CreativeAI: Deep Learning for Graphics

Motion & Physics

Niloy Mitra	Iasonas Kokkinos	Paul Guerrero	Nils Thuerey	Tobias Ritschel
UCL	UCL/Facebook	UCL	TUM	UCL

facebook Artificial Intelligence Research

Technische Universität München

Computer Animation

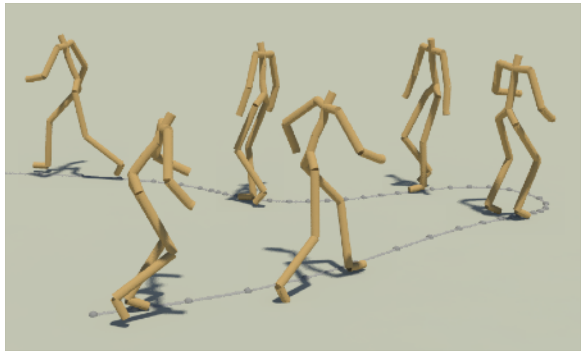
- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smooth $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Distar $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$
- Rendering $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$

- Motion over time
- Loads of data, expensive
- Relationships between spatial and temporal changes

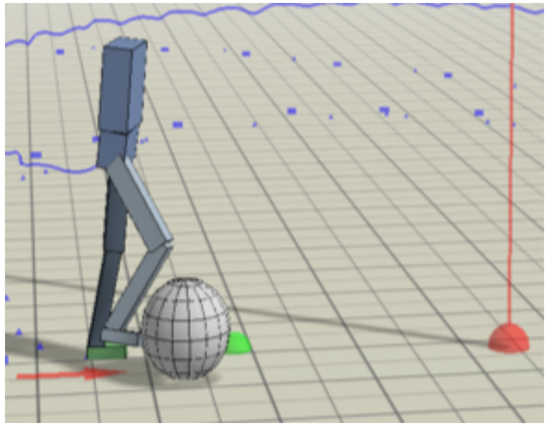
Character Animation

- Learn controllers for character rigs
- Powerful and natural
- Beyond the scope of this course…

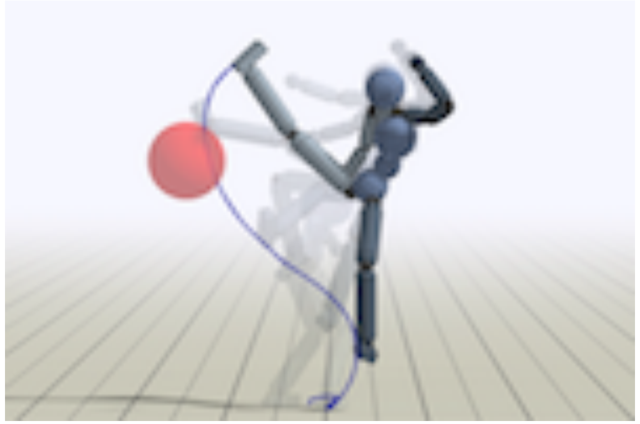
[Mode-Adaptive Neural Networks for Quadruped Motion Control, SIGGRAPH 2018]



[A Deep Learning Framework for Character Motion Synthesis and Editing, SIGGRAPH 2016]



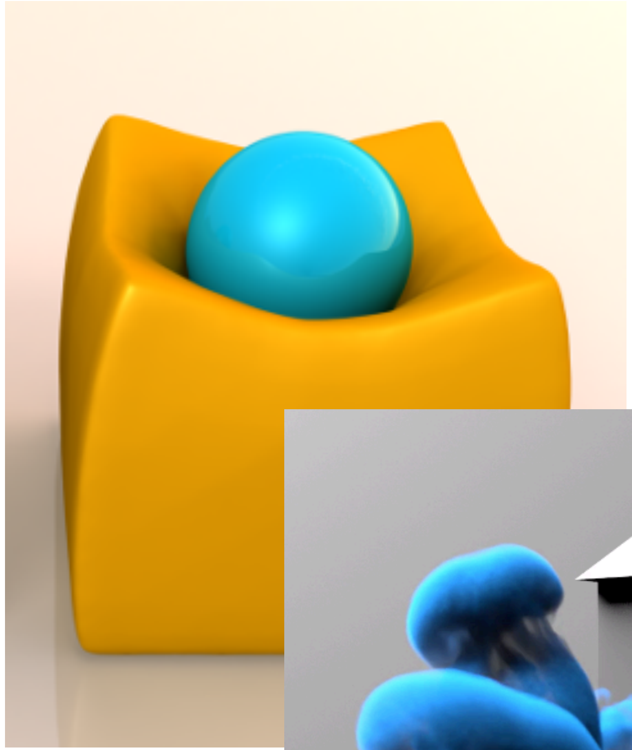
[DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning, SIGGRAPH 2017]

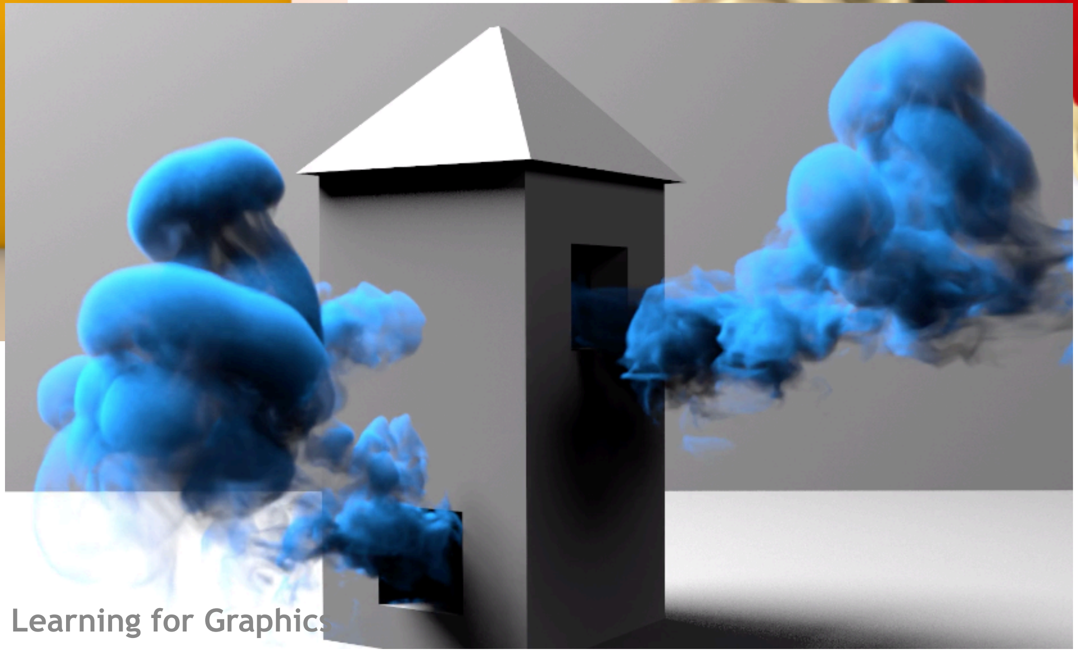


[DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, SIGGRAPH 2018]

Physics-Based Animation

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





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 we are aiming for solving PDEs with deep learning (DL)
- Requirement: “regularity” of the targeted function

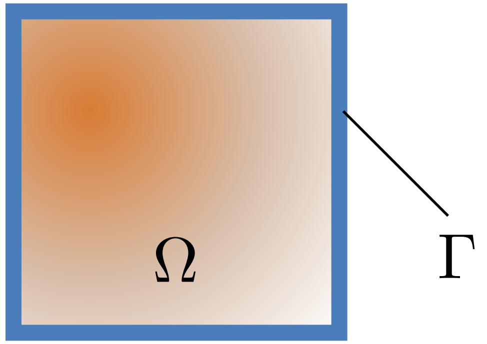
> “Bypass the solving of evolution equations when these equations conceptually exist but are not available or known in closed form.” [Kevrekidis et al.]

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…



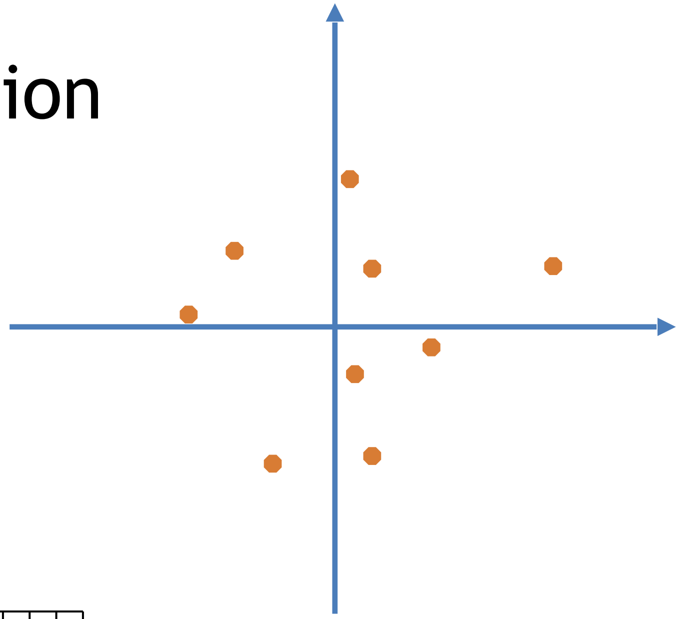
Methodology 1

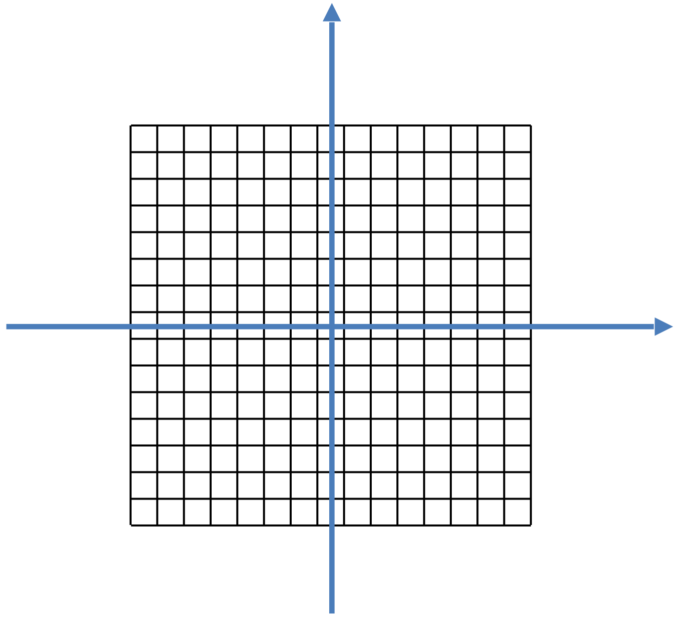
- Viewpoints: *holistic* or *partial*

  [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



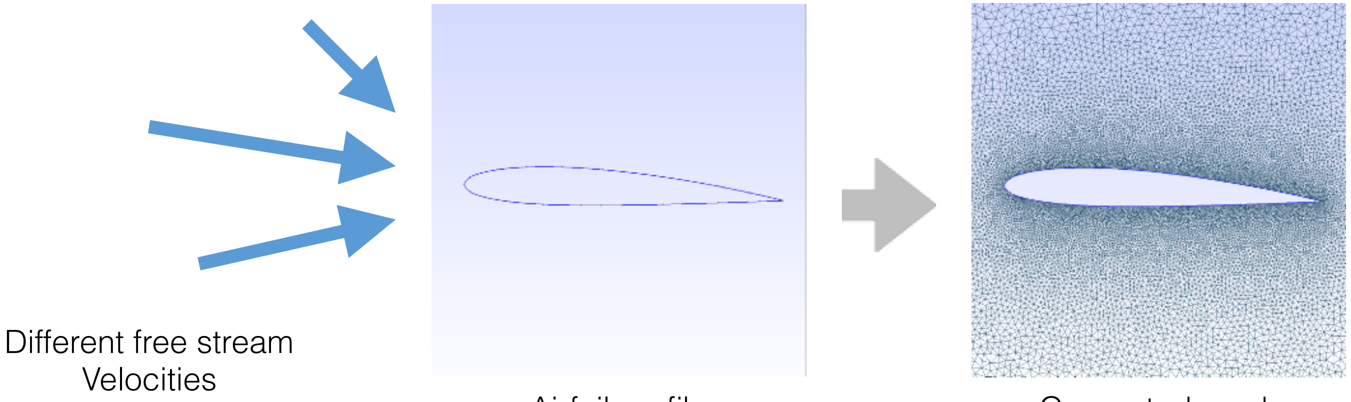


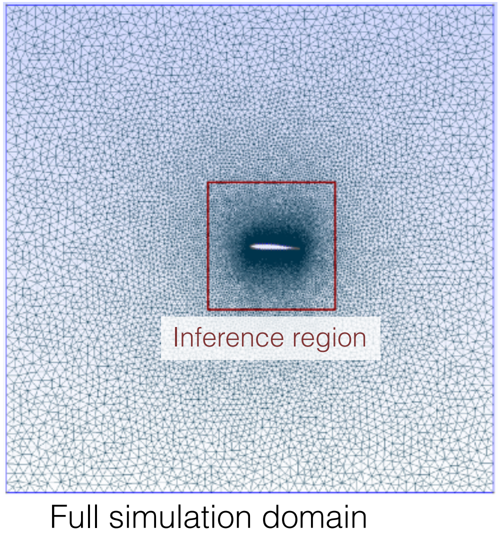
Solving PDEs with DL

- 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

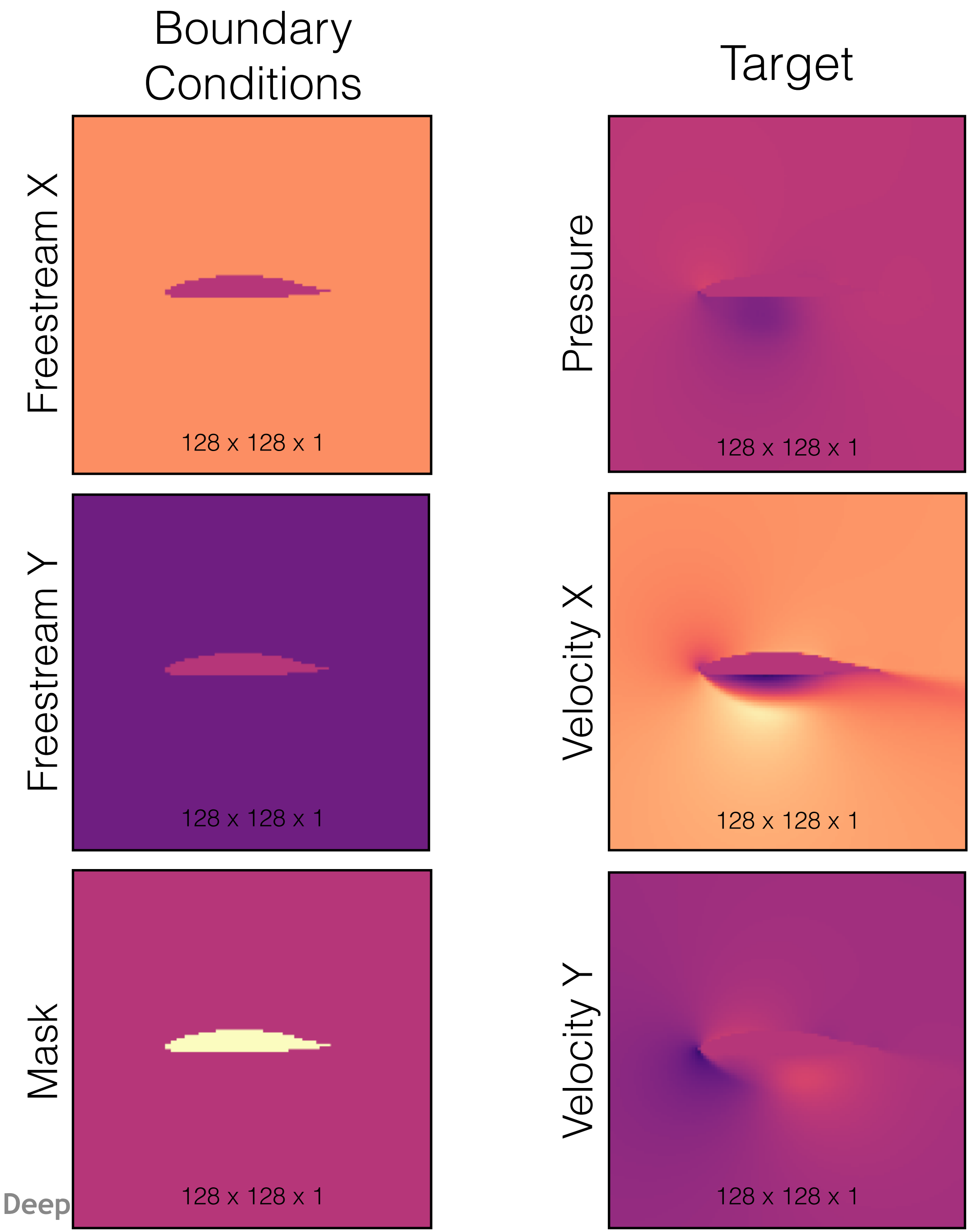
- Data generation
- Large number of pairs: input (BCs) - targets (solutions)





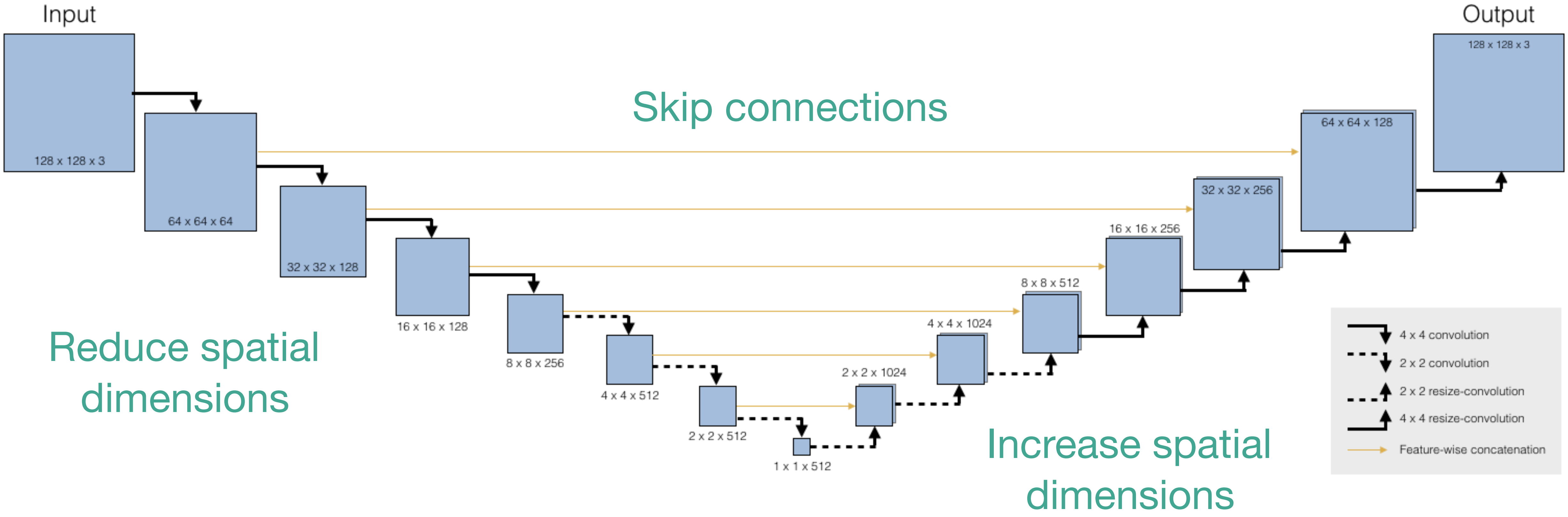
Solving PDEs with DL

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



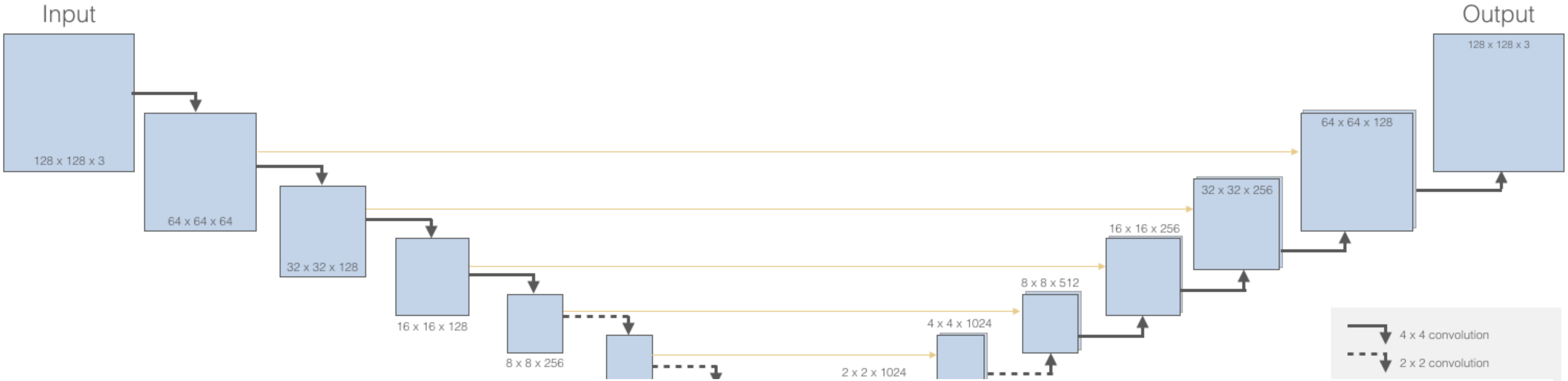
Solving PDEs with DL

- U-net NN architecture



Solving PDEs with DL

- U-net NN architecture



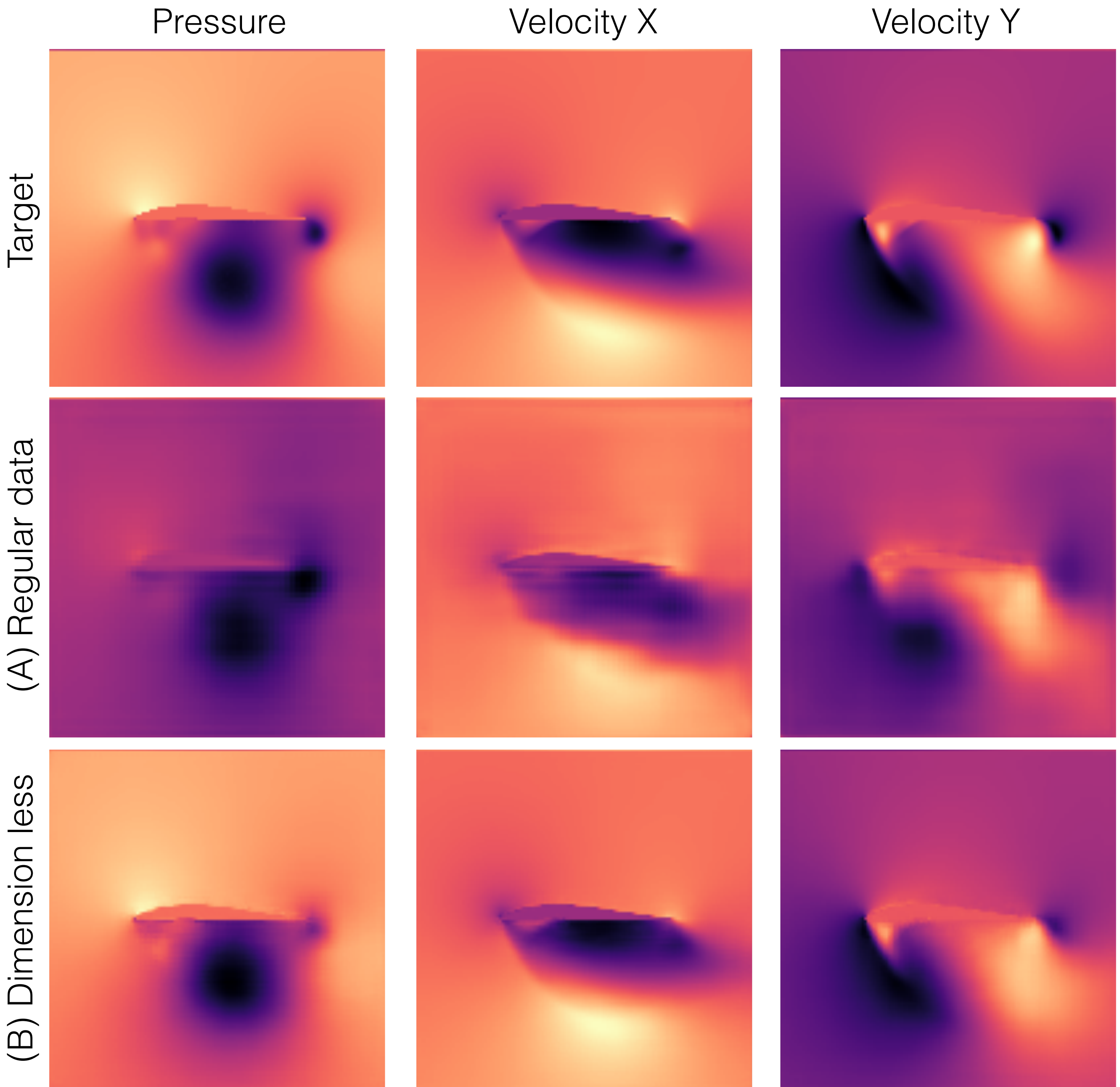
- Unet structure highly suitable for PDE solving
- Makes boundary condition information available throughout
- Crucial for inference of solution

Solving PDEs with DL

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

Results

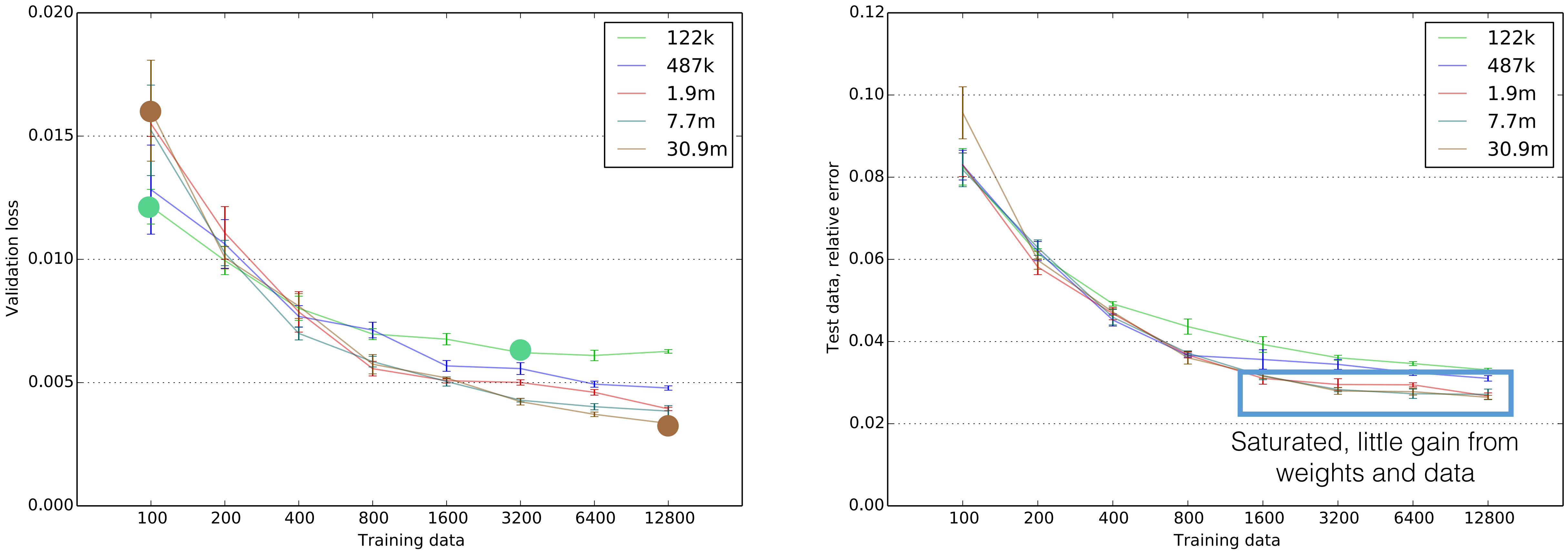
- Use knowledge about physics to simplify space of solutions: make quantities dimension- less
- Significant gains in inference accuracy



SIGGRAPH ASIA 2018 TOKYO

Solving PDEs with DL

- Validation and test accuracy for different model sizes



Code example

Solving PDEs with DL

Solving PDEs with DL

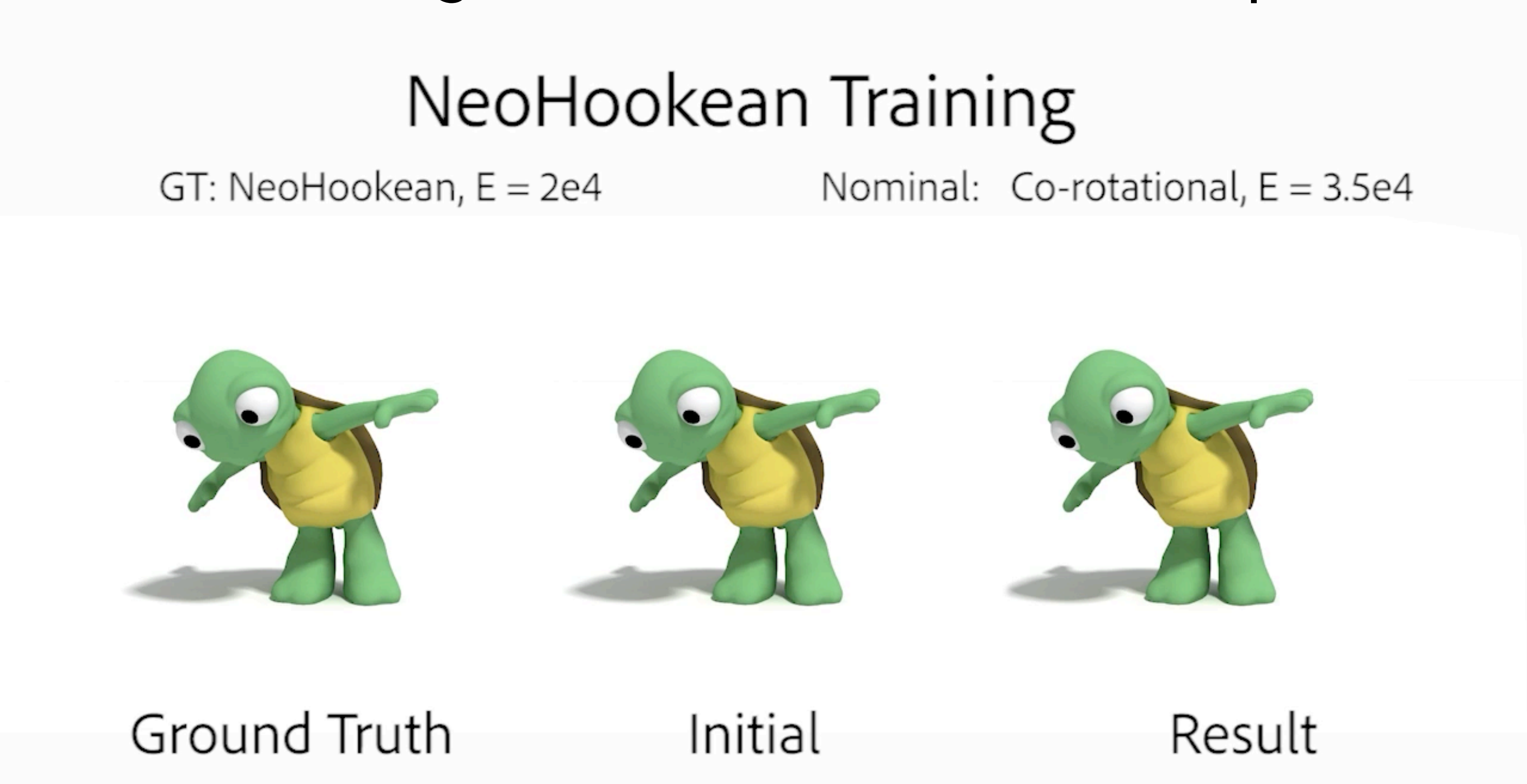
- Source code and training data available
- Requirements: numpy / pytorch , *OpenFOAM* for data generation
- Details at:
  https://github.com/thunil/Deep-Flow-Prediction and
  http://geometry.cs.ucl.ac.uk/creativeai/

Additional Examples

- Elasticity: material models
- Fluids: up-res algorithm & dimensionality reduction
- By no means exhaustive…

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]

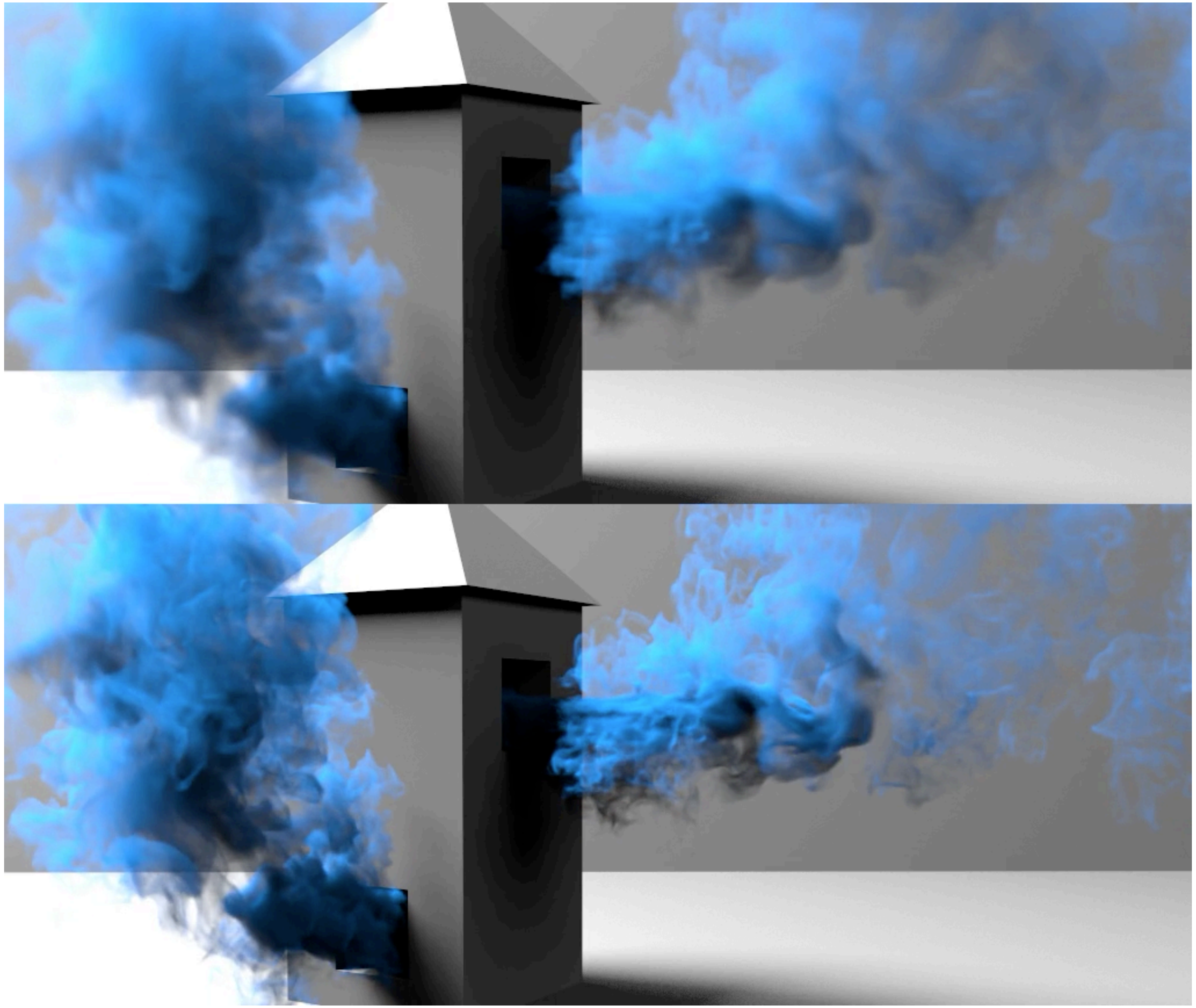
Neural Material - Elasticity

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

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

Temporal Data

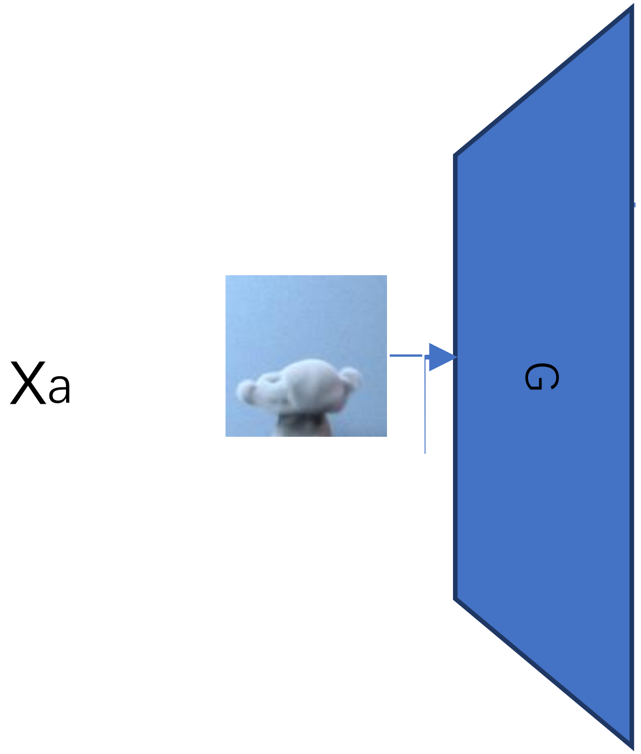
- tempoGAN: 3D GAN with temporal coherence



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

Temporal Data

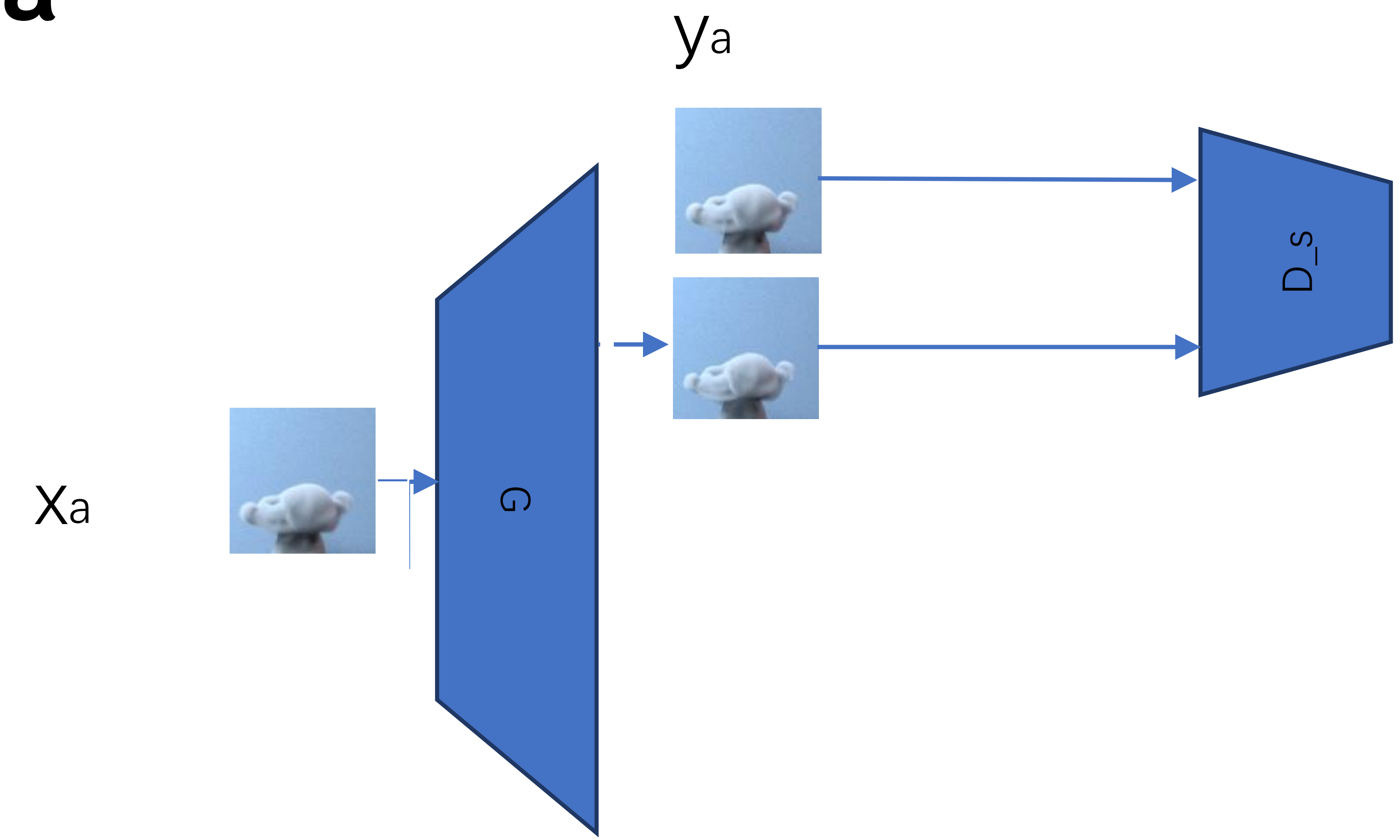
- tempoGAN: 3D GAN with temporal coherence



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

Temporal Data

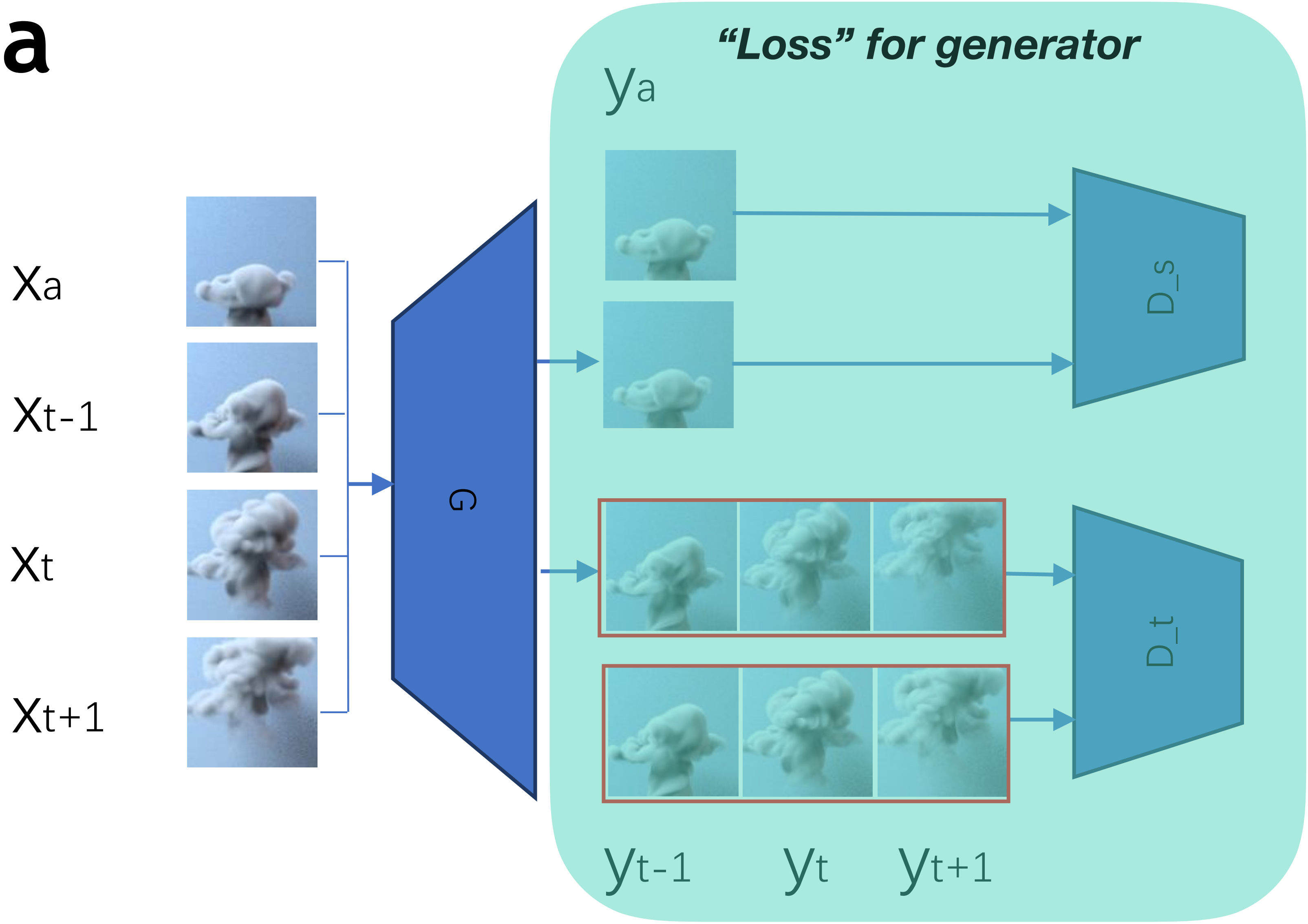
- tempoGAN: 3D GAN with temporal coherence



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

Temporal Data

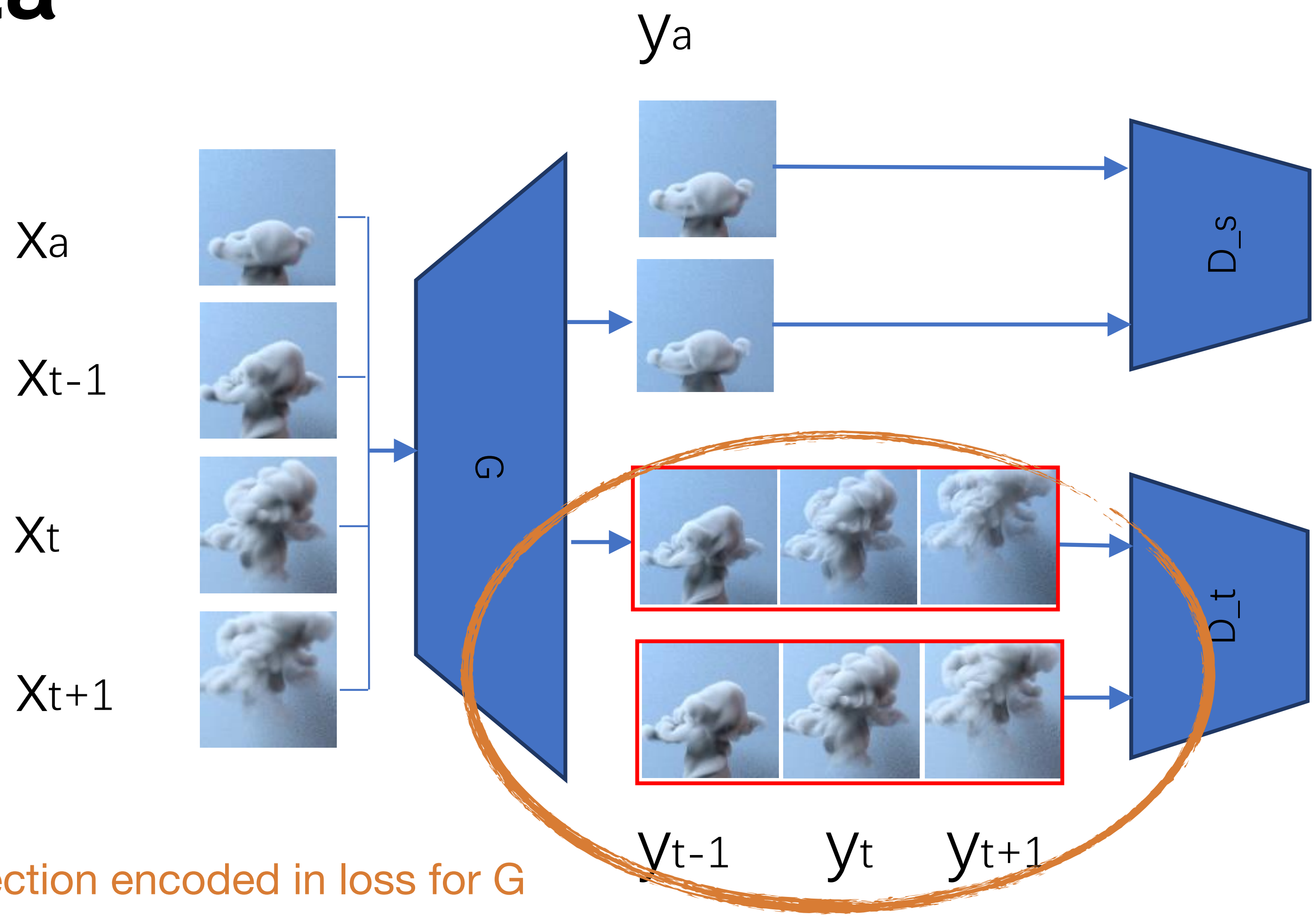
- tempoGAN: 3D GAN with temporal coherence



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

Temporal Data

- tempoGAN: 3D GAN with temporal coherence



Advection encoded in loss for G

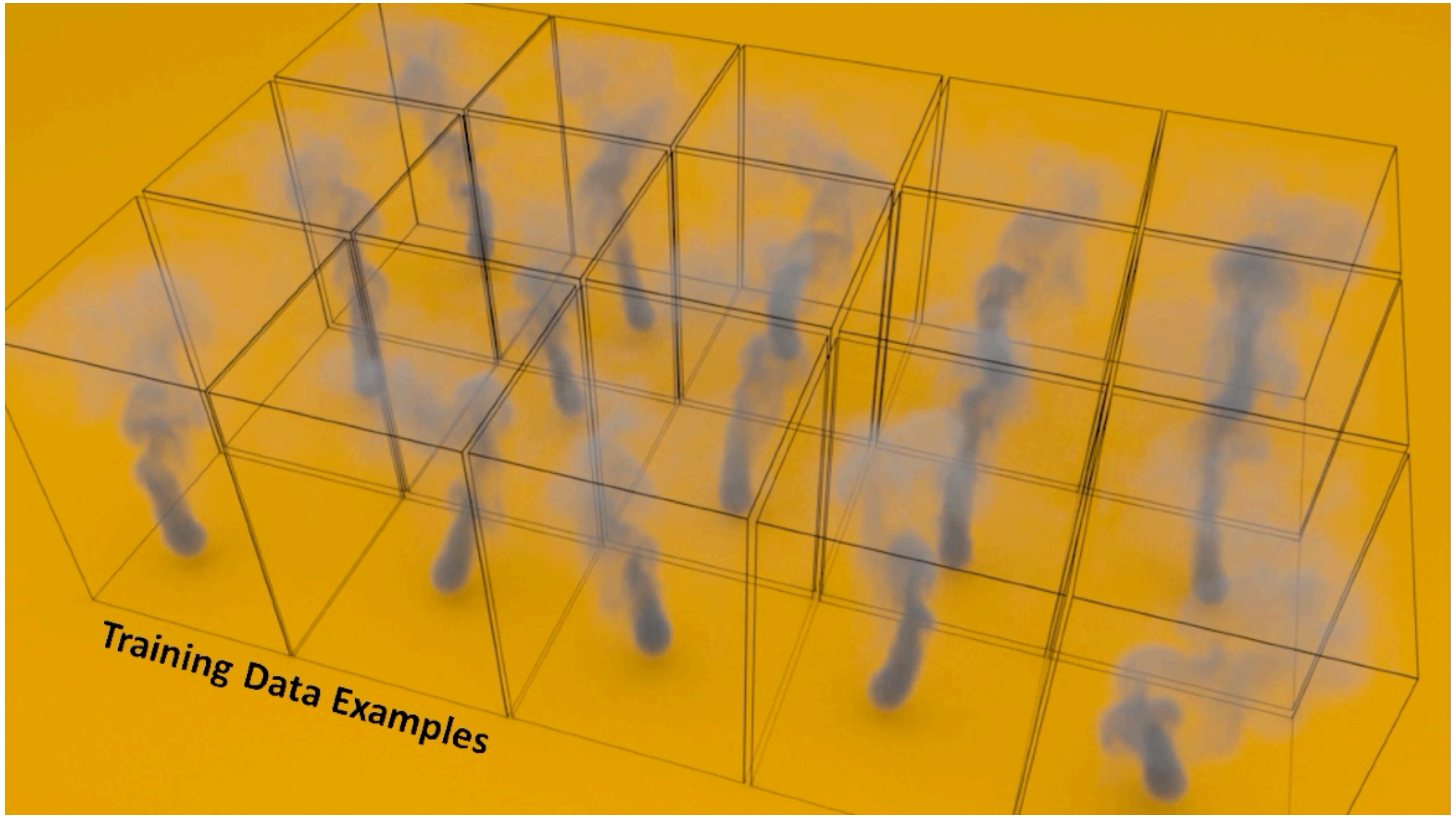
[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data

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

Latent Spaces

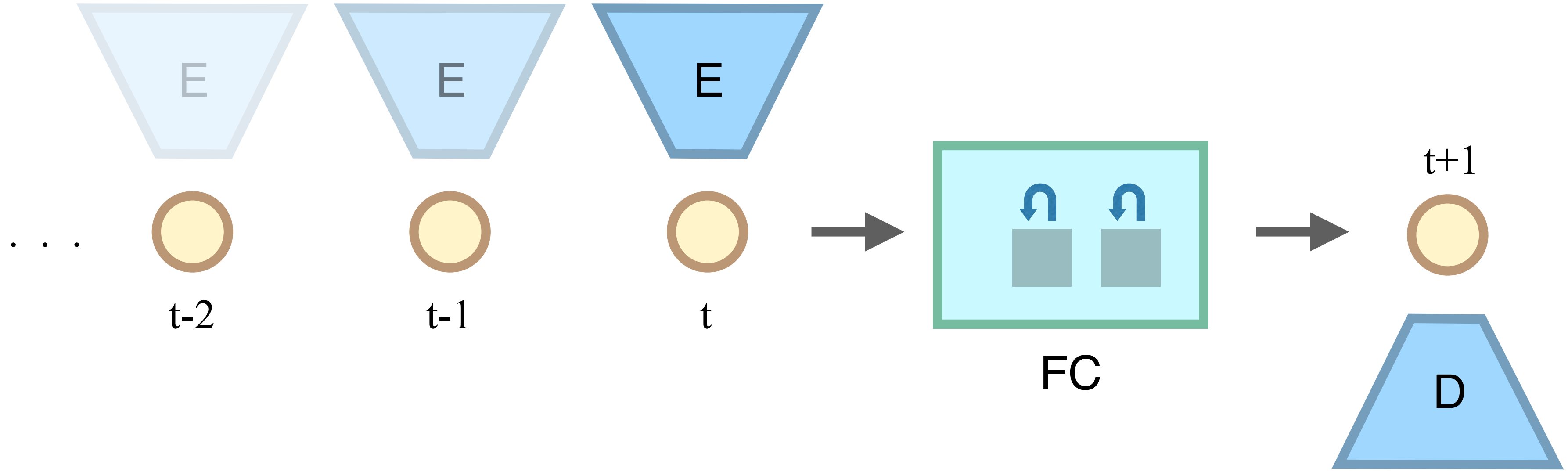
- Learn flexible reduced representation for physics problems



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, arXiv 2018]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, arXiv 2018]

Latent Spaces

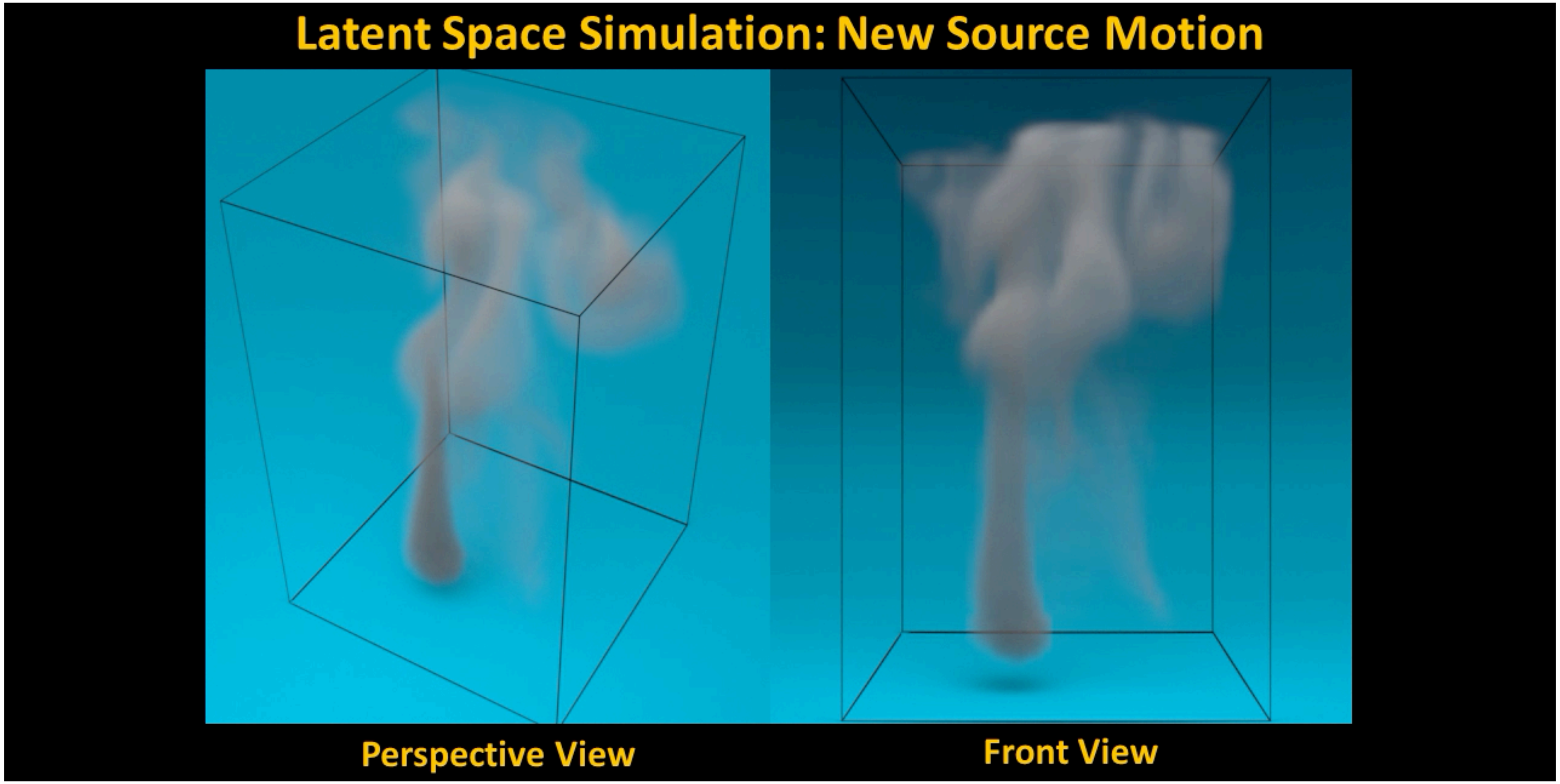
- Learn flexible reduced representation for physics problems
  - Employ Encoder part (E) of Autoencoder network to reduce dimensions
  - Predict future state in latent space with FC network
  - Use Decoder (D) of Autoencoder to retrieve volume data



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, arXiv 2018]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, arXiv 2018]

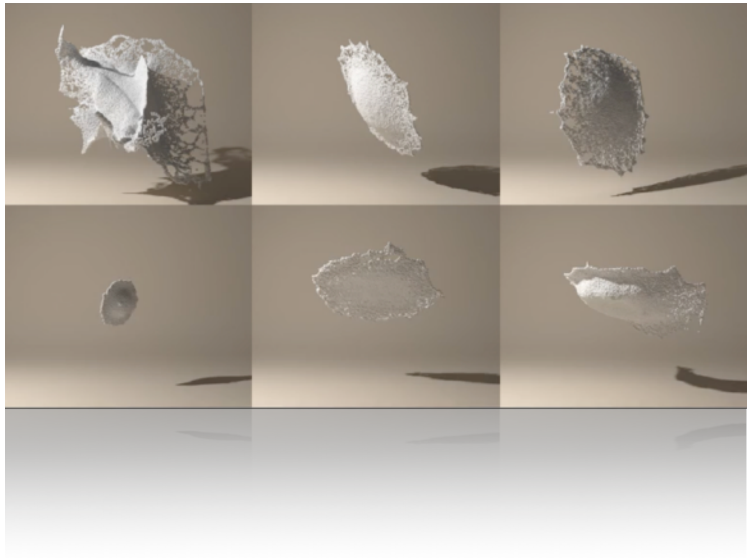
Latent Spaces

- Learn flexible reduced representation for physics problems



[Deep Fluids: A Generative Network for Parameterized Fluid Simulations, arXiv 2018]
[Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow, arXiv 2018]

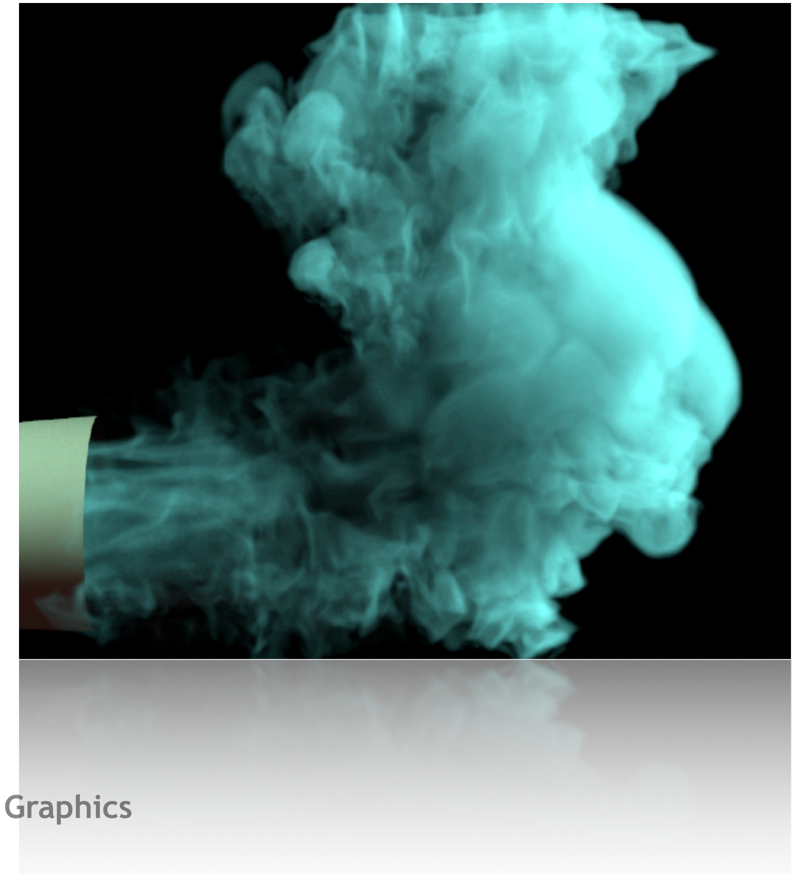
Summary



- 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 …



Thank you!

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