

1. Supplementary Material

1.1. Core Components Review

In this section, we review the core concepts of our system: the 3D Gaussian Splatting (3DGS) representation, the Material Point Method (MPM), and the Genesis simulation framework.

1.1.1. 3D Gaussian Splatting (3DGS)

3DGS [12] reconstructs a scene from multi-view images using a set of anisotropic Gaussian primitives. Each Gaussian is parameterized by a mean μ , scaling coefficients s , rotation matrix R , and opacity α . The covariance matrix is:

$$\Sigma = R S S^T R^T, \quad \text{where } S = \text{diag}(s). \quad (6)$$

Appearance is represented using spherical harmonics (SH) coefficients for view-dependent color. During rendering, Gaussians are projected, depth-sorted, and blended using α -compositing.

1.1.2. Material Point Method (MPM)

The **Material Point Method** (MPM) [7, 34] is a hybrid Lagrangian–Eulerian solver for continuum dynamics. Materials are discretized into particles that store mass, velocity, deformation gradient, and stress. These particles transfer data to a background grid for solving the equations of motion, then interpolate updated properties back to particles. Recent works [9, 15, 41, 44] have applied MPM to 3DGS by treating Gaussians as particles whose covariance evolves with the deformation gradient.

1.1.3. Genesis Simulation Framework

Genesis [1] is a GPU-accelerated physics engine supporting multiple solvers and primitives, making it suitable for hybrid simulation. Key features include:

- **Multi-solver support:** MPM, Smoothed Particle Hydrodynamics (SPH), Position-Based Dynamics (PBD), FEM, rigid bodies, and robotics.
- **Geometry support:** Analytic primitives (spheres, cylinders, planes) and polygonal meshes.
- **High performance:** Modular C++ codebase with GPU kernels for scalability.

1.2. 3DGS Acquisition Details for individual objects

In this section, we provide additional details on the acquisition and post-processing of 3D Gaussian Splatting (3DGS) models for individual objects.

1.2.1. 3DGS training

All 3DGS models in this work were trained using the `nerfstudio` framework [35], which offers an end-to-end pipeline for both camera pose estimation and model optimization. The process consists of two main stages: (i)

camera pose recovery and sparse point cloud reconstruction with COLMAP via `ns-process-data` images, and (ii) 3DGS model training using the `splatfacto` configuration.

COLMAP parameters. Table 1 summarizes the parameters used for pose estimation. In practice, COLMAP successfully recovered camera poses for approximately 90% of the images; the remaining views were discarded due to failed estimation.

| Parameter | Value |
|------------------------|-----------------|
| camera-type | perspective |
| matching-method | vocab_tree |
| sfm-tool | any |
| refine-pixsfm | False |
| refine-intrinsics | True |
| feature-type | any |
| matcher-type | any |
| num-downscales | 3 |
| same-dimensions | True |
| use-single-camera-mode | True |
| gpu | True |
| use-sfm-depth | False |
| crop-factor | 0.0 0.0 0.0 0.0 |
| percent-radius-crop | 1.0 |

Table 1. COLMAP parameters (via `ns-process-data` images).

3DGS training parameters. All models were trained with `ns-train splatfacto`. Table 2 lists the default hyperparameters for this configuration, which correspond to the unmodified “vanilla 3DGS” baseline used in Section 5.1 for comparison.

The setup used for all final experiments presented in the paper—denoted as “Increased Gaussian budget” in Section 5.1—we modified two parameters:

1. `densify_grad_thresh` was reduced from 0.0008 to 0.0002.
2. `cull_alpha_thresh` was reduced from 0.1 to 0.005.

As discussed in Section 3.1.1, these changes encourage a denser allocation of Gaussians on the target object, resulting in more stable reconstructions. Figure 1 provides a qualitative comparison.

1.2.2. Filling methods

Figure 2 provides additional examples comparing different filling methods against the baseline model. Figure 3 illustrates the temporal evolution of a simulation for a non-filled object versus PhysGaussian filling and our proposed filling method.

| Parameter | Value |
|------------------------------|--|
| Splatfacto Model Config: | |
| warmup_length | 500 |
| refine_every | 100 |
| resolution_schedule | 3000 |
| background_color | random |
| num_downscales | 2 |
| cull_alpha_thresh | 0.1 |
| cull_scale_thresh | 0.5 |
| reset_alpha_every | 30 |
| densify_grad_thresh | 0.0008 |
| use_absgrad | True |
| densify_size_thresh | 0.01 |
| n_split_samples | 2 |
| sh_degree_interval | 1000 |
| cull_screen_size | 0.15 |
| split_screen_size | 0.05 |
| stop_screen_size_at | 4000 |
| random_init | False |
| num_random | 50000 |
| random_scale | 10 |
| ssim_lambda | 0.2 |
| stop_split_at | 15000 |
| sh_degree | 3 |
| use_scale_regularization | False |
| max_gauss_ratio | 10.0 |
| output_depth_during_training | False |
| rasterize_mode | classic |
| camera_optimizer | CameraOptimizerConfig(mode="off") |
| use_bilateral_grid | False |
| grid_shape | (16, 16, 8) |
| color_corrected_metrics | False |
| strategy | default |
| max_opacity_reg | 0.01 |
| mcmc_scale_reg | 0.01 |
| Trainer Config: | |
| max_num_iterations | 30000 |
| steps_per_save | 2000 |
| eval_single_image_every | 100 |
| eval_all_images_every | 1000 |
| mixed_precision | False |
| use_grad_scaler | False |
| save_only_latest_checkpoint | True |
| Optimizers parameters: | |
| Optimizer LRs | means=0.00016; features-dc=0.0025; features-rest=0.000125; opacities=0.05; scales=0.005; quaternions=0.001; camera=0.0001; bilateral-grid=0.005 |

Table 2. Default splatfacto training parameters.

1.3. Hole Filling for 3DGS Environments: densification algorithm

As discussed in Section 3.1.2, we introduce a dedicated algorithm designed to address two key challenges in large-scale 3DGS reconstructions: the presence of holes and the occurrence of sparsely populated regions. Our method identifies such regions and applies targeted densification strategies to improve surface continuity and simulation robustness. The complete procedure is presented in Algorithm 1.

1.4. Material Parameters of Evaluation Objects

In this section, we present a comprehensive description of the parameters employed to configure the simulation solvers across the different physical domains and object types illustrated in our experiments. These parameters were carefully

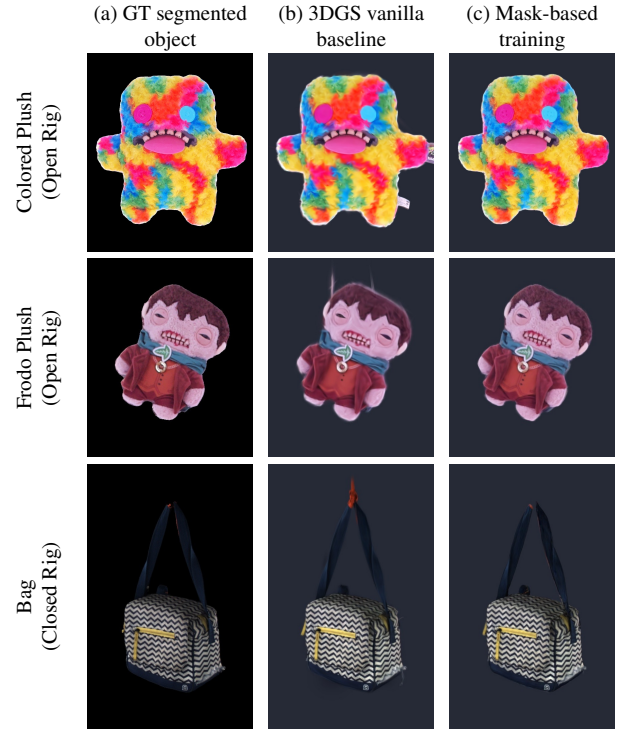


Figure 1. Comparative reconstruction results for three objects. Columns: (a) ground-truth segmented object; (b) vanilla 3DGS reconstruction with manual segmentation; (c) 3DGS trained on masked images. Rows (vertical labels) indicate the object name and rig setup. Our method reduces floating artifacts and produces more accurate geometry, resulting in sharper and more faithful renderings.

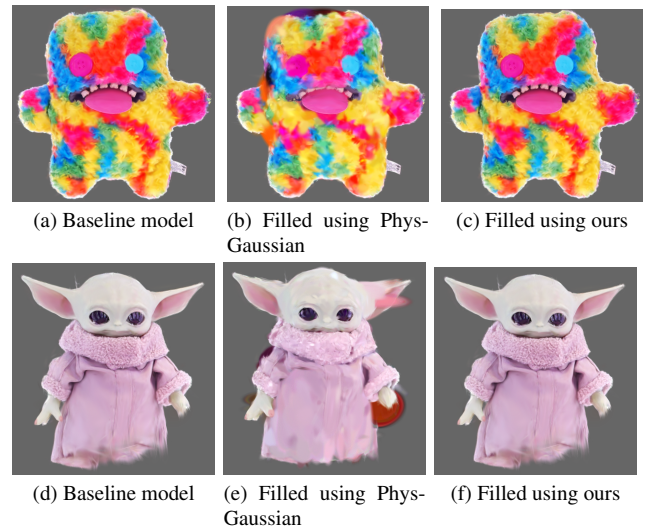


Figure 2. Comparative filling results. Each row corresponds to a different object, while the columns show: (a,d) the baseline empty model, (b,e) the model filled using PhysGaussian, and (c,f) the model filled using our method. Notice that PhysGaussian introduces visible external artifacts, whereas our approach produces results that fill the inner space of the object while maintaining the external appearance.

| Object | Density (kg/m ²) | Static friction (-) | Kinetic friction (-) | Stretch compliance (m/N) | Bending compliance (rad/N) | Stretch relaxation (-) | Bending relaxation (-) |
|-----------------|---------------------------------|---------------------------|----------------------------|--------------------------------|----------------------------------|------------------------------|------------------------------|
| Silk-like cloth | 1.5 | 0.05 | 0.03 | 1×10^{-6} | 1×10^{-4} | 0.05 | 0.02 |

Table 3. PBD cloth parameters.

Algorithm 1 Hole Filling / Densification for 3DGS Environments

Require: 3DGS export $\mathcal{G} = \{(\mu_i, \Sigma_i)\}$; target spacing h ; Poisson depth; empty-space threshold τ_d ; adaptive params $(\alpha, r_0, \gamma, p_{\min})$; dense-region pruning params (NN cutoff δ or voxel size v and max occupancy m).

Ensure: Densified point set \mathcal{D} ; augmented 3DGS.

```

1:  $P \leftarrow \{\mu_i\}$ 
2: Estimate normals on  $P$  (kNN)
3:  $\mathcal{M} \leftarrow \text{SCREENEDPOISSON}(P)$ 
4:  $\mathcal{M} \leftarrow \text{CROPTOAABB}(\mathcal{M}, P)$  // optional density
   pruning before crop
5: if  $\alpha > 1$  then
6:    $N_{\text{base}} \leftarrow \text{area}(\mathcal{M}) / ((\sqrt{3}/2) h^2)$ 
7:    $N_{\text{os}} \leftarrow \lfloor \alpha \cdot N_{\text{base}} \rfloor$ 
8:    $\mathcal{C} \leftarrow \text{POISSONDISKSAMPLEN}(\mathcal{M}, N_{\text{os}})$ 
9: else
10:   $\mathcal{C} \leftarrow \text{POISSONDISKSAMPLEH}(\mathcal{M}, h)$ 
11: for all  $x \in \mathcal{C}$  do
12:   $d(x) \leftarrow \text{NEARESTNEIGHBORDISTANCE}(x, P)$ 
13:   $\mathcal{C} \leftarrow \{x \in \mathcal{C} \mid d(x) \leq \tau_d\}$  // empty-space guard
14: if adaptive thinning then
15:   for all  $x \in \mathcal{C}$  do
16:     $p(x) \leftarrow \max(p_{\min}, \min(1, (d(x)/r_0)^\gamma))$ 
17:    Keep  $x$  with probability  $p(x)$ 
18: if dense-region pruning then
19:   if strategy = NN then
20:     $\mathcal{C} \leftarrow \{x \in \mathcal{C} \mid d(x) \geq \delta\}$ 
21:   else
22:    Build occupancy grid on  $P$  with voxel size  $v$ 
23:     $\mathcal{C} \leftarrow \{x \in \mathcal{C} \mid n(\text{vox}(x)) < m\}$ 
24:  $\mathcal{D} \leftarrow \mathcal{C}$ 
25:  $\text{AUGMENT3DGSWITHTRANSPARENTGAUSSIANS}(\mathcal{D})$ 
   return  $\mathcal{D}$  and augmented 3DGS

```

selected through manual tuning to achieve simulations that closely resemble real-world behavior. All reported values are expressed in SI units for consistency and reproducibility.

Table 3 reports the PBD solver parameters for the cloth simulations presented in Section 5.2.2 and further illustrated in Supplemental Section 1.6.

Table 4 lists the MPM elastoplastic material parameters for the 3DGS objects and environments used in the experiments.

| Object | Young's modulus (Pa) | Poisson's ratio (-) | Density (kg/m ³) | Yield stress (Pa) |
|--------------|-------------------------|------------------------|---------------------------------|----------------------|
| Can | 2×10^6 | 0.22 | 30 | 3×10^4 |
| Plushies | 0.25×10^4 | 0.09 | 50 | 1×10^4 |
| Lunch bag | 1×10^4 | 0.20 | 120 | 5×10^3 |
| Car | 2×10^6 | 0.22 | 30 | 3×10^4 |
| Environments | 5×10^{10} | 0.49 | 2.6×10^4 | 1×10^8 |

Table 4. MPM elastoplastic material parameters.

Table 5 summarizes the rigid body solver parameters for the meshes presented in Section 5.2.1 and further illustrated in Supplemental Section 1.5.

| Object | Density (kg/m ³) | Coupling friction (-) | Coupling softness (-) | Coupling restitution (-) |
|------------|---------------------------------|-----------------------------|-----------------------------|--------------------------------|
| Brown Rock | 5000 | 3.0 | 0.002 | 0.0 |
| Gray Rock | 2500 | 3.0 | 0.002 | 0.0 |
| Log | 2500 | 3.0 | 0.002 | 0.0 |

Table 5. Rigid bodies parameters.

Finally, Table 6 provides the SPH solver parameters for the mudslide simulations presented in Sections 5.2.3 and 5.2.5, and further illustrated in Supplemental Section 1.7.

| Object | Density (kg/m ³) | Stiffness (Pa) | Viscosity (Pa s) | Surface tension (N/m) |
|----------|---------------------------------|-------------------|---------------------|--------------------------|
| Mudslide | 1000 | 60 000 | 0.1 | 0.01 |

Table 6. SPH fluid parameters.

1.5. More results on 3DGS-rigid mesh interaction

Figure 4 shows more examples of the interaction 3DGS and meshes.

1.6. More results on 3DGS-cloth interaction

Figure 5 shows more examples of the interaction of 3DGS and cloths.

1.7. More results on 3DGS-fluid interaction

Figure 6 shows more experiments on the interaction of 3DGS and fluids

1.8. More results on complex use cases

Figure 7 shows more examples of complex use cases.

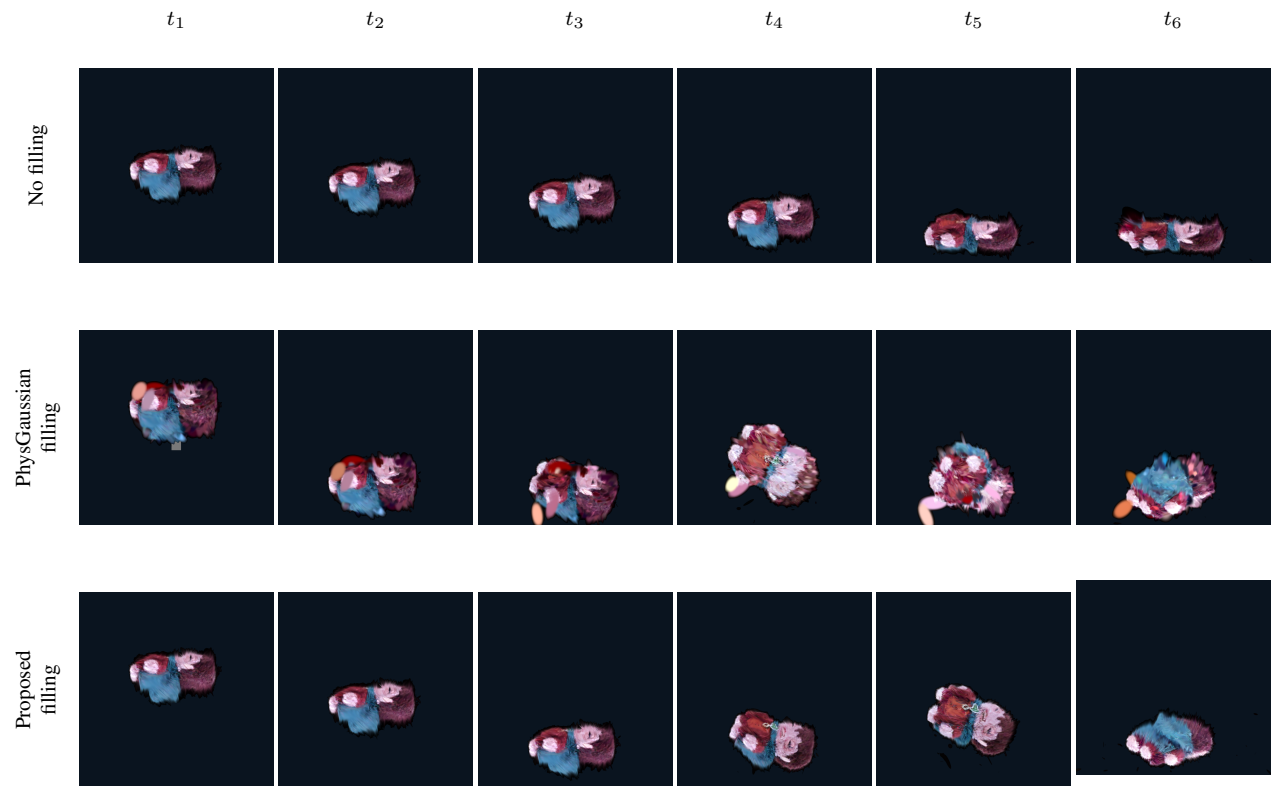


Figure 3. Temporal comparison during an object drop/deformation across filling strategies. Columns show consecutive timesteps (t_1 – t_6) of the object falling to the ground and bouncing. Rows: No filling; PhysGaussian filling; Proposed filling. The proposed method preserves visual quality and avoids protrusions while giving the simulation physical plausibility.



Figure 4. Temporal comparison of mesh-3DGS interactions (transposed layout). Columns correspond to timesteps (t_1 - t_3). Rows correspond to distinct interaction scenarios (rock or log impacting different 3DGS objects). It can be seen that this interaction results in physically plausible simulations.



Figure 5. Sampled frames from the cloth PDB solver and 3DGS interaction at three timesteps (t_1 – t_3). Columns (a)–(c) show three 3DGS models for each instant.

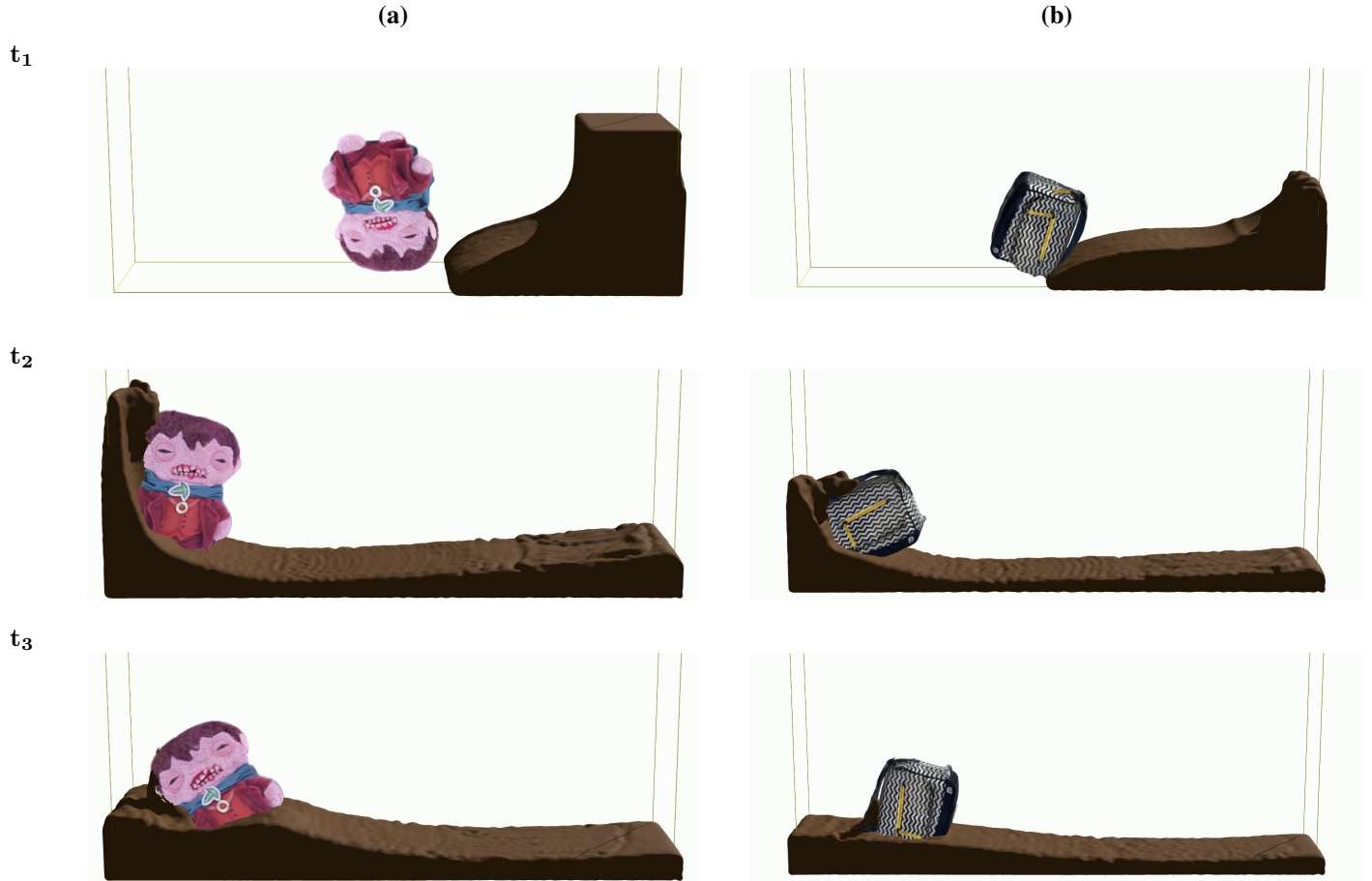


Figure 6. Sampled frames from the liquid and 3DGS interaction simulation. Rows show three manually selected timesteps to better capture the dynamics of the simulation(t_1-t_3). Columns depict two different real world objects modeled as a 3DGS.

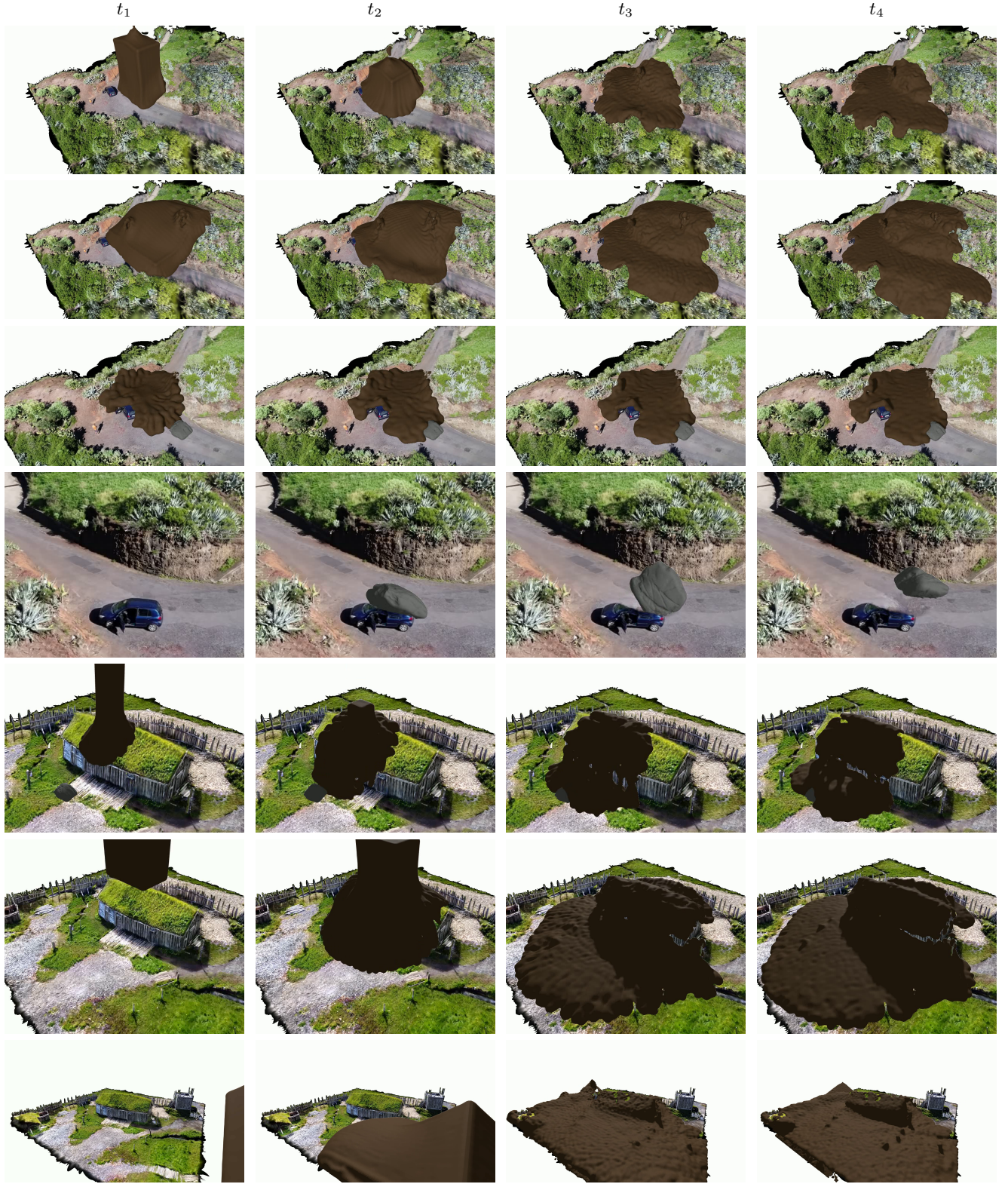


Figure 7. Different experiments of terrain–liquid interaction across time. The first four rows correspond to a road surrounded by vegetation, while the last three rows show a flat terrain with a house in the middle. Columns represent four timesteps (t_1 – t_4). Each row illustrates an example where a simulated liquid, modeled with an SPH solver, interacts with one of the two captured terrains. The terrains are real-world large scales environments, and they were reconstructed using 3DGS and further densified with our densification algorithm 3.1.2. In addition, Row 4 includes the interaction with a rigid mesh rock: when dropped onto the car, the impact causes the car to deform. Row 3 further demonstrates a three-way interaction, where liquid, 3DGS terrain, and mesh rock interact simultaneously.