CS570 Team 19

KAIST

Point cloud learning for accelerating large deformation modeling

Changjun Lee Computer Science
Yidong Zhao Civil and Environmental Engineering
Seongbin An Mechanical Engineering
Sihyun Kim Computer Science

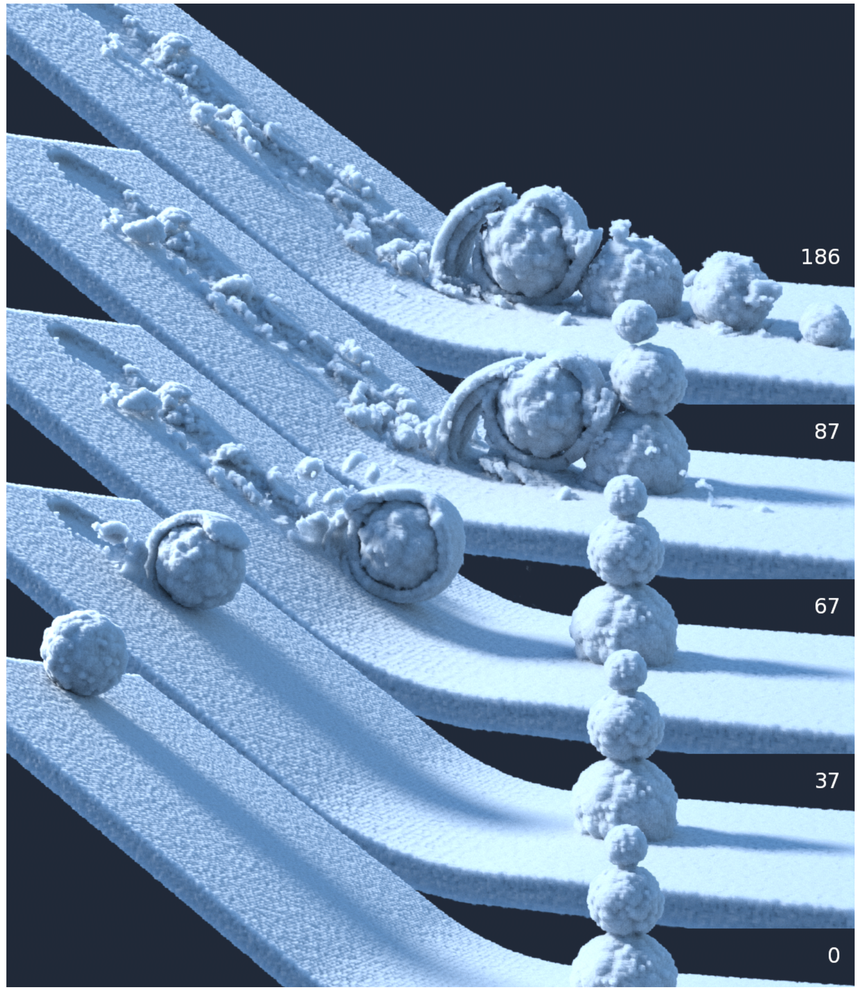
Apr. 28, 2022

Abstract:

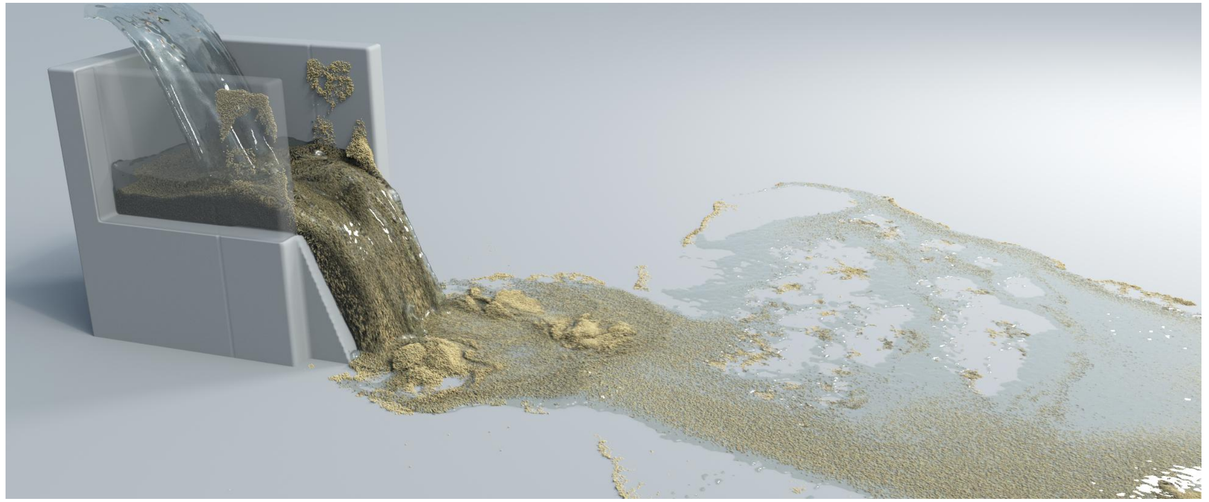
- Large deformation modeling is important but challenging.
- Reducing computational cost is necessary.
- Previous works have limitations.
- Our idea is based on learning from points cloud directly.

Background knowledge on large deformation modeling

Large deformation modeling is exciting!



Snow ([Stomakhin, et al., SIGGRAPH, 2013])



Multi-species ([Tampubolon, et al., SIGGRAPH, 2017])

Dongpo Pork ([Wolper, et al., SIGGRAPH, 2020])

Large deformation modeling is also important!

But modeling is also challenging...

1. Which method shall we use?

2. Simulation is time consuming...

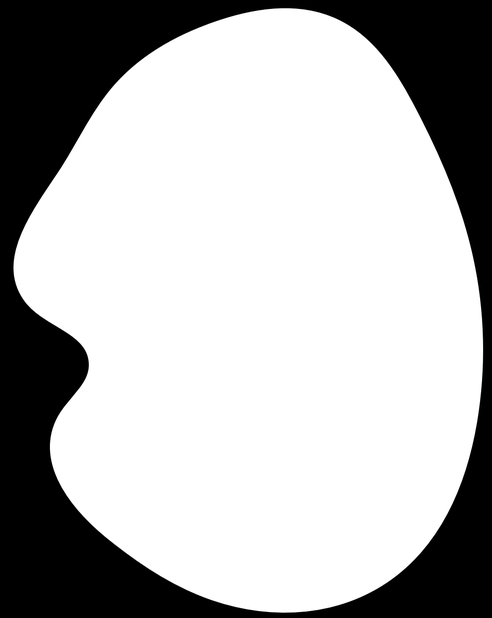
Let's look at them one-by-one:

1. Which method shall we use?

2. Simulation is time consuming...

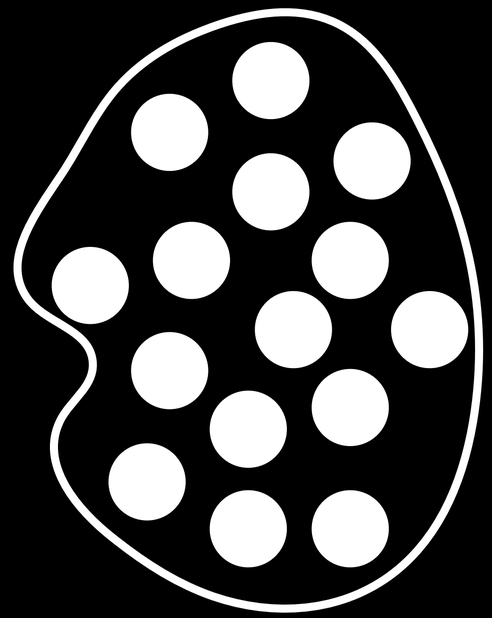
Material point method (MPM):

Material point method (MPM):



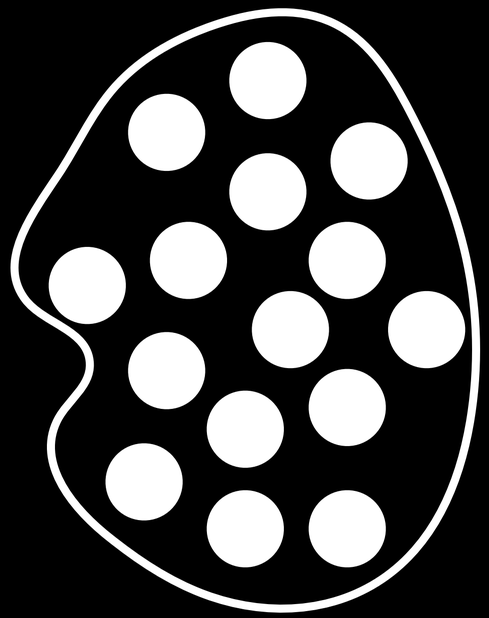
A continuum body

Material point method (MPM):

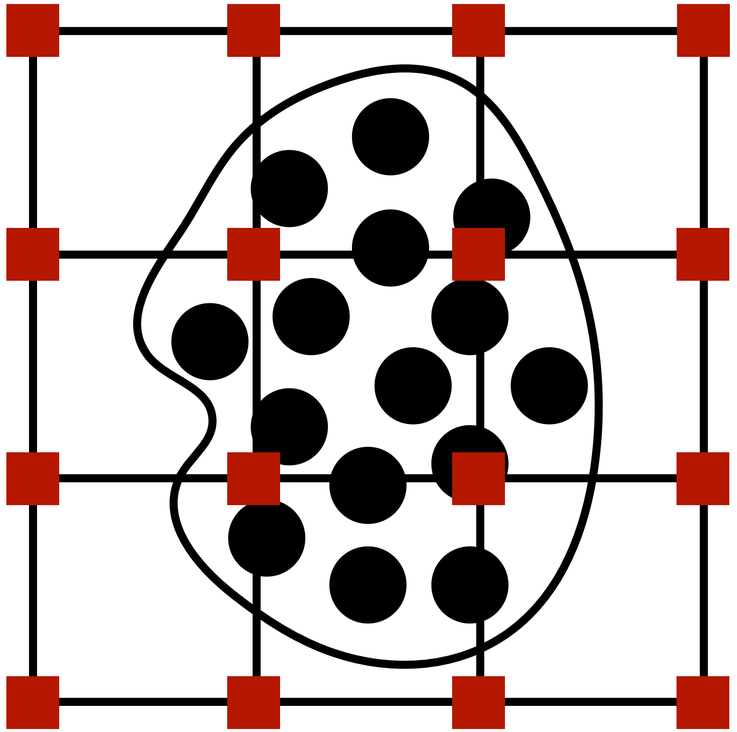


A continuum body is discretized with Lagrangian particles

Material point method (MPM):



A continuum body is discretized with Lagrangian particles



An Eulerian grid is used for interpolation.

We use MPM for modeling

1. Which method shall we use?

MPM

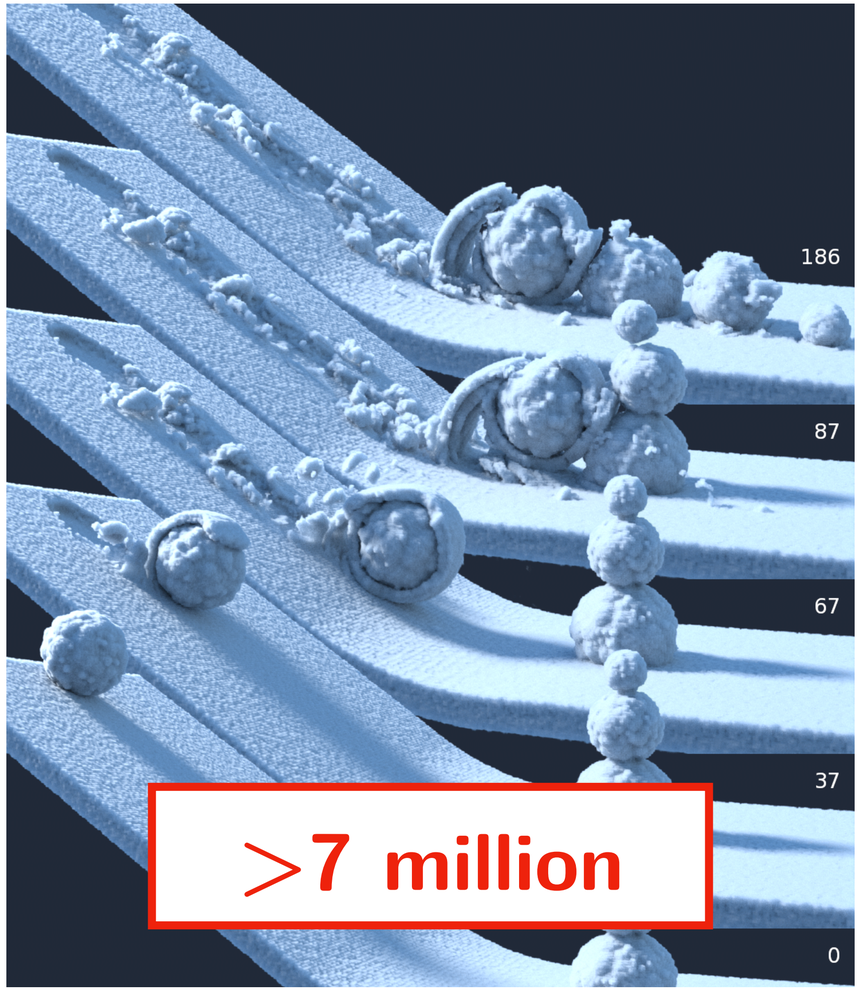
2. Simulation is time consuming...

Computational cost is very high:

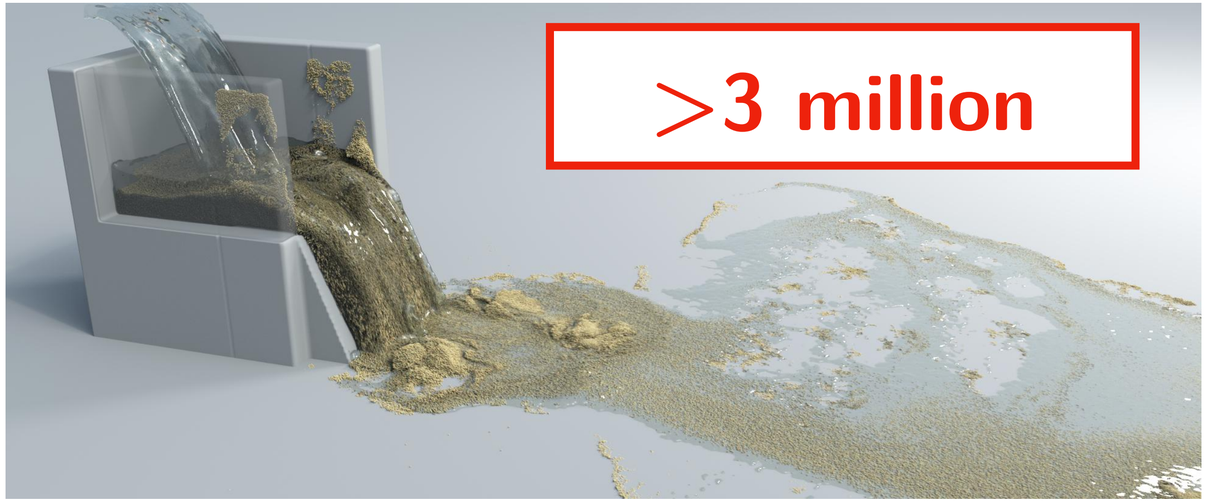
1. Which method shall we use?

2. Simulation is time consuming...

A large number of particles are used...



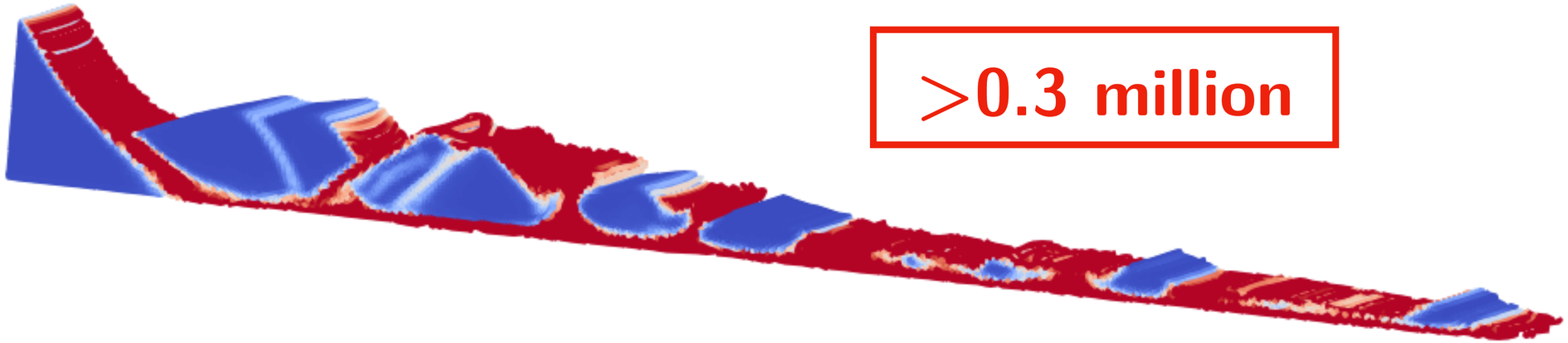
Snow ([Stomakhin, et al., SIGGRAPH, 2013])



Multi-species ([Tampubolon, et al., SIGGRAPH, 2017])

Dongpo Pork ([Wolper, et al., SIGGRAPH, 2020])

A large number of particles are used…

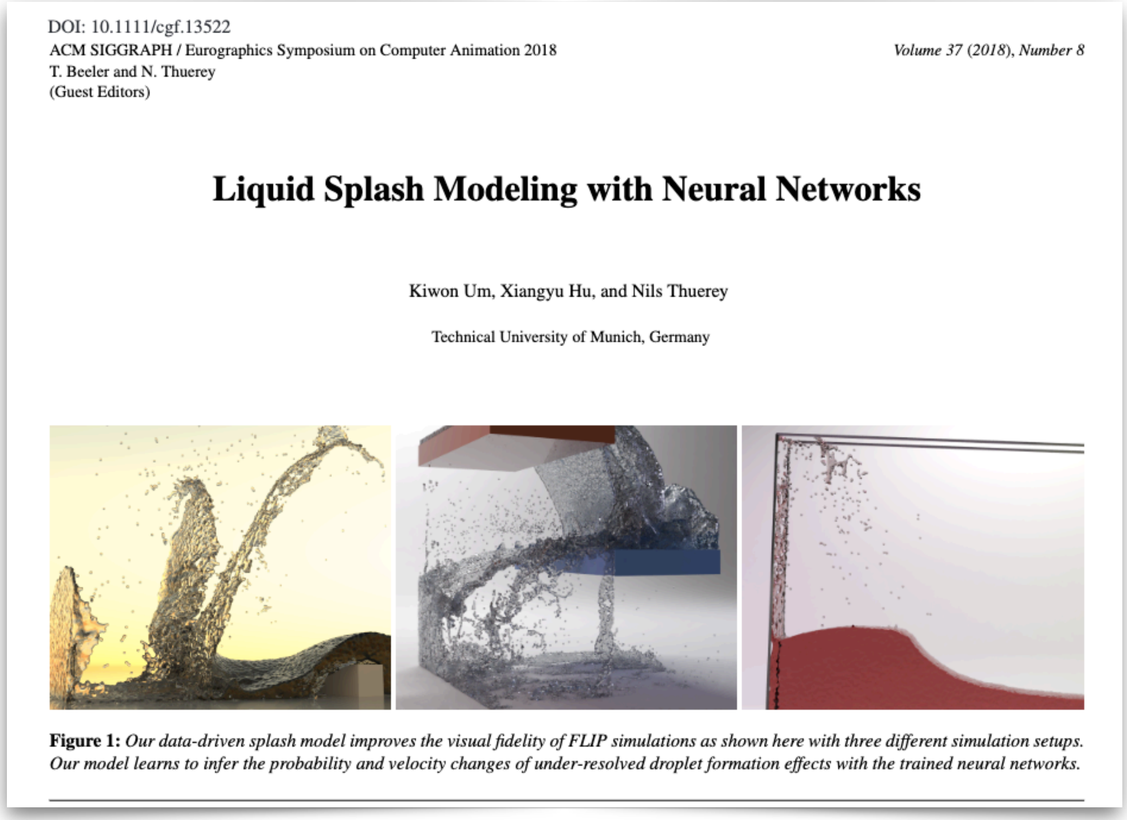


>0.3 million

Can we use machine learning techniques to reduce computational cost?

Review on existing methods:

MLFLIP: learning fluid splash



DOI: 10.1111/cgf.13522
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2018
T. Beeler and N. Thuerey
(Guest Editors)
Volume 37 (2018), Number 8

Liquid Splash Modeling with Neural Networks

Kiwon Um, Xiangyu Hu, and Nils Thuerey

Technical University of Munich, Germany

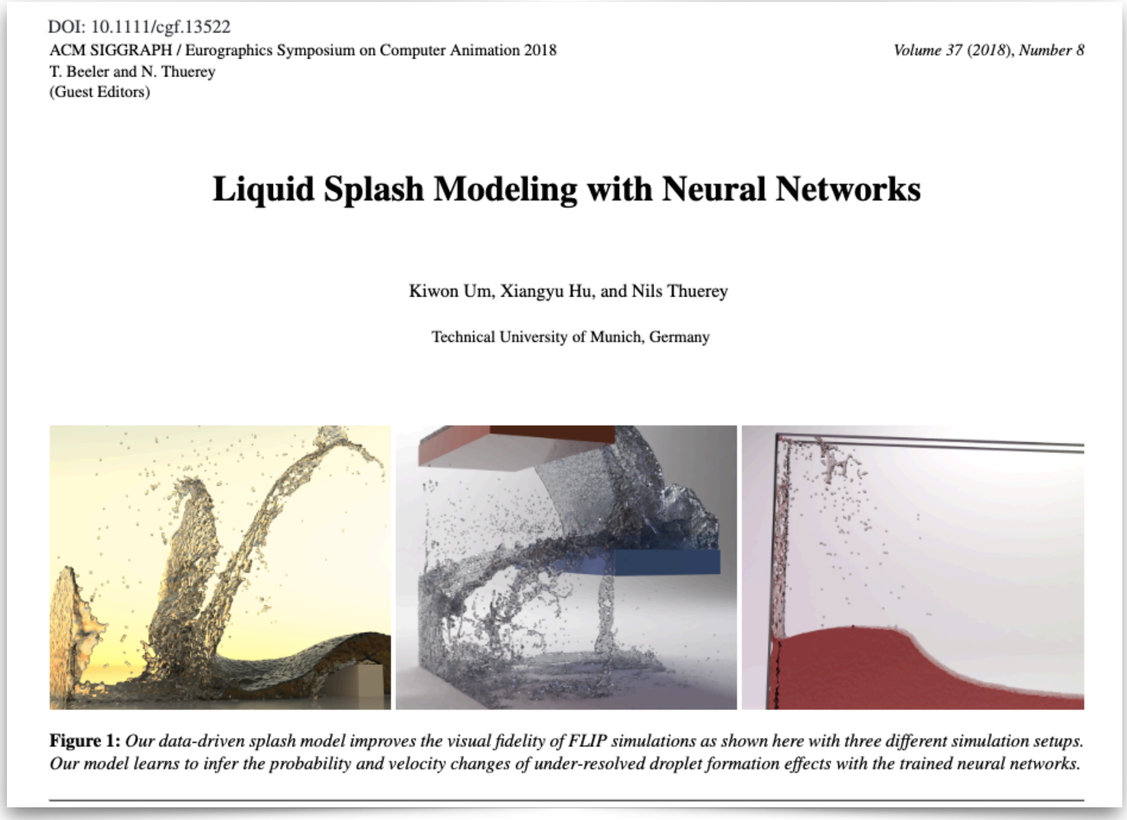
Figure 1: *Our data-driven splash model improves the visual fidelity of FLIP simulations as shown here with three different simulation setups. Our model learns to infer the probability and velocity changes of under-resolved droplet formation effects with the trained neural networks.*

[Um, et al, 2018]

- A detachment classification + velocity modification model is used for splash modeling

Review on existing methods:

MLFLIP: learning fluid splash



DOI: 10.1111/cgf.13522
ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2018
T. Beeler and N. Thuerey
(Guest Editors)
Volume 37 (2018), Number 8

Liquid Splash Modeling with Neural Networks

Kiwon Um, Xiangyu Hu, and Nils Thuerey

Technical University of Munich, Germany

Figure 1: *Our data-driven splash model improves the visual fidelity of FLIP simulations as shown here with three different simulation setups. Our model learns to infer the probability and velocity changes of under-resolved droplet formation effects with the trained neural networks.*

[Um, et al, 2018]

- A detachment classification + velocity modification model is used for splash modeling

- Limitation: the model can be only used for fluid splash modeling.

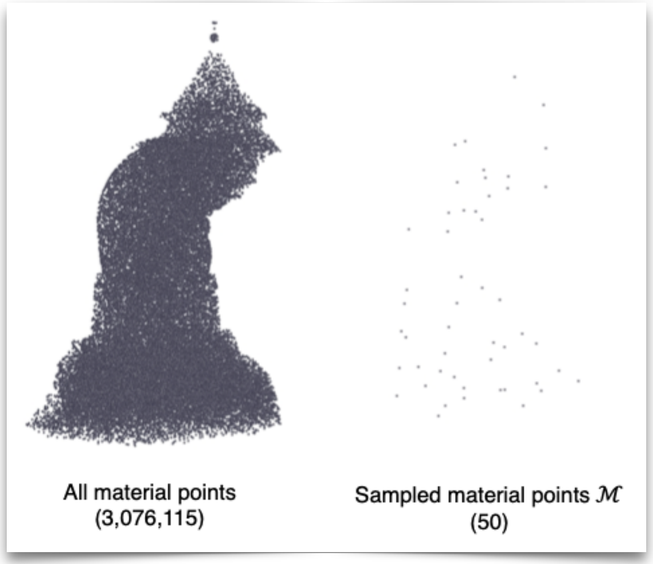
Review on existing methods:

Learning deformation map

Model reduction for the material point method via learning the deformation map and its spatial-temporal gradients

Peter Yichen Chen[a,*], Maurizio Chiaramonte[b], Eitan Grinspun[c,a], Kevin Carlberg[b]

[a] *Columbia University, 116th St & Broadway, New York, NY 10027, USA*
[b] *Facebook Reality Labs Research, 9845 Willows Road, Redmond, WA 98052, USA*
[c] *University of Toronto, 40 St. George Street, Room 4283, Toronto, ON M5S 2E4, Canada*



[Chen, et al, 2021]

- Achieving model reduction by learning the deformation map.

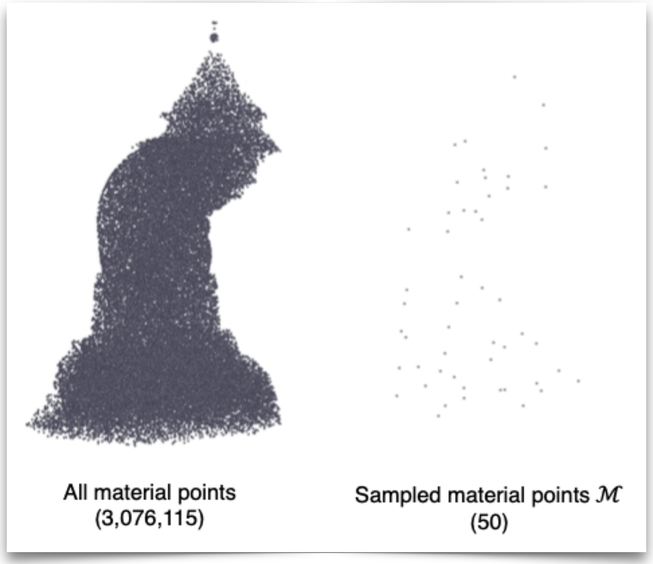
Review on existing methods:

Learning deformation map

Model reduction for the material point method via learning the deformation map and its spatial-temporal gradients

Peter Yichen Chen[a,*], Maurizio Chiaramonte[b], Eitan Grinspun[c,a], Kevin Carlberg[b]

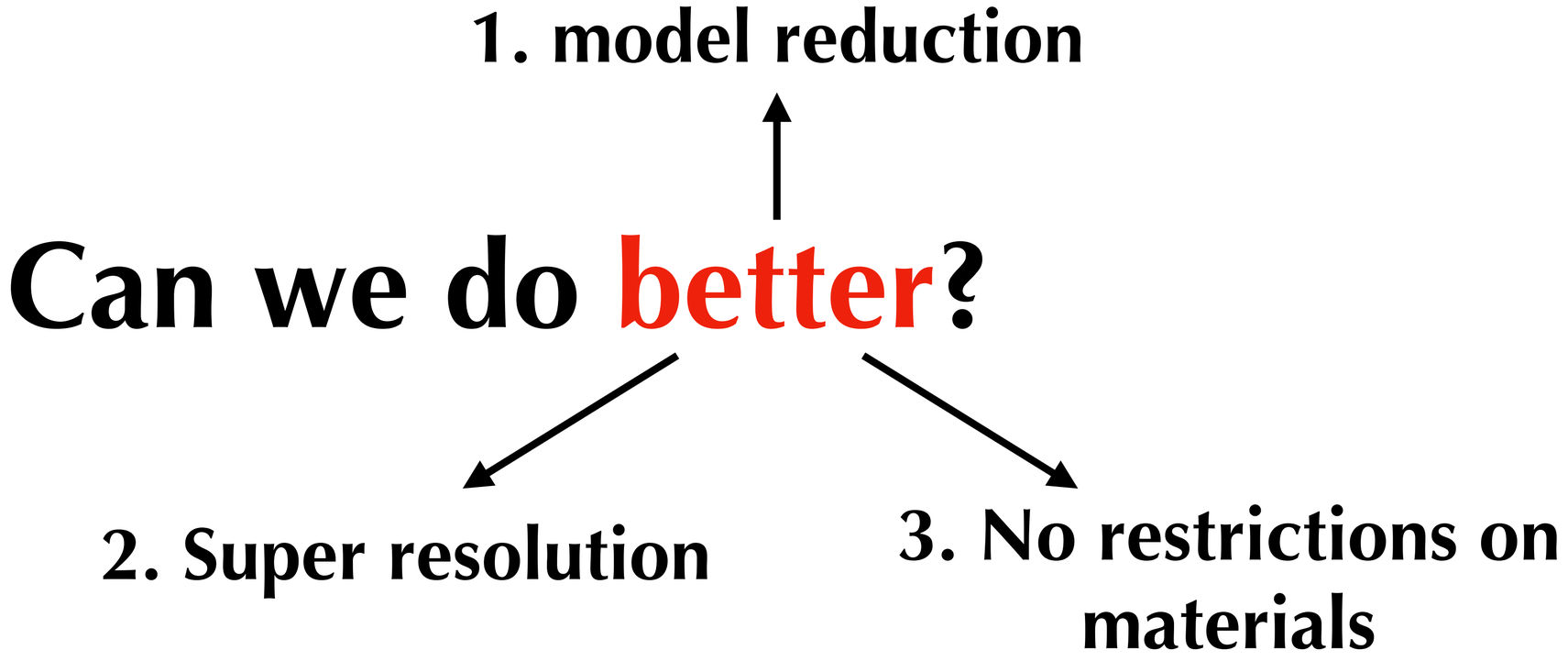
[a] *Columbia University, 116th St & Broadway, New York, NY 10027, USA*
[b] *Facebook Reality Labs Research, 9845 Willows Road, Redmond, WA 98052, USA*
[c] *University of Toronto, 40 St. George Street, Room 4283, Toronto, ON M5S 2E4, Canada*



[Chen, et al, 2021]

- Achieving model reduction by learning the deformation map.
- Limitation: the work only shows performances on elasticity, not sure its performance on other materials (e.g., plasticity)

Can we do better?


1. model reduction
Can we do better?
2. Super resolution
3. No restrictions on materials

Motivation:

learning directly from points cloud

- Numerical method (MPM) uses particles to discretize a continuum body.

- It is naturally to borrow ideas from points cloud learning.

Motivation:

learning directly from points cloud

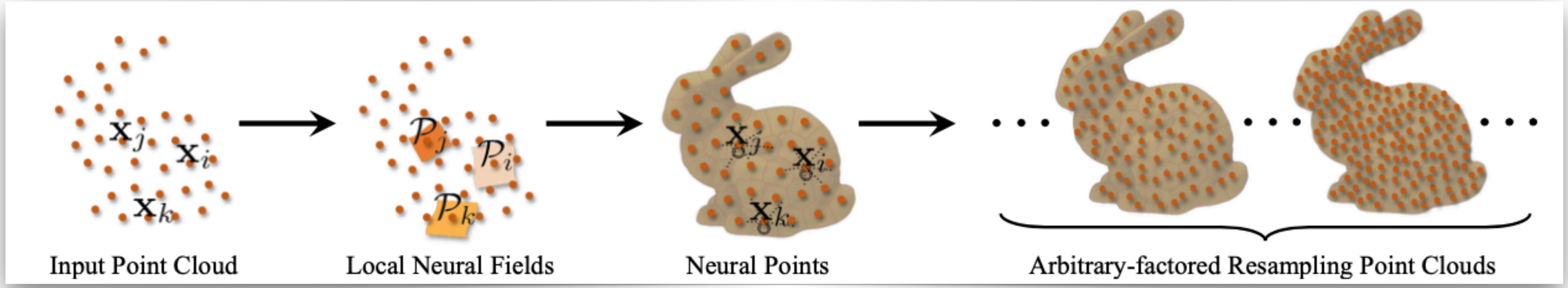
- Numerical method (MPM) uses particles to discretize a continuum body.
- It is naturally to borrow ideas from points cloud learning.
- There are many advanced points cloud learning techniques supporting model reduction with super-resolution at the same time.

Motivation:

Neural Points: using Neural Fields to represent point cloud

Neural Points: Point Cloud Representation with Neural Fields for Arbitrary Upsampling

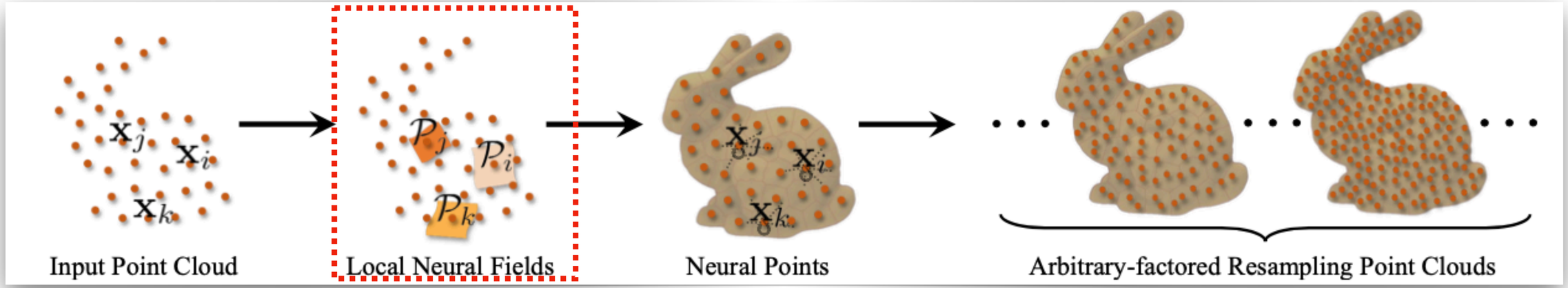
Wanquan Feng[1] Jin Li[2] Hongrui Cai[1] Xiaonan Luo[2] Juyong Zhang[1*]
[1]University of Science and Technology of China [2]Guilin University Of Electronic Technology
lcfwq@mail.ustc.edu.cn, 20032201014@mails.guet.edu.cn, hrcai@mail.ustc.edu.cn,
luoxn@guet.edu.cn, juyong@ustc.edu.cn



[Feng, et al, CVPR 2022]

A brief summary of the paper:

Step 1: extracting local feature using dynamic graph CNN

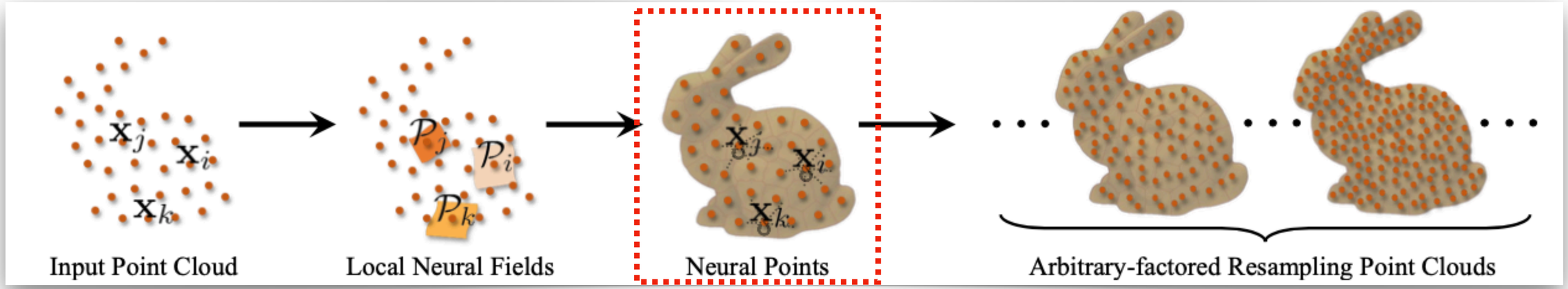


$$\boldsymbol{f}_i = \boldsymbol{f}_i^* \oplus MaxPool\{\boldsymbol{f}_k^*\}_{k \in \mathcal{N}(\boldsymbol{x}_i)}$$

[Feng, et al, CVPR 2022]

A brief summary of the paper:

Step 2: integrating each Neural field

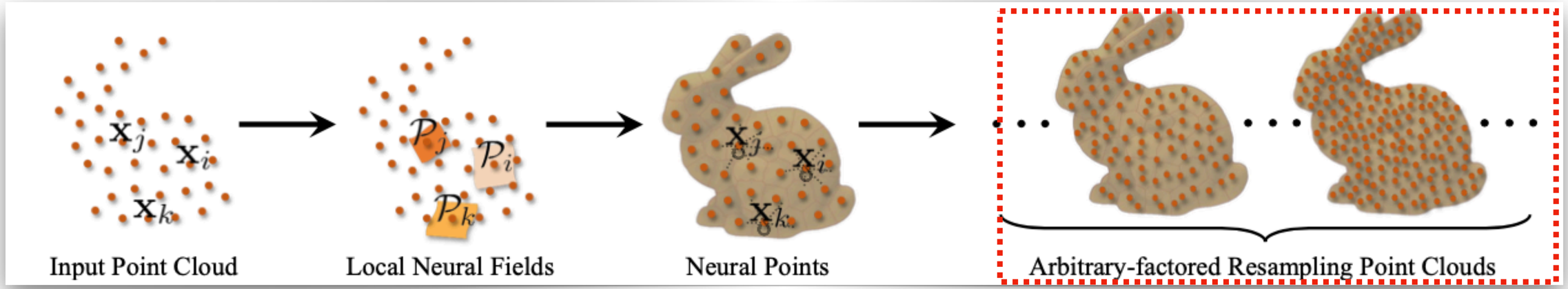


Integrating local neural fields to form the global shape

[Feng, et al, CVPR 2022]

A brief summary of the paper:

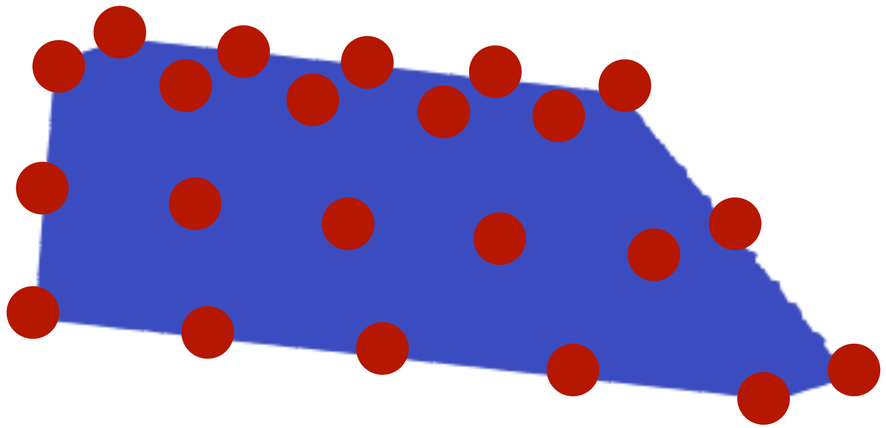
Step 3: resampling arbitrary point clouds



[Feng, et al, CVPR 2022]

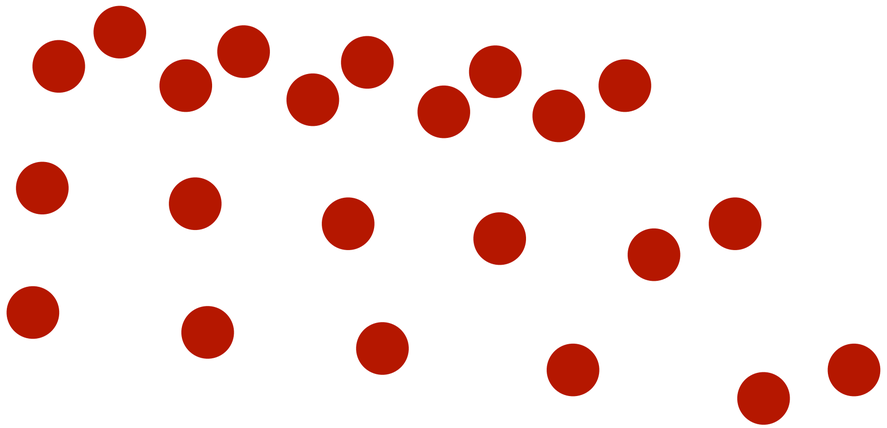
Applying neural points for MPM:

Getting very sparse sampling points



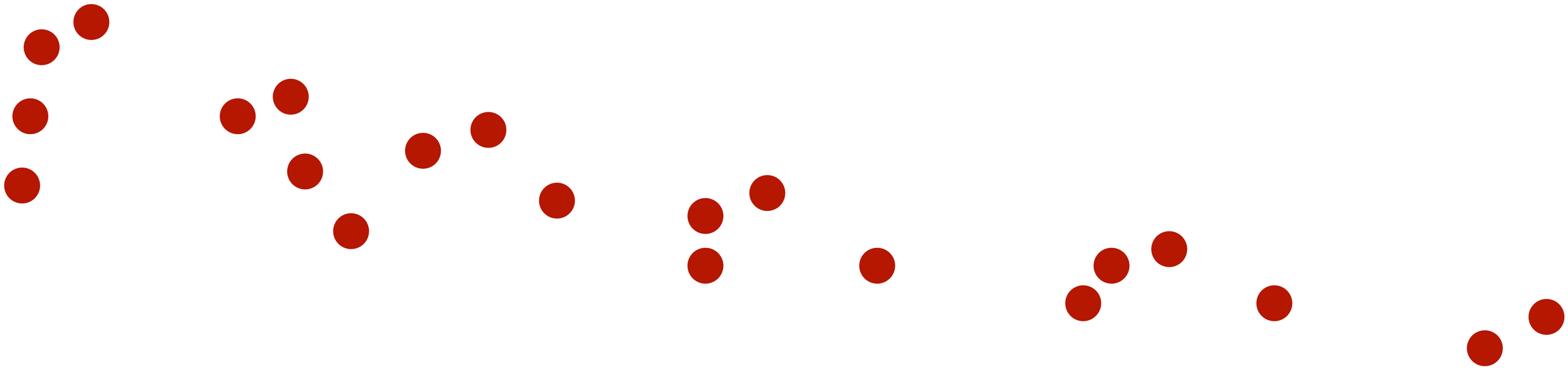
Applying neural points for MPM:

Getting very sparse sampling points



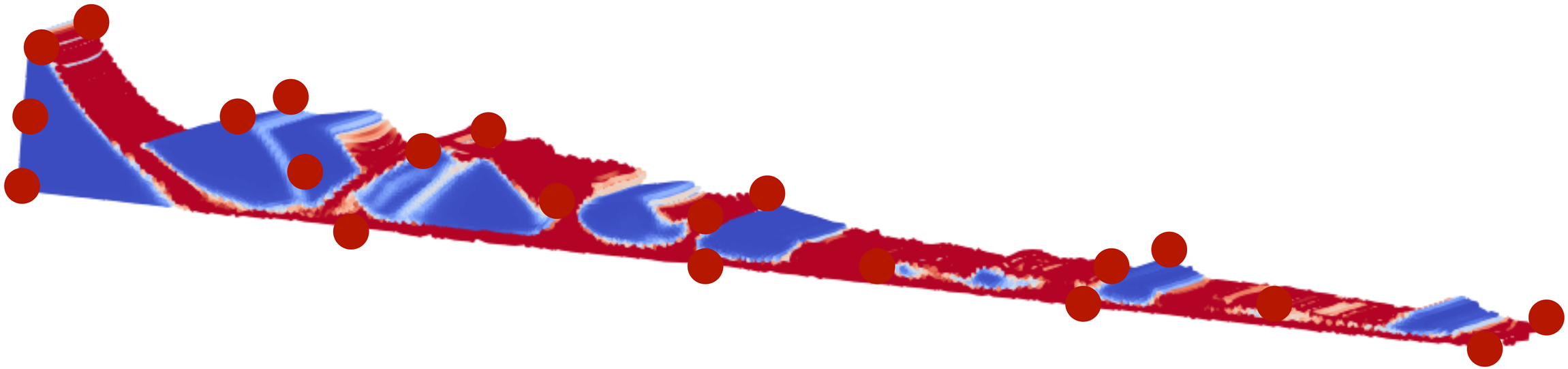
Applying neural points for MPM:

Only simulate these very sparse points



Applying neural points for MPM:

Upsampling using neural points



Plan:

- Try examples from the paper.
- Generate our own data set using MPM.
- Apply Neural points for our own data set.

Thank you very much!