Momentum-Conserving Graph Neural Networks for Deformable Objects

Appendix

A. Completeness of Per-Edge Momentum Basis

We show that per-edge momentum basis is complete in the sense that any momentum-preserving impulse can be generated in this way.

To this end, we assume that there exist momentum-conserving impulses impulses $\mathbf{q}$ that cannot be generated using the basis $\mathbf{B}$, i.e.,

$$\sum_i \mathbf{q}_i = \mathbf{0}\,, \quad \sum_i (\mathbf{q}_i \times \mathbf{c}) = \mathbf{0}\,, \text{ and } \mathbf{B}\mathbf{B}^T\mathbf{q} - \mathbf{q} = \mathbf{b} \neq \mathbf{0}\,, \tag{A27}$$

where $\mathbf{c}$ is an arbitrary reference point. Without loss of generality, we consider a two-dimensional example and show by induction that $\mathbf{B}$ always has rank $2|V| - 3$ with the three-dimensional null-space of $\mathbf{B}^T$ corresponding to rigid transformations. To see this, consider a single triangle with six nodal degrees of freedom. The three gradients of edge length are linearly independent vectors and thus form a three-dimensional subspace of $\mathbb{R}^6$. Since translations and (first-order) rotations are orthogonal to these gradients, the set of vectors $\mathbf{v}$ for which $\mathbf{B}^T\mathbf{v} = \mathbf{0}$ is a three-dimensional null-space corresponding to (first-order) rigid transformations. Now consider adding a vertex to an existing triangle mesh. This vertex adds two degrees of freedom, but we must also add at least two additional edges to maintain a proper triangulation. The two additional edges yield two linearly independent basis vectors because the corresponding edges are not collinear. Since the new basis vectors involve the new vertex, they are also linearly independent of all pre-existing basis vectors. Consequently, the output dimension of $\mathbf{B}$ increases by 2, but so does its rank, implying that the dimensionality and structure of the null-space remains unchanged.

Using this observation, it is clear that either $\mathbf{B}^T\mathbf{b} \neq \mathbf{0}$ or that $\mathbf{b}$ corresponds to a first-order rigid transformation and therefore changes momentum. We therefore conclude that all momentum-conserving impulses can be expressed with the per-edge basis $\mathbf{B}$.

B. Discussion on method

Impulse Basis from Intrinsic Properties instead of Energy We prefer to think in terms of intrinsic properties such as edge lengths and dihedral angles because their gradients are always well-defined and nonzero. Elastic energies due to stretching and bending build on these intrinsic properties but are at least quadratic functions of them, such that their gradients vanish at rest (when there is no deformation), resulting in degenerate impulse directions.

Momentum Conservation and Dissipation Our method predicts nodal updates for a momentum step that preserve total linear and angular momentum. It is important to note that momentum conservation does not imply the conservation of energy. To see this, consider a discretized elastic bar that is stretched along its axis and then released. As the bar oscillates, its total linear momentum remains zero— regardless of the amplitude oscillation. It is evident that scaling all nodal velocities by a given constant does not change the total momentum. However, the kinetic energy of the system can be changed arbitrarily in this way. Since momentum-conserving impulses can dissipate energy, our method learns the numerical damping inherent to implicit Euler even when using a purely elastic material.

Implicit Euler as Basis It is well established that implicit Euler conserves neither energy nor momentum. While this seemingly disqualifies implicit Euler as a basis for our momentum-preserving GNN, its stability properties nevertheless make it a very attractive choice. For example, while symplectic Euler conserves momentum by construction, it has poor stability properties. Similarly, the implicit midpoint scheme preserves momentum but does not enjoy the same stability properties as implicit Euler. By building our loss function on implicit Euler, our GNN learns to predict momentum-preserving corrections that lead to stable behavior even for very long roll-outs.

C. Additional results

C.1. Bouncing Tennis Ball

We extend our evaluation by simulating a tennis ball dropped onto a table. As shown in Figure A1, our approach closely matches the behavior of implicit Euler, generating realistic bouncing behavior. Additionally, it accurately captures the damping effects observed in the reference simulation, resulting in a gradual reduction of rebound amplitude over time. By contrast, MeshGraphNets is not able to generate physically plausible motion for the ball. Incorporating velocity projection improves the trajectory, but the ball ultimately bounces off the table due to spurious momentum changes.

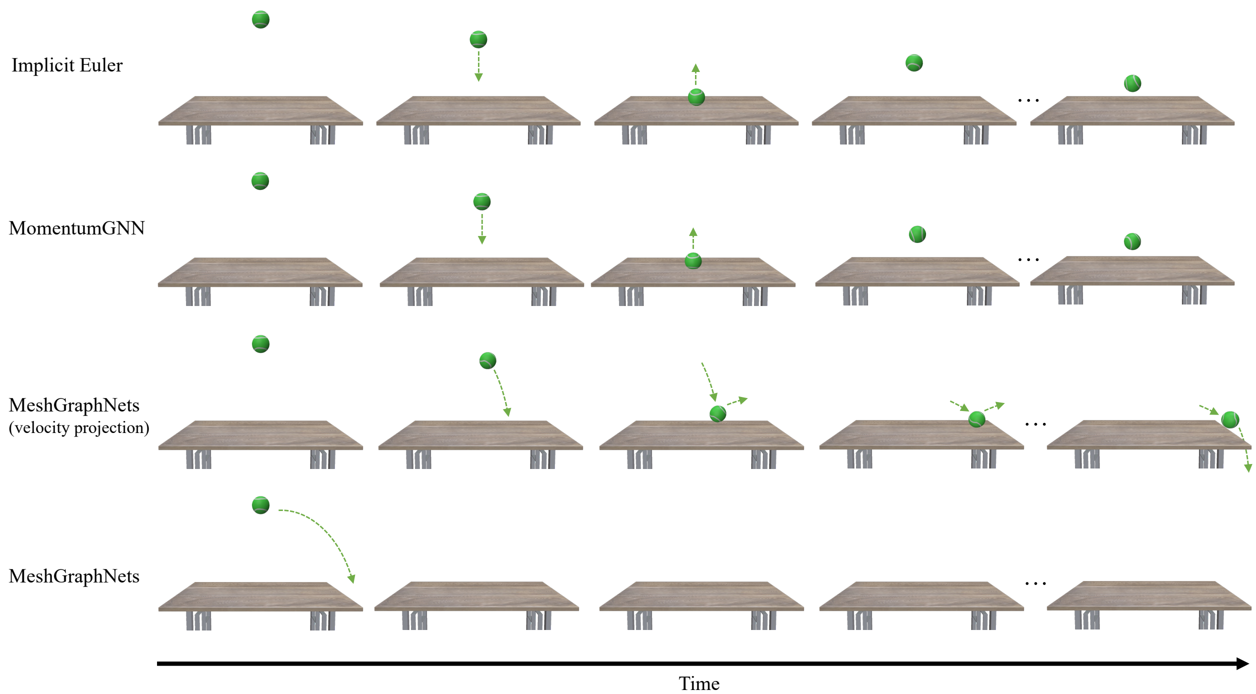

Figure A1. We simulate a bouncing tennis ball on a table. While MomentumGNN and Implicit Euler produce natural bouncing motions, both MeshGraphNets drifts the ball off the table.