint in a position-based solver (Müller et al., 2007). Three limitations arise: (i) the exponential kernel has nonzero tails; (ii) ReLU activations and max-radius normalization introduce nonsmooth branch boundaries; and (iii) the native evaluator's gradient proxy generally differs from the derivative of its world-space constraint because it omits normalization and scale derivatives (Appendix A.4). These issues motivate SNCP's compact support, smooth parameterization, and consistent world-space differentiation.

4 Method

SNCP approximates a finite-range surface contact energy with a neural coefficient multiplied by a prescribed geometric factor. The construction has four parts: a triangle-pair integral, a conservative support coordinate, coefficient learning in log space, and analytical differentiation of the complete potential for Newton-based implicit simulation. See Fig. 1 for an illustration.

4.1 FINITE-RANGE TRIANGLE-PAIR ENERGY

Let $T_A = (x_0, x_1, x_2)$ and $T_B = (x_3, x_4, x_5)$ be two nondegenerate linear triangles, with vertices in $\mathbb{R}^3$. For an interaction radius $r > 0$, we use the compactly supported C^2 polynomial of Wendland form (Wendland, 1995)

$$\psi_r(\rho) = \begin{cases} (1-\rho/r)^3(3\rho/r+1), & 0 \le \rho < r, \\ 0, & \rho \ge r. \end{cases} \tag{5}$$

The reference contact energy integrates this kernel over both surfaces:

$$E^*(T_A, T_B; r) = \int_{T_A}\int_{T_B} \psi_r(\|p-q\|_2)\ \mathrm{d}A_q\ \mathrm{d}A_p. \tag{6}$$

The integral aggregates interactions across the primitives without explicitly selecting a vertex–face or edge–edge contact feature. It is nonnegative and exactly zero when the triangle distance is at least r, defining a finite-range, frictionless penalty. We distinguish this exact integral E^*, its stored numerical estimate E_{num}, and the learned prediction $\widehat{E}_\theta$ throughout the construction.

Normalization and scale. We remove translation and joint scale variation using the centroid and RMS radius of the six vertices:

$$c = \frac{1}{6}\sum_{i=0}^{5} x_i, \qquad s = \left(\frac{1}{6}\sum_{i=0}^{5}\|x_i - c\|_2^2\right)^{1/2}, \qquad \bar{x}_i = \frac{x_i - c}{s}, \qquad \bar{r} = \frac{r}{s}. \tag{7}$$

The network input is $u = (\bar{x}_0, \ldots, \bar{x}_5, \bar{r}) \in \mathbb{R}^{19}$. Let $\bar{T}_A$ and $\bar{T}_B$ denote the normalized triangles. Each surface measure contributes a factor s^2, giving

$$E^*(T_A, T_B; r) = s^4 \bar{E}^*(\bar{T}_A, \bar{T}_B; \bar{r}). \tag{8}$$

We thus learn the normalized energy and use the same exact s^4 lift for its world-space prediction.

4.2 CONSERVATIVE SUPPORT FACTORIZATION

Unconstrained neural regression does not enforce compact support when approximating targets with finite interaction cutoffs. We structurally enforce this boundary behavior by factorizing the prediction into an analytic geometric factor and a learned coefficient

$$\widehat{\bar{E}}_\theta(u) = g_B(u)^4 a_\theta(u), \qquad a_\theta(u) > 0, \tag{9}$$

where the support coordinate g_B is determined by geometry, and a_θ is parameterized through the network to learn the integral's remaining geometric variation.

Requirements for the support coordinate. Beyond strictly yielding a cutoff that encloses the true interaction support, the resulting potential must have continuous first and second derivatives for Newton optimization. We use the fourth power in Eq. (9) to enforce the smoothness condition at the outer cutoff, but this alone would not be sufficient against derivative discontinuities occurring inside the active zone. For this reason, the canonical exact-distance gap

$$g_{\mathrm{exact}} = [\bar{r} - d]_+, \qquad [a]_+ = \max(a, 0), \tag{10}$$

where $d = d_{\min}(\bar{T}_A, \bar{T}_B)$ is the distance between the normalized triangles, would fail to satisfy the above desiderata because d can have discontinuous derivatives when the triangles touch or when the closest vertex–face or edge–edge configuration changes.

Our smooth conservative approximation. We construct the implemented coordinate from a smooth conservative approximation to the squared distance $q_B \le d^2$ and define

$$g_B = \frac{[\bar{r}^2 - q_B]_+}{2\bar{r}}. \tag{11}$$

The envelope remains active wherever the exact integral is positive and vanishes beyond a conservative outer cutoff. An interior barrier problem supplies this approximation; its construction, support bound, boundary regularity, and derivatives are given in Appendix A.2.

4.3 LEARNING THE RESIDUAL COEFFICIENT

We fit the coefficient in log space to accommodate its variation in magnitude. For an active pair ($g_B > 0$), the numerical coefficient target is

$$C_{\mathrm{num}} = \frac{\bar{E}_{\mathrm{num}}}{g_B^4}. \tag{12}$$

Let z^* denote its logarithmic encoding. We standardize this target as

$$y^* = \frac{z^* - \mu}{\sigma}, \tag{13}$$

where μ and σ are estimated from the training split only. The network $f_\theta(u)$ predicts y^*. We define the analytical coefficient by reversing the standardization and exponentiating:

$$a_\theta(u) = 10^{\mu + \sigma f_\theta(u)}. \tag{14}$$

This strictly positive coefficient captures the integral's remaining geometric variation. All analytical derivations use Eq. (14).

Network and supervision. We use a 19-input coordinate SIREN (Sitzmann et al., 2020) with 8 hidden layers of width 512, sine frequency $\omega = 1$, and a linear scalar output. Its smooth activations support analytical input derivatives through second order. The training objective is

$$\mathcal{L} = \mathrm{Huber}_1(f_\theta, y^*) + 0.5\,\mathcal{L}_{\mathrm{Sob}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{Sob}}$ compares the direction and logarithmic magnitude of $\nabla_u f_\theta$ and a transformed derivative target h^* computed from the numerical labels. This first-order supervision includes all 19 input channels and the derivative of g_B in the target transformation. It does not directly supervise world-space forces or Hessians. Proper rotations, independent within-triangle vertex permutations, and triangle exchange augment the data; these symmetries are encouraged rather than enforced by the network. Appendix C specifies the loss and training procedure.

Data generation. We sample nondegenerate triangle pairs at varying scales, position them in overlapping or near-contact configurations, and apply joint centroid/RMS normalization. Adaptive quadrature supplies the numerical energy label $\bar{E}_{\mathrm{num}}$ and estimates of its 19 input derivatives. At each refinement step, the algorithm compares an order-15 estimate with the sum over four child pairs formed by longest-edge bisection of both triangles. It recursively refines based on the change in estimated energy and prunes pairs outside the kernel support; derivative labels use the same quadrature partition. The dataset contains 100 million examples, with 85M/10M/5M training, validation, and test splits. Sampling and quadrature details are given in Appendix C.

4.4 ANALYTICAL DERIVATIVES AND NEWTON ASSEMBLY

Differentiating the complete potential. The barrier problem defining q_B has four independent barycentric variables z (Appendix A.2). Let w collect the 18 normalized vertex coordinates and $F(z, w) = 0$ denote the stationarity system. Its positive-definite Jacobian $K = F_z$ permits implicit differentiation of the optimizer $z_B(w)$:

$$\frac{\partial z_B}{\partial w} = -K^{-1} F_w. \tag{16}$$

Differentiating this system again gives the optimizer's second derivatives. These responses are necessary because q_B uses the squared separation at the optimizer, rather than the minimized barrier objective. We combine them with the analytical SIREN derivatives and propagate through q_B, g_B, and the coefficient decoder to obtain the normalized energy gradient and Hessian.

Let $X_{AB} \in \mathbb{R}^{18}$ collect a pair's world-space vertices and $J = \partial u / \partial X_{AB}$. Even with world-space r fixed, moving a vertex changes the normalized coordinates, $\bar{r}$, and the scale lift. For $\widehat{E}_{AB} = s^4 \widehat{\bar{E}}_\theta(u)$, the full gradient is

$$\nabla_{X_{AB}} \widehat{E}_{AB} = s^4 J^\top \nabla_u \widehat{\bar{E}}_\theta + 4 s^3 \widehat{\bar{E}}_\theta \nabla_{X_{AB}} s. \tag{17}$$

The Hessian includes the corresponding second derivatives and cross terms. Appendix A gives the complete analytical formulas. Derivative evaluation holds candidate-set membership fixed.

Implicit solve. Let X be the global degrees of freedom, $\widehat{X}$ their inertial prediction, M the mass matrix, Ψ the elastic energy, and $\mathcal{C}$ the triangle-pair candidate set. Implicit Euler minimizes

$$\Phi(X) = \frac{1}{2}(X - \widehat{X})^\top M(X - \widehat{X}) + \Delta t^2 \left[\Psi(X) + \kappa \sum_{(T_A, T_B) \in \mathcal{C}} \widehat{E}_{AB}(X_{AB}(X)) \right], \tag{18}$$

where κ is the contact stiffnesses.

5 Experiments

We evaluate SNCP in both Newton and XPBD frameworks. Our Newton implementation uses libuipc (Huang et al., 2024; 2025a), and our XPBD implementation extends the reproduced NCF pipeline (Zesch et al., 2023). Appendix B describes the potential evaluator and its integration into each solver. All experiments run on a workstation with an Intel Xeon w7-3445 CPU and NVIDIA RTX 5880 Ada GPU. Please refer to our supplementary video for high-quality dynamic visualizations.

5.1 Comparison

Baselines. We distinguish the NCF method (Zesch et al., 2023) from its Newton adaptation and compare both with SNCP:

(i) **NCF/XPBD:** we reproduce the full NCF pipeline with the network and the XPBD simulator, retaining the native contact rule and gradient proxy.
(ii) **NCF/Newton:** our adaptation of NCF's contact energy to the Newton solver, with full world-space derivatives. Its ReLU and max-radius branch boundaries lack a global C^2 guarantee, which can hinder Newton convergence. We continue the simulation after the solver reaches the limit of 100 Newton iterations to examine the resulting trajectory, recording convergence separately from advancement.
(iii) **SNCP/XPBD:** our potential inserted into the same reproduced position-based pipeline through an energy-derived contact constraint.
(iv) **SNCP/Newton (ours):** our potential with analytical gradients and Hessians in the Newton runtime.

See Appendix B for implementation details of these methods. The scene configurations and solver settings are described in Appendix G.

Large mass ratio. A heavy elastic cube (side length 10 m, mass $2{,}000{,}000$ kg) falls onto a smaller cube (side length 1 m, mass $1{,}000$ kg) above a fixed ground surface. Both cubes have Young's modulus 3 MPa and Poisson's ratio 0.49, with 21^3 and 3^3 mesh vertices, respectively.

As shown in Fig. 2(a), the XPBD solver fails to resolve the extreme mass ratio, causing inversion, a fundamental artifact stemming from the dual formulation of XPBD (Macklin et al., 2020). This failure highlights that compatibility with Newton's method is essential. When paired with the Newton solver, our method demonstrates clear superiority over NCF: NCF/Newton failed to converge in 167 out of 200 frames. SNCP/Newton simulates the scene robustly and efficiently without any penetrations or inversions, underscoring both the critical need for Newton-compatible formulations and the superior robustness of SNCP under severe mass imbalances.

Three-cube stacking. Three elastic cubes fall onto a fixed truncated-pyramid base. We compare the four contact/solver configurations to test their ability to form and maintain a stack.

As illustrated in Fig. 2(b), NCF/XPBD ejects the cubes and NCF/Newton leaves an unnaturally elevated stack, whereas SNCP/XPBD retains the stack with minor residual drift and only SNCP/Newton successfully maintains a stable stack. The difference stems from NCF's nonzero exponential tail continually pushing separated pairs apart. SNCP resolves this issue via a smooth cutoff that renders the constraint and its gradient strictly zero outside the active region, thereby eliminating artificial long-range corrections.

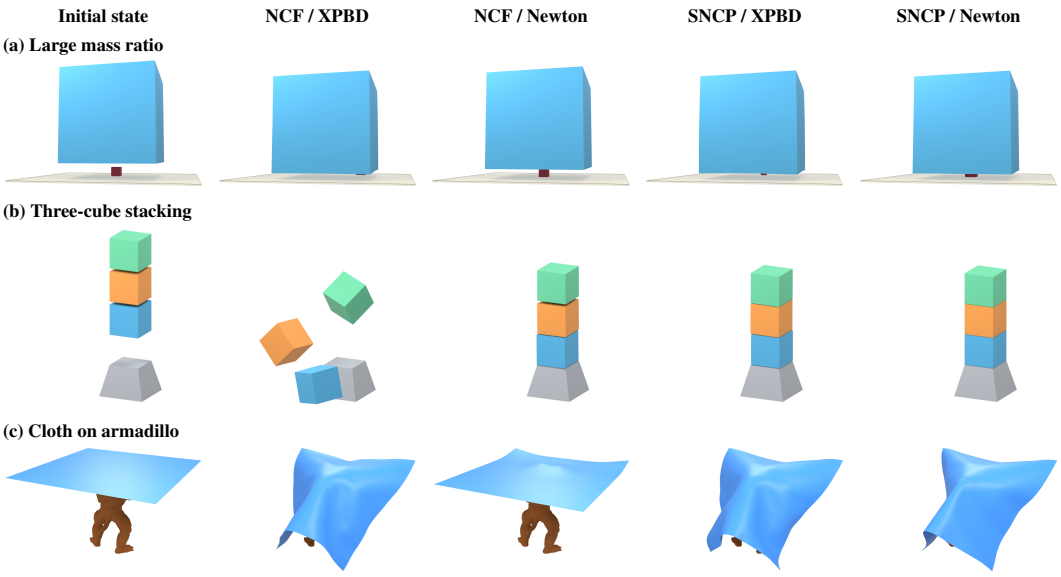

Figure 2: **Comparison across contact models and solvers.** (a) A heavy cube falls onto a light cube with a 2000:1 mass ratio; both cubes have a Young's modulus of 3 MPa. (b) Three cubes fall onto a fixed truncated-pyramid base; NCF/XPBD ejects the cubes. (c) A pinned cloth drapes over an armadillo. The first column shows the initial state; subsequent columns compare the four contact/solver combinations. NCF/Newton first fails to converge at frames 34, 45, and 30 for (a)–(c), respectively. The simulation stuck at early collisions and difficult to step forward. Table 3 in Appendix G reports the solver settings, runtimes, and convergence counts.

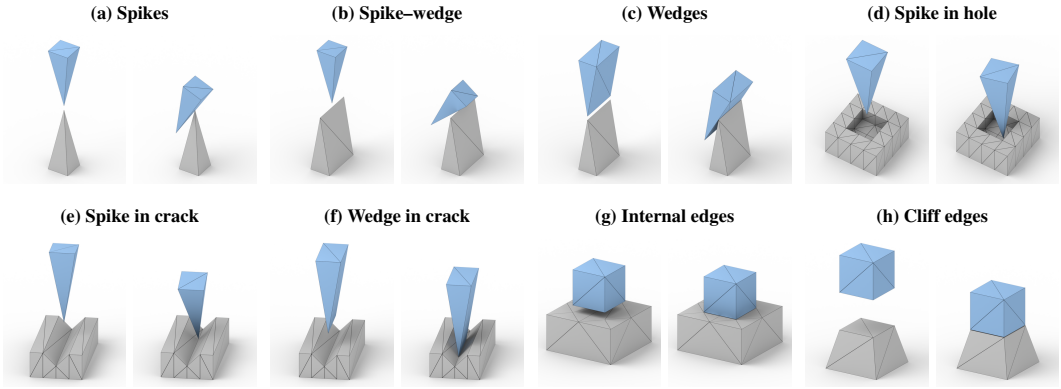

Figure 3: **Erleben mesh-contact tests.** Each pair shows the initial state (left) and the labeled simulation frame at the time of contact (right); blue denotes the moving body and gray the fixed geometry. The eight cases exercise sharp tips, narrow gaps, and internal or exposed edges. Our method successfully handles these challenging cases.

Cloth on armadillo. Figure 2(c) compares all four configurations on a larger scenario where a square cloth is draped on a fixed armadillo model. Both XPBD methods uses only one position iteration per substep, resulting in visible cloth jitter. SNCP/Newton handles the contact more stably and efficiently, while NCF/Newton remains near early-contact configurations after convergence failures.

Efficiency analysis. SNCP demonstrates a clear efficiency advantage as shown in Table 3. Under XPBD, our method is consistently faster in resolving sustained contact, achieving a nearly $5\times$ speedup on the contact-rich cloth-on-armadillo scene, with the sole exception in cube stacking being an artifact of NCF wrongly ejecting the cubes out of contact. The runtime gap is even more dramatic under the Newton solver (up to an order of magnitude in contact-rich cases), which is caused by NCF repeatedly failing to converge and hitting the maximum iteration count, whereas SNCP converges natively with zero failed frames.

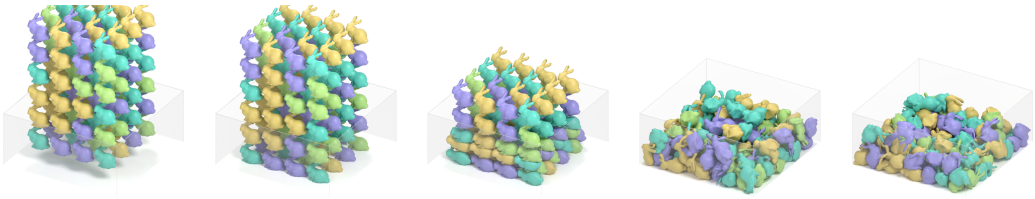

Figure 4: **Falling bunnies.** A $4 \times 6 \times 4$ array of 96 deformable bunnies falls into a fixed square well, with $53k$ tetrahedra and $60k$ degrees of freedom in total. From left to right, the bodies collide, deform, and settle into a dense pile.

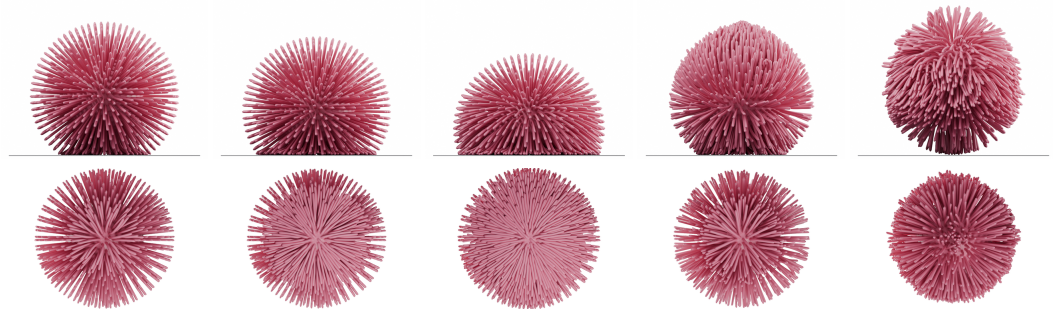

Figure 5: **Squishy-ball impact and deformation.** A tentacled elastic ball drops onto fixed ground. Front views (top) and views from underneath (bottom) show the same five frames, progressing from left to right through impact and rebound. The lower views reveal how the tendrils spread and bend beneath the ball. Our method handles the resulting volumetric deformation together with dense self-contact.

5.2 Simulation results

We demonstrate SNCP with a Newton solver in challenging scenes covering affine-body contacts, thin shells, volumetric deformation, and many-body interactions. Appendix G reports the scene settings and runtimes. Additional demonstrations appear in Appendix D.

Erleben test cases. Erleben (2018) proposed simple geometric unit tests designed to expose failures in mesh-based collision handling. We evaluate SNCP on eight configurations involving spikes, wedges, holes, cracks, internal edges, and cliff edges. As shown in Fig. 3, SNCP handles these challenging contacts successfully.

Falling bunnies. To evaluate SNCP with many interacting deformable bodies, we simulate 96 elastic bunnies falling under gravity into a fixed square well as shown in Fig. 4. Initially arranged in a $4 \times 6 \times 4$ grid, the bunnies comprise 60,192 degrees of freedom and 53,760 tetrahedra, with up to 14,624 active contacts recorded per frame. SNCP handles the evolving contacts as the bunnies collide, deform, and settle into a dense pile.

Squishy ball. To test SNCP on a detailed deformable model with many simultaneous contacts, we drop a tentacled elastic ball onto fixed ground (Fig. 5). The model contains $163k$ vertices and $444k$ tetrahedra. Its flexible tendrils bend and spread during impact, with up to $34k$ active contacts recorded per frame. This example demonstrates SNCP's handling of volumetric deformation coupled with dense self-contact.

5.3 Neural integrator versus quadrature

We benchmark the proposed SNCP model against the NCF-style quadrature (Zesch et al., 2023) adapted to the same Wendland contact integral. The NCF-style quadrature samples a uniform barycentric lattice on each triangle and averages the kernel over all resulting point pairs. We adopt a discretization resolution of 40, yielding 820 quadrature points per triangle, and evaluate both ap-

proaches on a test set of 65,536 triangle pairs using a GPU batch size of 8,096. Detailed results are given in Appendix E.

As shown in Table 1, SNCP decisively outperforms NCF-style quadrature across all derivative orders. It cuts MRE by roughly two-thirds for energy (16.78% vs. 59.37%) and gradients (18.67% vs. 52.11%), and by nearly a factor of three for Hessians (29.49% vs. 86.69%), while maintaining high directional alignment (≥ 0.953 mean cosine). Computationally, SNCP evaluates E and $E + g$ calls $2.35\times$ and $1.21\times$ faster than quadrature, respectively. Although complete Hessian evaluation ($E + g + H$) incurs higher runtime (10.48 s vs. 4.69 s), SNCP can provide a higher quality hessian for simulation.

5.4 Ablation study: factorization and Sobolev training

We conduct a 2×2 ablation over two design axes: **conservative support factorization** and **Sobolev supervision**. This yields four configurations: unfactored value-only (**UV**), unfactored Sobolev (**US**), factored value-only (**FV**), and our proposed factored Sobolev (**FS**). All variants share an 8-layer MLP architecture (width 512, 200 epochs) and are evaluated across three seeds on the 10M validation set.

As shown in Appendix F, Table 2, models without gradient supervision completely fail to recover meaningful force fields: 50.0% of their predicted gradients point in the wrong direction (negative cosine), and derivative relative p_{95} errors explode beyond 10^5. Sobolev training eliminates this failure, slashing inverted gradients to $\approx 7\%$ and reducing derivative p_{95} errors by over five orders of magnitude down to ≈ 1.0. Combining Sobolev supervision with architectural factorization achieves the best performance and tightest variance across all metrics.

6 Conclusion

We present Second-Order Neural Contact Potentials (SNCP), a learned contact formulation that supports implicit, second-order physical simulation without geometric sampling. We define the contact energy via a compactly supported C^2 Wendland surface integral, which is further factorized into an analytic geometric factor and a strictly positive learned coefficient. Analytical differentiation of the complete reconstructed potential provides consistent world-space gradients for XPBD and Hessians for Newton-based solvers. Once trained, the model transfers directly across affine bodies, deformable solids, and thin-shell scenes without retraining.

Empirically, SNCP significantly outperforms the existing formulation NCF in both numerical robustness and runtime efficiency, eliminating artificial separation artifacts and reliably recovering true resting equilibria. Our ablations confirm that combining an analytic geometric factor with Sobolev supervision is useful for learning conservative, derivative-accurate potentials. These results demonstrate the power of embedding structural geometric priors into neural representations to enable stable second-order physics simulation.

Limitations and future work. Despite its advantages, SNCP has several limitations that suggest promising directions for future research. Currently, our potential is restricted to frictionless contact between triangles. Extending the formulation to incorporate tangential friction and curved surface primitives would broaden its applicability. In addition, the coordinate-based network does not enforce rotation invariance: rotation augmentation encourages this symmetry but does not guarantee it. A key direction for future work is to develop rotation-invariant representations that retain the smoothness required for second-order simulation, so that contact energies remain invariant and contact forces transform consistently under rigid rotations.

AI Use Statement

In this work, we used generative AI tools for the following tasks: to formulate mathematical claims, to implement methods, to design or provide feedback on research methodology or experiments and to assist with translation. **We have not used generative AI tools for these tasks:** to generate synthetic data sets, to help develop theoretical models or conceptual frameworks, to provide critical ingredients for proving mathematical claims, to assist in the writing of proofs, or to

interpret results. **The following subtasks are not applicable to this work:** to propose or refine hypotheses, to clean and reformat dataset, and to support qualitative and thematic data analysis. **Additionally, we used generative AI tools for the following purposes:** to create or modify scientific figures or images and to identify relevant literature. **We have reviewed all AI-assisted work.** The AI-generated images serve only as reference material. The versions included in the article are all manually designed. LLM-generated code was verified and tested for correctness. The related works found by AI were also verified for authenticity. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

To support reproducibility, Sec. 4 describes the SNCP formulation, and Appendix A provides the mathematical derivations and assumptions underlying its support guarantee, regularity, and analytical derivatives. Appendix C documents the synthetic data sampling procedure, dataset splits, quadrature-based label generation, target normalization, network architecture, loss function, and optimization settings. Appendix B describes potential evaluation and integration into the Newton and XPBD solvers. Sec. 5 reports the experimental hardware and comparison setup; Appendices E and F specify the potential fidelity and timing protocols and report ablation results across independent training seeds. Appendix G lists simulation scene parameters, mesh sizes, material properties, time steps, and runtimes. We will publicly release the dataset, source code, and trained model checkpoints upon acceptance of the paper.

References

Francisco Armero and Eva Petőcz. Formulation and analysis of conserving algorithms for frictionless dynamic contact/impact problems. *Computer Methods in Applied Mechanics and Engineering*, 158(3–4):269–300, 1998. doi: 10.1016/S0045-7825(97)00256-9.

Jinshuai Bai, Zhongya Lin, Yizheng Wang, Jiancong Wen, Yinghua Liu, Timon Rabczuk, YuanTong Gu, and Xi-Qiao Feng. Energy-based physics-informed neural network for frictionless contact problems under large deformation. *Computer Methods in Applied Mechanics and Engineering*, 437:117787, 2025. doi: 10.1016/j.cma.2025.117787.

Xinhao Cai, Eulalie Coevoet, Alec Jacobson, and Paul Kry. Active learning neural c-space signed distance fields for reduced deformable self-collision. In *Graphics Interface 2022*, pp. 92–100, 2022. doi: 10.20380/GI2022.11.

Anka He Chen, Ziheng Liu, Yin Yang, and Cem Yuksel. Vertex block descent. *ACM Transactions on Graphics*, 43(4), 2024. doi: 10.1145/3658179.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017. doi: 10.1126/sciadv.1603015.

Wojciech M. Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kenny Erleben. Methodology for assessing mesh-based contact point methods. *ACM Transactions on Graphics*, 37(3), 2018. doi: 10.1145/3096239.

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*, 40(4), 2021.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

Kemeng Huang, Floyd M. Chitalu, Huancheng Lin, and Taku Komura. GIPC: Fast and stable Gauss–Newton optimization of IPC barrier energy. *ACM Transactions on Graphics*, 43(2), 2024. doi: 10.1145/3643028.

Kemeng Huang, Xinyu Lu, Huancheng Lin, Taku Komura, and Minchen Li. StiffGIPC: Advancing GPU IPC for stiff affine-deformable simulation. *ACM Transactions on Graphics*, 44(3), 2025a. doi: 10.1145/3735126.

Zizhou Huang, Maxwell Paik, Zachary Ferguson, Daniele Panozzo, and Denis Zorin. Geometric contact potential. *ACM Transactions on Graphics*, 44(4), 2025b. doi: 10.1145/3731142.

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), July 2022. doi: 10.1145/3528223.3530064.

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 Transactions on Graphics*, 39(4), 2020. doi: 10.1145/3386569.3392425.

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics*, 40(4), 2021.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. Convergent incremental potential contact. *arXiv preprint arXiv:2307.15908*, 2023.

Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in Adam. *CoRR*, abs/1711.05101, 2017.

M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and T.Y. Kim. Primal/dual descent methods for dynamics. *Computer Graphics Forum*, 39(8):89–100, 2020. doi: 10.1111/cgf.14104.

Jiří Minarčík, Michael Liu, Keenan Crane, and Minchen Li. Untangling surfaces via shape and mesh repulsion. *ACM Trans. Graph.*, 45(4), July 2026. doi: 10.1145/3811382.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2007.01.005.

Jeong Joon Park, Peter R. Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Cristian Romero, Dan Casas, Maurizio M. Chiaramonte, and Miguel A. Otaduy. Contact-centric deformation learning. *ACM Transactions on Graphics*, 41(4), 2022. doi: 10.1145/3528223.3530182.

Cristian Romero, Dan Casas, Maurizio M. Chiaramonte, and Miguel A. Otaduy. Learning contact deformations with general collider descriptors. In *ACM SIGGRAPH Asia 2023 Conference Proceedings*, 2023. doi: 10.1145/3610548.3618229.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473, 2020.

N. Sukumar and Ankit Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 389:114333, 2022. doi: 10.1016/j.cma.2021.114333.

Oliver T. Unke and Markus Meuwly. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of Chemical Theory and Computation*, 15(6):3678–3693, 2019. doi: 10.1021/acs.jctc.9b00181.

Xinyu Wang, Weipeng Xu, and Tianju Xue. Solving contact problems using Fiber Monte Carlo. *Computational Mechanics*, 2026. doi: 10.1007/s00466-026-02822-6.

Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995. doi: 10.1007/BF02123482.

Ryan S. Zesch, Vismay Modi, Shinjiro Sueda, and David I. W. Levin. Neural collision fields for triangle primitives. In *ACM SIGGRAPH Asia 2023 Conference Proceedings*, 2023. doi: 10.1145/3610548.3618225.

A Mathematical Derivations

This section derives the analytical derivatives used to evaluate SNCP at runtime. The support radius r is treated as fixed when differentiating with respect to the 18 world-space vertex coordinates.

A.1 Kernel Regularity and Surface Scaling

For $\tau = \rho/r \in [0, 1)$, the Wendland radial kernel $\psi_r(\rho) = 1 - 6\tau^2 + 8\tau^3 - 3\tau^4$ has derivatives:

$$\psi_r'(\rho) = -\frac{12}{r}\tau(1-\tau)^2, \qquad \psi_r''(\rho) = -\frac{12}{r^2}(1-\tau)(1-3\tau). \tag{19}$$

At $\rho = r$, the kernel function and its first two derivatives vanish smoothly, matching the zero exterior branch. At $\rho = 0$, $\psi_r'(0) = 0$ and $\lim_{\rho\to 0^+} \psi_r'(\rho)/\rho = \lim_{\rho\to 0^+} \psi_r''(\rho) = -12/r^2$. Thus, the radial profile $\psi_r(\|v\|)$ is globally C^2 across the origin with gradient zero and Hessian $-12I/r^2$.

Under a joint uniform scaling transformation $p = c + s\bar{p}$, $q = c + s\bar{q}$, and $\bar{r} = r/s$, the relative distance ratio is scale-invariant. Consequently,

$$\psi_r(\|p - q\|) = \psi_{\bar{r}}(\|\bar{p} - \bar{q}\|), \qquad \mathrm{d}A_p\,\mathrm{d}A_q = s^4\,\mathrm{d}\bar{A}_p\,\mathrm{d}\bar{A}_q, \tag{20}$$

which establishes the surface scale-lifting relation. Derivatives with respect to scale change are included when s depends on vertex configurations.

A.2 Barrier Geometry and Implicit Differentiation

Let $\bar{A} = [\bar{x}_0, \bar{x}_1, \bar{x}_2]$ and $\bar{B} = [\bar{x}_3, \bar{x}_4, \bar{x}_5]$ denote the normalized vertices of two interacting triangles. We solve the strictly convex interior barrier optimization problem:

$$(\alpha^*, \beta^*) = \underset{\alpha,\beta\in\Delta_3^\circ}{\arg\min} \|\bar{A}\alpha - \bar{B}\beta\|_2^2 - \kappa_B \sum_{i=0}^{2} (\log\alpha_i + \log\beta_i), \qquad \kappa_B = 10^{-6}, \tag{21}$$

where α and β are barycentric coordinates in $\Delta_3^\circ = \{\gamma \in \mathbb{R}^3_{>0} : \sum_i \gamma_i = 1\}$. The conservative proximity score q_B is obtained by subtracting the theoretical slack error bound from the optimized squared separation:

$$q_B = \|\bar{A}\alpha^* - \bar{B}\beta^*\|_2^2 - 6\kappa_B. \tag{22}$$

We use $w \in \mathbb{R}^{18}$ to denote the concatenated normalized coordinates, and use $z = (\alpha_0, \alpha_1, \beta_0, \beta_1)^\top$ to express the four degrees of freedom for the barycentric coordinates $\xi(z) = (z_0, z_1, 1 - z_0 - z_1, z_2, z_3, 1 - z_2 - z_3)^\top$ with Jacobian $R = \partial\xi/\partial z \in \mathbb{R}^{6\times 4}$. Define $D(w) = [\bar{x}_0 - \bar{x}_2, \bar{x}_1 - \bar{x}_2, \bar{x}_5 - \bar{x}_3, \bar{x}_5 - \bar{x}_4]$ and $t(w) = \bar{x}_2 - \bar{x}_5$. The objective functions and optimality conditions are:

$$Q(z, w) = \|D(w)z + t(w)\|_2^2, \tag{23}$$

$$B(z, w) = Q(z, w) - \kappa_B \sum_{i=0}^{5} \log\xi_i(z), \tag{24}$$

$$F(z, w) = \nabla_z B = 2D^\top(Dz + t) - \kappa_B R^\top \xi^{-1} = 0, \tag{25}$$

$$K(z, w) = F_z = 2D^\top D + \kappa_B R^\top \operatorname{diag}(\xi_i^{-2}) R. \tag{26}$$

Because $\kappa_B > 0$ and R has full column rank, K is positive definite everywhere. By the implicit function theorem, $z_B(w)$ and $q_B(w)$ are smooth functions of the normalized geometry.

Support Bound. Let z_* minimize Q over the closed simplex $\xi(z) \geq 0$, and let $q^* = Q(z_*, w)$. By convexity of Q and stationarity of B, we have:

$$0 \leq Q(z_B, w) - q^* \leq -\nabla_z Q(z_B, w)^\top (z_* - z_B) = \kappa_B \sum_{i=0}^{5} \left(1 - \frac{\xi_i(z_*)}{\xi_i(z_B)}\right) \leq 6\kappa_B, \tag{27}$$

establishing the conservative error bound:

$$q^* - 6\kappa_B \leq q_B \leq q^*. \tag{28}$$

Thus the active set $\mathcal{A}_B = \{q_B < \bar{r}^2\}$ strictly encloses the exact support domain $\{q^* < \bar{r}^2\}$.

Optimizer and Score Derivatives. First- and second-order derivatives of $z_B(w)$ with respect to coordinate components w_a, w_b are obtained by differentiating $F(z_B(w), w) = 0$:

$$Kz_a = -F_a, \tag{29}$$

$$Kz_{ab} = -\left(F_{ab} + F_{az}z_b + F_{bz}z_a + F_{zz}[z_a, z_b]\right), \tag{30}$$

where the subscript a (or z) denotes a partial derivative with respect to w_a (or z). Score derivatives follow via the full chain rule:

$$(q_B)_a = Q_a + Q_z^\top z_a, \tag{31}$$

$$(q_B)_{ab} = Q_{ab} + Q_{az}z_b + Q_{bz}z_a + z_a^\top Q_{zz}z_b + Q_z^\top z_{ab}. \tag{32}$$

A.3 Normalized Energy and World-Space Chain Rule

For normalized inputs $u = (\bar{x}, \bar{r})$, the smooth gap function $g_B = (\bar{r}^2 - q_B)/(2\bar{r})$ has active-branch derivatives:

$$g_{\bar{x}} = -\frac{q_{B,\bar{x}}}{2\bar{r}}, \qquad g_{\bar{r}} = \frac{1}{2} + \frac{q_B}{2\bar{r}^2}, \tag{33}$$

$$g_{\bar{x}\bar{x}} = -\frac{q_{B,\bar{x}\bar{x}}}{2\bar{r}}, \qquad g_{\bar{x}\bar{r}} = \frac{q_{B,\bar{x}}}{2\bar{r}^2}, \qquad g_{\bar{r}\bar{r}} = -\frac{q_B}{\bar{r}^3}. \tag{34}$$

Let $f_\theta(u)$ denote the scalar output of an L-layer SIREN network with activations $v^{(\ell)} = \sin(\omega a^{(\ell)})$. Write $a = a_\theta(u) = 10^{\mu + \sigma f_\theta(u)}$ and $\eta = \sigma \ln 10$. The coefficient derivatives are

$$\nabla_u a = \eta a \nabla_u f_\theta, \qquad H_a = \eta a H_{f_\theta} + \eta^2 a \nabla_u f_\theta \nabla_u f_\theta^\top. \tag{35}$$

The normalized energy $\hat{\bar{E}} = g_B^4 a$ therefore has derivatives

$$\nabla_u \hat{\bar{E}} = g_B^4 \nabla_u a + 4 g_B^3 a p, \tag{36}$$

$$H_{\hat{\bar{E}}} = g_B^4 H_a + 4 g_B^3 \left(p(\nabla_u a)^\top + (\nabla_u a) p^\top\right) + 12 g_B^2 a p p^\top + 4 g_B^3 a H_g, \tag{37}$$

where $p = \nabla_u g_B$ and $H_g = \nabla_u^2 g_B$ on the active branch.

Regularity of the analytical potential. For $\bar{r} > 0$, let $h(u) = (\bar{r}^2 - q_B)/(2\bar{r})$, so that $g_B = [h]_+$. The exact stationary barrier solution makes h smooth, and a_θ is smooth and strictly positive for finite network inputs. Since $t \mapsto [t]_+^4$ is C^3, their product $\hat{\bar{E}} = [h]_+^4 a_\theta$ is C^3, including across $h = 0$. Its value, gradient, and Hessian are zero on the exterior branch and at the outer cutoff. There is no interior coefficient-clamp boundary in this analytical formulation. Smooth centroid/RMS normalization and the s^4 lift preserve this regularity for $s > 0$ and fixed world-space $r > 0$.

Transforming to world-space coordinates $X \in \mathbb{R}^{18}$ via Jacobian $J = \partial u / \partial X$ and scale vector $b = \nabla_X s$ yields the world-space energy derivatives:

$$\nabla_X (s^4 \bar{E}) = s^4 J^\top \nabla_u \bar{E} + 4 s^3 \bar{E} b, \tag{38}$$

$$H_X(s^4\bar{E}) = s^4 \left(J^\top H_{\bar{E}} J + \sum_a (\partial_{u_a} \bar{E}) H_{u_a} \right) + 4 s^3 \left[(J^\top \nabla_u \bar{E}) b^\top + b (J^\top \nabla_u \bar{E})^\top + \bar{E} H_s \right] + 12 s^2 \bar{E} b b^\top. \tag{39}$$

A.4 World-Space Gradient Consistency

The inconsistency in NCF. NCF (Zesch et al., 2023) rescales both its normalized potential and its normalized-coordinate gradient by the same factor s_N. However, the gradient of the world-space potential $s_N E_{\mathrm{NCF}}$ is not $s_N E'_{\mathrm{NCF}}$: differentiating with respect to world-space vertices requires the chain rule through normalization and the derivative of s_N itself.

To make this explicit, let $X = (x_0, \ldots, x_5) \in \mathbb{R}^{18}$ collect the world-space vertices. NCF centers and normalizes them as

$$P = \left(I_6 - \tfrac{1}{6}\mathbf{1}\mathbf{1}^\top\right) \otimes I_3, \qquad y = PX, \qquad s_N = \max_{0 \le i < 6} \|y_i\|_2, \qquad u_N = y / s_N, \tag{40}$$

where $y_i \in \mathbb{R}^3$ is the ith vertex block. Write $E_{\mathrm{NCF}}(u_N)$ for the normalized neural potential and $E'_{\mathrm{NCF}} = \nabla_{u_N} E_{\mathrm{NCF}}$ for its input gradient. Suppressing a fixed multiplicative gain, the native evaluator uses

$$E_N^{\mathrm{world}}(X) = s_N E_{\mathrm{NCF}}(u_N), \qquad \widetilde{G}_N(X) = s_N E'_{\mathrm{NCF}}(u_N). \tag{41}$$

The scalar is used as a contact constraint in NCF's position-based solver, with $\widetilde{G}_N$ supplied as its world-space gradient.

On a differentiable branch with $s_N > 0$, define $b_N = \nabla_X s_N$. The normalization Jacobian and the correct world-space gradient are

$$J_N = \frac{\partial u_N}{\partial X} = \frac{P - u_N b_N^\top}{s_N}, \tag{42}$$

$$\begin{aligned} \nabla_X E_N^{\mathrm{world}} &= s_N J_N^\top E'_{\mathrm{NCF}} + E_{\mathrm{NCF}} b_N \\ &= P E'_{\mathrm{NCF}} + \left(E_{\mathrm{NCF}} - u_N^\top E'_{\mathrm{NCF}}\right) b_N. \end{aligned} \tag{43}$$

Thus the input-normalization Jacobian cancels the explicit s_N in the first term and contributes a scale-dependent correction; differentiating the output scale contributes $E_{\mathrm{NCF}} b_N$. Simply multiplying E'_{NCF} by s_N omits these dependencies and is generally incorrect, even away from ReLU and max-radius branch boundaries.

Uniform scaling exposes the mismatch directly. For $t > 0$, u_N is unchanged and $E_N^{\mathrm{world}}(tX) = tE_N^{\mathrm{world}}(X)$, so

$$(\nabla_X E_N^{\mathrm{world}})(tX) = (\nabla_X E_N^{\mathrm{world}})(X), \qquad \widetilde{G}_N(tX) = t\,\widetilde{G}_N(X). \tag{44}$$

The true gradient is invariant to this scaling, whereas the native NCF gradient scales linearly. This inconsistency concerns native NCF/XPBD; our NCF/Newton adaptation uses full world-space derivatives.

SNCP differentiates the world-space potential. SNCP accounts for world-space scaling in both its potential and its derivatives. Its potential is $\widehat{E}_{AB}(X; r) = s^4 \widehat{\bar{E}}_\theta(u)$, where $s = \sqrt{y^\top y/6}$ and $u = (\bar{x}, \bar{r}) = (y/s, r/s)$. With world-space r fixed, the complete normalization Jacobian is

$$b = \nabla_X s = \frac{y}{6s}, \qquad J = \frac{\partial u}{\partial X} = \begin{bmatrix} (P - \bar{x} b^\top)/s \\ -\bar{r} b^\top / s \end{bmatrix}. \tag{45}$$

Applying the chain rule gives

$$\nabla_X \widehat{E}_{AB} = s^4 J^\top \nabla_u \widehat{\bar{E}}_\theta + 4 s^3 \widehat{\bar{E}}_\theta b. \tag{46}$$

This includes the effects of centering, input scaling, the normalized radius, and the s^4 output scale. SNCP therefore supplies the correct gradient of its learned world-space potential. Its XPBD constraint obeys the same rule: writing $C_S = s^2 F_S(u)$ gives

$$\nabla_X C_S = s^2 J^\top \nabla_u F_S + 2 s F_S b. \tag{47}$$

B Implementation Details

We compare NCF and SNCP in two solver frameworks. The XPBD experiments use the reproduced NCF position-based pipeline (Zesch et al., 2023); the Newton experiments use our C++/CUDA integration in libuipc (Huang et al., 2024; 2025a).

B.1 Contact Field Evaluation

SNCP. The evaluator constructs the 19-dimensional input using the centroid/RMS normalization in Eq. (7). Its analytical potential is

$$\widehat{E}_{AB} = s^4 g_B^4 a_\theta(u), \qquad a_\theta(u) = 10^{\mu + \sigma f_\theta(u)}. \tag{48}$$

Barrier solve. We solve Eq. (21) with $\kappa_B = 10^{-6}$ using damped Newton iterations, initialized at uniform barycentric coordinates $\alpha_i = \beta_i = 1/3$. Each iteration solves a reduced 4×4 system by $LDL^\top$ factorization, with diagonal damping if needed. A fraction-to-boundary factor of 0.995 preserves positive barycentric coordinates; Armijo backtracking uses coefficient 10^{-4} and at most 40 step halvings. We allow 80 iterations and stop when $-\frac{1}{2}F^\top \delta z \le 10^{-15}$, where $F = \nabla_z B$ and δz is the Newton direction.

NCF. We reproduced NCF pipeline using a triangle-pair network with 18 coordinate inputs, four width-512 ReLU hidden layers, and a sigmoid output. Let $F_N(u_N)$ denote its decoded normalized scalar. As defined in Appendix A.4, $u_N = PX/s_N$ uses joint-centroid centering and the maximum vertex radius s_N. The same NCF weights are used in both solvers.

B.2 XPBD Integration

Update schedule. Each frame of duration Δt contains S substeps of size $h = \Delta t/S$. Each position iteration applies the position constraint projection. The neural contact energies and gradients are evaluated on the GPU with fixed-size batch, while other parts of the simulation are parallelized by OpenMP on the CPU. The velocity-constraint iterations run after all substeps.

NCF/XPBD. Following NCF, we use the scalar field $C_N = E_{\mathrm{N}}^{\mathrm{world}}$ as the constraint and use $\widetilde{G}_N$ in Eq. (41) as the gradient proxy. So for inverse vertex masses w_i and contact-update time step h, the native projection applies

$$\Delta\lambda = -\frac{C_N}{\sum_{j=0}^{5} w_j \|\widetilde{G}_N\|_2^2 + \alpha/h^2}, \qquad \Delta x_i = w_i \Delta\lambda \widetilde{G}_N. \tag{49}$$

SNCP/XPBD. For contact stiffness κ, the analytical constraint is

$$C_S = \sqrt{2\alpha \widehat{E}_{AB}} = s^2 g_B^2 \sqrt{2\alpha a_\theta(u)}, \qquad \alpha = \frac{1}{\kappa}. \tag{50}$$

For inverse vertex masses w_i and contact-update time step h, the native projection applies

$$\Delta\lambda = -\frac{C_S}{\sum_{j=0}^{5} w_j \|\nabla_{x_j} C_S\|_2^2 + \alpha/h^2}, \qquad \Delta x_i = w_i \Delta\lambda \nabla_{x_i} C_S. \tag{51}$$

B.3 Newton Integration

Contact energies and derivatives. For each candidate pair, the implicit-Euler objective receives $\Delta t^2 U(X)$. SNCP uses $U_S = \kappa_S \widetilde{E}_{AB}$ and its analytical world-space gradient and Hessian.

Our NCF adaptation defines

$$U_N = \frac{\kappa_N}{2} C_N^2, \tag{52}$$

where κ_N is the contact stiffness for NCF/Newton. We differentiate this energy on each ReLU/max-radius branch:

$$\nabla_X U_N = \kappa_N C_N \nabla_X C_N, \tag{53}$$

$$\nabla_X^2 U_N = \kappa_N \left[(\nabla_X C_N)(\nabla_X C_N)^\top + C_N \nabla_X^2 C_N \right]. \tag{54}$$

These derivatives include the sigmoid, centering, input normalization, and output scale.

Assembly and convergence. For both fields, we evaluate an 18-coordinate gradient and a full 18×18 pair Hessian, and clamp negative eigenvalues to zero before assembly. This positive-semidefinite projection stabilizes the solve. Pair contributions are mapped to global degrees of freedom, and the Newton system is solved by preconditioned conjugate gradients with block Jacobi preconditioning and an energy-based line search. NCF/Newton uses caps of 100 Newton iterations and 32 line-search trials. At the cap, the comparison continues from the last finite iterate.

B.4 Candidate Generation and Numerical Settings

Both frameworks use a dynamic triangle BVH. For SNCP, a triangle with RMS radius σ_f and maximum radius ρ_f about its centroid has its axis-aligned bounding box expanded by

$$e_f = \max\left\{r, \frac{r/2 + c_B(\sigma_f/\sqrt{2} + \rho_f/2)}{1 - c_B/2}\right\}, \qquad c_B = \sqrt{6\kappa_B}, \tag{55}$$

with additional shell thickness where applicable. After BVH pruning, SNCP activation requires $q_B < \bar{r}^2$. NCF uses geometric candidate tolerances rather than this barrier-support test; pruning its candidate set does not make the neural field compactly supported.

The XPBD comparisons evaluate constraints and gradients in batches of up to 32,768 pairs. The Newton evaluator processes 32,768-pair chunks with Hessian sub-batches of 4,096. These evaluator settings are separate from the substep and iteration budgets in Table 3.

C Training Data and Optimization Details

C.1 Sampling Protocol

The dataset consists of 100 million triangle-pair samples partitioned into 85M training, 10M validation, and 5M test split. Vertex coordinates are initialized from independent standard normal distributions, pruned of degenerate triangles, and scaled log-uniformly by $s \sim \text{LogUniform}[0.1, 1.0]$. Triangle centroids are aligned at the origin.

Starting from these centroid-aligned pairs, we postprocess each sample to construct two equally probable configurations: interpenetration and near-contact. For interpenetrating configurations, the secondary triangle is translated by ξv along the penetration vector v, where $\xi \sim \mathcal{U}[0, 0.8]$. For near-contact configurations, we separate the pair just outside the contact boundary by applying an offset of $v + \delta\frac{v}{\|v\|_2}$, where $\delta \sim \mathcal{U}[0, 0.3r]$ introduces a controlled separation gap governed by the interaction radius $r = 0.01$.

C.2 Logarithmic Targets and Loss Formulation

Numerical energy labels $\bar{E}_{\text{num}}$ and derivative estimates $d_{\text{num}} \in \mathbb{R}^{19}$ are generated using adaptive order-15 product quadrature with relative tolerance 10^{-6}. To handle target scale variation, labels are normalized by:

$$z^* = \log_{10}\frac{\bar{E}_{\text{num}}}{g_B^4} = \log_{10}\bar{E}_{\text{num}} - 4\log_{10} g_B, \qquad y^* = \frac{z^* - \mu}{\sigma}, \tag{56}$$

where the training-split normalization constants μ and $\sigma > 0$ are held fixed when differentiating. Applying the chain rule, with d_{num} supplying the estimate of $\nabla_u \bar{E}_{\text{num}}$, gives the derivative target

$$h^* = \frac{1}{\sigma \ln 10}\left(\frac{d_{\text{num}}}{\bar{E}_{\text{num}}} - 4\frac{\nabla_u g_B}{g_B}\right) = \frac{d_{\text{num}} - 4\bar{E}_{\text{num}}\nabla_u g_B/g_B}{\sigma \ln(10)\bar{E}_{\text{num}}}. \tag{57}$$

The combined loss is $\mathcal{L} = \mathcal{L}_{\text{Huber}} + 0.5\mathcal{L}_{\text{Sob}}$, where the value term compares $f_\theta(u_i)$ with y_i^* using unit-threshold Huber loss. Writing $\hat{h}_i = \nabla_u f_\theta(u_i)$, the derivative term is

$$\mathcal{L}_{\text{Sob}} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\left[1 - \frac{\hat{h}_i^\top h_i^*}{\|\hat{h}_i\|_2\|h_i^*\|_2} + \text{Huber}_1(\ln\|\hat{h}_i\|_2, \ln\|h_i^*\|_2)\right], \tag{58}$$

with $\mathcal{V} = \{i : u_i \in \mathcal{A}_+,\ \|h_i^*\|_2 > 10^{-20}\}$ for this unshifted formulation.

C.3 Architecture and Training Parameters

We deploy an 8-layer SIREN architecture (Sitzmann et al., 2020) with 512 hidden units per layer ($\omega = 1.0$, 1,849,345 total parameters). Optimization uses AdamW (Loshchilov & Hutter, 2017) with weight decay 10^{-5}, batch size 4096, and gradient clipping norm 1.0. The learning rate follows a cosine decay schedule from 10^{-3} to 10^{-6} over 200 epochs. Training includes random 3D rotation, vertex permutation, and triangle order swap augmentations.

D Additional Simulation Results

We present two additional SNCP/Newton demonstrations of sustained self-contact and many-body interactions. Table 4 reports their scene parameters, contact counts, and runtimes.

Mat twist. To test large deformation and dense self-contact, we twist a thin rectangular mat (Fig. 6). One short edge is fixed, while the opposite edge softly tracks a $144°$/s rotation about the mat's longitudinal axis. As twisting proceeds, the mat folds into a tightly wound central region. SNCP handles the resulting sustained self-contact throughout the sequence.

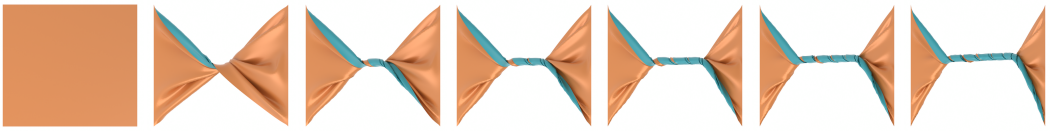

Figure 6: **Extreme twisting of a mat.** One short edge is fixed while the opposite edge softly tracks a prescribed rotation about the mat's longitudinal axis. Left to right, snapshots show zero to six full turns of the prescribed rotation at one-turn intervals. The tightly wound central region develops sustained self-contact, testing our method under large deformation and closely packed folds.

Struck cubes. A heavy ball suspended from a chain of 13 interlocking rings strikes a stack of 560 blocks, scattering them across the ground (Fig. 7). The scene contains 575 objects in total, including the fixed ground. SNCP handles contact between the chain links together with the many collisions generated by the collapsing stack.



Figure 7: **Struck cubes.** A ball suspended from 13 interlocking rings strikes a stack of 560 blocks. Left to right, figures show the initial state, impact, and subsequent collapse and scattering. The sequence tests our method on persistent contacts between chain links alongside the changing contacts among the ball, blocks, and ground.

E Potential Fidelity Benchmarks

We compare SNCP with NCF-style Wendland quadrature ($n = 40$, 820 nodes per triangle) on the unchanged panel of 65,536 test configurations with $E \geq \tau = 2.97 \times 10^{-7}$. Energy and gradient errors use stored adaptive-quadrature ground truth in dataset; Hessian errors use a separate analytical reference computed with the same adaptive-quadrature algorithm. The gradient g and Hessian H include all 18 normalized vertex coordinates and the support radius.

Table 1 reports mean relative error (MRE), using absolute scalar error for energy, Euclidean norm error for g, and Frobenius norm error for H, each divided by the corresponding reference norm. Mean cosine uses defined nonzero pairs and the Frobenius inner product for Hessians.

We report median synchronized wall-clock times for the full panel over seven runs after five batch warmups. Each timed call includes all prerequisite computations.

For reference, adaptive quadrature on the CPU takes 38.14 s for energy and 58.40 s for energy plus all 19 input derivatives on the same panel (medians over seven runs). These measurements use 32 threads on CPU.

F Ablation Studies

We assess the relative contributions of structural energy factorization (F) and Sobolev derivative supervision (S) across four model configurations: Unfactored Value-only (UV), Unfactored Sobolev

Table 1: **Potential fidelity and runtime.** The data were recorded on 65,536 configurations. Quad. denotes NCF-style quadrature ($n = 40$). Runtime rows for E, g, and H measure complete E, $E + g$, and $E + g + H$ calls, respectively.

Quantity	MRE (%) ↓		Mean cosine ↑		Runtime (ms) ↓	
	SNCP	Quad.	SNCP	Quad.	SNCP	Quad.
Energy E	**16.78**	59.37	–	–	**255.99**	602.41
Gradient g_{19}	**18.67**	52.11	**0.992779**	0.937439	**529.54**	641.99
Hessian H_{19}	**29.49**	86.69	**0.952922**	0.780637	10476.87	**4686.26**

(US), Factored Value-only (FV), and Factored Sobolev (FS). All models are evaluated across 3 independent training seeds on the 10M validation set.

For each model, p_{95} denotes the 95th percentile of the per-configuration errors on the applicable evaluation mask: approximately 95% of errors are at or below this value, and 5% are above it. Similarly, p_{01} denotes the first percentile of the per-configuration cosine similarities.

Table 2: **Validation performance across ablation models.** The results are reported in mean ± std across 3 seeds. Relative metrics are dimensionless; opposite-direction rate indicates negative gradient cosine percentage.

Metric	UV	US	FV	FS (Ours)
Energy Relative p95	707.72 ± 0.19	1.973 ± 0.115	39.236 ± 0.005	$\mathbf{1.951 \pm 0.012}$
Coordinate Relative p95	$(2.77 \pm 0.26) \times 10^6$	1.379 ± 0.040	$(4.69 \pm 0.25) \times 10^5$	$\mathbf{1.304 \pm 0.004}$
Coordinate Cosine p01	-0.529 ± 0.008	-0.711 ± 0.002	-0.527 ± 0.003	$\mathbf{-0.693 \pm 0.000}$
Opposite Direction (%)	49.99 ± 0.01	7.75 ± 0.13	50.01 ± 0.01	$\mathbf{7.26 \pm 0.01}$
Radius Relative p95	$(9.82 \pm 0.76) \times 10^5$	1.142 ± 0.076	$(1.17 \pm 0.06) \times 10^5$	$\mathbf{0.984 \pm 0.001}$

G Simulation Performance and Scene Parameters

Table 3 reports the XPBD iteration settings, per-frame runtimes, and Newton convergence counts for the three comparison scenes in Figure 2. Table 4 lists scene geometry, material parameters, contact counts, and runtimes for the SNCP/Newton simulations.

XPBD divides each frame of duration Δt into S substeps of size $h = \Delta t/S$. More substeps improve temporal resolution and can reduce integration error. Within each substep, I_p position iterations repeatedly correct elastic and contact constraints; more iterations generally reduce residual stretching and contact-constraint violations. I_v velocity iterations refine active velocity constraints after all substeps. Increasing these counts raises computational cost; additional iterations refine the constraint solve without advancing time.

Table 3: **Comparison solver settings and simulation performance.** XPBD uses S substeps per frame, I_p position iterations and I_v velocity iterations per substep. Runtime is the mean synchronized simulation time over all recorded frames, excluding initialization and export. Unconverged counts refer to Newton frames that did not establish native convergence; – denotes a setting or diagnostic that does not apply.

Scene	Method	S	I_p	I_v	Time/frame (s)	Unconverged
Large mass ratio	NCF/XPBD	10	24	5	1.9818	–
	NCF/Newton	–	–	–	3.8603	167/200
	SNCP/XPBD	10	24	5	1.6127	–
	SNCP/Newton	–	–	–	0.5019	0/200
Three-cube stacking	NCF/XPBD	20	5	5	0.0306	–
	NCF/Newton	–	–	–	1.1303	155/200
	SNCP/XPBD	20	5	5	0.8805	–
	SNCP/Newton	–	–	–	0.0367	0/200
Cloth on armadillo	NCF/XPBD	5	1	5	1.3540	–
	NCF/Newton	–	–	–	6.7529	571/600
	SNCP/XPBD	5	1	5	0.2773	–
	SNCP/Newton	–	–	–	0.1692	0/600

Table 4: **Scene parameters and simulation performance.** SNH denotes stable Neo-Hookean elasticity, and ABD denotes affine body dynamics. Mesh sizes include fixed mesh obstacles. Contacts are sampled at the final Newton assembly of each frame. The two densities for Large Mass Ratio refer to the heavy and light cubes, respectively. Time/frame excludes initialization and export.

	Mesh size			Δt	Model	ρ	E	ν	Contacts		Time/frame
Scene	Vertices	Faces	Tets	(s)		(kg/m^3)	(Pa)		Avg.	Max.	(s)
Spikes (Fig. 3(a))	10	12	0	0.01	ABD	1000	–	–	2.6	13	0.0118
Spike–Wedge (Fig. 3(b))	11	14	0	0.01	ABD	1000	–	–	1.8	6	0.0142
Wedges (Fig. 3(c))	12	16	0	0.01	ABD	1000	–	–	5.1	23	0.0107
Spike in Hole (Fig. 3(d))	58	108	0	0.01	ABD	1000	–	–	1.8	13	0.0090
Spike in Crack (Fig. 3(e))	25	42	0	0.01	ABD	1000	–	–	4.4	8	0.0096
Wedge in Crack (Fig. 3(f))	26	44	0	0.01	ABD	1000	–	–	6.7	12	0.0090
Internal Edges (Fig. 3(g))	22	36	5	0.01	ABD	1000	–	–	34.6	53	0.0146
Cliff Edges (Fig. 3(h))	16	24	5	0.01	ABD	1000	–	–	11.8	62	0.0105
Large Mass Ratio (Fig. 2(a))	9288	4848	40040	0.01	SNH	2000/1000	3×10^6	0.49	256.1	655	0.5019
Three Cube Stack (Fig. 2(b))	32	48	15	0.01	SNH	1000	10^{12}	0.49	12.7	54	0.0367
Armadillo Cloth (Fig. 2(c))	7427	14652	0	0.005	Cloth	200	10^7	0.49	178.3	616	0.1692
Mat Twist (Fig. 6)	7200	14042	0	0.01	Cloth	200	10^7	0.49	13271.0	25966	22.4651
Falling Bunnies (Fig. 4)	20064	38016	53760	0.01	SNH	1000	7×10^5	0.4	5475.7	14624	15.7961
Squishy Ball (Fig. 5)	163814	285080	444491	0.01	SNH	1000	5.67×10^7	0.417	6014.1	34582	19.5471
Struck Cubes. (Fig. 7)	12510	18636	29343	0.01	ABD	1000	–	–	9899.4	27216	4.9436