Boundary Graph Neural Networks for 3D Simulations

Andreas Mayr[1] Sebastian Lehner[1] Arno Mayrhofer[2] Christoph Kloss[2] Sepp Hochreiter[1,3] Johannes Brandstetter[1,*]

[1] ELLIS Unit Linz & LIT AI Lab, Johannes Kepler University Linz, Linz, Austria [2] DCS Computing GmbH, Linz, Austria [3] Institute of Advanced Research in Artificial Intelligence (IARAI), Vienna, Austria
[*] now at Microsoft Research AI4Science

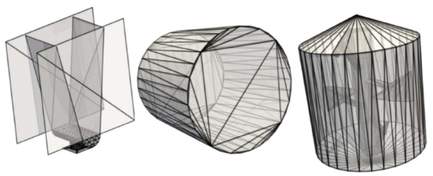
Abstract

The abundance of data has given machine learning considerable momentum in natural sciences and engineering, though modeling of physical processes is often difficult. A particularly tough problem is the efficient representation of geometric boundaries. Triangularized geometric boundaries are well understood and ubiquitous in engineering applications. However, it is notoriously difficult to integrate them into machine learning approaches due to their heterogeneity with respect to size and orientation. In this work, we introduce an effective theory to model particle-boundary interactions, which leads to our new Boundary Graph Neural Networks (BGNNs) that dynamically modify graph structures to obey boundary conditions. The new BGNNs are tested on complex 3D granular flow processes of hoppers, rotating drums and mixers, which are all standard components of modern industrial machinery but still have complicated geometry. BGNNs are evaluated in terms of computational efficiency as well as prediction accuracy of particle flows and mixing entropies. BGNNs are able to accurately reproduce 3D granular flows within simulation uncertainties over hundreds of thousands of simulation timesteps. Most notably, in our experiments, particles stay within the geometric objects without using handcrafted conditions or restrictions.

Problem Statement & Main Contributions

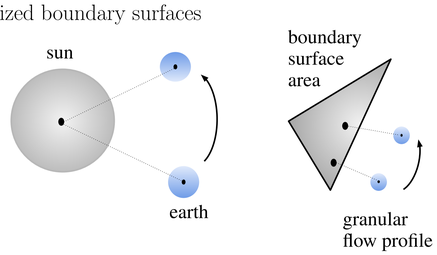
Machine Learning for Physical Processes:

- Relevant for natural sciences and engineering
- Run-time intensive computations
- Research to speedup computations
- Boundary conditions relevant for dynamics
- Triangular meshes as boundary description



Goal: Development of an Effective Theory for modeling particle-boundary interaction within triangularized boundary surfaces

Basic idea:
Approximate physical system by factoring out degrees of freedom not relevant in the given setting and problem



GNNs for Simulation Dynamics

Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Nodes $v_i \in \mathcal{V}$ with node features $\boldsymbol{p}_{v_i} \in \mathbb{R}^N$
Edges $e_{ij} \in \mathcal{E}$ with edge features $\boldsymbol{a}_{ij} \in \mathbb{R}^M$ between a pair of nodes (v_i, v_j)

Nearest neighbor graphs: local interactions to build arbitrary global dynamics

$$e_{ij} \in \mathcal{E} \iff d(v_i, v_j) \leqslant r_{\text{cut-off}}$$

Message passing:

$$\boldsymbol{m}'_{ij} = \phi(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{m}_{ij}), \qquad \boldsymbol{h}'_i = \psi\Big(\boldsymbol{h}_i, \square_{e_{ij} \in \mathcal{E}}\, \boldsymbol{m}'_{ij}\Big)$$

with aggregation $\square_{e_{ij} \in \mathcal{E}}$ at node v_i across all nodes connected to v_i via e_{ij}

Time transition model [1] from time t to time $t+1$ given by:

$$\dot{\boldsymbol{x}}^{t+1} = \dot{\boldsymbol{x}}^t + \Delta t\ \ddot{\boldsymbol{x}}^t$$
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \Delta t\ \dot{\boldsymbol{x}}^{t+1}$$

$\boldsymbol{x}$ ... particle location
$\dot{\boldsymbol{x}}$ ... particle velocity
Δt fixed to 1
prediction of acceleration $\ddot{\boldsymbol{x}}^t$ with a GNN

Construction of BGNNs: Modification of Graph

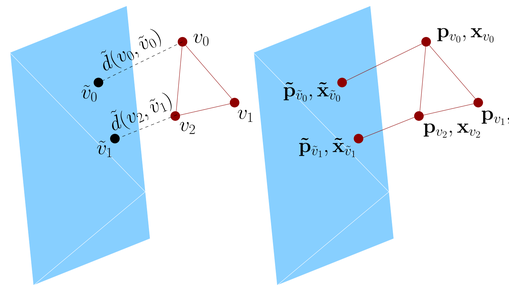
Dynamically add $\tilde{n}$ virtual nodes $\tilde{v}_j \in \tilde{\mathcal{V}}$ for boundary regions, iff the corresponding boundary region is within a cut-off radius to any other particle

Augment the set of edges $e_{ij} \in \mathcal{E}$ by boundary edges $\tilde{e}_{ij} \in \tilde{\mathcal{E}}$ giving an enhanced edge set $\hat{\mathcal{E}} = \mathcal{E} \cup \tilde{\mathcal{E}}$

Particle-boundary edges $\tilde{e}_{ij}$ defined via:

$$e_{ij} \in \mathcal{E} \subseteq \hat{\mathcal{E}} \iff d(v_i, v_j) \leqslant r_{\text{cut-off}}$$
$$\tilde{e}_{ij} \in \tilde{\mathcal{E}} \subseteq \hat{\mathcal{E}} \iff \tilde{d}(v_i, \tilde{v}_j) \leqslant \tilde{r}_{\text{cut-off}}$$

with $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and $\tilde{d} : \mathcal{V} \times \tilde{\mathcal{V}} \to \mathbb{R}$



Extended node and edge features:

- $\hat{\boldsymbol{P}} = \{\boldsymbol{p}_{v_0}, \ldots, \boldsymbol{p}_{v_{n-1}}, \tilde{\boldsymbol{p}}_{\tilde{v}_0}, \ldots, \tilde{\boldsymbol{p}}_{\tilde{v}_{\tilde{n}-1}}\}$
- $\hat{\boldsymbol{X}} = \{\boldsymbol{x}_{v_0}, \ldots, \boldsymbol{x}_{v_{n-1}}, \tilde{\boldsymbol{x}}_{\tilde{v}_0}, \ldots, \tilde{\boldsymbol{x}}_{\tilde{v}_{\tilde{n}-1}}\}$
- $\tilde{N}$-dimensional node features $\tilde{\boldsymbol{p}}_{\tilde{v}_i}$ encode information about triangles
- $\tilde{\boldsymbol{x}}_{\tilde{v}_i}$ chosen to minimize distance between boundary points and real particles
- $\hat{\boldsymbol{p}}_i$ with default values for missing features
- $\hat{\boldsymbol{p}}_i \in \mathbb{R}^{N+\tilde{N}}$, $\hat{\boldsymbol{x}}_i \in \mathbb{R}^3$ denote elements of $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{X}}$

Experiment	$\lvert\mathcal{V}\rvert$	$\lvert\tilde{\mathcal{E}}\rvert$	% increase
Hopper	1113 ± 738	5475 ± 3547	72.2
Drum	3283 ± 282	1678 ± 188	54.8

Dynamic Graph Message Passing:

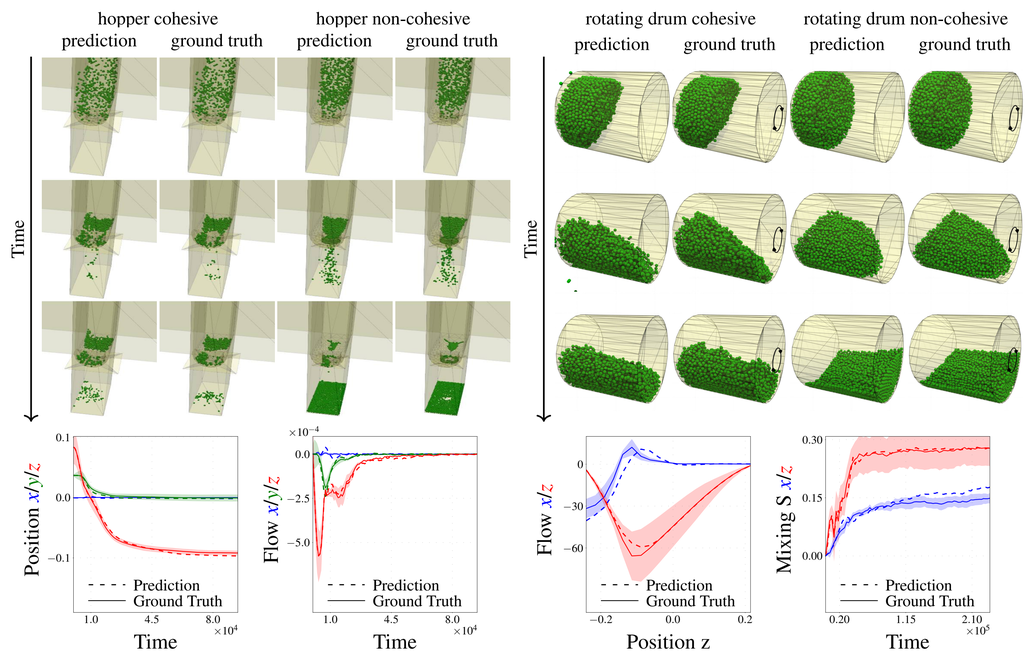
$$\hat{\boldsymbol{m}}'_{ij} = \hat{\phi}\Big(\hat{\boldsymbol{h}}_i, \hat{\boldsymbol{h}}_j, \hat{\boldsymbol{m}}_{ij}\Big)$$
$$\hat{\boldsymbol{h}}'_i = \hat{\psi}\Big(\hat{\boldsymbol{h}}_i, \square_{\hat{e}_{ij} \in \hat{\mathcal{E}}}\, \hat{\boldsymbol{m}}'_{ij}\Big)$$

aggregation $\square_{\hat{e}_{ij} \in \hat{\mathcal{E}}}$ at v_i across all real or virtual nodes connected via an edge $\hat{e}_{ij}$

usage of pairwise distances: $\big\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\big\|^2$, $\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j$, and, deterministic functions thereof as edge attributes $\hat{\boldsymbol{a}}_{ij}$

Experimental Evaluation: Learning Simulations with 2 Materials

Granular flow simulations are obtained by an Discrete Element Method (DEM) [2] with simulator LIGGGHTS [3]



Computation of Samllest Distances

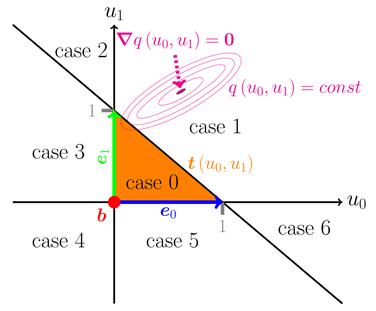
Efficient GPU implementation [4]:

- Triangle parameterized by $\boldsymbol{t}$ u_0, $u_1 \in \mathbb{R}$ with

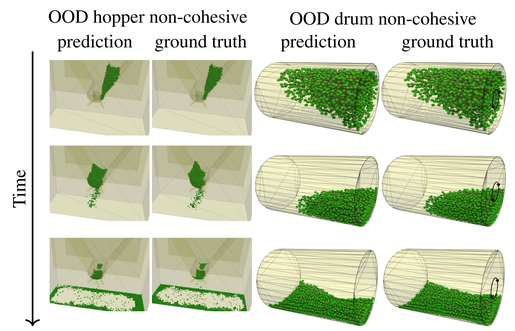
$$\boldsymbol{t}(u_0, u_1) = \boldsymbol{b} + u_0\, \boldsymbol{e}_0 + u_1\, \boldsymbol{e}_1$$

  where $u_0 \geq 0$, $u_1 \geq 0$, and $u_0 + u_1 \leqslant 1$
- Minimal Euclidean squared distance d of the point $\boldsymbol{p}$ to triangle given by optimization problem:

$$d = \min_{u_0, u_1} q(u_0, u_1) = \|\boldsymbol{t}(u_0, u_1) - \boldsymbol{p}\|^2$$
$$\text{s.t.} \quad u_0 \geq 0\,, \quad u_1 \geq 0\,, \quad u_0 + u_1 \leqslant 1$$



OOD Generalization



domain	cohesive μ	cohesive σ	non-cohesive μ	non-cohesive σ
in-distribution	0.34	0.09	0.89	0.03
OOD	0.14	0.11	0.73	0.15

Comparison of the proportion (mean μ and std σ) of particles beyond the outlet of the hopper

Runtime Comparison

method	time steps	real world time	wall-clock time [s]
LIGGGHTS	250.000	12.5s	356
BGNNs	2.500	12.5s	158

[1] Sanchez-Gonzalez A. et al. Learning to Simulate Complex Physics with Graph Networks. 2020.
[2] Cundall P.A. and Strack O.D.L. A discrete numerical model for granular assemblies. 1979.
[3] Kloss C. et al. Models, algorithms and validation for opensource DEM and CFD–DEM. 2012.
[4] Eberly D. Distance Between Point and Triangle in 3D. Technical report, 1999.