FRACFUSION: A HETEROGENEOUS SIMULATION-RENDERING FRAMEWORK FOR VEHICLE IMPACT AND STRUCTURAL BREAKAGE

Anonymous authors
Paper under double-blind review



Figure 1: **Representative hybrid crash sequence of our FracFusion.** Our heterogeneous framework enables **mutual structural breakage**: the nodebeam vehicle undergoes componentwise structural bending and fracture, while the MPM3DGS obstacle produces materiallevel arbitrary cracks and flying fine debris.Existing baseline solvers can only realize one of these two structural behaviours at a time.

ABSTRACT

High-fidelity simulation of simultaneous structural breakage in vehicleobstacle collisions is critical for autonomousdriving safety testing and neural world modeling. Real crash data is scarce and costly to acquire, making simulation the main source of diverse accident scenarios. Existing pipelines face a fundamental tradeoff: nodebeam solvers capture realistic vehiclecomponent structural fracture but cannot generate arbitrary continuum fragmentation for obstacles, whereas MPMbased 3D Gaussian Splatting (3DGS) produces debris but discards calibrated vehicle mechanics when vehicles are converted to particles. Naively combining solvers cannot resolve heterogeneous contact, consistent bidirectional momentum transfer, and persistent interpenetration artifacts. We present FracFusion, a heterogeneous simulationrendering framework. It keeps calibrated nodebeam vehicle assets intact and uses MPMaugmented 3DGS for photorealistic brittle obstacle fracture. Lightweight shell proxy particles enable stable bidirectional impulse coupling within CollisionMPM without redundant vehicle dynamics. We further incorporate particletriangle antipenetration correction and physicsaware depth compositing to mitigate crossrepresentation artifacts. Experiments and ablations validate our design. Physical checks verify momentum and energy stability under highenergy impacts, and perceptual studies show improved visual plausibility. FracFusion delivers crash simulation featuring both vehiclecomponent structural fracture and obstacle continuum fragmentation for safetycritical autonomous simulation and worldmodel applications.

1 INTRODUCTION

Safetycritical simulation and neural world models demand faithful reproduction of highenergy structural breakage events. For autonomousdriving virtual testing, realworld crash data is scarce and costly to obtain; simulation is therefore essential for generating diverse accident scenarios to validate perception and planning systems under vehicle damage and flying debris (Maul et al., 2021; Ge et al., 2026). Meanwhile, neural world models can reconstruct largescale realworld scenes, yet most fail to predict scene evolution once highenergy impacts and structural fracture occur (Zhang et al., 2024; Xie et al., 2024). A key unresolved limitation is the lack of simulation tools that simultaneously reproduce vehicle componentwise structural fracture and continuumlevel obstacle fragmentation.

Realworld vehicleobstacle collisions couple two distinct physical responses: vehicles undergo componentlevel bending, plasticity and fracture governed by perpart mechanical properties, while brittle

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.

obstacles produce arbitrarily oriented cracks and fine flying fragments. Todays dominant simulation paradigms present an unavoidable tradeoff.

Nodebeam frameworks (e.g., BeamNG) reuse engineeringcalibrated vehicle assets and faithfully simulate componentwise vehicle fracture, yet fracture is constrained to predefined beam topology and cannot generate arbitrary obstacle fragmentation (Maul et al., 2021). By contrast, physicsaugmented 3D Gaussian Splatting pipelines such as PhysGaussian, FractureGS and GaussianFluent natively support continuum fragmentation by treating all objects as MPM particles (Xie et al., 2024; Wang et al., 2026; Huang et al., 2026). However, converting vehicles fully into particles discards engineered component topology and mechanical parameters, and recalibrating these parameters is prohibitively expensive; treating vehicles as rigid bodies removes vehicle deformation entirely. Simply stitching two independent solvers cannot resolve core difficulties including crossrepresentation contact proxying, bidirectional impulse mapping, persistent interpenetration and multibackend occlusion inconsistency.

To address this gap, we present **FracFusion**, a heterogeneous simulationrendering pipeline for highenergy vehicleobstacle structural breakage. Our divisionoflabor strategy preserves native nodebeam representation for vehicles to retain calibrated mechanical behaviour, while brittle obstacles are simulated via MPM and rendered with 3D Gaussian Splatting for photorealistic fragmentation. We introduce shellproxyparticle primitives on vehicle surfaces to interface nodebeam structures with CollisionMPM splitfield contact, enabling complete bidirectional impulse feedback without converting vehicles to particles. A complementary particletriangle penalty mitigates heterogeneous interpenetration artifacts. The whole pipeline runs on GPU with sparsegrid acceleration, and physicsaware depth compositing resolves meshGaussian occlusion conflicts. We construct multiple highenergy impact test scenarios and conduct qualitative comparisons, user studies, ablations and momentumenergy conservation validation against representative baselines. Figure 1 shows representative hybrid crash outputs produced by our FracFusion.

Our main contributions are summarized as follows:

- **Heterogeneous cosimulation architecture.** A divisionoflabor paradigm coupling nodebeam structural solver and MPM3DGS fracture solver, reusing precalibrated vehicle assets to realize simultaneous vehiclecomponent structural fracture and continuum obstacle fragmentation.
- **Crossrepresentation bidirectional contact coupling.** A shellproxyparticle interface connecting nodebeam structures with CollisionMPM for full bidirectional impulse transfer, augmented by particletriangle penalty to suppress persistent interpenetration.
- **Endtoend GPU simulationrendering and occlusion handling.** Sparsegrid accelerated hybrid GPU implementation for our heterogeneous coupling; we propose physicsaware depth compositing to eliminate meshGaussian rendering occlusion errors.

2 Related Work

2.1 PhysicsIntegrated 3D Gaussian Splatting for Dynamics

3D Gaussian Splatting (3DGS) enables highfidelity realworld reconstruction and fast differentiable rendering (Kerbl et al., 2023). PhysGaussian pioneers embedding 3D Gaussian primitives as Lagrangian MPM particles, unifying simulation and rendering primitives (Xie et al., 2024). Building upon this paradigm, a family of followup works targets fracture, material parameter inference and scenelevel simulation (Borycki et al., 2024; Cai et al., 2024; Wang et al., 2026; Huang et al., 2026; Liu et al., 2026). Additional lines of work combine Gaussian representations with rigidbody simulation, dynamic scene generation and urban driving reconstruction, though they lack support for componentwise structural fracture under highenergy impact (Ramlal & Zelek, 2026; Dong et al., 2026; Zhang et al., 2024; Qiu et al., 2024; Ge et al., 2026).

Despite producing compelling fracture visuals, these methods rely on homogeneous particleparticle contact. They cannot directly ingest precalibrated nodebeam structural assets: converting vehicles to particles discards componentlevel mechanical parameters, while treating vehicles as rigid bodies eliminates vehicle deformation and structural breakage.

2.2 MPM AND CONTINUUM DYNAMIC FRACTURE

The Material Point Method (MPM) is a wellestablished particlegrid solver for large deformation and dynamic fracture (Sulsky et al., 1994). APIC improves particlegrid interpolation for rotational motions (Jiang et al., 2015). CDMPM introduces the NACC constitutive model together with splitfield CollisionMPM contact, forming the foundation for modern GaussianMPM fracture pipelines (Wolper et al., 2019).

Multiple MPM variants extend the framework toward anisotropic damage, snow simulation, phase change, brittle crack propagation and mesoscale inelastic flow (Wolper et al., 2020; Stomakhin et al., 2013; 2014; Fan et al., 2022; Qu et al., 2023). Further developments cover highorder boundary treatments, implicit formulations and numerical optimization for geomechanics and impact problems (Liang et al., 2026; Wang et al., 2016; Gavin et al., 2024).

Conventional MPM workflows discretize all objects into particles, which incurs heavy parametercalibration overhead when discrete engineered structures are present. GPUaccelerated sparsegrid MPM improves runtime efficiency (Huang et al., 2026), a technique also adopted in our implementation.

2.3 NODEBEAM DISCRETE STRUCTURAL SIMULATION

The nodebeam model uses lumpedmass nodes connected by deformable beam elements, wellsuited for framelike and vehicle structures with tunable stiffness, plasticity and fracture thresholds. Early work established the core computational framework for inelastic structural simulation (Terzopoulos & Fleischer, 1988). BeamNG.tech widely adopts nodebeam assets to reproduce chassis collapse and component detachment in vehicle crash scenarios (Maul et al., 2021). Reducedorder nodebeam models have also been studied for electricvehicle control simulation (Dalboni & Soldati, 2019).

A fundamental limitation remains: fracture is restricted to predefined beam connections. Nodebeam representations cannot natively generate arbitrarily oriented internal cracks or fine flying debris for continuous media such as walls and ruins.

2.4 HETEROGENEOUS SOLVER COUPLING AND HYBRID PHYSICS

Coupling distinct physical solvers is a longstanding topic in computational mechanics. Representative combinations include FEMMPM, DEMMPM and smootheddynamicsMPM for geomechanical impact and fluidstructure interaction (Lian et al., 2011; Ren et al., 2023; He et al., 2017; Sun et al., 2024; Chen et al., 2015). These existing coupling approaches target offline finiteelement analysis and do not integrate 3DGaussian neural rendering pipelines.

Standard CollisionMPM requires every colliding body to be represented as MPM particles. No existing mechanism enables an external nodebeam solver to interface with splitfield contact via surface proxies and realize complete bidirectional impulse feedback. Heterogeneous representation coupling further brings persistent interpenetration challenges. On the rendering side, naive alpha compositing between meshes and Gaussian splats suffers from depthleakage artifacts (Xie et al., 2024; Wang et al., 2026).

Taken together, existing methods suffer an inherent tradeoff: nodebeam solvers restrict obstacle fragmentation, while MPM3DGS loses calibrated vehicle mechanics. Our FracFusion resolves this gap with shellproxy coupling, antipenetration correction and physicsaware depth compositing.

3 METHOD

Notation We adopt the following notation throughout the paper: r denotes a physical node in the nodebeam system; s denotes a shell proxy particle; p denotes an MPM obstacle particle; i denotes a background MPM grid node. $T(\cdot)$ returns the parentnode set associated with an object; w_{sr} is the barycentric weight from shell proxy s to parent node r; N_{si} is the quadratic Bspline interpolation kernel from shell particle s to grid node i; Δt is the simulation timestep.

Overview Figure 2 illustrates the full simulationrendering pipeline of FracFusion. Targeting highenergy vehicleobstacle impacts with mutual structural breakage, we propose a heterogeneous coupled simulation paradigm. The vehicle is represented by a nodebeam formulation that captures componentlevel structural deformation and fracture (Terzopoulos & Fleischer, 1988; Maul et al., 2021). Brittle obstacles such as walls and ruins are solved by the Material Point Method (MPM) and rendered with 3D Gaussian Splatting (3DGS) for continuum fragmentation and photorealistic appearance (Kerbl et al., 2023; Xie et al., 2024).

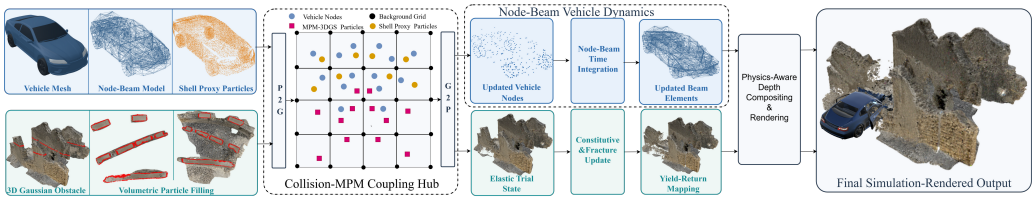


Figure 2: System overview. A vehicle is represented by structural nodes, beams and shell proxy particles, while a 3DGS obstacle is volumetrically filled with MPM particles. Hybrid MPM couples the two representations; subsequent nodebeam structural integration, MPM constitutive fracture solving, and physicsaware depth compositing produce occlusionconsistent final rendering outputs.

Directly feeding sparse physical nodebeam nodes into MPM projection yields poor surface contact sampling (Wang et al., 2026). Instead, **only lightweight shell proxy particles participate in P2G projection for the vehicle side**. Shellproxy total mass matches the aggregate mass of parent physical nodes: $\sum_s m_s^{\text{proxy}} = \sum_r m_r$, preserving vehicleside mass conservation. Proxies carry no independent dynamics: their positions and velocities are interpolated from parent nodes at the beginning of every timestep, and they are never updated during GridtoParticle (G2P). Bidirectional momentum exchange between the nodebeam vehicle and the MPM obstacle is performed on a shared CollisionMPM grid (Wang et al., 2026). After contact solving, the collision impulse $\boldsymbol{I}_i$ is scattered back via proxies only to physical nodebeam nodes to drive structural integration. Obstacle MPM particles undergo constitutive update and yieldreturn to reproduce large plastic deformation, crack propagation and fragment separation. Finally, a physicsaware depth compositing strategy fuses the deformed vehicle mesh and fractured 3DGS obstacles to produce occlusionconsistent imagery. The pipeline runs iteratively, timestep by timestep.

Full simulation hyperparameters, asset statistics and reproduction configurations are summarized in Appendix A. Complete asset preprocessing workflows for obstacles and nodebeam vehicles are documented in Appendix B. Details of our GPU kernel implementation using Warp are given in Appendix F.

3.1 Heterogeneous Bidirectional Coupling via CollisionMPM

The workflow of our sharedgrid bidirectional coupling is visualized in Figure 3. We adopt CollisionMPM as the mechanical hub between the two representations (Wang et al., 2026). No explicit pairwise collision detection is run; all momentum and impulse exchange occurs on the shared background grid. The coupling has three substeps: P2G projection, grid splitfield contact solving, and G2P backprojection.

3.1.1 P2G Projection

Two particle sets contribute to P2G: (1) all MPM obstacle particles $\{p\}$ from obstacles; (2) shell proxy particles $\{s\}$ sampled on vehicle collision surfaces. **Physical nodebeam nodes $\{r\}$ are not directly projected onto the background grid; all vehicleside massmomentum information originates purely from shell proxies.** Using the quadratic Bspline kernel of APICMPM (Jiang et al., 2015), mass and momentum are projected onto grid node i:

$$m_i^a = \sum_{p \in P_a} m_p \, N(\boldsymbol{x}_p - \boldsymbol{z}_i), \tag{1}$$

$$m_i^b = \sum_s m_s^{\text{proxy}} \, N(\boldsymbol{x}_s - \boldsymbol{z}_i). \tag{2}$$

Here P_a is the obstacle particle set; P_b denotes vehicleside shellproxy particle set; $\boldsymbol{z}_i$ is the gridnode coordinate; m_i^a, m_i^b are the two splitfield channel masses. Proxy mass satisfies massmatching constraint $\sum_s m_s^{\text{proxy}} = \sum_{r \in \text{vehicle}} m_r$, ensuring exact totalmass conservation for vehicle bodies. Two independent fields P_a, P_b are kept on the grid to avoid unphysical momentum mixing before contact solving. Shell proxies provide dense surface sampling over the car body and mitigate missing contact caused by sparselydistributed physical structural nodes.

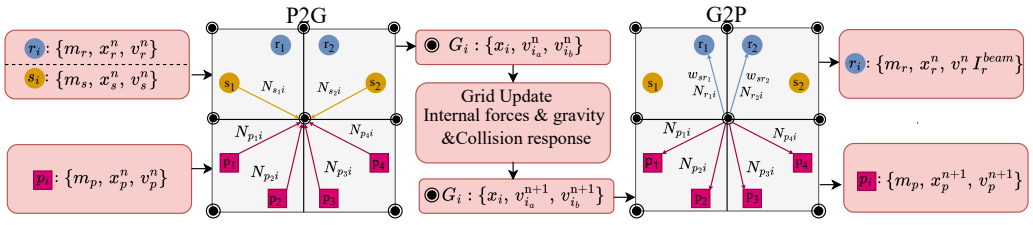


Figure 3: **Sharedgrid heterogeneous coupling mechanism.** Only shell proxy particles (vehicle-side) and obstacle particles are projected onto two independent massmomentum fields on the shared background grid; physical nodebeam nodes are never fed into P2G projection. Collision impulse is computed on overlapping grid cells, and then propagated back via shell proxies exclusively to nodebeam structural nodes for structural integration. Shell proxies remain readonly and receive no G2P updates.

3.1.2 Grid SplitField Contact

Splitfield contact is solved only on cells satisfying both $m_i^a > 0$ and $m_i^b > 0$. The normaldirection collision impulse is

$$\boldsymbol{I}_i = \gamma \frac{m_i^a m_i^b}{m_i^a + m_i^b} d_i \, \boldsymbol{n}_i^a, \tag{3}$$

where d_i is the relative velocity along interface normal $\boldsymbol{n}_i^a$ and γ is the contact response coefficient. The impulse is recorded separately on each field without premature field merging (Wolper et al., 2019).

3.1.3 G2P BackProjection

Differentiated handling is applied to the two families:

1. For obstacle field P_a, standard G2P interpolates grid velocities back to particles, updating $\boldsymbol{v}_p$, $\boldsymbol{x}_p$, deformation gradient $\boldsymbol{F}_p$ and velocity gradient $\boldsymbol{C}_p$, which feed the constitutive solve.
2. For vehicle field P_b, shell proxies $\{s\}$ **receive no state update from G2P**. The grid impulse is first interpolated onto proxies,

$$\boldsymbol{I}_s = m_s^{\text{proxy}} \sum_i N_{si} \frac{\boldsymbol{I}_i}{m_i^b}, \tag{4}$$

then scattered to parent nodes and converted into a nodal force:

$$\boldsymbol{I}_r^{\text{beam}} = \sum_{s:\, r \in T(s)} w_{sr} \, \boldsymbol{I}_s, \qquad \boldsymbol{f}_r^c = \frac{\boldsymbol{I}_r^{\text{beam}}}{\Delta t}. \tag{5}$$

The force $\boldsymbol{f}_r^c$ is accumulated into the nodal buffer for the next nodebeam integration.

3.2 NodeBeam Vehicle Dynamics

Receiving $\boldsymbol{f}_r^c$, the nodebeam system is integrated explicitly. Figure 4 (a) illustrates our nodebeam update scheme.

1. **Nodal integration**:

$$\boldsymbol{v}_r^{t+\Delta t} = \boldsymbol{v}_r^t + \frac{\Delta t}{m_r}\left(\boldsymbol{f}_r^{\text{beam}} + \boldsymbol{f}_r^c\right), \quad \boldsymbol{x}_r^{t+\Delta t} = \boldsymbol{x}_r^t + \Delta t \, \boldsymbol{v}_r^{t+\Delta t}, \tag{6}$$

where $\boldsymbol{f}_r^{\text{beam}}$ is the beaminternal force assembled in this step.

2. **Beam force assembly**: for each intact beam with current length l and rest length l_0, the axial force is

$$f^{\text{beam}} = k\,(l - l_0) + d\,\frac{\Delta l}{\Delta t}, \tag{7}$$

applied as equal and opposite forces on the two end nodes.

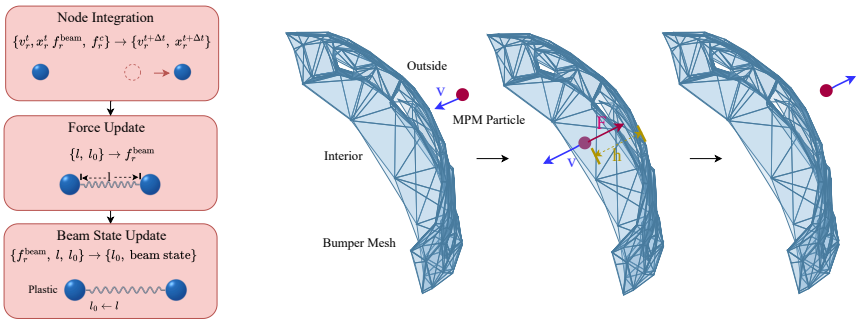


Figure 4: **Nodebeam integration and particletriangle antipenetration correction.** (a) Explicit update pipeline for nodebeam vehicle: nodal integration, beam internal force assembly, and plasticitydriven beam fracture. (b) Illustration of particletriangle penalty: when MPM obstacle particles penetrate thin vehicle shell meshes, outward velocity correction is applied and reaction impulse is fed back to structural nodes to mitigate hidden interpenetration artifacts missed by gridonly CollisionMPM contact.

3. **Plasticity and fracture**: when stretching exceeds the plastic threshold, the rest length is updated $l_0 \leftarrow l$ to record permanent deformation; if it exceeds the breaking strength, the beam is removed to simulate connection failure (Terzopoulos & Fleischer, 1988; Maul et al., 2021).

At the end of the step, shellproxy positions and velocities are reinterpolated from updated physical nodes:

$$\boldsymbol{x}_s = \sum_{r \in T(s)} w_{sr} \boldsymbol{x}_r, \qquad \boldsymbol{v}_s = \sum_{r \in T(s)} w_{sr} \boldsymbol{v}_r. \tag{8}$$

3.3 MPM Constitutive Update and Fracture Evolution

Obstacle particles receive trial deformation $\boldsymbol{F}_p^{\text{trial}}$ from G2P. After SVD, a returnmapping under the NACC cap yield surface yields the admissible deformation $\boldsymbol{F}_p$, Kirchhoff stress $\boldsymbol{\tau}_p$ and plastic volume J_p (Wolper et al., 2019).

The accumulated hardening variable evolves as

$$H_p^{t+\Delta t} = H_p^t + \Delta H(\boldsymbol{F}_p^{\text{trial}}, \boldsymbol{F}_p). \tag{9}$$

When $H_p > H_{\text{thresh}}$, particle p is marked fractured and contributes to fracturesurface Gaussian reconstruction. Its rendering attributes $(\boldsymbol{A}_p, \rho_p, \boldsymbol{c}_p)$ travel with it, so fragments render directly without extra geometry reconstruction (Xie et al., 2024; Wang et al., 2026).

3.4 Particle–Triangle AntiPenetration Penalty

CollisionMPM contact relies on both mass channels coexisting on a cell (Wolper et al., 2019). Once an MPM particle penetrates inside a thin vehicle shell, the cell no longer records both masses and grid contact fails. We add a local particletriangle penalty as a complementary guard. Figure 4 (b) illustrates our complementary particletriangle antipenetration correction.

A uniform spatial hash is built on the deformed collision triangles for fast candidate queries. Let $h_{p\tau}$ be the penetration depth and $\boldsymbol{n}_\tau$ the triangle normal. Velocity correction is applied only when the particle lies inside the triangle and continues inward:

$$\Delta \boldsymbol{v}_p = \min\left(k h_{p\tau},\, \Delta v_{\max}\right) \boldsymbol{u}_{p\tau}, \tag{10}$$

where $\boldsymbol{u}_{p\tau}$ is the outward unit normal.

The resulting impulse $\Delta \boldsymbol{I}_p = m_p \Delta \boldsymbol{v}_p$ is scattered back to the parent nodes by barycentric weights. For quantitative ablation we define a timeaveraged penetration burden

$$F = \frac{1}{NT} \sum_{p=1}^{N} \sum_{t=1}^{T} \max\left(h_{p,t} - h_0, 0\right), \tag{11}$$

where h_0 is a visually acceptable tolerance, N the particle count and T the step count. This module suppresses persistent visible penetration rather than enforcing zero instantaneous penetration.

3.5 PHYSICSAWARE DEPTH COMPOSITING

Two inputs are rendered: the nodebeamdeformed vehicle mesh and the fractured MPM3DGS obstacle particles (Kerbl et al., 2023). Native weighted depth from 3DGS rasterization leaks along semitransparent Gaussian edges. We instead use the physical cameraspace depth z_p^{phys} of MPM particles. For pixel (u, v), $Z_{\text{GS}}(u, v)$ is the minimum depth among covering particles, and $Z_{\text{mesh}}(u, v)$ is the meshrendered depth.

The perpixel occlusion decision is

$$O(u,v) = \begin{cases} \text{draw GS}, & Z_{\text{GS}}(u,v) < Z_{\text{mesh}}(u,v), \\ \text{draw mesh}, & \text{otherwise}. \end{cases} \tag{12}$$

Alpha compositing guided by $O(u, v)$ guarantees imagespace occlusion consistency with the physical simulation state.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmark Scenarios We construct three highenergy impact scenarios: (i) a sedan crashing into a reinforced concrete wall at high speed; (ii) a car hitting a masonry ruin; (iii) a vehicle colliding with a table. Each scenario pairs a calibrated nodebeam vehicle asset with a 3DGS obstacle reconstructed from realworld capture and filled with interior MPM particles. Full simulation hyperparameters, asset statistics and reproduction configurations are summarized in Appendix A.

Baselines We compare against two baselines: (1) **Nodebeam only**: the vehicle and obstacle are both represented by nodebeam elements, as in the BeamNG ecosystem (Maul et al., 2021); (2) **FractureGS**: the vehicle and obstacle are represented as CollisionMPM particles following the FractureGS paradigm (Xie et al., 2024; Wang et al., 2026). Additional qualitative simulation sequences are provided in Appendix D.

Metrics We report both visual and physical metrics. Visually, we conduct a user study asking 30 participants to rate structural response, material fracture, and appearance/rendering quality on separate 5point Likert scales, as shown in Table 1. Physically, we measure (a) **momentum deviation** $\|\Delta \boldsymbol{P}\|/\|\boldsymbol{P}_0\|$ over the full sequence, (b) **energy stability** as the coefficient of variation of total energy after contact, and (c) the **penetration burden** F from Eq. (10). All methods run on a single GPU with identical sparse grid resolution and timestep configuration.

4.2 QUALITATIVE COMPARISON

Figure 5 provides qualitative comparison and coupling ablations at matched simulation timestamps. The pure nodebeam baseline yields realistic vehicle deformation, yet obstacle fracture is strictly topologylocked with limited fine debris. Conversely, FractureGS generates rich obstacle fragmentation, but vehicle structural component response vanishes, deforming as a homogeneous particle lump. Our hybrid FracFusion achieves simultaneous vehicle componentlevel collapse and continuum obstacle fragmentation without persistent interpenetration. Detailed perrow analysis for each ablation variant can be found in Appendix C.

Table 1: Simulationonly time and mean userstudy scores (1–5; higher is better for the three perceptual dimensions).

Method	Simulation only (ms/step)	Structural response	Material fracture	Appearance / rendering
Pure node–beam	1.28	3.9	1.2	3.0
FractureGS	16.95	2.1	3.7	3.3
Ours (hybrid)	9.61	3.8	3.9	4.0

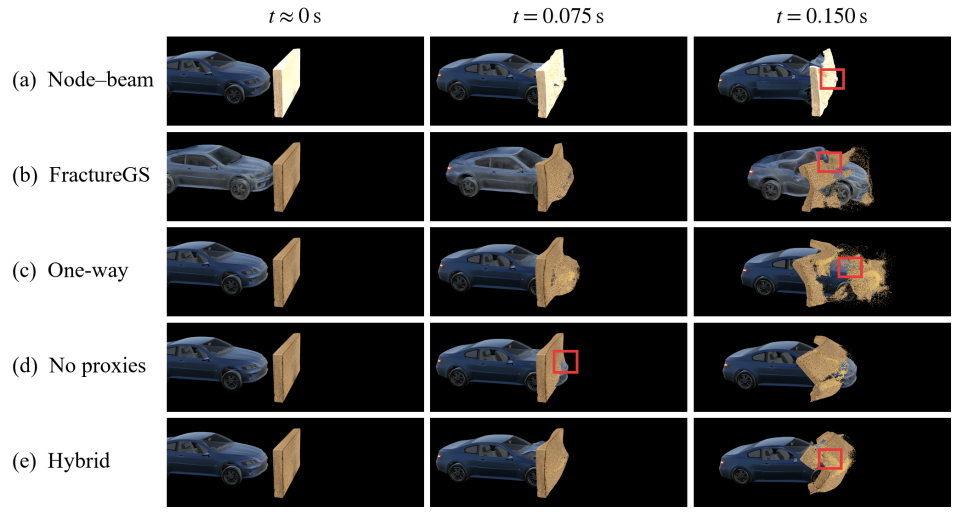


Figure 5: **Qualitative comparison and coupling ablation under matched timestamps.** Rows (a) and (b) are the two baseline methods (Node-beam and FractureGS); rows (c) and (d) ablate impulse feedback and shell proxy particles, respectively; row (e) is the full hybrid method. The red boxes highlight representative contact regions.

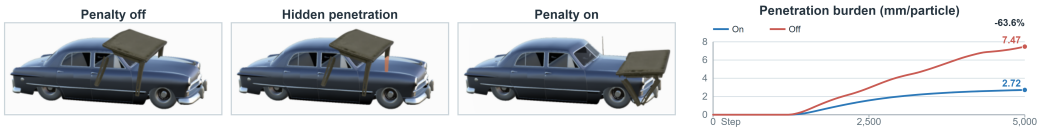


Figure 6: **Ablation of particletriangle penalty antipenetration module.** Left three panels compare visual outputs with/without penalty; righthand curve reports penetration burden over simulation steps.

4.3 Ablation Study

We evaluate four key ablations on representative impact scenarios: oneway coupling, removal of shell proxy particles, removal of antipenetration correction, and removal of physicaldepth compositing.

- **Oneway coupling**: shell proxies still transfer vehicle motion onto the grid, but reaction impulse is not fed back to nodebeam nodes, leading to negligible vehicle deformation upon collision (Fig. 5 (c)).
- **Shell proxies off**: direct node projection causes insufficient contact sampling, an increased penetration burden and asymmetric obstacle cracking (Fig. 5 (d)).
- **Antipenetration off**: hidden interpenetration artifacts occur between vehicle shell and obstacle particles (Fig. 6).
- **Physicaldepth compositing off**: naive Gaussian alpha compositing produces depthleakage artifacts at fragment edges (Fig. 7).

Disabling any single component visibly degrades simulation or rendering quality. Full subfigure interpretations for these ablation and validation plots can be found in Appendix C. Aggregated quantitative ablation metrics are reported in Appendix E.

4.4 Physical Validation

We examine momentum and energy conservation of our heterogeneous coupling pipeline under highenergy crash loading in Figure 8. Overall system energy remains bounded without spurious energy explosion, and total system momentum is well preserved throughout impact and fragmentation. Shellproxy particles introduce mild and controllable numerical dissipation. Stepbystep subplot interpretation is provided in Appendix C.

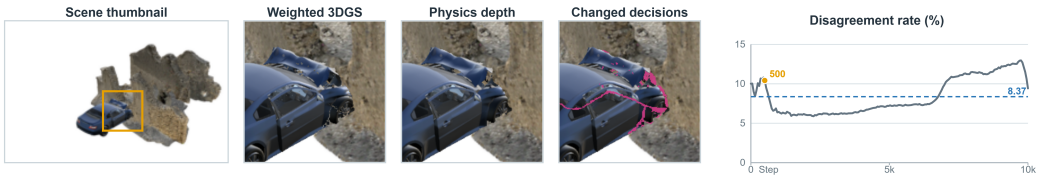


Figure 7: **Consistency evaluation between weighted 3DGS rendering and physicsaware depth.** Left panels show scene overview and zoomed collision region; right curve reports pixelwise depthdisagreement rate across timesteps.

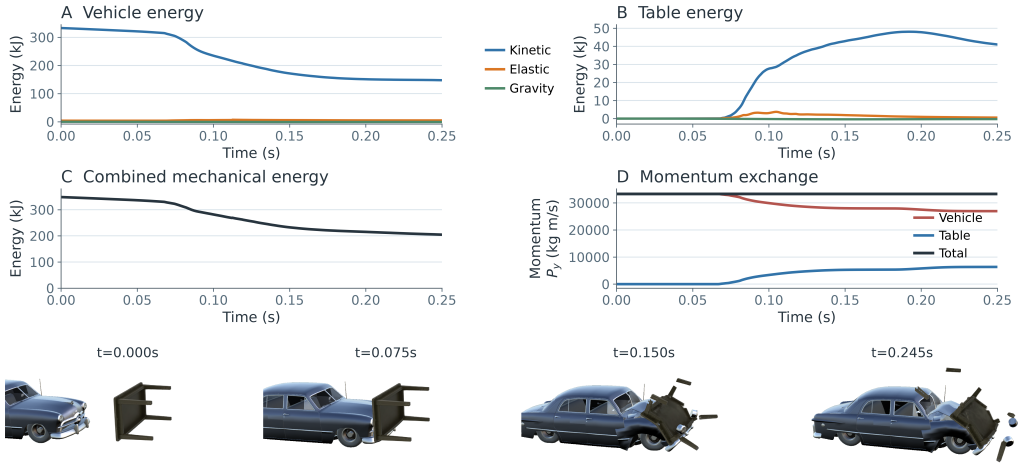


Figure 8: **Physical sanity validation: energy stability and momentum exchange on vehicletable crash scenario.** Subplots decompose subsystem energy and momentum; bottom row shows simulation snapshots at representative timestamps.

5 Conclusion and Future Work

We introduced FracFusion, a heterogeneous simulationrendering pipeline for highenergy vehicleobstacle structural breakage. Retaining native nodebeam vehicle assets and employing MPM3DGS for brittle obstacles, our shellproxyparticle interface built on CollisionMPM grid contact realizes bidirectional structural responses: componentlevel vehicle collapse alongside continuum obstacle fragmentation. We mitigate interpenetration via a particletriangle penalty and resolve meshGaussian occlusion conflicts with physicsaware depth compositing. Our design is validated by qualitative comparisons, user studies, ablations, and momentumenergy conservation checks.

Broader impact FracFusion targets virtual crash testing, neural world modeling and highfidelity destructive simulation. It reduces reliance on costly realworld crash trials to generate diverse impact scenarios, enriching safetycritical training data for perception and planning models. The shellproxy coupling generalizes beyond vehicles to other nodebeam structures (trusses, articulated robots, frames). Nevertheless, two limitations remain . Material parameters are visually tuned rather than fitted to real crash measurements, so output forcedisplacement results are not engineeringCAE grade. Moreover, our coupling targets nodebeam / MPM3DGS; supporting FEM or SPH would require redesigning the surfaceproxy and impulsemapping layers. Appendix G further discusses observable failure modes and practical boundaries of our hybrid framework.

Future work We highlight three promising research directions. First, calibrating material parameters against real measured crash data can better align visual plausibility with physical accuracy. Second, endtoend differentiable coupling would enable gradientdriven impactscenario optimization for stressexposure evaluation. Third, extending the shellproxy interface to additional structural solvers, such as FEM shells, SPH fluids and rigidbody hulls, can expand the scope of mixedrepresentation interactions beyond nodebeam / MPM3DGS.

AI USE STATEMENT

This work leverages artificial intelligence tools **only for grammatical correction, wording polishing, and fluent expression optimization** of the manuscript text. No AI model or tool is involved in the core research process, including but not limited to research idea design, theoretical derivation, algorithm design, experimental scheme formulation, data collection and processing, model training and testing, result analysis, and conclusion summarization. All intellectual contributions, core innovations, experimental designs, and analytical conclusions in this paper are independently completed by the authors. The use of AI tools does not affect the originality, integrity, and academic rigor of the research work.

ETHICS STATEMENT

This study strictly abides by academic ethics and ICLR ethical guidelines throughout the entire research and manuscript preparation process. There is no data forgery, tampering, plagiarism, duplicate submission, or improper academic behavior in this work. All referenced literatures, methods, and experimental materials are clearly cited and attributed in the manuscript. The research does not involve sensitive personal data, animal experiments, or any potentially harmful, discriminatory, or risky content that violates social ethics and academic norms. All authors have confirmed the rationality and compliance of the research content, and undertake full responsibility for the academic integrity and ethical compliance of this work.

REPRODUCIBILITY STATEMENT

All experiments, analyses, and conclusions presented in this paper are **fully reproducible**. The complete experimental settings, including model architecture details, hyperparameter configurations, training strategies, evaluation metrics, and experimental environment (software version, hardware configuration, operating system) are comprehensively and clearly described in the manuscript. There are no hidden experimental parameters, unstated operations, or unaccountable experimental processes. Researchers can completely replicate all experimental results and verify the corresponding conclusions of this paper by following the detailed steps provided in the manuscript. We ensure the stability and reliability of the experimental results, and all core findings are robust and repeatable under standard experimental conditions.

REFERENCES

Piotr Borycki, Weronika Smolak, Joanna Waczyńska, Marcin Mazur, Sławomir Tadeja, and Przemysław Spurek. GASP: Gaussian splatting for Physic-Based simulations. arXiv preprint arXiv:2409.05819, 2024.

Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. GIC: Gaussian-informed continuum for physical property identification and simulation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 75035–75063, 2024.

Z. P. Chen, X. M. Qiu, Xiong Zhang, and Yanping Lian. Improved coupling of finite element method with material point method based on a particle-to-surface contact algorithm. *Computer Methods in Applied Mechanics and Engineering*, 293:1–19, 2015. doi: 10.1016/j.cma.2015.04.005.

Matteo Dalboni and Alessandro Soldati. Soft-body modeling: A scalable and efficient formulation for control-oriented simulation of electric vehicles. In *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, pp. 1–6, 2019.

Xin Dong, Lihan Zhang, Tianru Dai, Wenfeng Deng, and Yansong Tang. Occlusion-robust multi-object decoupling for physics-based robotic interaction. arXiv preprint arXiv:2606.29303, 2026.

Linxu Fan, Floyd M. Chitalu, and Taku Komura. Simulating brittle fracture with material points. *ACM Transactions on Graphics*, 41(5):177, 2022. doi: 10.1145/3522573.

N. Gavin, G. Pretti, W. Coombs, J. Brigham, and C. Augarde. On the implementation of a material point-based arc-length method. *International Journal for Numerical Methods in Engineering*, 125(9):e7438, 2024.

Fudong Ge, Dingning Liu, Hanshi Wang, Yiwei Zhang, Jin Gao, Weiming Hu, and Zhipeng Zhang. SurfaceGS: Dynamic surface gaussian splatting for urban driving scenes. In *CVPR*, 2026. Findings track.

N. He, Y. Liu, and X. Zhang. Seamless coupling of molecular dynamics and material point method via smoothed molecular dynamics. *International Journal for Numerical Methods in Engineering*, 112(4):380–400, 2017.

Bei Huang, Yixin Chen, Ruijie Lu, Gang Zeng, Hongbin Zha, Yuru Pei, and Siyuan Huang. GaussianFluent: Gaussian simulation for dynamic scenes with mixed materials. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21583–21593, 2026.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023.

Yanping Lian, Xiong Zhang, and Yan Liu. Coupling of finite element method with material point method by local multi-mesh contact method. *Computer Methods in Applied Mechanics and Engineering*, 200(47–48):3482–3494, 2011. doi: 10.1016/j.cma.2011.07.014.

Y. Liang, J. Given, B. Chandra, Z. Zeng, X. Zhang, and K. Soga. The virtual stress boundary method to impose nonconforming neumann boundary conditions in the high-order material point method. *International Journal for Numerical Methods in Engineering*, 127(5):e70291, 2026.

Xiaoyang Liu, Shangzhe Wu, and Kai Han. Scene-level heterogeneous physics simulation with 3D gaussian splats. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6456–6465, 2026. Findings track.

Miles Macklin. Warp: A High-performance Python Framework for GPU Simulation and Graphics. https://github.com/NVIDIA/warp, March 2022. NVIDIA GPU Technology Conference (GTC).

Pascale Maul, Marc Mueller, Fabian Enkler, Eva Pigova, Thomas Fischer, and Lefteris Stamatogiannakis. BeamNG.tech technical paper. Technical report, BeamNG GmbH, 2021.

Ri-Zhao Qiu, Ge Yang, Weijia Zeng, and Xiaolong Wang. Language-driven physics-based scene synthesis and editing via Feature Splatting. In *European Conference on Computer Vision (ECCV)*, pp. 368–383, 2024. doi: 10.1007/978-3-031-72940-9_21.

Ziyin Qu, Minchen Li, Yin Yang, Chenfanfu Jiang, and Fernando de Goes. Power Plastics: A hybrid lagrangian/eulerian solver for mesoscale inelastic flows. *ACM Transactions on Graphics (TOG)*, 42(6):242:1–242:11, 2023. doi: 10.1145/3618344.

A. Ramlal and J. S. Zelek. PersistGS: Differentiable physics for object permanence in 4D gaussian splatting. In *CVPR Workshops*, 2026.

S. Ren, P. Zhang, Y. Zhao, X. Tian, and S. Galindo-Torres. A coupled metaball discrete element material point method for fluid-particle interactions with free surface flows and irregular shape particles. *Computer Methods in Applied Mechanics and Engineering*, 417:116440, 2023.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics (TOG)*, 32(4):1–10, 2013.

Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. Augmented MPM for phase-change and varied materials. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.

Deborah Sulsky, Zhen Chen, and Howard L. Schreyer. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):179–196, 1994.

Z. Sun, Y. Hua, Y. Xu, and X. Zhou. Simulation of fluid-structure interaction using the density smoothing B-spline material point method with a contact approach. *Computers & Mathematics with Applications*, 176:525–544, 2024.

Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *SIGGRAPH*, pp. 269–278, 1988.

B. Wang, P. J. Vardon, M. A. Hicks, and Z. Chen. Development of an implicit material point method for geotechnical applications. *Computers and Geotechnics*, 71:159–167, 2016. doi: 10.1016/j.compgeo.2015.08.008.

Xiaogang Wang, Hongyu Wu, Wenfeng Song, and Kai Xu. Fracture-GS: Dynamic fracture simulation with physics-integrated gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2026.

Joshuah Wolper, Yu Fang, Minchen Li, Jiecong Lu, Ming Gao, and Chenfanfu Jiang. CD-MPM: Continuum damage material point methods for dynamic fracture animation. *ACM Transactions on Graphics (TOG)*, 38(4):1–15, 2019.

Joshuah Wolper, Yunuo Chen, Minchen Li, Yu Fang, Ziyin Qu, Jiecong Lu, Meggie Cheng, and Chenfanfu Jiang. AnisoMPM: Animating anisotropic damage mechanics. *ACM Transactions on Graphics (TOG)*, 39(4):37:1–37:16, 2020. doi: 10.1145/3386569.3392428.

Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D gaussians for generative dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4389–4398, 2024.

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. Physdreamer: Physics-based interaction with 3D objects via video generation. In *European Conference on Computer Vision (ECCV)*, pp. 388–406, 2024.

A Simulation Parameters and Reproduction Settings

This appendix summarizes material parameters, mesh resolution, timestep configuration and asset specifications used in all main paper experiments. All quantitative evaluations are performed on a single RTX 3090 GPU.

A.1 MPM Object Material Configuration

Tab. 2 lists MPM constitutive parameters for major obstacle assets. E denotes Young's modulus, ν Poisson's ratio, ρ density. For NACC capyield constitutive model (Wolper et al., 2019), J_p is initial plastic volume, β tensioncompression weight factor, ξ compaction hardening steepness, and ϕ friction angle. Simulation domain size is 8 m side length unless noted, background ground friction coefficient 0.8.

Table 2: MPM material parameters for obstacle objects.

Object	E [Pa]	ν	Density ρ [kg/m^3]	NACC (J_p, β, ξ, ϕ)
Table	2×10^6	0.39	32	$(0.98, 0.50, 2.0, 45^\circ)$
Vase	2×10^4	0.39	2	$(0.98, 0.50, 2.0, 45^\circ)$
Ruins	8×10^6	0.39	600	$(0.98, 0.30, 2.0, 45^\circ)$
Wall	1×10^7	0.39	2000	$(0.96, 0.50, 3.0, 45^\circ)$

A.2 NodeBeam Vehicle Asset Specifications

Tab. 3 reports statistics for our two primary nodebeam vehicles parsed from community assets. Proxy particle mass denotes pershellproxy mass used during P2G projection. Individual beamlevel stiffness, damping, plastic and fracture thresholds are stored perbeam inside original asset files.

Table 3: Statistics of nodebeam vehicle assets.

Vehicle	Nodes	Beams	Collision Triangles
49 Ford	533	3018	392
Gavril	369	2100	185

Tab. 4 and Tab. 5 give beamparameter statistics for vehicle body beams. Derived beams for suspension, hydraulic dampers and tires are excluded from these statistics.

Table 4: Beam statistics for 49 Ford (2478 body beams).

Parameter	Min	Median	Max
Rest length L_0 [m]	0.0141	0.4684	1.7400
Stiffness k	1.272×10^6	1.656×10^6	3.0024×10^7
Damping d	30	37	68
Deformation threshold T_{def}	1500	9600	6.675×10^{10}
Break threshold T_{break}	4950	4.26×10^{10}	4.26×10^{10}
Plastic deformation coefficient c_{pl}	0	0	0.03

Table 5: Beam statistics for Gavril (1620 body beams).

Parameter	Min	Median	Max
Rest length L_0 [m]	0.0346	0.6269	1.6800
Stiffness k	68750	1.8×10^6	7.225×10^6
Damping d	110	253	1232
Deformation threshold T_{def}	1900	16625	3.8×10^6
Break threshold T_{break}	7150	7.15×10^8	1.29987×10^{11}
Plastic deformation coefficient c_{pl}	0	0.1	0.8

A.3 Detailed Reproduction Settings

Default hybrid simulation uses sparse background grid of resolution 256^3 within 8 m domain. Simulation timestep configurations:

- Ruin scenarios: $\Delta t = 10^{-4}$ s, total 20000 steps.
- Wall scenarios: $\Delta t = 10^{-4}$ s, total 2500 steps.
- Table / flowerpot scenarios: $\Delta t = 5 \times 10^{-5}$ s, total 5000 steps.

Rendered visual outputs adopt 1920×1080 resolution. All ablation studies modify exactly one component while keeping all other hyperparameters identical. Simulation kernels (nodebeam integration, P2G/G2P, splitfield CollisionMPM, shellproxy update, spatialhash antipenetration) are implemented on GPU within Warp (Macklin, 2022). Rendering pipeline splits workload: structural mesh rendered via Blender offscreen; MPM3DGS rasterization runs on separate GPU device; physicsaware depth compositing merges two image streams. Timing measurements exclude asset preprocessing, 3DGS interior appearance pretraining.

B Asset Construction Pipeline

Our pipeline accepts two families of input assets: MPM3DGS obstacle assets and nodebeam structural assets.

B.1 MPM3DGS Obstacle Asset Workflow

1. If only images are available, reconstruct glTF mesh via multiview reconstruction (e.g., TRELLIS).

2. Configure multiview camera trajectory around object, train surfaceonly 3DGS on glTF geometry.
3. For thinshell objects (vase, table): reuse surface Gaussian kernels directly as MPM surface particles.
4. For thick fractured objects (wall, fruit): perform volumetric particle filling inside bounding envelope.
5. Interior appearance completion following (Huang et al., 2026): generate multiview reference for hidden interior surfaces, propagate color/opacity/scale onto interior particles. Interior learning only produces rendering attributes; physical mass, constitutive parameters remain independent of appearance.
6. Cache final combined particle set (surface + interior) with paired 3DGS attributes for simulation reuse.

B.2 NodeBeam Asset Construction

Automatic generation for simple obstacles For simple topology objects such as walls, we generate nodebeam assets automatically from glTF mesh:

1. Bake worldspace transform into vertex positions, deduplicate spatiallycoincident vertices.
2. Perform quadric mesh simplification with facecount budget; reject configuration producing overly short beams and iteratively lower budget until valid mesh obtained.
3. Simplified mesh vertices become structural nodes; triangle edges become beam elements; triangles form collision geometry used for shellproxy sampling.
4. Distribute total object mass uniformly over nodes; assign global homogeneous beam stiffness/damping/fracture parameters.
5. Establish binding between physical nodes and render mesh vertices to drive mesh deformation during simulation.
6. Sanitycheck: parse generated asset again and verify consistency of node count, beam count, collision geometry and total mass before simulation.

B.3 Vehicle Preprocessing and Shell Proxy Generation

Given vehicle assets containing nodebeam topology and collision triangular meshes, we parse physical nodes with mass m_r, beam elements with perbeam stiffness, damping, plastic yield threshold and breaking strength, together with collision faces on the car body (Maul et al., 2021).

Shell proxy particles s are uniformly sampled over each collision triangle. Each proxy stores its parentnode set $T(s)$ and barycentric weights w_{sr} satisfying $\sum_{r\in T(s)} w_{sr} = 1$. **Proxy mass assignment: we distribute total vehicle mass onto shell proxies such that $\sum_s m_s^{\text{proxy}} = \sum_r m_r$, to preserve exact vehicleside mass for P2G projection.** Proxies maintain no independent dynamic state; at each timestep their position and velocity are interpolated from parent physical nodes via Eq. equation 8. Shell proxies serve exclusively as contactsampling primitives for P2G; no G2P update is applied to proxy particles.

Community vehicle assets Complex vehicles (49 Ford, Gavril, Magnum) originate from precalibrated Rigs of Rods community nodebeam assets. We parse node mass, beam topology, perbeam mechanical parameters and collision triangles. Render mesh vertices store offset relative to reference node triplets, guaranteeing consistent visual deformation under structural fracture. Future work aims to predict nodebeam topology and parameters directly from raw vehicle meshes.

C Supplementary Experiment Interpretations

C.1 Qualitative baseline & coupling ablation (Figure 5)

Row (a) Pure nodebeam baseline: Vehicle chassis deforms plausibly, yet obstacle fracture is strictly bounded by predefined beam topology, yielding blocky cracks without fine flying fragments. Row (b) FractureGS baseline: Obstacle produces abundant debris; however, converting the whole vehicle into MPM particles erases componentwise mechanical response, and the vehicle deforms as

a featureless particle lump. Row (c) Oneway coupling ablation: Vehicle motion is projected onto grid, but collision impulse is never fed back to structural nodes. Obstacle fractures normally, while the vehicle barely deforms under contact. Row (d) No shellproxy ablation: Without dense surface sampling from shell proxies, contact coverage is insufficient. An increased penetration burden is observed and the obstacle exhibits asymmetric, localized crack patterns. Row (e) Full FracFusion: Simultaneous vehicle componentwise structural deformation and obstacle continuum fragmentation, with mitigated interpenetration. Red boxes highlight key contact zones for each case.

C.2 PARTICLETRIANGLE ANTIPENETRATION ABLATION (FIGURE 6)

This ablation validates our particletriangle penalty correction for hidden interpenetration that gridlevel CollisionMPM cannot fully eliminate.

Visual comparison (left three subpanels): When the penalty module is disabled (*Penalty off*), MPM particles from fractured obstacles penetrate inside the thinshell vehicle mesh. This penetration remains visually hidden in early timesteps, termed *hidden penetration*. The middle panel marks this invisible overlapping region with red segments. When our particletriangle penalty is activated (*Penalty on*), outward velocity correction is applied to penetrating MPM particles, and equalmagnitude reaction impulse is scattered back to vehicle structural nodes. Obstacle geometry correctly separates from vehicle surfaces, removing persistent interpenetration artifacts.

Quantitative curve (right): The curve plots average penetration burden (mm per particle) across simulation timesteps. Red curve (penalty disabled) peaks at 7.47 mm/particle; blue curve (penalty enabled) peaks only at 2.72 mm/particle, corresponding to 63.6% reduction of penetration magnitude. Gridonly contact cannot fully resolve thinshell hidden penetration. Our lightweight penalty correction substantially lowers penetration error and preserves geometric consistency for highenergy impacts.

C.3 PHYSICSAWARE DEPTH COMPOSITING EVALUATION (FIGURE 7)

Figure 7 evaluates renderinggeometry consistency between native weighted3DGS rasterization and our physicsaware depth strategy under crashinduced fragmentation.

Visual panels (left four subplots): The leftmost thumbnail gives fullscene overview; orange box crops the vehiclewall collision area for zoomed inspection.

- *Weighted 3DGS*: Photorealistic rasterized output from standard weighted 3DGS.
- *Physics depth*: Depth map sampled directly from physical simulation, treated as geometric reference.
- *Changed decisions*: Magenta pixels mark positions where vanilla 3DGS rasterization produces wrong depth ordering relative to physical simulation. These pixels would suffer depthleakage artifacts without our compositing scheme.

Quantitative curve (right): The curve records pixelwise depthdisagreement rate (%) over simulation timesteps. The orange marker at step 500 marks initial vehicleobstacle contact. After contact transients, disagreement rate stabilizes near the 8.37% baseline (blue dashed line). Disagreement gradually rises in later timesteps when massive fragmentation rearranges Gaussian primitives. Our compositing strategy maintains good alignment between rendering output and physical geometry for most of the impact sequence.

C.4 PHYSICAL CONSERVATION VALIDATION (FIGURE 8)

Subplot A (Beam vehicle): Energy decomposition for nodebeam vehicle: kinetic (blue), elastic strain (orange), relative gravitational potential (green). Upon collision, vehicle kinetic energy drops rapidly, partially converted into elastic strain energy due to component deformation; gravitational potential energy barely changes.

Subplot B (MPM table): Energy curves for MPM obstacle. After contact onset, obstacle gains kinetic energy and accumulates moderate elastic strain energy. As fracture proceeds, stored strain energy releases and converts into kinetic energy of flying fragments.

Subplot C (Combined mechanical energy): Total mechanical energy of coupled system. Solid black: primary total energy (vehicle nodes + MPM obstacle particles). Dashed purple: extended total energy counting shellproxy particles. Total energy stays bounded through impact with no spurious energy explosion; shell proxies introduce mild, controllable numerical dissipation.

Subplot D (Momentum exchange): Momentum evolution of closed simulation system. Solid black line: total momentum, nearly constant. Red curve: vehicle momentum decreasing after collision. Blue curve: MPM obstacle momentum increasing synchronously. Momentum is bidirectionally exchanged between vehicle and obstacle, satisfying momentum conservation.

Bottom snapshots visualize timestamps $t = 0.000\,\mathrm{s}$ (precollision), $t = 0.075\,\mathrm{s}$ (initial contact), $t = 0.150\,\mathrm{s}$ (deep penetration with fracture onset), $t = 0.245\,\mathrm{s}$ (latestage impact with debris ejection). Snapshots visually correspond to the timeevolution of energymomentum curves above.

Overall, our bidirectional coupling exhibits stable physical conservation properties under highenergy impact and fragmentation without severe numerical instability.

D Additional Qualitative Results

D.1 Diverse VehicleObstacle Impact Scenarios

Fig. 9 collects supplementary simulation sequences covering different vehicle models and obstacle materials: 49 Ford against a flowerpot, vehicle against gate wall, and Magnum against a metal roadblock. Our unified hybrid pipeline handles thinshell bending, brittle fracture and metallike obstacle yielding without modification.

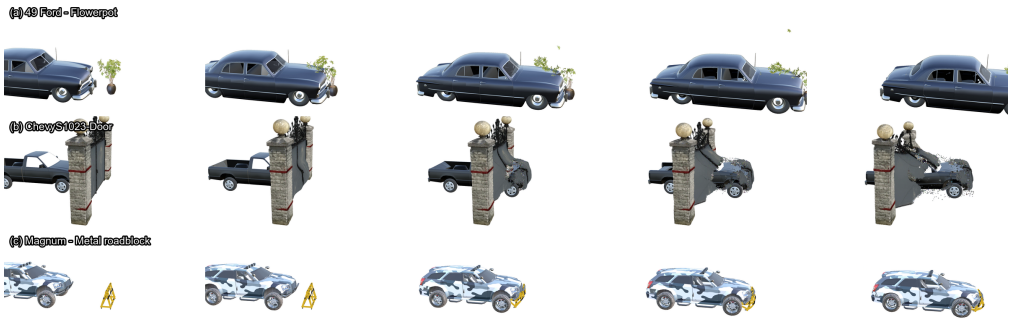


Figure 9: Supplementary impact sequences. Top: 49 Ford colliding with a flowerpot. Middle: vehicle hitting a gate wall. Bottom: Magnum against a metallic roadblock. Time increases from left to right.

D.2 Interior 3DGS Appearance for Fractured Soft Objects

Fig. 10 demonstrates interiorGaussian completion on fruitlike objects (watermelon, pineapple, dragonfruit). After fracture, newly exposed interior surfaces exhibit consistent appearance learned in preprocessing stage.



Figure 10: Fracture of soft volumetric objects with prelearned interior 3DGS appearance. From top to bottom: watermelon, pineapple, dragonfruit. First column: fullscene overview; remaining columns cropped to collision contact region.

E Full Quantitative Ablation Summary

We aggregate key quantitative metrics across all ablation variants in Tab. 6. Metrics include penetration burden F and average perstep wallclock time. "Oursw/oshell" denotes removing shell proxy particles; "Oursoneway" removes gridtovehicle impulse feedback while retaining vehicletogrid contact; "Oursw/oantipen" disables particletriangle antipenetration penalty; "Oursw/ophysdepth" uses native weighted 3DGS depth for compositing.

Table 6: Aggregated quantitative ablation metrics. Lower is better for F.

Variant	Penetration burden F [mm/particle]	Avg step time [ms]
Ours (full)	2.72	9.61
Oursoneway	3.63	8.56
Oursw/oshell	8.11	8.84
Oursw/oantipen	7.47	9.12
Oursw/ophysdepth	2.68	9.53

Observations:

1. Removing gridtovehicle impulse feedback raises penetration burden, consistent with a missing structural reaction rather than a depthcompositing change.
2. Removing shell proxies substantially raises penetration burden, confirming shell proxies improve surface contact sampling.
3. Antipenetration penalty reduces visible persistent penetration without large runtime overhead.
4. Replacing physicsaware depth with native weighted 3DGS depth leaves the penetration metric unaffected, but reintroduces the depthleakage artifacts discussed in the main paper; the occlusiondisagreement rate is defined for this variant only, since the other ablations do not alter depth compositing.

F Additional Implementation Notes on Warp GPU Kernels

All core simulation logic is implemented using NVIDIA Warp (Macklin, 2022), a CUDAbased Python framework for highperformance physics simulation. Key kernel responsibilities:

1. Shell proxy update kernel: gather parentnode positions/velocities, perform barycentric interpolation for every shell proxy at start of each timestep. No G2P write to shell proxy data array.
2. P2G kernel: scatter mass and momentum from shell proxies and MPM particles onto two independent splitfield background grid buffers.
3. Splitfield CollisionMPM kernel: iterate active grid cells only; compute normaldirection collision impulse where both fields have nonzero mass.
4. G2P scattering kernel: update MPM obstacle particles; scatter collision impulse from grid $\rightarrow$ shell proxy $\rightarrow$ physical nodebeam nodes via barycentric weights. Shell proxy array is readonly inside G2P kernel.
5. Nodebeam integration kernel: sum beaminternal forces plus collision external impulse, perform explicit nodal integration; evaluate beam plasticity and fracture condition, mark broken beam indices.
6. Spatialhash antipenetration kernel: rebuild hash tables from deformed collision triangles each timestep; perform particletriangle intersection test; compute velocity correction and backscatter reaction impulse onto structural nodes.

Memory layout remark: shellproxy arrays live only on GPU; data transfer to CPU is avoided during simulation loop to preserve throughput.

G Failure Modes and Limitation Analysis

We document observable failure modes of our hybrid framework for completeness, to clarify boundary of applicability.

1. **Severe tangential slip and shear penetration**: CollisionMPM handles normaldirection contact well. Under extreme large tangential shear, thin shell proxy layer may slide through obstacle particles. Antipenetration reduces but cannot fully eliminate this artifact. Very highshear crash scenarios may still exhibit transient visual penetration.
2. **Shell proxy sampling density tradeoff**: Sparser shell proxies accelerate simulation but degrade contact resolution; overly dense shell proxies increase P2G overhead. Our paper uses fixed pertriangle sampling count; adaptive sampling for highlydeformed regions remains future work.
3. **Nodebeam asset dependency**: Vehicle mechanical performance is fully inherited from input nodebeam assets. If source asset contains unrealistic beam stiffness or fracture thresholds, simulation outputs will propagate these biases. Our method does not calibrate beam parameters from realworld crash measurement data.
4. **3DGS interior appearance limitation**: Interior Gaussian appearance is imagebased completion, not physically learned material property. Under very large shear fracture, interior Gaussian texture may show implausible visual discontinuity.
5. **No twoway appearance feedback onto structural mesh**: Structural vehicle mesh deformation purely driven by nodebeam solver; we do not simulate meshlevel tearing or surface cracking on vehicle body geometry, only beamlevel structural disconnection.

These limitations suggest promising future directions: adaptive shellproxy resampling, joint calibration from real crash data, and meshsurface damage modeling for nodebeam structures.