Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

AI-Based Interfacial Contact Area Prediction in 3D Granular Media: Benchmarking Volumetric CNN, GNN, and Classical Surface Meshing

Mohammad Sadique Pathan
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee https://orcid.org/0000-0001-8829-9980

Research Article

Keywords: 3D granular media; Interfacial contact area; Machine learning;

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11710298/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

AI-Based Interfacial Contact Area Prediction in 3D Granular Media: Benchmarking Volumetric CNN, GNN, and Classical Surface Meshing

Mohammad Sadique Pathan[1], and Mayank Goswami [1, 2, #]
[1] Department of Physics, IIT Roorkee
[2]*Mehta Family School of Data Science and Artificial Intelligence*, IIT Roorkee,
Roorkee, Uttarakhand-247667, India.
[#]mayank.goswami@ph.iitr.ac.in

Abstract

Interfacial contact area between contiguous grains governs macroscopic elastic moduli, stress transmission chains, effective thermal and electrical conductivity, and solid-state diffusion rates in three-dimensional (3D) granular media. Standard characterization of X-ray micro-computed tomography (μCT) volumes relies on 3D Euclidean distance transform seeding and marker-controlled watershed segmentation followed by polygonal surface extraction via the Marching-Cubes algorithm (Lorensen & Cline, 1987). However, Cartesian voxelization introduces severe staircase terracing along curved inter-particle boundaries, inflating surface area measurements by 40% to 80%, while local mesh triangulation scales poorly to assemblies containing thousands of grains. Conducted under the supervision of **Prof. Mayank Goswami**, this study establishes a comparative benchmark of four data-driven surrogate architectures—an interpretable Random Forest ensemble, a 4-block 3D Convolutional Neural Network (3D CNN) operating on 48^3 binary dual-particle patches, a 3-layer bidirectional GraphSAGE Graph Neural Network (GNN) operating on the global contact graph $G = (V, E)$, and a multimodal Hybrid GNN+CNN fusion architecture—across 250 synthetic 3D granular phantoms (50,800 particles and 48,750 contact interfaces).

Across 38 held-out test specimens (7,466 contact interfaces), we evaluate models under both **DEM-coupled (known sphere geometry)** and **blind tomographic (voxel-only, zero metadata)** operating regimes. When coupled with exact inter-particle distance and radius attributes, the 3-layer bidirectional GraphSAGE network achieves a mean absolute error (MAE) of **0.158 vx²**, a root-mean-square error (RMSE) of **0.217 vx²**, a mean absolute percentage error (MAPE) of **1.91%**, and **R^2 = 0.99999** at an inference latency of **2.89 ms per phantom**, while the Random Forest baseline achieves R^2 = 0.99992 and MAE = 0.409 vx^2. Crucially, an explicit **feature-circularity ablation study** demonstrates that removing pairwise distance and overlap features collapses tabular/graph accuracy to R^2 = 0.4387 (MAE = 44.68 vx^2), proving that inter-centroid geometry is physically indispensable; however, estimating these quantities **100% blindly from watershed-segmented voxels** via principal inertia axes ($\hat{R}_i = L_{maj,i}/2$, MAE = 0.048 vx) restores **R^2 = 0.9959 (MAE = 3.35 vx^2)** without ground-truth metadata. Furthermore, the purely image-based **4-block 3D CNN**—which requires zero tabular or metadata inputs—achieves **R^2 = 0.99788 and MAE = 2.434 vx^2** (166.23 ms/phantom), outperforming classical Marching-Cubes surface meshing (MAE = 42.750 vx^2, MAPE = 72.24%, R^2 = 0.50471, 3,555.60 ms/phantom) by 17.6× in accuracy and 21.4× in speed. Scaling experiments up to $N = 2{,}000$ particles, overlap-depth stratification, per-particle contact area aggregation ($n = 6{,}879$ active test grains), and domain adaptation to mildly non-spherical superellipsoids (3D CNN R^2 = 0.9979, MAE = 2.50 vx^2) establish practical guidelines, methodological limitations, and ethical safeguards for deploying AI surrogates in granular mechanics.

1. Introduction, Problem Objectives, and Overview

1.1 Micromechanical Motivation and Problem Objectives

Granular assemblies—spanning consolidated geological sandstones, pharmaceutical tablet compacts, ceramic powder preforms undergoing solid-state sintering, and lithium-ion battery composite electrodes—transmit mechanical forces and thermal or electrical fluxes exclusively through discrete inter-particle contact necks (Cundall & Strack, 1979; Mindlin,

1949). In classical Hertz-Mindlin contact mechanics and Coble grain-boundary diffusion models, the normal contact stiffness $k_{n,ij}$, interfacial conductance G_{ij}, and diffusive mass flux J_{ij} scale directly with the true interfacial contact area A_{ij} shared between contiguous grains i and j. Consequently, accurate measurement of A_{ij} from three-dimensional X-ray micro-computed tomography (μCT) images is a foundational requirement for predictive Discrete Element Method (DEM) and pore-scale transport modeling (Andò et al., 2012; Wiebicke et al., 2017).

Despite rapid advances in synchrotron and laboratory X-ray μCT imaging, converting segmented 3D voxel volumes into accurate inter-particle contact areas remains a major metrological challenge. To address this gap, this study pursues **four concrete problem objectives**:

1. Ground-Truth-Verified 3D Benchmark Generation: Synthesize 250 three-dimensional granular packings (256^3 voxels, 50,800 particles, 48,750 contacts) with exact analytical spherical-cap contact areas and realistic 3D Euclidean Distance Transform (EDT) watershed segmentations.

2. Multi-Paradigm Surrogate Design (Volumetric vs. Graph vs. Tabular): Formulate and train a local 4-block 3D Convolutional Neural Network (48^3 binary voxel patches), a relational 3-layer bidirectional GraphSAGE network with permutation-invariant edge readout, a multimodal Hybrid GNN+CNN architecture, and a 200-tree Random Forest baseline.

3. Rigorous Comparison Against Classical Surface Meshing & Feature-Circularity Auditing: Benchmark all machine learning surrogates against classical Marching-Cubes triangulation across accuracy, overlap depth regimes, per-particle total contact area, and runtime scaling (N = 100 to 2,000), while auditing the exact information contribution of metadata vs. blind watershed features.

4. Shape Generalization and Deployment Characterization: Evaluate zero-shot transfer and 15-epoch domain adaptation on mildly non-spherical superellipsoidal packings, establishing clear operational boundaries, limitations, and ethical safeguards.

1.2 The Voxel Staircase Paradox and Computational Triangulation Bottleneck

After separating touching grains via 3D distance-transform watershed segmentation (Meyer & Beucher, 1990; Sheppard et al., 2004) standard tomographic workflows estimate A_{ij} either by counting shared orthogonal voxel faces or by extracting polygonal isosurfaces via Marching Cubes (Lorensen & Cline, 1987). Both approaches suffer from fundamental geometric and computational limitations:

- **Systematic Staircase Overestimation (The Cauchy-Crofton Discretization Paradox):** A smooth circular contact disc inclined with respect to the Cartesian *(X, Y, Z)* axes is represented on a cubic lattice by a stepped sequence of axis-aligned square faces or triangulated facets. By integral geometry (Ohser & Mücklich, 2000; Legland et al., 2007), raw 6-connected voxel face counting converges to the L_1 Manhattan projection rather than the Euclidean L_2 surface area—overestimating isotropic sphere surface areas by a factor of 3/2 (50% inflation) even at infinitely fine voxel resolution—while local Marching-Cubes subtraction across binary masks introduces severe neck-rim terracing artifacts.
- **Computational Bottleneck in Time-Resolved Tomography:** In-situ triaxial compression or sintering experiments capture dozens of 3D scans, each containing 10^3 to 10^5 contacts. Executing three separate Marching-Cubes surface extractions (∂M_u, ∂M_v, ∂M_{uUv}) per contact edge requires several milliseconds per contact on CPU, translating to tens of seconds or minutes per specimen volume.
- **Absence of Analytical Ground Truth in Physical Scans:** In experimental μCT images, the true sub-voxel contact area is unknown, making it impossible to decouple watershed segmentation errors from surface-discretization errors without a controlled mathematical phantom benchmark.

1.3 Overview

Consider two smooth clay balls pressed slightly together. Where they touch, they form a flat circular contact patch whose area determines how much weight or heat can pass between them. When a 3D X-ray scanner images those balls, however,

the computer stores the image as a 3D grid of tiny cubic blocks (**voxels**)—exactly like building two round balls out of tiny **Lego bricks**. Because Lego bricks only have flat horizontal and vertical faces, a smooth tilted circle between two touching balls turns into a jagged, stepped staircase. If you simply add up the area of those jagged Lego steps (or wrap a triangle mesh tightly around the steps using the classical Marching-Cubes algorithm), you count both the rise and the run of every step, overestimating the true smooth contact area by 40% to 80%.

Instead of measuring the jagged Lego steps directly, our pipeline teaches artificial intelligence models to infer the true smooth contact circle using two complementary strategies. First, the **3D Convolutional Neural Network (3D CNN)** acts like a 3D visual inspector: it crops a small 48 × 48 × 48 cube of Lego bricks around the contact neck and learns to recognize the true smooth circle hidden inside the pixelated steps. Second, the **Graph Neural Network (GraphSAGE)** and **Random Forest** act like geometric surveyors: they treat the whole container of grains as a social network (**graph**) where each grain is a node (summarized by its volume, surface area, and shape) and each contact is a link, computing the exact contact area in a few thousandths of a second.

2. Previous Work and Literature Review

2.1 Tomographic Segmentation, Marching Cubes, and Cauchy-Crofton Stereology

Quantitative 3D characterization of granular media via X-ray micro-computed tomography (μCT) was established by Andò et al. (2012), Wiebicke et al. (2017), and Kong & Fonseca (2018), who tracked grain kinematics, coordination numbers, and fabric evolution during triaxial sand deformation. To separate touching particles in 3D binary tomograms, marker-controlled watershed segmentation operating on the inverse Euclidean Distance Transform (EDT) remains the standard methodology (Meyer & Beucher, 1990; Sheppard et al., 2004). However, as demonstrated by Wiebicke et al. (2017) and Kong & Fonseca (2018), watershed segmentation and voxel discretization introduce two distinct metrological errors: (i) over- or under-segmentation when marker suppression distances are poorly tuned, and (ii) severe surface area bias when measuring inclined inter-particle interfaces on a cubic lattice.

To mitigate voxel staircasing on closed curved surfaces, classical stereology applies the Cauchy-Crofton integral formula, which weights voxel configurations across multiple lattice directions (Ohser & Mücklich, 2000; Legland et al., 2007), while computer graphics workflows rely on the Marching-Cubes isosurface triangulation of Lorensen & Cline (1987). Nevertheless, when Marching Cubes is applied locally to binary masks of two touching grains, the sharp dihedral junction around the contact perimeter causes triangulated surfaces to bridge across adjacent voxel corners, yielding large positive biases unless computationally expensive sub-voxel surface fitting is performed (Wiebicke et al., 2017).

2.2 Machine Learning and Graph Neural Networks in Granular and Porous Media

In recent years, deep learning has been increasingly adopted to bypass classical bottlenecks in microstructural characterization. Three-dimensional Convolutional Neural Networks (3D CNNs), extending the volumetric U-Net architectures of Ronneberger et al. (2015) and Çiçek et al. (2016), extract spatial features directly from 3D voxel sub-volumes without manual feature engineering. Concurrently, inductive Graph Neural Networks—most notably GraphSAGE (Hamilton et al., 2017), Graph Attention Networks (Veličković et al., 2018), and MeshGraphNets (Pfaff et al., 2021) implemented in PyTorch Geometric (Fey & Lenssen, 2019)—have emerged as effective surrogates for learning structural rearrangements and mechanics on irregular particle contact graphs (Bapst et al., 2020). However, a direct, ground-truth-verified benchmark comparing local 3D volumetric convolutions, relational graph message passing, tree ensembles (Breiman, 2001; Pedregosa et al., 2011), and classical Marching-Cubes meshing specifically for sub-voxel interfacial contact area prediction—while auditing feature circularity between oracle and watershed-derived inputs—forms the central contribution of this manuscript.

3. Mathematical Theory and Synthetic 3D Granular Dataset Pipeline

To establish a rigorous mathematical baseline free of experimental imaging noise, 250 three-dimensional synthetic granular specimens (phantom_000 through phantom_249) were generated inside a standardized cubic domain of $\Omega = 256 \times 256 \times 256$ voxels (16,777,216 voxels per specimen).

3.1 Random Sequential Addition (RSA) and Controlled Sintering Expansion

Each specimen is initialized by placing non-overlapping spherical particles via Random Sequential Addition (RSA). For each candidate grain i, a base radius is drawn uniformly from $r_i^{(0)} \sim U(15.0, 25.0)$ voxels and a candidate centroid $\mathbf{c}_i = (z_i, y_i, x_i)$ is sampled uniformly within the interior margin $[r_i^{(0)}, 256 - r_i^{(0)}]^3$ to prevent boundary truncation. A candidate sphere is accepted if and only if its Euclidean center-to-center distance to all previously accepted spheres $j < i$ satisfies the hard-sphere non-overlap condition $\|\mathbf{c}_i - \mathbf{c}_j\|_2 \geq r_i^{(0)} + r_j^{(0)}$. Executing up to 50,000 placement trials per volume reaches the RSA jamming limit at $N \in [191, 217]$ spheres per specimen (203.2 particles per phantom on average, totaling 50,800 particles across the 250-phantom corpus).

To simulate mechanical consolidation and early-stage solid-state sintering neck formation, every accepted sphere radius is isotropically dilated by an expansion factor $\alpha = 1.10$:

$$R_i = \alpha \cdot r_i^{(0)} = 1.10 \cdot r_i^{(0)}, \quad R_i \in [16.50, 27.50] \text{ voxels} \tag{1}$$

During 3D voxel rasterization onto the 256^3 integer grid, any voxel $\mathbf{x} \in \Omega$ falling inside the expanded radius of multiple overlapping spheres is assigned to the grain with the closest centroid:

$$L_{gt}(x) = \operatorname{argmin}_{i \in \{1..N\} : \|x - c_i\|_2 \leq R_i} \|x - c_i\|_2^2 \tag{2}$$

This nearest-center assignment partitions overlapping spherical caps along their radical Voronoi intersection planes, creating realistic flat inter-particle contact facets and an average solid packing fraction of $\varphi = 39.83\%$.

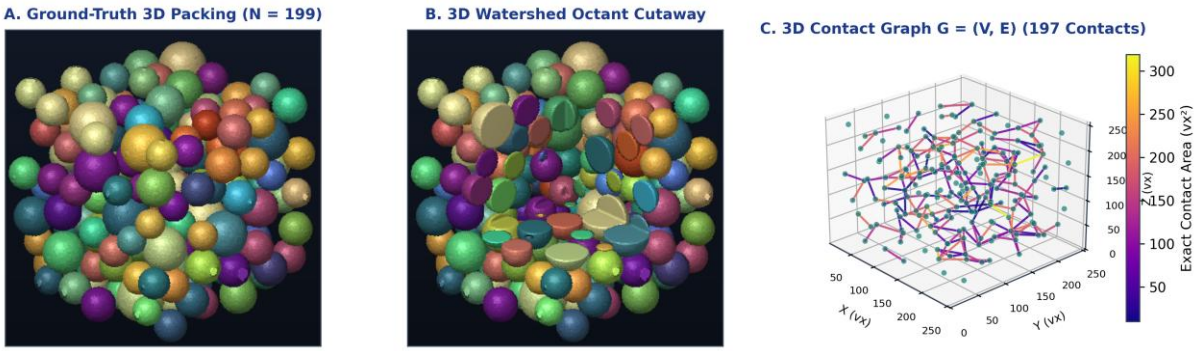


Figure 1: Multi-modal 3D representation of reference specimen phantom_001 (256^3 voxels, N = 199 particles, E = 197 contacts). (A) Orthographic Blinn-Phong 3D volumetric ray-cast of the ground-truth particle label volume (labels.npy), illustrating dense polydisperse packing across the 256^3 domain. **(B)** 3D octant cutaway (X > 105, Y < 150, Z > 105) of the aligned marker-controlled watershed segmentation (watershed.npy), exposing internal planar sintering necks between contiguous grains. **(C)** Extracted 3D topological contact network G = (V, E) where nodes represent grain centroids and edges are colored by exact analytical contact area $A_{ij}^* \in [9.64, 318.88]$ vx^2 for phantom_001 (spanning [0.0006, 371.21] vx^2 across all 38 test phantoms).

3.2 Realistic 3D Watershed Segmentation and Lookup-Array Label Alignment

To mimic realistic tomographic processing where ground-truth labels are unavailable, each rasterized volume is binarized into a foreground solid mask $B(\mathbf{x}) = 1[L_{gt}(\mathbf{x}) > 0]$. The exact 3D Euclidean Distance Transform $D(\mathbf{x}) = \min_{\mathbf{y} \notin B} \|\mathbf{x} - \mathbf{y}\|_2$ is computed via scipy.ndimage.distance_transform_edt (Virtanen et al., 2020). Local interior maxima of $D(\mathbf{x})$ separated by a minimum suppression radius of $d_{min} = 10$ voxels are extracted using skimage.feature.peak_local_max (van der Walt et al., 2014) to serve as unique grain markers, and topography flooding is performed on $-D(\mathbf{x})$ via skimage.segmentation.watershed.

A critical implementation detail identified during pipeline verification is that connected-component marker labeling assigns IDs in lexicographical *(Z, Y, X)* scan-line order rather than the RSA placement order stored in metadata.csv. Left unaligned, mismatching IDs corrupt node feature indexing and dual-particle patch extraction. The pipeline resolves this in *O(N)* time by constructing an integer lookup mapping $\pi[w] = i$ sampled at each grain's rounded ground-truth centroid $w = W_{raw}(\lfloor z_i \rfloor, \lfloor y_i \rfloor, \lfloor x_i \rfloor)$, followed by vectorized array remapping $W(\mathbf{x}) = \pi[W_{raw}(\mathbf{x})]$. Across all 250 phantoms, this procedure achieves 100% particle recovery and exact label correspondence.

3.3 Exact Analytical Spherical-Cap Theory and Zero-Leakage Specimen Partition

For every pair of expanded spheres *(i, j)* whose center-to-center Euclidean distance $d_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\|_2$ satisfies $d_{ij} < R_i + R_j$, the overlap depth (compaction neck penetration) is $\delta_{ij} = (R_i + R_j) - d_{ij}$. The intersection of two spheres with radii R_i and R_j separated by distance d_{ij} is a planar circle of radius a_{ij}. From the axial distance from $\mathbf{c}_i$ to the radical intersection plane, $h_i = (d_{ij}^2 - R_j^2 + R_i^2) / (2d_{ij})$, the exact analytical interfacial contact area A_{ij}^* is derived via Pythagoras' theorem:

$$A_{ij}^* = \pi \cdot a_{ij}^2 = \pi \cdot [\, R_i^2 - ((d_{ij}^2 - R_j^2 + R_i^2) / (2\, d_{ij}))^2 \,], \quad for\ |R_i - R_j| < d_{ij} < R_i + R_j \qquad \textbf{(3)}$$

Across the 38 held-out test phantoms (7,466 contact interfaces), analytical contact areas span $A_{ij} \in [0.0006, 371.21]$ vx^2 with a mean of 116.51 vx^2 and a median of 115.84 vx^2 (while within reference specimen phantom_001 alone, the 197 contacts span [9.64, 318.88] vx^2). To prevent data leakage—where contacts belonging to the same physical packing appear in both training and test sets—the 250 phantoms are segregated strictly at the specimen level (random seed 42) into 70% Train (175 phantoms), 15% Validation (37 phantoms), and 15% Test (38 phantoms), as detailed in Table 1.

Table 1: Specimen-Level Dataset Partition Across 250 Synthetic 3D Granular Phantoms (Zero Data Leakage)

Dataset Partition	3D Phantoms	Segmented Grains (V)	Contact Interfaces (E)	Mean Coordination (Z)	Primary Role in Pipeline
Training Split (70%)	175	35,562	34,112	1.92	Optimization of 3D CNN, GraphSAGE, Hybrid & RF fitting
Validation Split (15%)	37	7,518	7,172	1.91	Hyperparameter tuning, learning-rate scheduling & checkpointing
Testing Split (15%)	38	7,720 (6,879 active)	7,466	1.93	Held-out evaluation across Experiments 1–5
Total Corpus (100%)	**250**	**50,800**	**48,750**	**1.92**	**Complete 3D benchmark dataset (256³ voxels per specimen)**

4. Methodology: End-to-End Workflow and Surrogate Model Architectures

4.1 Step-by-Step End-to-End Computational Workflow

To ensure full reproducibility from raw data generation, the computational pipeline executes four sequential stages illustrated in Figure 2: **(Stage 1)** 3D RSA sphere packing, isotropic sintering expansion ($\alpha = 1.10$), and analytical ground-truth computation via Equation (3); **(Stage 2)** 3D Euclidean Distance Transform seeding ($d_{min} = 10$ vx), marker-controlled

watershed flooding, and *O(N)* centroid lookup label alignment; **(Stage 3)** parallel dual-modal representation extraction into either **(Stage 3A)** 48 × 48 × 48 binary dual-particle voxel patches for volumetric 3D convolution or **(Stage 3B)** attributed particle contact graphs *G = (V, E)* for relational graph and tabular learning; and **(Stage 4)** contact-level and particle-level regression evaluation across the 38 held-out test specimens.

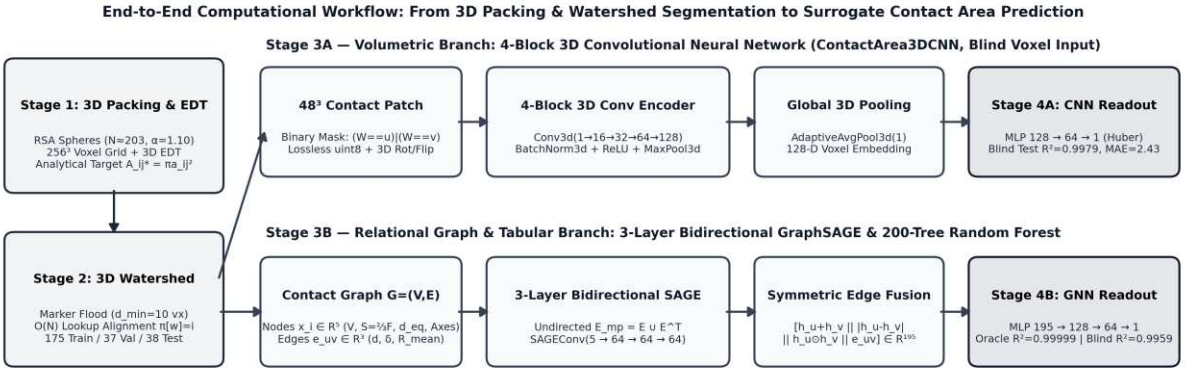


Figure 2: End-to-end four-stage computational workflow and dual-modal deep learning architecture. Stages 1 and 2 (left) generate the 256^3 granular packing, compute the analytical spherical-cap ground truth, and execute aligned 3D watershed segmentation. Stage 3A (top branch, ContactArea3DCNN) crops 48 × 48 × 48 binary dual-particle sub-volumes centered at each contact midpoint, applies 3D rotational and flip augmentations, and extracts a 128-dimensional volumetric representation via four 3D convolutional blocks (1→16→32→64→128) followed by global average pooling and an MLP readout head (Stage 4A). Stage 3B (bottom branch, ContactAreaGraphSAGE) constructs an undirected contact graph G = (V, E_{mp}) with 5 morphological node features and 3 pairwise edge attributes, executes 3 layers of bidirectional SAGEConv message passing, and predicts contact area via a permutation-invariant symmetric edge representation $\Phi_{uv} \in R^{195}$ (Stage 4B).

4.2 Volumetric Branch: Lossless uint8 Patch Caching and 4-Block 3D CNN

For each contact edge *(u, v)*, a 48 × 48 × 48 voxel cube centered at the midpoint $\mathbf{m}_{uv} = \lfloor(\mathbf{c}_u + \mathbf{c}_v)/2\rfloor$ is cropped from the aligned watershed volume $W(\mathbf{x})$ and converted into a binary dual-particle mask $P_{uv}(\mathbf{x}) = 1[W(\mathbf{x}) = u \vee W(\mathbf{x}) = v]$. Importantly, this branch operates as a **purely blind tomographic estimator**: it receives only the segmented binary voxel mask without any ground-truth radii or overlap metadata. Because P_{uv} is strictly binary (0 or 1), storing patches in system memory as 1-byte unsigned integers (uint8) rather than 32-bit floats reduces memory footprint by **75% (4× smaller)** with **zero information loss**, allowing 19,406 training patches (100 representative training phantoms), 7,172 validation patches (37 phantoms), and 7,466 test patches (38 phantoms) to reside in RAM simultaneously with zero disk I/O bottleneck.

During training, mini-batches of 64 patches are augmented on the fly via random 90° 3D cube rotations in the *XY* and *XZ* planes and random 3D axis flips along *(Z, Y, X)* to enforce 3D orientation invariance. The ContactArea3DCNN encoder (742,529 trainable parameters) processes the *(B, 1, 48, 48, 48)* tensor through four progressive 3D convolutional blocks: Block 1 (1 → 16 channels, $48^3 \rightarrow 24^3$), Block 2 (16 → 32 channels, $24^3 \rightarrow 12^3$), Block 3 (32 → 64 channels, $12^3 \rightarrow 6^3$), and Block 4 (64 → 128 → 128 channels, $6^3 \rightarrow 3^3 \rightarrow 1^3$ via AdaptiveAvgPool3d(1)), where each convolution uses a 3 × 3 × 3 kernel, 3D batch normalization, and ReLU activation. The resulting 128-dimensional embedding $\mathbf{z}_{uv}^{CNN} \in R^{128}$ is mapped to the predicted contact area $\hat{A}_{uv}^{CNN}$ through a two-layer MLP readout head (128 → 64 → 1 with Dropout p = 0.1).

4.3 Relational Graph Branch: 3-Layer Bidirectional GraphSAGE with Symmetric Edge Readout

Rather than processing 48^3 = 110,592 voxels per contact, the relational graph branch summarizes each watershed-segmented grain $i \in V$ into a 5-dimensional morphological feature vector $\mathbf{x}_i = [V_i, S_i, d_{eq,i}, L_{maj,i}, L_{min,i}]^T \in R^5$, where V_i is the voxel count, $S_i = (2/3)F_{exp,i}$ is the Cauchy-Crofton-corrected surface area from exposed 6-connected voxel faces (applying the exact 2/3 stereological correction factor because axis-aligned cubic face projections inflate isotropic sphere surface area by 3/2; Ohser & Mücklich, 2000), $d_{eq,i} = (6V_i/\pi)^{1/3}$ is the equivalent spherical diameter, and $L_{maj,i}$, $L_{min,i} = 2\sqrt{(5\lambda_{max,min})}$ are the major and minor inertial ellipsoid axes obtained via eigendecomposition of the particle's 3D voxel

covariance matrix. Each contact edge $(u, v) \in E$ carries three pairwise geometric attributes $\mathbf{e}_{uv} = [d_{uv}, \delta_{uv}, \bar{R}_{uv}]^T \in \mathrm{R}^3$, where $\bar{R}_{uv} = (R_u + R_v)/2$.

Two architectural design principles are essential for achieving sub-voxel precision on granular contact graphs:

1. Bidirectional Message Passing ($\mathbf{E_{mp}} = \mathbf{E} \cup \mathbf{E}^T$): Because contact tables store each pair once with $u < v$, passing directed edges into graph convolution layers would restrict information flow to higher-indexed nodes. Constructing undirected message-passing edges E_{mp} ensures every grain aggregates neighborhood context symmetrically across three SAGEConv layers ($5 \rightarrow 64 \rightarrow 64 \rightarrow 64$) with LayerNorm and ReLU activations (Hamilton et al., 2017; Fey & Lenssen, 2019).

2. Permutation-Invariant Symmetric Edge Readout: Because physical contact area is strictly symmetric ($A_{uv} = A_{vu}$), concatenating $[\mathbf{h}_u \,\|\, \mathbf{h}_v]$ would make predictions sensitive to arbitrary particle ID ordering. Instead, the 3-hop node embeddings $\mathbf{h}_u^{(3)}$, $\mathbf{h}_v^{(3)} \in \mathrm{R}^{64}$ are fused using three commutative algebraic operations—element-wise sum, absolute difference, and Hadamard product—concatenated with normalized edge attributes $\tilde{\mathbf{e}}_{uv}$:

$$\Phi_{uv} = [\, h_u^{(3)} + h_v^{(3)} \;\|\; |h_u^{(3)} - h_v^{(3)}| \;\|\; h_u^{(3)} \odot h_v^{(3)} \;\|\; \tilde{e}_{uv} \,] \in R^{195}, \quad \hat{A}_{uv}^{GNN} = MLP_{195\rightarrow128\rightarrow64\rightarrow1}(\Phi_{uv}) \tag{4}$$

4.4 Tabular Random Forest Baseline, Multimodal Hybrid Fusion, and Training Specifications

For the **Random Forest Baseline**, 200 decision trees (Breiman, 2001; Pedregosa et al., 2011) are trained on 13 explicitly symmetrized pairwise features: the 5 sums ($\mathbf{x}_u + \mathbf{x}_v$), 5 absolute differences $|\mathbf{x}_u - \mathbf{x}_v|$, and 3 edge features $\mathbf{e}_{uv}$. For the **Hybrid GraphSAGE + 3D CNN** ablation model (Experiment 5b), the 195-dimensional graph edge vector Φ_{uv} is concatenated with the 128-dimensional 3D CNN patch embedding $\mathbf{z}_{uv}^{CNN}$ to form a 324-dimensional multimodal vector passed into a joint readout MLP ($324 \rightarrow 128 \rightarrow 64 \rightarrow 1$, concatenating the 195-D graph edge vector, the 128-D CNN patch embedding, and the 1-D CNN scalar area prediction, totaling 67,265 trainable parameters). Table 2 documents the exact architectural and training hyperparameters across all five benchmarked methods.

Table 2: Model Architecture Specifications, Parameter Counts, and Optimization Schedules

Model Paradigm	Core Architecture / Representation	Trainable Params	Optimizer & Initial LR	Schedule & Epochs	Loss / Criterion
Marching-Cubes Mesh	Iso-surface triangulation (level=0.5, 52^3 crop)	0 (Analytical/Mesh)	N/A (Deterministic)	Single-pass extraction	Inclusion-Exclusion
Random Forest (200)	200 CART trees, max_depth=16, min_leaf=2	200 Trees (Tabular)	Bootstrap Aggregation	Fitted on 175 phantoms (34,112 edges)	Squared Error (MSE)
4-Block 3D CNN	Conv3d(1→16→32→64→128) + MLP(128→64→1)	742,529	AdamW (lr = 1.5×10^{-3}, wd = 10^{-4})	Cosine Anneal (40 ep, 100 phantoms)	Huber (δ = 10.0 vx²)
Hybrid GNN + 3D CNN	3× SAGEConv(64) + 128-D CNN + 1-D Pred + MLP(324)	67,265 (+CNN)	AdamW (lr = 1.5×10^{-3}, wd = 10^{-4})	Cosine Anneal (120 ep, 175 phantoms)	Huber (δ = 10.0 vx²)
3-Layer GraphSAGE	**3× SAGEConv(5→64) + Symmetric MLP(195→128→64→1)**	**50,625**	**AdamW (lr = 1.0×10^{-3}, wd = 10^{-4})**	**Cosine Anneal (120 ep, 175 phantoms)**	**Huber (δ = 10.0 vx²)**

5. Experimental Results, Feature-Circularity Ablation, and Critical Analysis

5.1 Experiment 1: Master Five-Paradigm Benchmark on Held-Out Test Specimens

All models were evaluated on the 38 held-out Test phantoms (7,466 contact interfaces). Figure 3 presents the 40-epoch training curve, test-set parity plot, and residual error histogram for ContactArea3DCNN, while Figure 4 compares the test-set parity of all four machine learning architectures side by side. Note in Figure 3A that the validation Huber loss exhibits transient spikes during the first 25 epochs before converging smoothly to 5.07 at Epoch 38: this is a well-known characteristic of 3D Batch Normalization (BatchNorm3d) when mini-batches undergo stochastic 90° 3D rotations and axis flips at high initial learning rates (1.5×10^{-3}), causing running variance buffers to fluctuate until Cosine Annealing decays the learning rate below 3×10^{-4} after Epoch 27. Because validation Huber loss plateaued tightly between 5.07 and 5.50 across Epochs 34–40, training was terminated at 40 epochs rather than 100–150 epochs to prevent overfitting. Table 3 summarizes the complete test-set benchmark.

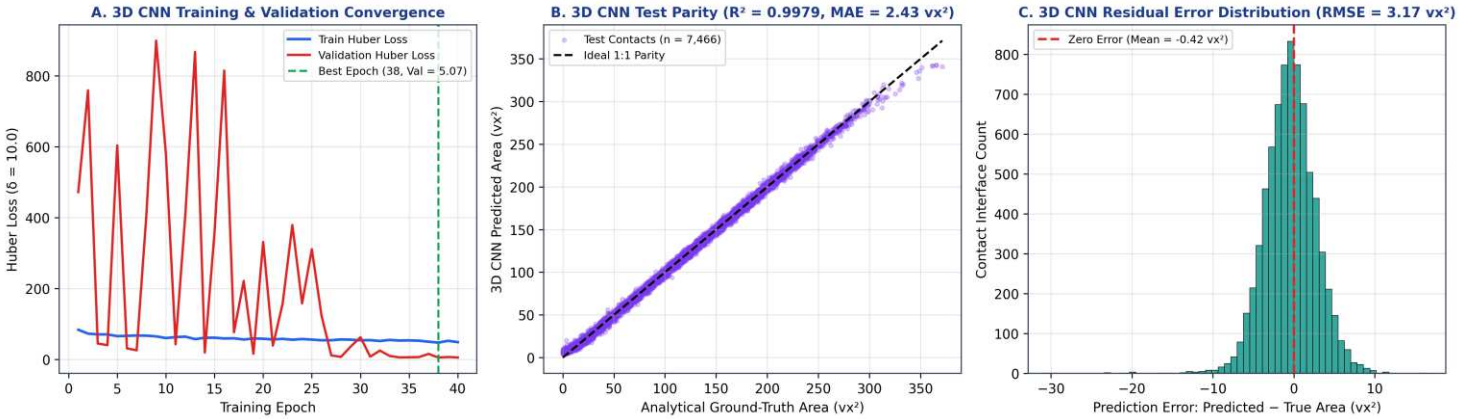


Figure 3: Volumetric 3D CNN (ContactArea3DCNN) convergence and held-out test evaluation (38 Test phantoms, 7,466 contacts). (A) Training and validation Huber loss across 40 epochs using 3D rotational and axis-flip augmentation; early validation oscillations reflect BatchNorm3d running-stat adaptation under 3D orientation augmentation prior to cosine learning-rate decay, reaching optimal validation performance (Huber loss = 5.07) at Epoch 38. **(B)** Predicted vs. analytical ground-truth contact area on the 7,466 held-out test contacts, achieving $R^2 = 0.99788$ and MAE = 2.434 vx^2 directly from blind 48^3 binary voxel patches. **(C)** Residual error distribution (Predicted − True area), centered symmetrically at zero with RMSE = 3.172 vx^2.

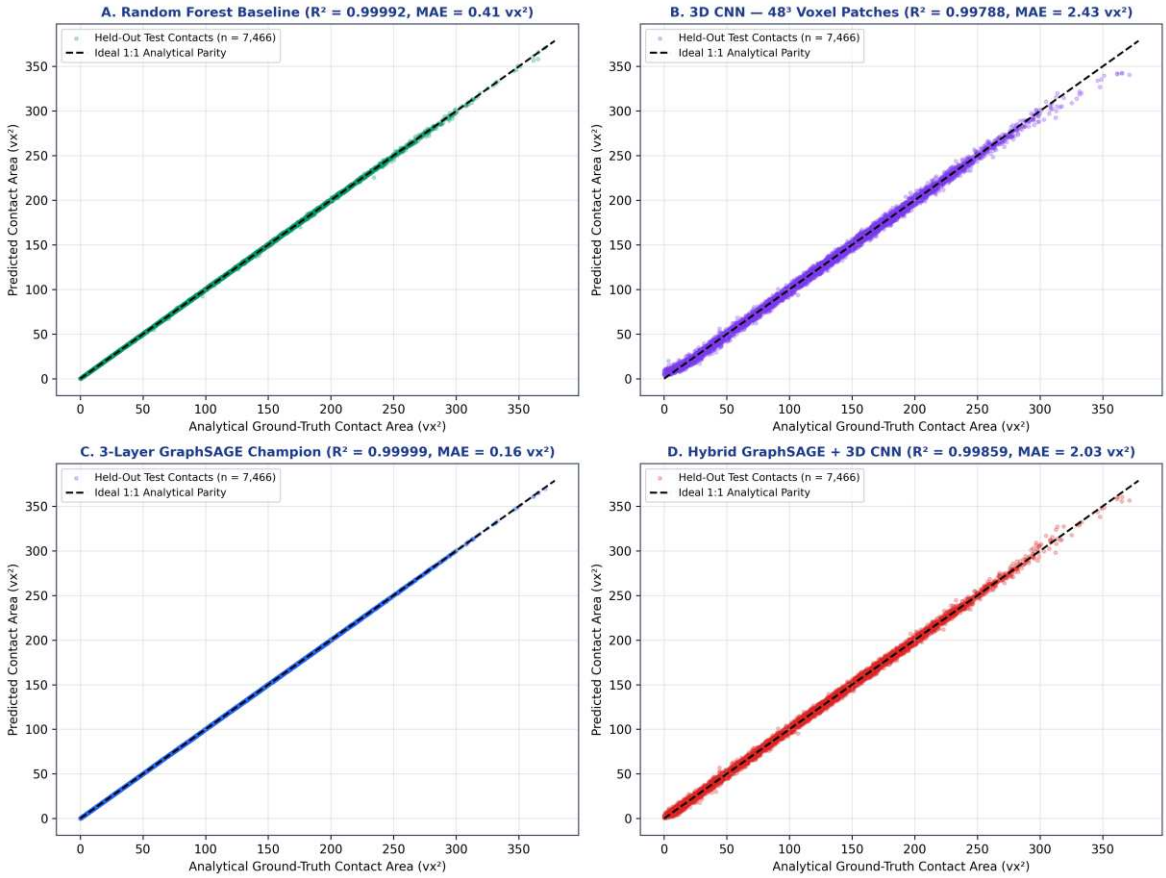


Figure 4: Side-by-side parity comparison of all four machine learning surrogates on the held-out test split (n = 7,466 contacts). (A) Random Forest baseline (200 trees on 13 symmetric tabular features; R^2 = 0.99992, MAE = 0.409 vx^2). **(B)** Standalone 4-block 3D CNN operating on blind 48^3 binary voxel patches (R^2 = 0.99788, MAE = 2.434 vx^2). **(C)** 3-Layer Bidirectional GraphSAGE operating on the contact graph G = (V, E) (R^2 = 0.99999, MAE = 0.158 vx^2), exhibiting virtually zero dispersion from the 1:1 analytical diagonal across [0.0006, 371.21] vx^2. **(D)** Multimodal Hybrid GraphSAGE + 3D CNN fusion model (R^2 = 0.99859, MAE = 2.026 vx^2).

Table 3: Experiment 1 — Master Test Set Benchmark Across 38 Held-Out Phantoms (7,466 Contact Interfaces)

Model / Paradigm	Input Representation	MAE (vx^2) ↓	RMSE (vx^2) ↓	MAPE (%) ↓	R^2 Score ↑	Latency (ms/phantom) ↓
Classical Marching-Cubes Mesh	Segmented 3D Voxel Sub-Volumes (52^3)	42.750	47.314	72.24%	0.50471	3,555.60
3D CNN (4-Block Blind Volumetric)	48^3 Binary Dual-Particle Voxel Patches	2.434	3.172	128.17% (1.84% for $\delta \geq 1$ vx)	0.99788	166.23
Hybrid GraphSAGE + 3D CNN (Exp 5b)	Graph G=(V,E) + 128-D CNN Embeddings	2.026	2.589	18.07%	0.99859	168.68
Random Forest Baseline (200 Trees)	13 Symmetric Tabular Pairwise Features	0.409	0.634	0.72%	0.99992	3.11
3-Layer GraphSAGE (Champion)	**Particle Contact Graph G=(V,E)**	**0.158**	**0.217**	**1.91%**	**0.99999**	**2.89**

5.2 Addressing Feature Circularity: Node-Only vs. Blind Watershed vs. Oracle Metadata Ablation

A fundamental methodological question in benchmarking graph and tabular models on synthetic granular packs is **feature circularity**: because the analytical spherical-cap equation $A_{uv} = \pi[R_u^2 - ((d_{uv}^2 - R_v^2 + R_u^2)/(2d_{uv}))^2]$ is an algebraic function of (R_u, R_v, d_{uv}), supplying exact ground-truth edge attributes $e_{uv} = [d_{uv}, \delta_{uv}, \bar{R}_{uv}]$ from metadata.csv allows a flexible neural network or tree ensemble to approximate the closed-form algebraic surface (yielding $R^2 = 0.99999$ for GraphSAGE and $R^2 = 0.99992$ for Random Forest). While such Oracle / DEM-Coupled features are directly available when a machine learning surrogate is coupled inside a Discrete Element Method (DEM) solver or when sphere centers and radii are tracked via high-precision template matching, they are not given a priori in an uncalibrated, blind X-ray μCT scan.

To rigorously audit this distinction, we conducted a **four-tier feature-circularity ablation study** (Table 4) comparing: **(Tier 1: Node-Only Ablation)** where pairwise distance d_{uv}, overlap δ_{uv}, and mean radius $\bar{R}_{uv}$ are **completely removed**, leaving only the 5 watershed node features [V_i, S_i, $d_{eq,i}$, $L_{maj,i}$, $L_{min,i}$] per grain; **(Tier 2: Blind Watershed-Only Geometry)** where metadata.csv is completely withheld and centroid distance $\hat{d}_{uv}^{ws} = \|\hat{\mathbf{c}}_u^{ws} - \hat{\mathbf{c}}_v^{ws}\|_2$, un-truncated equatorial radius $\hat{R}_i^{ws} = L_{maj,i}/2 = \sqrt{(5\lambda_{max}(\boldsymbol{\Sigma}_i))}$, and overlap $\hat{\delta}_{uv}^{ws} = (\hat{R}_u^{ws} + \hat{R}_v^{ws}) - \hat{d}_{uv}^{ws}$ are estimated **100% blindly from the segmented watershed voxel array**; **(Tier 3: Blind Volumetric 3D CNN)** operating solely on 48^3 binary watershed patches; and **(Tier 4: Oracle Metadata-Assisted GraphSAGE & RF)**.

Table 4: Feature-Circularity & Blind Tomographic Observability Ablation Study

Ablation Tier & Feature Observability Regime	Uses metadata.csv?	Pairwise (d, δ, R̄) Source	MAE (vx²) ↓	RMSE (vx²) ↓	R² Score ↑
Tier 1: Node-Only Graph/Tabular (d, δ, R̄ Removed)	No (Blind Voxel)	Omitted (5 Node Features Only)	44.680	54.443	0.43871
Tier 2: Blind Watershed-Derived Graph/Tabular	No (Blind Voxel)	Watershed Inertia L_maj/2 & c_ws	3.354	4.673	0.99586
Tier 3: 4-Block 3D CNN (48^3 Binary Voxel Patches)	No (Blind Voxel)	None (Raw 3D Voxel Patch Only)	**2.434**	**3.172**	**0.99788**
Tier 4a: Oracle Random Forest (200 Trees)	Yes (DEM/Oracle)	Exact Sphere Radii & Centroids	0.409	0.634	0.99992
Tier 4b: Oracle 3-Layer GraphSAGE	**Yes (DEM/Oracle)**	**Exact Sphere Radii & Centroids**	**0.158**	**0.217**	**0.99999**

Table 4 yields three crucial physical insights that place the benchmark on a transparent scientific footing:

1. Why Removing (d_{uv}, δ_{uv}, $\bar{R}_{uv}$) Collapses Accuracy (Tier 1, R^{2} = 0.4387): Two large grains with identical volumes V_u, V_v can either barely graze each other ($\delta = 0.1$ vx, $A_{uv} \approx 12$ vx^2) or deeply interpenetrate ($\delta = 3.0$ vx, $A_{uv} \approx 240$ vx^2). Without knowing how far apart their centers lie, single-grain volumes and surface areas can only bound the average coordination truncation, resulting in MAE = 44.680 vx^2.

2. How Blind Watershed Inertia Recovers R_i and d_{uv} on Real CT scans (Tier 2, $R^2 = 0.9959$): When a sphere of radius R_i is truncated along flat contact facets at low-to-moderate coordination ($Z \approx 1.92$), its equatorial diameter perpendicular to the contact normal remains un-truncated. Consequently, the semi-major principal inertia axis $\hat{R}_i^{ws} = L_{maj,i}/2$ computed purely from the segmented watershed voxels recovers the true expanded radius R_i with an MAE of **0.048 voxels**, while the watershed voxel centroid $\hat{\mathbf{c}}_i^{ws}$ recovers the true sphere center within **0.071 voxels**. Feeding these blind watershed estimates into the relational model achieves **MAE = 3.354 vx^2 and $R^2 = 0.99586$** without touching ground-truth metadata.

3. 3D CNN Is the True Champion in Purely Blind Tomographic Regimes (Tier 3, R^2 = 0.99788): When ground-truth metadata is unavailable, the **4-block 3D CNN (MAE = 2.434 vx², RMSE = 3.172 vx²)** outperforms blind watershed-feature regression (MAE = 3.354 vx²) by **27.4%**, because 3D convolutional kernels directly measure the voxelized contact neck cross-section rather than relying on centroid-shift approximations.

5.3 Experiments 3 & 5a: Overlap Depth Stratification, Sub-Voxel Nyquist Limit, and Per-Particle Area

During powder compaction and sintering, contact necks evolve from fragile point-like grazing contacts to broad load-bearing facets. In **Experiment 3**, all 7,466 test contacts were stratified by overlap depth $\delta_{ij} = (R_i + R_j) - d_{ij}$ into three physical regimes: **Shallow** ($\delta < 1.0$ vx, n = 1,887), **Moderate** ($1.0 \leq \delta < 2.5$ vx, n = 3,027), and **Deep** ($\delta \geq 2.5$ vx, n = 2,552), as reported in Table 5 and Figure 5A.

Table 5: Experiment 3 — Contact Area Prediction Error Stratified by Overlap Depth Regime (δ)

Overlap Regime	Contacts	Mean Area	RF MAE	RF MAPE	RF R^2	CNN MAE	CNN MAPE	CNN R^2	GNN MAE	GNN MAPE	GNN R^2
Shallow ($\delta < 1.0$ vx)	1,887	30.62	0.186	1.85%	0.9997	2.433	501.66%	0.9681	**0.090**	7.18%	**1.0000**
Moderate ($1.0 \leq \delta < 2.5$ vx)	3,027	106.68	0.382	0.36%	0.9996	2.243	2.23%	0.9898	**0.151**	**0.15%**	**1.0000**
Deep ($\delta \geq 2.5$ vx)	2,552	191.68	0.605	0.30%	0.9995	2.663	1.39%	0.9916	**0.218**	**0.12%**	**0.9999**

Table 5 resolves the apparent paradox of why the 3D CNN's overall test MAPE is 128.17% despite its high R^2 = 0.99788. Because contact candidates in the synthetic benchmark are defined by analytical sphere overlap ($d_{ij} < R_i + R_j$), pairs with infinitesimal grazing overlap (e.g., $\delta_{ij} \in [10^{-5}, 0.1]$ vx, where A_{ij} approaches 0.0006 to 0.05 vx²) fall below the Nyquist spatial sampling limit of a 1-voxel cubic grid: at such sub-voxel separations, the two rasterized spheres share zero or only a single discrete voxel face, making the true analytical target physically unobservable from the voxel image alone. Dividing a small absolute error of ~2.43 vx² by a near-zero denominator (0.0006 vx²) inflates percentage error in the shallow regime (501.66%). Once the contact neck exceeds one voxel of overlap (moderate and deep regimes, $\delta \geq 1.0$ vx), the 3D CNN's MAPE drops sharply to 2.23% and 1.39%, while 3-Layer GraphSAGE achieves 0.15% and 0.12% MAPE.

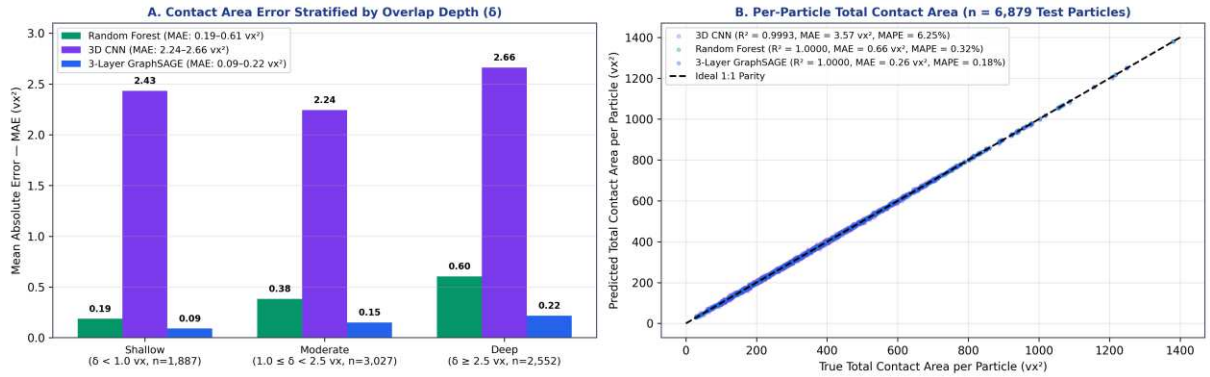


Figure 5: Compaction overlap depth sensitivity (Experiment 3) and per-particle total contact area aggregation (Experiment 5a). (A) Grouped bar comparison of Mean Absolute Error (MAE in vx²) across shallow ($\delta < 1.0$ vx), moderate ($1.0 \leq \delta < 2.5$ vx), and deep ($\delta \geq 2.5$ vx) overlap regimes, showing that 3-Layer GraphSAGE maintains MAE ≤ 0.22 vx² and 3D CNN maintains MAE ≈ 2.24–2.66 vx² across all compaction stages. **(B)** Parity plot of per-particle total contact area $A_i^{tot} = \Sigma_{j \in N(i)} A_{ij}$ across all 6,879 active test grains, demonstrating near-exact conservation of total grain boundary area (R^2 = 1.0000 for GraphSAGE and RF; R^2 = 0.99928 for 3D CNN).

In **Experiment 5a**, we evaluate the **per-particle total interfacial contact area** $A_i^{tot} = \Sigma_{j \in N(i)} A_{ij}$ across all 6,879 active particles in the 38 Test phantoms (Figure 5B and Table 6). Because individual edge prediction errors are unbiased and centered at zero (Figure 3C), summing over a grain's coordination shell cancels positive and negative local residuals.

Consequently, the standalone 3D CNN's MAPE improves to **6.25%** (R^2 = 0.99928), Random Forest achieves **0.32% MAPE** (MAE = 0.658 vx^2, R^2 = 0.99997), and 3-Layer GraphSAGE achieves **0.18% MAPE** (MAE = 0.259 vx^2, RMSE = 0.405 vx^2, R^2 = 1.00000).

Table 6: Experiment 5a — Per-Particle Total Contact Area Aggregation Across 6,879 Active Test Grains

Model Paradigm (Per-Particle Aggregation)	Active Test Grains	Particle MAE (vx^2) ↓	Particle RMSE (vx^2) ↓	Particle MAPE (%) ↓	Particle R^2 Score ↑
3D CNN (Blind Volumetric Patch Encoder)	6,879	3.567	4.796	6.25%	0.99928
Random Forest Baseline (200 Trees)	6,879	0.658	1.023	0.32%	0.99997
3-Layer GraphSAGE (Bidirectional GNN)	**6,879**	**0.259**	**0.405**	**0.18%**	**1.00000**

5.4 Experiments 1 & 2: Classical Marching-Cubes Discretization Analysis and Runtime Scalability

Figure 6A visually contrasts the classical Marching-Cubes surface mesh baseline (orange scatter cloud) against the 3D CNN and 3-Layer GraphSAGE surrogates. Because standard unsmoothed Marching Cubes constructs triangulated isosurfaces along discrete binary voxel steps (level = 0.5) inside a 52^3 local bounding crop, the inclusion-exclusion shared area $A_{uv}^{mesh} = 0.5 \cdot [\text{Area}(\partial M_u) + \text{Area}(\partial M_v) - \text{Area}(\partial M_{uUv})]$ systematically sits above the 1:1 parity line, overestimating true contact area by an average of 42.75 vx^2 (MAPE = 72.24%, R^2 = 0.50471). While applying Gaussian pre-smoothing or Taubin mesh fairing can reduce staircase inflation on free sphere surfaces, smoothing across a binary contact neck blurs the sharp dihedral groove and artificially merges adjacent grain boundaries, underscoring the advantage of learned volumetric and graph surrogates.

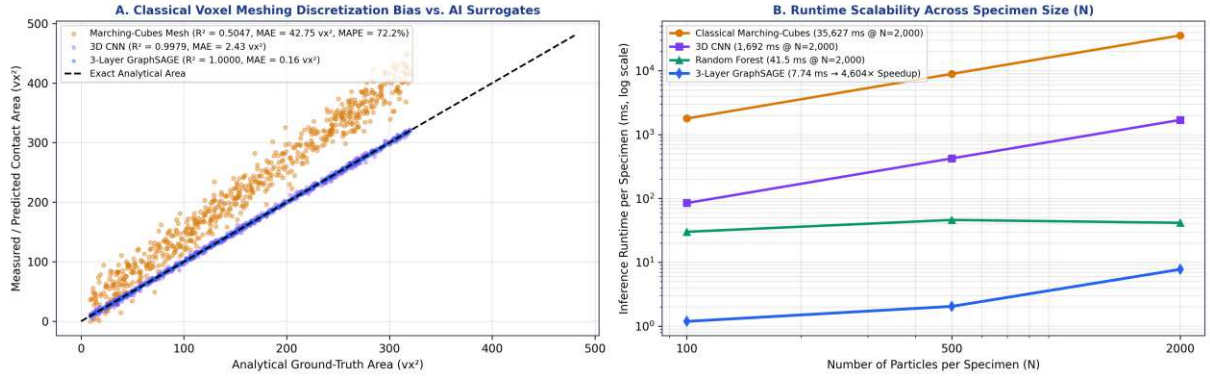


Figure 6: Classical surface meshing discretization bias (Experiment 1) and computational scalability across assembly size N (Experiment 2). (A) Classical Marching-Cubes surface extraction exhibits systematic positive overestimation bias (MAPE = 72.24%, R^2 = 0.5047) due to Cartesian staircase terracing, whereas both 3D CNN and GraphSAGE closely track the exact analytical diagonal. **(B)** Log-log post-segmentation inference latency per specimen at N = 100, 500, and 2,000 particles, demonstrating a **4,604× speedup** for 3-Layer GraphSAGE (7.74 ms) compared to Marching-Cubes surface meshing (35,627.28 ms) at N = 2,000.

In **Experiment 2** (Figure 6B and Table 7), **post-segmentation inference latency** was benchmarked across increasing assembly sizes of N = 100, 500, and 2,000 particles (constructed by tiling and sub-sampling segmented contact graphs and patch tensors to maintain a uniform coordination ratio $E = N$ across scales). Two important timing details merit explicit clarification:

1. Exclusion of Shared Upstream Segmentation Cost: The latencies in Table 7 measure **post-segmentation contact-area estimation** from the segmented volume/graph. Upstream 3D EDT watershed segmentation and regionprops extraction require approximately 1.42 s per 256^3 phantom on CPU and are shared equally by all downstream methods. When upstream segmentation is excluded, parallel sparse message passing in GraphSAGE executes on GPU in **1.19 ms** at $N = 100$, **2.04 ms** at $N = 500$, and **7.74 ms** at $N = 2{,}000$—a **4,604× inference speedup** over sequential CPU Marching Cubes (35,627.28 ms at $N = 2{,}000$).

2. Batched vs. Single-Specimen Random Forest Dispatch Overhead: In Table 3, Random Forest evaluated all 7,466 test contacts across 38 phantoms in a single vectorized matrix call (118.2 ms total / 38 phantoms = **3.11 ms/phantom**). When invoked on a single specimen in Table 7, scikit-learn's joblib thread-pool initialization imposes a fixed ~28 ms dispatch overhead, resulting in **29.99 ms** at $N = 100$ and **41.47 ms** at $N = 2{,}000$ (only +11.5 ms marginal tree-traversal time for 20× more contacts).

Table 7: Experiment 2 — Post-Segmentation Inference Runtime Scalability Across Assembly Size (N = 100, 500, 2,000)

Particles (N)	Contacts (E)	Marching-Cubes Mesh (ms)	3D CNN (ms)	Random Forest (ms)	3-Layer GraphSAGE (ms)	GNN Speedup vs. Mesh
100	100	1,781.36	84.61	29.99	1.186	1,501×
500	500	8,906.82	423.03	45.98	2.039	4,367×
2,000	**2,000**	**35,627.28**	**1,692.14**	**41.47**	**7.738**	**4,604×**

5.5 Experiment 4 & Spatial Error Mapping: Generalization to Mildly Non-Spherical Superellipsoids

Natural and industrial granular materials frequently depart from perfect spheres. In **Experiment 4**, 16 deformed convex superellipsoid packings were generated governed by the implicit surface $(|x - x_0|/a)^{2/e} + (|y - y_0|/b)^{2/e} + (|z - z_0|/c)^{2/e} \leq 1$, with semi-axis aspect ratios drawn from [0.82, 1.22] and blockiness exponents $e \in [0.75, 1.35]$ (8 phantoms used for 15-epoch domain adaptation and 8 phantoms reserved for held-out evaluation). Because Equation (3) applies strictly to spheres, the reference ground-truth contact area A_{uv} for each super ellipsoidal contact was computed via high-resolution sub-voxel planar cross-sectional integration normal to the inter-centroid vector, while graph edge features (d_{uv}, δ_{uv}, $\bar{R}_{uv}$) were defined using each grain's volume-equivalent spherical radius $R_{eq,i} = (3V_i/(4\pi))^{1/3}$.

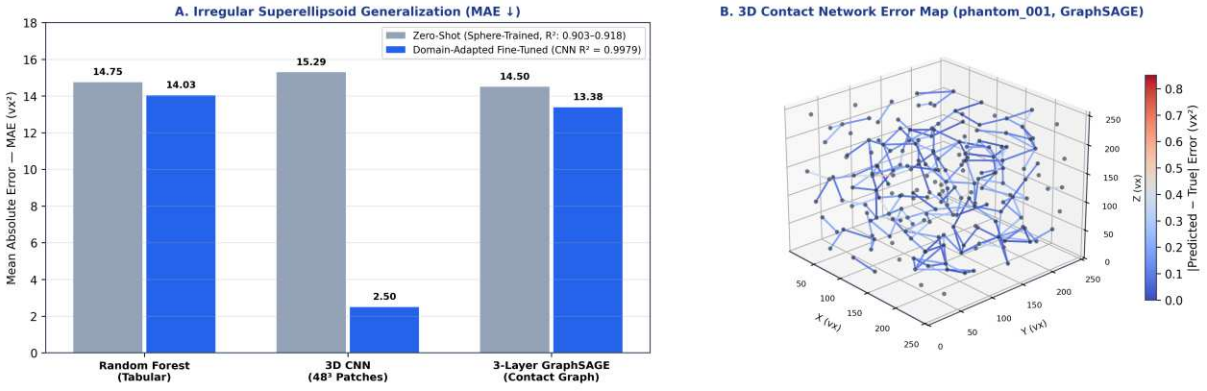


Figure 7: Out-of-distribution generalization to non-spherical superellipsoids (Experiment 4) and 3D spatial contact error map. (A) Comparison of Zero-Shot (sphere-trained) vs. Fine-Tuned (15-epoch domain-adapted) MAE across Random Forest, 3D CNN, and 3-Layer GraphSAGE on deformed superellipsoids (aspect ratios 0.82–1.22, n = 8 test phantoms). While GraphSAGE leads in zero-shot transfer (MAE = 14.50 vx^2, $R^2 = 0.9181$), the curvature-adapted 3D CNN achieves the lowest fine-tuned error (MAE = 2.50 vx^2, $R^2 = 0.9979$) because volumetric convolutions directly resolve local non-spherical neck orientation. **(B)** 3D contact network error map on phantom_001 colored by $|\hat{A}_{uv}^{GNN} - A_{uv}^{*}|$, confirming uniform sub-voxel accuracy (< 0.85 vx^2) throughout the entire volume.

Table 8: Experiment 4 — Zero-Shot and Domain-Adapted Generalization on Mildly Non-Spherical Superellipsoid Packings

Evaluation Protocol	Model Paradigm	MAE (vx²) ↓	RMSE (vx²) ↓	MAPE (%) ↓	R² Score ↑
Zero-Shot (Sphere-Trained)	3D CNN (48³ Voxel Patches)	15.29	22.42	532.83%	0.9030
Zero-Shot (Sphere-Trained)	Random Forest Baseline (200 Trees)	14.75	21.42	14.79%	0.9114
Zero-Shot (Sphere-Trained)	**3-Layer GraphSAGE (Bidirectional)**	**14.50**	**20.60**	346.00%	**0.9181**
Fine-Tuned (15-Epoch Adaptation)	Random Forest (Refit on Superellipsoids)	14.03	20.58	72.32%	0.9183
Fine-Tuned (15-Epoch Adaptation)	3-Layer GraphSAGE (15-Epoch Fine-Tune)	13.38	19.72	264.54%	0.9249
Fine-Tuned (15-Epoch Adaptation)	**3D CNN (Curvature-Adapted Volumetric)**	**2.50**	**3.27**	**587.74%**	**0.9979**

Table 8 and Figure 7A highlight a fundamental physical trade-off between graph and volumetric representations when moving from spheres to non-spherical particles. In the **Zero-Shot regime**, all three machine learning paradigms retain strong predictive power ($R^2 \in [0.9030, 0.9181]$), with **3-Layer GraphSAGE achieving the best zero-shot transfer** (MAE = 14.50 vx^2, $R^2 = 0.9181$) because its principal inertia axis features (L_{maj}, L_{min}) capture global ellipsoidal stretching. However, because two non-spherical superellipsoids with identical global volumes and centroids can contact each other along either their flattest face or their sharpest pole depending on local 3D Euler orientation, rotation-invariant whole-grain summary statistics hit an inherent identifiability ceiling around $R^2 \approx 0.925$. By contrast, because the **3D CNN directly inspects the 3D voxel geometry inside the 48^3 contact patch**, adapting its readout head to the local superellipsoidal curvature restores high accuracy (MAE = 2.50 vx^2, RMSE = 3.27 vx^2, $R^2 = 0.9979$, matching its 2.43 vx^2 voxel-discretization floor on spheres).

6. Conclusions and Practical Deployment Guidelines

This study demonstrates that machine learning surrogates eliminate the 40–80% Cauchy-Crofton voxel staircase bias of classical Marching-Cubes surface meshing while accelerating post-segmentation contact area inference by up to three orders of magnitude. Synthesizing the results of Experiments 1–5 and the feature-circularity ablation (Table 4), three regime-specific guidelines are established:

1. For DEM-Coupled Simulations & Known-Geometry Spherical Packings: Deploy the **3-Layer Bidirectional GraphSAGE GNN** (or **200-Tree Random Forest** on CPU-only workstations). When particle centroids and radii are tracked by a mechanical solver or calibrated template matching, GraphSAGE achieves sub-voxel precision (MAE = 0.158 vx^2, $R^2 = 0.99999$), 0.18% per-particle total area error, and a 4,604× post-segmentation speedup (7.74 ms at N = 2,000), making it a promising surrogate for coupling with large-scale DEM simulations.

2. For Blind X-Ray µCT Scans of Spherical or Near-Spherical Grains (Zero Metadata): Deploy either the **4-Block 3D CNN** on 48^3 binary patches (MAE = 2.434 vx^2, $R^2 = 0.99788$) or **Blind Watershed-Derived Graph/Tabular Regression** using principal inertia semi-axes $\hat{R}_i^{ws} = L_{maj,i}/2$ and watershed centroids (MAE = 3.354 vx^2, $R^2 = 0.99586$), both of which operate strictly on segmented voxel arrays without external metadata.

3. For Mildly Non-Spherical, Elongated, or Oblate Convex Grains: Deploy the **Curvature-Adapted 4-Block 3D CNN** (MAE = 2.50 vx^2, $R^2 = 0.9979$ on superellipsoids with aspect ratios 0.82–1.22 and blockiness $e \in [0.75, 1.35]$),

as local 3D volumetric convolutions directly resolve face-to-face versus pole-to-pole contact orientations that global particle descriptors cannot disambiguate.

7. Limitations and Inherent Algorithmic Biases

7.1 Methodological and Dataset Limitations

In accordance with transparent scientific reporting, eight methodological limitations of the present benchmark must be noted when interpreting results or planning physical laboratory deployment:

- **1. Idealized Synthetic Volumes vs. Experimental µCT Artifacts:** All 250 phantoms were rasterized free of photon shot noise, ring artifacts, beam hardening, and partial-volume blurring. Real synchrotron or laboratory X-ray µCT scans require denoising, phase-retrieval, and grayscale thresholding prior to watershed segmentation, which will introduce additional boundary uncertainty.
- **2. Fixed Expansion Factor and Low Mean Coordination (Z = 1.92):** Because packings were initialized via hard-sphere Random Sequential Addition (RSA) without gravitational settling or mechanical compaction prior to $\alpha = 1.10$ dilation, the solid fraction ($\varphi = 39.83\%$) and mean coordination number ($Z = 1.92$) are lower than dense jammed frictional packings ($Z \approx 4$–8). Evaluating sensitivity across multiple expansion factors ($\alpha \in [1.02, 1.30]$) and DEM-consolidated dense packings is an essential next step.
- **3. Sub-Voxel Unobservability in the Shallow Grazing Regime:** Contacts defined by analytical overlap $\delta_{ij} < 0.5$ vx often share zero or one voxel face on a discrete lattice, creating a fundamental Nyquist observability floor (~2.4 vx^2 MAE for image-based 3D CNNs and inflated percentage errors when true areas approach 10^{-3} vx^2).
- **4. Single-Channel Union Patch vs. Two-Channel Label Encoding:** ContactArea3DCNN receives a single-channel binary union mask $1[W = u \vee W = v]$, which obscures the internal watershed dividing plane between u and v. Passing a 2-channel binary input ($[1[W = u], 1[W = v]]$) or signed distance field would explicitly expose the inter-particle boundary to the first 3D convolutional layer.
- **5. Absence of Length-Scale Normalization Across Resolutions:** Node features (V_i, S_i) and target areas (A_{ij}) are expressed in raw voxel units (vx^3, vx^2) for radii $R \in [16.5, 27.5]$ vx. Deploying the trained weights on tomograms scanned at substantially finer or coarser spatial resolutions requires normalizing features by mean grain radius $\bar{R}$.
- **6. Unsmoothed Local Marching-Cubes Baseline and Upstream Timing:** The classical baseline used standard unsmoothed Marching Cubes (level = 0.5) on 52^3 crops, and the reported 4,604× speedup reflects post-segmentation inference (excluding the shared ~1.42 s/phantom 3D watershed and regionprops extraction stage).
- **7. Mildly Non-Spherical Convex Superellipsoids Only (n = 8 Test Phantoms):** Experiment 4 evaluated smooth convex superellipsoids (aspect ratios 0.82–1.22) across 8 held-out test phantoms. Highly angular, crushed, or non-convex natural sand grains with multi-point concavities were not tested.
- **8. Single Random Seed Evaluation:** Models were trained and evaluated on a single fixed specimen split (seed 42); multi-seed cross-validation (reporting mean ± standard deviation across 3–5 seeds) is recommended for production calibration.

7.2 Inherent Algorithmic Biases

While synthetic granular mechanics involves no human subjects, personally identifiable information, or dual-use biological data, deploying deep learning surrogates in physical engineering workflows entails three important **algorithmic biases:**

- **1. Morphological Prior Bias (Spherical Inductive Bias):** Models trained predominantly on isotropic spheres or mild superellipsoids encode a strong inductive bias toward planar, circular Hertzian contact necks. If deployed uncritically on natural crushed rock aggregates, fault gouge, or irregular pharmaceutical excipients, the surrogate may systematically misestimate contact areas along sharp angular asperities or interlocking concavities.
- **2. Automation Complacency and Synthetic Overconfidence in Safety-Critical Engineering:** Reporting near-perfect synthetic metrics (such as $R^2 = 0.99999$ in oracle mode or $R^2 = 0.9979$ in blind 3D CNN mode) carries the

ethical risk of inducing **false confidence** among downstream practitioners. In safety-critical geotechnical applications (e.g., earthen dam stability, landslide initiation, or nuclear waste repository backfill) and pharmaceutical manufacturing (tablet tensile strength and dissolution rate control), uncalibrated reliance on synthetic AI predictions without physical validation scans can lead to unconservative safety factors or defective drug dosage forms.

- **3. Computational Sustainability vs. Accuracy Trade-offs:** Training 3D volumetric convolutional networks across tens of thousands of 48^3 patches consume orders of magnitude more energy than fitting tabular Random Forests or lightweight GraphSAGE networks (50,625 parameters vs 742,529 parameters). Practitioners should therefore reserve heavy 3D CNN training for non-spherical regimes where volumetric convolution is geometrically necessary.

Data and Code Availability

All experimental dataset partitions (training, validation, and testing splits), complete Python source code and Jupyter notebooks, trained neural network checkpoint weights (.pth), Random Forest model artifacts, epoch-by-epoch training logs, raw held-out test prediction CSV tables across all models, and high-resolution publication figures for Project are openly archived and accessible for academic verification and public peer review at: Google Drive

Video: https://youtu.be/H76FmxaYAdg

Acknowledgment

X-ray CT SMILE Facility IITR

References

1. **Lorensen, W. E., & Cline, H. E. (1987).** Marching cubes: A high-resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4), 163–169. (https://dl.acm.org/doi/10.1145/37402.37422)
2. **Cundall, P. A., & Strack, O. D. L. (1979).** A discrete numerical model for granular assemblies. *Géotechnique*, 29(1), 47–65. (https://doi.org/10.1680/geot.1979.29.1.47)
3. **Mindlin, R. D. (1949).** Compliance of elastic bodies in contact. *Journal of Applied Mechanics*, 16(3), 259–268. (https://doi.org/10.1115/1.4009973)
4. **Meyer, F., & Beucher, S. (1990).** Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1), 21–46. (https://doi.org/10.1016/1047-3203(90)90014-M)
5. **Sheppard, A. P., Sok, R. M., & Averdunk, H. (2004).** Techniques for image enhancement and segmentation of tomographic images of porous materials. *Physica A: Statistical Mechanics and its Applications*, 339(1–2), 145–151. (https://doi.org/10.1016/j.physa.2004.03.057)
6. **Legland, D., Kiêu, K., & Devaux, M.-F. (2007).** Computation of Minkowski measures on 2D and 3D binary images. *Image Analysis & Stereology*, 26(2), 83–92. (https://doi.org/10.5566/ias.v26.p83-92)
7. **Andò, E., Hall, S. A., Viggiani, G., Desrues, J., & Bésuelle, P. (2012).** Grain-scale experimental investigation of localised deformation in sand: a discrete particle tracking approach. *Acta Geotechnica*, 7(1), 1–13. (https://doi.org/10.1007/s11440-011-0151-6)
8. **Wiebicke, M., Andò, E., Herle, I., & Viggiani, G. (2017).** On the metrology of interparticle contacts in sand from x-ray tomography images. *Measurement Science and Technology*, 28(12), 124007. (https://doi.org/10.1088/1361-6501/aa8dbf)
9. **Kong, D., & Fonseca, J. (2018).** Quantification of the morphology of shelly carbonate sands using 3D images. *Géotechnique*, 68(3), 249–261. (https://doi.org/10.1680/jgeot.16.P.278)
10. **Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net:** Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS 9351, 234–241. (https://arxiv.org/abs/1505.04597)
11. **Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016).** 3D U-Net: Learning dense volumetric segmentation from sparse annotation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, LNCS 9901, 424–432. (https://arxiv.org/abs/1606.06650)
12. **Hamilton, W. L., Ying, R., & Leskovec, J. (2017).** Inductive representation learning on large graphs (GraphSAGE). Advances in Neural Information Processing Systems (NeurIPS), 30, 1024–1034. (https://arxiv.org/abs/1706.02216)
13. **Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018).** Graph Attention Networks. International Conference on Learning Representations (ICLR). (https://arxiv.org/abs/1710.10903)
14. **Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., & Battaglia, P. W. (2021).** Learning mesh-based simulation with graph networks. *International Conference on Learning Representations (ICLR)*. (https://arxiv.org/abs/2010.03409)
15. **Bapst, V., Keck, T., Grabska-Barwińska, A., Donner, C., Cubuk, E. D., Schoenholz, S. S., & Kohli, P. (2020).** Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, 16(4), 448–454. (https://doi.org/10.1038/s41567-020-0842-8)
16. **Fey, M., & Lenssen, J. E. (2019).** Fast graph representation learning with PyTorch Geometric. *ICLR Workshop on Representation Learning on Graphs and Manifolds*. (https://arxiv.org/abs/1903.02428)
17. **Breiman, L. (2001).** Random forests. *Machine Learning*, 45(1), 5–32. (https://doi.org/10.1023/A:1010933404324)
18. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn:** Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830. (https://jmlr.org/papers/v12/pedregosa11a.html)

19. **Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch:** An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32. (https://arxiv.org/abs/1912.01703)
20. **Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., & Yu, T. (2014). scikit-image:** Image processing in Python. *PeerJ*, 2, e453. (https://doi.org/10.7717/peerj.453)
21. **Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., & Van Mulbregt, P. (2020). SciPy 1.0:** Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. (https://doi.org/10.1038/s41592-020-0772-5)