Supplementary Material: Fragility Modeling of Precariously Balanced Rocks: Calibration, Benchmarking, and Sensitivity

Zhiang Chen * [1], **Akshay Sopan Mahalle**[1], **M. Khalid Saifullah**[2], **Christine Wittich**[3], **Jnaneshwar Das**[1], **Christopher Madugo**[4], **Albert Kottke**[4], **Ramón Arrowsmith**[1]

[1]Arizona State University, Tempe, Arizona, USA, [2]Kinemetrics, Pasadena, California, USA, [3]University of Nebraska–Lincoln, Lincoln, Nebraska, USA, [4]Pacific Gas and Electric Company, California, USA

S1 Technical Implementation Details

S1.1 System Architecture

The VSR2 platform organizes two custom ROS2 nodes developed in this study that operate in coordination: the Simulation Node, which manages the physical environment and model dynamics using the PyBullet engine, and the Control Node, which governs test logic and input trajectory execution. These two components are loosely coupled through ROS2's native service and action interfaces, which allow flexible experiment definition and execution.

The Simulation Node is responsible for instantiating the virtual environment within the PyBullet physics engine. Upon initialization, the node constructs a complete simulation scene, consisting of a static base, a horizontally actuated pedestal, and a PBR positioned atop the pedestal. These entities are modeled as rigid bodies and configured based on user-specified parameters supplied via YAML configuration files. The parameter configuration includes the selection of pedestal and PBR types, as well as their associated geometric and physical properties, which are described in Section S1.2. The pedestal is connected to the base via a single prismatic joint that allows translation in the horizontal x-axis. This joint is actuated using control commands from the Control Node.

The Control Node is responsible for receiving high-level ground motion commands and generating corresponding trajectories for the pedestal motion. VSR2 supports four types of ground motions: single cosine displacement, grid cosine displacements, single earthquake displacement recording, and all earthquake displacement recordings, which are described in Section S1.3. Based on a high-level ground motion command, the Control Node generates a trajectory and publishes it to the Simulation Node using a ROS2 Action interface. The action goal message includes the list of desired positions, velocities when applicable, and their corresponding timestamps. Upon receiving this input, the Simulation Node executes the motion by applying the desired setpoints to the pedestal's prismatic joint. At each simulation timestep, the Simulation Node records both the commanded and actual motions of the pedestal, as well as the dynamic response of the PBR. The PBR position is tracked at its center of mass in world coordinates, and its orientation is represented by Euler angles (roll, pitch, yaw) derived from the body's rotation matrix relative to the fixed world frame (x–y–z). The system stores these logs for subsequent analysis, such as the detection of overturning events and the evaluation of complex rocking behavior.

To ensure repeatability and consistency across simulation trials, the Simulation Node exposes a custom ROS2 service, manage_model, which manages the lifecycle of the PBR model. This service handles spawning and resetting the PBR and is called by the Control Node before and after each test. During a spawn action, the Simulation Node loads the predefined rock model into the environment using either a URDF file or a collision mesh, placing it at a user-defined location on the pedestal surface. A reset action returns the rock to its initial pose and clears all accumulated velocity, ensuring mechanical neutrality at the start of each run.

S1.2 Physical Modeling of Pedestal and Rock

VSR2 allows for flexible specification of the physical and geometric properties of both the pedestal and the rock, accommodating a wide range of experimental scenarios. All physical properties, such as mass, moment of inertia,

*Corresponding author: zch@asu.edu

contact friction, restitution, and surface stiffness, are defined in YAML configuration files. This design separates model setup from simulation execution and enables systematic variation of physical parameters without modifying the underlying codebase.

The pedestal may be defined as either a simple prism or as a triangular mesh derived from observed 3D topographic data. For idealized tests, the pedestal geometry is defined as a prism with explicit dimensions and uniform density. In mesh-based simulations, a 3D file, such as an OBJ file, is supplied and scaled appropriately. The rock is modeled as a separate rigid body placed freely atop the pedestal. Like the pedestal, the rock can be defined either as a box or a detailed mesh model. The mesh version is typically sourced from lidar-scanned or photogrammetric models of field rocks or lab analogs. When using a mesh, the geometry must be watertight to ensure correct computation of inertial properties. The principal moments of inertia are critical for capturing accurate rock behavior. Our GitHub repository (https://github.com/ZhiangChen/virtual_shake_robot_pybullet) provides a Python script to compute the inertia matrix using trimesh.

The physical interaction between the rock and pedestal is defined by a set of contact parameters, including friction, restitution, spinning friction, contact stiffness, and contact damping. Parameters such as lateral friction and restitution correspond to measurable physical properties and can be experimentally determined through sliding or impact tests. Other parameters, such as spinning friction, contact stiffness, and contact damping, are implemented in PyBullet as numerical models for torsional resistance, material rigidity, and energy dissipation rather than directly measurable physical properties. For example, spinning friction represents torsional resistance at the contact patch, while stiffness and damping arise from the spring-damper formulation used to approximate deformation and energy loss during contact. Together, these parameters capture both the measurable and numerically approximated aspects of rock-pedestal interactions, enabling simulations that reproduce complex rock behavior. VSR2 simulations employ a rigid-body assumption, meaning that no deformation, fatigue, or fracture occurs within either the rock or the pedestal.

VSR2 allows users to modify parameters for the physics-engine environment in a YAML file. These parameters are inherited from PyBullet, and detailed descriptions of their functions and configuration can be found in the PyBullet Quickstart Guide (Coumans and Bai, 2023). For the experiments in this study, we used a timestep of 0.001 s (1 kHz integration), with 150 solver iterations applied at every timestep. Contacts are detected using convex-convex collision detection with support for contact margins. PyBullet computes contact points, normals, and approximate penetration depth via iterative methods, then resolves these constraints through an impulse-based solver. The maximum penetration depth correction was controlled by the split impulse penetration threshold using the default value of -0.04 m in our setup, which prevents unrealistically deep interpenetrations while maintaining energy conservation. These settings, along with the use of cone friction and deterministic pair management, provided consistent and stable resolution of contact forces across the large simulation ensemble.

S1.3 Control Strategies and Motion Modes

The VSR2 platform supports four motion modes: (i) single cosine, (ii) grid cosine, (iii) single earthquake displacement recording, and (iv) all earthquake displacement recordings. These motion modes represent high-level ground-motion commands. Based on the selected motion model, specified via the motion_mode configuration parameter, the Control Node generates a corresponding trajectory and publishes it to the Simulation Node using a ROS2 Action interface. The Simulation Node then applies this trajectory to the prismatic joint, controlling the horizontal displacement of the pedestal.

The single cosine mode generates a single-pulse cosine displacement trajectory based on user-defined amplitude and frequency. The trajectory is defined as $x(t) = A\,(1 - \cos(2\pi Ft))$, where A is the displacement amplitude and F is the frequency. The trajectory is discretized at 1 kHz to ensure smooth execution. This motion starts and ends with zero velocity, avoiding impulsive accelerations and ensuring the system returns to rest. The single cosine mode is particularly well suited for benchmarking rock behavior under controlled loading conditions.

The grid cosine mode extends the single cosine mode by systematically applying a series of cosine motions over a grid of ground motion parameters, specifically PGA and PGV/PGA ratios. These two intensity measures are widely used in PBR fragility studies: PGA characterizes shaking amplitude, and the PGV/PGA ratio reflects shaking-duration effects (Purvance et al., 2008). To construct the ground motion grid, users specify the ranges and the uniform spacing for PGA and PGV/PGA. The grid cosine mode then uniformly slices each interval based on the chosen spacing, generating a grid of paired (PGA, PGV/PGA) values. For each combination of PGA and PGV/PGA, the Control Node computes the corresponding amplitude and frequency, then invokes the single cosine mode to generate the motion trajectory. To ensure repeatability and consistency across experiments, the grid cosine mode also uses the manage_model service to reset the PBR before each motion trial. This procedure ensures the same initial conditions for the PBR, including position and orientation, across all runs. The grid mode enables a sweep over combinations of PGA and PGV/PGA to produce an overturning response diagram for assessing the fragility of a specific object.

In the single earthquake displacement recording and all earthquake displacement recordings modes, VSR2 generates ground motions from real-world earthquake data. Before executing either mode, users must provide a set of earthquake displacement seismograms, which the Control Node loads into memory. In the single earthquake displacement recording mode, users configure the motion_mode parameter and specify the index of the desired earthquake record. The Control Node then retrieves the corresponding ground motion displacement and publishes the resulting trajectory to the Simulation Node. In the all earthquake displacement recordings mode, the Control Node sequentially processes and executes all available earthquake records. For both modes, the Control Node invokes the manage_model service to reset the PBR to its initial position and orientation before each motion trial.

At the low level, VSR2 uses a closed-loop feedback controller on the prismatic joint to track the reference displacement trajectory produced by the Control Node. The Control Node samples the reference trajectory at 1 kHz. For the two cosine motion modes, the controller follows both position and velocity references from the analytic trajectory to achieve smooth starts and stops. For the two earthquake-recording motion modes, the controller tracks position reference only to avoid amplifying high-frequency noise while preserving the recorded displacement history. Fig. S1 illustrates the controller's performance for both a cosine input and a real displacement seismogram. This control method enables the pedestal to accurately and precisely track the desired ground motion trajectory, allowing reproduction of input ground motions in simulation and supporting stable and repeatable analysis of overturning responses.

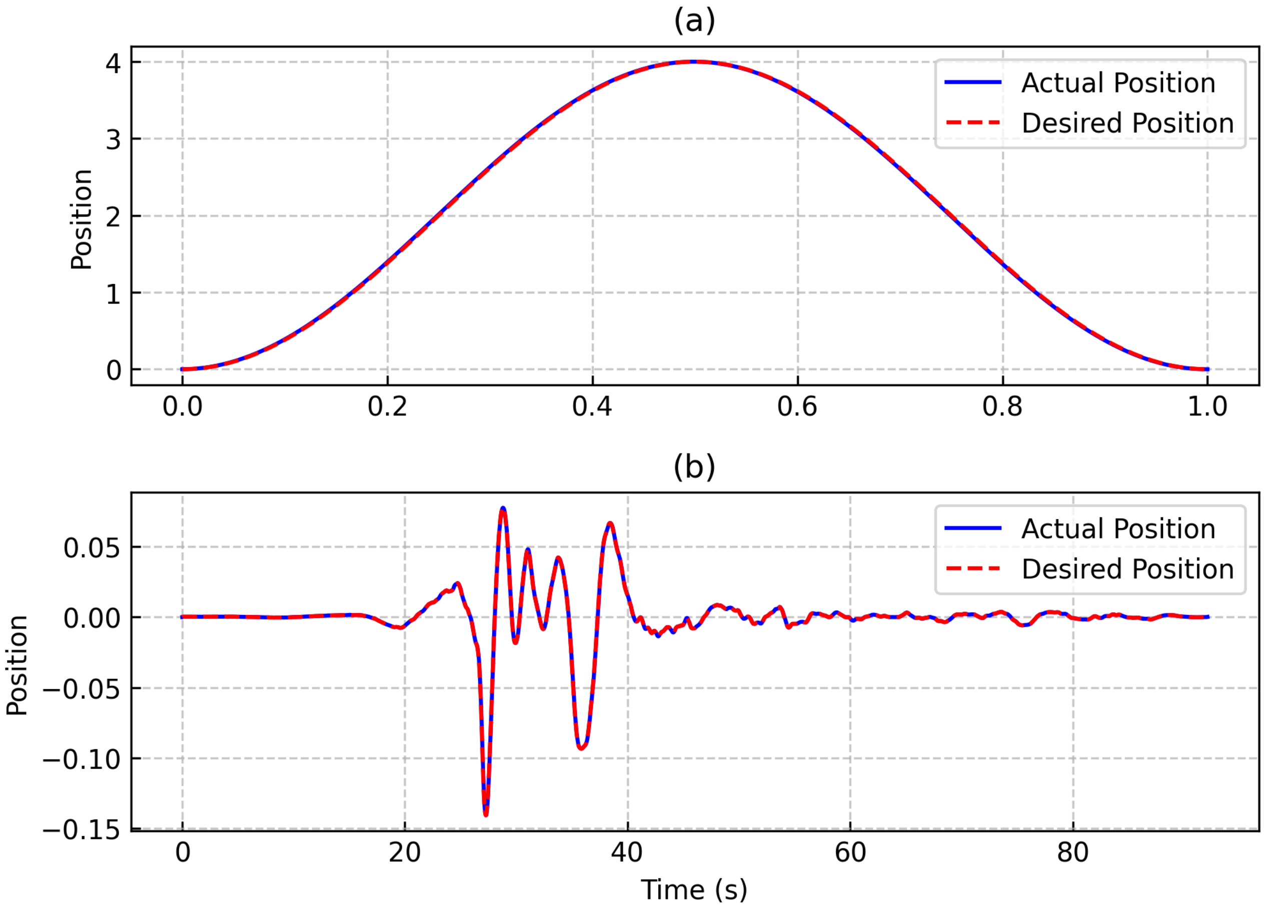

Figure S1 Control performance of VSR2. Comparison between actual and desired pedestal positions under (a) a cosine ground motion and (b) a real earthquake displacement recording. The close alignment illustrates the accuracy of the velocity controller in tracking the input trajectory.

S2 Contact Modeling in PyBullet

S2.1 Body-Level Inputs and Effective Contact-Pair Values

PyBullet contact parameters are specified at the body level, but contact constraints are resolved using effective pairwise values computed from the two contacting bodies. Let superscripts r and p denote the PBR and pedestal body-level inputs, respectively. Under the default Bullet material-combiner rules, the effective rock–pedestal contact parameters are

$$\begin{aligned}
\mu_{\text{lat}}^{rp} &= \mu_{\text{lat}}^{r}\mu_{\text{lat}}^{p}, \\
e^{rp} &= e^{r}e^{p}, \\
\mu_{\text{spin}}^{rp} &= \mu_{\text{spin}}^{r}\mu_{\text{lat}}^{p} + \mu_{\text{spin}}^{p}\mu_{\text{lat}}^{r}, \\
c^{rp} &= c^{r} + c^{p}, \\
k^{rp} &= \left(\frac{1}{k^{r}} + \frac{1}{k^{p}}\right)^{-1},
\end{aligned} \tag{1}$$

where μ_{lat} is lateral friction, e is restitution, μ_{spin} is spinning friction, c is contact damping, and k is contact stiffness. Bullet also internally bounds friction-like combined values to avoid unrealistically large constraint forces. In this study, no custom material-combiner callback was used.

Because the same body-level values were assigned to the PBR and pedestal in all VSR2 simulations, Eq. 1 reduces to

$$\begin{aligned}
\mu_{\text{lat}}^{rp} &= \mu_{\text{lat}}^{2}, \\
e^{rp} &= e^{2}, \\
\mu_{\text{spin}}^{rp} &= 2\mu_{\text{spin}}\mu_{\text{lat}}, \\
c^{rp} &= 2c, \\
k^{rp} &= \frac{k}{2}.
\end{aligned} \tag{2}$$

Thus, the optimized contact parameters reported in the main text are body-level PyBullet inputs, whereas the physical contact response is governed by the corresponding effective rock–pedestal pair values.

Table S1 Optimized body-level PyBullet contact inputs and corresponding effective rock–pedestal contact-pair values. The body-level inputs were assigned identically to both the PBR and pedestal.

Parameter	**Body-level input**	**Effective pair value**
Restitution, e	0.35	0.12
Lateral friction, μ_{lat}	0.7	0.49
Spinning friction, μ_{spin}	0.5	0.70
Contact damping, c	1.0×10^{7}	2.0×10^{7}
Contact stiffness, k	2.6×10^{9}	1.3×10^{9}

S2.2 Simplified Contact-Solver Interpretation

PyBullet uses the Bullet physics engine as its computational core and resolves contact response through a constraint-based solver that considers lateral friction, spinning friction, restitution, contact damping, and contact stiffness within the same contact-resolution step (Coumans, 2015). The formulation presented below is intentionally simplified to highlight the primary roles of the contact parameters; the actual Bullet implementation is more complex and includes additional details in the assembly and solution of the contact constraints. Let the pre-impact normal relative velocity at the contact point be

$$u_n^- = \mathbf{n} \cdot \mathbf{u}_{rel},$$

where $\mathbf{n}$ is the unit contact normal and $\mathbf{u}_{rel}$ is the relative velocity of body A with respect to body B at the contact point,

$$\mathbf{u}_{rel} = (\mathbf{v}_A + \boldsymbol{\omega}_A \times \mathbf{r}_A) - (\mathbf{v}_B + \boldsymbol{\omega}_B \times \mathbf{r}_B).$$

$\mathbf{v}_A$ and $\mathbf{v}_B$ are the linear velocities of bodies A and B, $\boldsymbol{\omega}_A$ and $\boldsymbol{\omega}_B$ are their angular velocities, and $\mathbf{r}_A$ and $\mathbf{r}_B$ are the vectors from the corresponding centers of mass to the contact point.

Restitution enters the solver through a restitution-induced target normal rebound term,

$$u_{n,\text{rest}} = \begin{cases} 0, & |u_n^-| < v_{th}, \\ -e\,u_n^-, & |u_n^-| \geq v_{th}, \end{cases}$$

where e is the coefficient of restitution and v_{th} is the restitution velocity threshold below which Bullet suppresses bounce. Both e and v_{th} are user-defined. Restitution controls the rebound component of the impact response: larger e produces a larger target rebound, while $e = 0$ corresponds to a fully inelastic normal response.

Contact stiffness and contact damping are treated differently. Rather than being applied directly as contact impulses, Bullet maps them into two solver parameters,

$$\text{cfm} = \frac{1}{\Delta t\, k + c}, \qquad \text{erp} = \frac{\Delta t\, k}{\Delta t\, k + c},$$

where k is the combined contact stiffness, c is the combined contact damping, Δt is the simulation time step, cfm is the constraint force mixing term, and erp is the error reduction parameter. The combined contact stiffness and damping are computed from the user-defined contact stiffness and damping values of the two contacting objects as

$$k = \left(\frac{1}{k_A} + \frac{1}{k_B}\right)^{-1}$$

and

$$c = c_A + c_B,$$

where k_A and k_B denote the user-defined contact stiffness values of objects A and B, respectively, and c_A and c_B denote their user-defined contact damping values. Physically, k controls the effective hardness of the contact, with larger k producing a less compliant, more rigid response. The parameter c controls effective dissipation during contact resolution, with larger c producing a more damped contact response. The quantity cfm acts as a compliance term in the contact constraint, whereas erp controls the fraction of positional contact error corrected during each time step.

A useful simplified expression for the normal contact impulse is

$$\lambda_n \approx \frac{(u_{n,\mathrm{rest}} - u_n^-) + \mathrm{erp}(-p/\Delta t)}{K_n + \mathrm{cfm}},$$

where λ_n is the normal contact impulse, p is the penetration depth, and K_n is the effective inverse mass term in the contact-normal direction,

$$K_n = m_A^{-1} + m_B^{-1} + \mathbf{n} \cdot \left[\mathbf{I}_A^{-1}(\mathbf{r}_A \times \mathbf{n}) \times \mathbf{r}_A\right] + \mathbf{n} \cdot \left[\mathbf{I}_B^{-1}(\mathbf{r}_B \times \mathbf{n}) \times \mathbf{r}_B\right]$$

where m_A and m_B are the masses of bodies A and B, $\mathbf{I}_A$ and $\mathbf{I}_B$ are their inertia tensors, $\mathbf{r}_A$ and $\mathbf{r}_B$ are the vectors from the corresponding centers of mass to the contact point, and $\mathbf{n}$ is the unit contact normal. This expression for λ_n shows that restitution, contact stiffness, and contact damping are coupled in the same normal-contact solve. Restitution contributes through $u_{n,\mathrm{rest}}$, while contact stiffness and damping contribute through erp and cfm. The post-impact normal velocity is then determined from the solved impulse,

$$u_n^+ = u_n^- + K_n \lambda_n,$$

where u_n^+ is the post-impact relative normal velocity.

Lateral friction acts on the tangential component of the relative contact velocity. Let

$$\mathbf{u}_t = \mathbf{u}_{rel} - u_n^- \mathbf{n}$$

denote the tangential relative velocity at the contact, and let $\mathbf{t}_1$ and $\mathbf{t}_2$ be orthogonal unit vectors spanning the local tangential plane. Bullet resolves friction through tangential constraint rows, which are commonly interpreted through a Coulomb-type bound,

$$|\lambda_{t_i}| \leq \mu \lambda_n, \qquad i = 1, 2,$$

where λ_{t_i} is the tangential impulse in direction $\mathbf{t}_i$, μ is the lateral friction coefficient, and λ_n is the normal contact impulse. Thus, lateral friction controls resistance to sliding at the contact patch.

Spinning friction resists relative rotation about the contact normal. Let

$$\omega_{spin}^- = \mathbf{n} \cdot (\boldsymbol{\omega}_A - \boldsymbol{\omega}_B)$$

denote the pre-impact relative angular velocity about the normal direction. Bullet introduces an additional torsional resistance associated with this mode, which can be interpreted through a torsional friction impulse or moment opposing ω_{spin}^-. Denoting this generalized spinning-friction impulse by λ_{spin}, the effect is to reduce relative twisting about the contact normal and thereby suppress rotational slip at the interface. Physically, larger spinning friction produces greater resistance to twist, while smaller spinning friction allows easier relative rotation around the normal direction.

S3 Optimization Process

Table S1 summarizes the iterative parameter optimization procedure. In each iteration, one contact parameter was varied while the remaining parameters were held fixed at their current best-fit values, and model performance was evaluated using accuracy, precision, recall, F1 score, and the objective function. The parameter combination with the highest objective value was carried forward to the next iteration. The optimization was terminated when further parameter updates failed to produce a meaningful improvement in the objective function.

Table S2 Iterative parameter optimization results. Metrics are reported for each candidate parameter set.

Iter.	Group	Rest.	Lat. Fric.	Spin. Fric.	Damp.	Stiff.	Acc.	Prec.	Recall	F1	Obj.
1	restitution	0.3	0.3	0.1	10000	1000000	0.770	0.991	0.453	0.621	0.738
1	restitution	0.35	0.3	0.1	10000	1000000	0.768	0.991	0.449	0.618	0.736
1	restitution	0.4	0.3	0.1	10000	1000000	0.768	0.991	0.449	0.618	0.736
1	restitution	0.45	0.3	0.1	10000	1000000	0.766	0.991	0.444	0.614	0.734
1	restitution	0.5	0.3	0.1	10000	1000000	0.770	0.991	0.453	0.621	0.738
1	restitution	0.6	0.3	0.1	10000	1000000	0.773	0.991	0.461	0.629	0.742
1	restitution	0.7	0.3	0.1	10000	1000000	0.770	0.991	0.453	0.621	0.738
1	restitution	0.8	0.3	0.1	10000	1000000	0.770	0.991	0.453	0.621	0.738
1	lateralFriction	0.6	0.31	0.1	10000	1000000	0.780	0.967	0.490	0.650	0.746
1	lateralFriction	0.6	0.41	0.1	10000	1000000	0.794	0.992	0.510	0.674	0.765
1	lateralFriction	0.6	0.51	0.1	10000	1000000	0.818	0.972	0.580	0.727	0.790
1	lateralFriction	0.6	0.55	0.1	10000	1000000	0.816	0.986	0.568	0.721	0.790
1	lateralFriction	0.6	0.66	0.1	10000	1000000	0.813	0.953	0.580	0.721	0.782
1	lateralFriction	0.6	0.7	0.1	10000	1000000	0.825	0.973	0.597	0.740	0.798
1	lateralFriction	0.6	0.8	0.1	10000	1000000	0.818	0.966	0.584	0.728	0.789
1	lateralFriction	0.6	1	0.1	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.01	10000	1000000	0.823	0.973	0.593	0.737	0.796
1	spinningFriction	0.6	1	0.05	10000	1000000	0.832	0.980	0.609	0.751	0.807
1	spinningFriction	0.6	1	0.1	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.2	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.3	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.4	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.5	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	spinningFriction	0.6	1	0.6	10000	1000000	0.833	0.987	0.609	0.753	0.810
1	contactDamping	0.6	1	0.5	1000000	1000000	0.833	0.987	0.609	0.753	0.810
1	contactDamping	0.6	1	0.5	10000000	1000000	0.725	1	0.342	0.509	0.689
1	contactDamping	0.6	1	0.5	100000000	1000000	0.706	0.986	0.300	0.461	0.664
1	contactDamping	0.6	1	0.5	1000000000	1000000	0.679	0.868	0.272	0.414	0.606
1	contactDamping	0.6	1	0.5	10000000000	1000000	0.644	0.630	0.358	0.457	0.544
1	contactStiffness	0.6	1	0.5	10000000	10000000	0.799	0.950	0.547	0.695	0.765
1	contactStiffness	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
1	contactStiffness	0.6	1	0.5	10000000	1000000000	0.873	0.942	0.741	0.829	0.852
1	contactStiffness	0.6	1	0.5	10000000	10000000000	0.873	0.912	0.770	0.835	0.852
2	restitution	0.2	1	0.5	10000000	1000000000	0.856	0.892	0.745	0.812	0.831
2	restitution	0.3	1	0.5	10000000	1000000000	0.871	0.916	0.761	0.831	0.849
2	restitution	0.4	1	0.5	10000000	1000000000	0.852	0.894	0.733	0.805	0.826
2	restitution	0.5	1	0.5	10000000	1000000000	0.873	0.908	0.774	0.836	0.852
2	restitution	0.6	1	0.5	10000000	1000000000	0.869	0.903	0.770	0.831	0.847
2	restitution	0.7	1	0.5	10000000	1000000000	0.874	0.909	0.778	0.838	0.854
2	lateralFriction	0.7	0.4	0.5	10000000	1000000000	0.804	0.951	0.560	0.705	0.772
2	lateralFriction	0.7	0.42	0.5	10000000	1000000000	0.818	0.931	0.609	0.736	0.786

Continued on next page

Table S2 continued from previous page

Iter.	Group	Rest.	Lat. Fric.	Spin. Fric.	Damp.	Stiff.	Acc.	Prec.	Recall	F1	Obj.
2	lateralFriction	0.7	0.45	0.5	10000000	1000000000	0.833	0.924	0.654	0.766	0.804
2	lateralFriction	0.7	0.5	0.5	10000000	1000000000	0.849	0.906	0.712	0.797	0.822
2	lateralFriction	0.7	0.55	0.5	10000000	1000000000	0.854	0.907	0.724	0.805	0.828
2	lateralFriction	0.7	0.6	0.5	10000000	1000000000	0.849	0.897	0.720	0.799	0.822
2	lateralFriction	0.7	0.65	0.5	10000000	1000000000	0.866	0.915	0.749	0.824	0.843
2	lateralFriction	0.7	0.7	0.5	10000000	1000000000	0.868	0.907	0.761	0.828	0.845
2	lateralFriction	0.7	1	0.5	10000000	1000000000	0.873	0.904	0.778	0.836	0.852
2	lateralFriction	0.7	1.25	0.5	10000000	1000000000	0.861	0.886	0.765	0.821	0.837
2	spinningFriction	0.7	1	0.01	10000000	1000000000	0.864	0.898	0.761	0.824	0.841
2	spinningFriction	0.7	1	0.05	10000000	1000000000	0.875	0.913	0.774	0.837	0.854
2	spinningFriction	0.7	1	0.1	10000000	1000000000	0.875	0.921	0.765	0.836	0.854
2	spinningFriction	0.7	1	0.2	10000000	1000000000	0.861	0.901	0.749	0.818	0.837
2	spinningFriction	0.7	1	0.3	10000000	1000000000	0.875	0.921	0.765	0.836	0.854
2	spinningFriction	0.7	1	0.4	10000000	1000000000	0.868	0.895	0.774	0.830	0.846
2	spinningFriction	0.7	1	0.5	10000000	1000000000	0.871	0.900	0.778	0.834	0.850
2	contactDamping	0.7	1	0.5	1000000	1000000000	0.419	0.418	1	0.590	0.612
2	contactDamping	0.7	1	0.5	10000000	1000000000	0.875	0.913	0.774	0.837	0.854
2	contactDamping	0.7	1	0.5	100000000	1000000000	0.854	0.895	0.737	0.808	0.829
2	contactDamping	0.7	1	0.5	1000000000	1000000000	0.818	0.936	0.605	0.735	0.786
2	contactStiffness	0.7	1	0.5	10000000	10000000	0.861	0.922	0.728	0.814	0.837
2	contactStiffness	0.7	1	0.5	10000000	100000000	0.875	0.913	0.774	0.837	0.854
2	contactStiffness	0.7	1	0.5	10000000	1000000000	0.419	0.418	1	0.590	0.612
2	contactStiffness	0.7	1	0.5	10000000	10000000000	0.419	0.418	1	0.590	0.612
3	restitution	0.3	1	0.5	10000000	100000000	0.864	0.918	0.741	0.820	0.841
3	restitution	0.35	1	0.5	10000000	100000000	0.863	0.931	0.724	0.815	0.839
3	restitution	0.4	1	0.5	10000000	100000000	0.854	0.929	0.704	0.801	0.829
3	restitution	0.45	1	0.5	10000000	100000000	0.864	0.932	0.728	0.818	0.841
3	restitution	0.5	1	0.5	10000000	100000000	0.861	0.922	0.728	0.814	0.837
3	restitution	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
3	restitution	0.7	1	0.5	10000000	100000000	0.863	0.918	0.737	0.817	0.839
3	restitution	0.8	1	0.5	10000000	100000000	0.861	0.931	0.720	0.812	0.837
3	lateralFriction	0.6	0.4	0.5	10000000	100000000	0.790	0.969	0.514	0.672	0.758
3	lateralFriction	0.6	0.42	0.5	10000000	100000000	0.797	0.937	0.551	0.694	0.762
3	lateralFriction	0.6	0.45	0.5	10000000	100000000	0.813	0.935	0.593	0.725	0.780
3	lateralFriction	0.6	0.5	0.5	10000000	100000000	0.837	0.930	0.658	0.771	0.808
3	lateralFriction	0.6	0.55	0.5	10000000	100000000	0.849	0.919	0.700	0.794	0.822
3	lateralFriction	0.6	0.6	0.5	10000000	100000000	0.851	0.915	0.708	0.798	0.824
3	lateralFriction	0.6	0.65	0.5	10000000	100000000	0.871	0.947	0.733	0.826	0.850
3	lateralFriction	0.6	0.7	0.5	10000000	100000000	0.866	0.927	0.737	0.821	0.843
3	lateralFriction	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
3	lateralFriction	0.6	1.25	0.5	10000000	100000000	0.859	0.900	0.745	0.815	0.835
3	spinningFriction	0.6	1	0.01	10000000	100000000	0.857	0.908	0.733	0.811	0.833

Continued on next page

Table S2 continued from previous page

Iter.	Group	Rest.	Lat. Fric.	Spin. Fric.	Damp.	Stiff.	Acc.	Prec.	Recall	F1	Obj.
3	spinningFriction	0.6	1	0.05	10000000	100000000	0.851	0.911	0.712	0.799	0.824
3	spinningFriction	0.6	1	0.1	10000000	100000000	0.852	0.911	0.716	0.802	0.826
3	spinningFriction	0.6	1	0.2	10000000	100000000	0.857	0.921	0.720	0.808	0.833
3	spinningFriction	0.6	1	0.3	10000000	100000000	0.854	0.911	0.720	0.805	0.829
3	spinningFriction	0.6	1	0.4	10000000	100000000	0.844	0.909	0.695	0.788	0.816
3	spinningFriction	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
3	contactDamping	0.6	1	0.5	1000000	100000000	0.856	0.892	0.745	0.812	0.831
3	contactDamping	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
3	contactDamping	0.6	1	0.5	100000000	100000000	0.809	0.929	0.588	0.720	0.775
3	contactDamping	0.6	1	0.5	1000000000	100000000	0.718	0.965	0.337	0.500	0.673
3	contactStiffness	0.6	1	0.5	10000000	10000000	0.799	0.950	0.547	0.695	0.765
3	contactStiffness	0.6	1	0.5	10000000	100000000	0.873	0.942	0.741	0.829	0.852
3	contactStiffness	0.6	1	0.5	10000000	600000000	0.854	0.907	0.724	0.805	0.828
3	contactStiffness	0.6	1	0.5	10000000	1100000000	0.856	0.888	0.749	0.812	0.831
3	contactStiffness	0.6	1	0.5	10000000	1600000000	0.868	0.884	0.786	0.832	0.846
3	contactStiffness	0.6	1	0.5	10000000	2600000000	0.868	0.867	0.807	0.836	0.847
3	contactStiffness	0.6	1	0.5	10000000	1000000000	0.869	0.903	0.770	0.831	0.847
3	contactStiffness	0.6	1	0.5	10000000	10000000000	0.419	0.418	1	0.590	0.612
4	restitution	0.3	1	0.5	10000000	2600000000	0.863	0.850	0.815	0.832	0.842
4	restitution	0.35	1	0.5	10000000	2600000000	0.876	0.873	0.823	0.847	0.858
4	restitution	0.4	1	0.5	10000000	2600000000	0.861	0.849	0.811	0.829	0.840
4	restitution	0.45	1	0.5	10000000	2600000000	0.857	0.845	0.807	0.825	0.836
4	restitution	0.5	1	0.5	10000000	2600000000	0.866	0.857	0.815	0.835	0.846
4	restitution	0.6	1	0.5	10000000	2600000000	0.868	0.867	0.807	0.836	0.847
4	restitution	0.7	1	0.5	10000000	2600000000	0.856	0.841	0.807	0.824	0.834
4	restitution	0.8	1	0.5	10000000	2600000000	0.851	0.831	0.807	0.818	0.829
4	lateralFriction	0.35	0.4	0.5	10000000	2600000000	0.799	0.875	0.605	0.715	0.760
4	lateralFriction	0.35	0.42	0.5	10000000	2600000000	0.813	0.853	0.667	0.748	0.777
4	lateralFriction	0.35	0.45	0.5	10000000	2600000000	0.826	0.878	0.679	0.766	0.794
4	lateralFriction	0.35	0.5	0.5	10000000	2600000000	0.861	0.879	0.774	0.823	0.838
4	lateralFriction	0.35	0.55	0.5	10000000	2600000000	0.866	0.873	0.794	0.832	0.845
4	lateralFriction	0.35	0.6	0.5	10000000	2600000000	0.863	0.862	0.798	0.829	0.841
4	lateralFriction	0.35	0.65	0.5	10000000	2600000000	0.861	0.849	0.811	0.829	0.840
4	lateralFriction	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
4	lateralFriction	0.35	1	0.5	10000000	2600000000	0.876	0.873	0.823	0.847	0.858
4	lateralFriction	0.35	1.25	0.5	10000000	2600000000	0.876	0.867	0.831	0.849	0.858
4	spinningFriction	0.35	0.7	0.01	10000000	2600000000	0.864	0.842	0.831	0.836	0.846
4	spinningFriction	0.35	0.7	0.05	10000000	2600000000	0.868	0.864	0.811	0.837	0.847
4	spinningFriction	0.35	0.7	0.1	10000000	2600000000	0.873	0.863	0.827	0.845	0.854
4	spinningFriction	0.35	0.7	0.2	10000000	2600000000	0.876	0.873	0.823	0.847	0.858
4	spinningFriction	0.35	0.7	0.3	10000000	2600000000	0.875	0.866	0.827	0.846	0.856
4	spinningFriction	0.35	0.7	0.4	10000000	2600000000	0.878	0.868	0.835	0.851	0.860

Continued on next page

Table S2 continued from previous page

Iter.	Group	Rest.	Lat. Fric.	Spin. Fric.	Damp.	Stiff.	Acc.	Prec.	Recall	F1	Obj.
4	spinningFriction	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
4	contactDamping	0.35	0.7	0.5	1000000	2600000000	0.419	0.418	1	0.590	0.612
4	contactDamping	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
4	contactDamping	0.35	0.7	0.5	100000000	2600000000	0.863	0.910	0.745	0.819	0.839
4	contactDamping	0.35	0.7	0.5	1000000000	2600000000	0.863	0.941	0.716	0.813	0.840
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.790	0.955	0.523	0.676	0.756
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.854	0.907	0.724	0.805	0.828
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.854	0.916	0.716	0.804	0.829
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.869	0.920	0.753	0.828	0.847
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.861	0.897	0.753	0.819	0.837
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.847	0.889	0.724	0.798	0.820
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.419	0.418	1	0.590	0.612
4	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.419	0.418	1	0.590	0.612
5	contactDamping	0.35	0.7	0.5	5000000	2600000000	0.419	0.418	1	0.590	0.612
5	contactDamping	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
5	contactDamping	0.35	0.7	0.5	15000000	2600000000	0.845	0.870	0.741	0.800	0.819
5	contactStiffness	0.35	0.7	0.5	10000000	1000000000	0.847	0.889	0.724	0.798	0.820
5	contactStiffness	0.35	0.7	0.5	10000000	2000000000	0.854	0.862	0.774	0.816	0.830
5	contactStiffness	0.35	0.7	0.5	10000000	2600000000	0.883	0.869	0.848	0.858	0.867
5	contactStiffness	0.35	0.7	0.5	10000000	3000000000	0.802	0.734	0.827	0.778	0.788
5	contactStiffness	0.35	0.7	0.5	10000000	4000000000	0.491	0.448	0.938	0.606	0.626

S4 Parallel Simulation

A central design goal of VSR2 is to enable high-throughput, large-scale experimentation for ground motion evaluation and parameter-space exploration. In addition to sequentially executing hundreds of individual simulations, VSR2 also supports parallel simulation runs, dramatically reducing total computation time. To support this, the system architecture allows each simulation instance, defined by a namespace with a specific set of rock, ground motions, and physics parameters, to run independently with its own configuration, control logic, and result logging. These instances are launched in parallel and executed simultaneously across available CPU cores. Each simulation independently logs its input trajectory and dynamic response, ensuring collision-free file I/O and eliminating race conditions.

Parallel simulations are orchestrated by a script-driven workflow that automates the full experimental cycle: generating input configurations, launching simulations in batch, collecting results, and refining the parameter space based on performance metrics. By leveraging parallelism at both desktop and cluster scales, VSR2 transforms computationally intensive tasks such as large-scale fragility analyses and physics parameter calibration into tractable and time-efficient processes. Figure S2 shows the runtime required to complete 582 earthquake recordings using 1 to 8 CPU cores on the same desktop system. The results exhibit a near-linear reduction in cumulative runtime with increasing core count. Parallel simulation on 8 cores completes all 582 recordings (Saifullah and Wittich, 2023) in approximately 1,550 s, compared to 9,300 s with serial execution on a single core, yielding a sixfold speed-up. These results highlight the substantial efficiency gains enabled by parallel simulation.

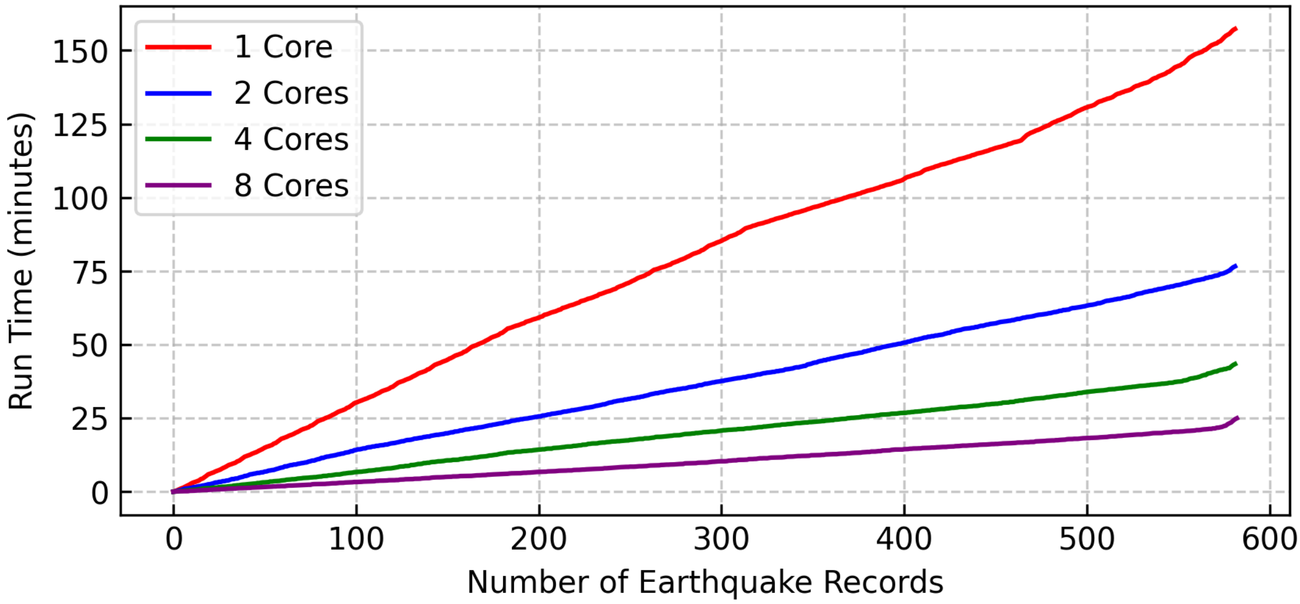

Figure S2 Parallel computing efficiency of VSR2. Total runtime (in minutes) for simulating 582 earthquake recordings using 1, 2, 4, and 8 CPU cores. Increasing the number of cores substantially reduces computation time, demonstrating the scalability and efficiency of the parallel simulation architecture.

References

Coumans, E. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15. ACM, 2015. doi: 10.1145/2776880.2792704.

Coumans, E. and Bai, Y. *PyBullet Quickstart Guide*. Bullet Physics SDK, 2023. Accessed 2025-10-13.

Purvance, M. D., Anooshehpoor, A., and Brune, J. N. Freestanding block overturning fragilities: Numerical simulation and experimental validation. *Earthquake Engineering & Structural Dynamics*, 37(5):791–808, 2008. doi: 10.1002/eqe.789.

Saifullah, M. K. and Wittich, C. E. Uncertainty in overturning of precariously balanced rocks due to basal contact. *Earthquake Engineering & Structural Dynamics*, 52(14):4562–4581, 2023. doi: 10.1002/eqe.3970.