Simulation & Animation

Niloy Mitra — UCL/ Adobe

Iasonas Kokkinos — UCL/ Ariel AI

Paul Guerrero — UCL/ Adobe

Vladimir Kim — Adobe

Nils Thuerey — TUM

Leonidas Guibas — Stanford/ FAIR

Timetable

		Niloy	Iasonas	Paul	Nils	Leonidas
Introduction	9:00	X				
Neural Network Basics	~9:15		X			
Supervised Learning in CG	~9:50	X				
Unsupervised Learning in CG	~10:20			X		
Learning on Unstructured Data	~10:55					X
Learning for Animation	~11:35				X	
Discussion	12:05	X	X	X	X	X

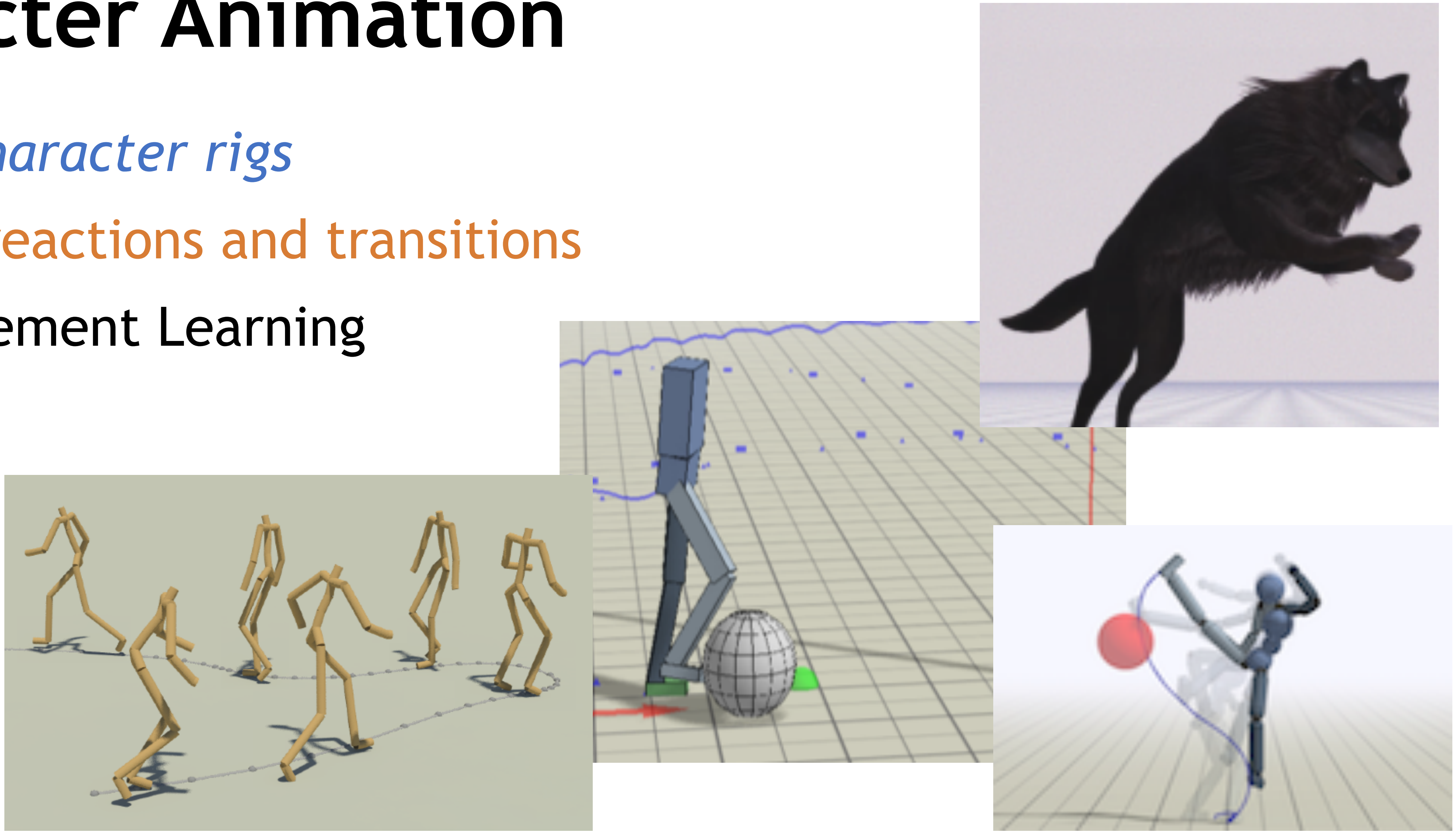
Computer Animation

- Feature detection (image features, point features) $\mathbb{R}^{m\times m} \to \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$
- Embedding, Distance computation $\mathbb{R}^{m\times m, m\times m} \to \mathbb{R}^{d}$
- Rendering $\mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$
- Animation $\mathbb{R}^{3m\times t} \to \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m\times t} \to \mathbb{R}^{3m}$

- Motion over time
- Lots of data - expensive…
- Relationships between spatial and temporal changes

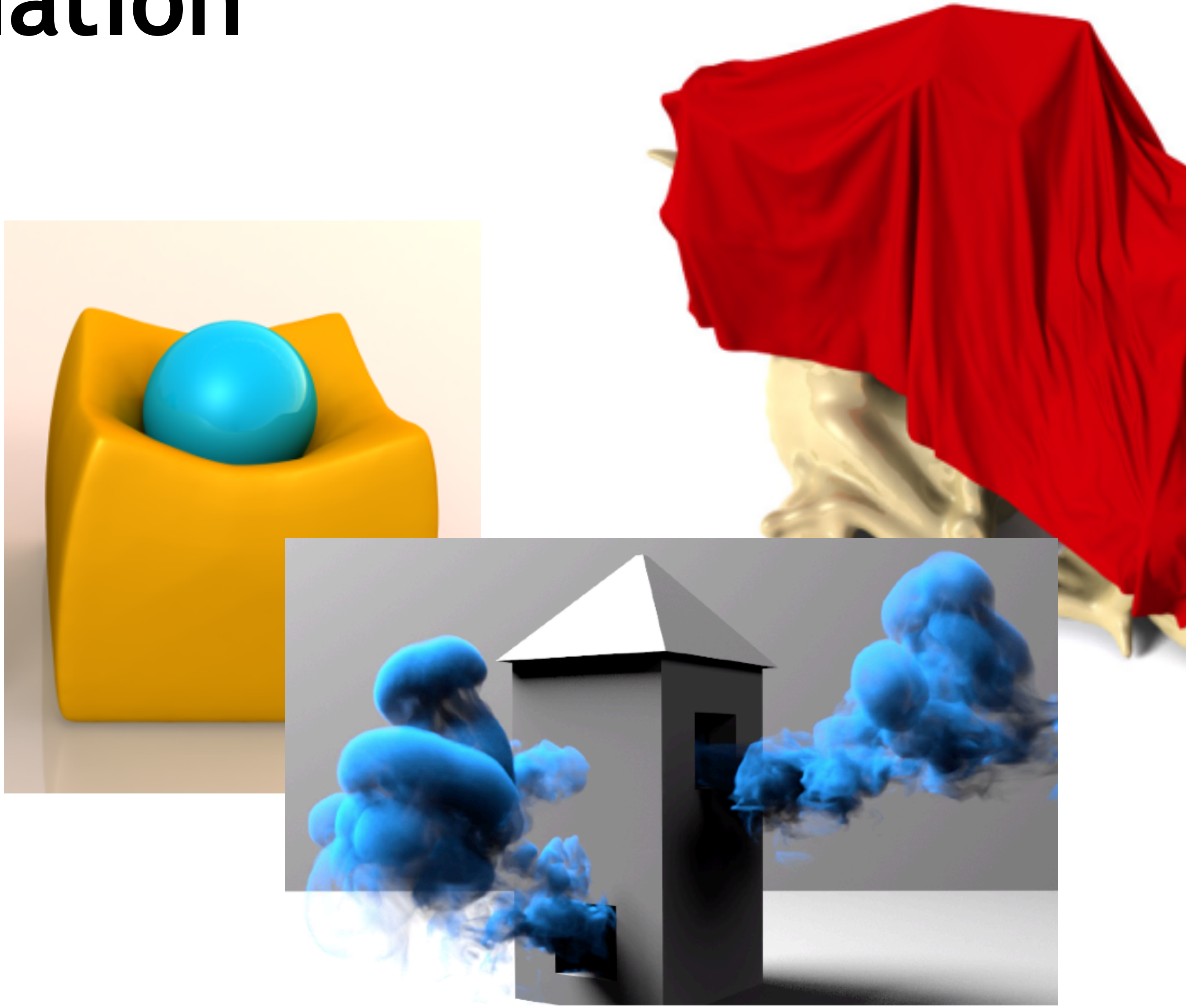
Character Animation

- Target *character rigs*
- Natural reactions and transitions
- Reinforcement Learning

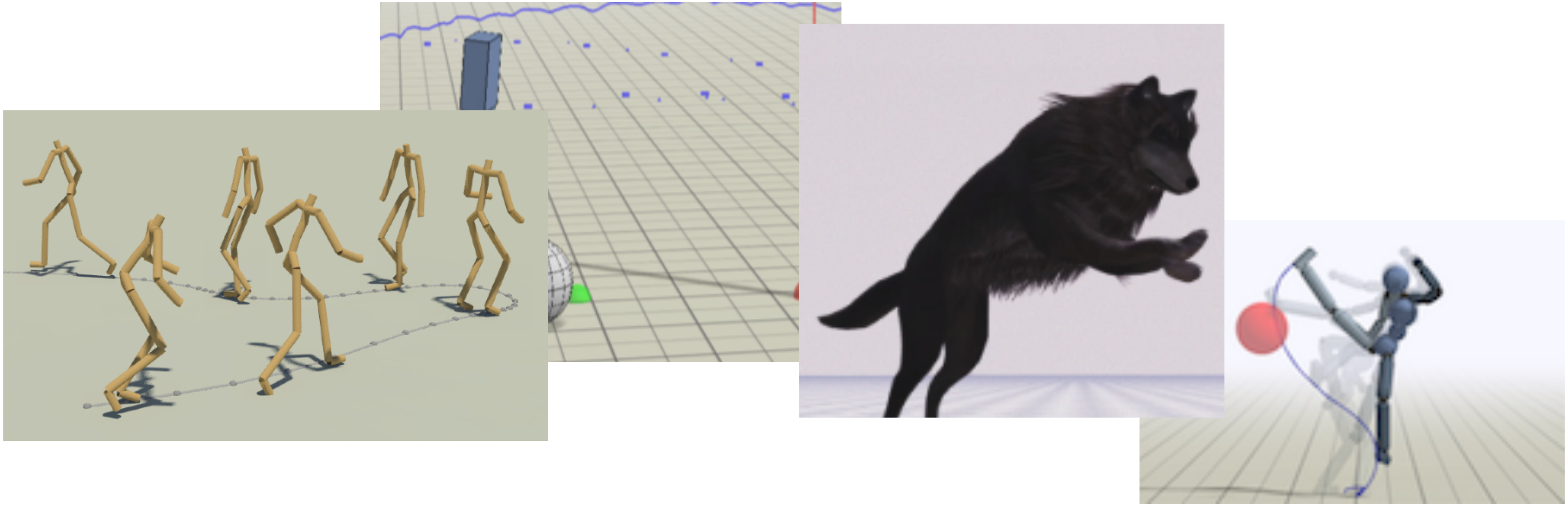


Physics-Based Animation

- Leverage *physical models*
- Examples:
  - Rigid bodies
  - Cloth
  - Deformable objects
  - Fluids



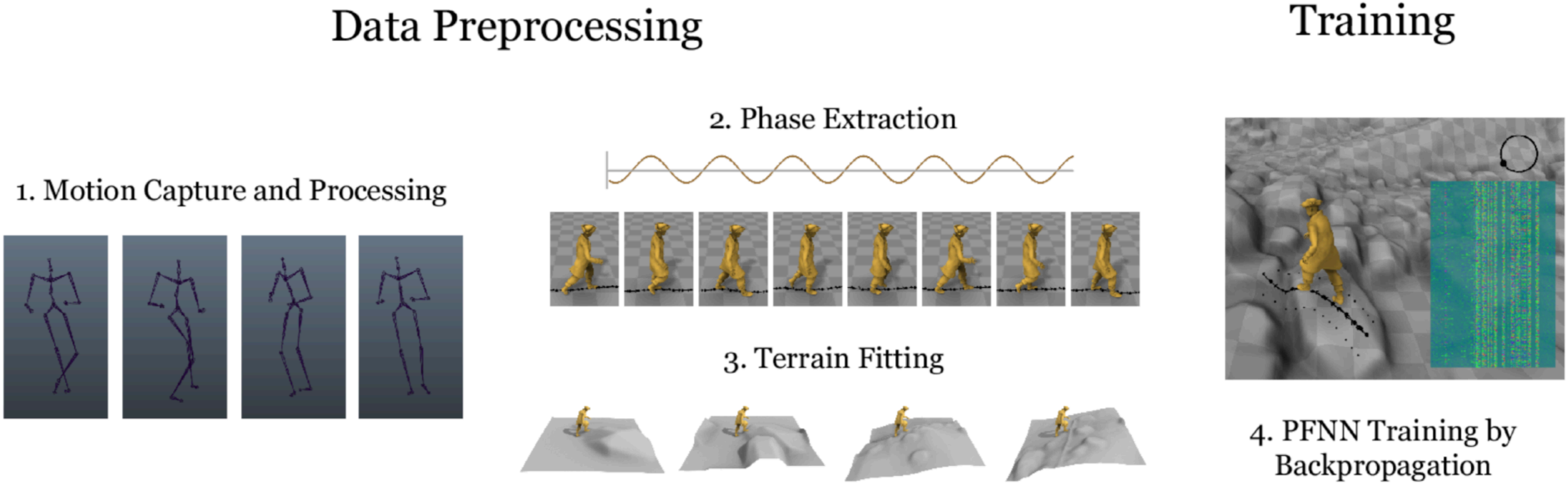
Character Animation



Existing Approaches

- Motion Representations
- Controllers

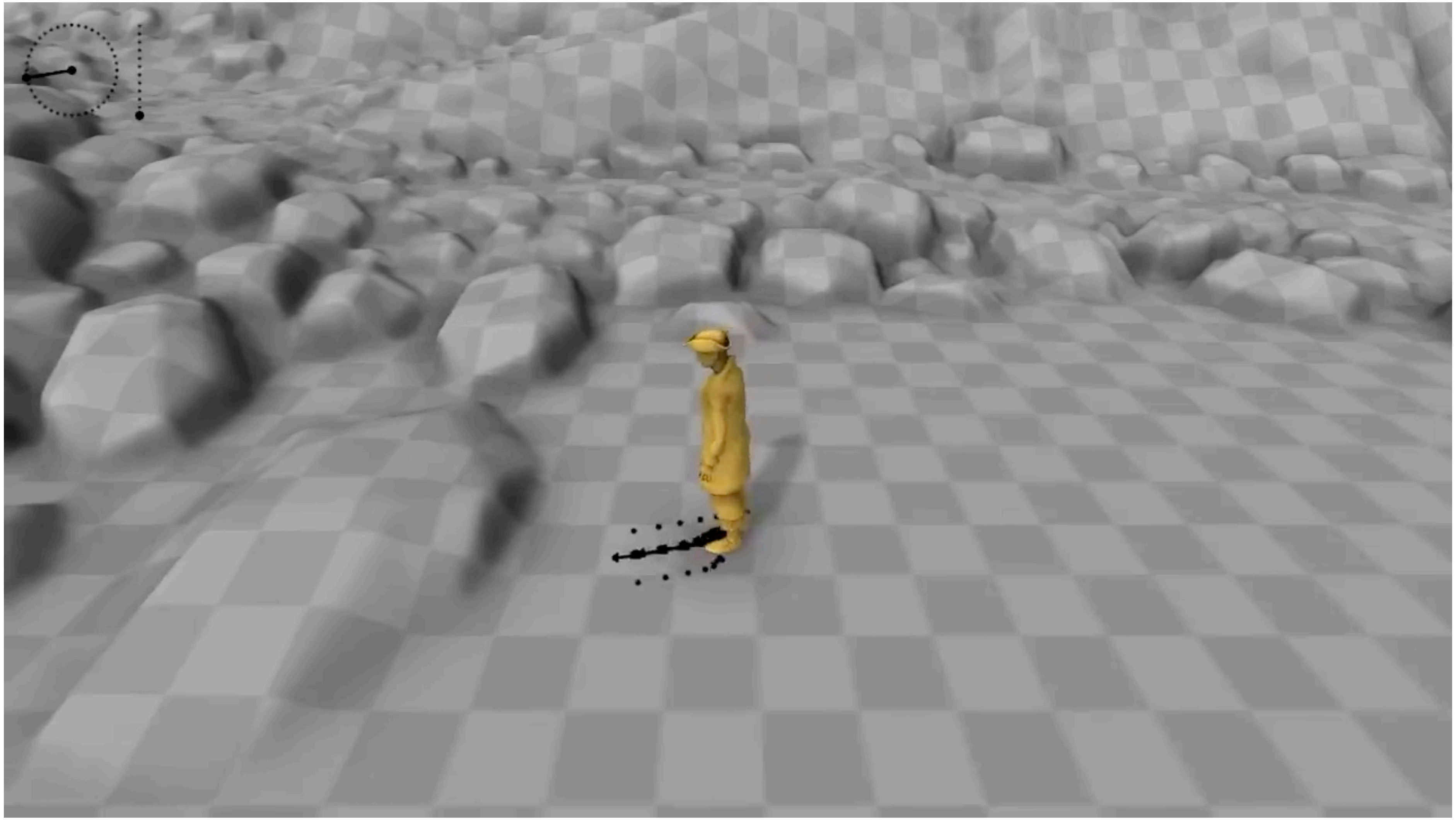
Learned Motion Manifolds



[Learning Motion Manifolds with Convolutional Autoencoders, SGA 2015 Tech. Briefs]

[Phase-functioned neural networks for character control, SIGGRAPH 2017]

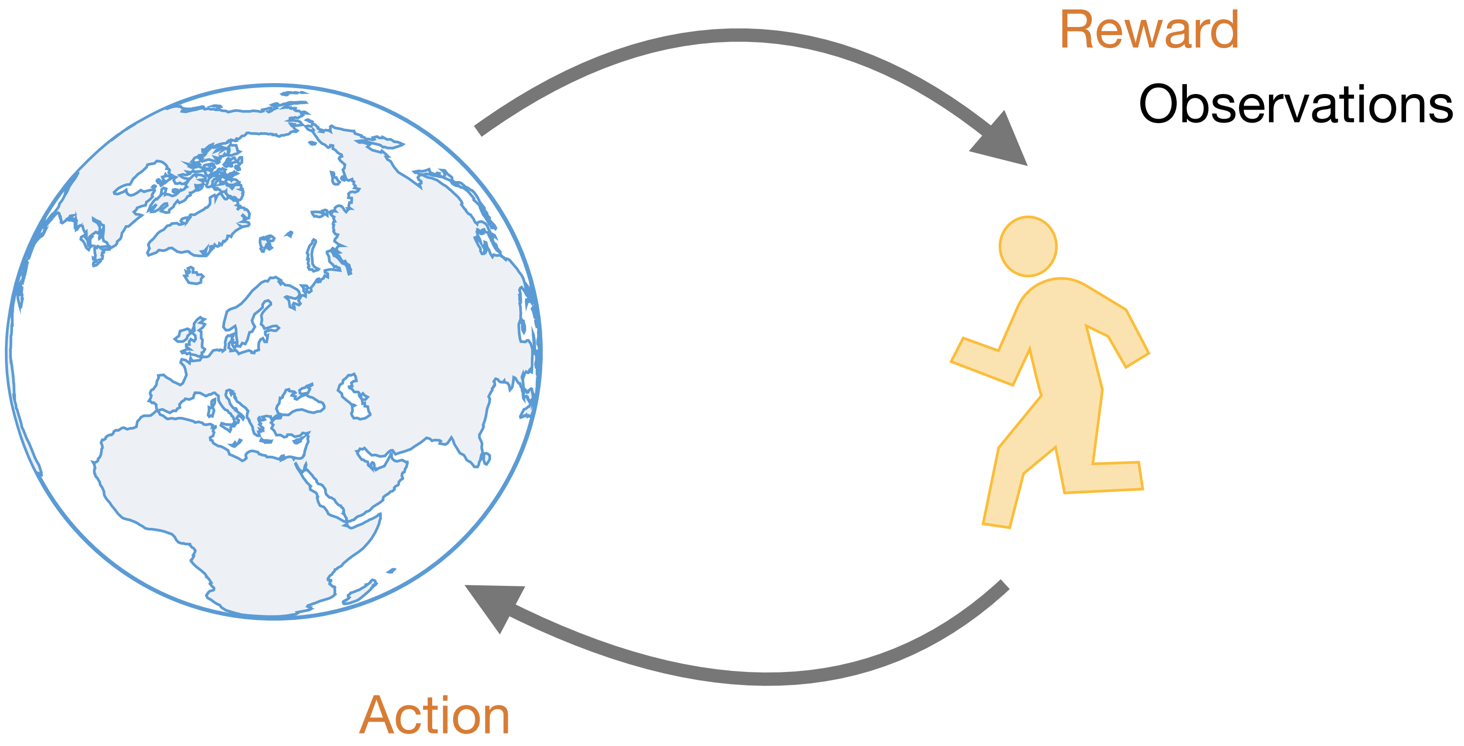
Learned Motion Manifolds



[Phase-functioned neural networks for character control, SIGGRAPH 2017]

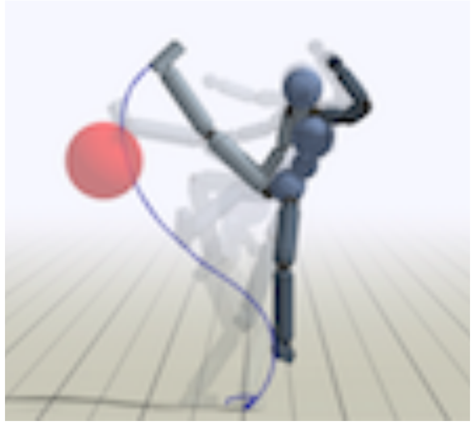
Reinforcement Learning

- Goal: maximize *reward* by performing *actions* in an *environment*



RL for Animation

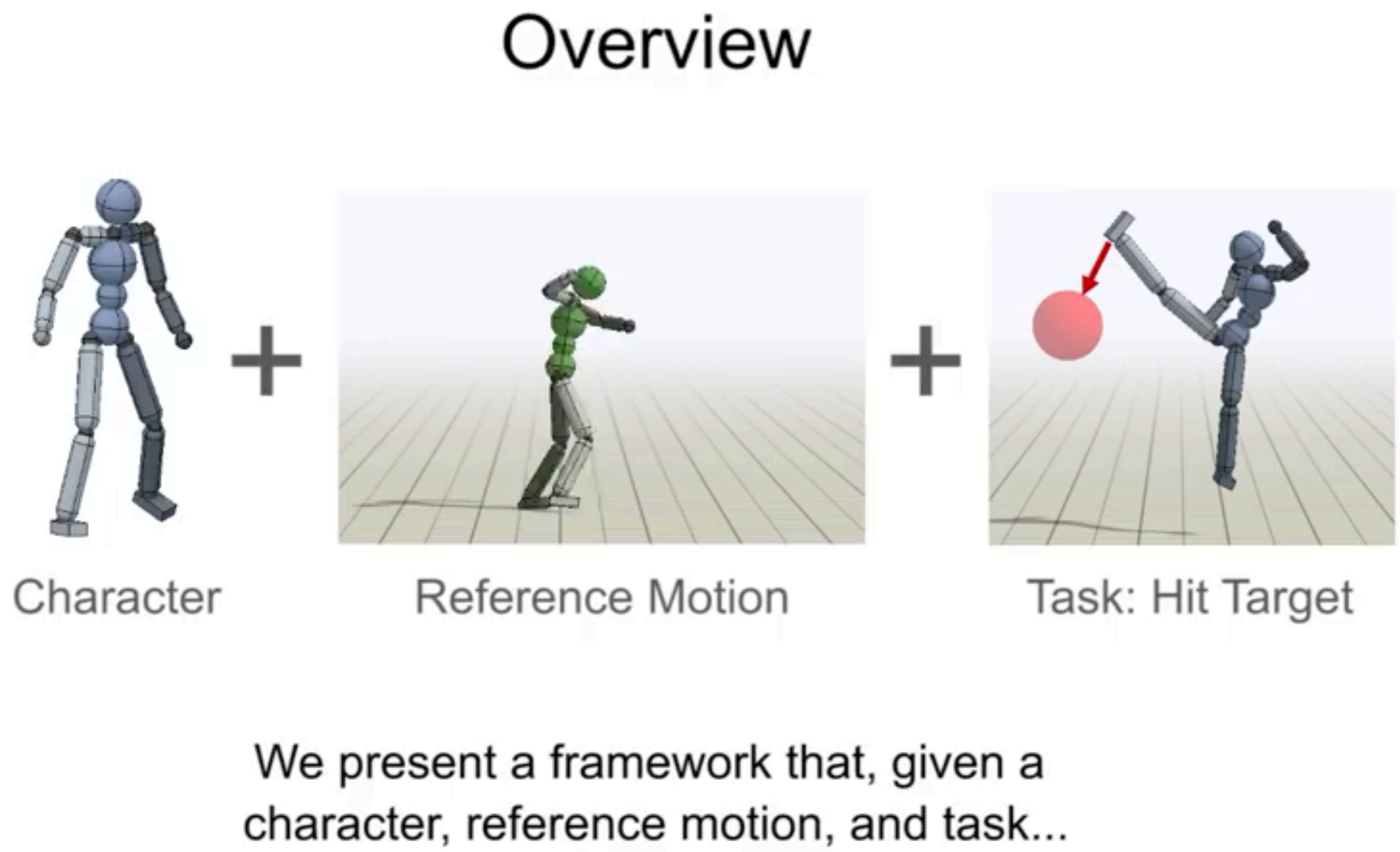
- Learn Controllers that steer character rigs
- Smooth and natural transitions
- Reactions to changes in the environment



[Terrain-Adaptive Locomotion Skills Using Deep Reinforcement Learning, SIGGRAPH 2016]

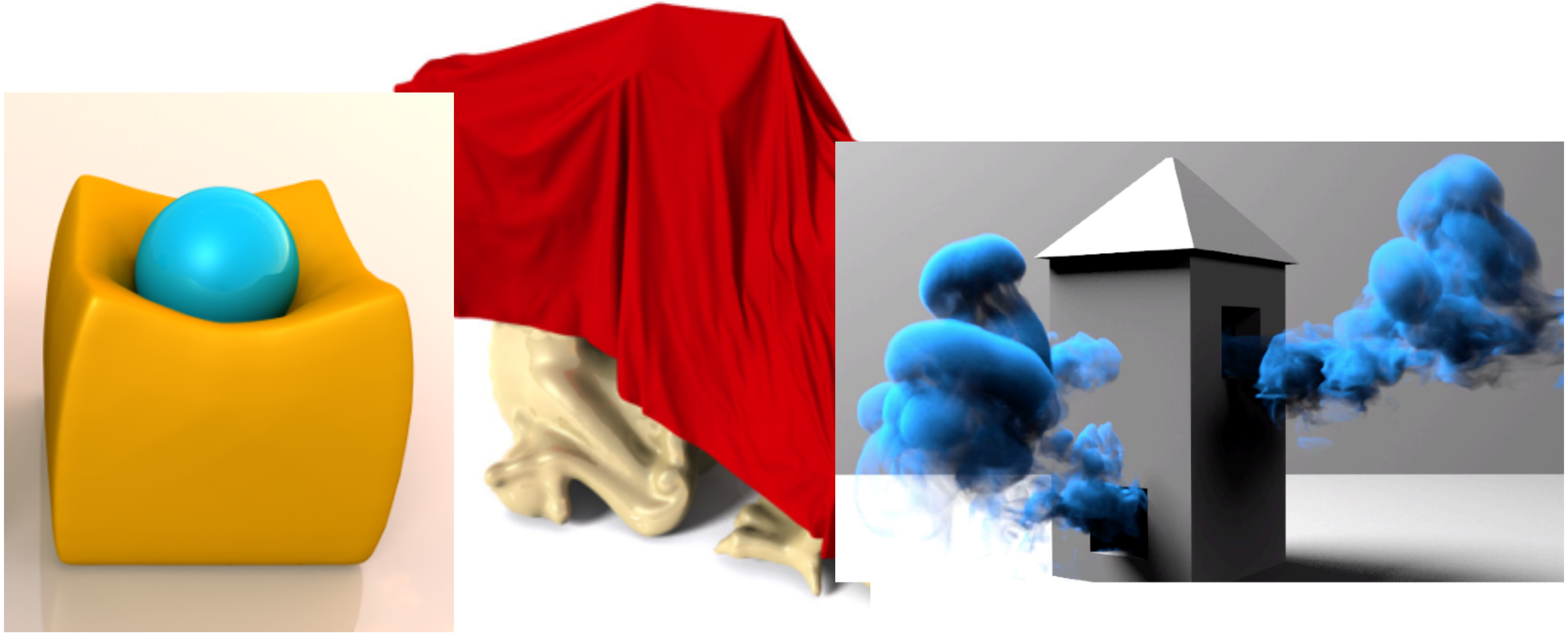
[DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, SIGGRAPH 2018]

Reinforcement Learning



[DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, SIGGRAPH 2018]

Physics-Based Animation



Physics-Based Animation

Skip Theory with Deep Learning?

[No! More on that later…]

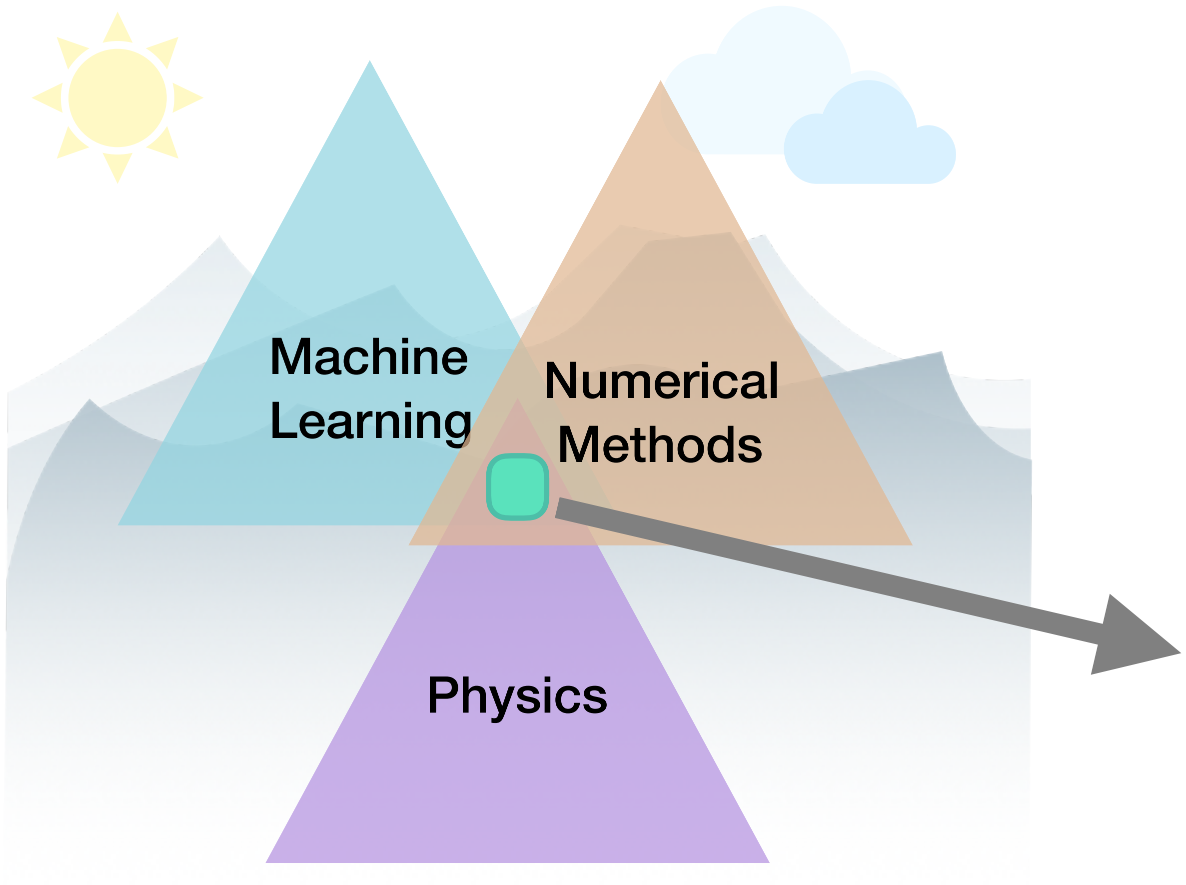
Experiment	Theory	Computation
Observations / data	Model equations	Discrete representation

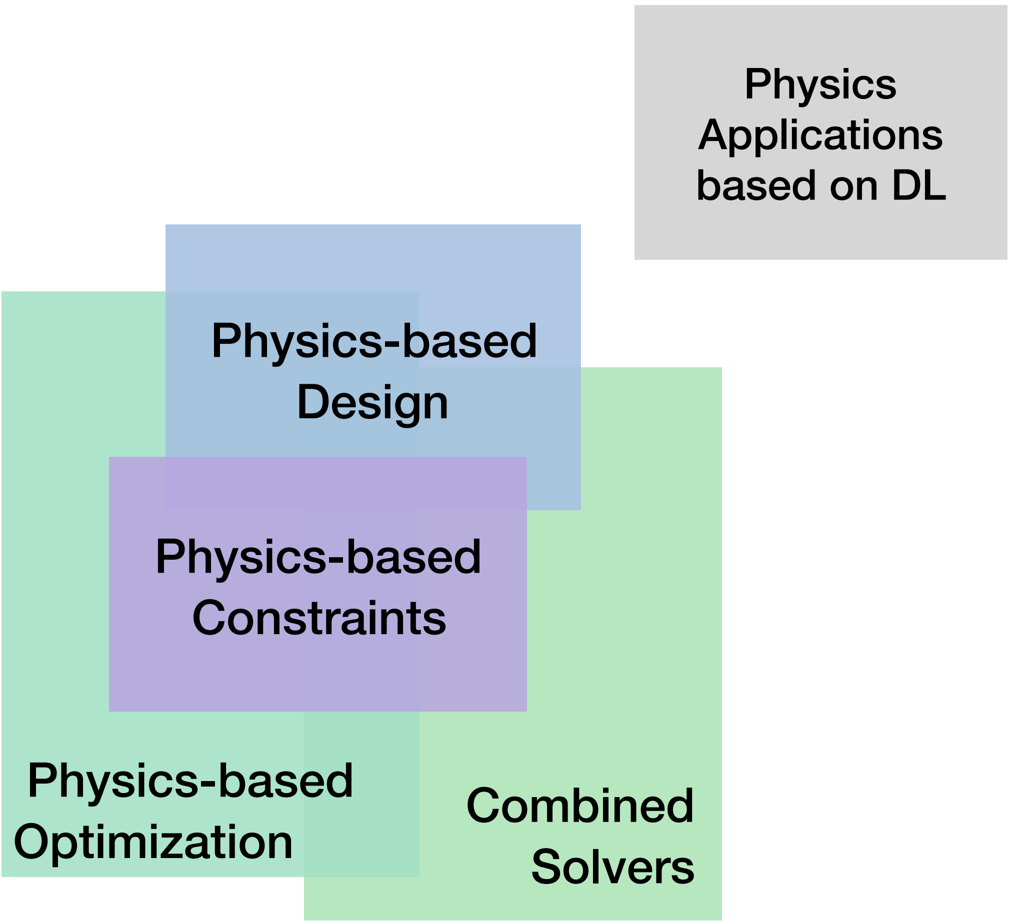
Physics-Based Animation

- Better goal: support solving suitable physical models
- Nature = Partial Differential Equations (PDEs)
- Hence can aim for solving PDEs with deep learning (DL)
- Requirement: “regularity” of the targeted function

> “Bypass the solving of evolution equations when these equations conceptually exist but are not available or known in closed form.” [Kevrekidis et al.]

Physics-Based Deep Learning

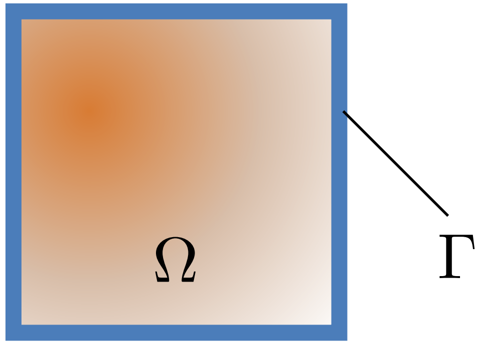




Partial Differential Equations

- Typical problem formulation: unknown function $u(x_1, ..., x_n)$
- PDE of the general form:

$$f\left(x_1, ..., x_n; \frac{\partial u}{\partial x_1}, ..., \frac{\partial u}{\partial x_n}; \frac{\partial^2 u}{\partial^2 x_1}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, ...\right) = 0$$



- Solve in domain Ω, with boundary conditions on boundary Γ
- Traditionally: discretize & solve numerically. Here: also discretize, but solve with DL…

Methodology 1

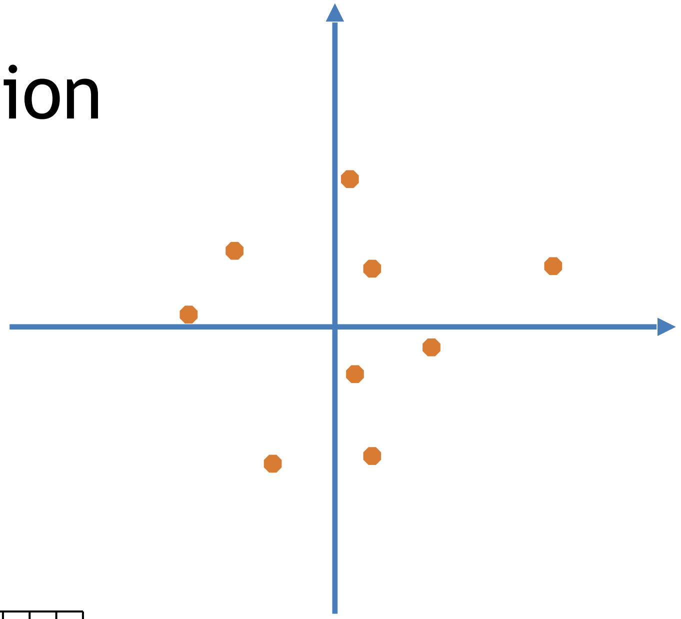
- Viewpoints: *holistic* or *partial*

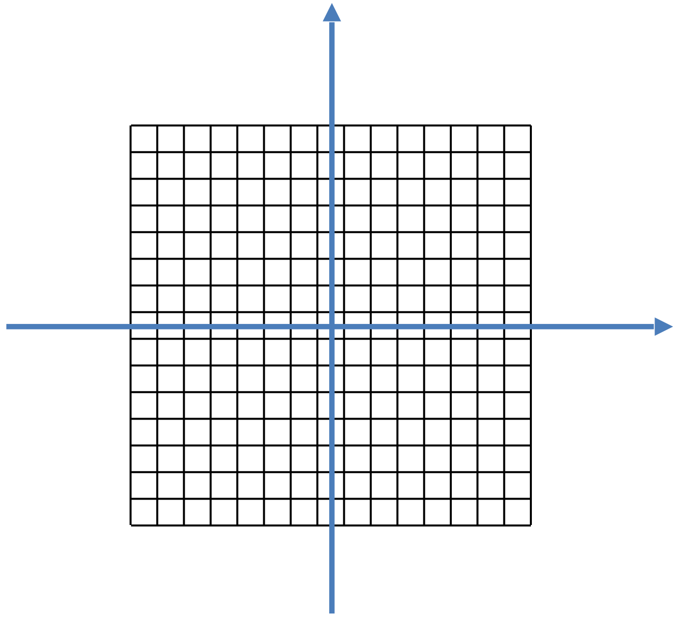
  [partial also meaning "coarse graining" or "sub-grid / up-res"]

- Influences complexity and non-linearity of solution space
- Trade off computation vs accuracy:
  - Target most costly parts of solving
  - Often at the expense of accuracy

Methodology 2

- Consider dimensionality & structure of discretization
- Small & unstructured
  - Fully connected NNs only choice
  - Only if necessary…
- Large & structured
  - Employ convolutional NNs
  - Usually well suited



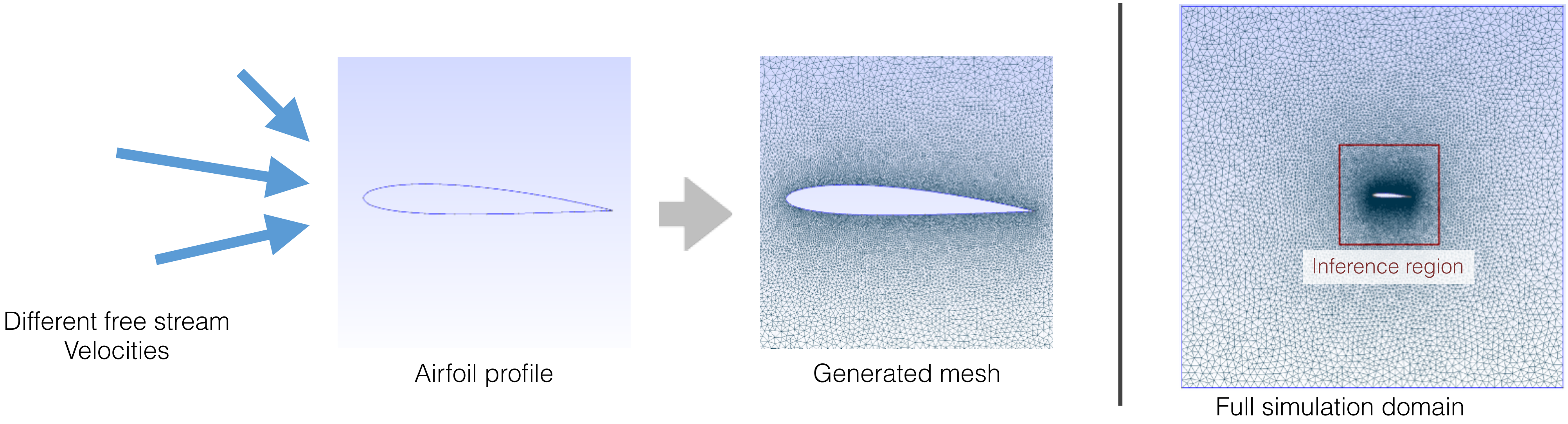


Solving PDEs with DL

- Practical example: *airfoil flow*
    - Given boundary conditions solve stationary flow problem on grid
    - Fully replace traditional solver
    - 2D data, no time dimension
    - I.e., holistic approach with structured data

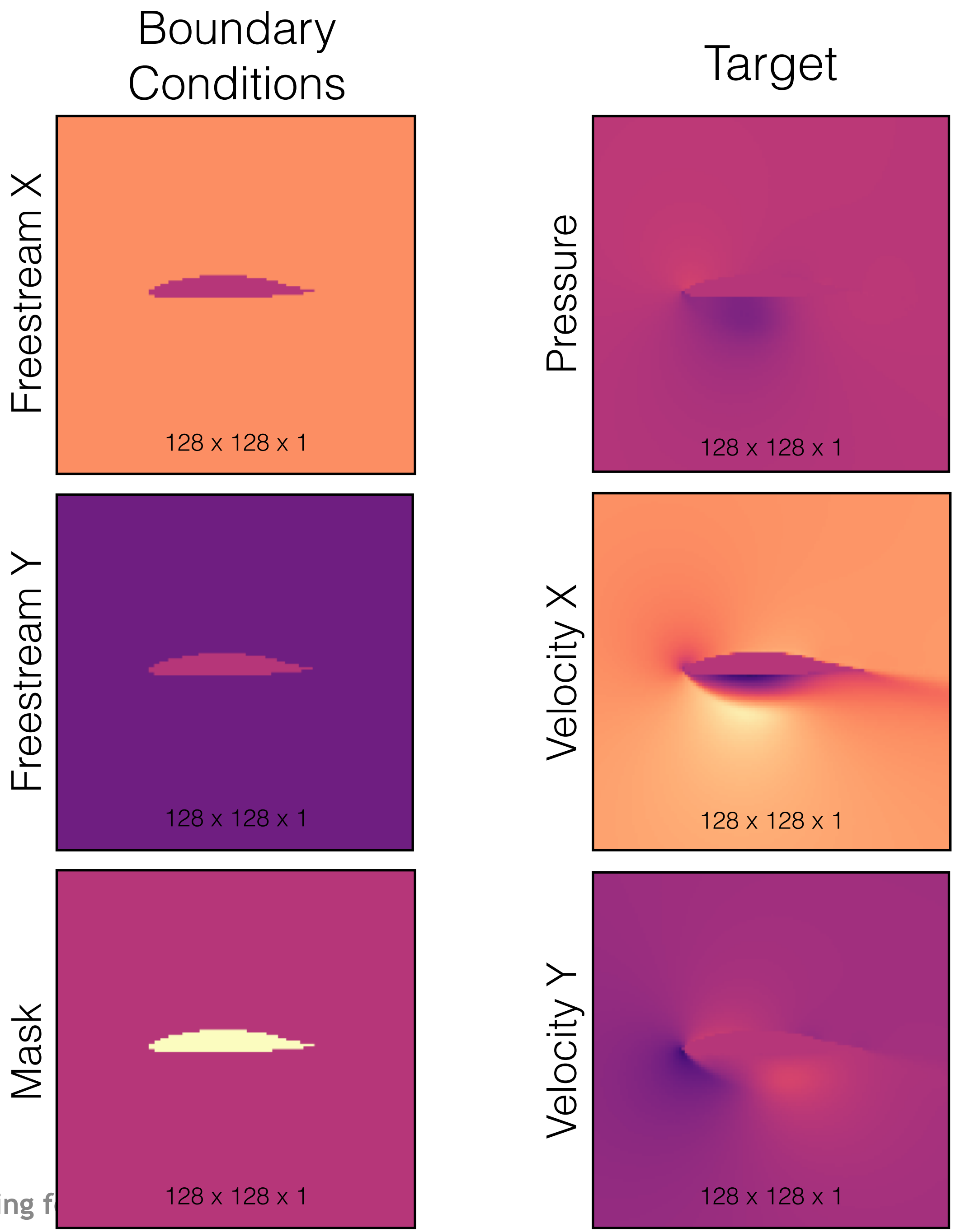
Solving PDEs with DL

- Data generation
- Large number of pairs: input (BCs) - targets (solutions)



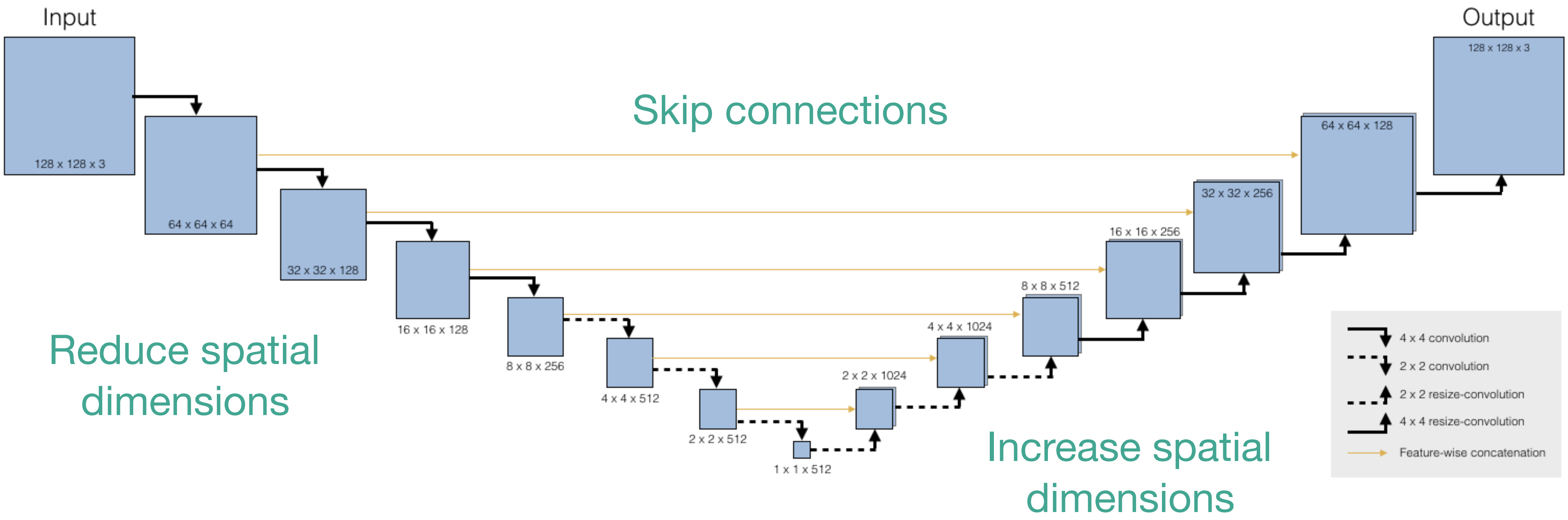
Solving PDEs with DL

- Data generation
- Example pair
- Note - boundary conditions (i.e. input fields) are typically constant
- Rasterized airfoil shape present in all three input fields



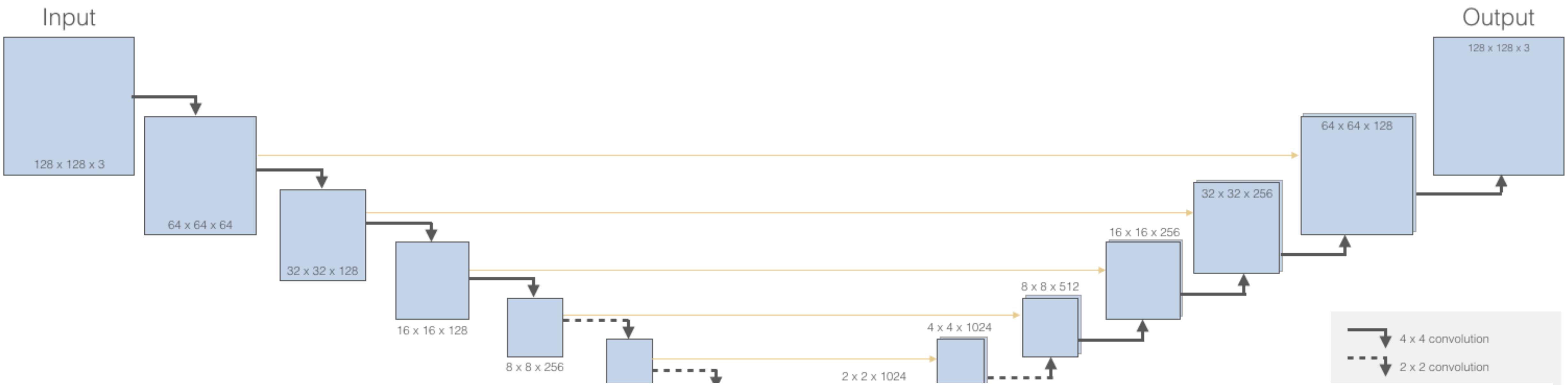
Solving PDEs with DL

- U-net NN architecture



Solving PDEs with DL

- U-net NN architecture



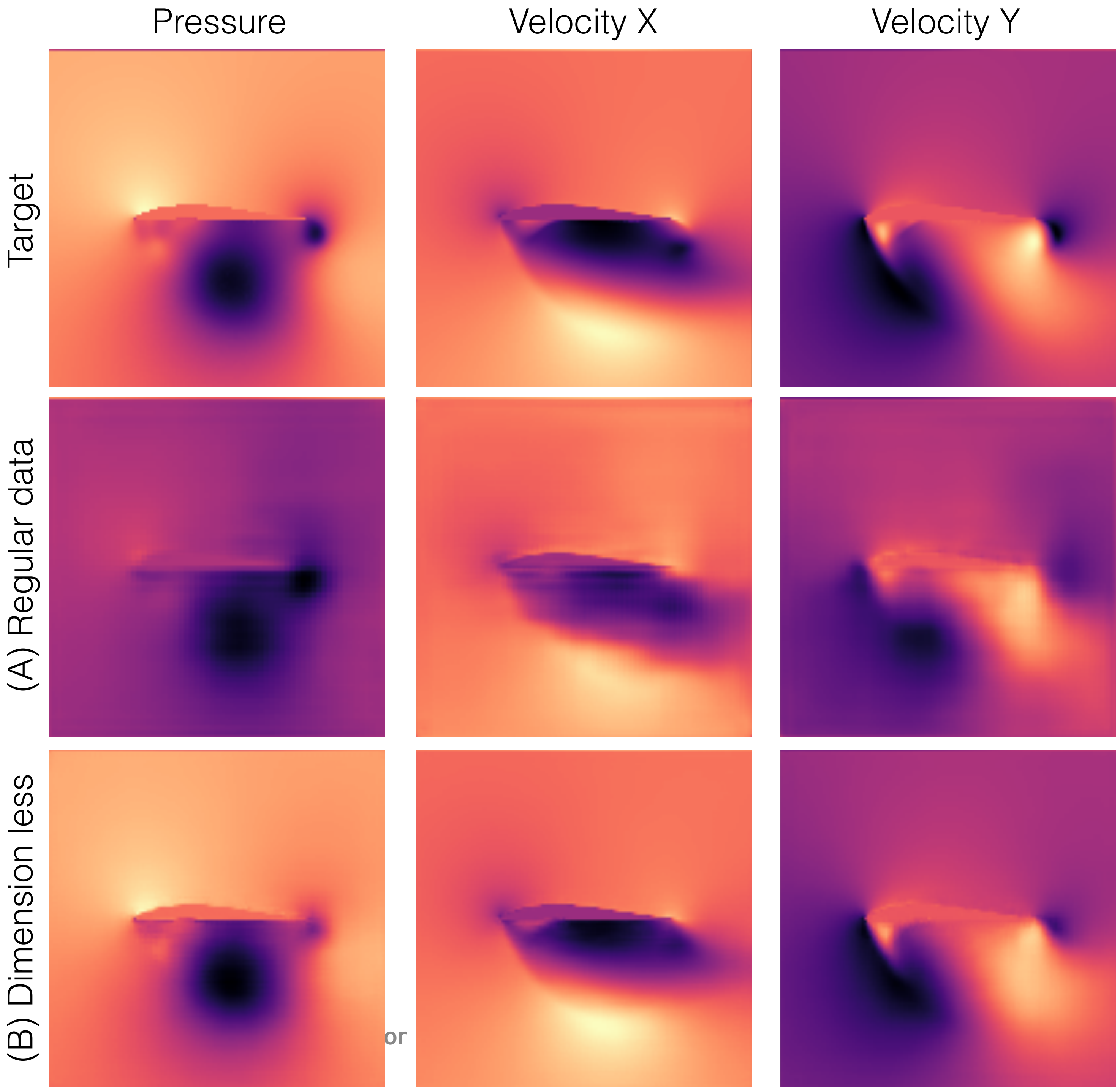
- Unet structure highly suitable for PDE solving
- Makes boundary condition information available throughout
- Crucial for inference of solution

Solving PDEs with DL

- Training: 80.000 iterations with ADAM optimizer
- Convolutions with enough data - no dropout necessary
- Learning rate decay stabilizes models

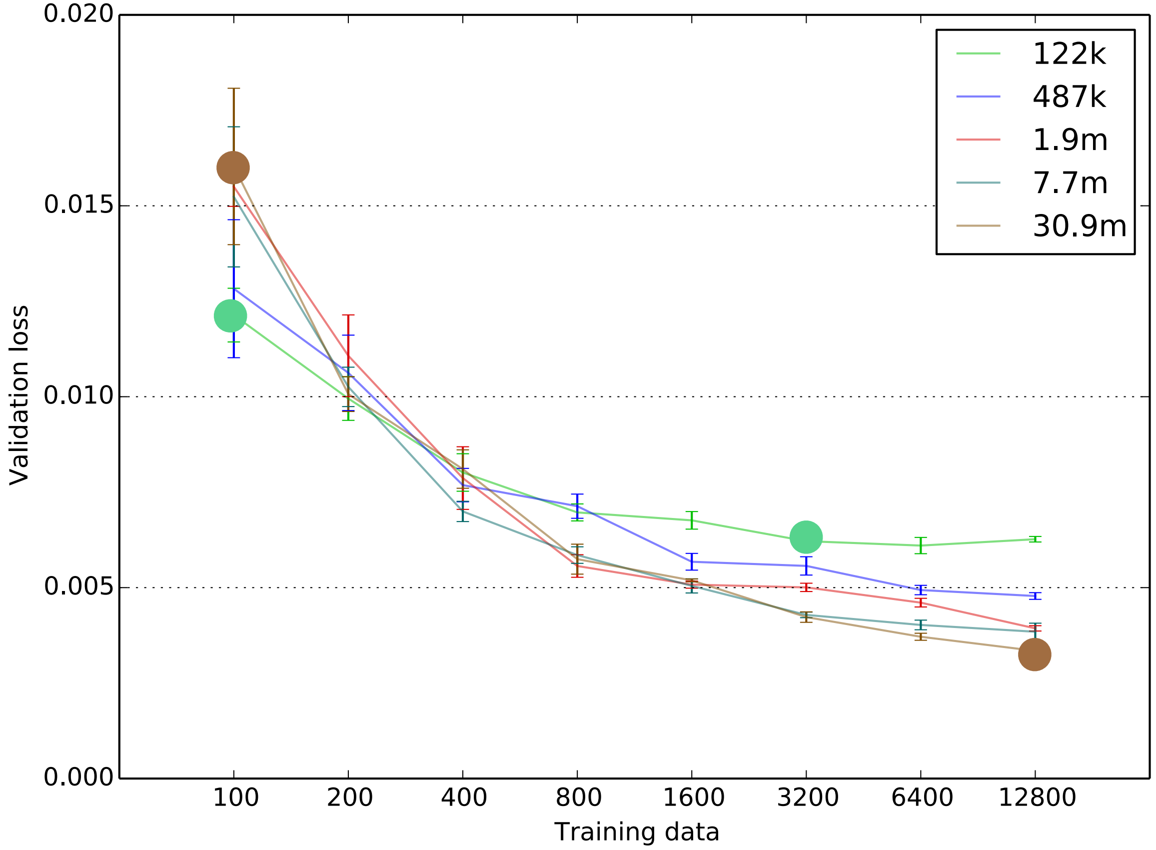
Results

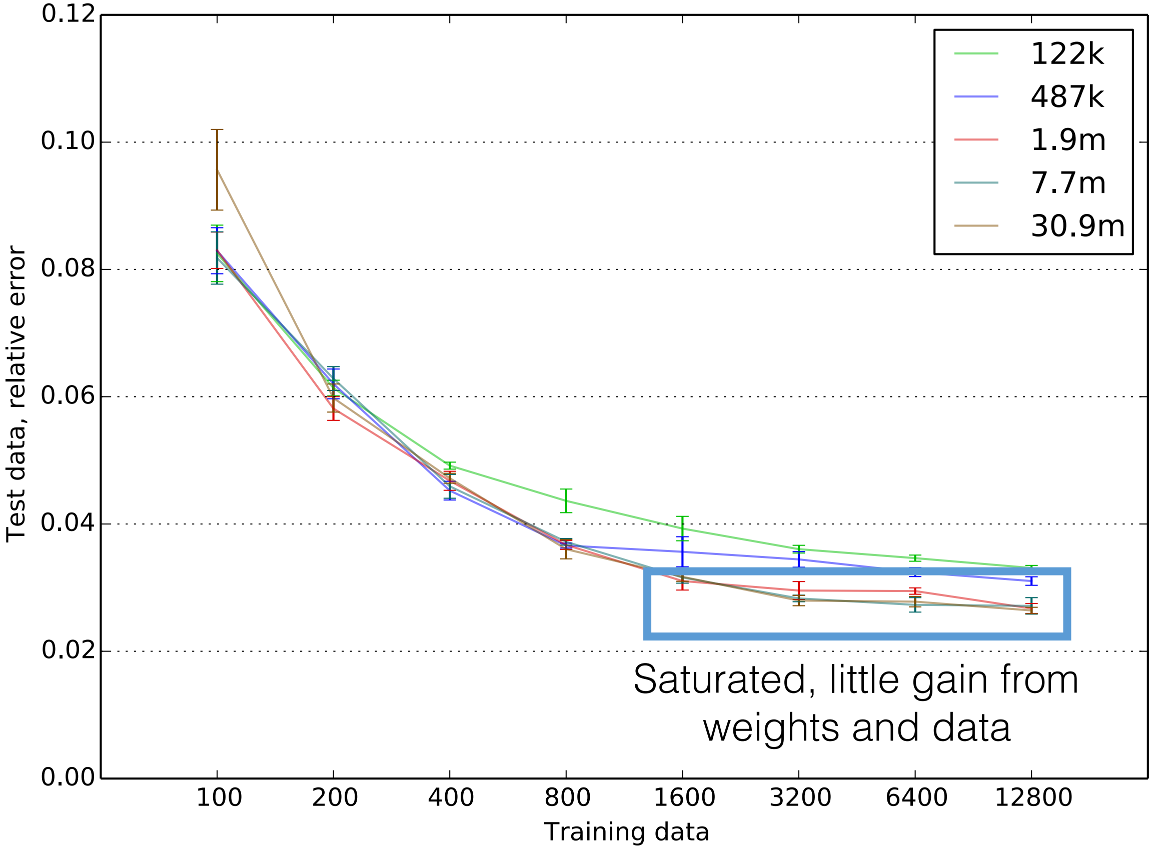
- Use knowledge about physics to simplify space of solutions: make quantities dimension- less
- Significant gains in inference accuracy



Solving PDEs with DL

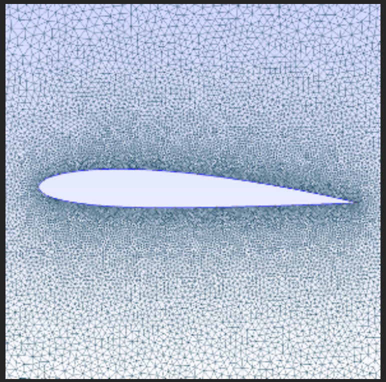
- Validation and test accuracy for different model sizes





Code example

Solving PDEs with DL

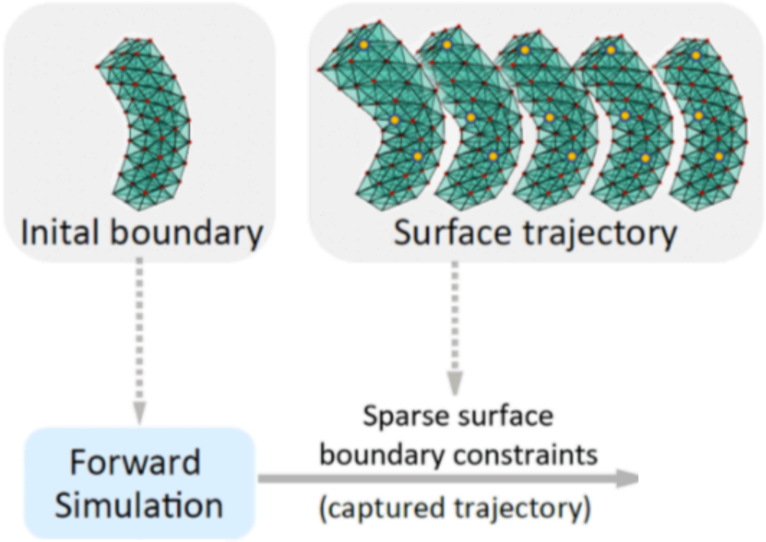


Existing Approaches

- Elasticity
- Cloth
- Fluids

Neural Material - Elasticity

- Learn correction of regular FEM simulation for complex materials
- Numerical simulation with flexible NN for material behavior

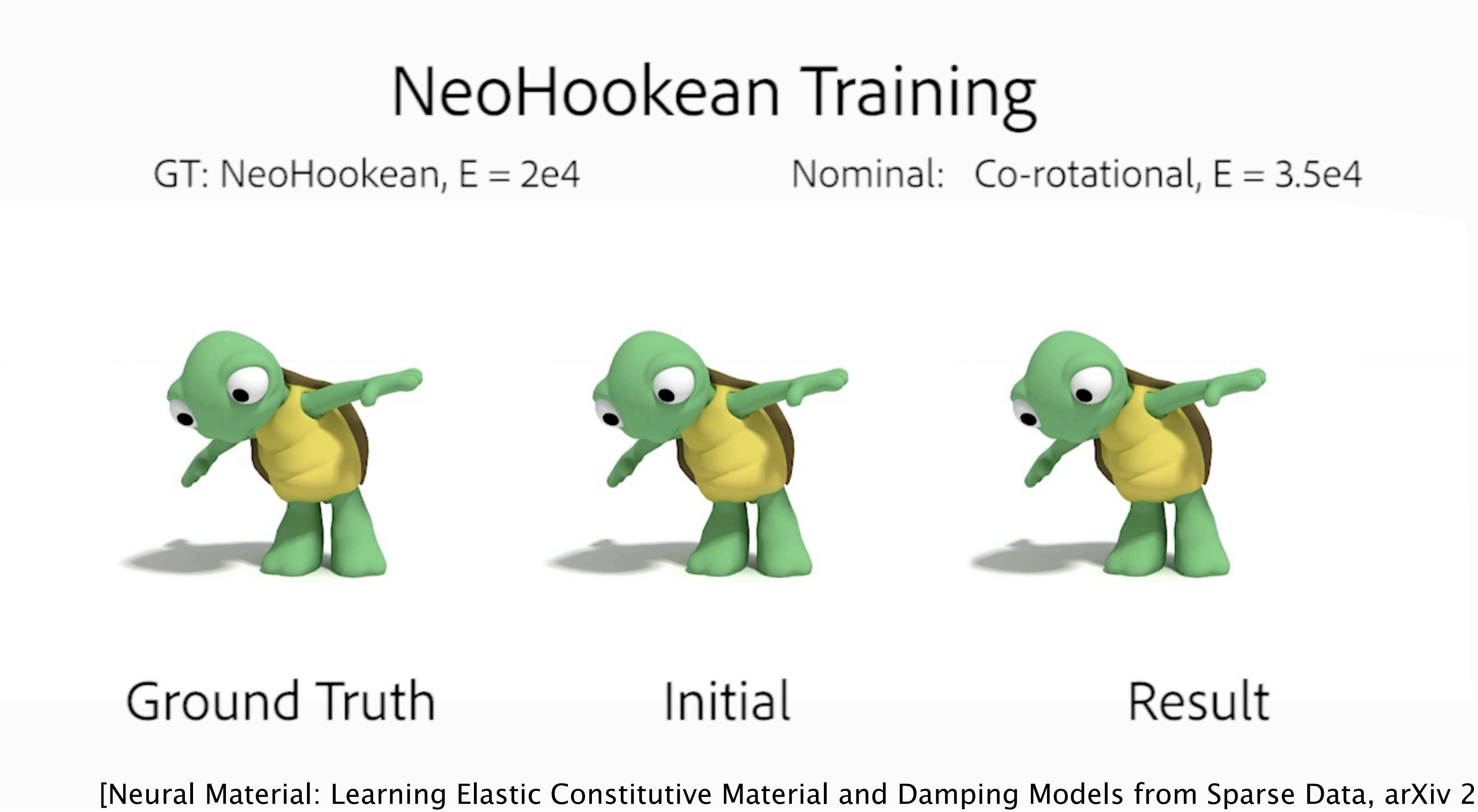


[NNWarp: Neural Network-based Nonlinear Deformation, TVCG 2018]

[Neural Material: Learning Elastic Constitutive Material and Damping Models from Sparse Data, arXiv 2018]

Neural Material - Elasticity

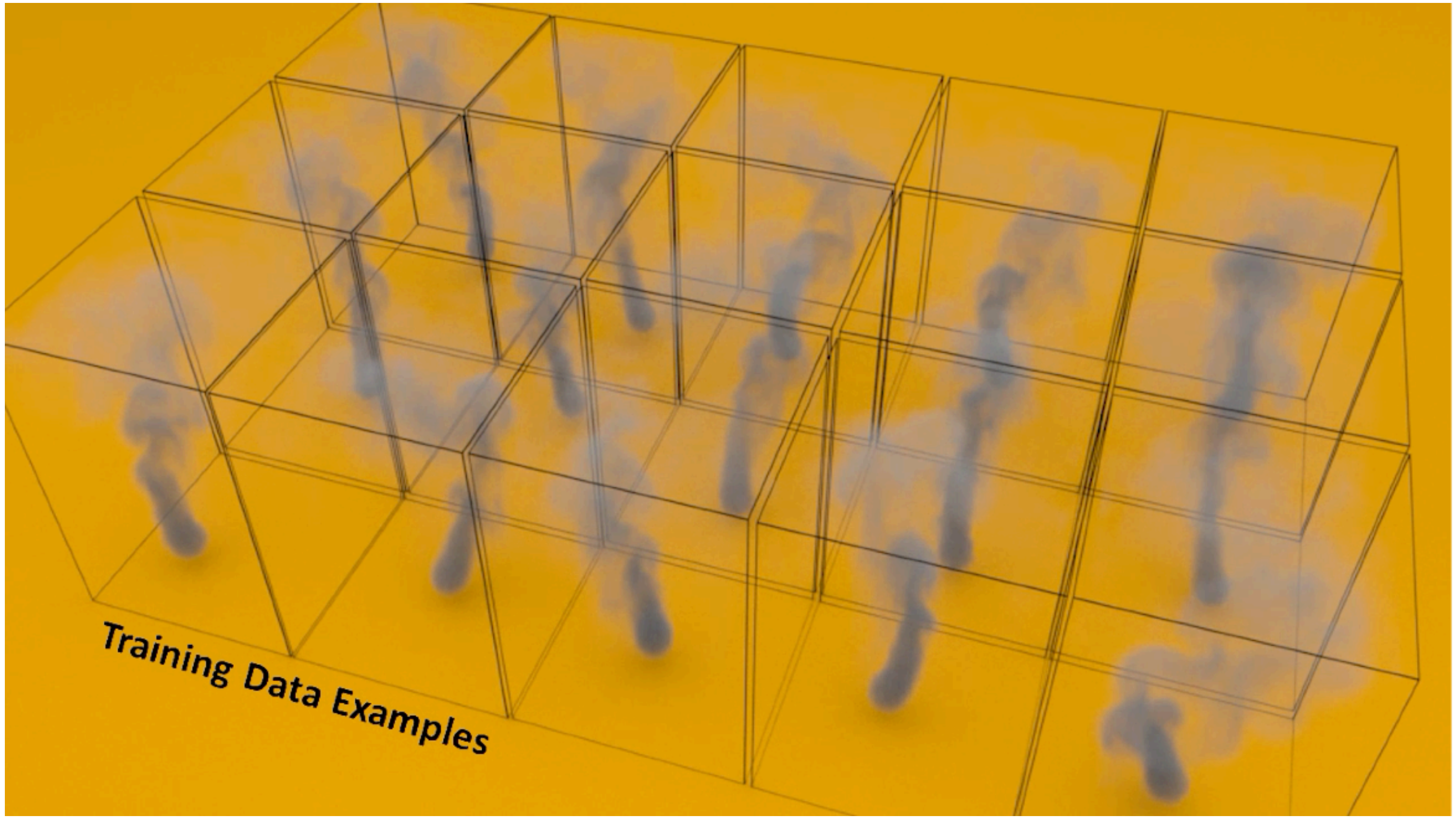
- Learn correction of regular FEM simulation for complex materials



[Neural Material: Learning Elastic Constitutive Material and Damping Models from Sparse Data, arXiv 2018]

Latent Spaces

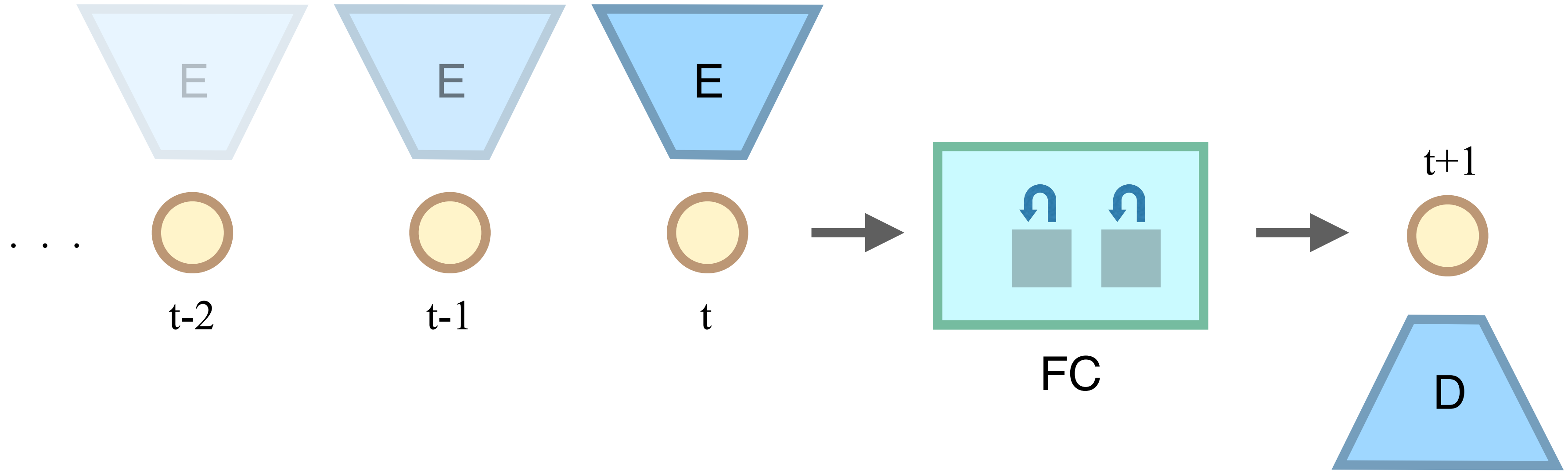
- Learn flexible reduced representation for physics problems



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, EG 2019]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, EG 2019]

Latent Spaces

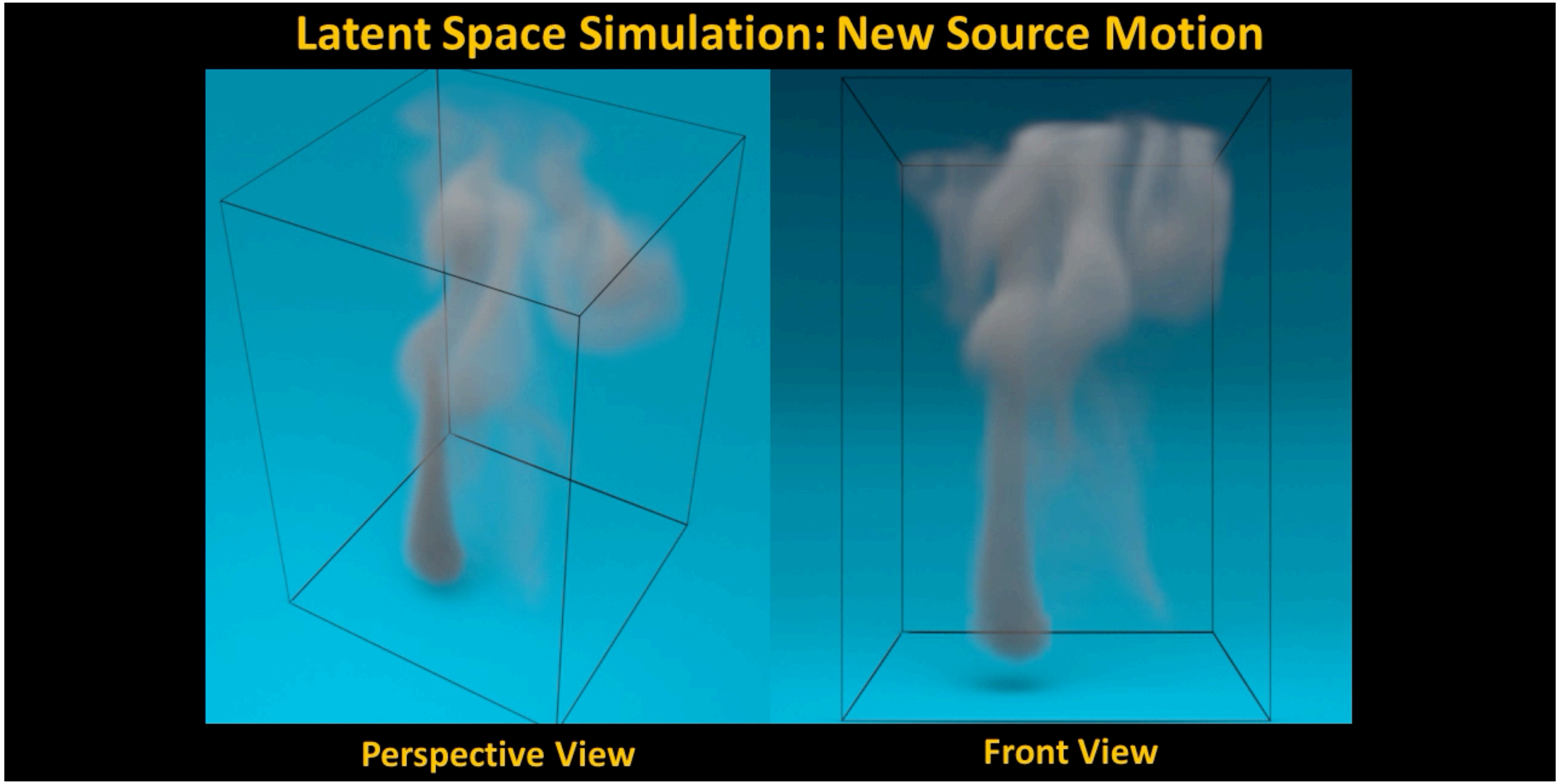
- Learn flexible reduced representation for physics problems
  - Employ Encoder part (E) of Autoencoder network to reduce dimensions
  - Predict future state in latent space with FC network
  - Use Decoder (D) of Autoencoder to retrieve volume data



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, EG 2019]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, EG 2019]

Latent Spaces

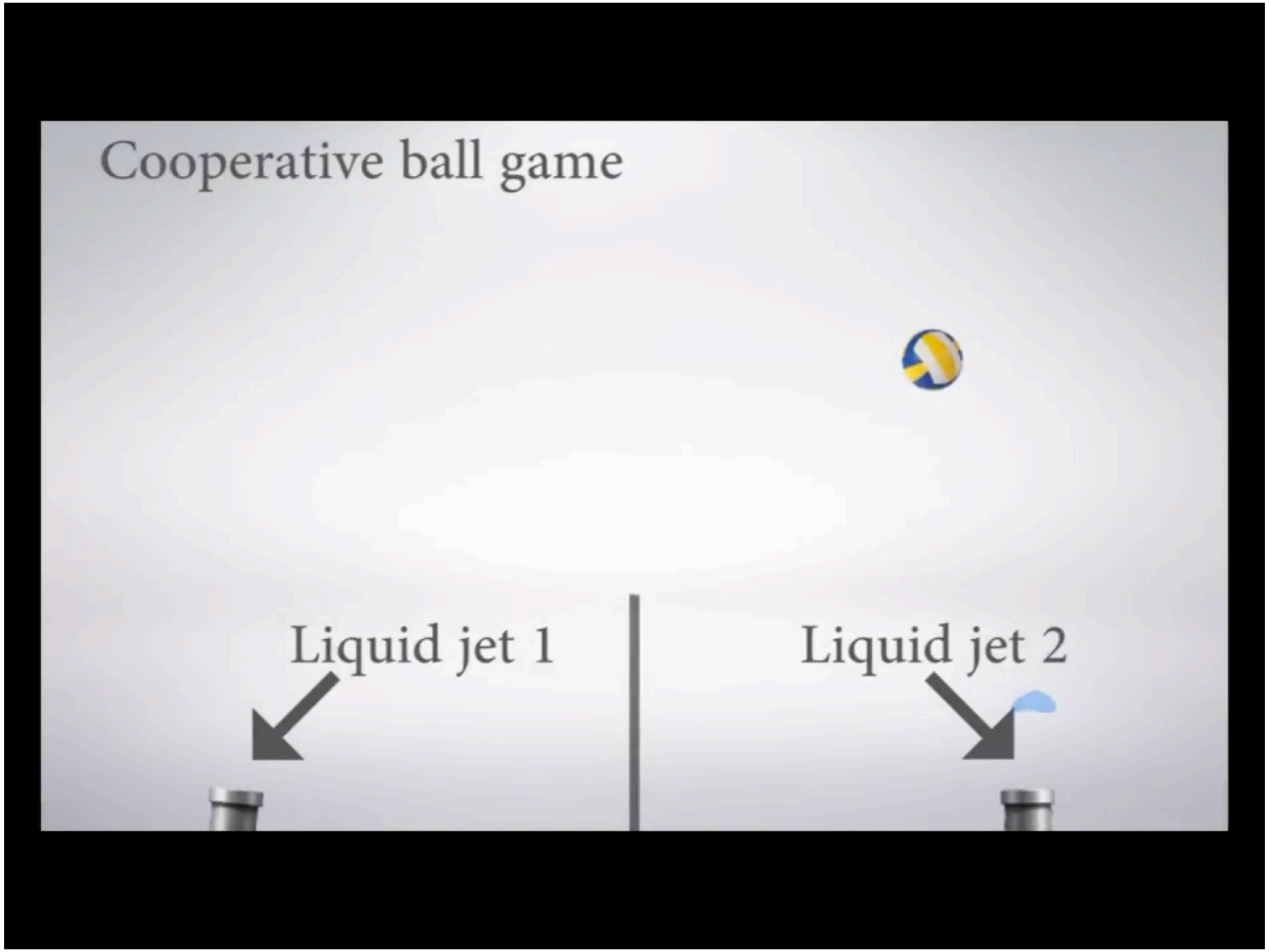
- Learn flexible reduced representation for physics problems



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, EG 2019]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, EG 2019]

Latent Spaces

- In combination with Reinforcement Learning



[Fluid Directed Rigid Body Control using Deep Reinforcement Learning, SIGGRAPH 2018]

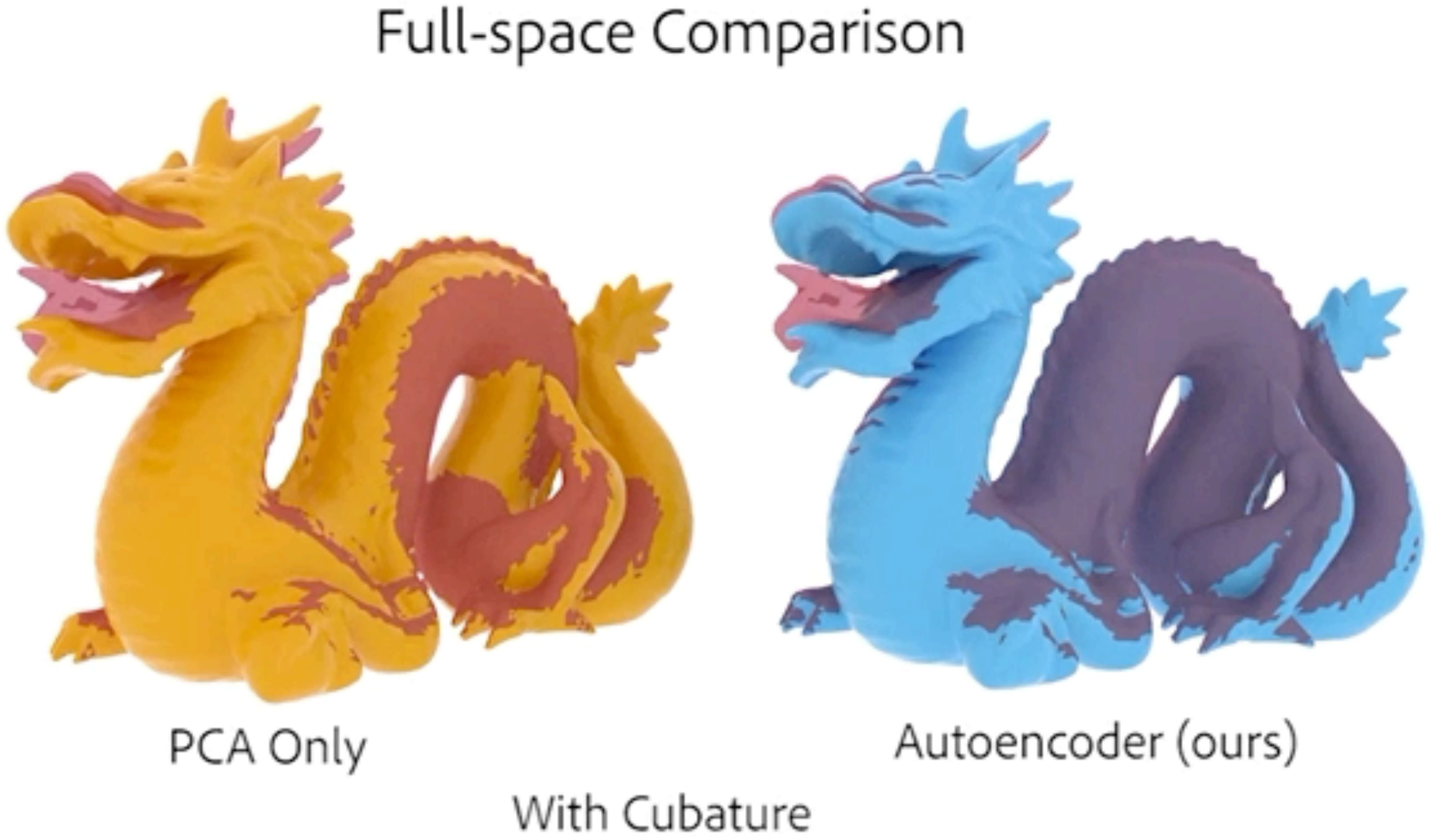
Latent Spaces

- For elasticity problems

[Latent-space Dynamics for Reduced Deformable Simulation, EG 2019]

Latent Spaces

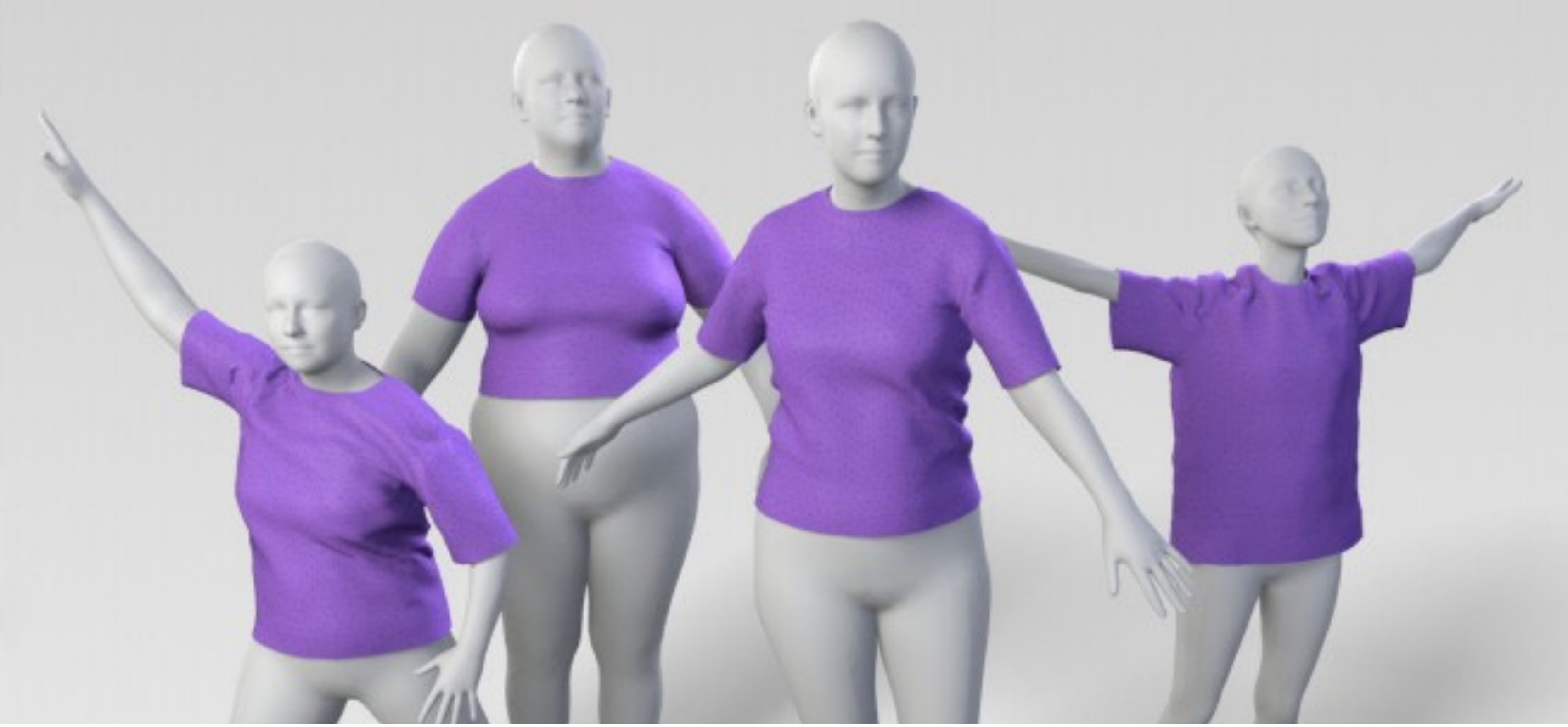
- For elasticity problems



[Latent-space Dynamics for Reduced Deformable Simulation, EG 2019]

Latent Spaces

- For cloth (adaptation to different body shapes)



[Learning-Based Animation of Clothing for Virtual Try-On, EG 2019]

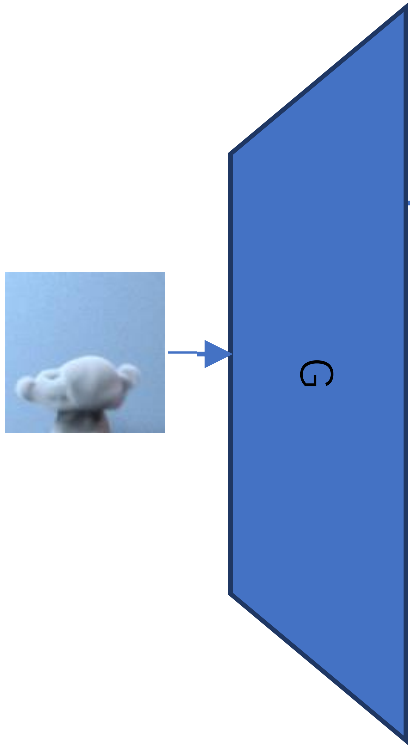
Temporal Data

- Generative model for 3D plus time
- Input domain: low resolution 3D volumes
- Output: high-resolution 3D volumes
- Auxiliary goal: match temporal evolution of target domain (high-res. data)

[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

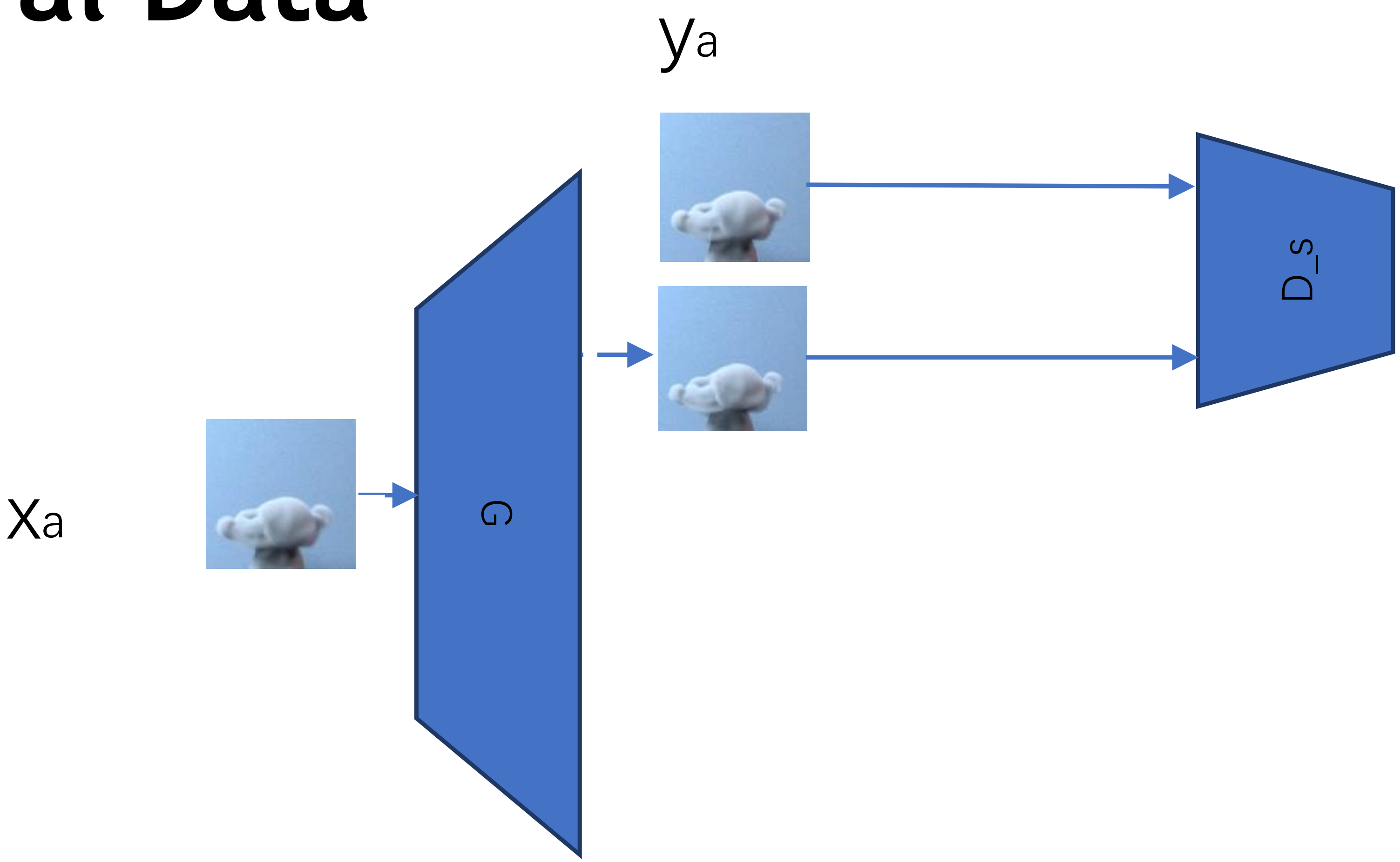
Temporal Data

Xa



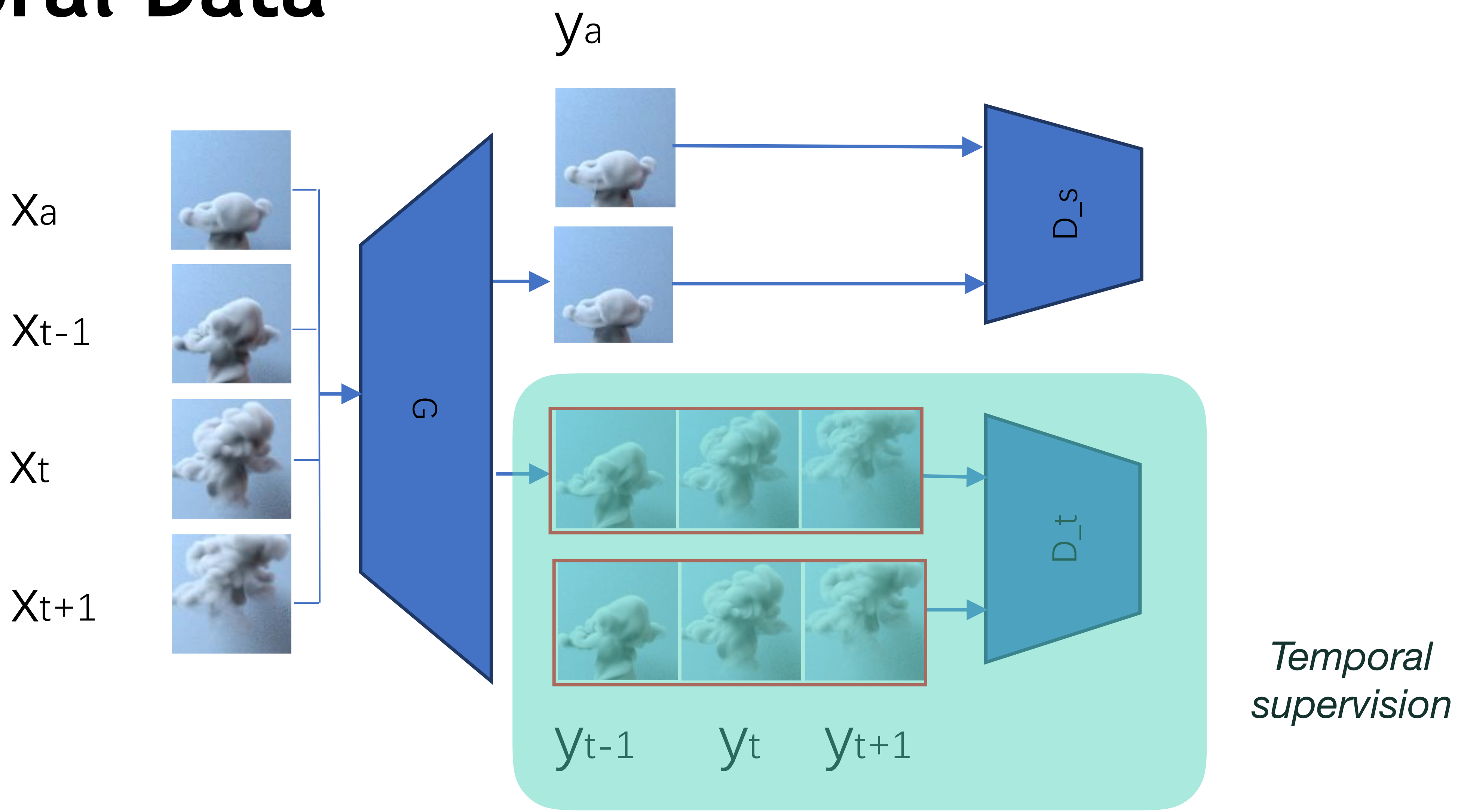
[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data



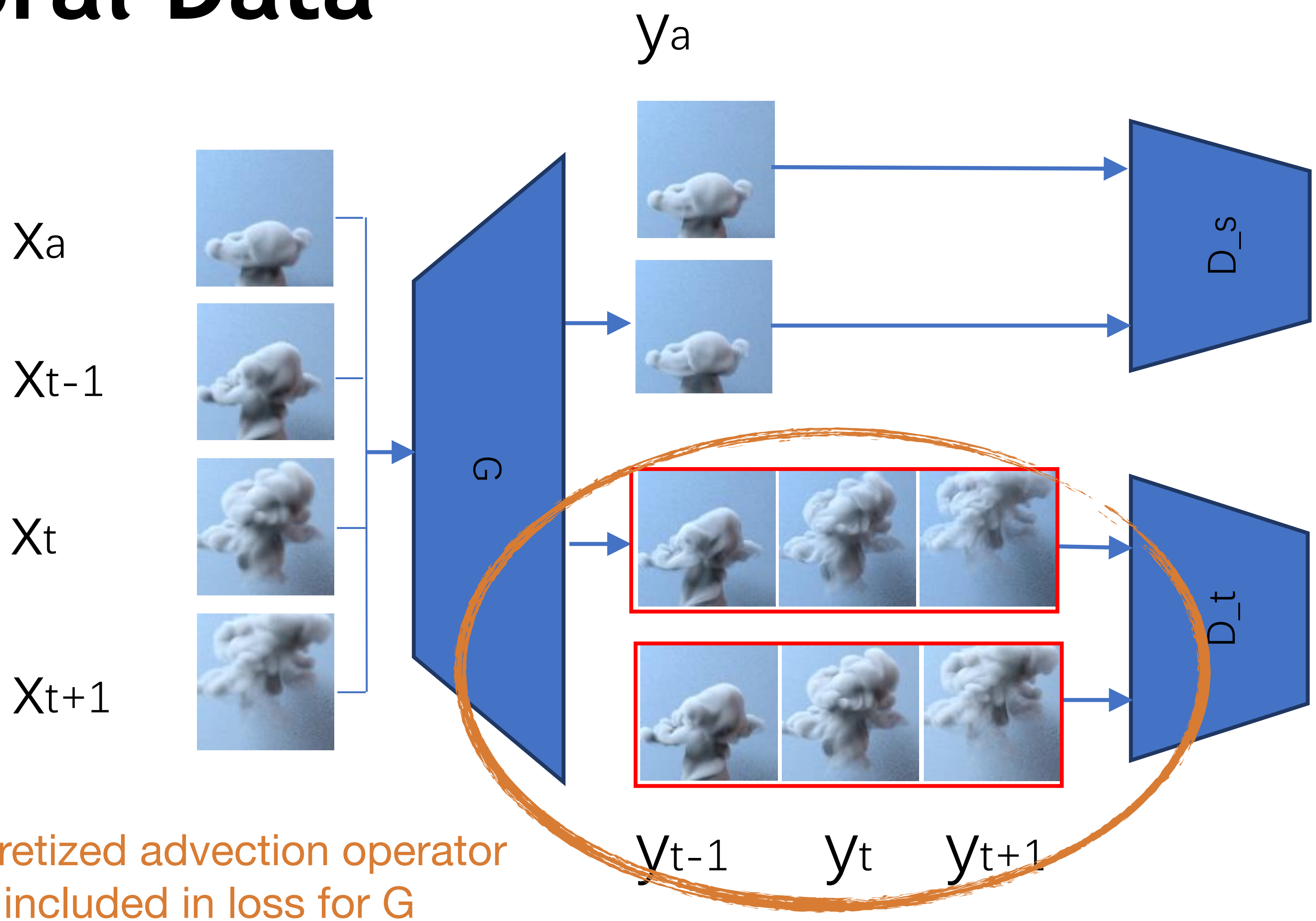
[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data



[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

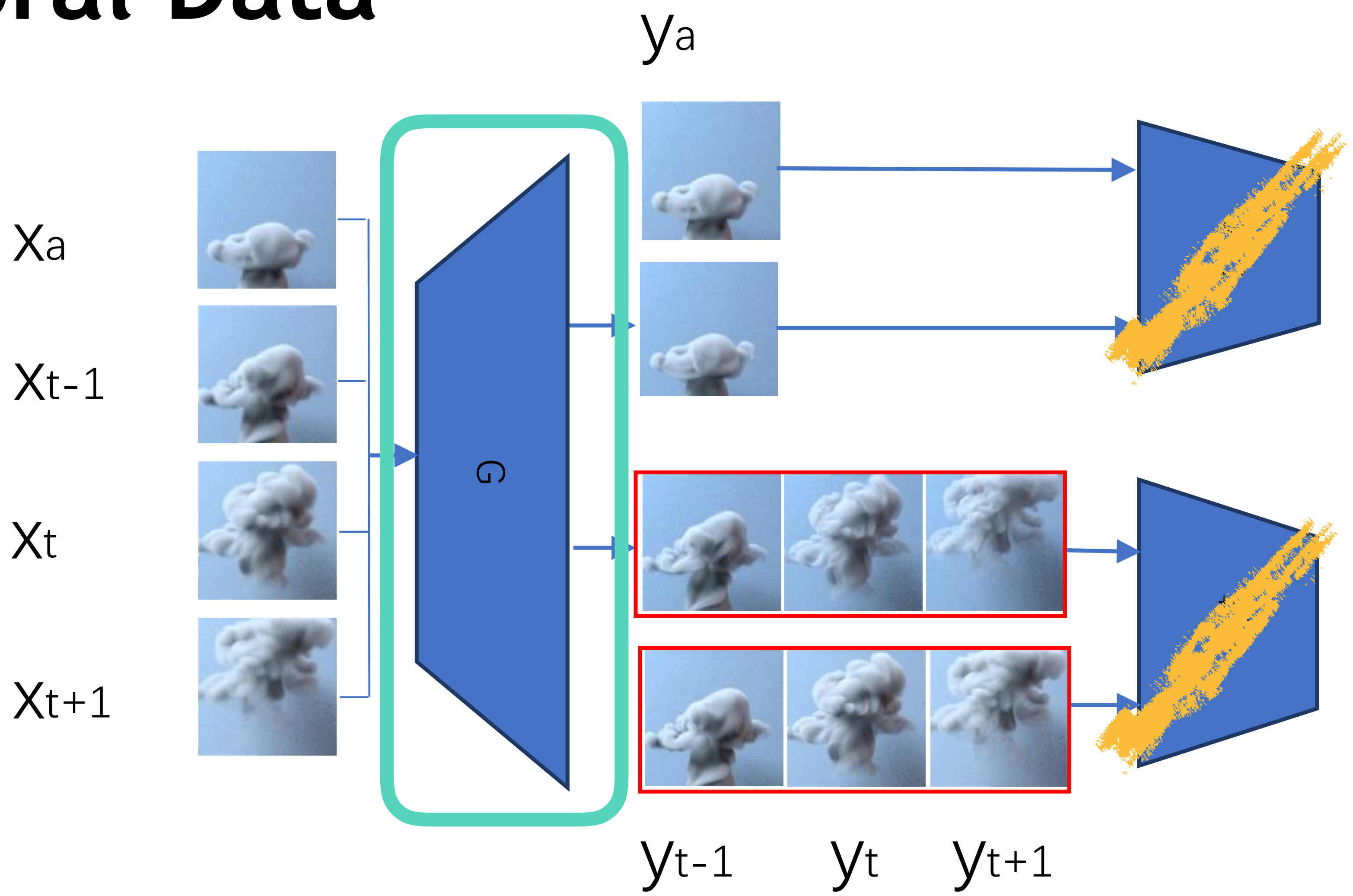
Temporal Data



Discretized advection operator
included in loss for G

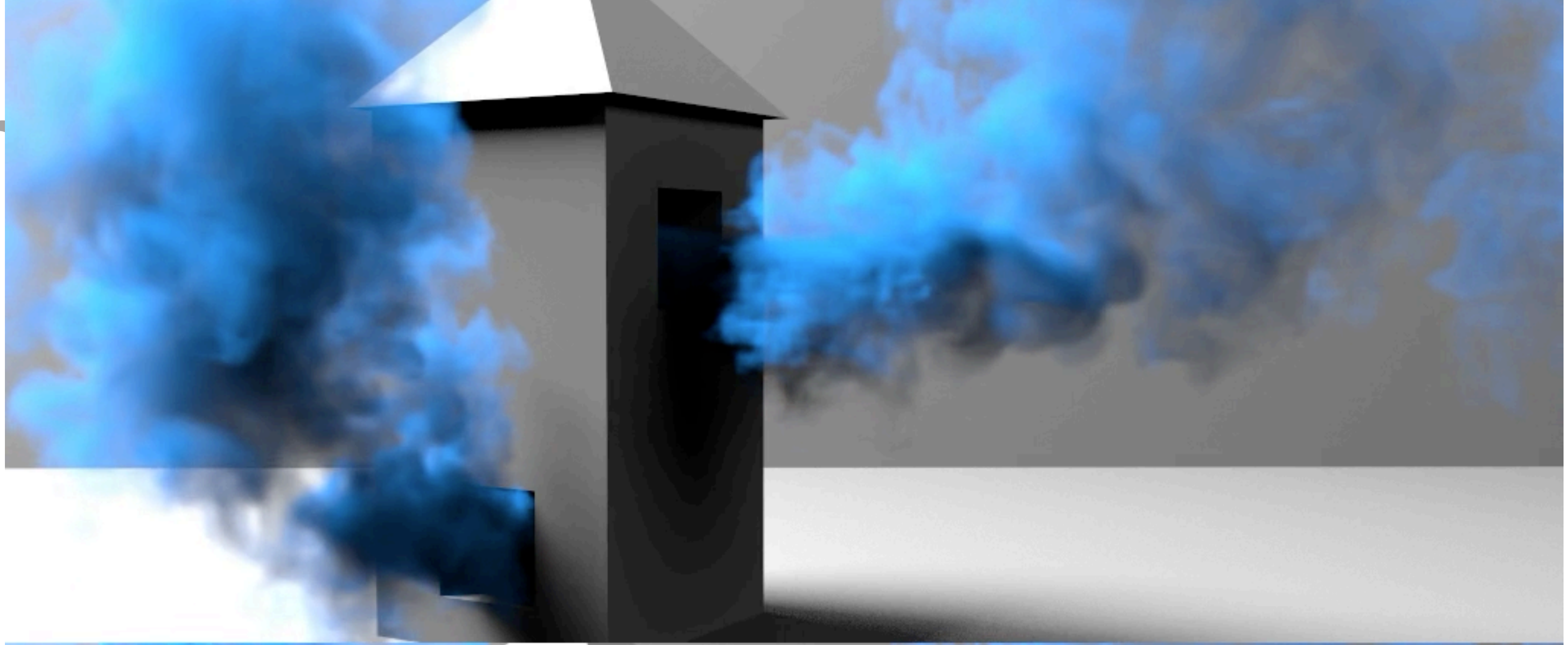
[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data



[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Tempor



Low-res Input

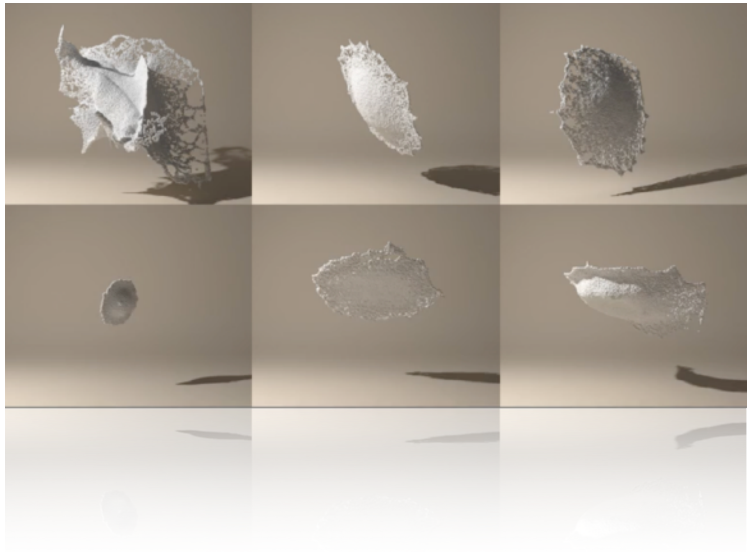
Result

[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data

[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

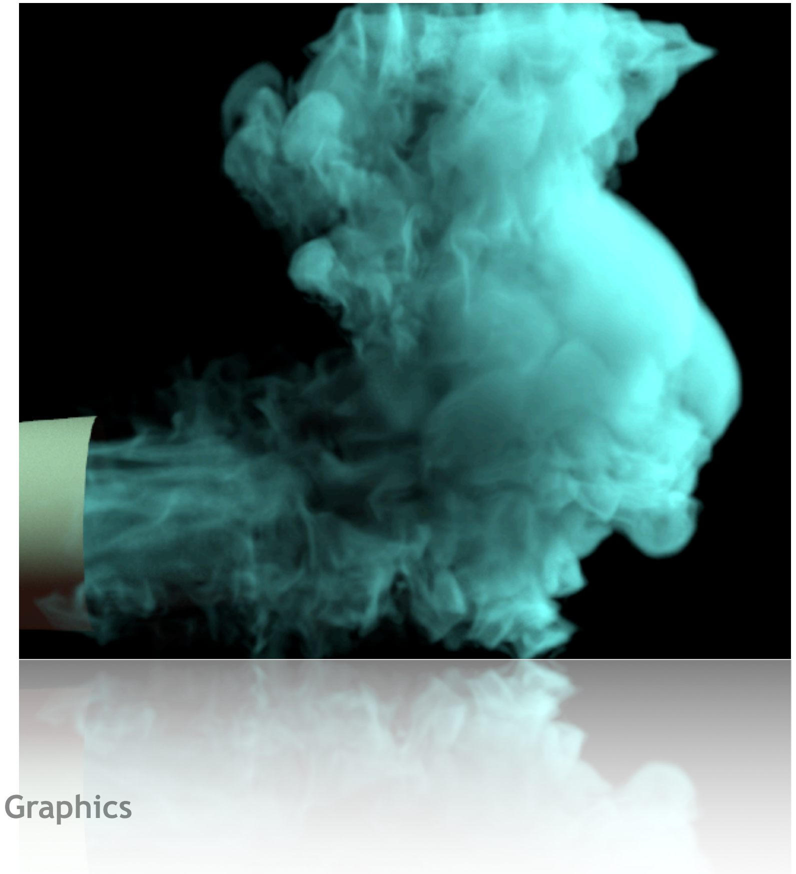
Summary



- Checklist for solving PDEs with DL:
  - ✓ Model? (Typically given)
  - ✓ Data? Can enough training data be generated?
  - ✓ Which NN Architecture?
  - ✓ Fine tuning: learning rate, number of layers & features?
  - ✓ Hyper-parameters, activation functions etc.?

Summary

- Approach PDE solving with DL like solving with traditional numerical methods:
  - Find closest example in literature
  - Reproduce & test
  - Then vary, adjust, refine …
- Main change: Data pipeline



Deep Learning - Outlook

- DL provides a powerful computational tool
- Open challenges:
  - Theoretical guarantees
  - Ethical questions
  - “Next level” of representation learning

The End - Thank you!

Scan me

Course Information (slides/code/comments)

http://geometry.cs.ucl.ac.uk/creativeai/

thrive
SIGGRAPH2019
LOS ANGELES • 28 JULY – 1 AUGUST