Content

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

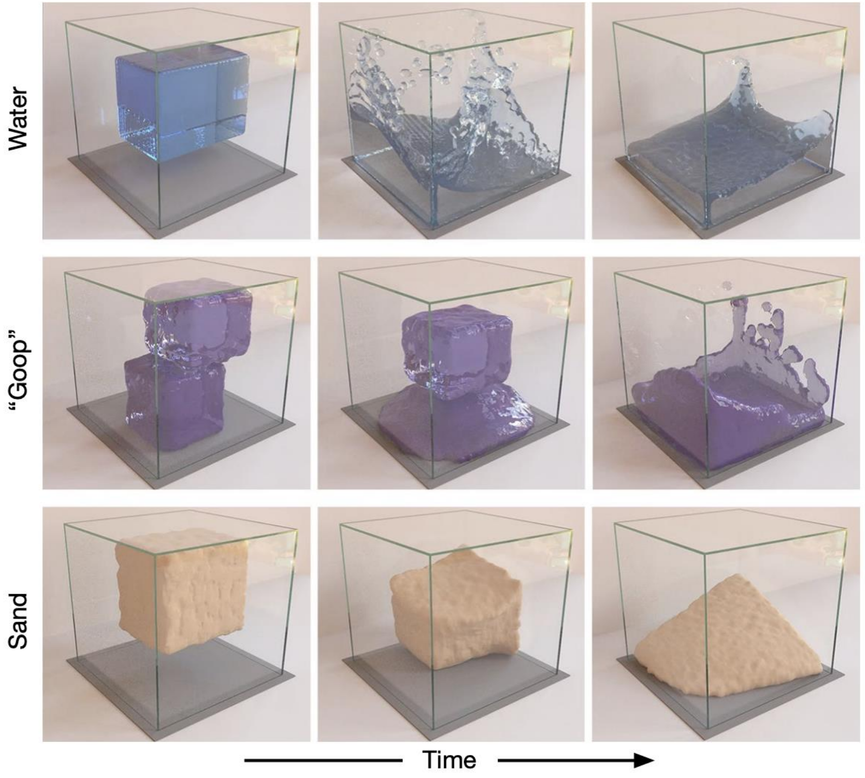
- Introduction & Related Work
  - Data-driven Physics Simulations
  - Graph-based Learning
  - Position-based Dynamics and Elastic Rods
- Methodology
  - Overview
  - Graph Encoding
  - Network Structure
- Evaluation
  - Data Generation & Ablation Study
  - Spatial Discretization
  - Temporal Evolution & Long-term Stability.
  - Complex Scenarios

Introduction & Related Work

- Data-driven Physics Simulations

Jonathan Tompson et al. 2016

A machine learning approach to directly update the state of the system or substiute part of the algorithm.

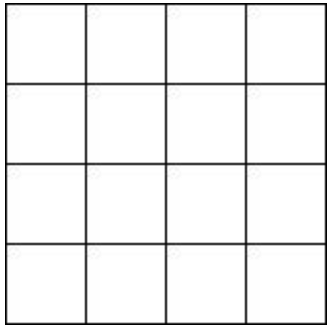


Alvaro Sanchez et al. 2020

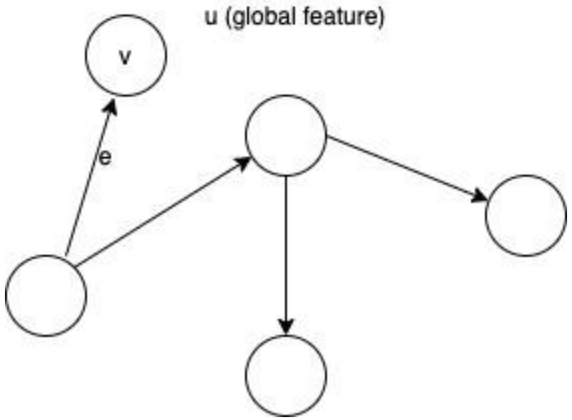
Introduction & Related Work

King Abdullah University of Science and Technology

- Graph-based Learning (Graph Network)



Cartesian Structure



Graph Structure

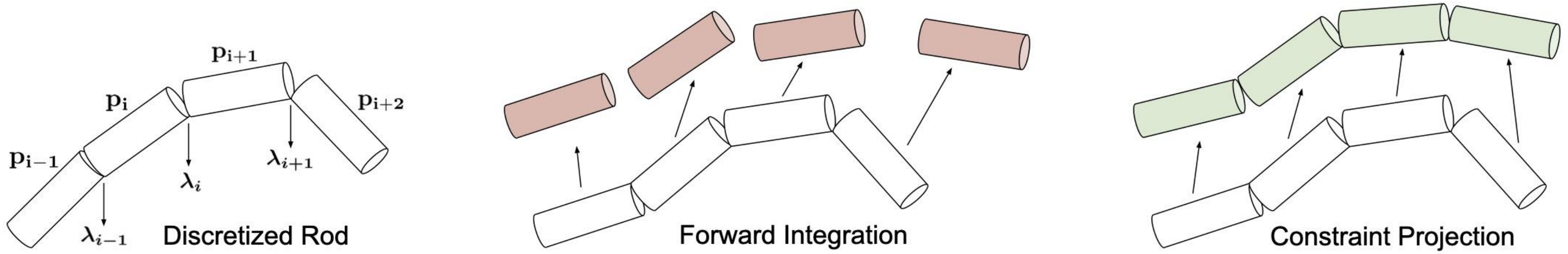
Graph Network: A mapping from G(V, E, U) to G'(V', E', U')

Peter W. Battaglia et al. 2018

function GraphNetwork($\mathbf{V}, \mathbf{E}, \mathbf{U}, \mathbf{sd}, \mathbf{rv}$)
  for $i \leftarrow 1$ to $|\mathbf{E}|$ **do**
    $\mathbf{v}_s$ = BroadCastVtoE($\mathbf{V}, \mathbf{sd}, i$)
    $\mathbf{v}_r$ = BroadCastVtoE($\mathbf{V}, \mathbf{rv}, i$)
    $\mathbf{u}$ = BroadCastUtoE($\mathbf{U}, i$)
    $\mathbf{e}'_i = \mathrm{MLP}_{\mathrm{edge}}([\mathbf{e}_i, \mathbf{v}_s, \mathbf{v}_r, \mathbf{u}])$
  end for
  for $i \leftarrow 1$ to $|\mathbf{V}|$ **do**
    $\mathbf{e}_s$ = AggregateEtoV($\mathbf{E}, \mathbf{sd}, i$)
    $\mathbf{e}_r$ = AggregateEtoV($\mathbf{E}, \mathbf{rv}, i$)
    $\mathbf{u}$ = BroadCastUtoV($\mathbf{U}, i$)
    $\mathbf{v}'_i = \mathrm{MLP}_{\mathrm{node}}([\mathbf{v}_i, \mathbf{e}_s, \mathbf{e}_r, \mathbf{u}])$
  end for
  for $i \leftarrow 1$ to $|\mathbf{U}|$ **do**
    $\mathbf{e}_u$ = AggregateEtoU($\mathbf{E}, i$)
    $\mathbf{v}_u$ = AggregateVtoU($\mathbf{V}, i$)
    $\mathbf{u}'_i = \mathrm{MLP}_{\mathrm{global}}([\mathbf{u}_i, \mathbf{e}_u, \mathbf{v}_u])$
  end for
  $\mathbf{E} = (\mathbf{e}'_1, \mathbf{e}'_2, ..., \mathbf{e}'_{|\mathbf{E}|})$
  $\mathbf{V} = (\mathbf{v}'_1, \mathbf{v}'_2, ..., \mathbf{v}'_{|\mathbf{V}|})$
  $\mathbf{U} = (\mathbf{u}'_1, \mathbf{u}'_2, ..., \mathbf{u}'_{|\mathbf{U}|})$
  return $(\mathbf{V}', \mathbf{E}', \mathbf{U}')$
end function

Introduction & Related Work

- Position-based Dynamics and Elastic Rods



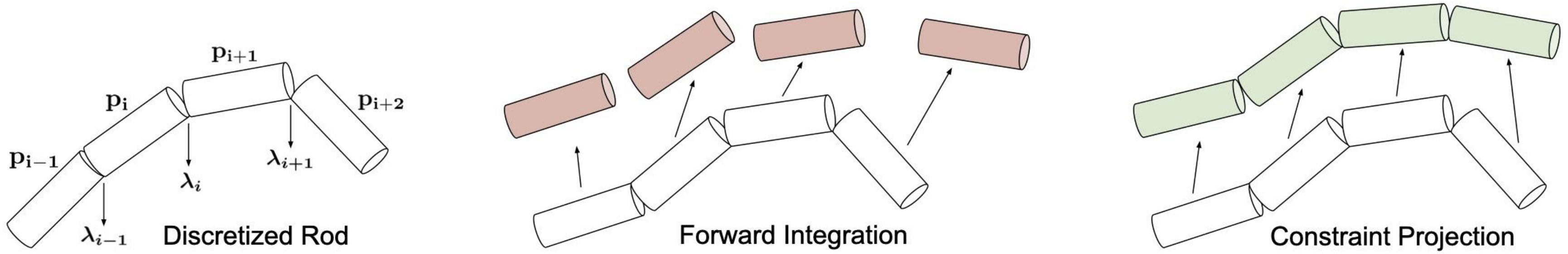
Position-based Dynamics(PBD): A fast and robust dynamic system simulation approach, Matthias Müller, 2006
Extended PBD: Improved version of PBD, Miles Macklin et al., 2016

Three steps: Forward Integration, Constraint Projection, Velocity Updating

1. Forward Integration: integration step to deal with external force
2. Constraint Projection: correction of the position based on the constraints, which deals with internal force or collisions
3. Updating: velocity update

Introduction & Related Work

- Position-based Dynamics and Elastic Rods



Position and Orientation Based Cosserat Rod, Tassilo Kugelstadt, 2016
Direct Position-Based Solver for Stiff Rods, Crispin Deul, 2018

A rod is discretized one dimensionally, each rod segment is described by position and orientation.
P in the figure represents the general coordinate.

Two types of constraint: stretch and shear constraint, bend and twist constraint.
Each constraint has a corresponding Lagrangian multiplier.

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Methodology

- Overview – one step in simulation

for all rod segments **do**
  $\boldsymbol{v}_i^* \leftarrow \boldsymbol{v}_{i,t} + \Delta t\, \mathbf{a}_{\text{ext}}$
  $\mathbf{p}_i^* \leftarrow \mathbf{p}_{i,t} + \Delta t\ \mathbf{H}(\mathbf{q}_{i,t})\, \boldsymbol{v}_i^*$ with $\mathbf{H}(\mathbf{q}_{i,t}) := [\mathbf{1}_{3\times3}, \mathbf{0}_{3\times3}; \mathbf{0}_{4\times3}, \mathbf{G}(\mathbf{q}_{i,t})]$
end for

Forward Integration

$\boldsymbol{\lambda}^0 \leftarrow \mathbf{0}, \mathbf{p}^0 \leftarrow \mathbf{p}^*$
$(\mathbf{Coll}_{\text{r-r}}, \mathbf{Coll}_{\text{r-p}}) \leftarrow \text{generateCollisionConstraints}(\mathbf{p}, \mathbf{p}^*)$
for $j \leftarrow 0$ to number of required solver iterations **do**
  for all rods **do**
    $(\Delta\mathbf{p}, \Delta\boldsymbol{\lambda}) \leftarrow \text{rodConstraintProjection}(\mathbf{p}^j, \boldsymbol{\lambda}^j)$
    $\boldsymbol{\lambda}^{j+1} \leftarrow \boldsymbol{\lambda}^j + \Delta\boldsymbol{\lambda}$
    $\mathbf{p}^{j+1} \leftarrow \mathbf{p}^j + \Delta\mathbf{p}$
  end for
  $\mathbf{p}^{j+1} \leftarrow \text{updateCollisionConstraintProjection}(\mathbf{p}^{j+1}, \mathbf{Coll}_{\text{r-r}}, \mathbf{Coll}_{\text{r-p}})$
  $j \leftarrow j + 1$
end for

Constraint Projection

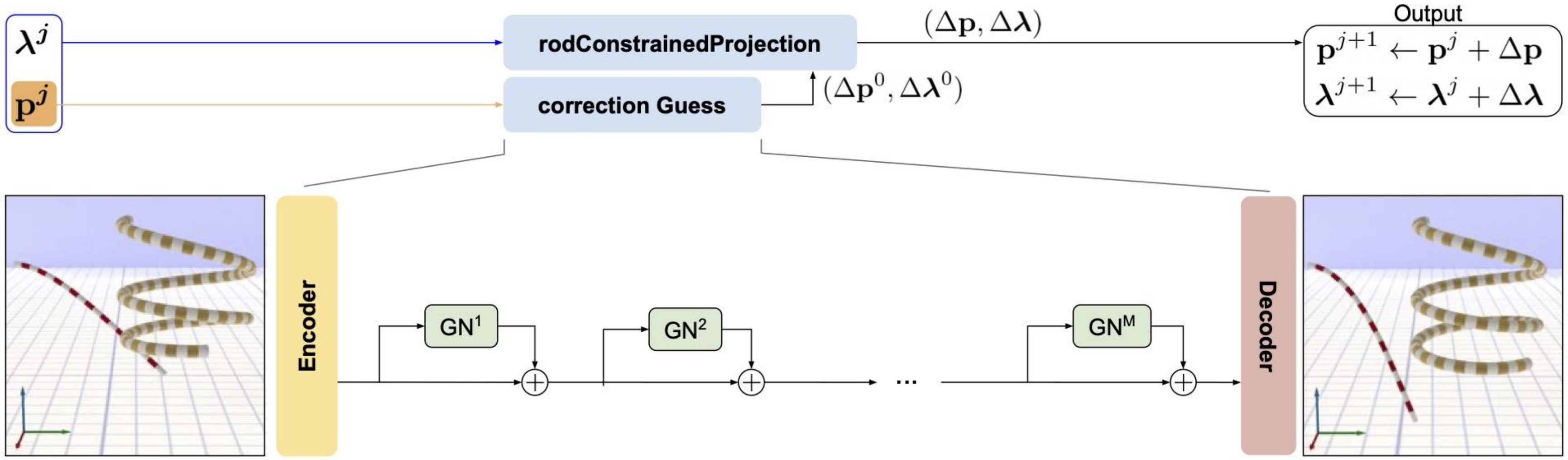
for all rod segments **do**
  $\mathbf{p}_{i,t+\Delta t} \leftarrow \mathbf{p}_i^j$
  $\boldsymbol{v}_{i,t+\Delta t} \leftarrow \mathbf{H}^{\mathsf{T}}(\mathbf{q}_{i,t})(\mathbf{p}_{i,t+\Delta t} - \mathbf{p}_{i,t})/\Delta t$
end for

Updating

Methodology

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

- Overview – rod constraint project



$$(\Delta\mathbf{p}, \Delta\boldsymbol{\lambda}) \leftarrow \text{rodConstraintProjection}(\mathbf{p}^j, \boldsymbol{\lambda}^j) \rightarrow \begin{cases} (\Delta\mathbf{p}^0, \Delta\boldsymbol{\lambda}^0) \leftarrow \text{correctionGuess}(\mathbf{p}^j) \\ (\Delta\mathbf{p}, \Delta\boldsymbol{\lambda}) \leftarrow \text{rodConstraintProjection}(\mathbf{p}^j, \boldsymbol{\lambda}^j, \Delta\mathbf{p}^0, \Delta\boldsymbol{\lambda}^0) \end{cases}$$

COPINGNet(COnstraint Projection INitial Guess Network)

CG ratio as performance indicator

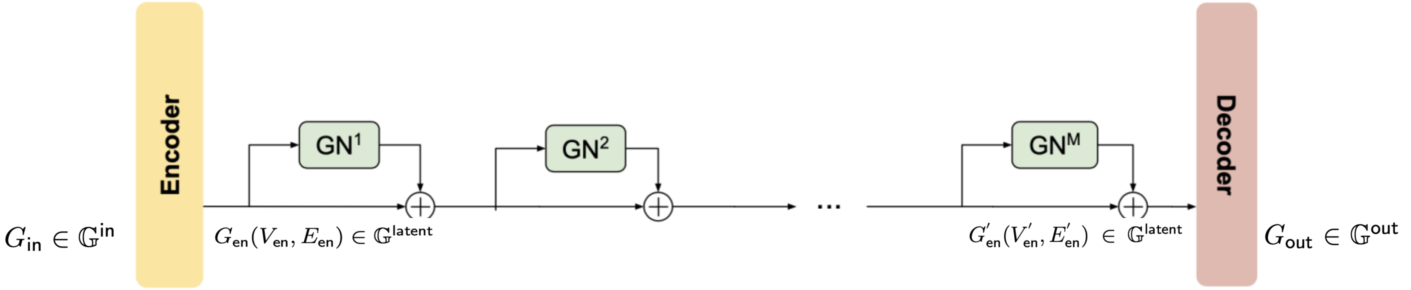
Methodology

- Graph Encoding

Final Goal, $\mathbb{G}^{\text{in}} \rightarrow \mathbb{G}^{\text{out}}$

$G_{\text{in}} \in \mathbb{G}^{\text{in}}$, and $G_{\text{in}} = G(V_{\text{in}}, E_{\text{in}})$

$G_{\text{out}} \in \mathbb{G}^{\text{out}}$, and $G_{\text{out}} = G(V_{\text{out}}, E_{\text{out}})$



Formulating the nodes features:

$$\mathbf{v}_{\text{in},i} = (\mathbf{x}_i, \mathbf{q}_i, r_i, \rho_i, \ell_i, \alpha_i, f_{0_i}, f_{1_i}, f_{2_i})^{\mathsf{T}} \in \mathbb{V}^{\text{in}} \subseteq \mathbb{R}^{14}$$, and $V_{\text{in}} = \cup_{i=1}^{n} \{\mathbf{v}_{\text{in},i}\}$

$$\mathbf{v}_{\text{out},i} = \Delta \mathbf{p}_i \in \mathbb{V}^{\text{out}} \subseteq \mathbb{R}^{6}$$, and $V_{\text{out}} = \cup_{i=1}^{n} \{\mathbf{v}_{\text{out},i}\}$

Formulating the edges features:

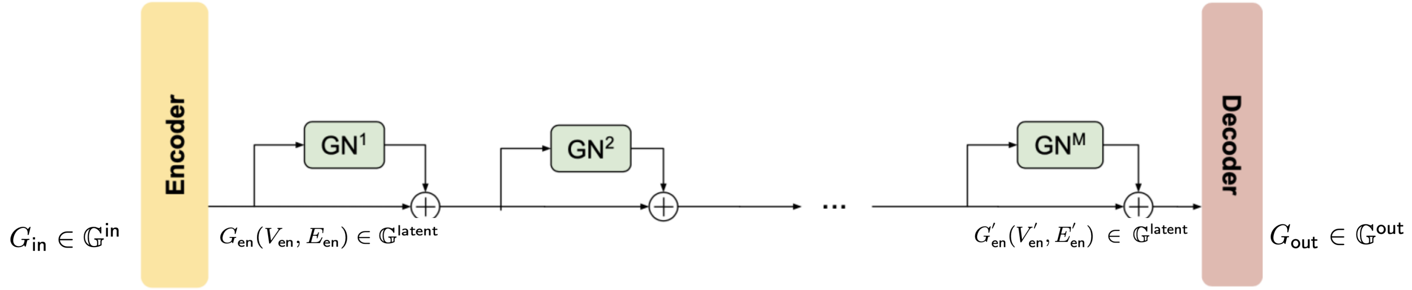
$$\mathbf{e}_{\text{in},i} = (\boldsymbol{\omega}_i, Y_i, T_i)^{\mathsf{T}} \in \mathbb{E}^{\text{in}} \subseteq \mathbb{R}^{5}$$, and $E_{\text{in}} = \cup_{i=1}^{m} \{\mathbf{e}_{\text{in},i}\}$

$$\mathbf{e}_{\text{out},i} = \Delta \boldsymbol{\lambda}_i \in \mathbb{E}^{\text{out}} \subseteq \mathbb{R}^{6}$$, and $E_{\text{out}} = \cup_{i=1}^{m} \{\mathbf{e}_{\text{out},i}\}$

Methodology

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Network Structure



Three parts of mapping

$$\mathbb{G}^{\text{in}} \to \mathbb{G}^{\text{out}} \quad \begin{cases} \mathbb{G}^{\text{in}} \to \mathbb{G}^{\text{latent}} \\ \mathbb{G}^{\text{latent}} \to \mathbb{G}^{\text{latent}} \\ \mathbb{G}^{\text{latent}} \to \mathbb{G}^{\text{out}} \end{cases}$$

$$G_{\text{en}}(V_{\text{en}}, E_{\text{en}}) \in \mathbb{G}^{\text{latent}}$$
$$G'_{\text{en}}(V'_{\text{en}}, E'_{\text{en}}) \in \mathbb{G}^{\text{latent}}$$

Encoding MLPs:

$$E_{\text{en}} = \text{MLP}_{\text{edge}}(E_{\text{in}})$$
$$V_{\text{en}} = \text{MLP}_{\text{node}}(V_{\text{in}})$$

Decoding MLPs:

$$E_{\text{out}} = \text{MLP}_{\text{edge}}(E'_{\text{en}})$$
$$V_{\text{out}} = \text{MLP}_{\text{node}}(V'_{\text{en}})$$

Final loss: $L := \text{MSE}(V_{\text{out}}, \tilde{V}_{\text{out}}) + \text{MSE}(E_{\text{out}}, \tilde{E}_{\text{out}})$

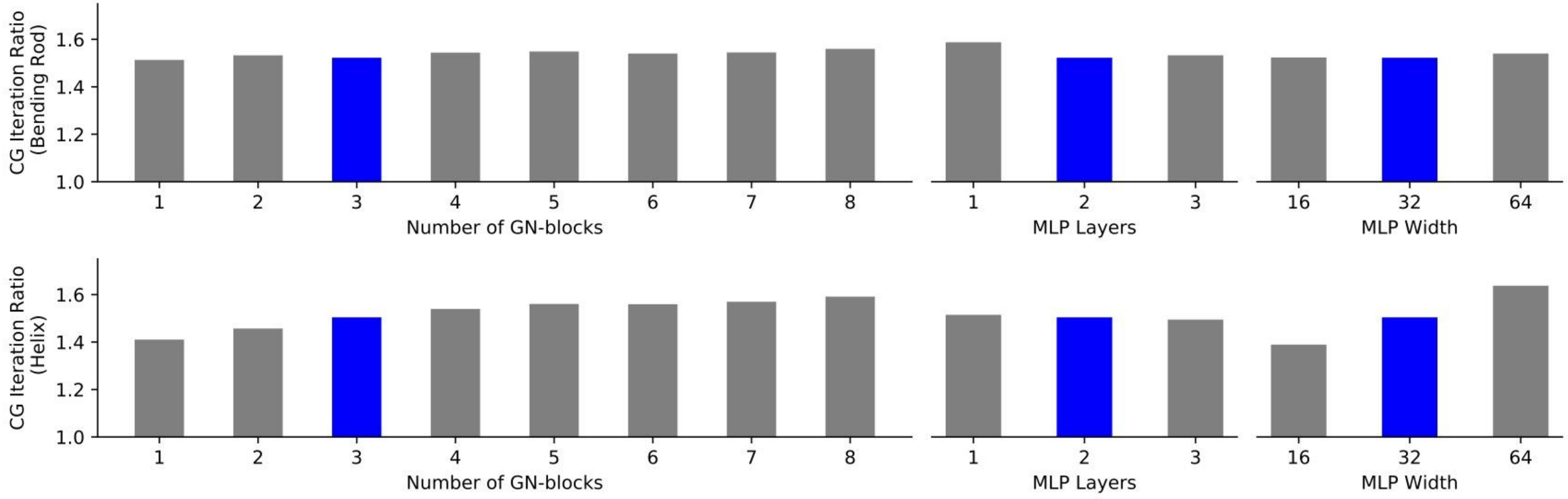
Evaluation

- Data Generation & Ablation study

Train/Val	#Steps	#Nodes N	Young's Modulus Y	Initial Angle ϕ_0	Rod Length ℓ
256/100	50	$\mathcal{U}_d(10, 55)$	10^a Pa, $a \sim \mathcal{U}(4.0, 6.0)$	$\mathcal{U}(0°, 45.0°)$	$\mathcal{U}(1.0\ \text{m}, 5.0\ \text{m})$
Train/Val	**#Steps**	**#Nodes N**	**Torsion Modulus T**	**Helix Radius / Height**	**Winding Number**
256/100	50	$\mathcal{U}_d(45, 105)$	10^a Pa, $a \sim \mathcal{U}(4.0, 6.0)$	$\mathcal{U}(0.4\ \text{m}, 0.6\ \text{m})$ / $\mathcal{U}(0.4\ \text{m}, 0.6\ \text{m})$	$\mathcal{U}(2.0, 3.0)$

Bending rod

Helix



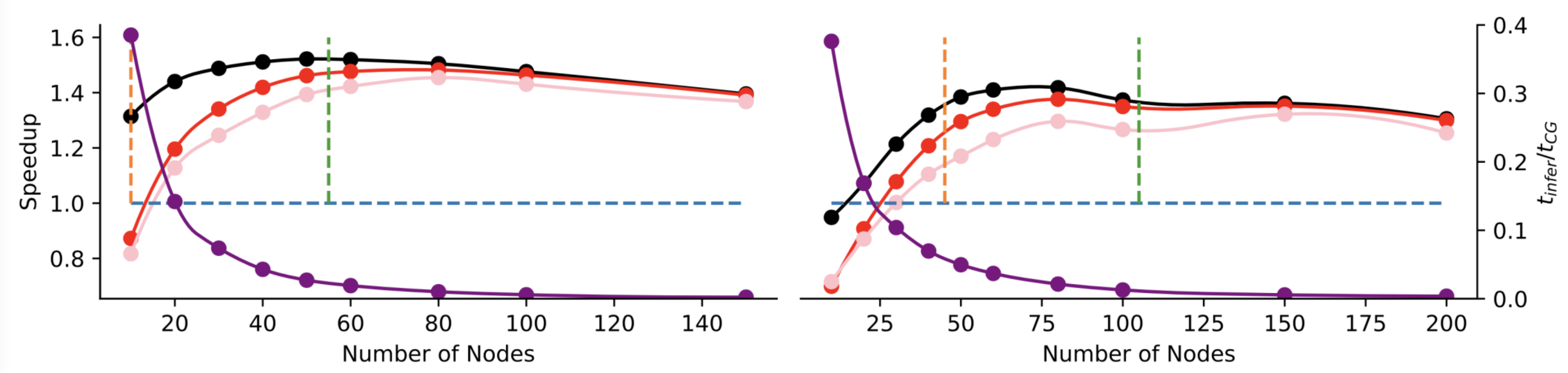
Bending rod

Helix

Averaged from 50 test simulations each running for 100 time steps

Evaluation

- Spatial Discretization



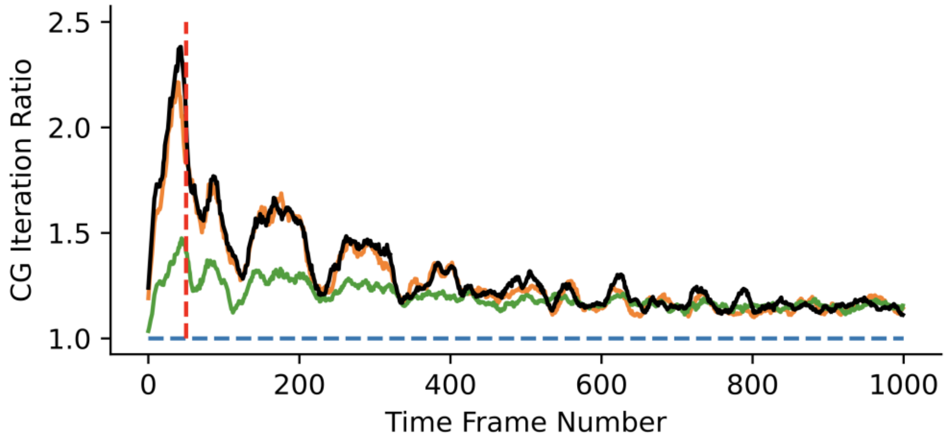
Black: CG iteration ratio
Purple: Inference time of neral network to CG time
Red: Speedup taking acount of inference time
Pink: Net speedup

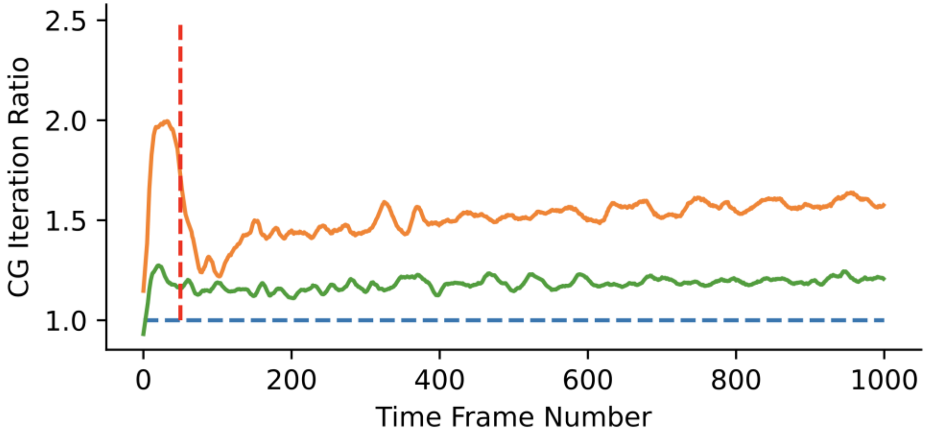
Left: bending rod
Right: helix

Averaged from 50 test simulations each running for 100 time steps

Evaluation

- Temporal Evolution





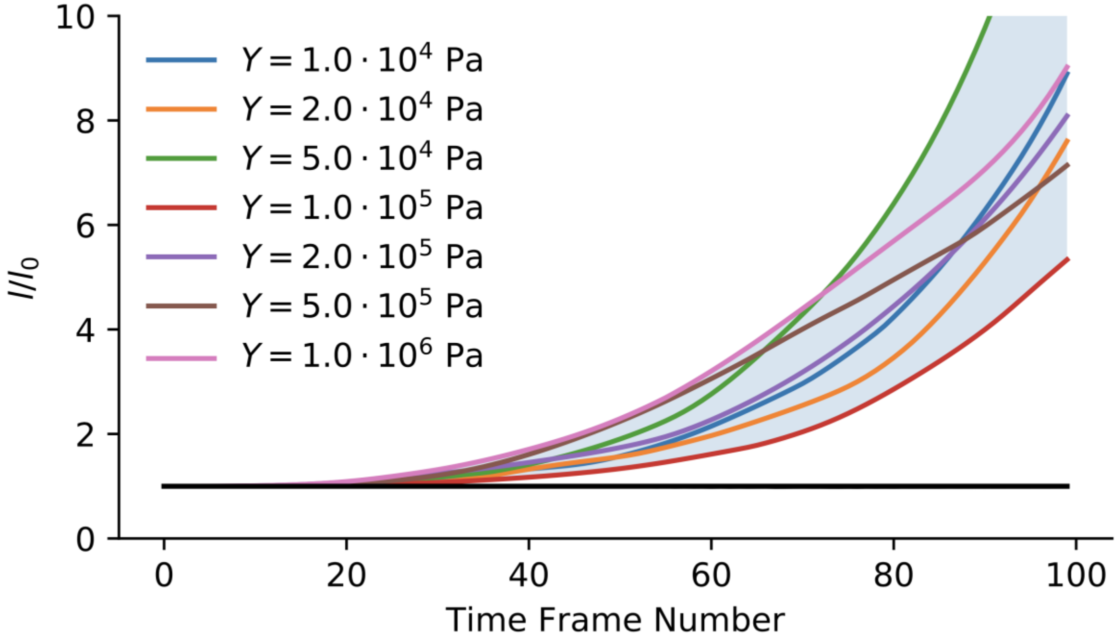
Orange: CG ratio using our framework
Green: CG ratio using k-nearest neighbours(k=3)
Black: CG ratio with our framework extra restriction

Left: bending rod
Right: helix

Select one set of simulation parameters

Evaluation

- Long-term Stability



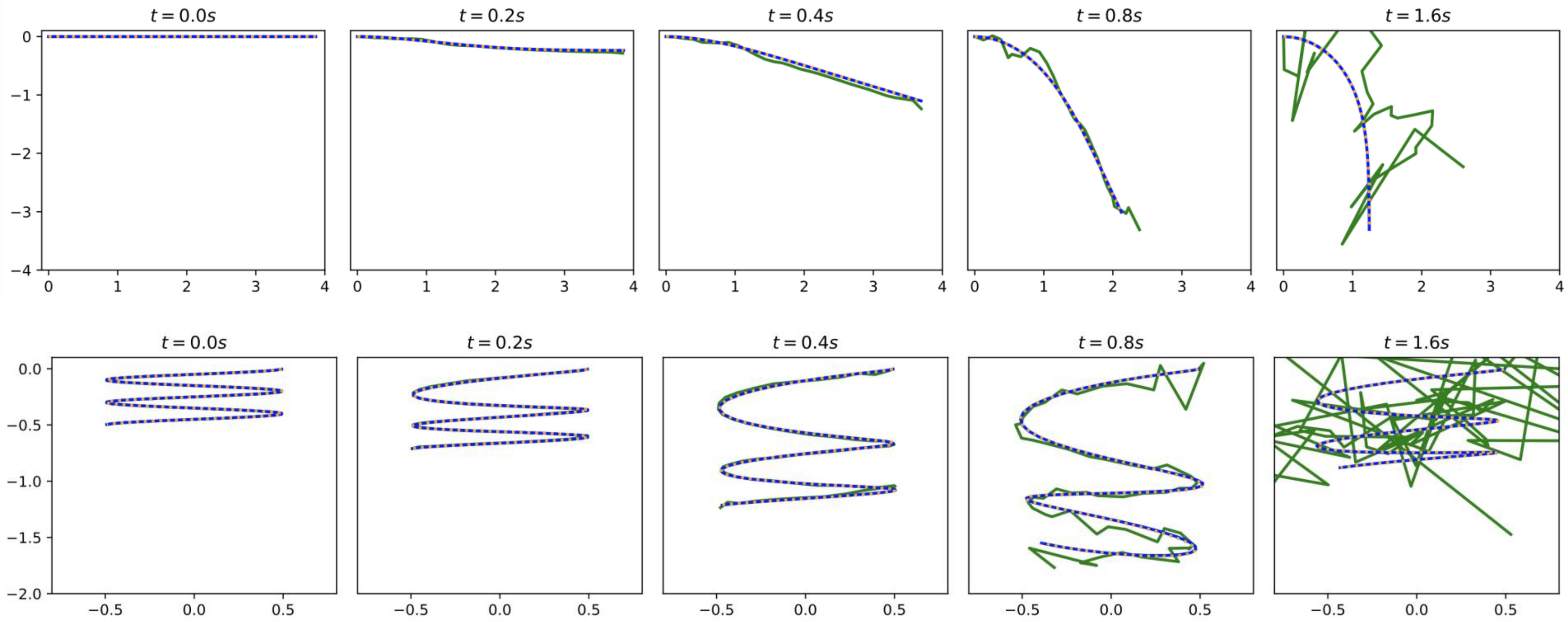
Relative change of the total rod length

Colored lines show different results for COPINGNet replacing the constraint projection, called GN-based end-to-end.

The thick black line represent the result for COPINGNet only offering the initial guess.

Evaluation

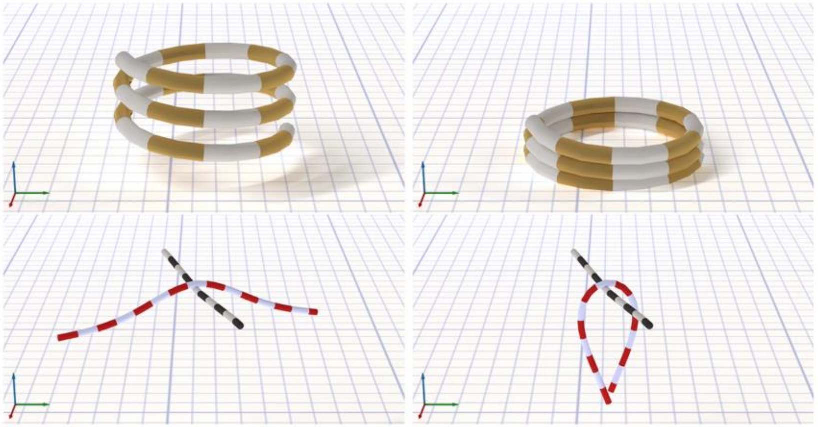
- Long-term Stability

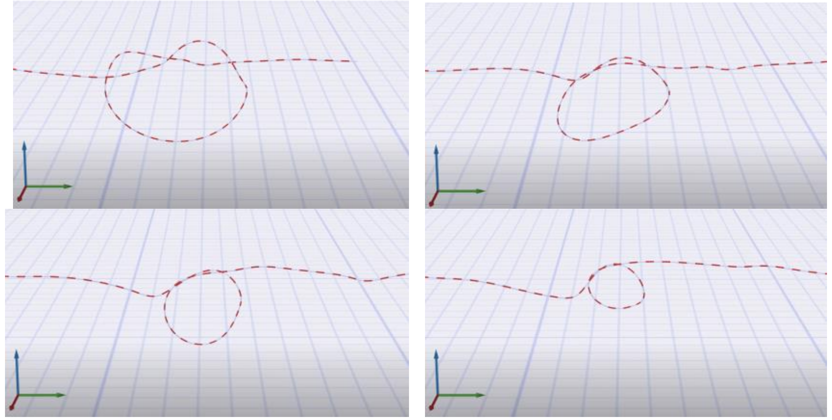


Orange: Traditional method
Blue: Our approach
Green: GN-based end-to-end

Evaluation

- Complex Scenario – Collisions, Knot, Trees





Conclusion

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

- We show how to accelerate iterative solvers with COPINGNet;
- We show that our network-assisted solver ensures long-term stability required for simulating physical systems;
- We demonstrate accuracy and generalizability of our approach by simulating different scenarios and various mechanical properties of rods including collisions and complex topologies;

- Future work for simulating other deformable objects or accelerate other iterative solvers