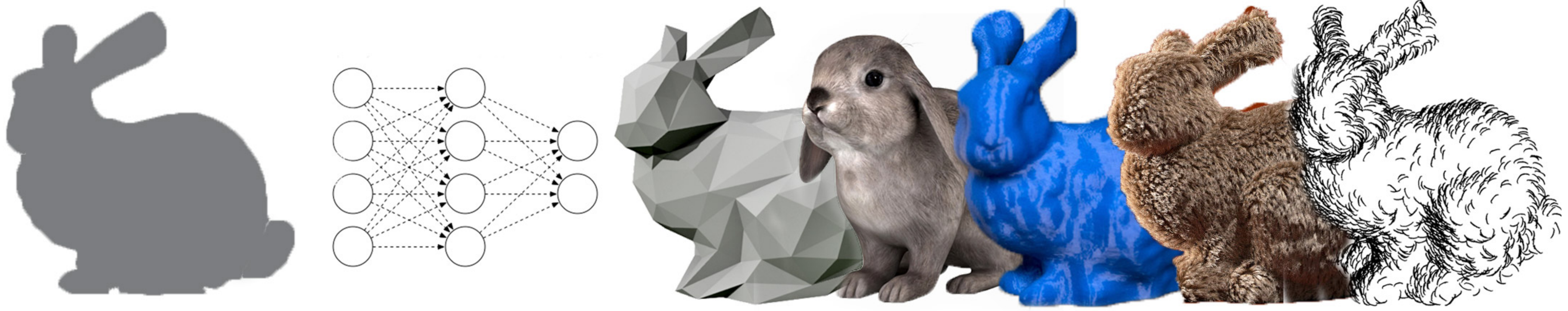


Physics and Animation

Niloy Mitra	Iasonas Kokkinos	Federico Monti	Emanuele Rodolà	Michael Bronstein	Or Litany	Leonidas Guibas
UCL	UCL	USI Lugano	La Sapienza	Imperial College USI Lugano	Stanford University Facebook	Stanford University

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

Timetable

			Niloy	Federico	Iasonas	Emanuele
Theory/Basics	Introduction	9:00	X	X	X	X
	Machine Learning Basics	~ 9:05	X			
	Neural Network Basics	~ 9:35		X		
	Alternatives to Direct Supervision (GANs)	~11:00			X	
State of the Art	Image Domain	~11:45			X	
	3D Domains (extrinsic)	~13:30	X			
	3D Domains (intrinsic)	~ 14:15				X
	Physics and Animation	~ 16:00	X			
	Discussion	~ 16:45	X	X	X	X

Sessions: A. 9:00-10:30 (**coffee**) B. 11:00-12:30 [**LUNCH**] C. 13:30-15:00 (**coffee**) D. 15:30-17:00

NN Cheatsheet

NN Cheatsheet

- Data, data, data

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible
- Overfit, ‘reproducible’, backprop (check if possible)

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible
- Overfit, ‘reproducible’, backprop (check if possible)
- ADAM

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible
- Overfit, ‘reproducible’, backprop (check if possible)
- ADAM
- Change one block/concept at a time

NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible
- Overfit, ‘reproducible’, backprop (check if possible)
- ADAM
- Change one block/concept at a time
- Regularize
  (e.g., latent representation, Spectral basis, image formation module)

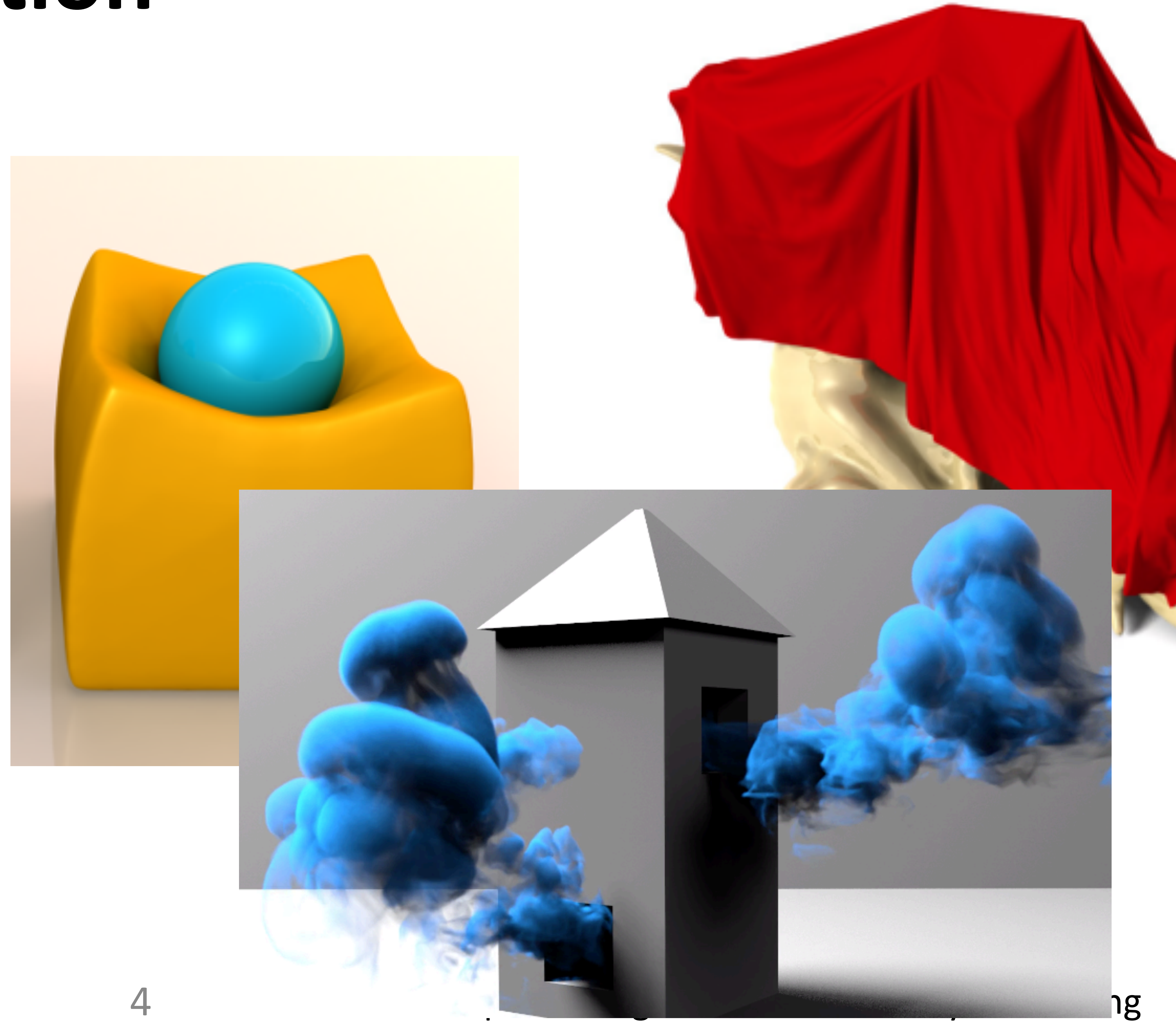
NN Cheatsheet

- Data, data, data
- Setup evaluation, benchmark, loss measures, baselines
- Initialize well, visualize intermediate results
- Use existing networks to start, if possible
- Overfit, ‘reproducible’, backprop (check if possible)
- ADAM
- Change one block/concept at a time
- Regularize
  (e.g., latent representation, Spectral basis, image formation module)
- Hyperparameter optimization

Physics-Based Animation

[Many of the following slides thanks to Nils Thuerey]

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



Physics-Based Animation

[Many of the following slides thanks to Nils Thuerey]

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

Physics-Based Animation

- Traditional approach:

Experiment

Theory

Computation

Physics-Based Animation

- Traditional approach:

Experiment

Theory

Computation

Observations / data

Physics-Based Animation

- Traditional approach:

Experiment

Observations / data

Theory

Model equations

Computation

Physics-Based Animation

- Traditional approach:

Experiment	Theory	Computation
Observations / data	Model equations	Discrete representation

Physics-Based Animation

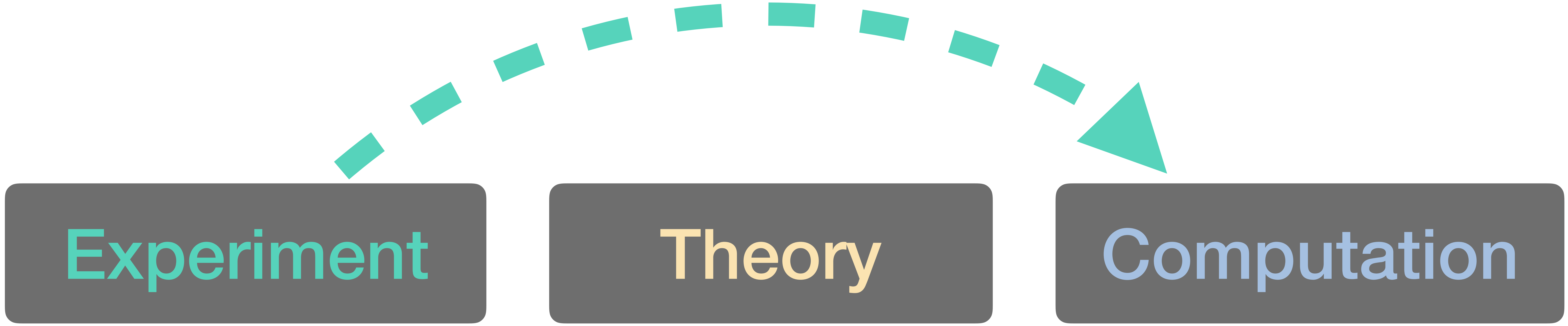
- Traditional approach:

Experiment	Theory	Computation
Observations / data	Model equations	Discrete representation

Physics-Based Animation

- Traditional approach:

Skip Theory with Deep Learning?



Observations / data

Model equations

Discrete representation

Physics-Based Animation

- Traditional approach:

Skip Theory with Deep Learning?

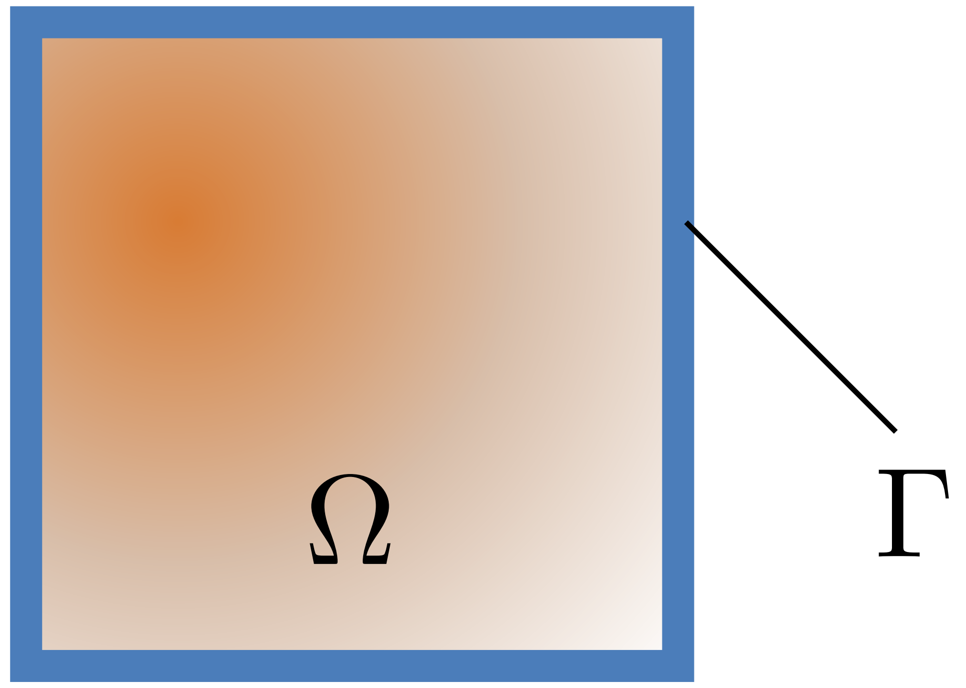
[No! More on that later…]

Experiment	Theory	Computation
Observations / data	Model equations	Discrete representation

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

Learing to Represent Mechanics

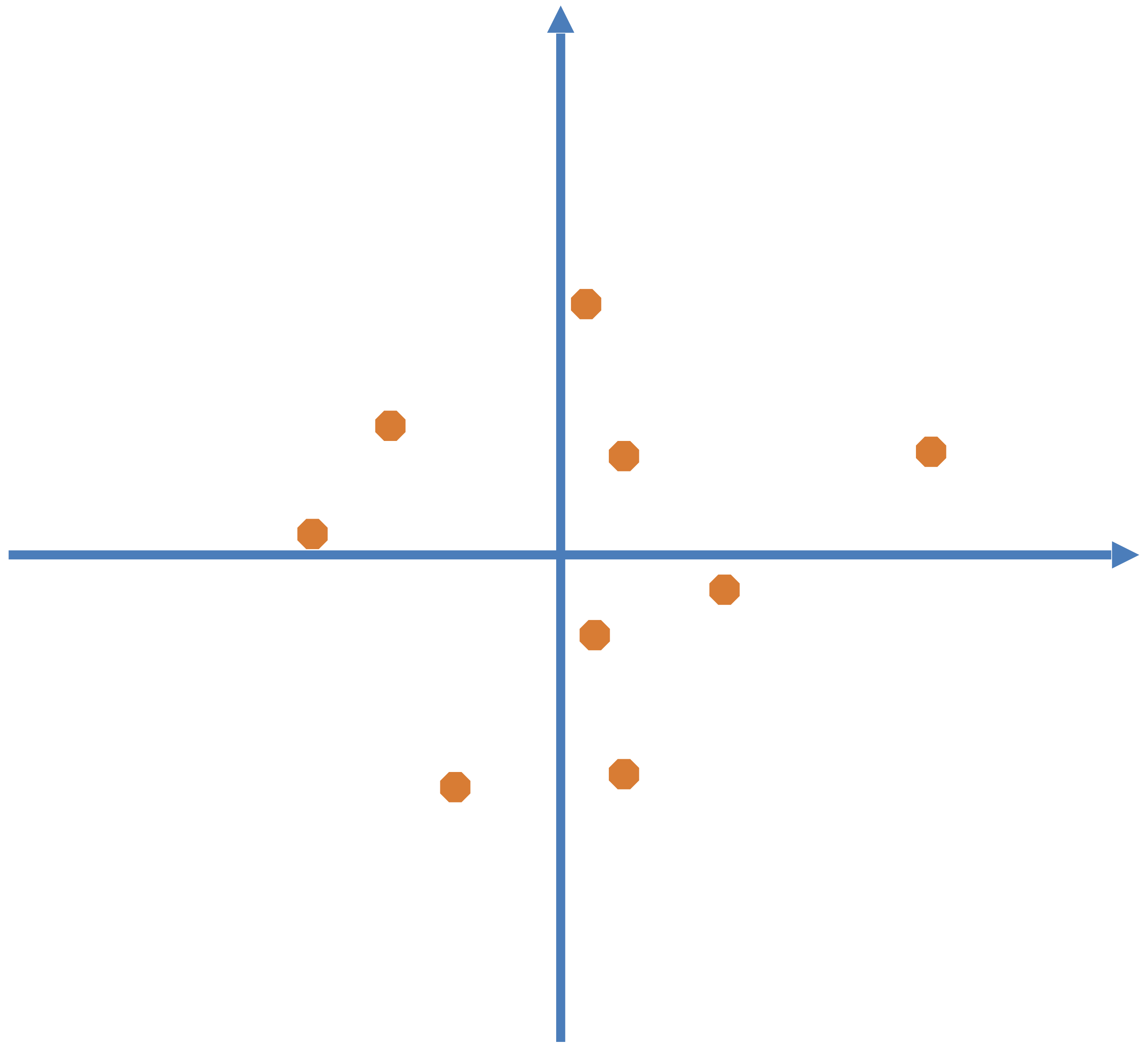
Learing to Represent Mechanics

Methodology 2

- Consider dimensionality & structure of discretization

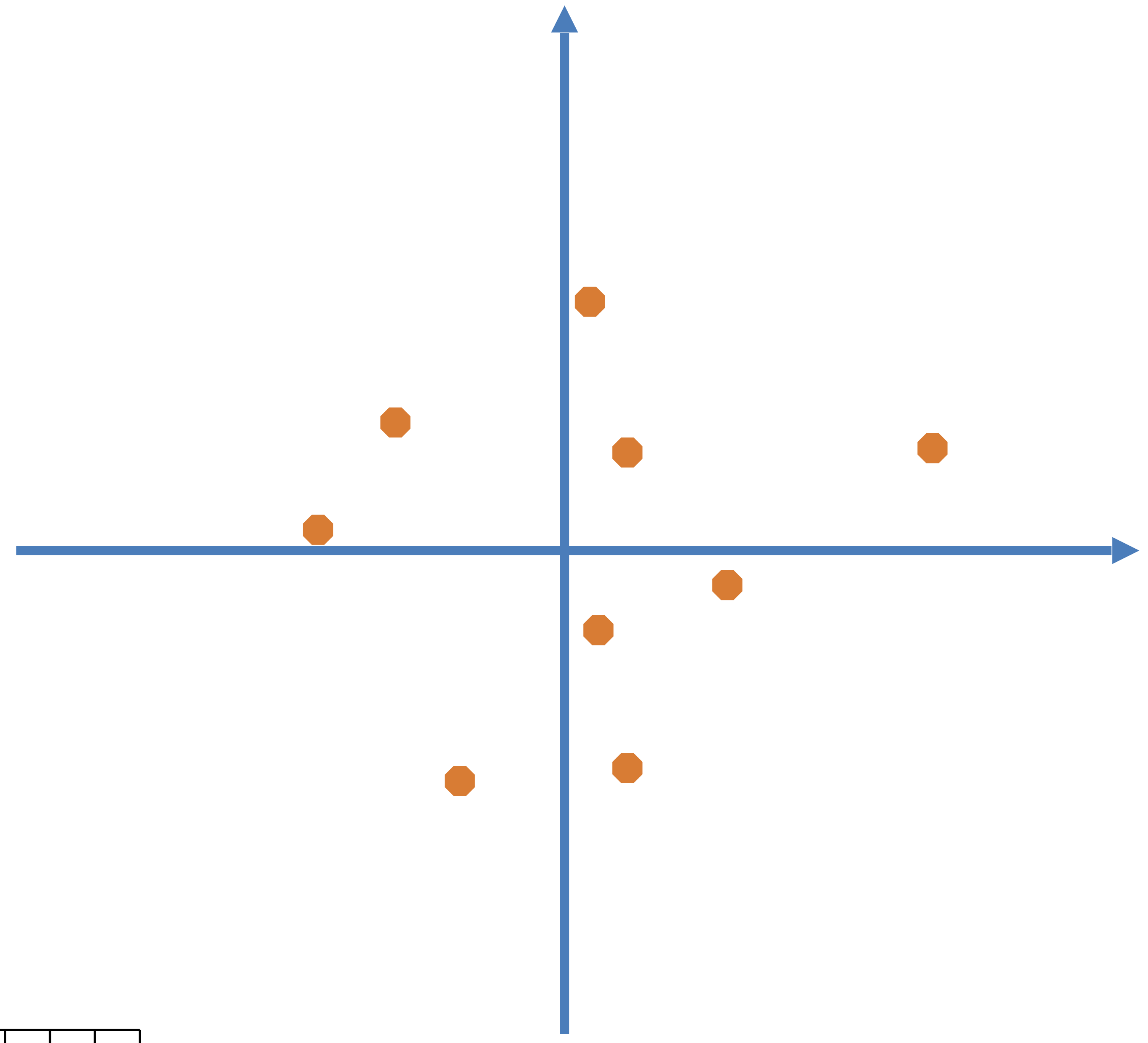
Methodology 2

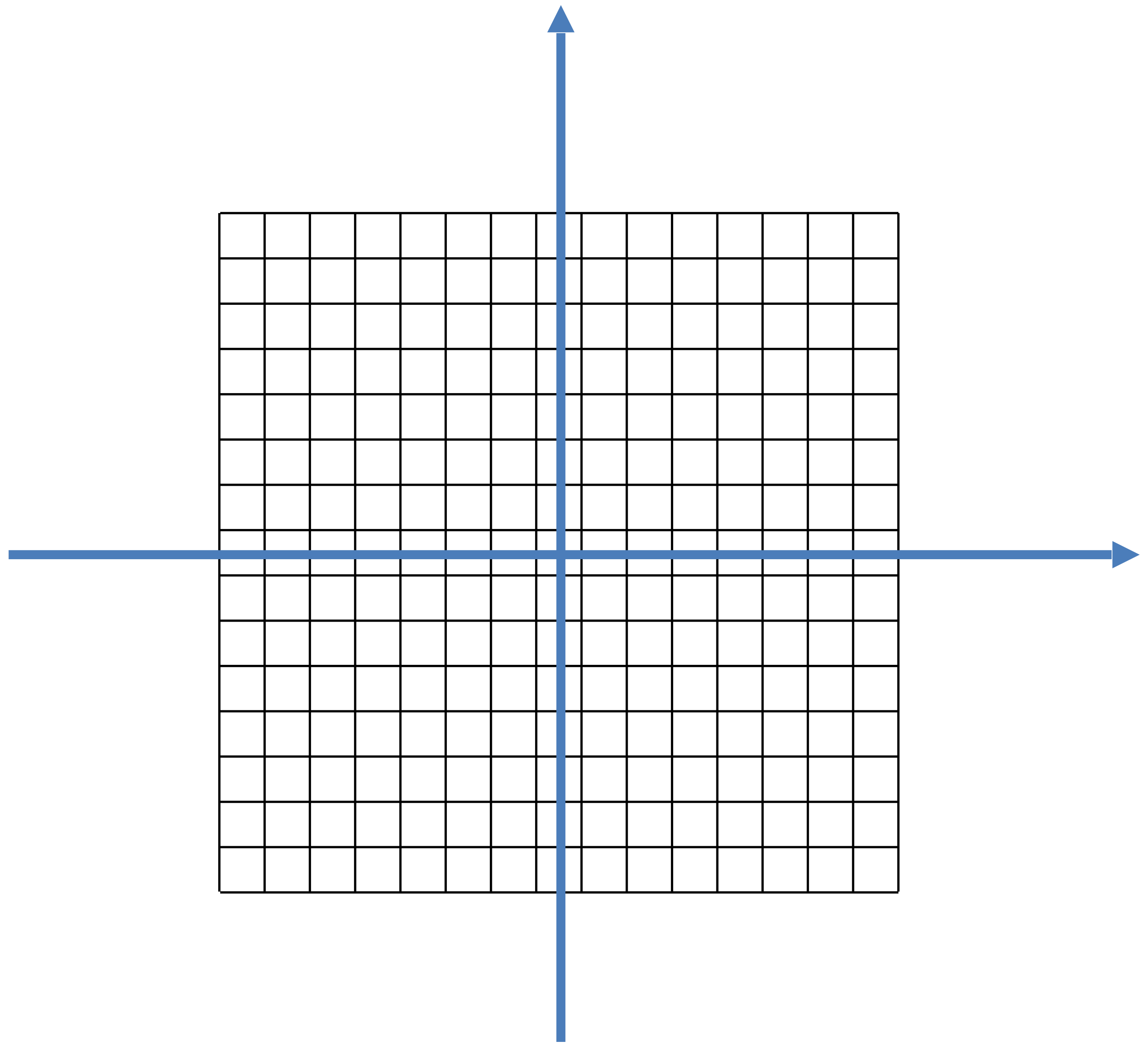
- Consider dimensionality & structure of discretization
- Small & unstructured
  - Fully connected NNs only choice
  - Only if necessary…



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



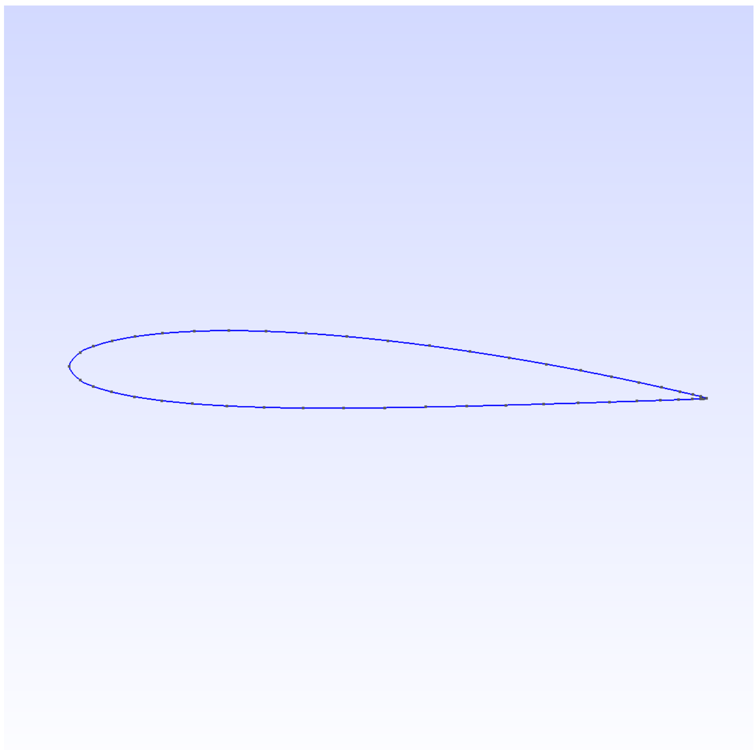


Solving PDEs with DL

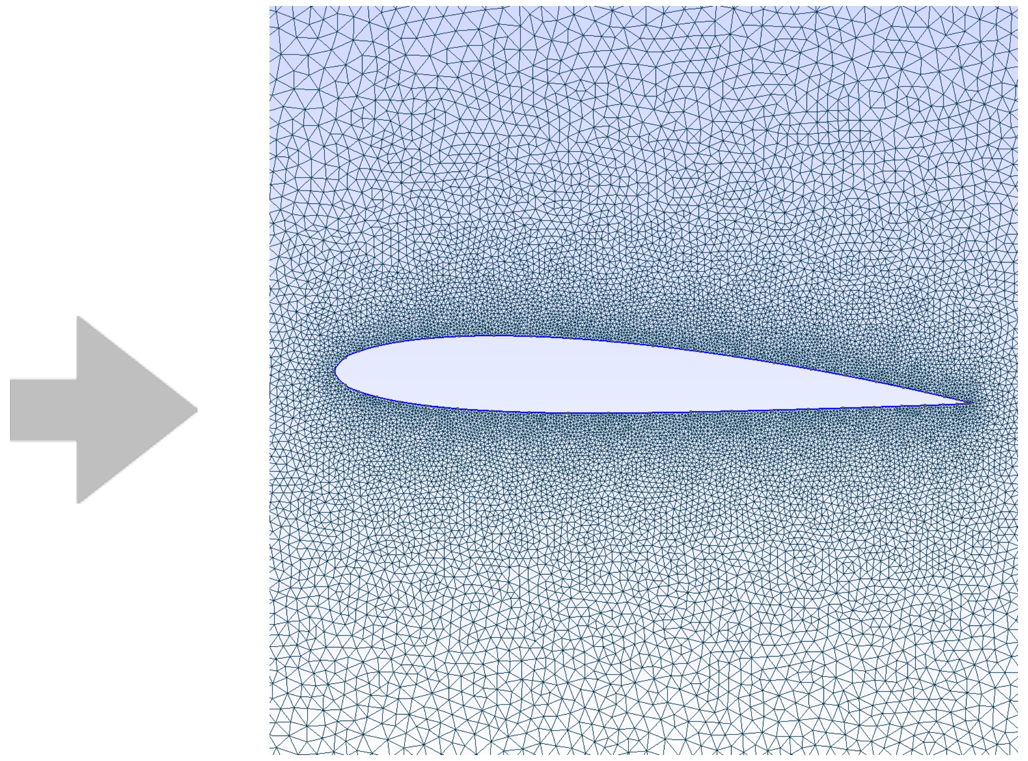
- Practical example: *airfoil flow*
  - Given boundary conditions solve stationary flow problem on grid
  - Fully replace traditional solver
  - 2D data, no time dimension
  - I.e., holistic approach with structured data

Solving PDEs with DL

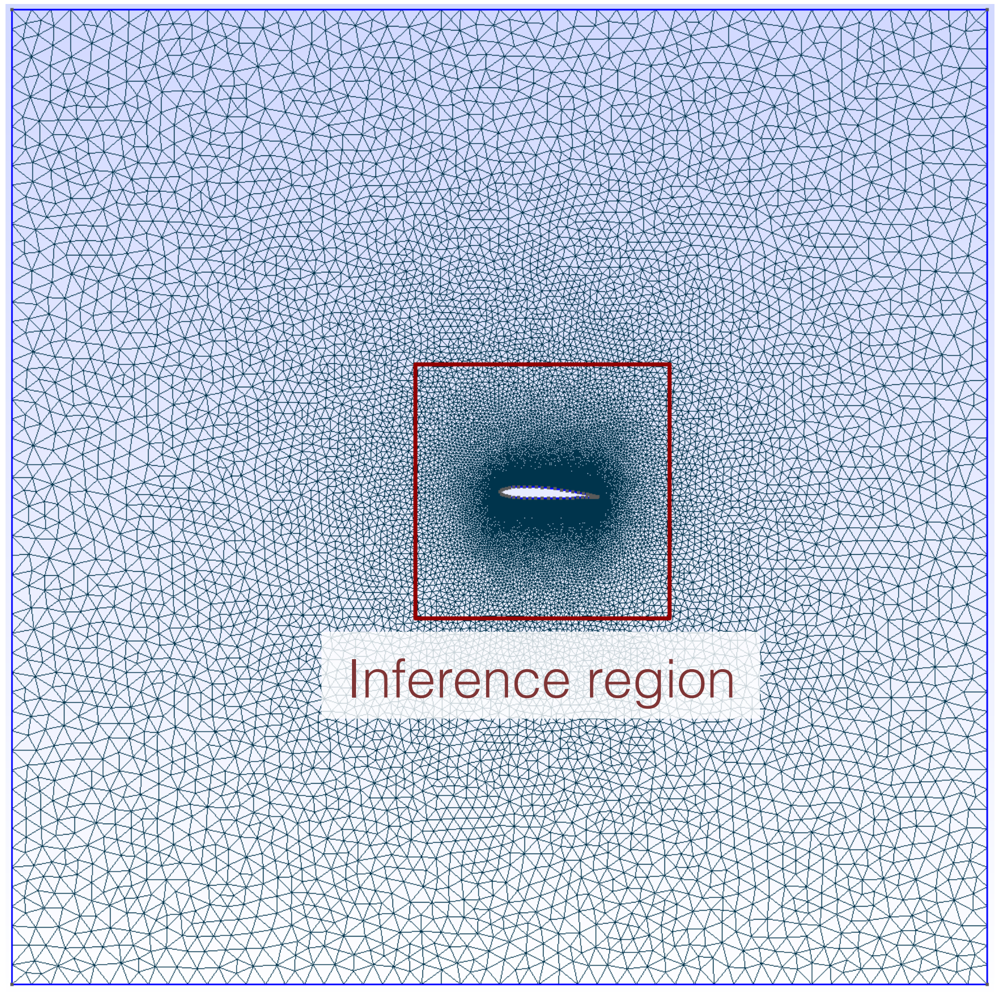
- Data generation
- Large number of pairs: input (BCs) - targets (solutions)



Airfoil profile



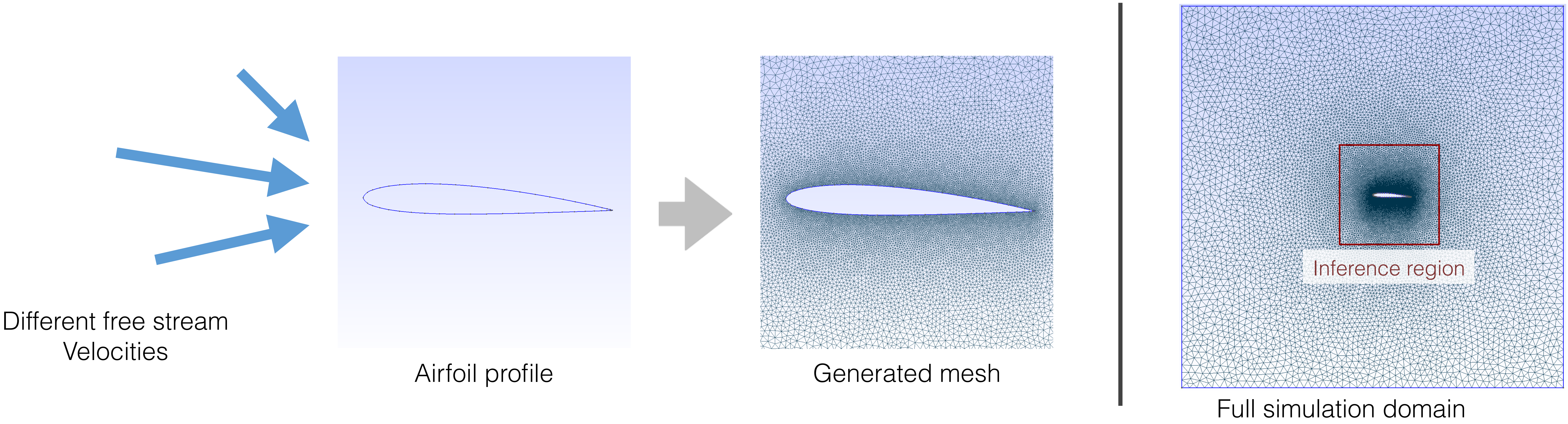
Generated mesh



Full simulation domain

Solving PDEs with DL

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

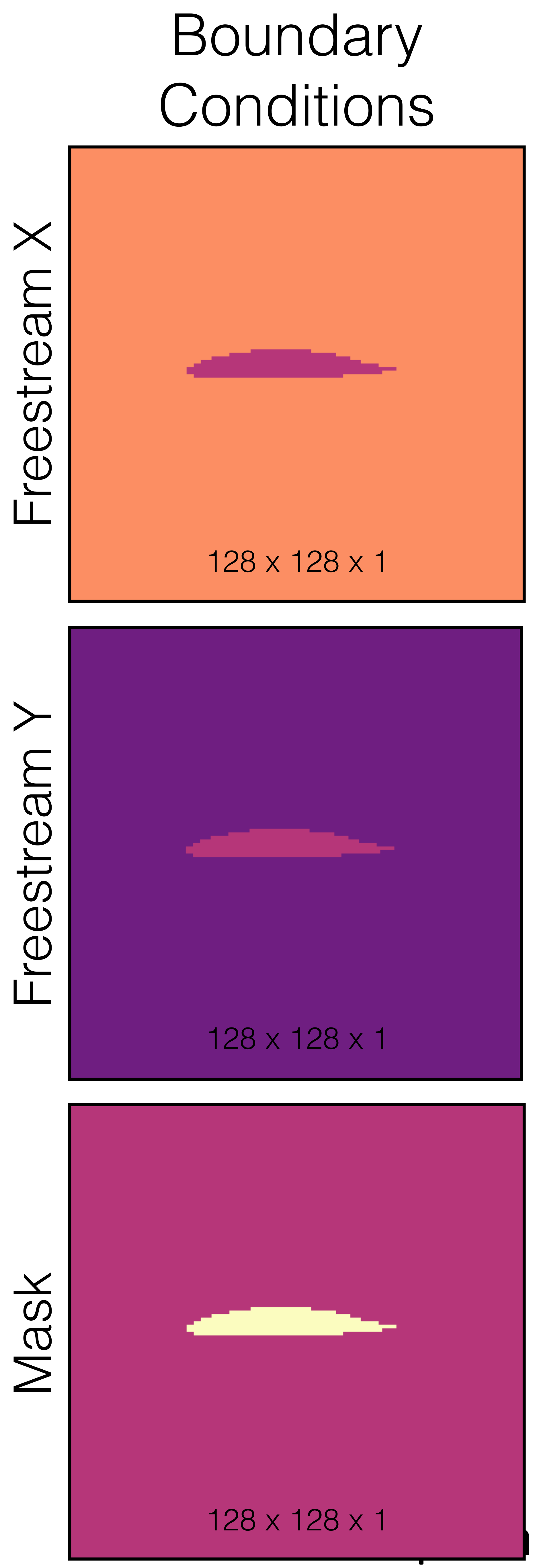


Solving PDEs with DL

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

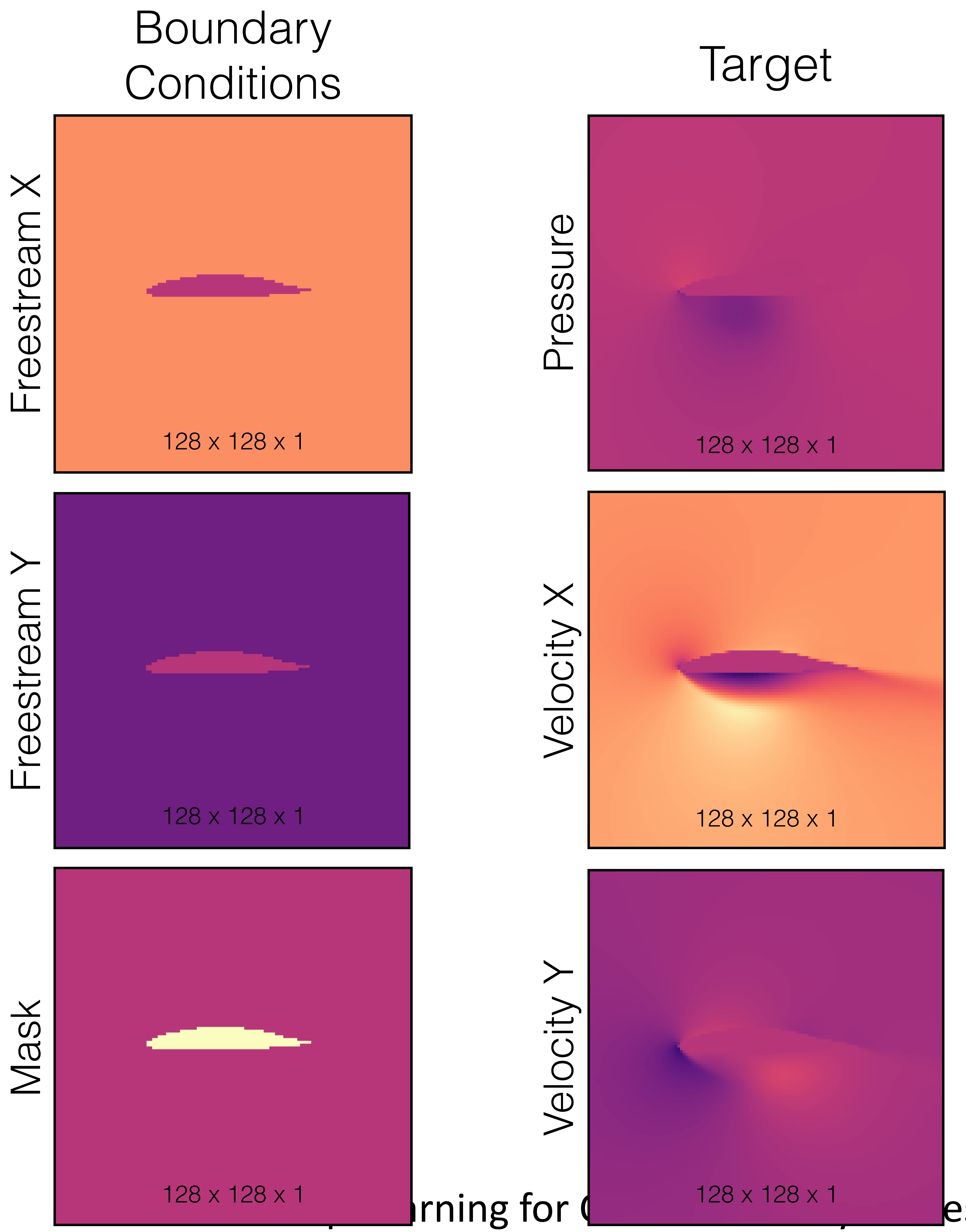
Solving PDEs with DL

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



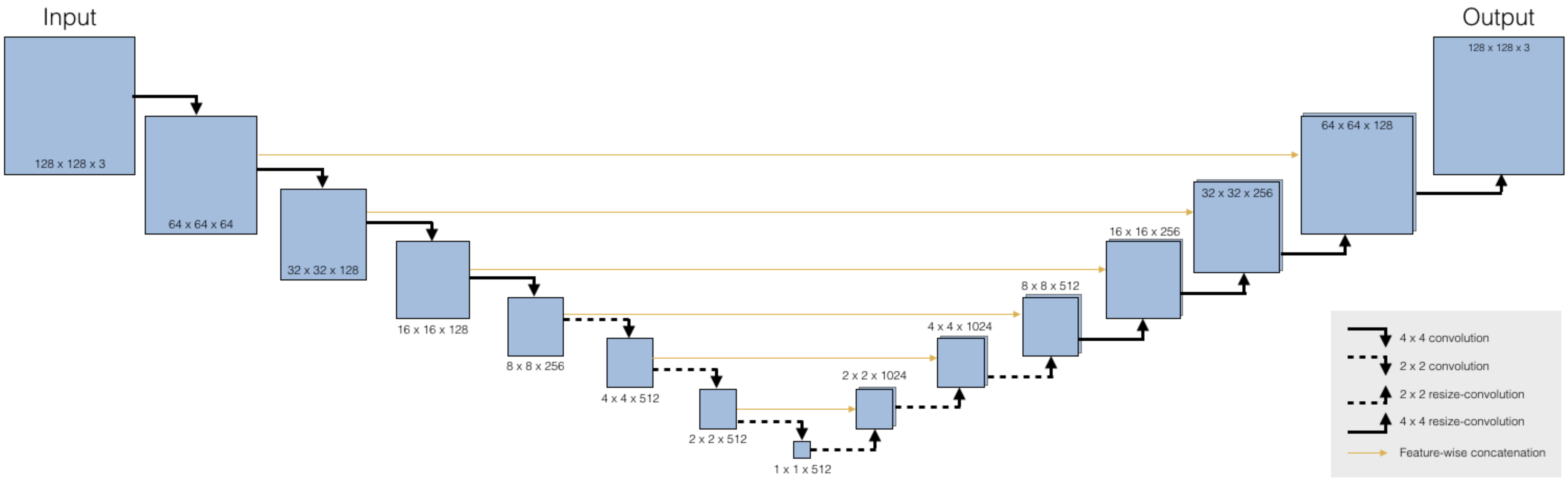
Solving PDEs with DL

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



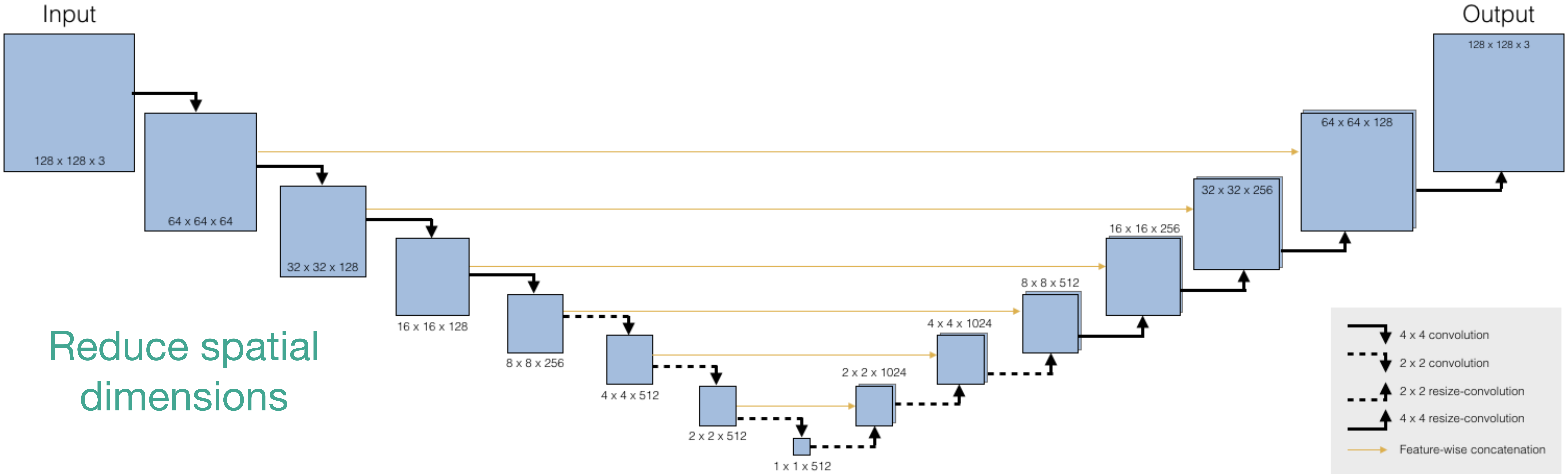
Solving PDEs with DL

- U-net NN architecture



Solving PDEs with DL

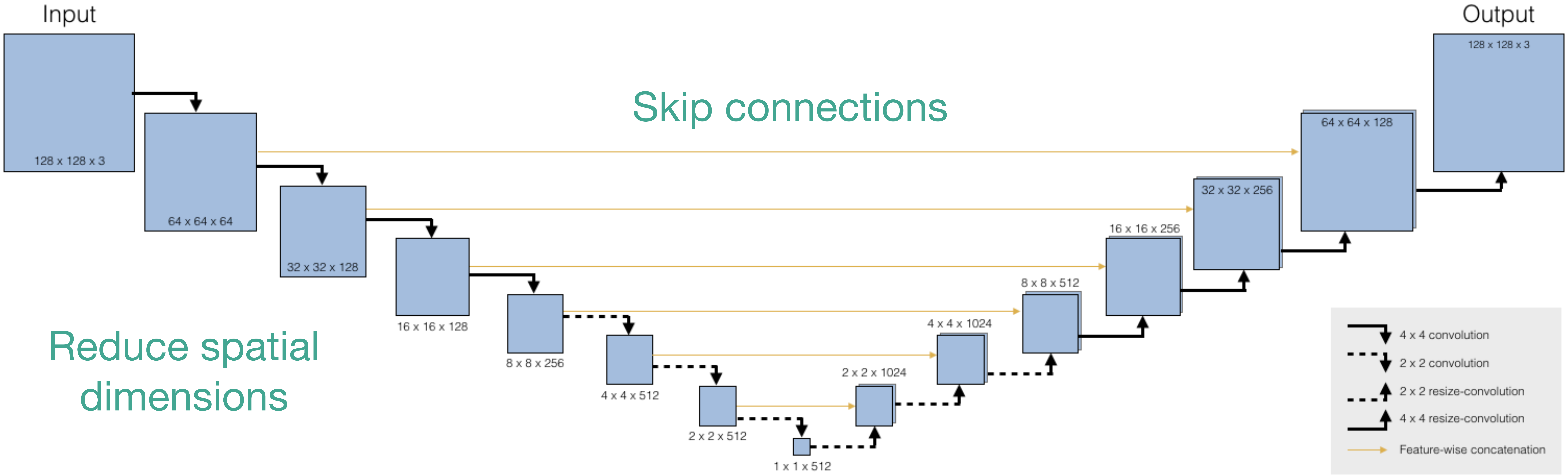
- U-net NN architecture



Reduce spatial dimensions

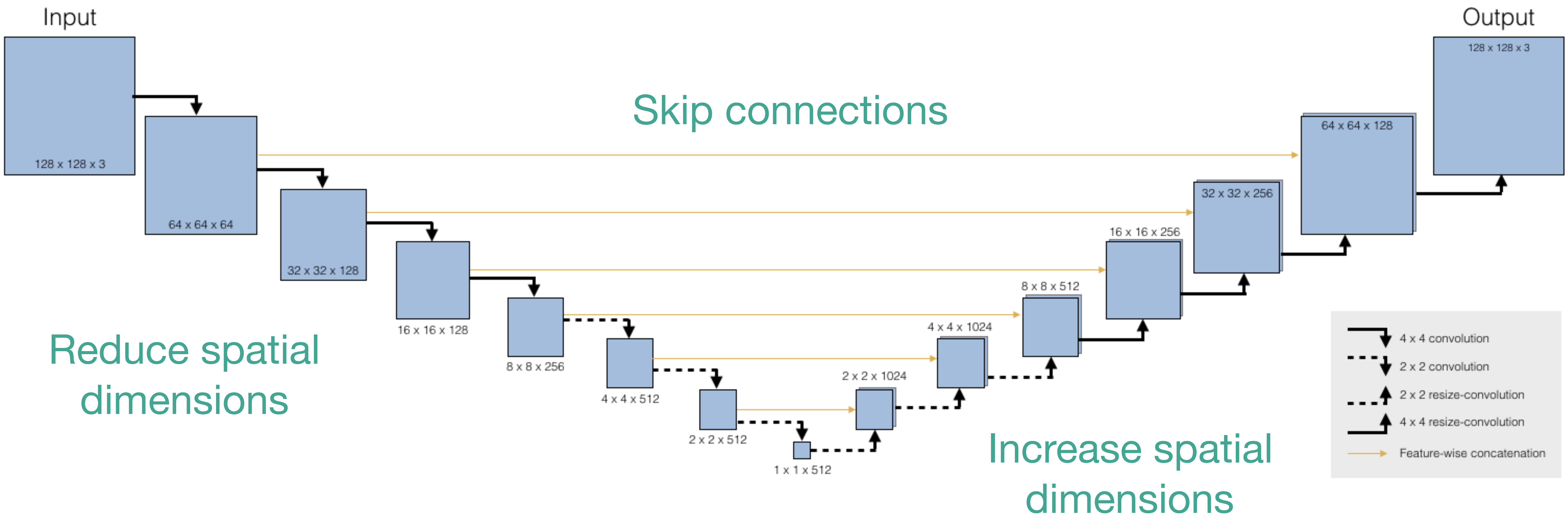
Solving PDEs with DL

- U-net NN architecture

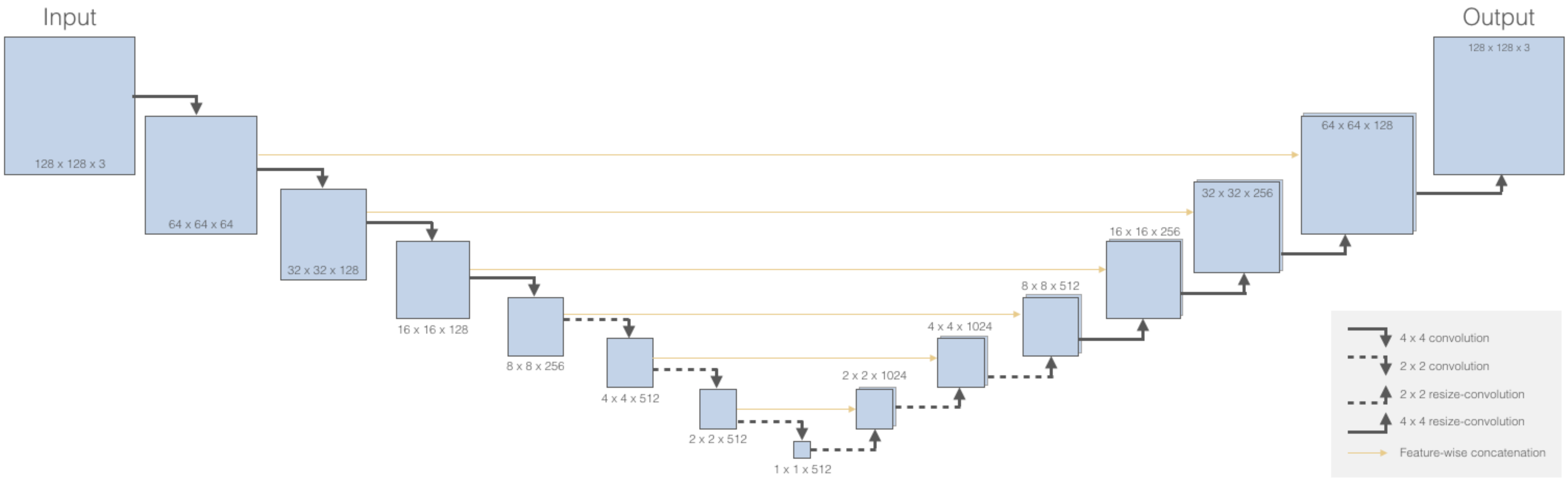


Solving PDEs with DL

- U-net NN architecture



Solving PDEs with DL



Solving PDEs with DL

- Unet structure highly suitable for PDE solving

Solving PDEs with DL

- Unet structure highly suitable for PDE solving
- Makes boundary condition information available throughout

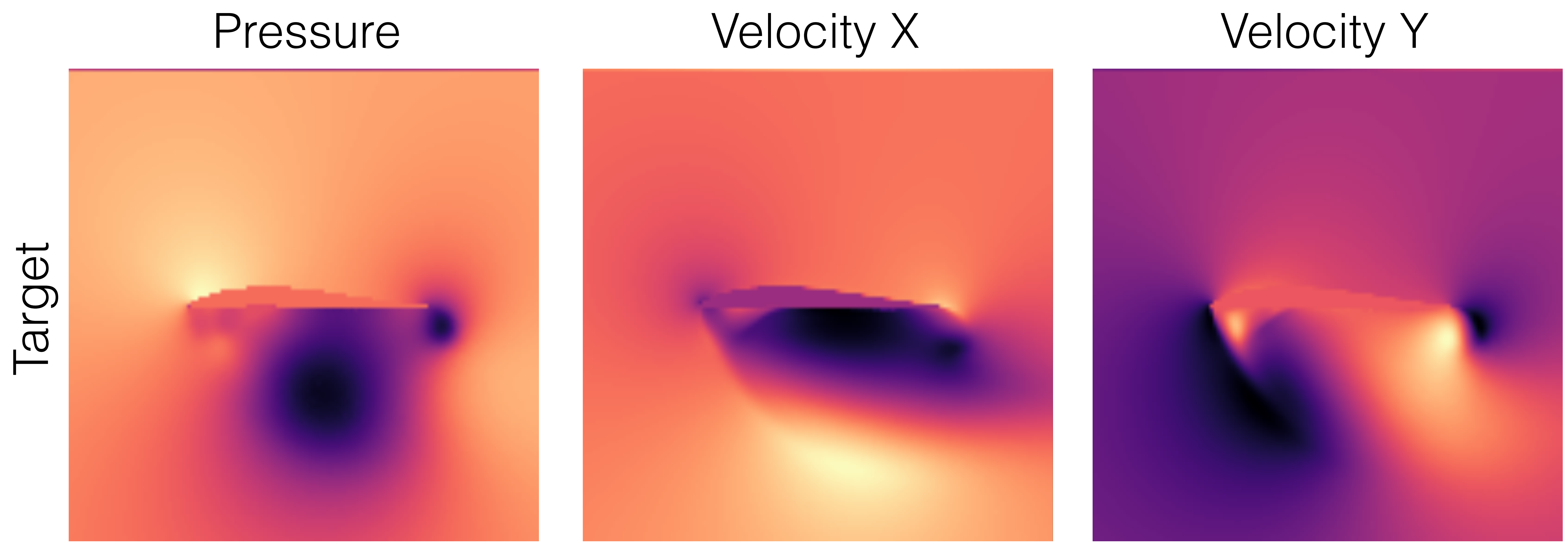
Solving PDEs with DL

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

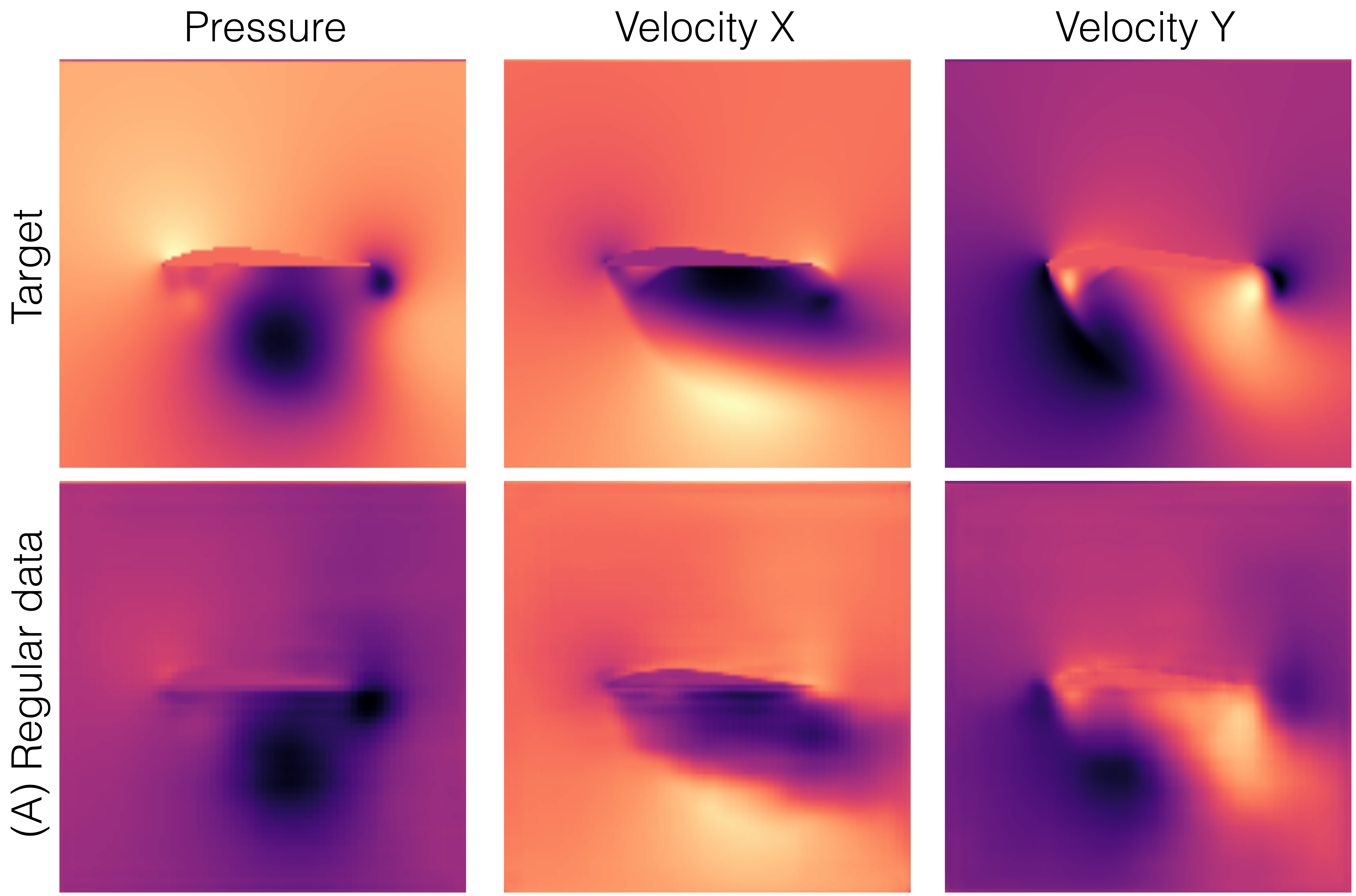
Solving PDEs with DL

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

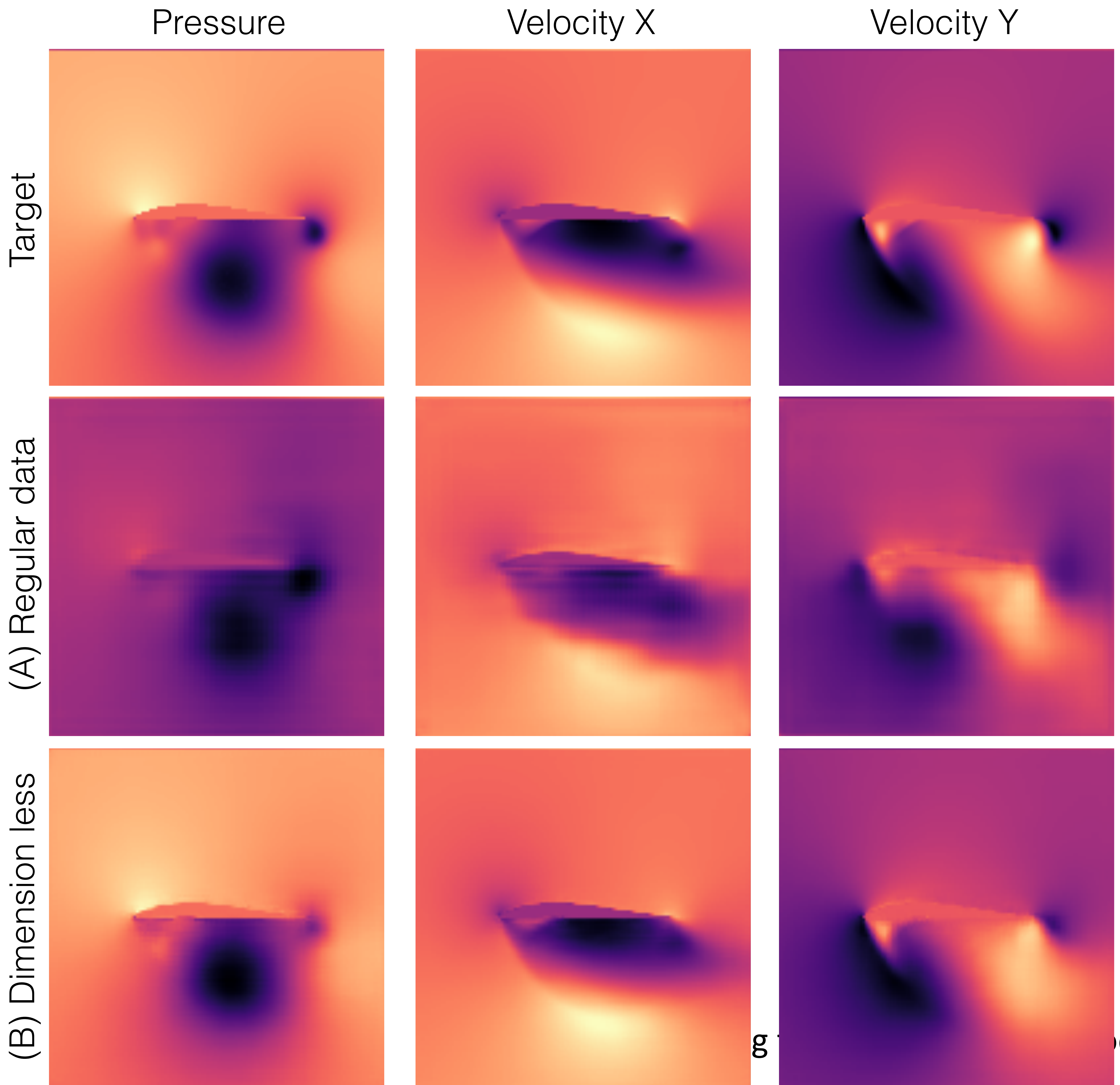
Results



Results



Results

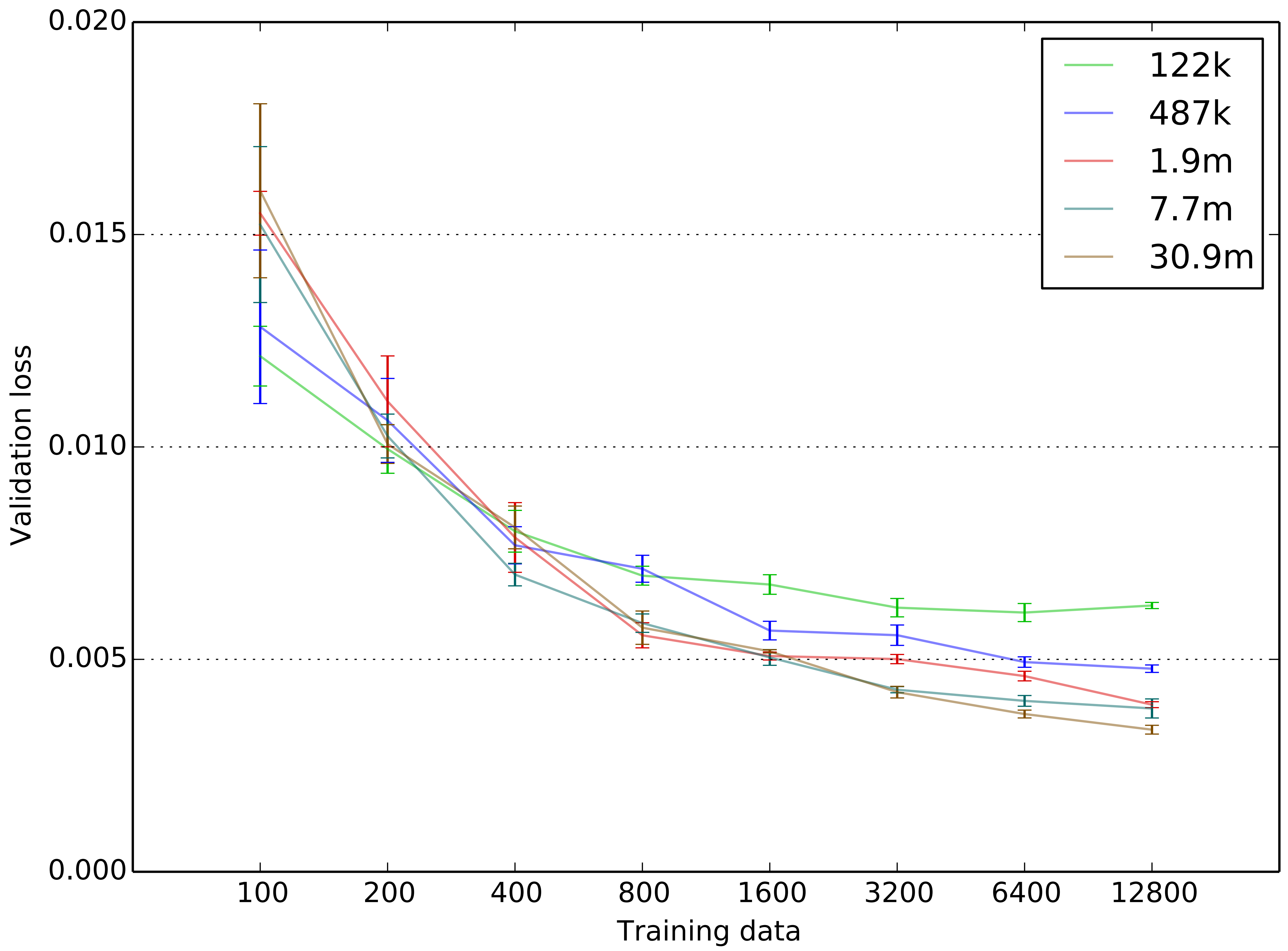


g

cessing

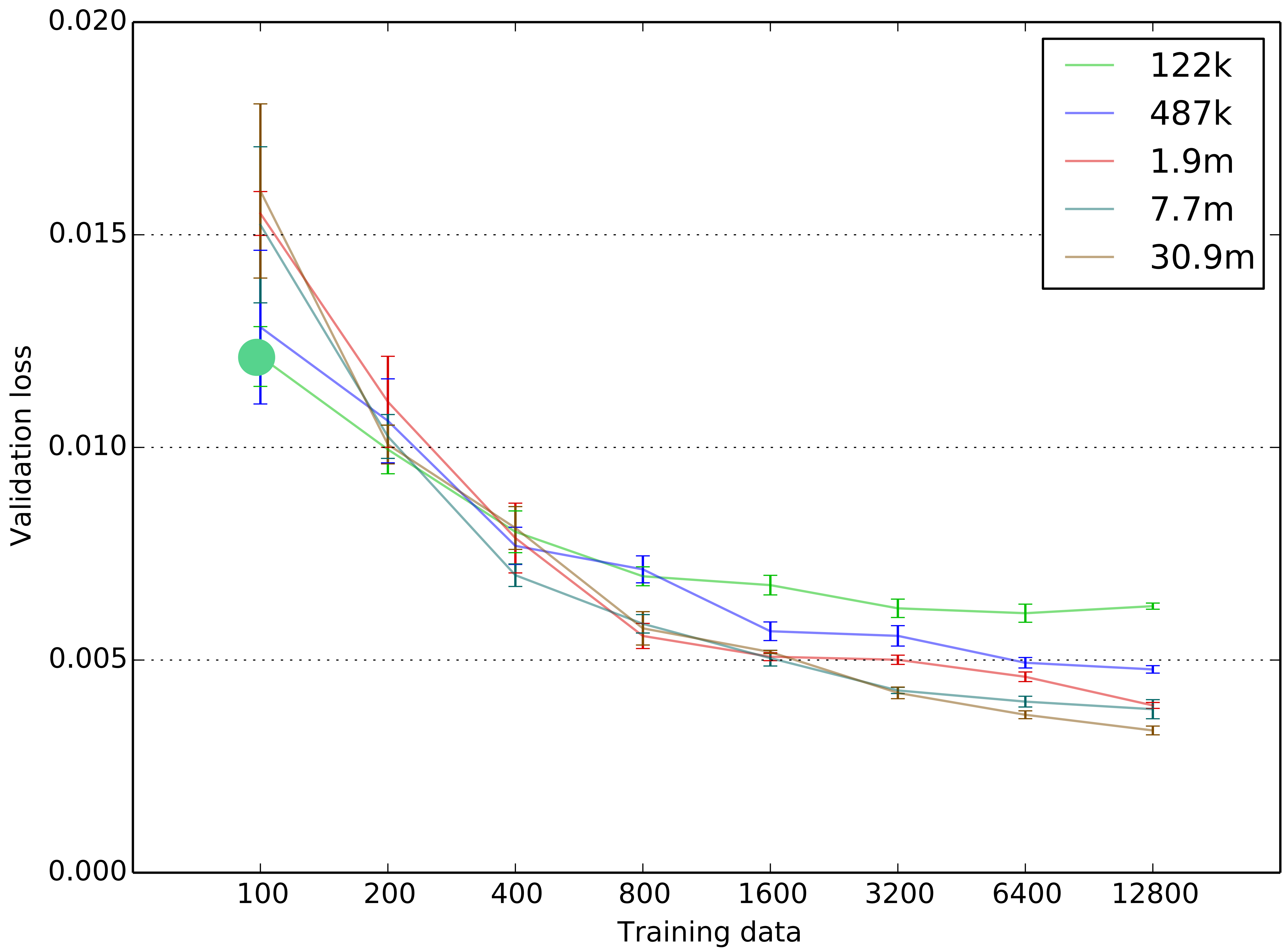
Solving PDEs with DL

- Validation and test accuracy for different model sizes



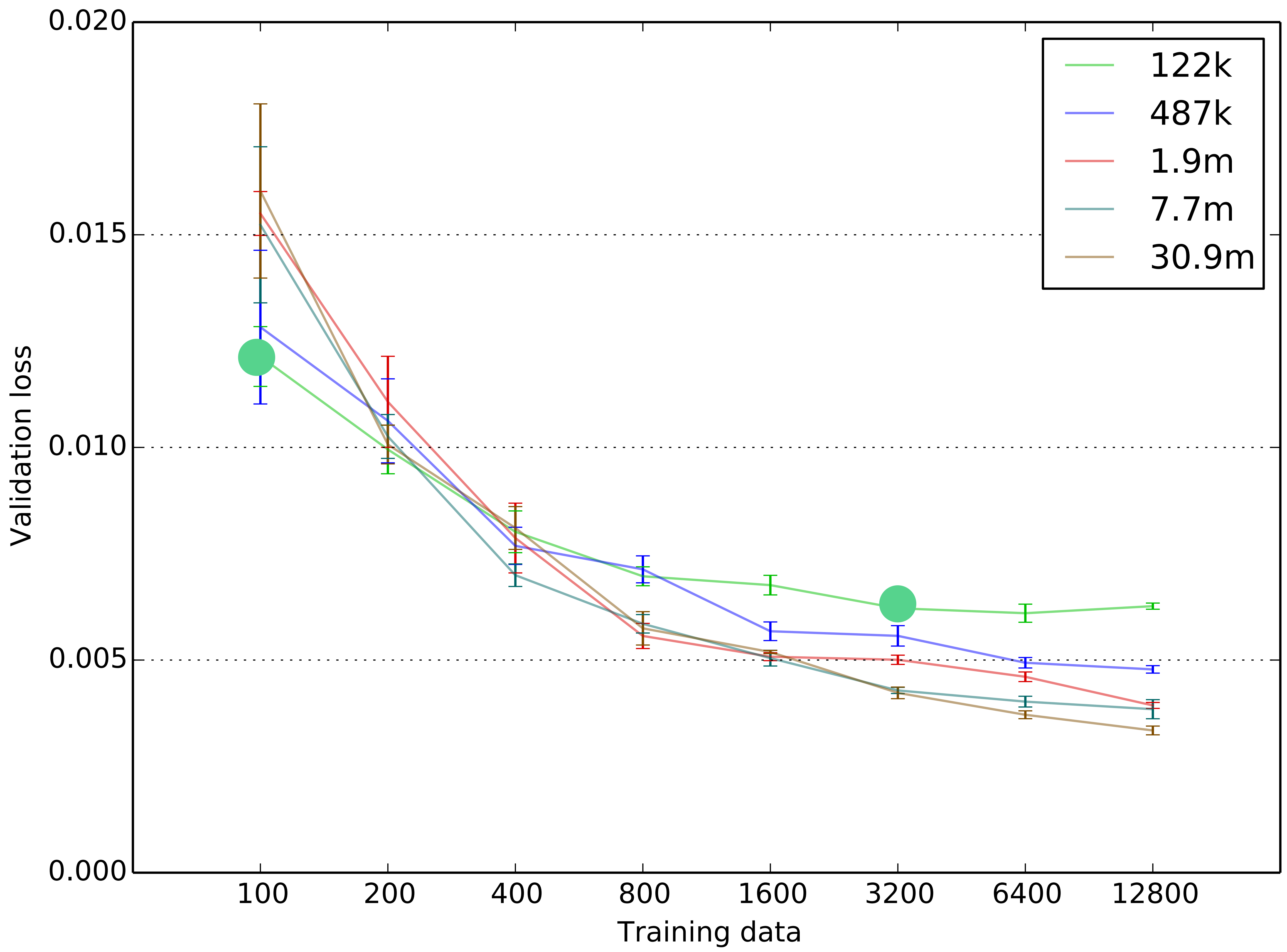
Solving PDEs with DL

- Validation and test accuracy for different model sizes



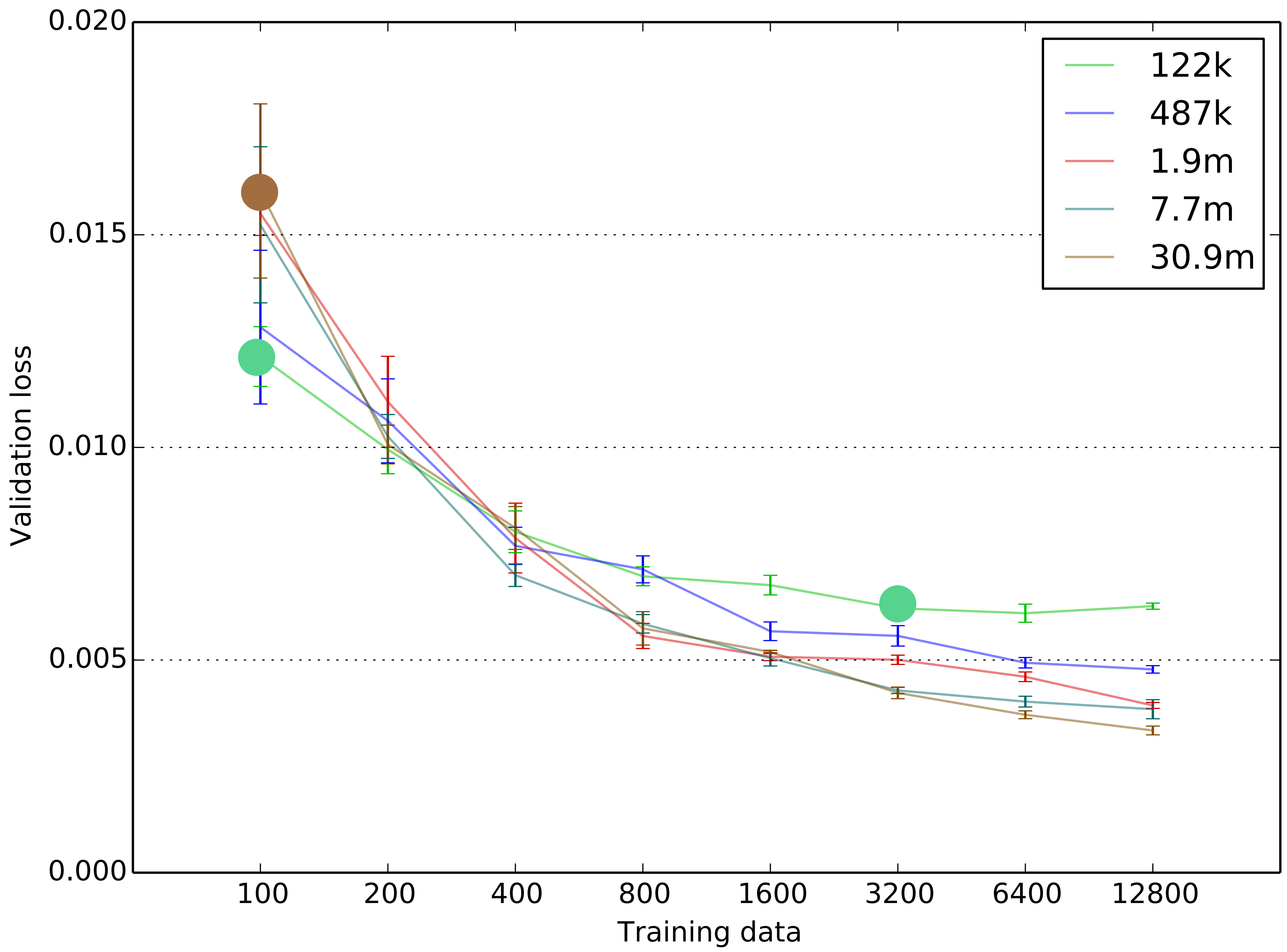
Solving PDEs with DL

- Validation and test accuracy for different model sizes



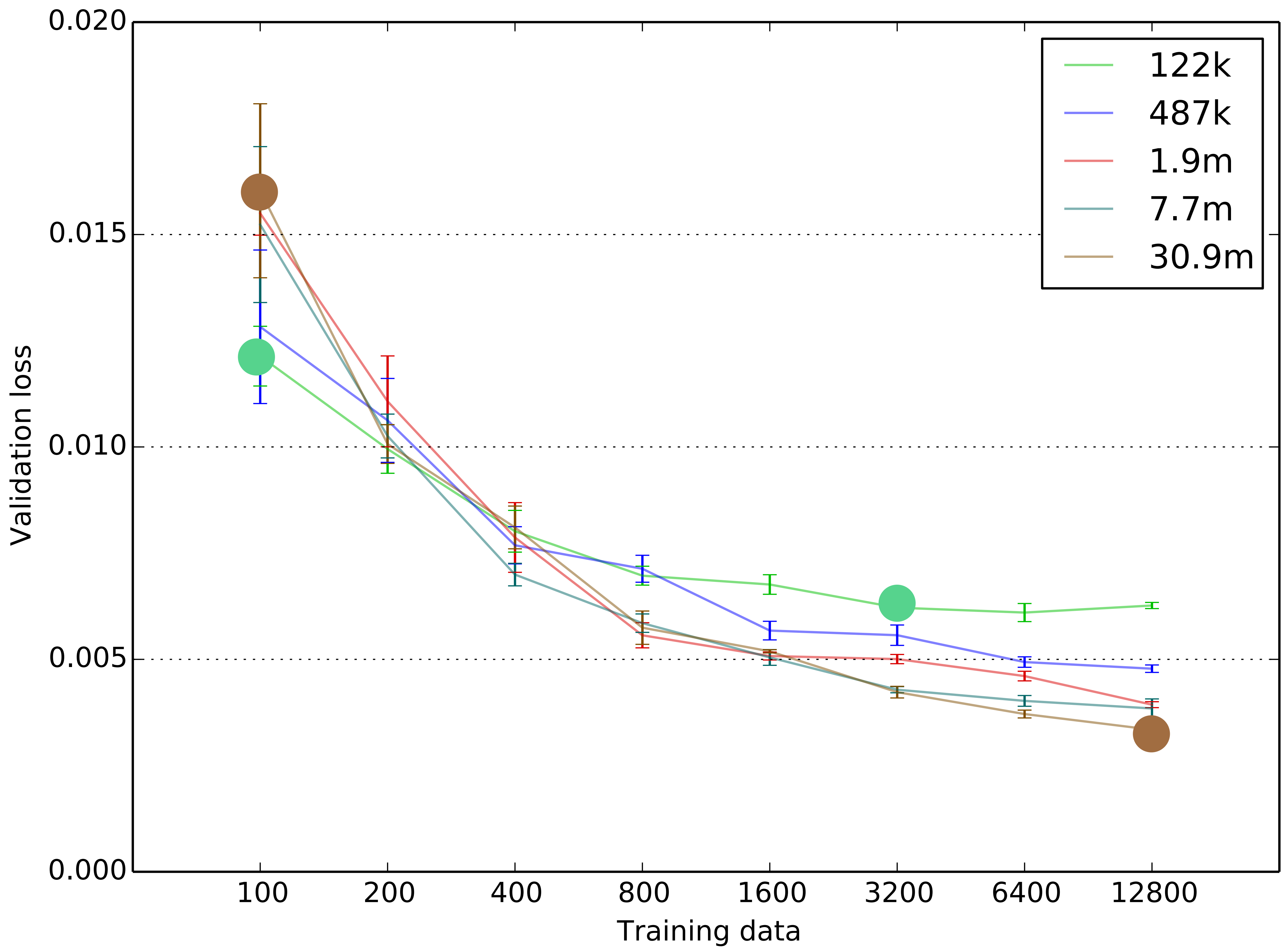
Solving PDEs with DL

- Validation and test accuracy for different model sizes



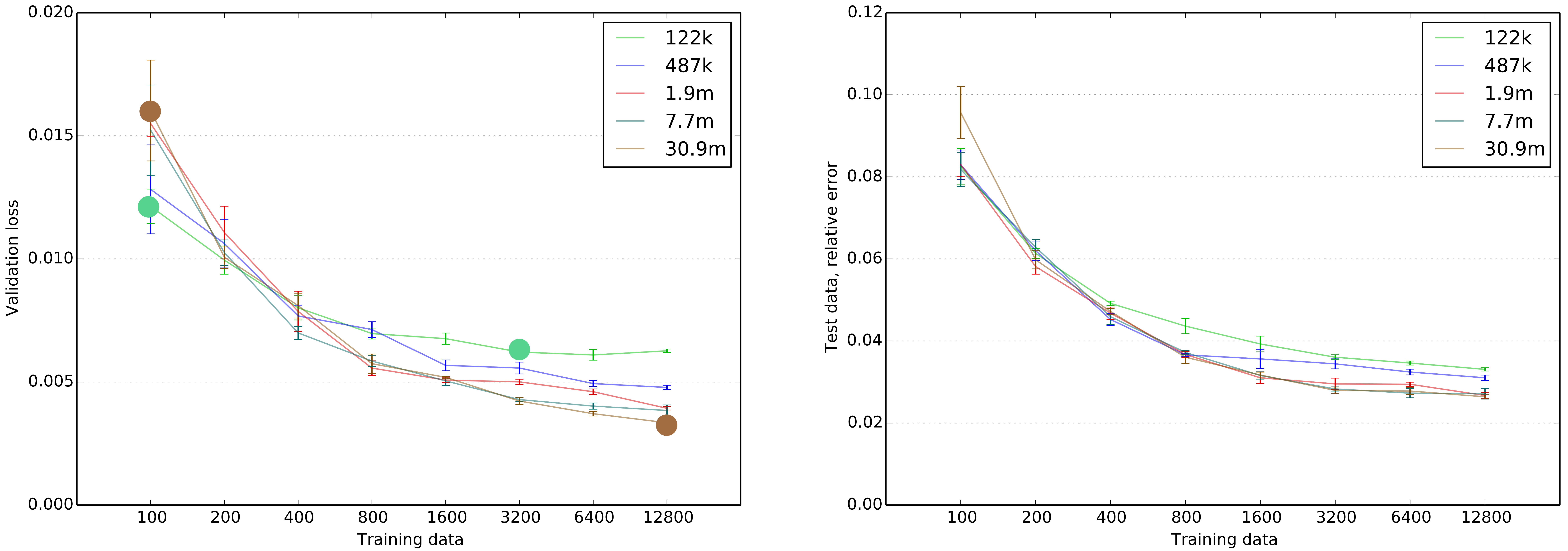
Solving PDEs with DL

- Validation and test accuracy for different model sizes



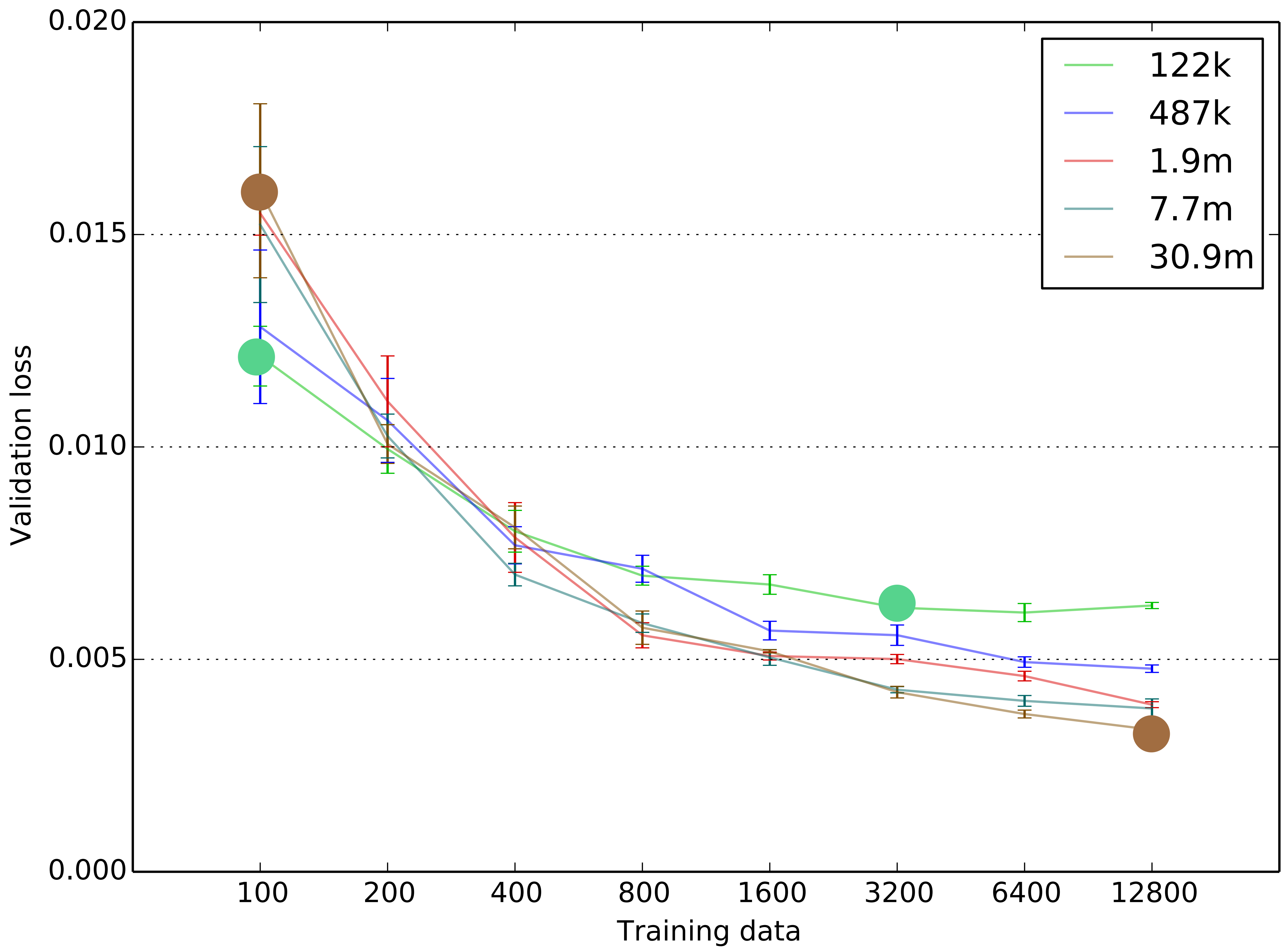
Solving PDEs with DL

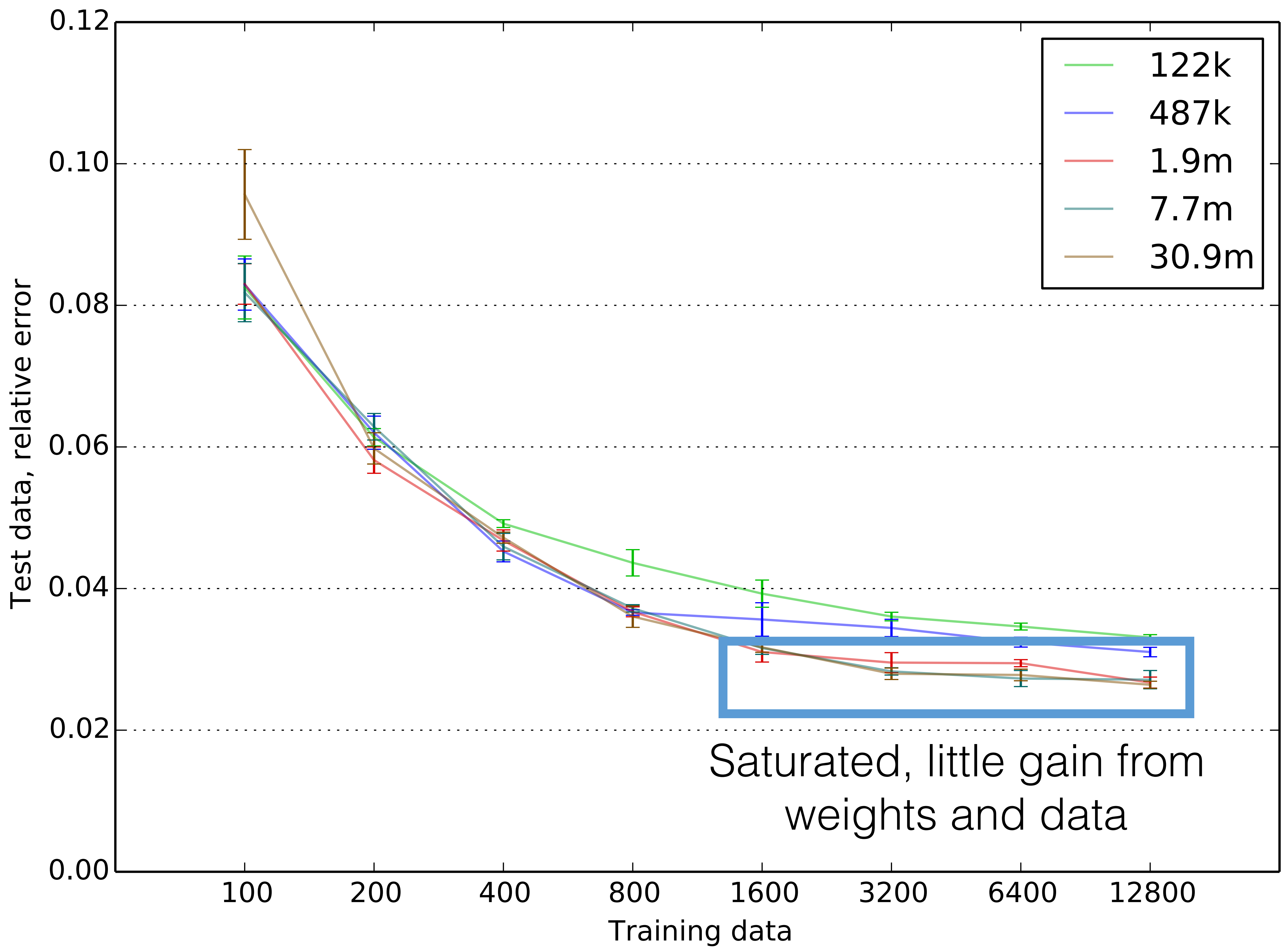
- Validation and test accuracy for different model sizes



Solving PDEs with DL

- Validation and test accuracy for different model sizes





Additional Examples

Additional Examples

- Elasticity: material models

Additional Examples

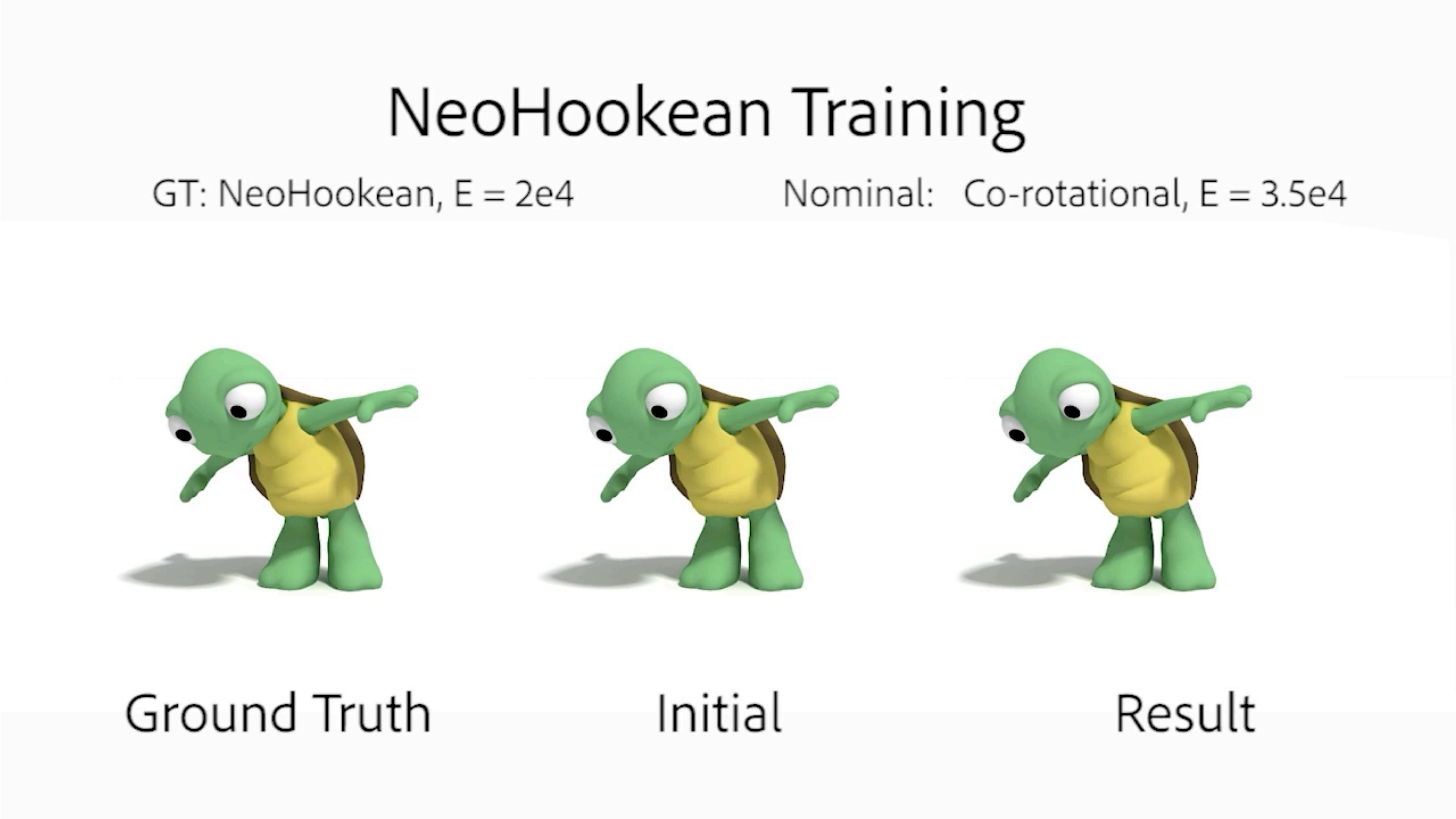
- Elasticity: material models
- Fluids: up-res algorithm & dimensionality reduction

Additional Examples

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

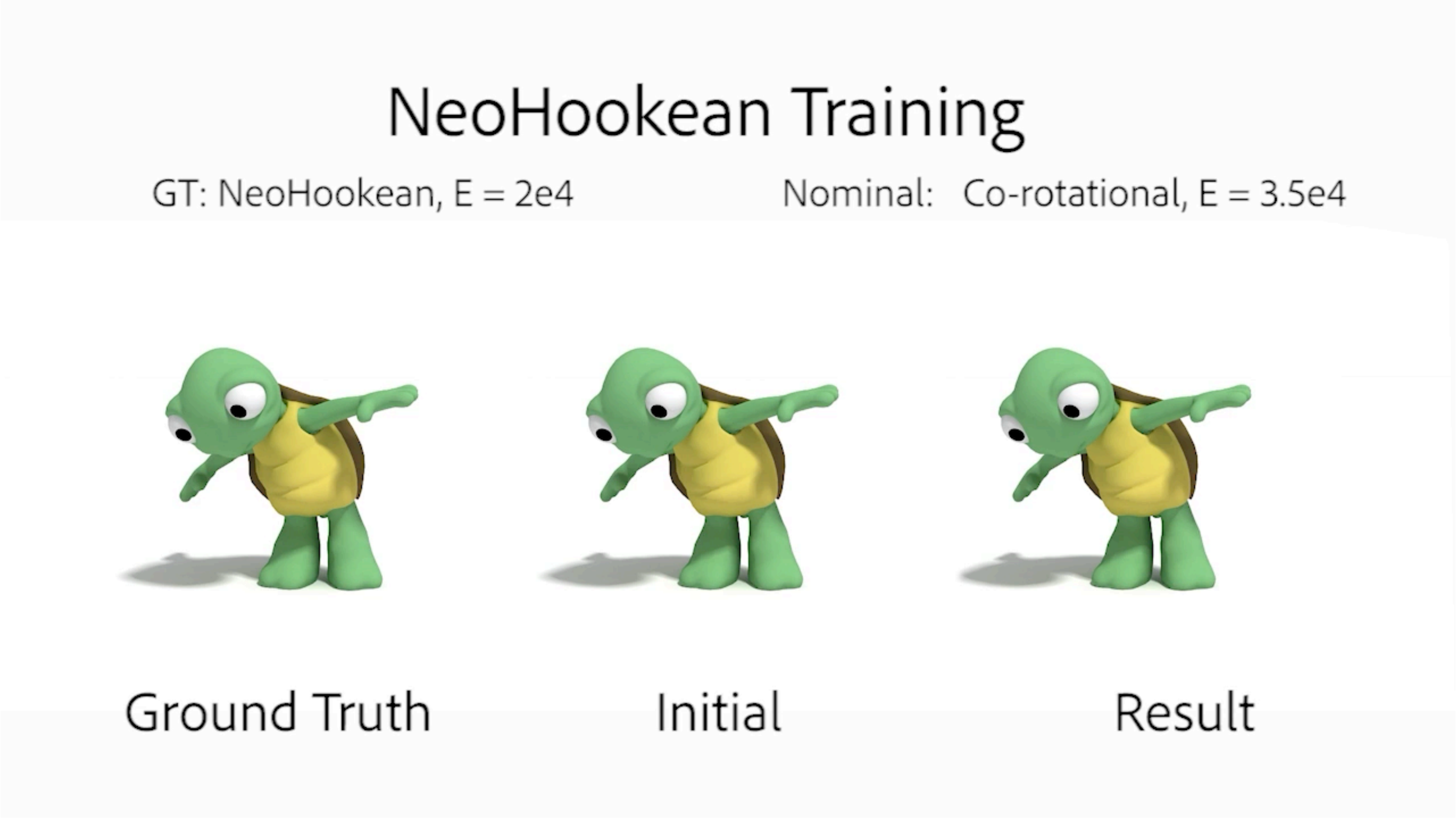
Neural Material - Elasticity

- Learn correction of regular FEM simulation for complex materials



Neural Material - Elasticity

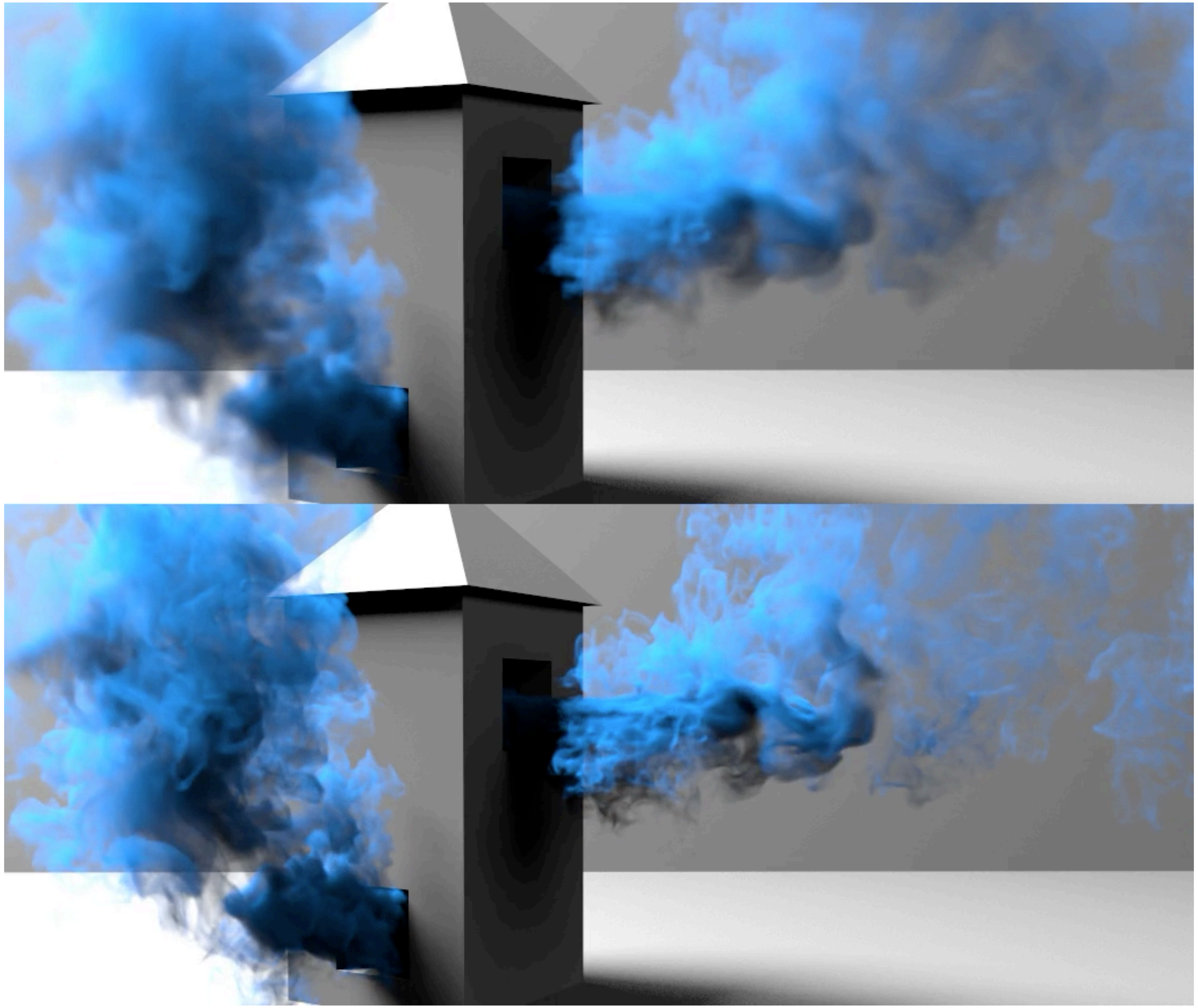
- Learn correction of regular FEM simulation for complex materials



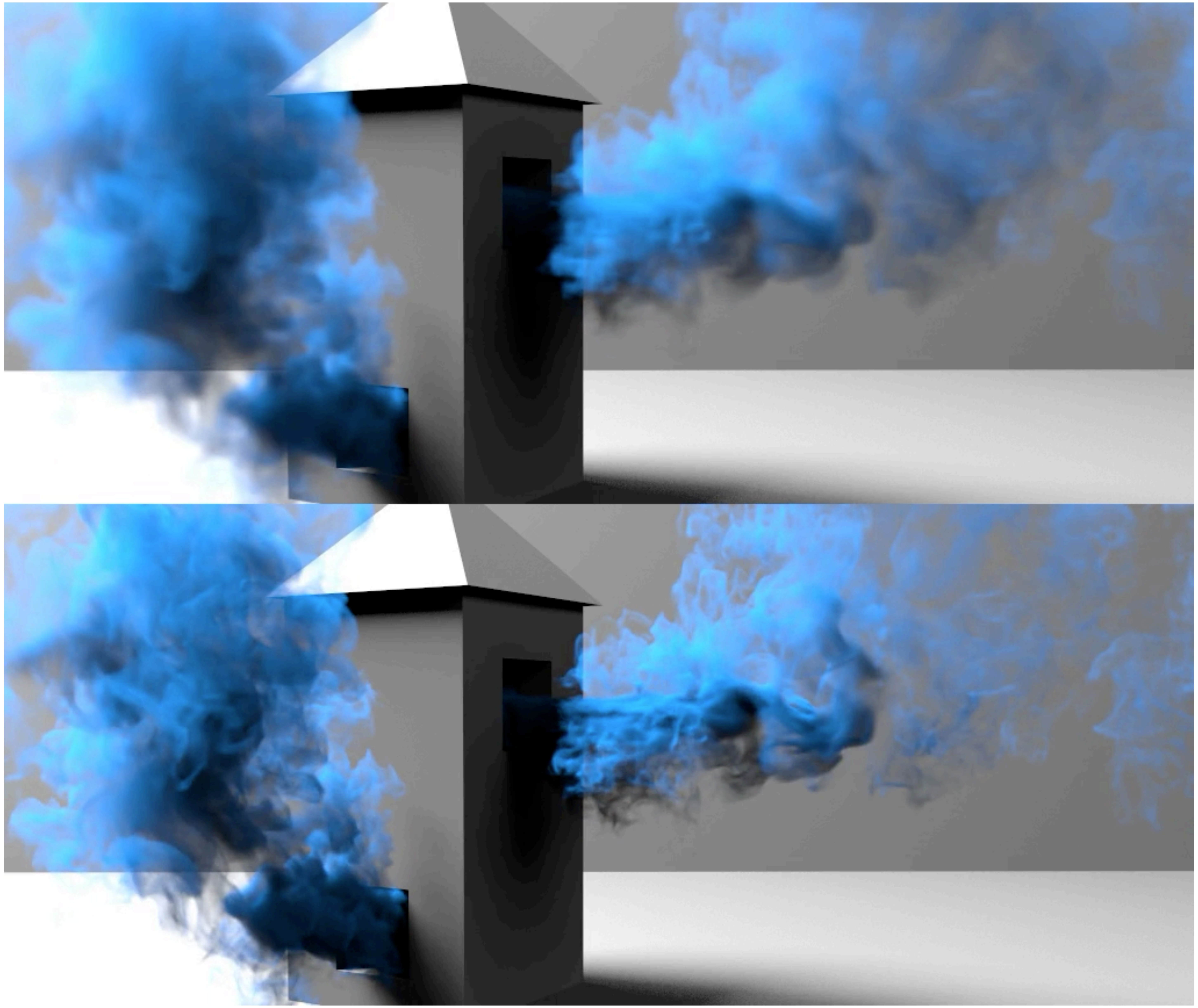
Neural Material - Elasticity

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

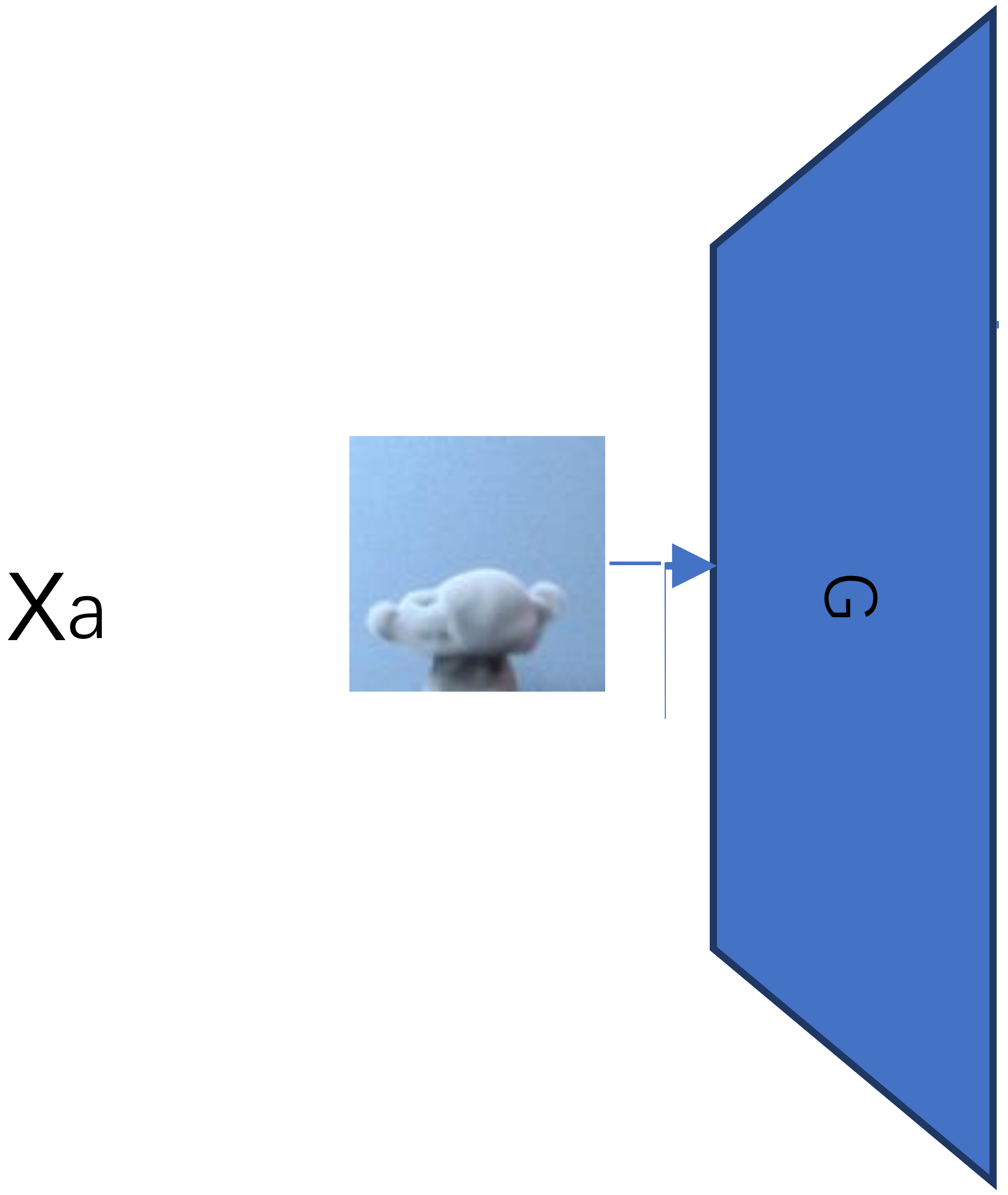
Temporal Data



Temporal Data

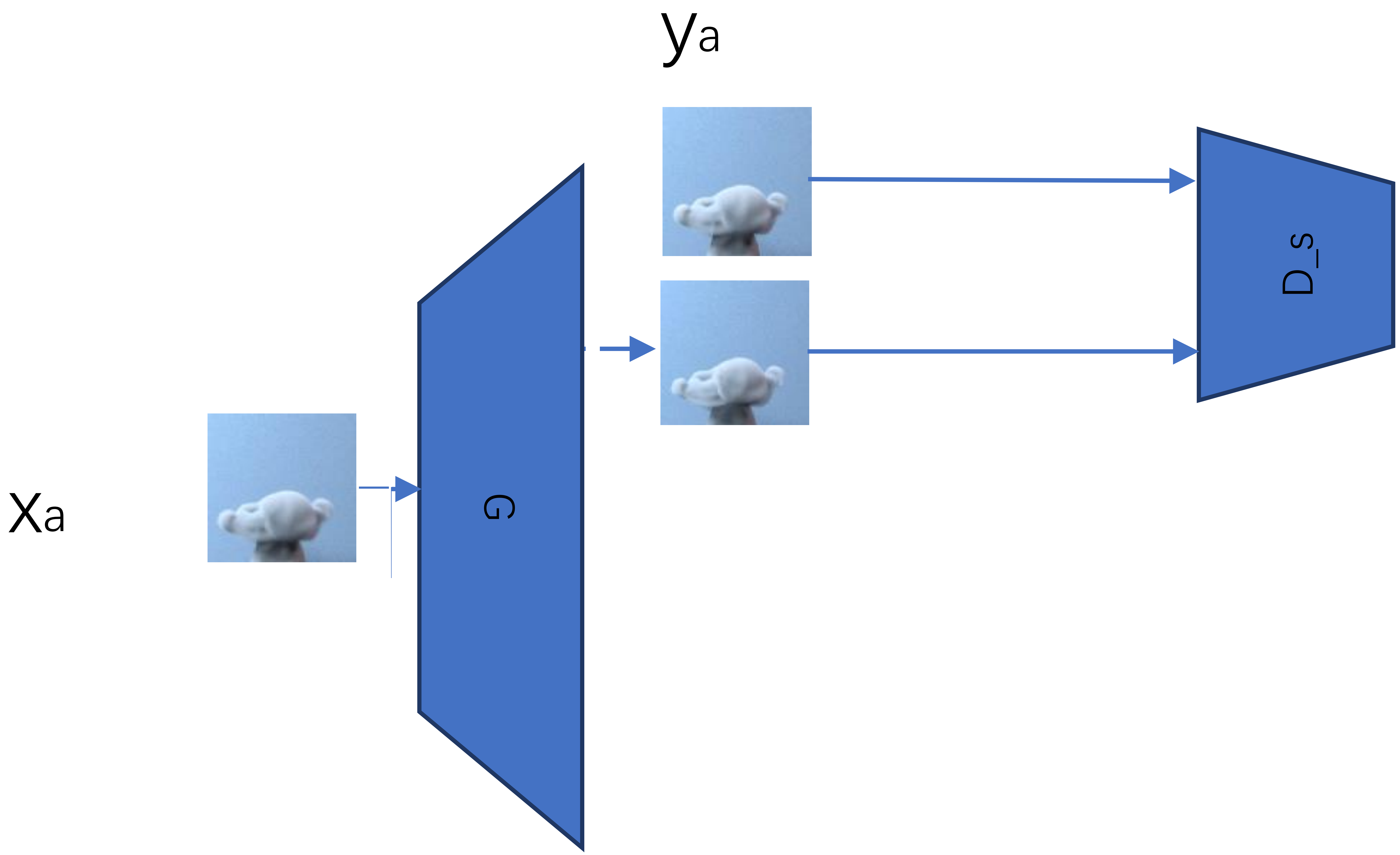


Temporal Data



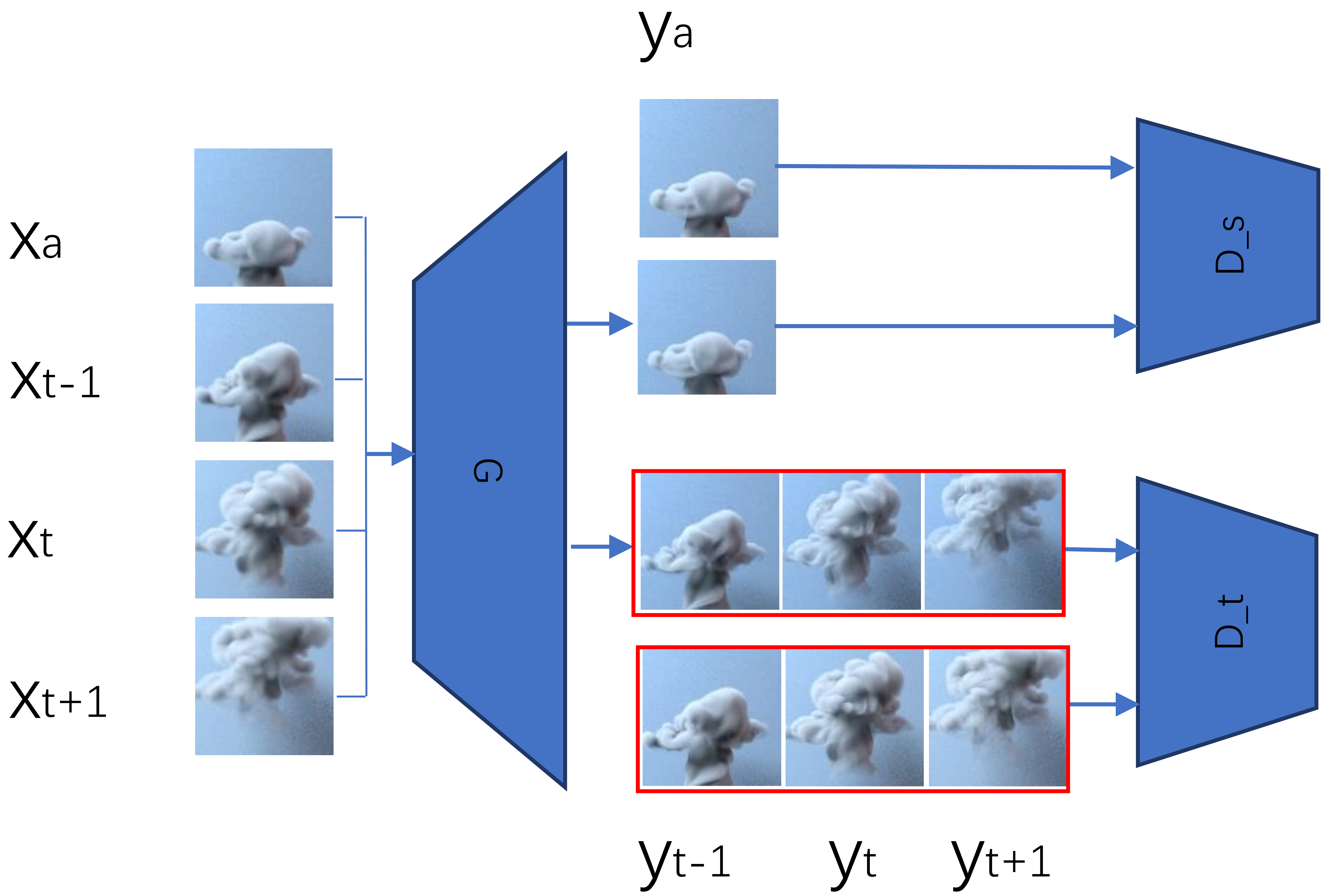
[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

Temporal Data

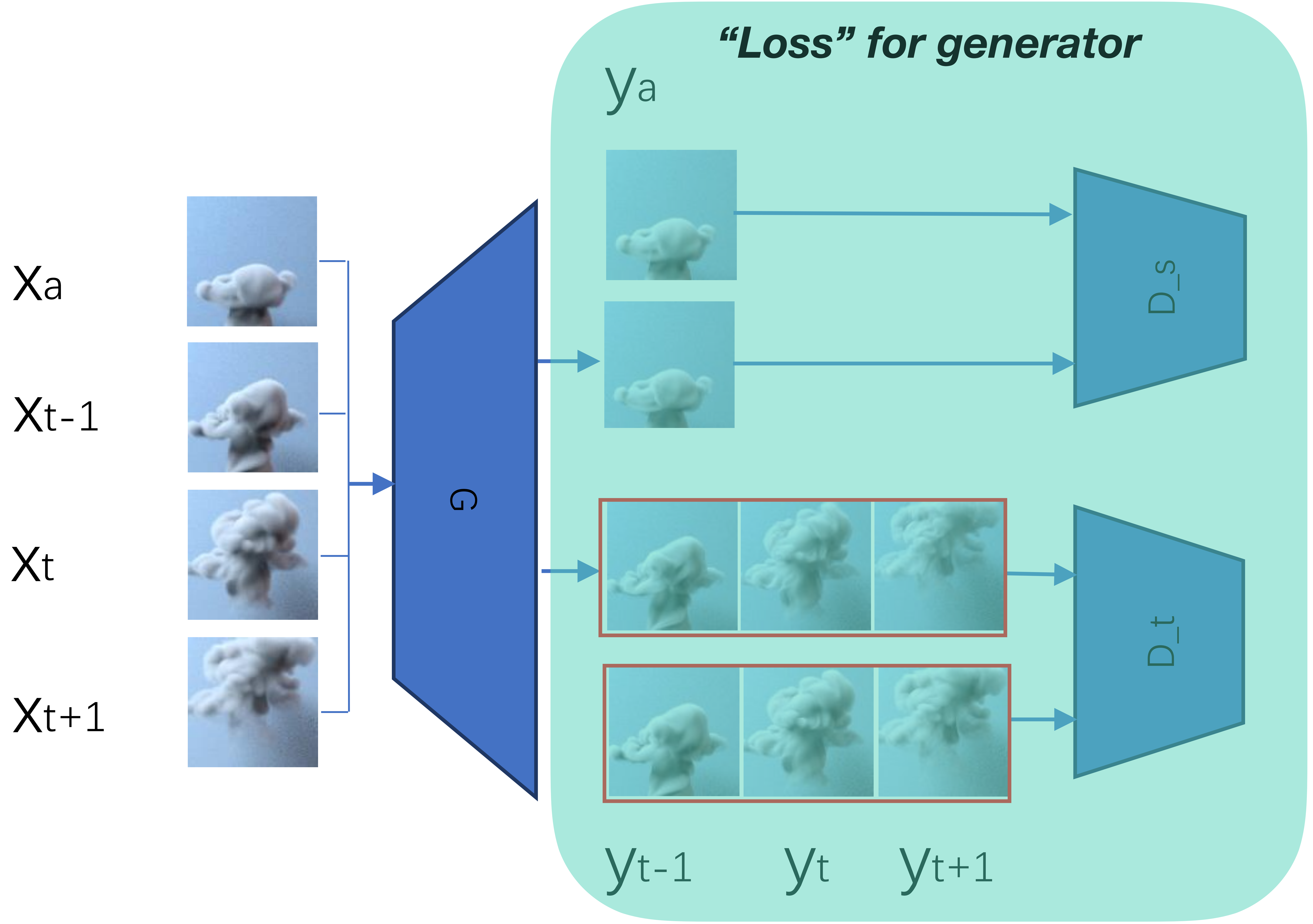


[tempoGAN: A Temporally Coherent, Volumetric GAN for Super-resolution Fluid Flow , SIGGRAPH 2018]

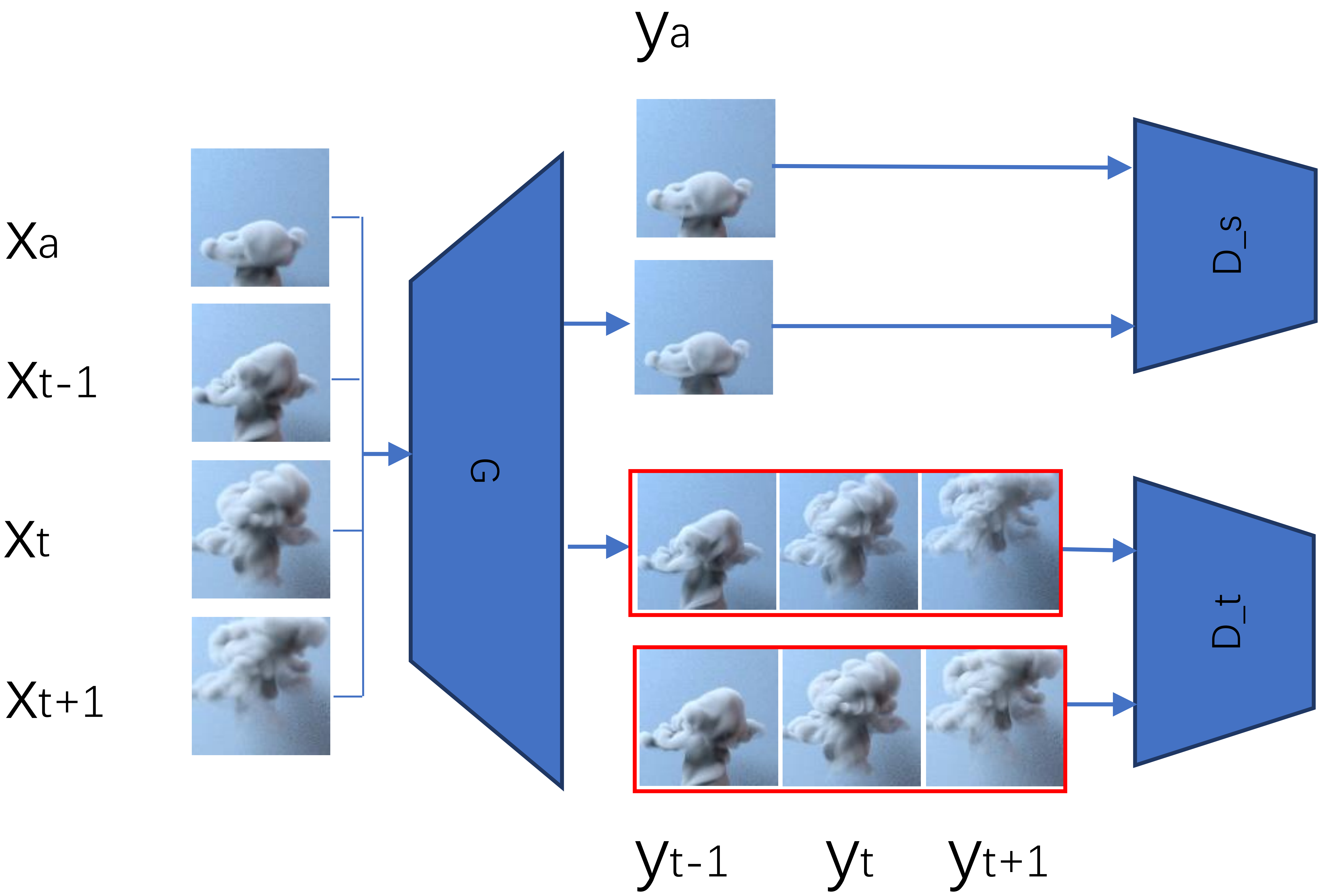
Temporal Dat



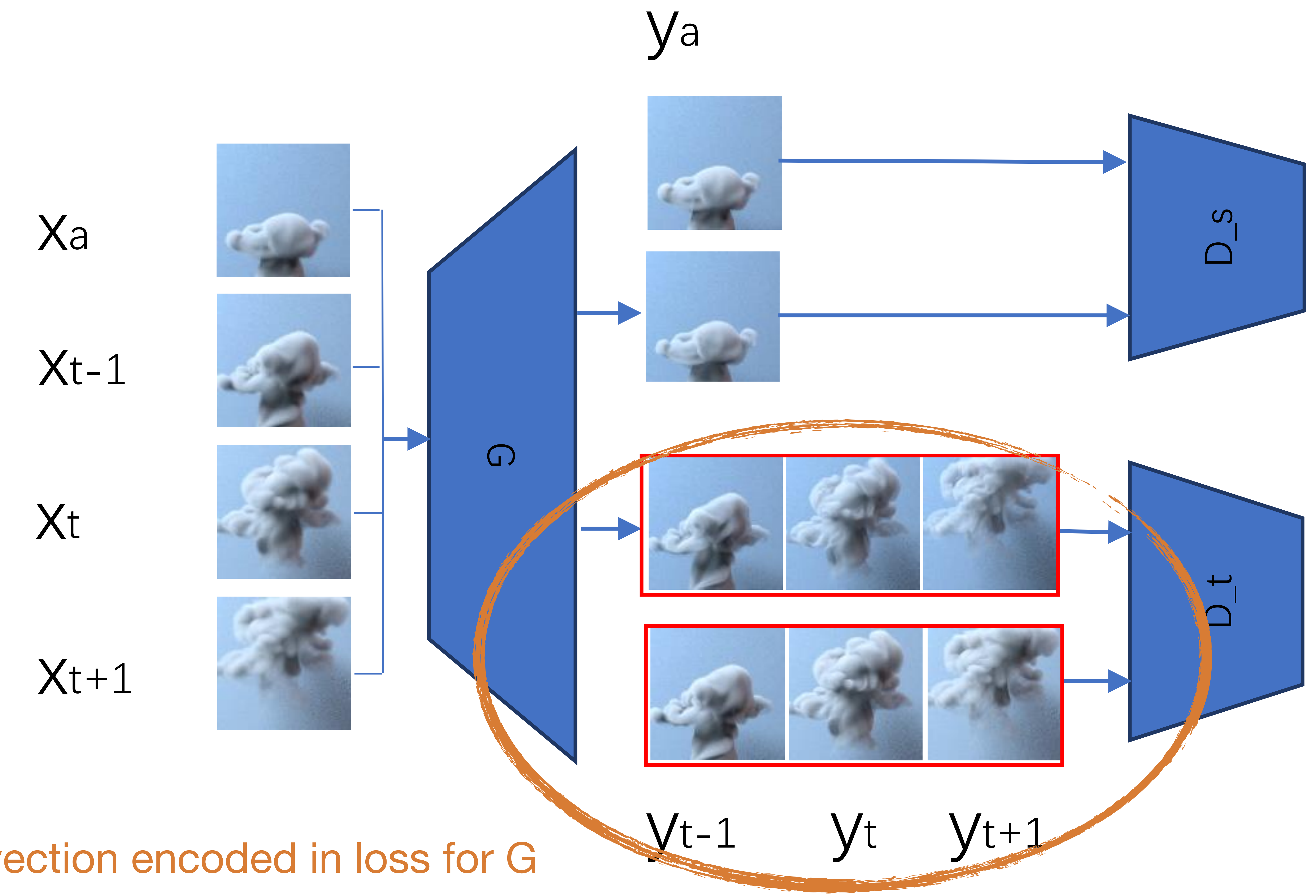
Temporal Dat



Temporal Dat



Temporal Dat



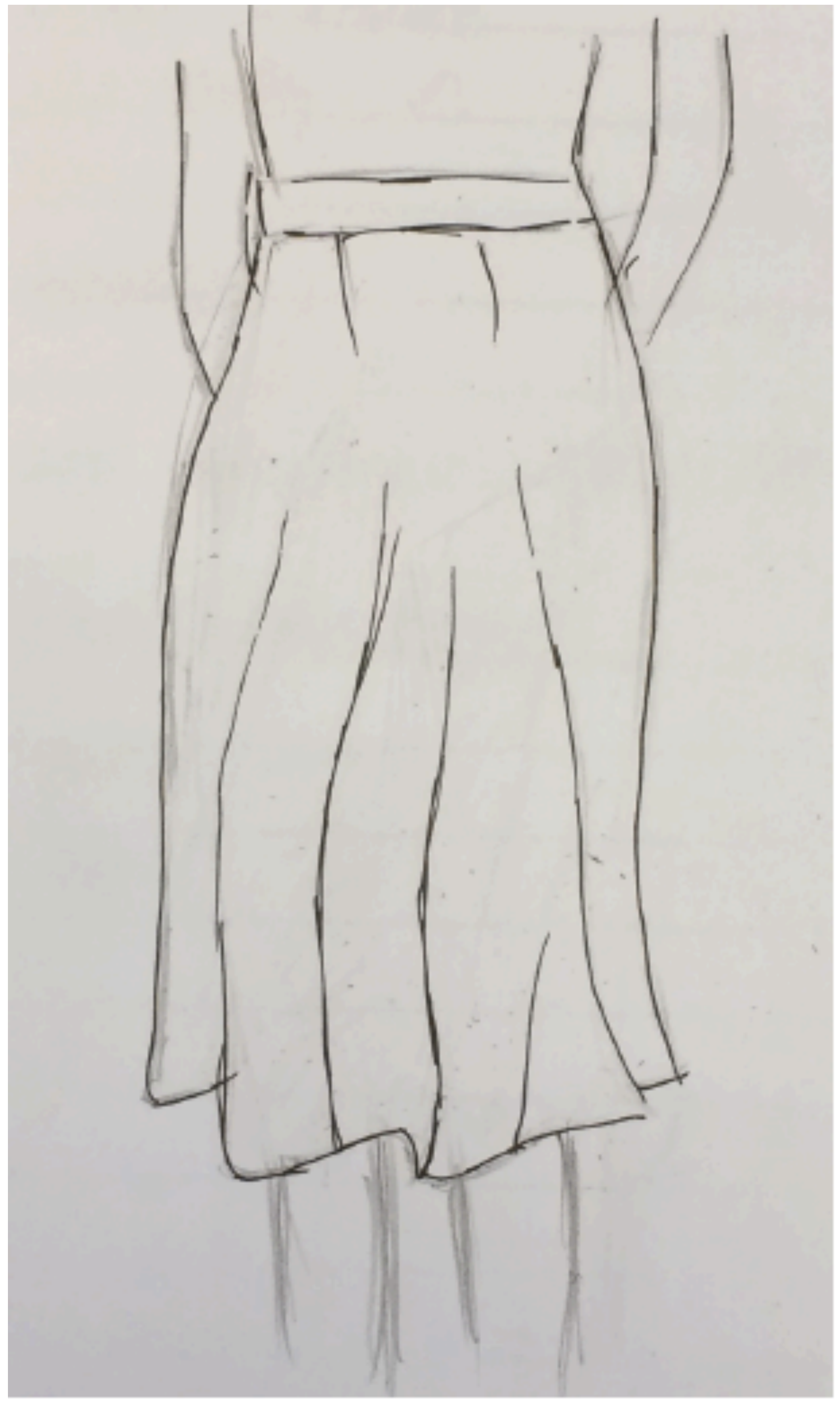
Advection encoded in loss for G

Temporal Data

Temporal Data

Design Options

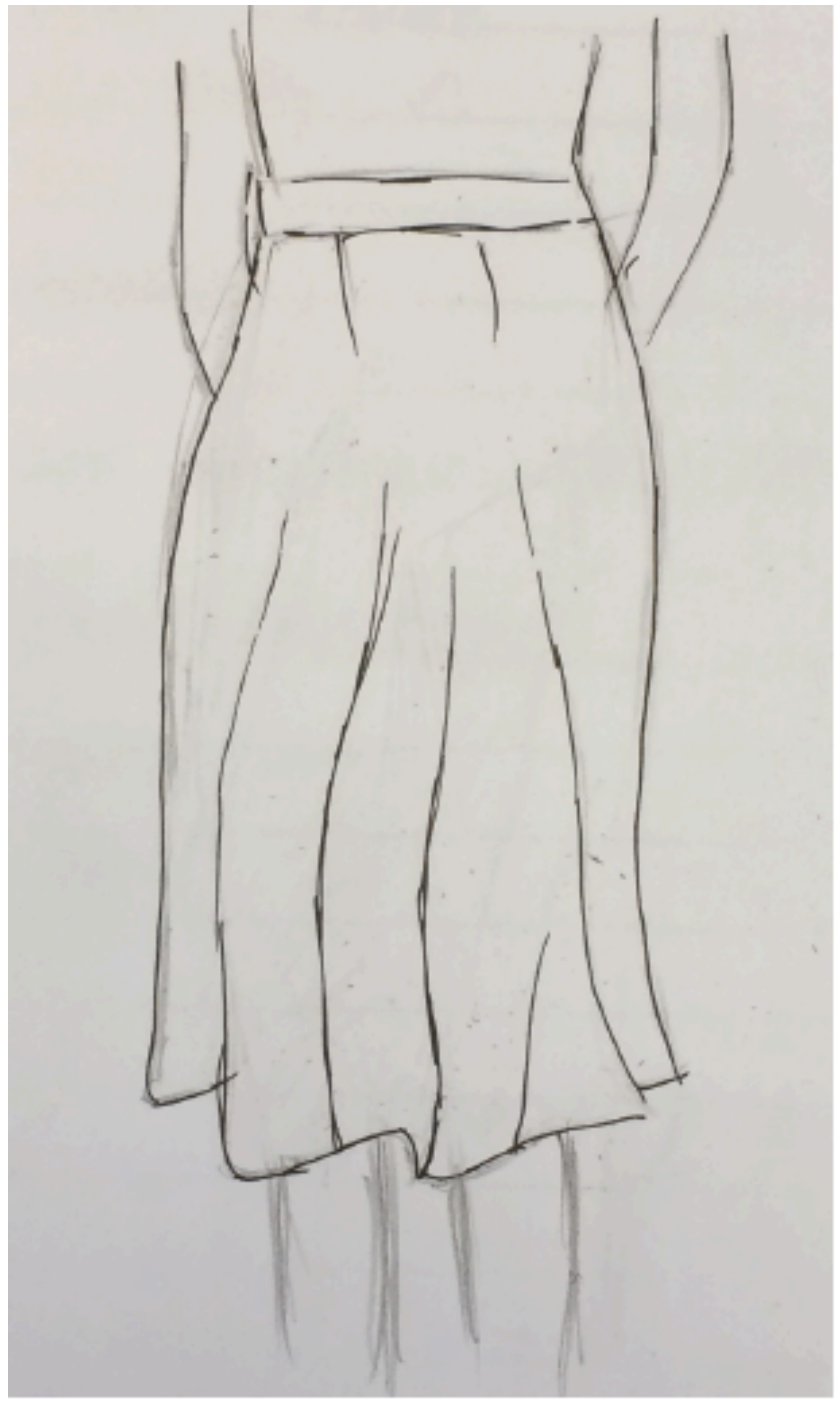
[Wang, Ceylan, Popovic, Mitra, Siggraph Asia, 2018]



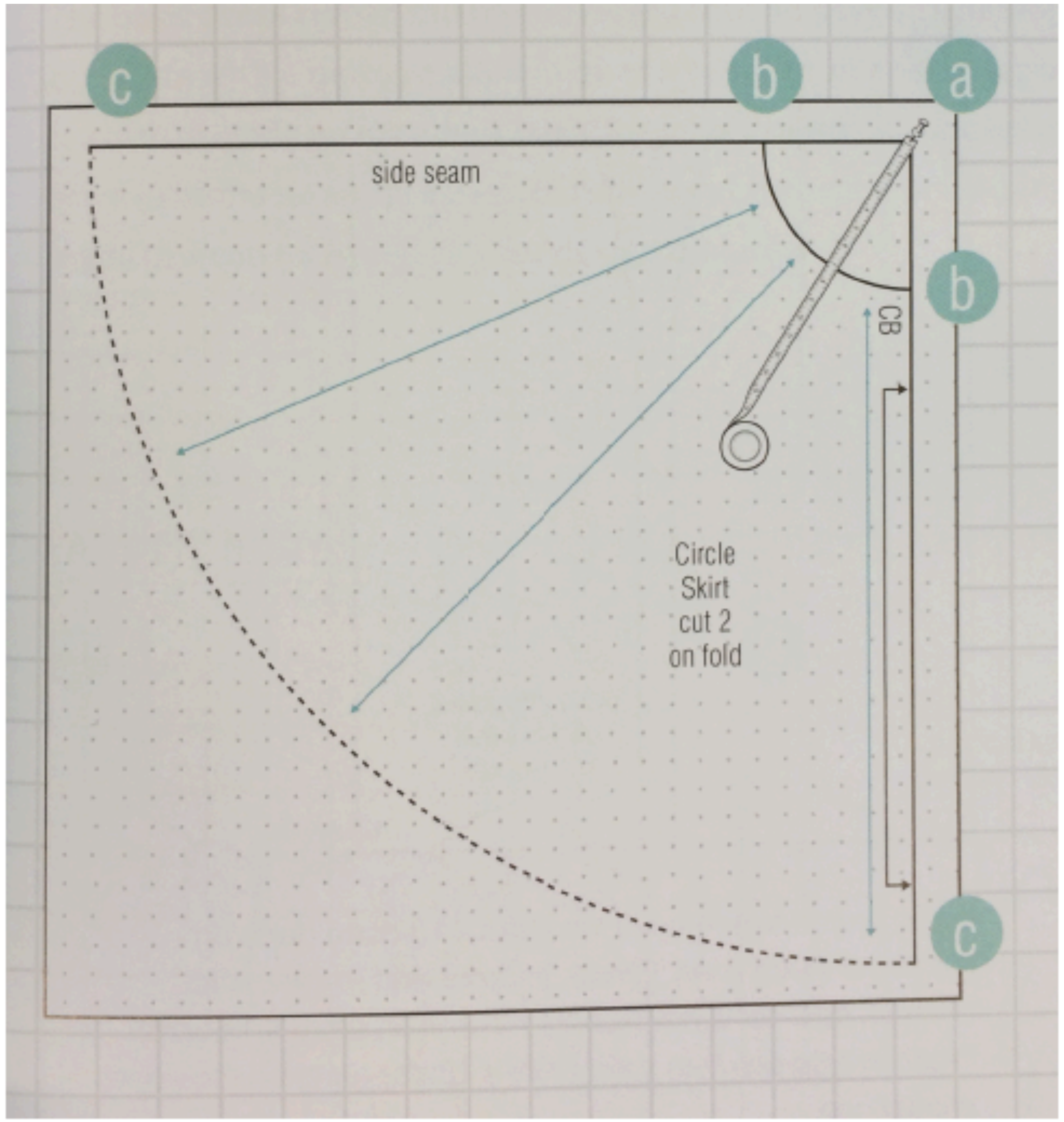
1. sketching

Design Options

[Wang, Ceylan, Popovic, Mitra, Siggraph Asia, 2018]



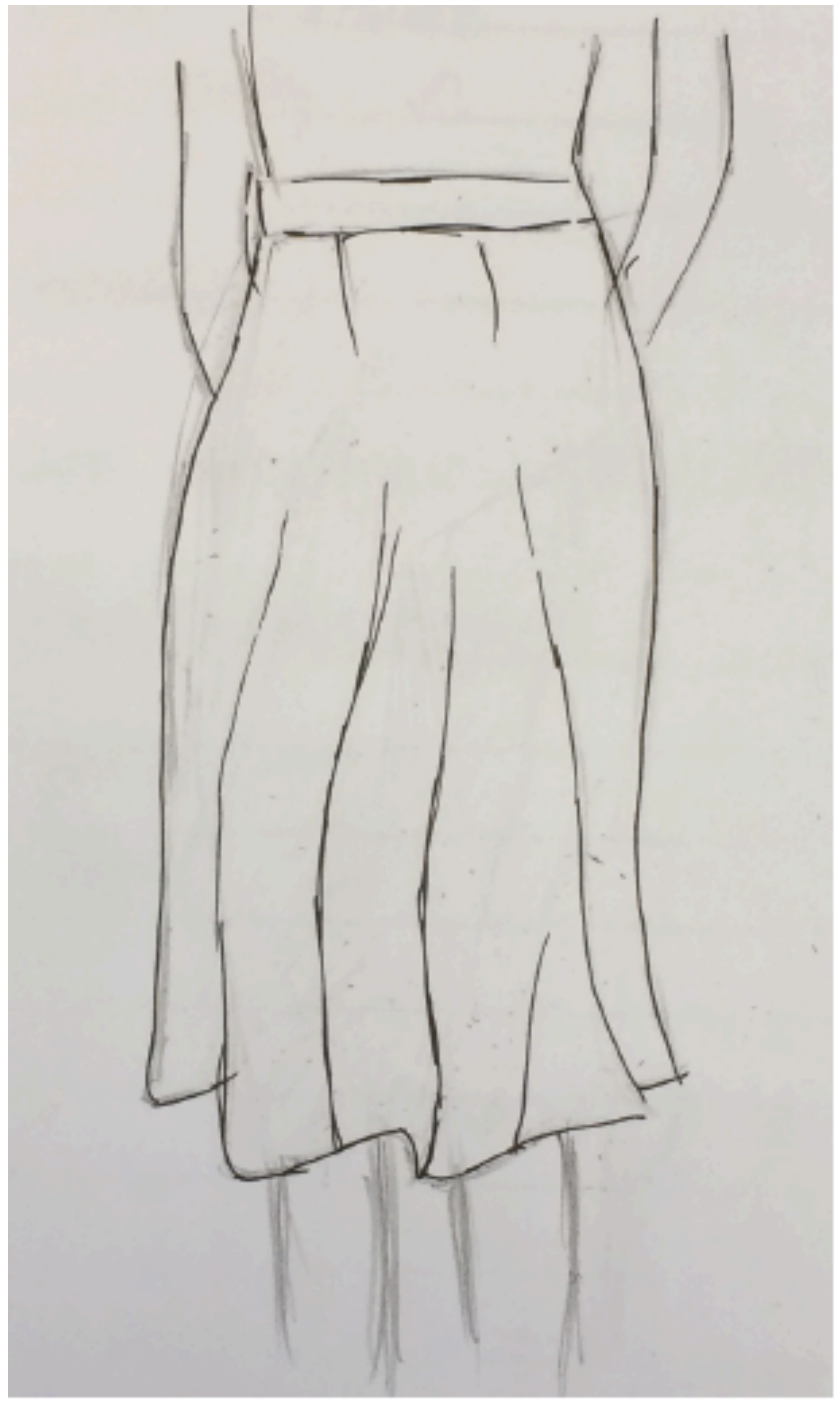
1. sketching



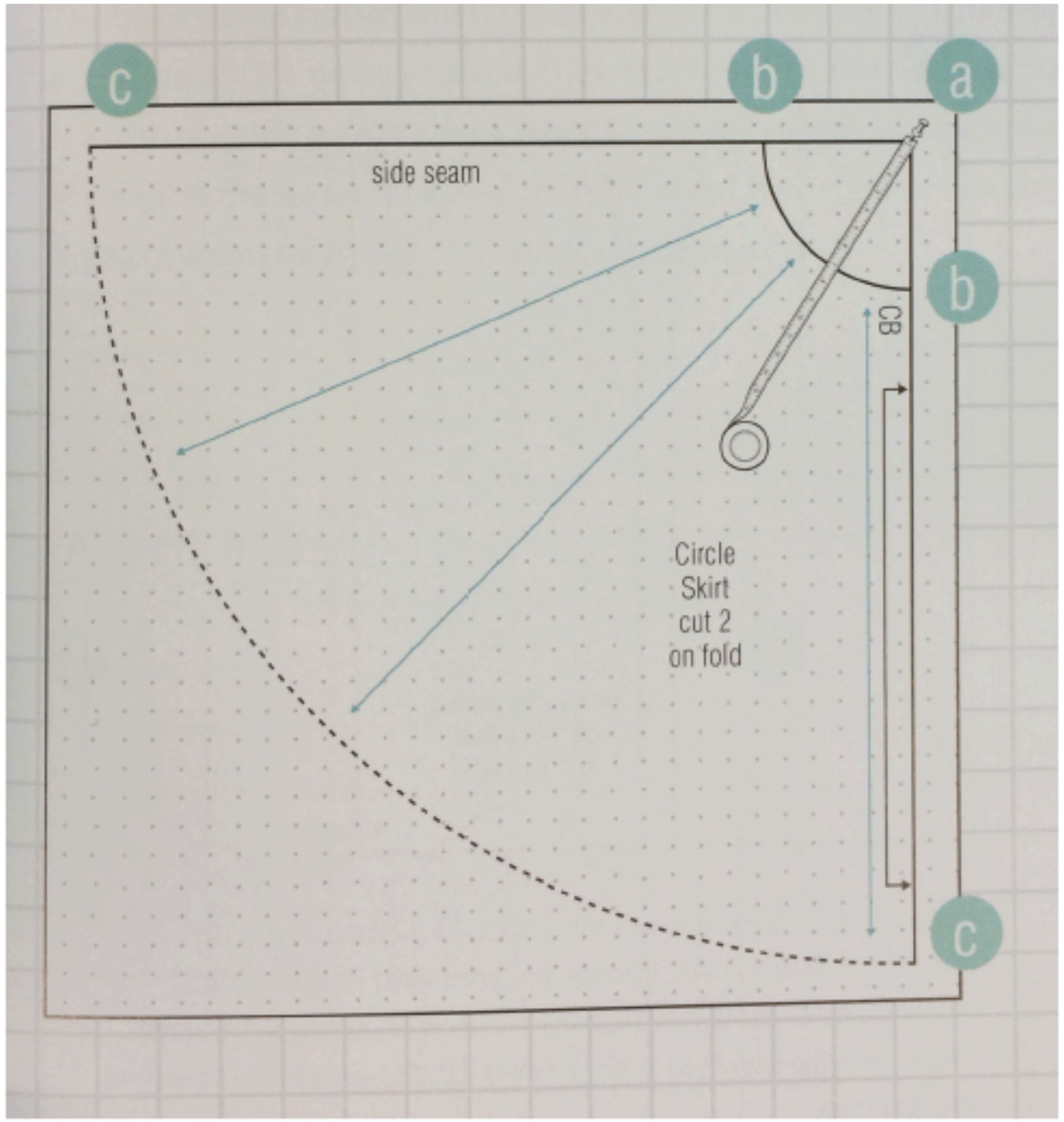
2. sewing patterns

Design Options

[Wang, Ceylan, Popovic, Mitra, Siggraph Asia, 2018]



1. sketching

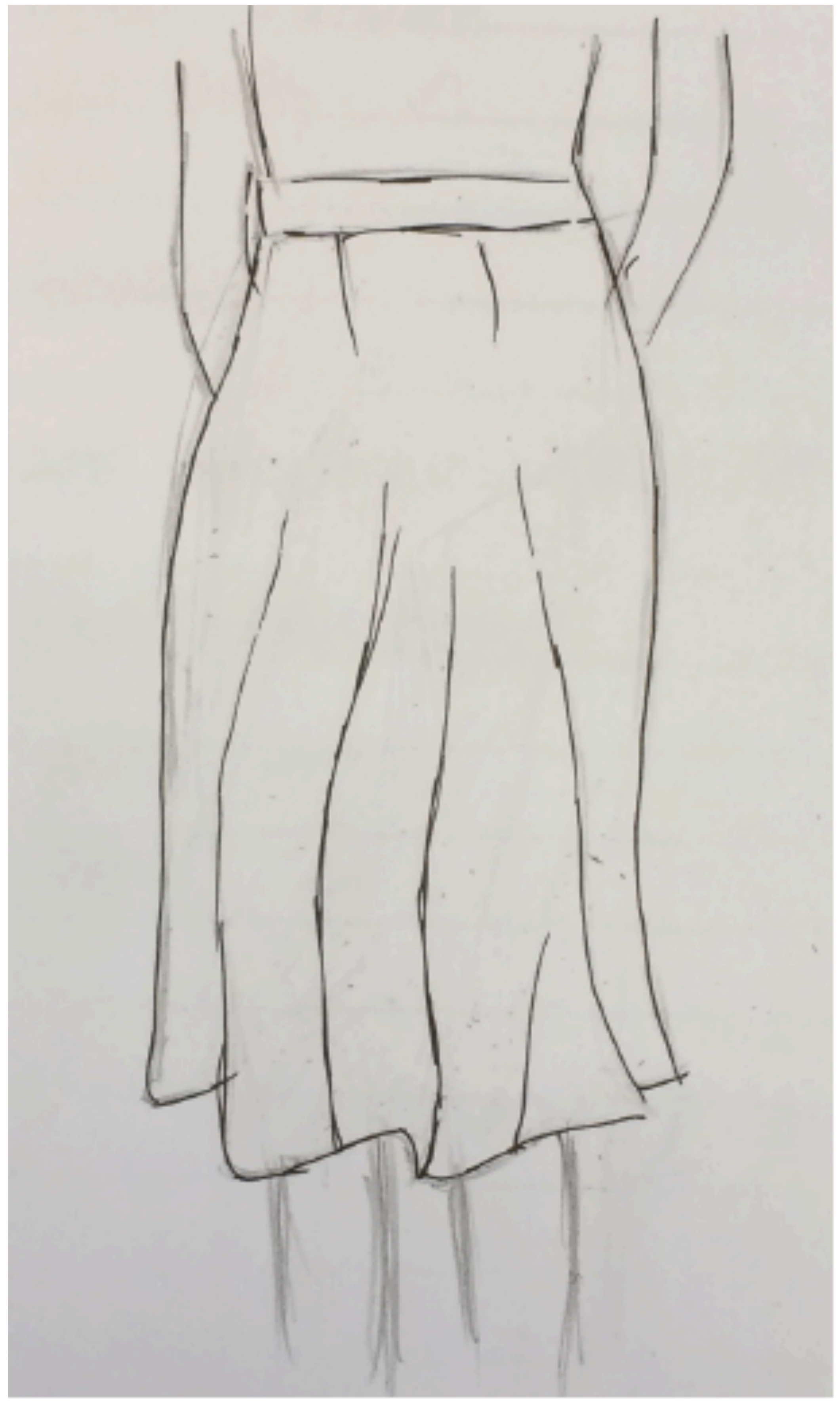


2. sewing patterns

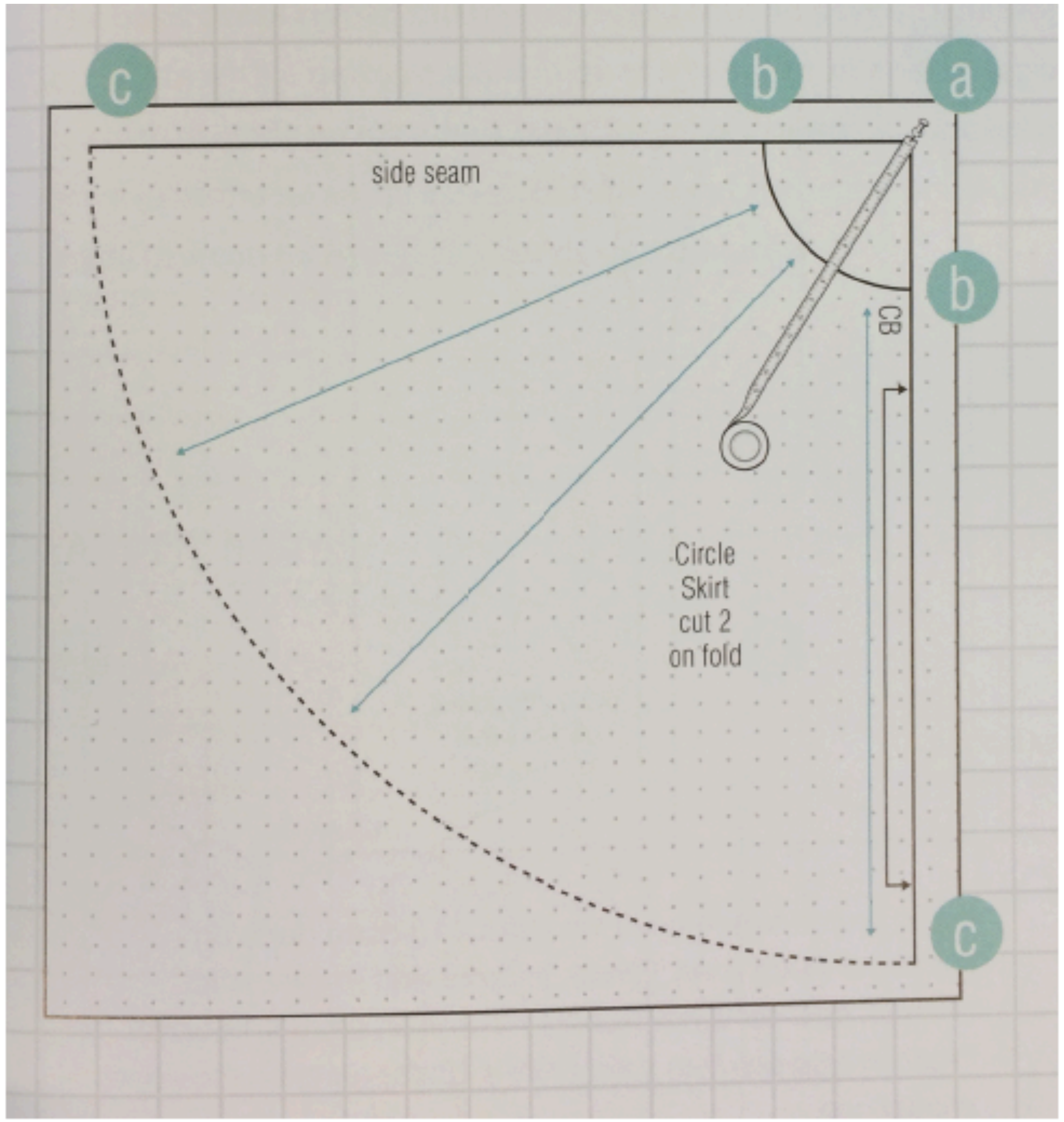
3. draped garment

Design Options

[Wang, Ceylan, Popovic, Mitra, Siggraph Asia, 2018]



1. sketching



2. sewing patterns

3. draped garment

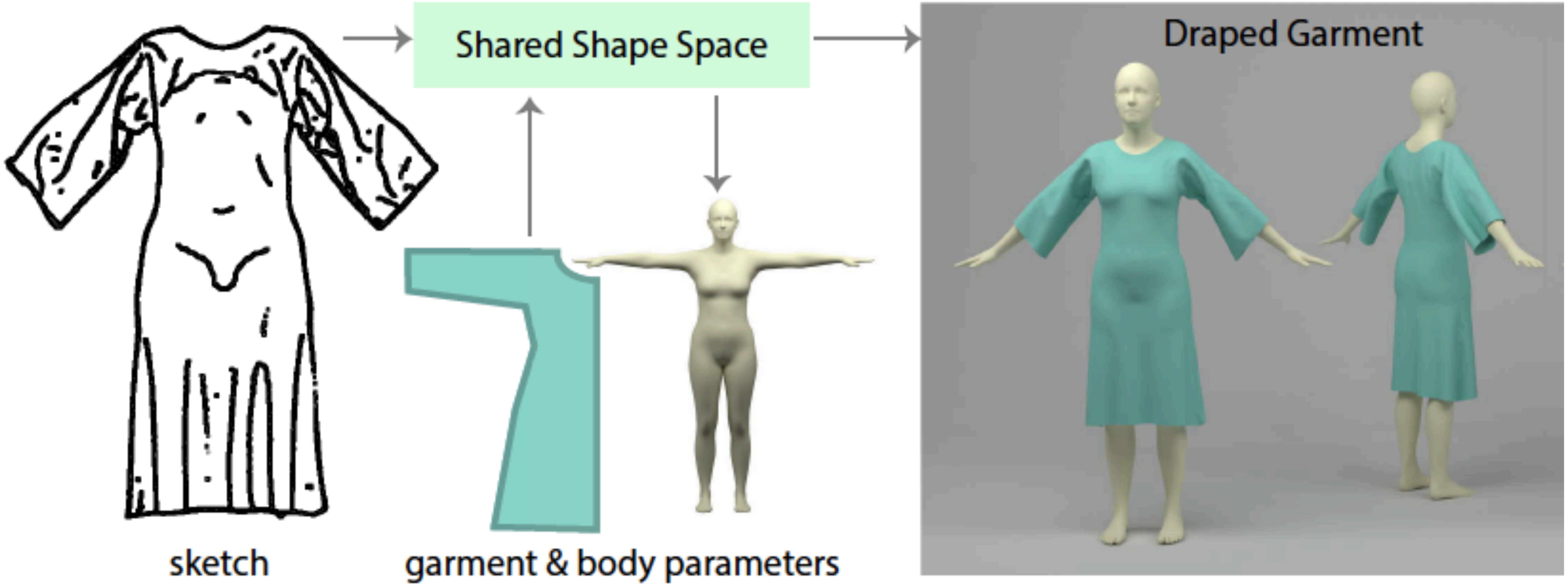
= interaction(sewing pattern, material, body shape)

Interaction through Simulation

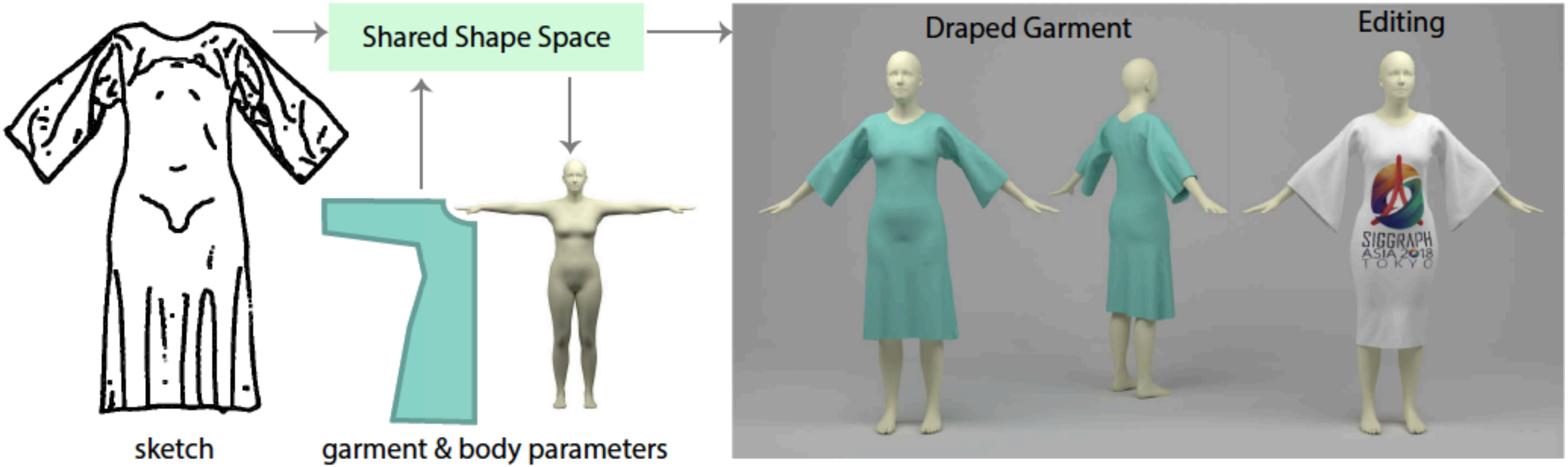
Interaction through Simulation

realistic simulations but NOT interactive

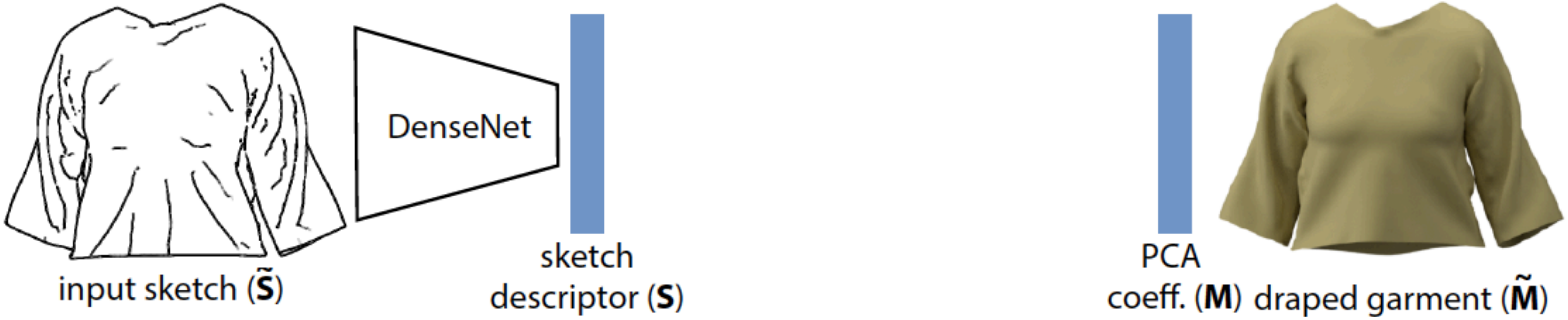
Multimodal Design



Multimodal Design

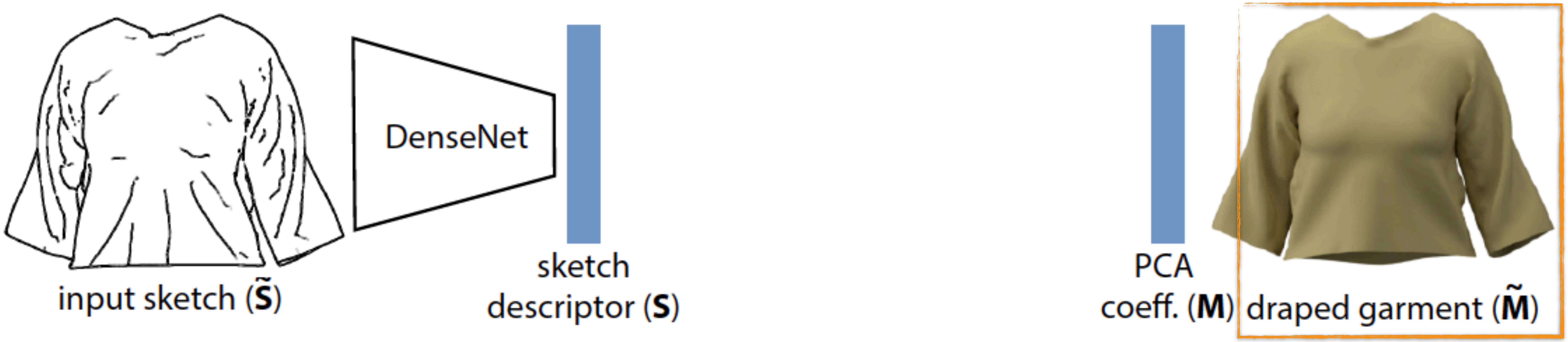


Learning a Latent Space (AutoEncoder)

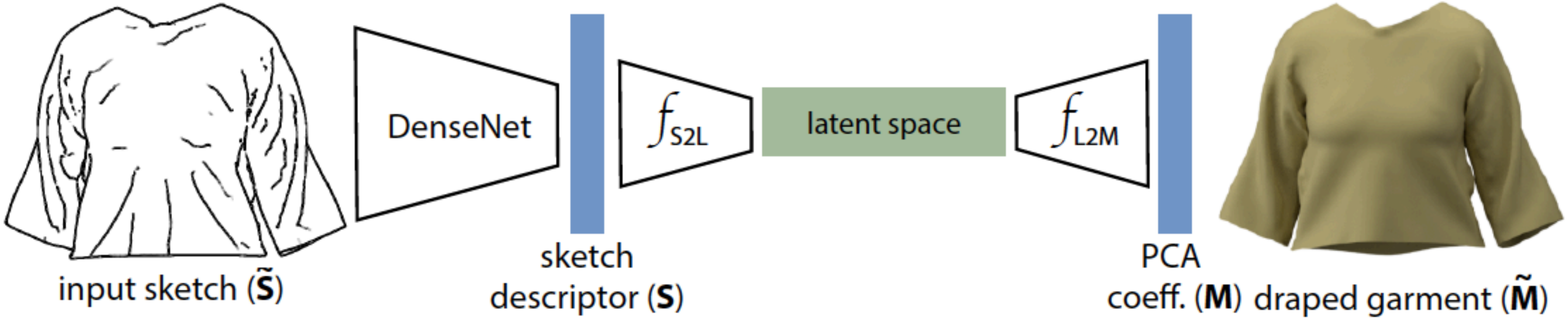


Learning a Latent Space (AutoEncoder)

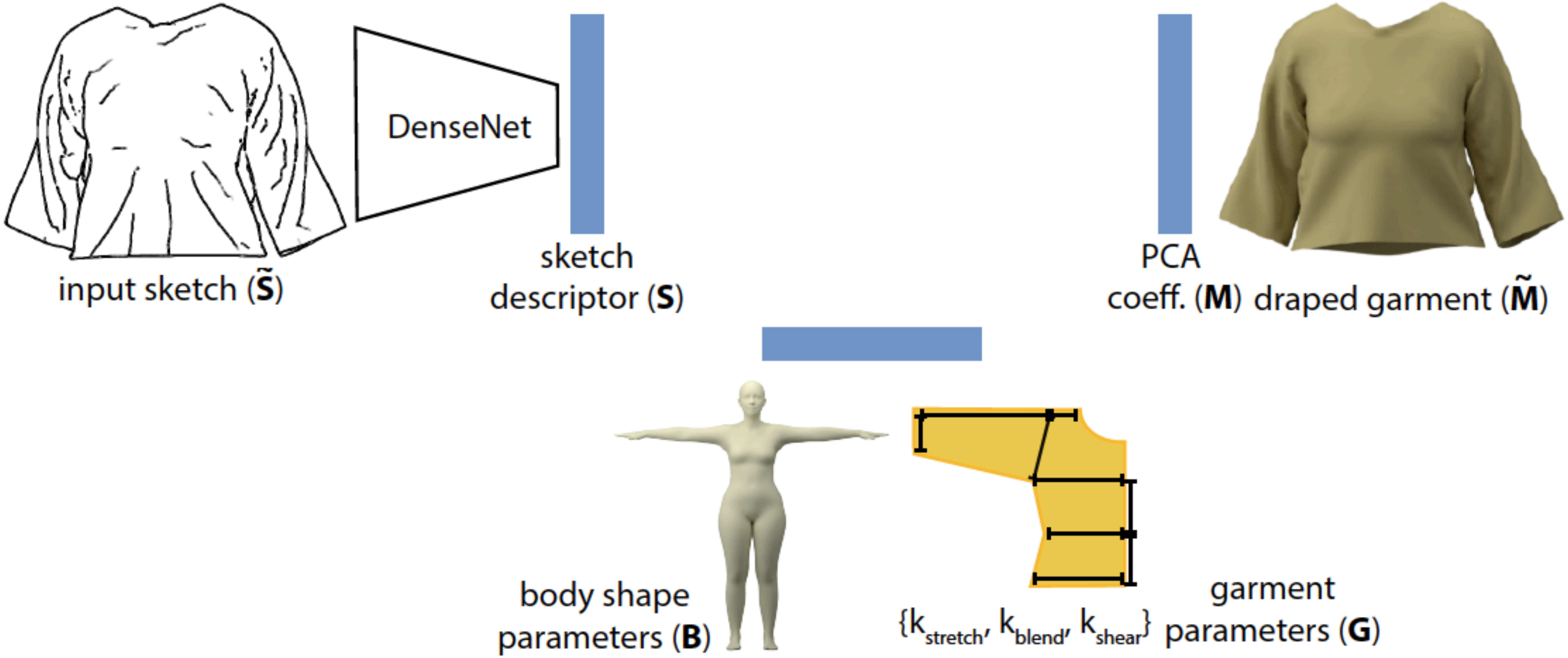
Learning a Latent Space (AutoEncoder)



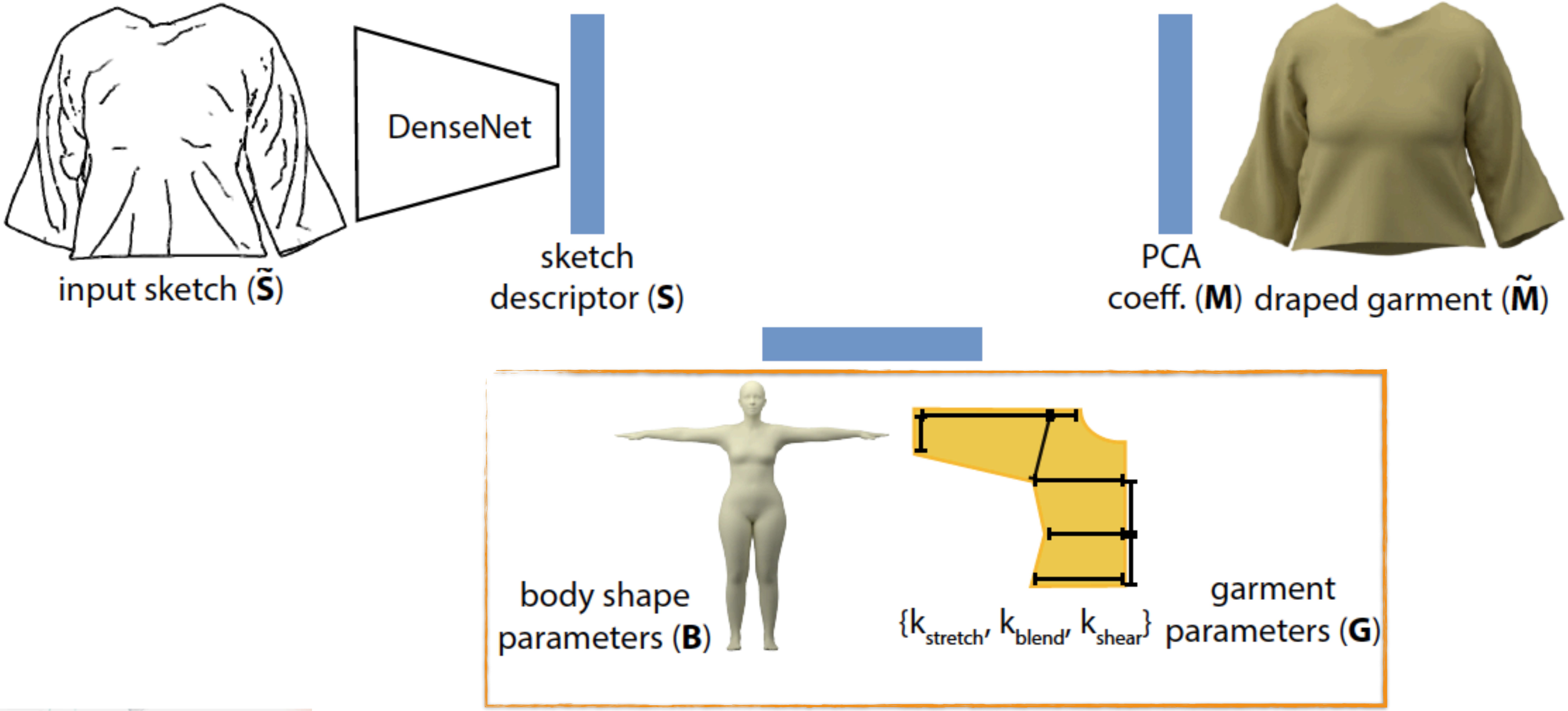
Learning a Latent Space (AutoEncoder)



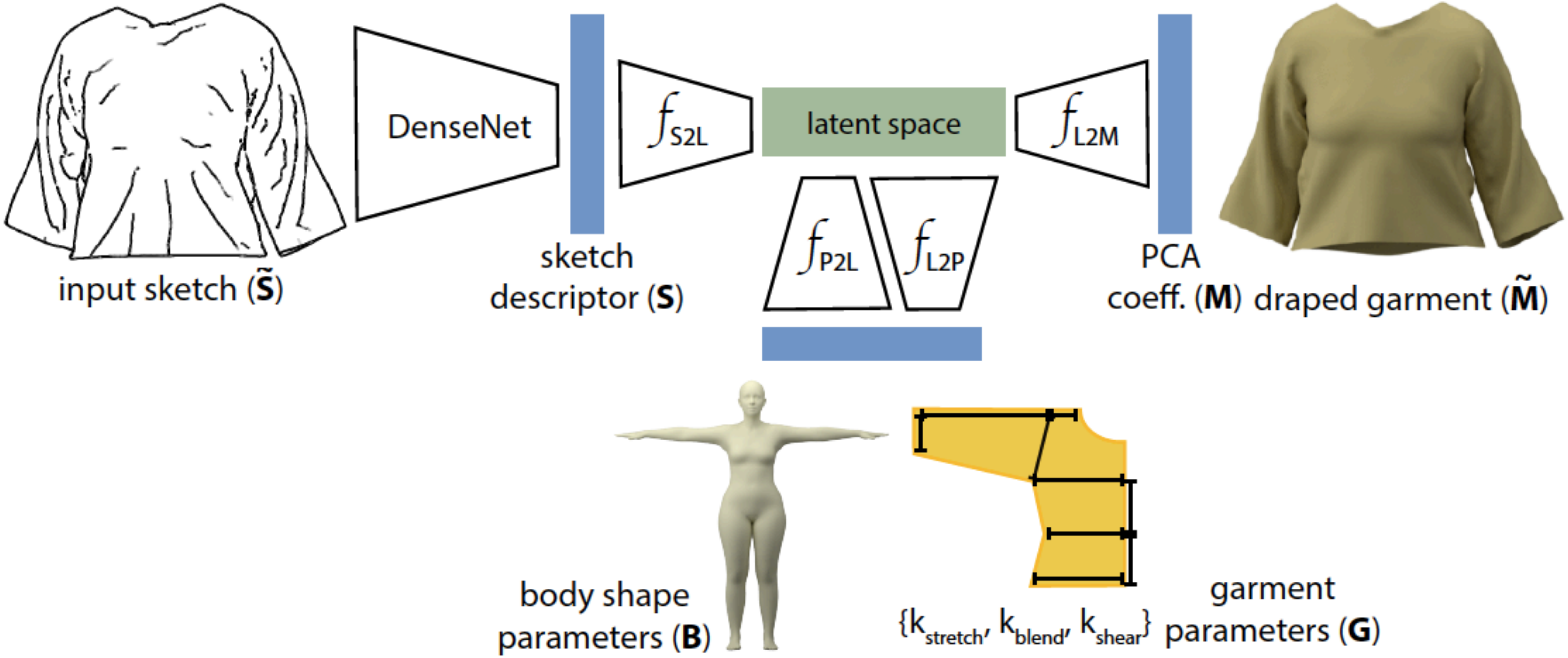
Learning a Shared Latent Space (3-way AutoEncoder)



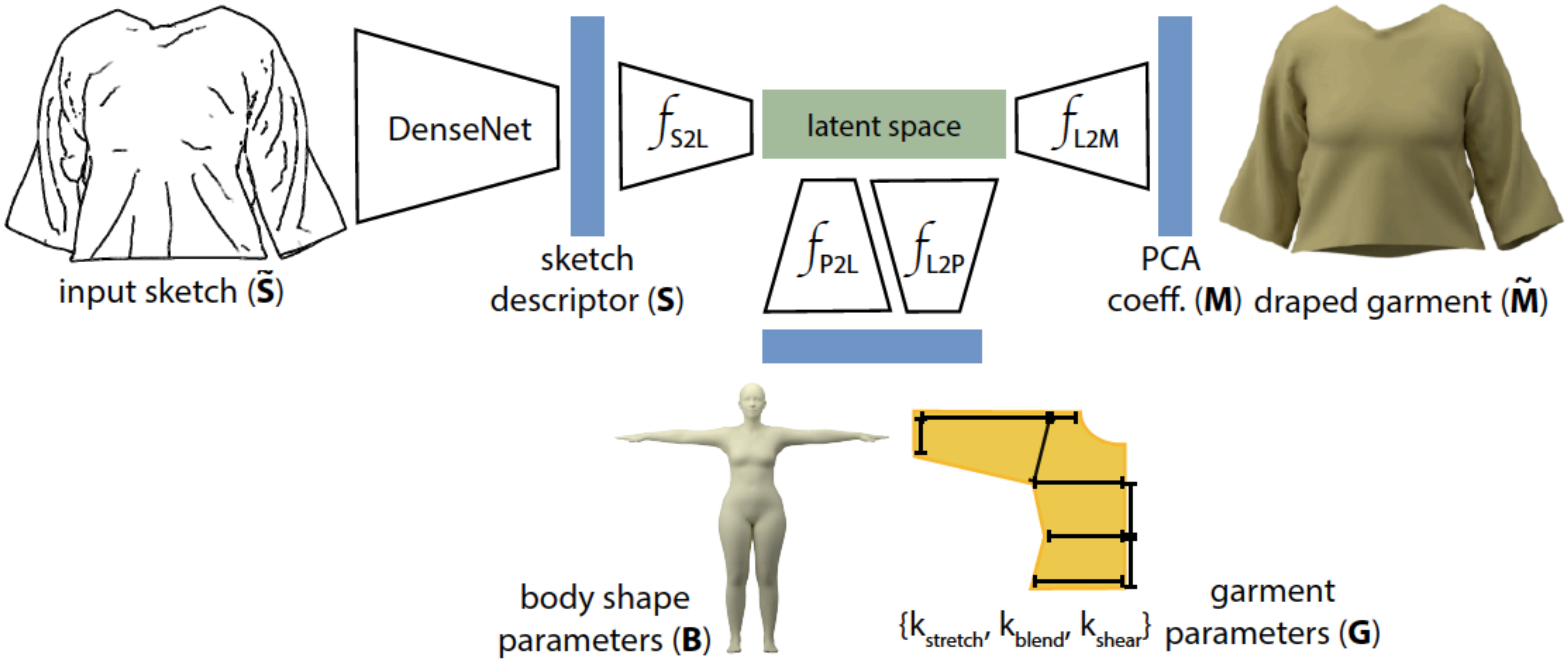
Learning a Shared Latent Space (3-way AutoEncoder)



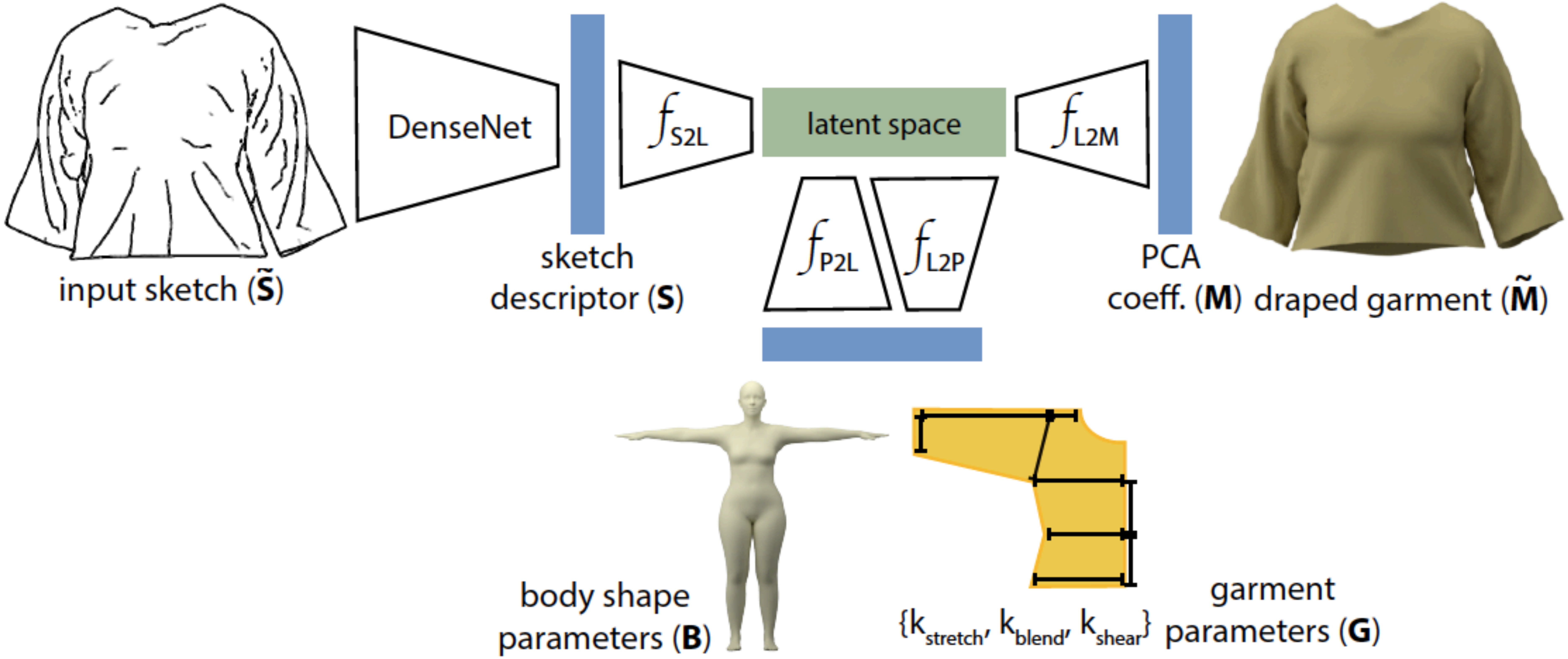
Learning a Shared Latent Space (3-way AutoEncoder)



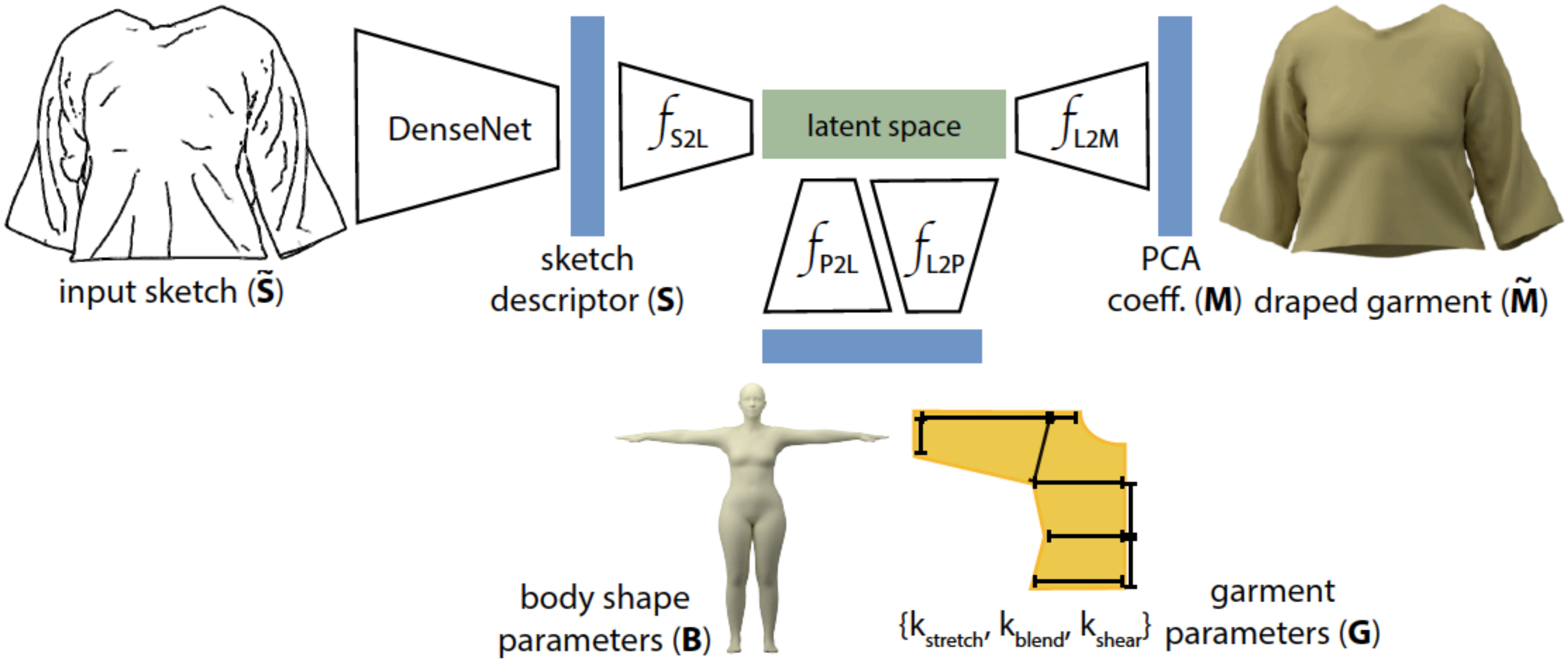
Loss Function Terms



Loss Function Terms

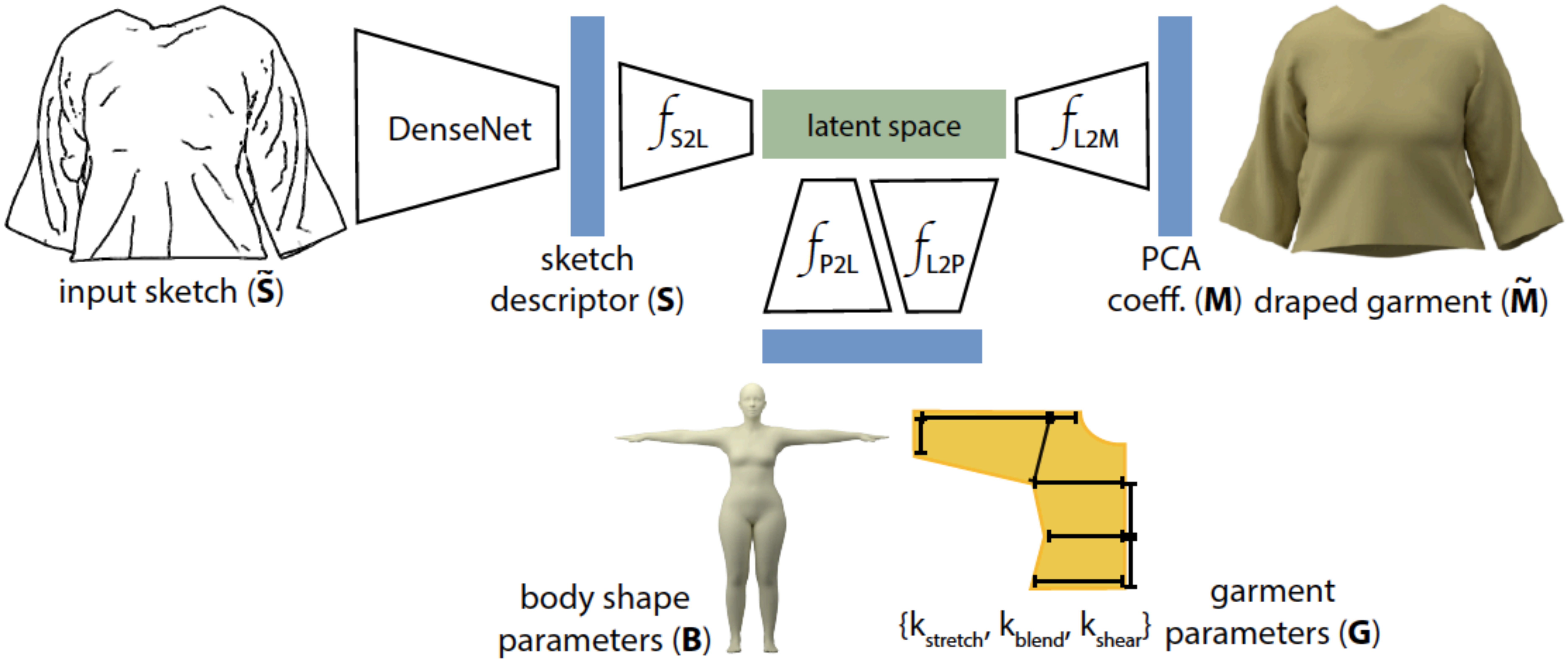


Loss Function Terms



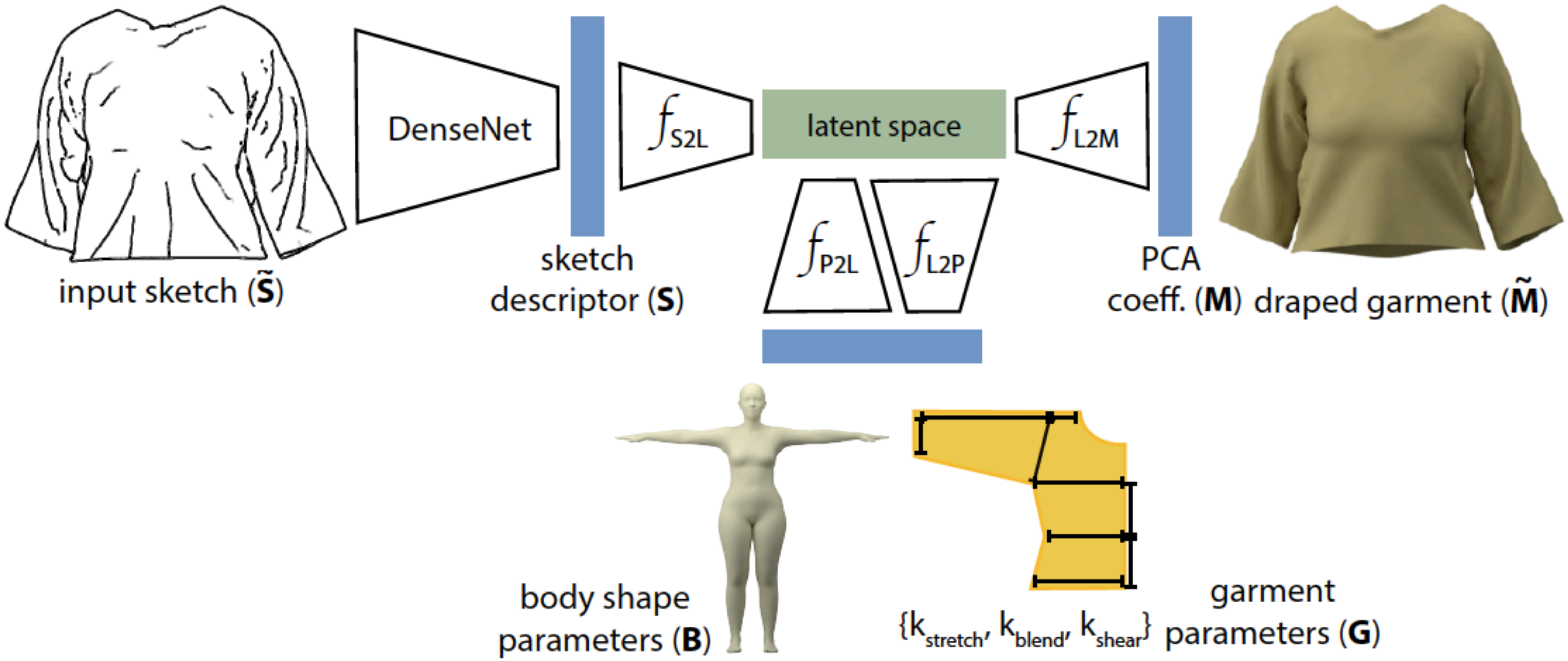
$$\|M - f_{L2M}(f_{S2L}(S))\|_2$$

Loss Function Terms



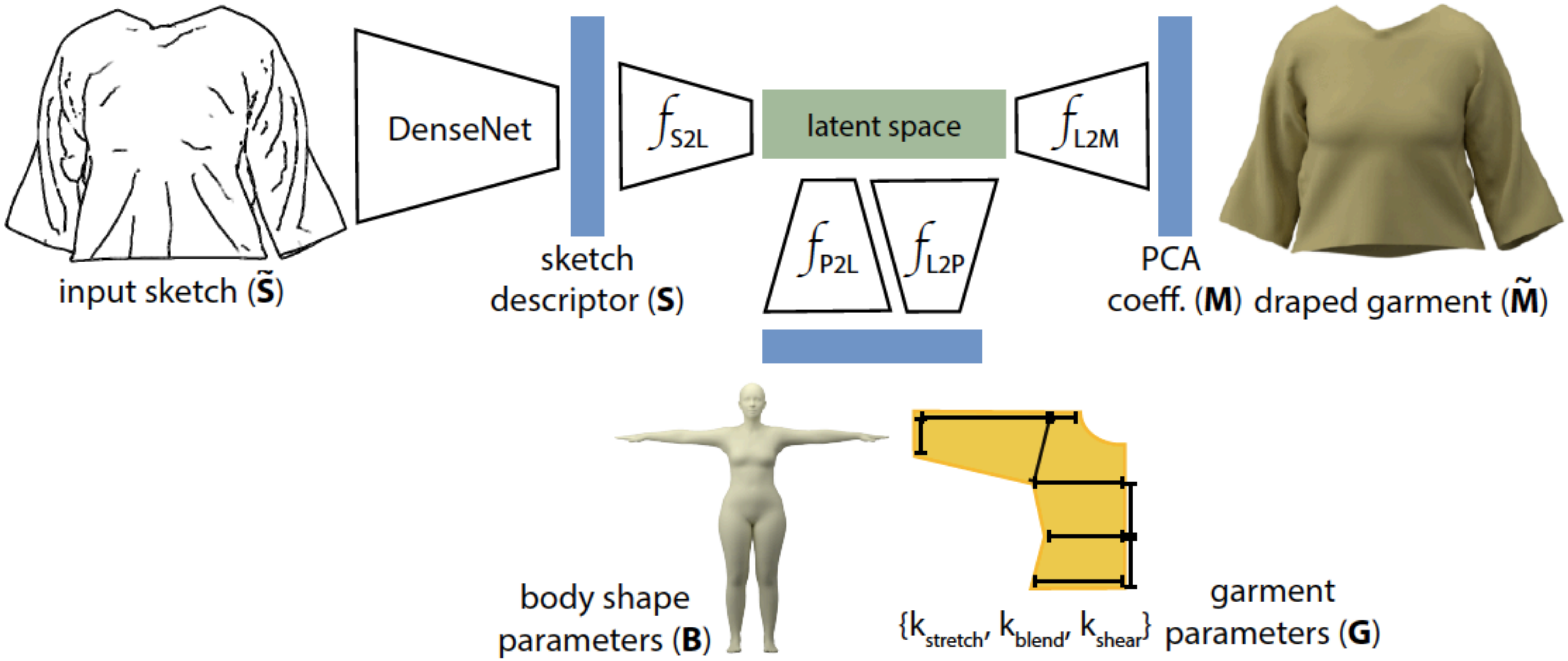
$$\|M - f_{L2M}(f_{S2L}(S))\|_2 \quad \|P - f_{L2P}(f_{S2L}(S))\|_2$$

Loss Function Terms



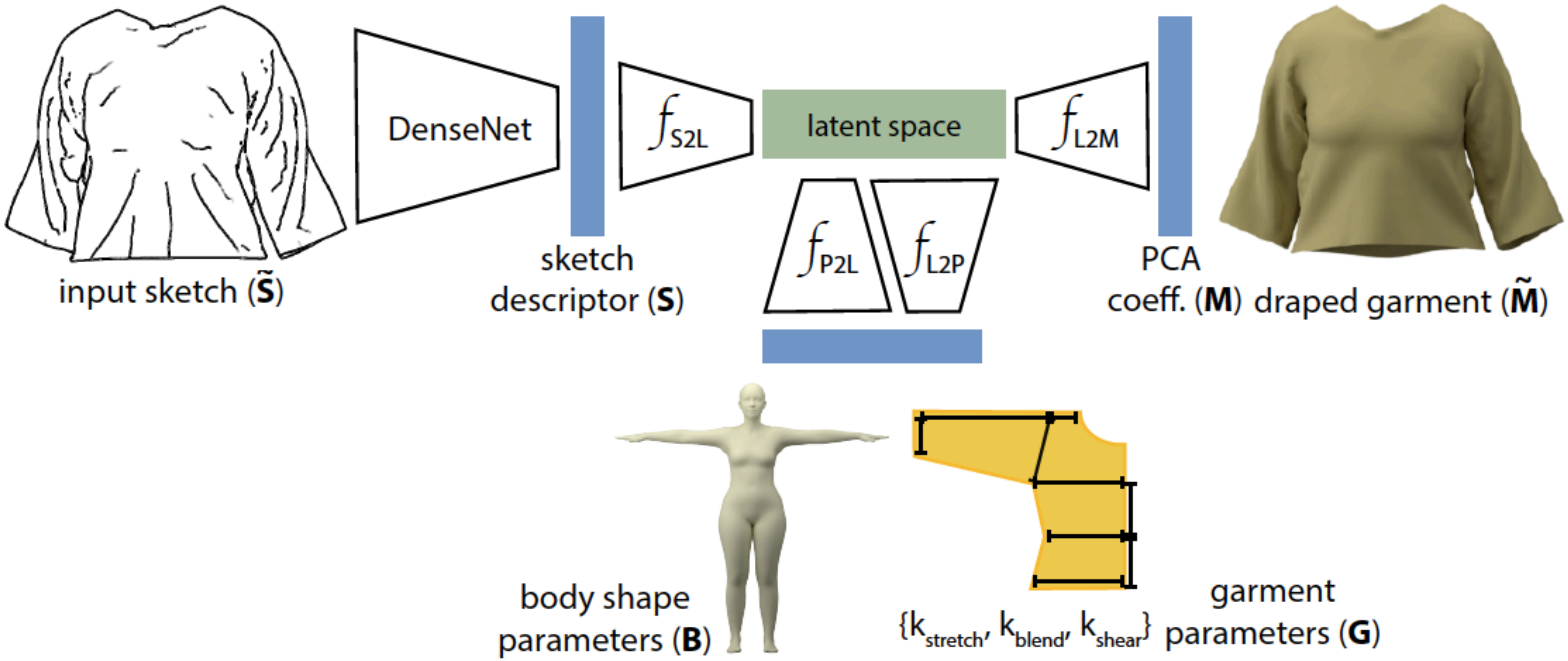
$$\|M - f_{L2M}(f_{S2L}(S))\|_2 \quad \|P - f_{L2P}(f_{S2L}(S))\|_2 \quad \|M - f_{L2M}(f_{P2L}(P))\|_2$$

Loss Function Terms



$$\|M - f_{L2M}(f_{S2L}(S))\|_2 \quad \|P - f_{L2P}(f_{S2L}(S))\|_2 \quad \|M - f_{L2M}(f_{P2L}(P))\|_2 \quad \|P - f_{L2P}(f_{P2L}(P))\|_2$$

Loss Function Terms

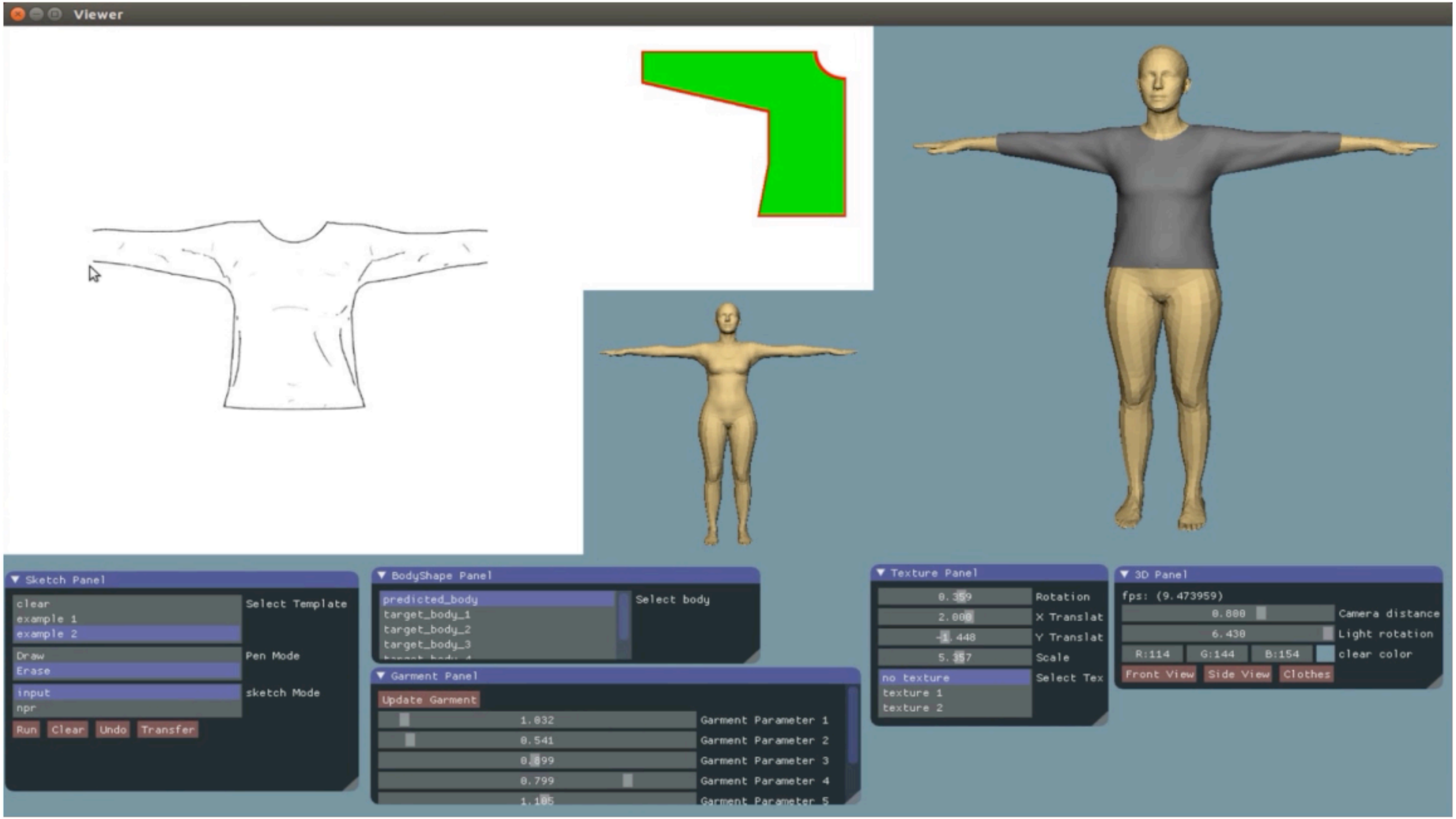


$$\mathscr{L}(\mathbf{P}, \mathbf{M}, \mathbf{S}) =$$

$$||M - f_{L2M}(f_{S2L}(S))||_2 + ||P - f_{L2P}(f_{S2L}(S))||_2 + ||M - f_{L2M}(f_{P2L}(P))||_2 + ||P - f_{L2P}(f_{P2L}(P))||_2$$

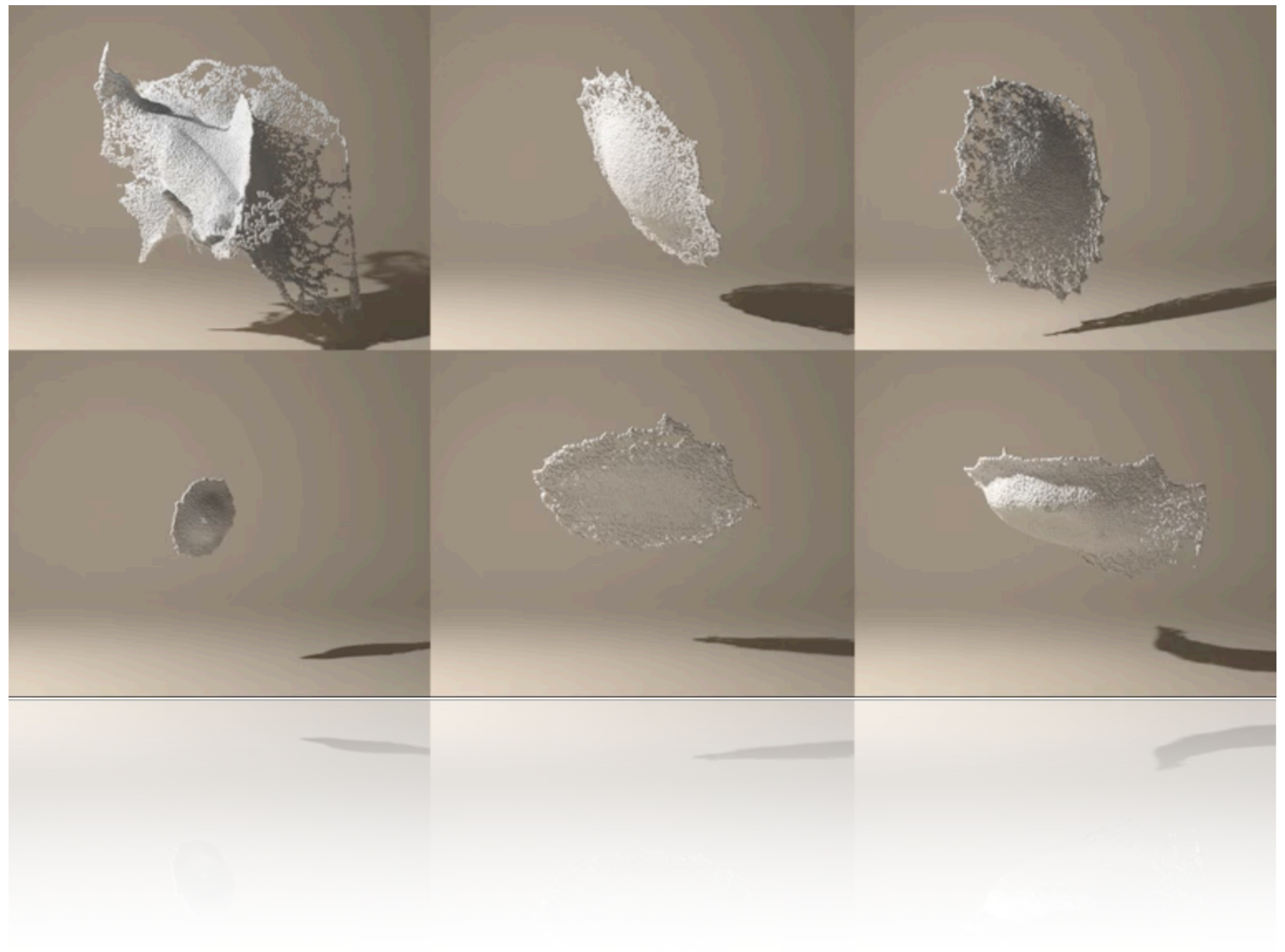
Sketch editing:

x4



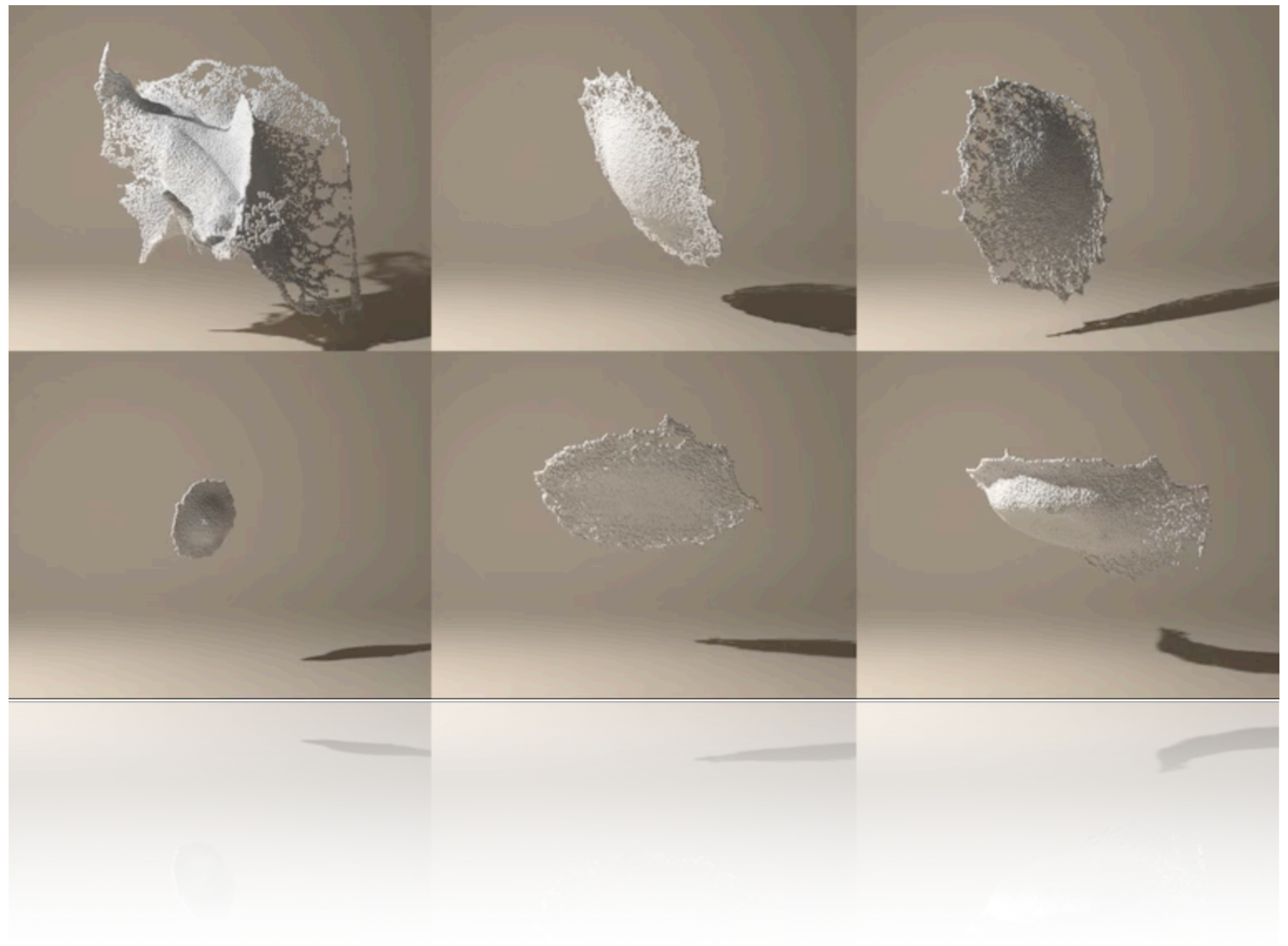
Recap

- Checklist for solving PDEs with DL:



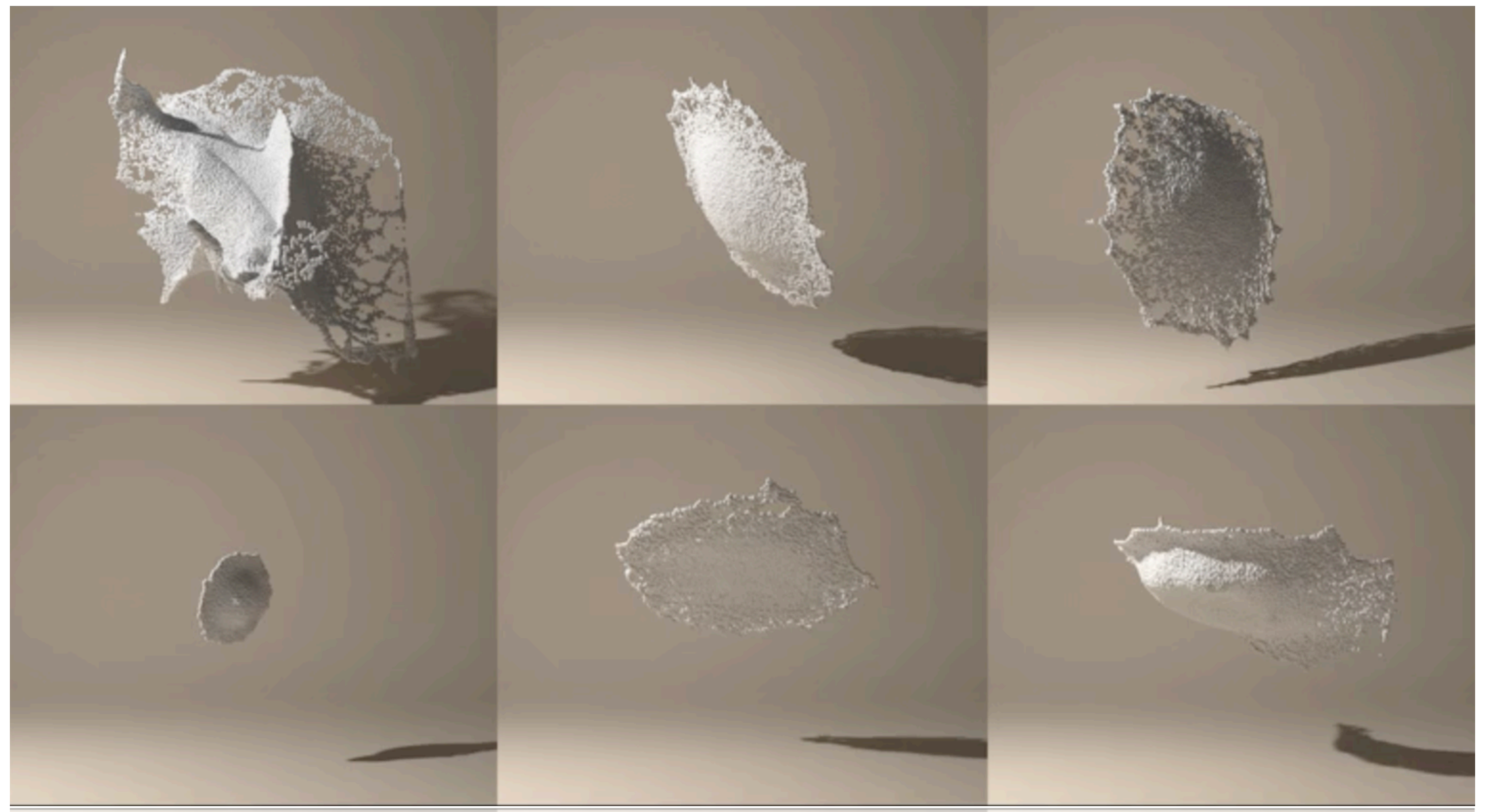
Recap

- Checklist for solving PDEs with DL:
  - ✓ Model? (Typically given)

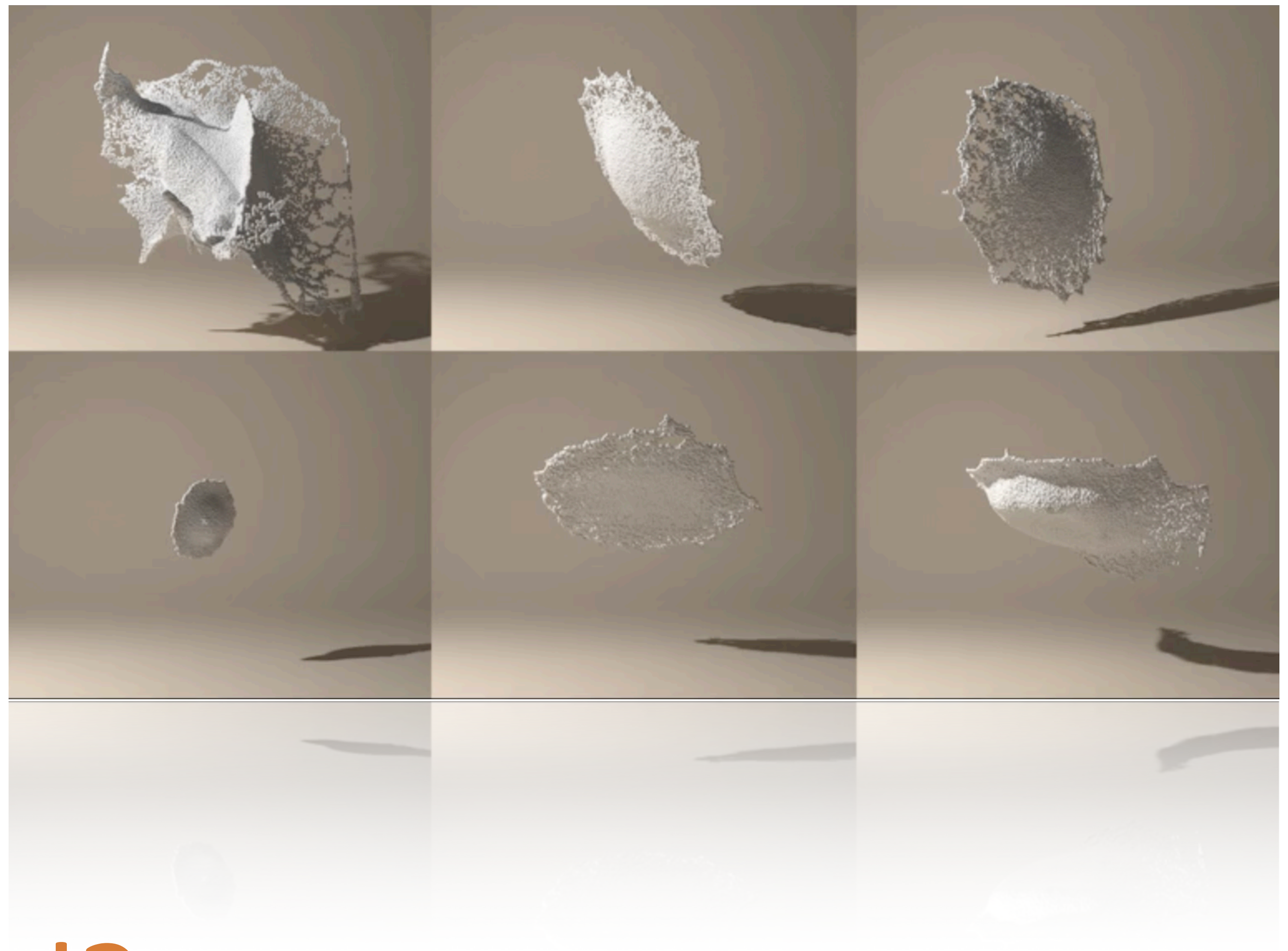


Recap

- Checklist for solving PDEs with DL:
  - ✓ Model? (Typically given)
  - ✓ Data? Can enough training data be generated?

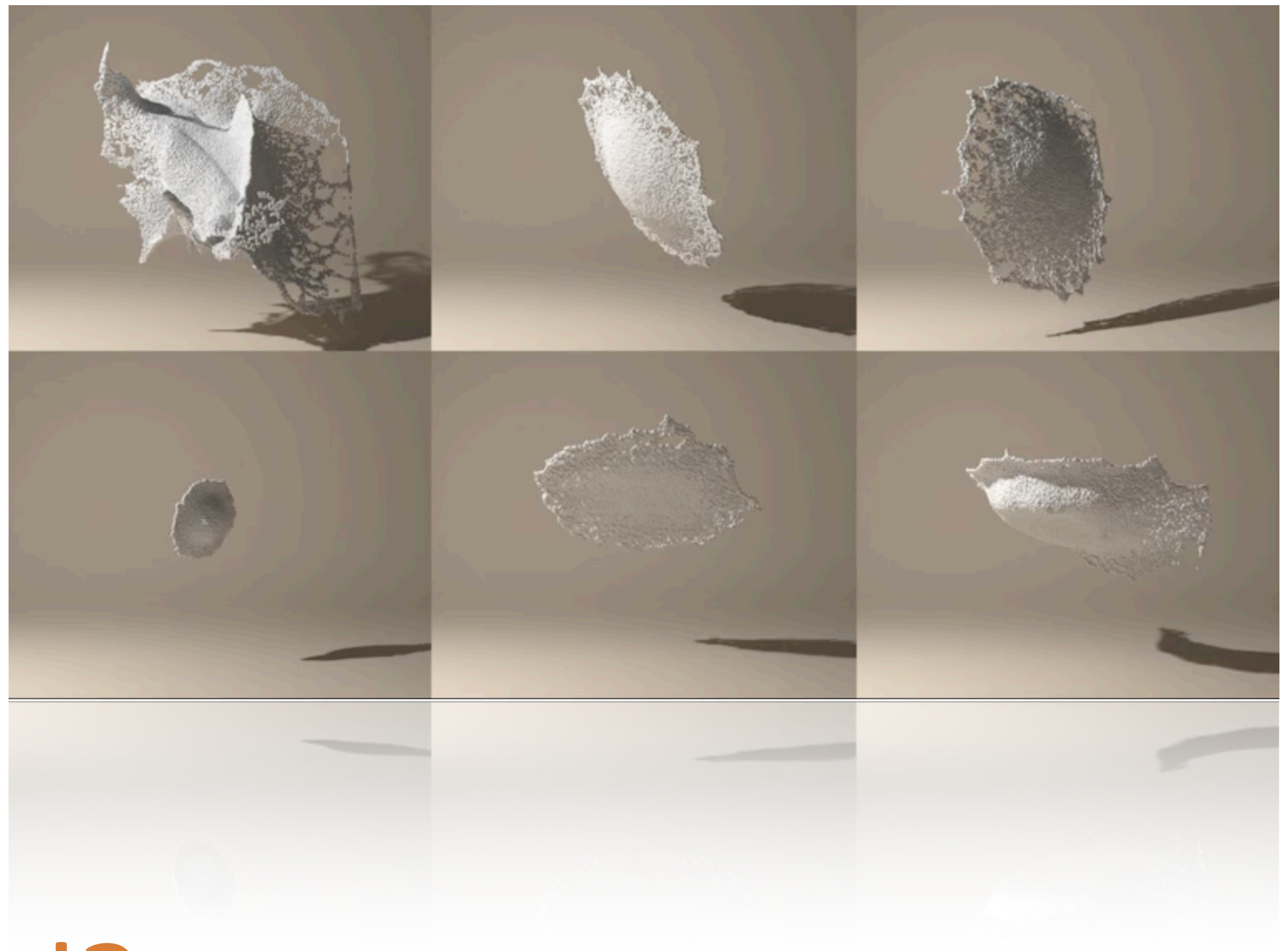


Recap



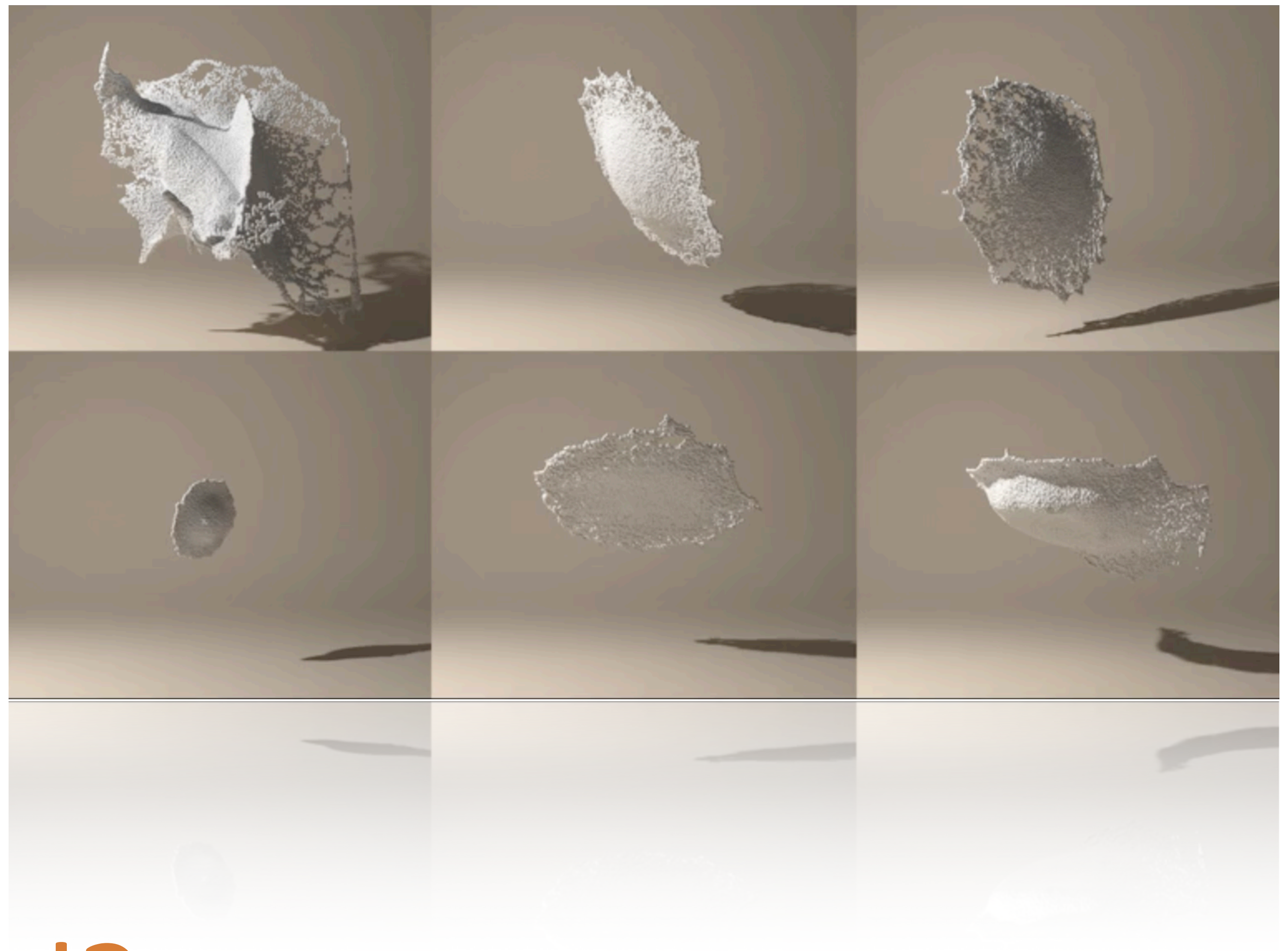
- Checklist for solving PDEs with DL:
  - ✓ Model? (Typically given)
  - ✓ Data? Can enough training data be generated?
  - ✓ Which NN Architecture?

Recap



- 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?

Recap



- 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.?

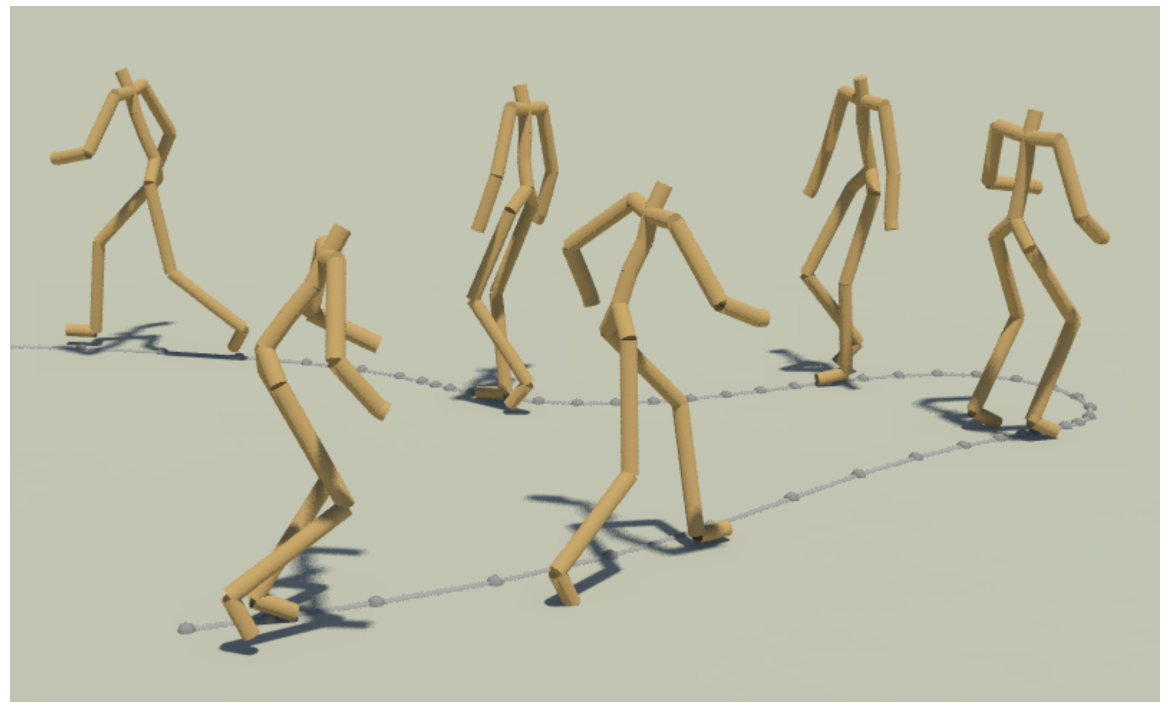
Character Animation

- Learn controllers for character rigs
- Powerful and natural

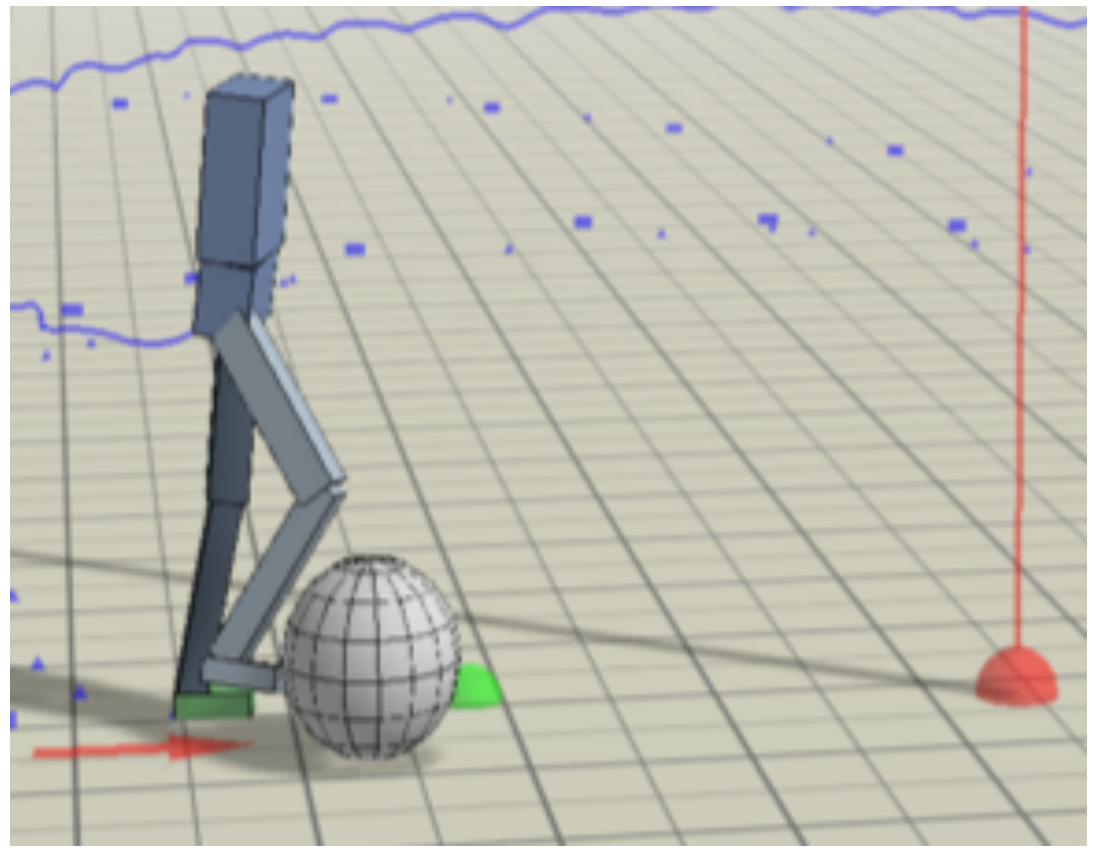
Character Animation

- Learn controllers for character rigs
- Powerful and natural

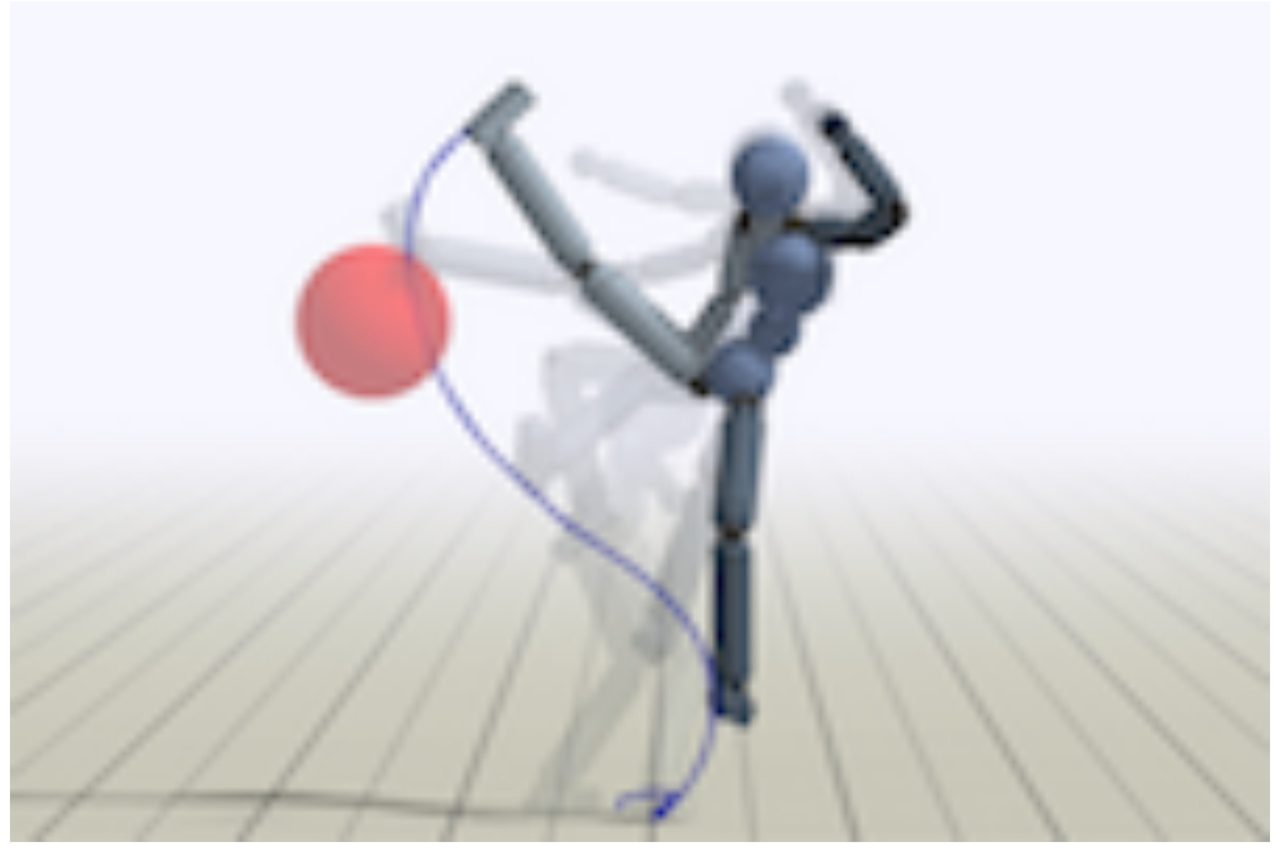
[Mode-Adaptive Neural Networks for Quadruped Motion Control, SIGGRAPH 2018]



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



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

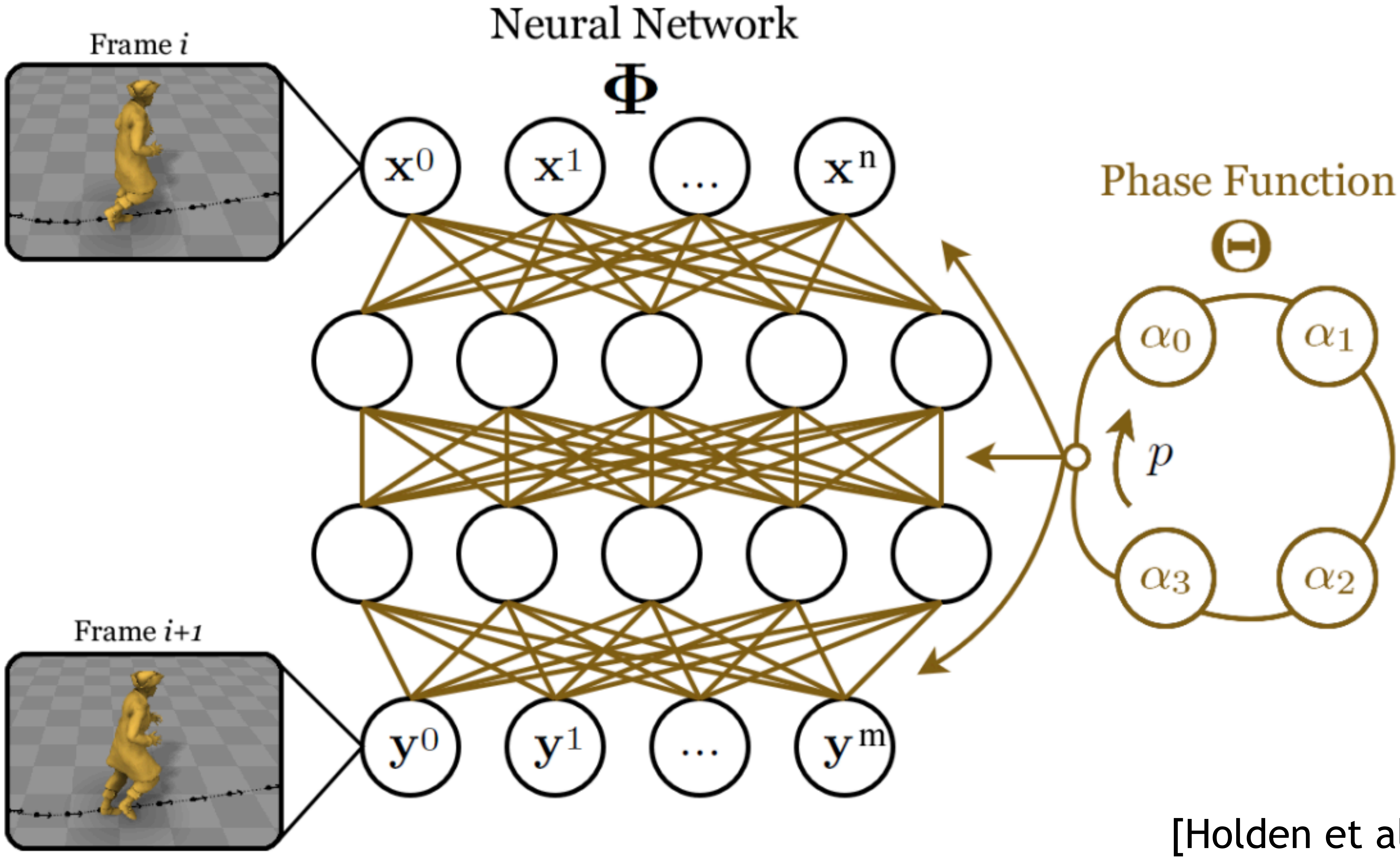


[DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills, SIGGRAPH 2018]

Result

Result

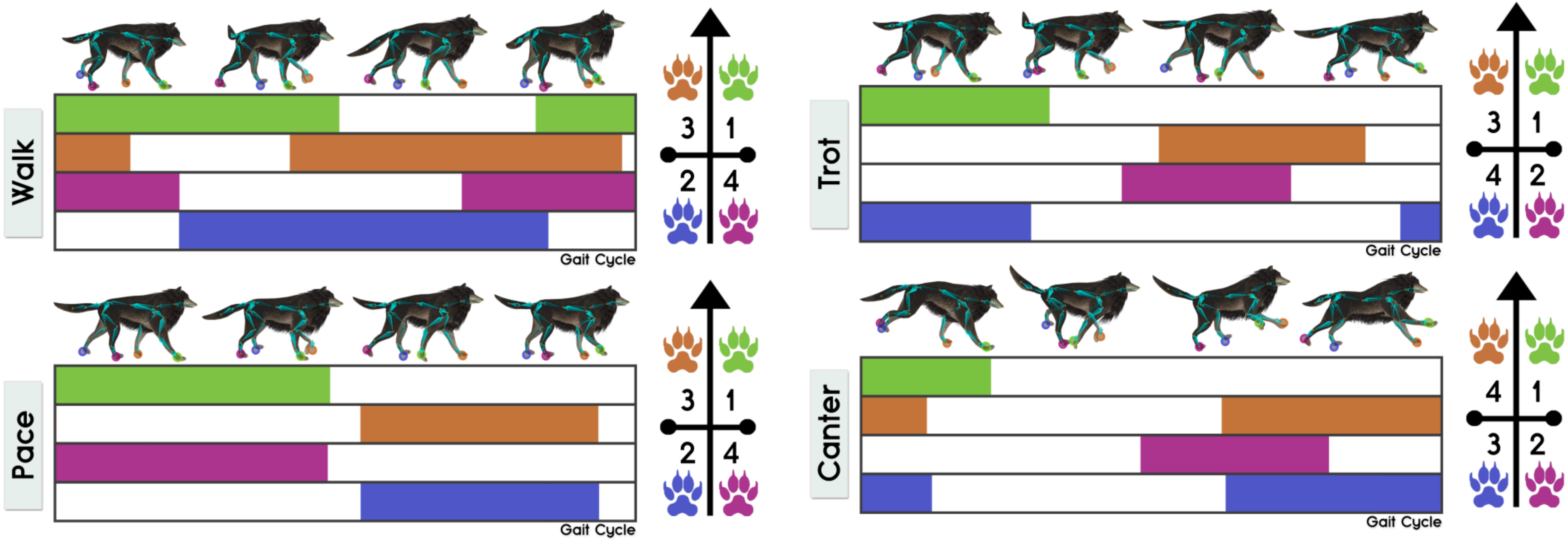
Phase-functioned Neural Network



[Holden et al., Siggraph, 2017]

What about Other Creatures?

Footfall Patterns

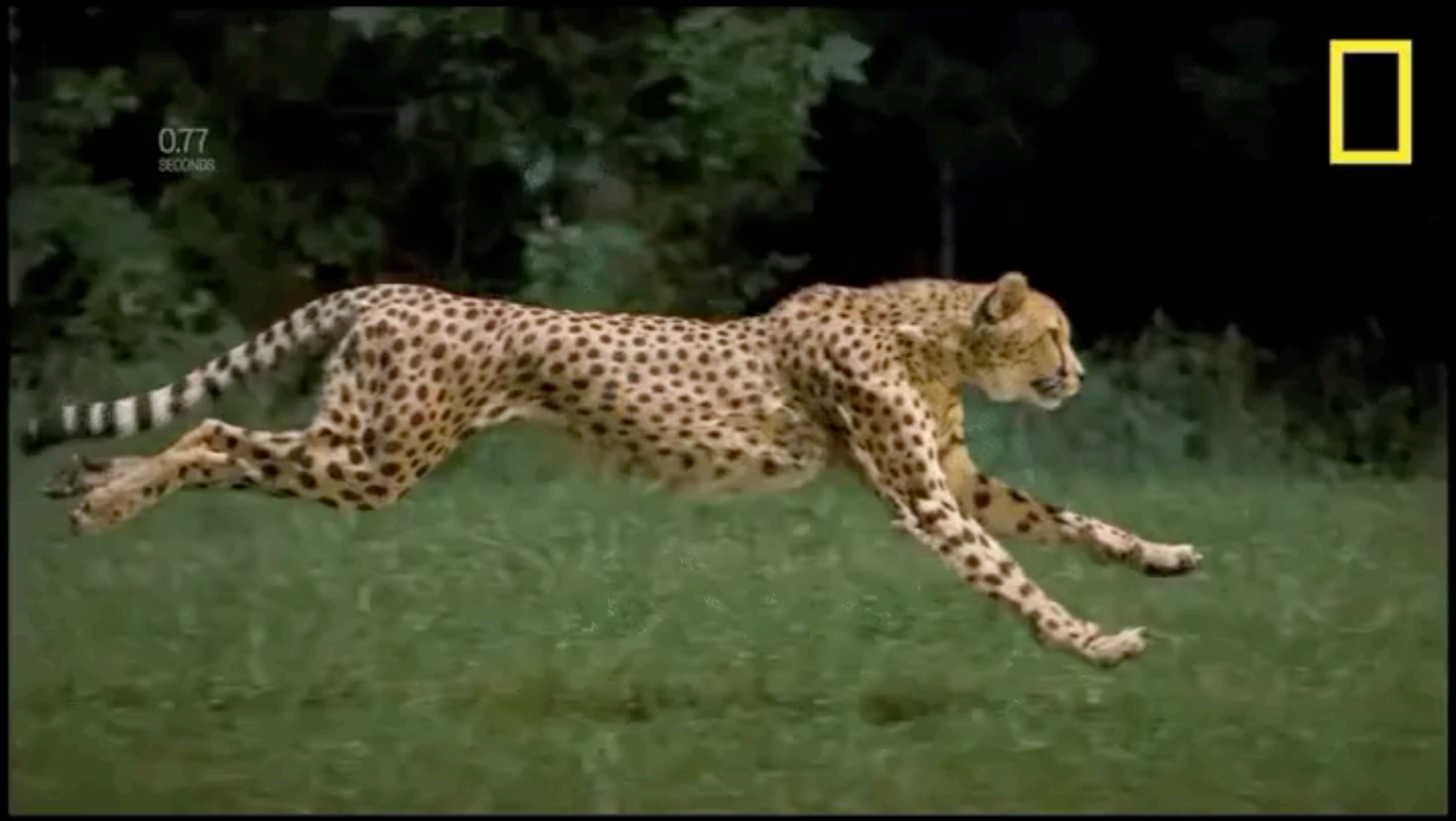


Gating + Motion Update Network


Pace -> Canter -> Walk -> Turn
0%
100%
0%
0%
0%
0%
SMOOTH FOLLOW
CONSTANT VIEW
STATIC
YAW: 0
PITCH: -5
FOV: 1.0

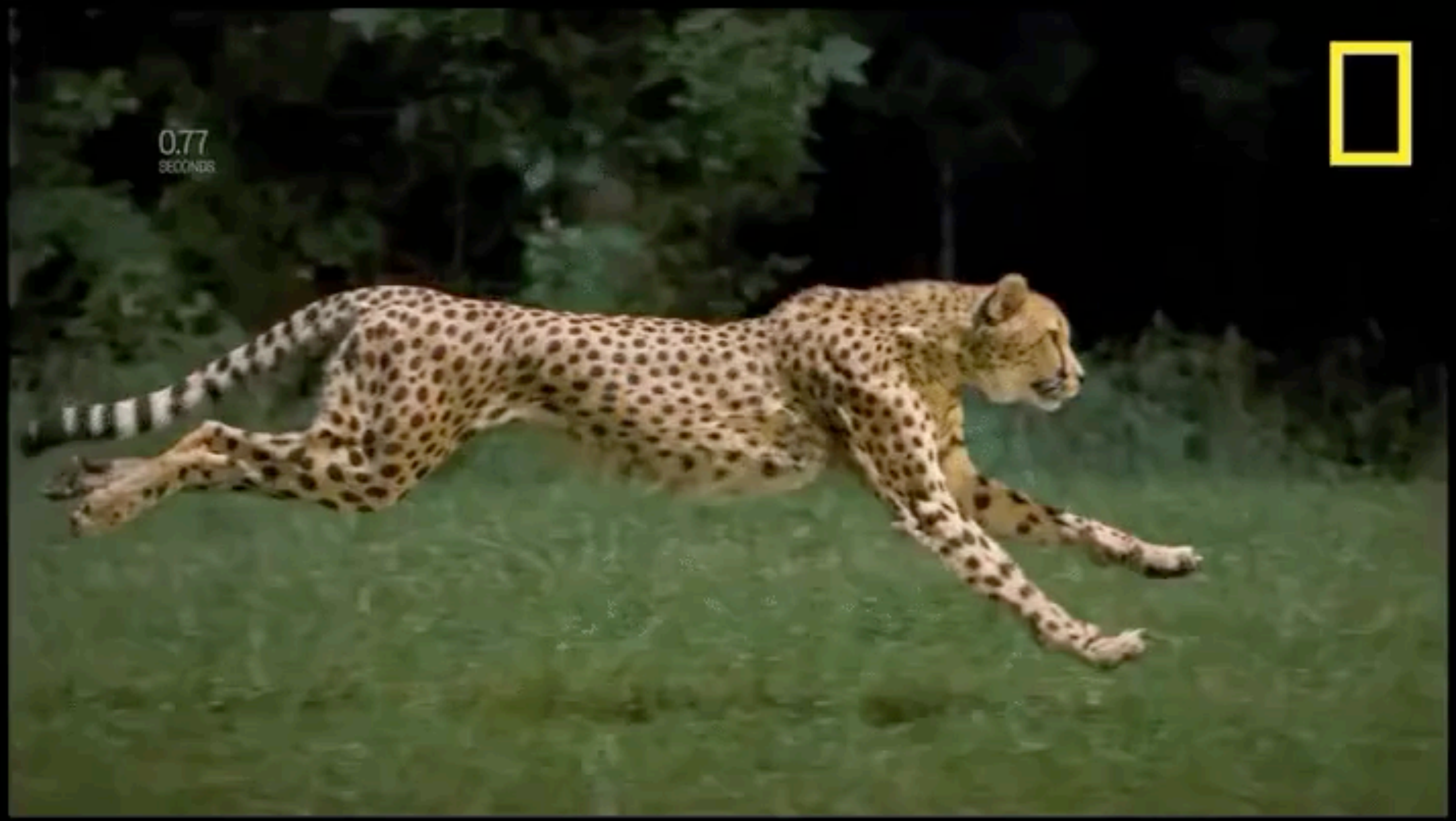

Pace -> Canter ->
Walk -> Turn
0%
100%
0%
0%
0%
0%
SMOOTH FOLLOW
CONSTANT VIEW
STATIC
YAW: 0
PITCH: -5
FOV: 1.0

Dynamic Motion Control



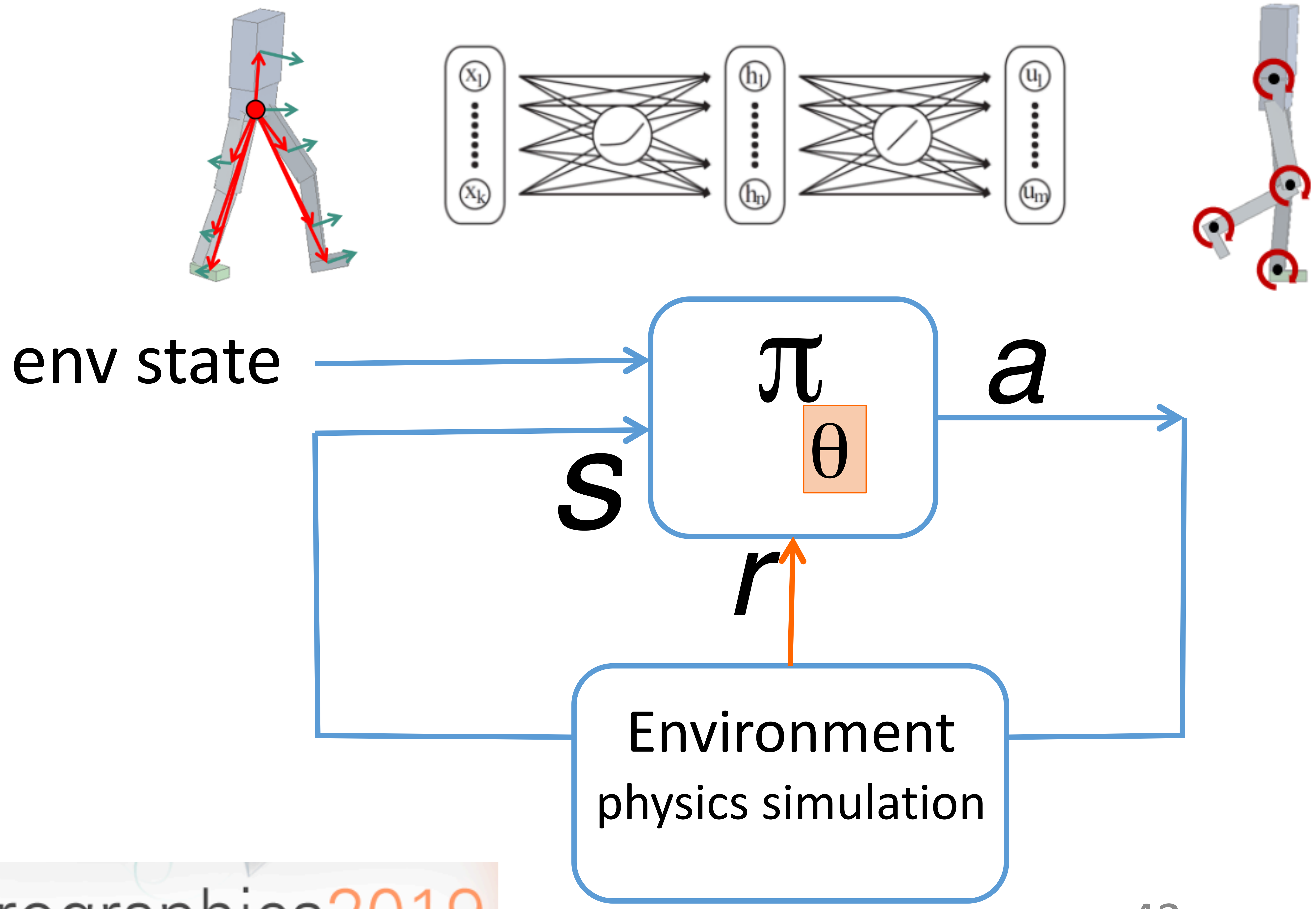
[Many of the following slides thanks to Michiel van de Panne]

Dynamic Motion Control



[Many of the following slides thanks to Michiel van de Panne]

REINFORCEMENT LEARNING FOR LOCOMOTION CONTROL



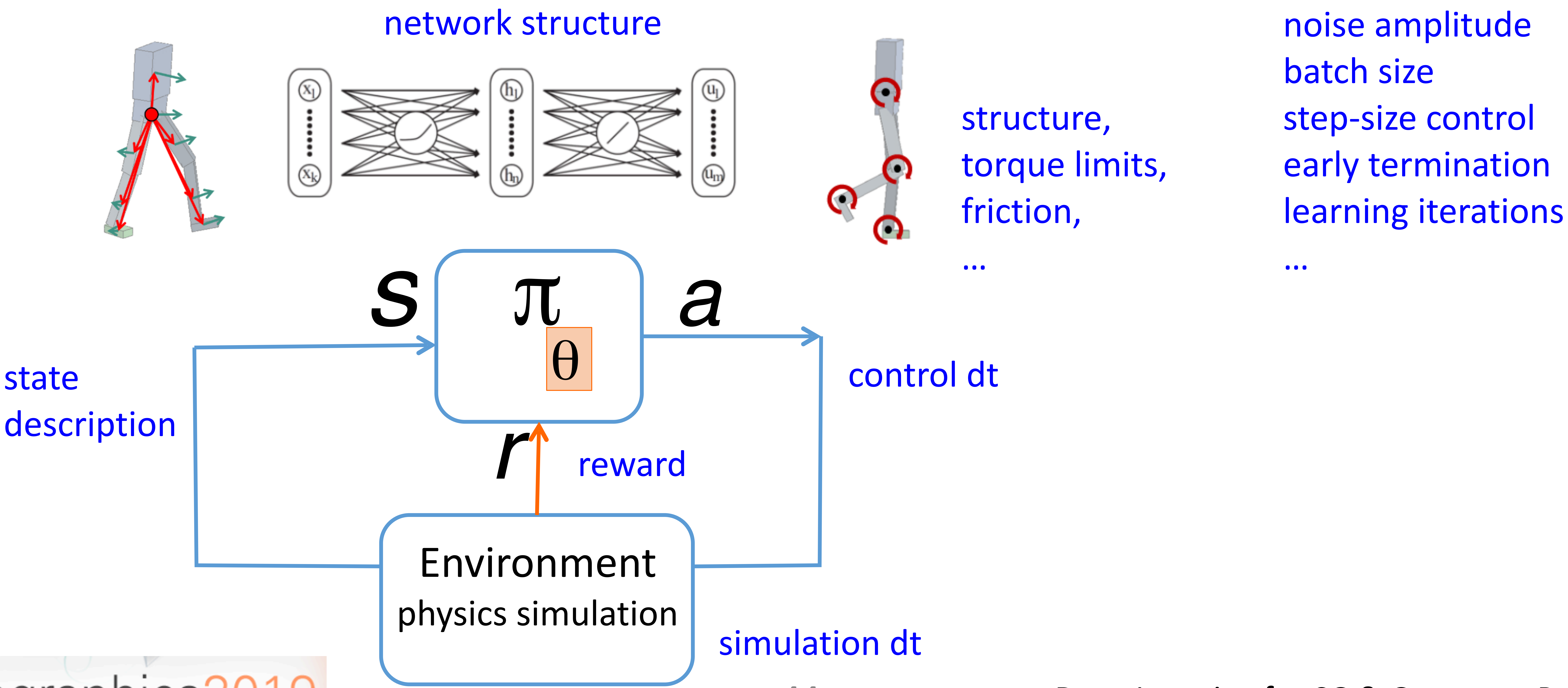
In principle:

- specify rewards
- "train" using RL algorithm

$$\max_{\theta} \ \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

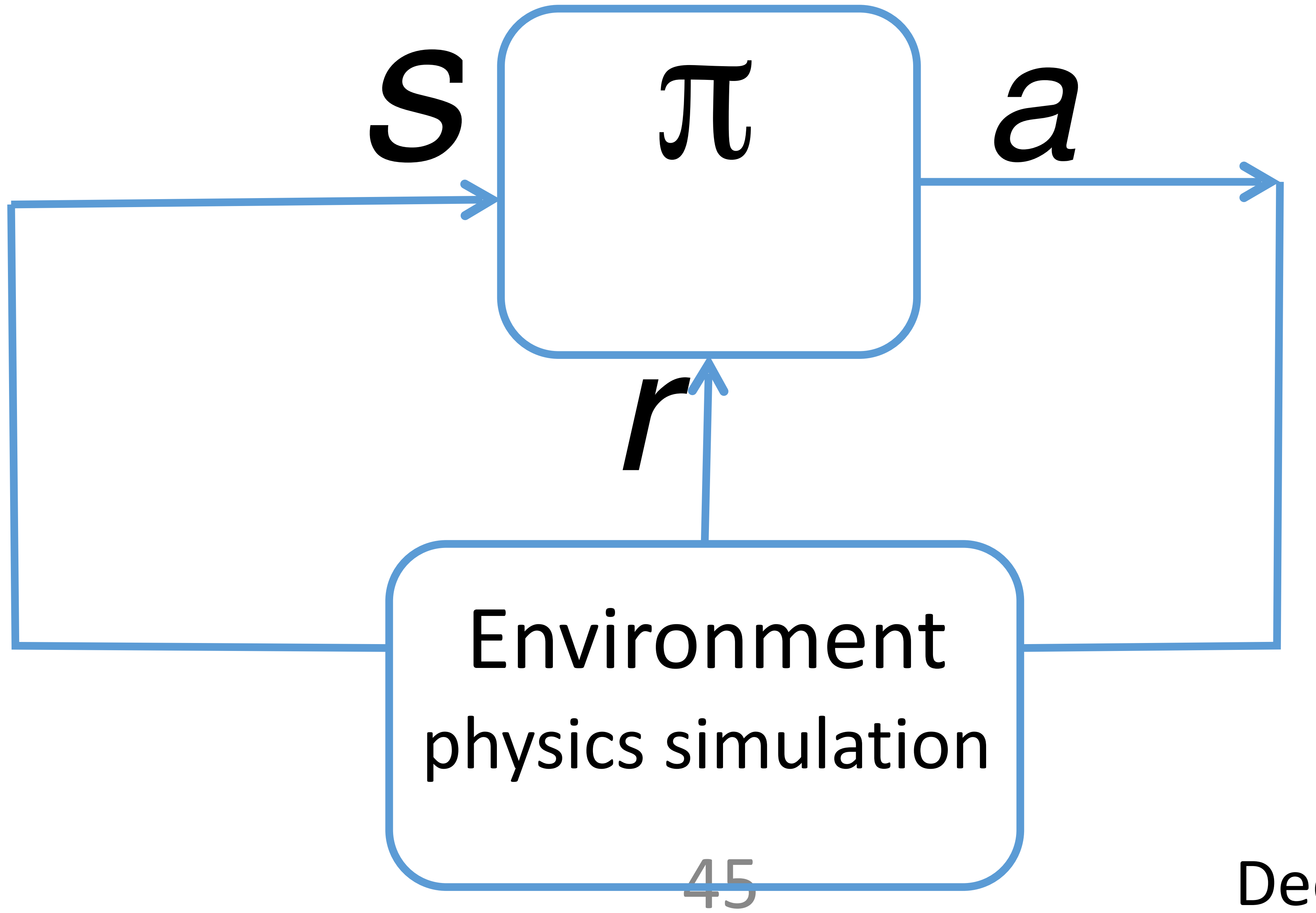
- use the solution

REINFORCEMENT LEARNING



MOTION IMITATION

- DeepLoco: SIGGRAPH 2017
- DeepMimic: SIGGRAPH 2018



DEEPMIMIC: EXAMPLE-GUIDED DEEP REINFORCEMENT LEARNING OF PHYSICS-BASED CHARACTER SKILLS

- Xue Bin Peng, University of California, Berkeley Pieter Abbeel, University of California, Berkeley Sergey Levine, University of California, Berkeley Michiel van de Panne, University of British Columbia

[SIGGRAPH 2018]

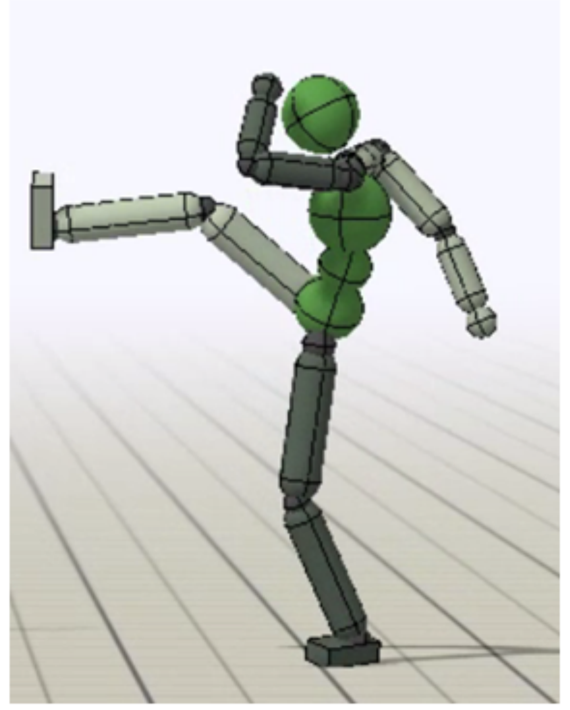
REWARD

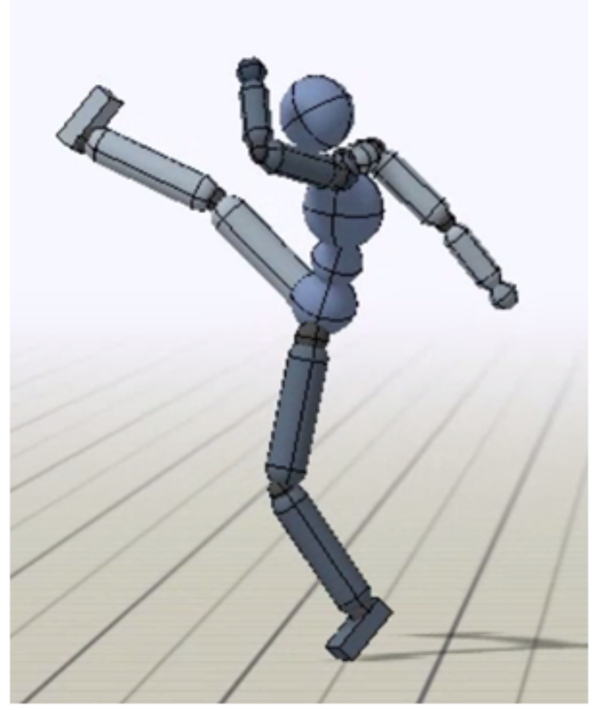
$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

REWARD

$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

Imitation Objective

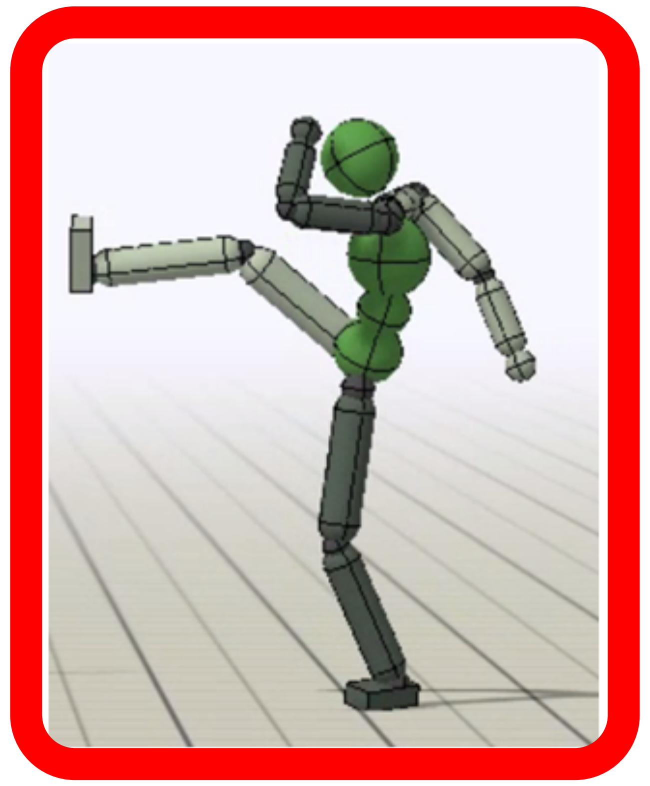
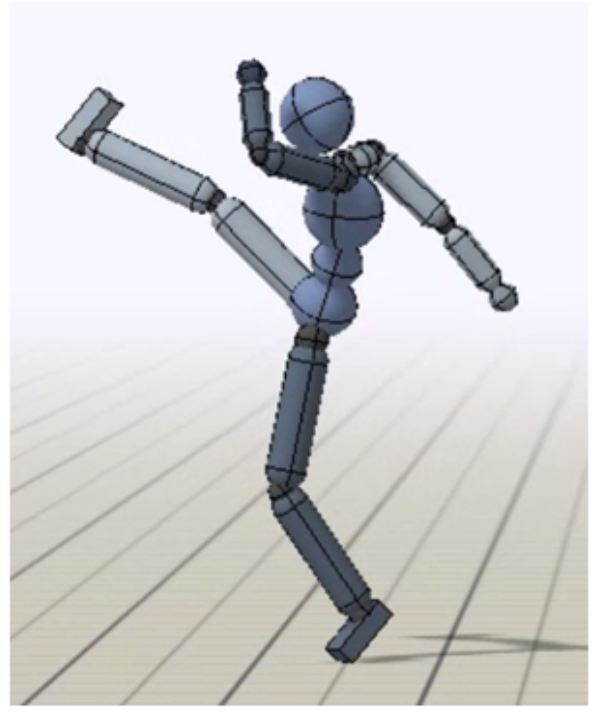




REWARD

$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

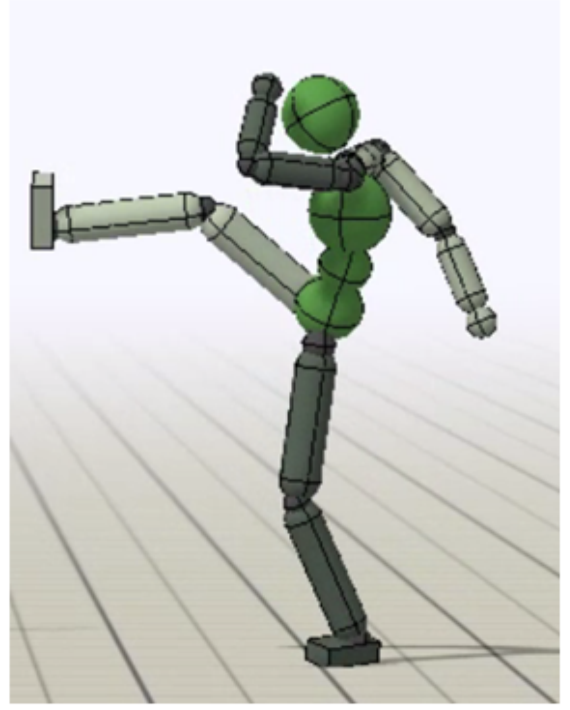
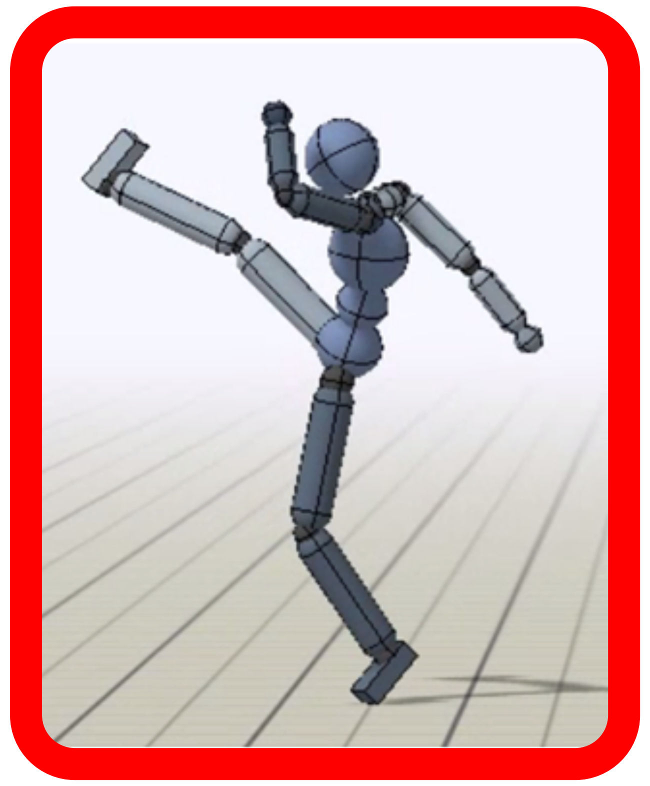
Imitation Objective

$\Big\|$  $-$  $\Big\|$

REWARD

$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

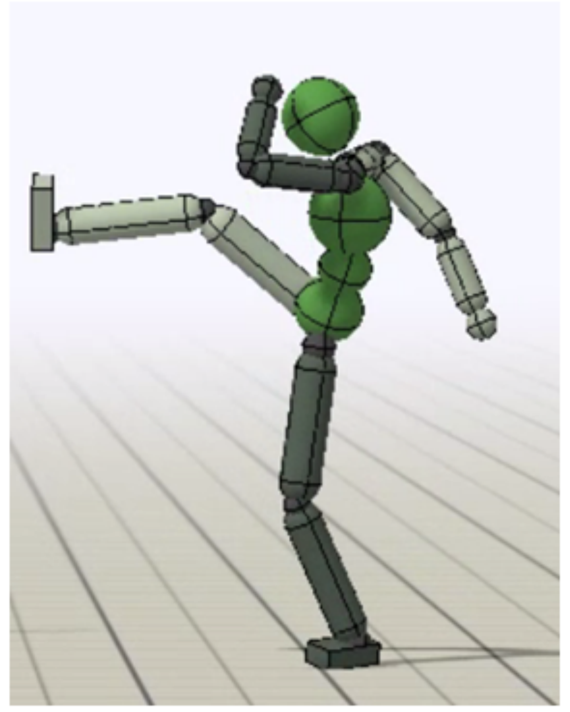
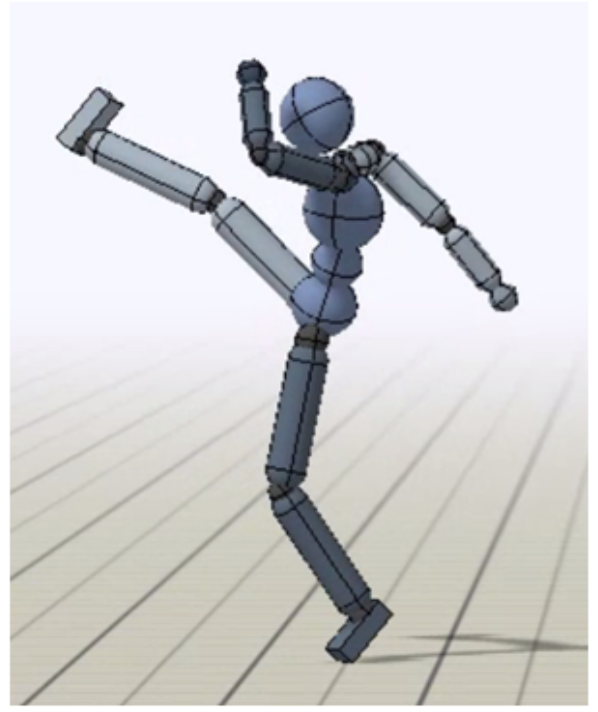
Imitation Objective

$\Big\|$  $-$  $\Big\|$

REWARD

$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

Imitation Objective

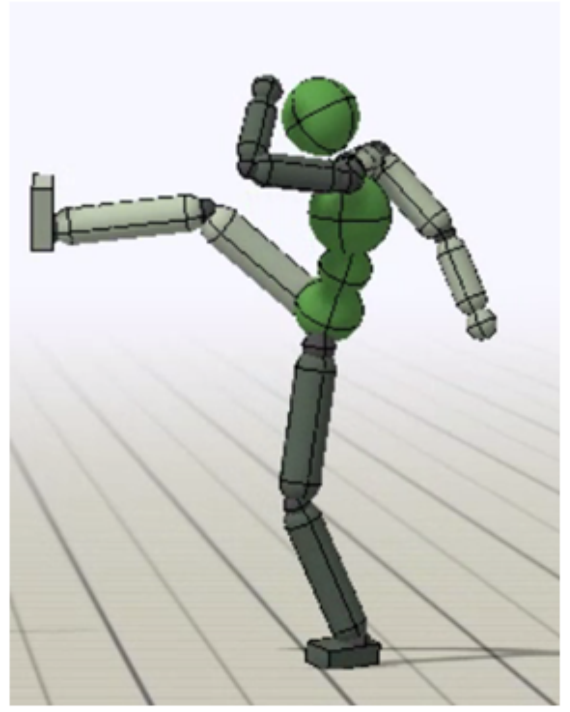
$\Big\|$  $-$  $\Big\|$

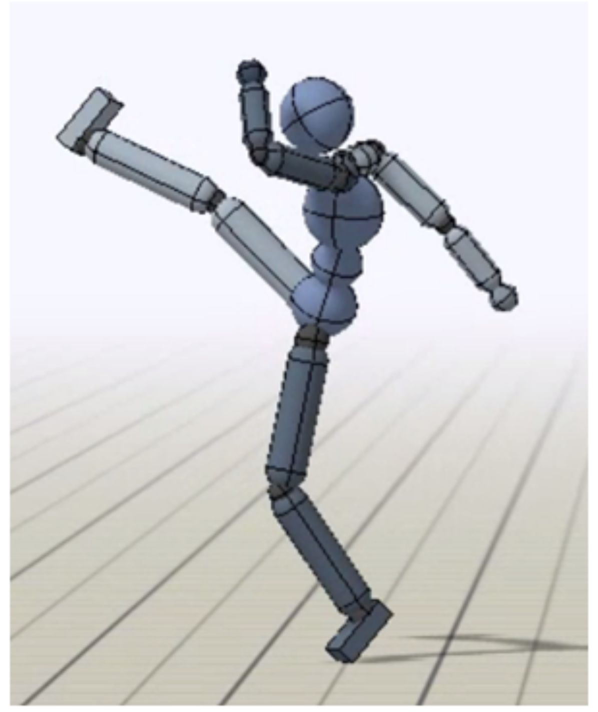
REWARD

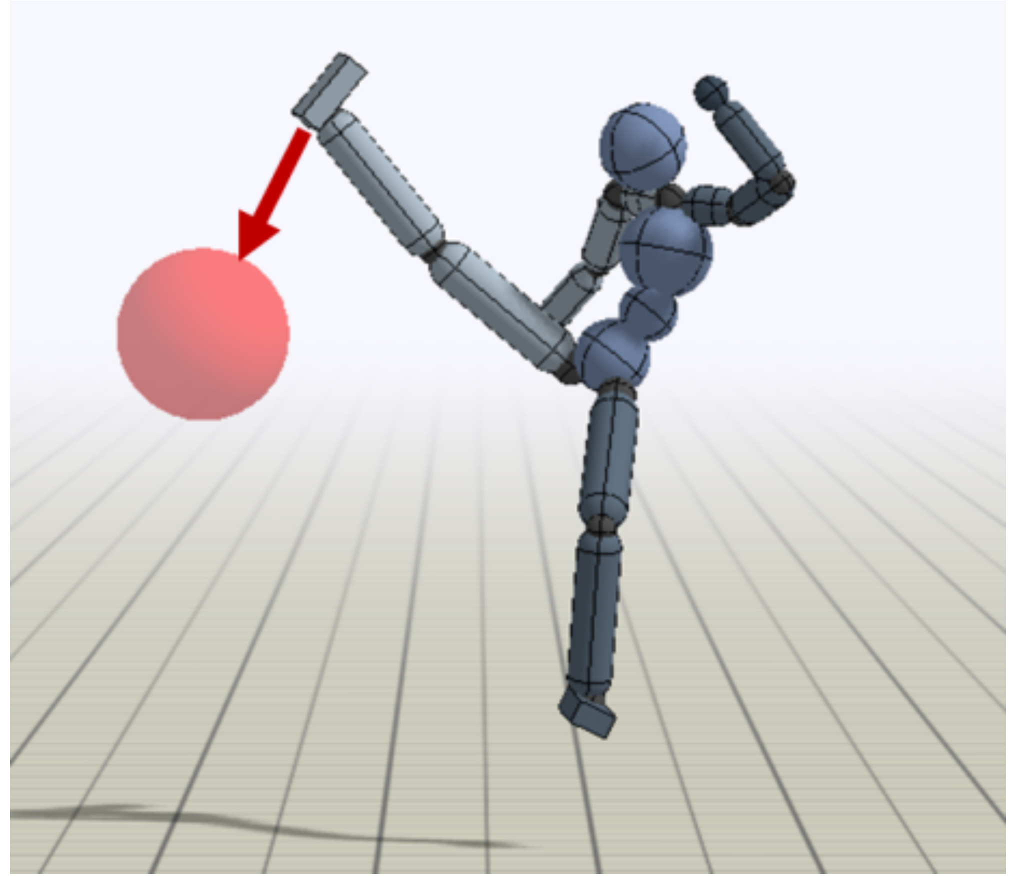
$$r_t = \omega^I r_t^I + \omega^G r_t^G$$

Imitation Objective

Task Objective





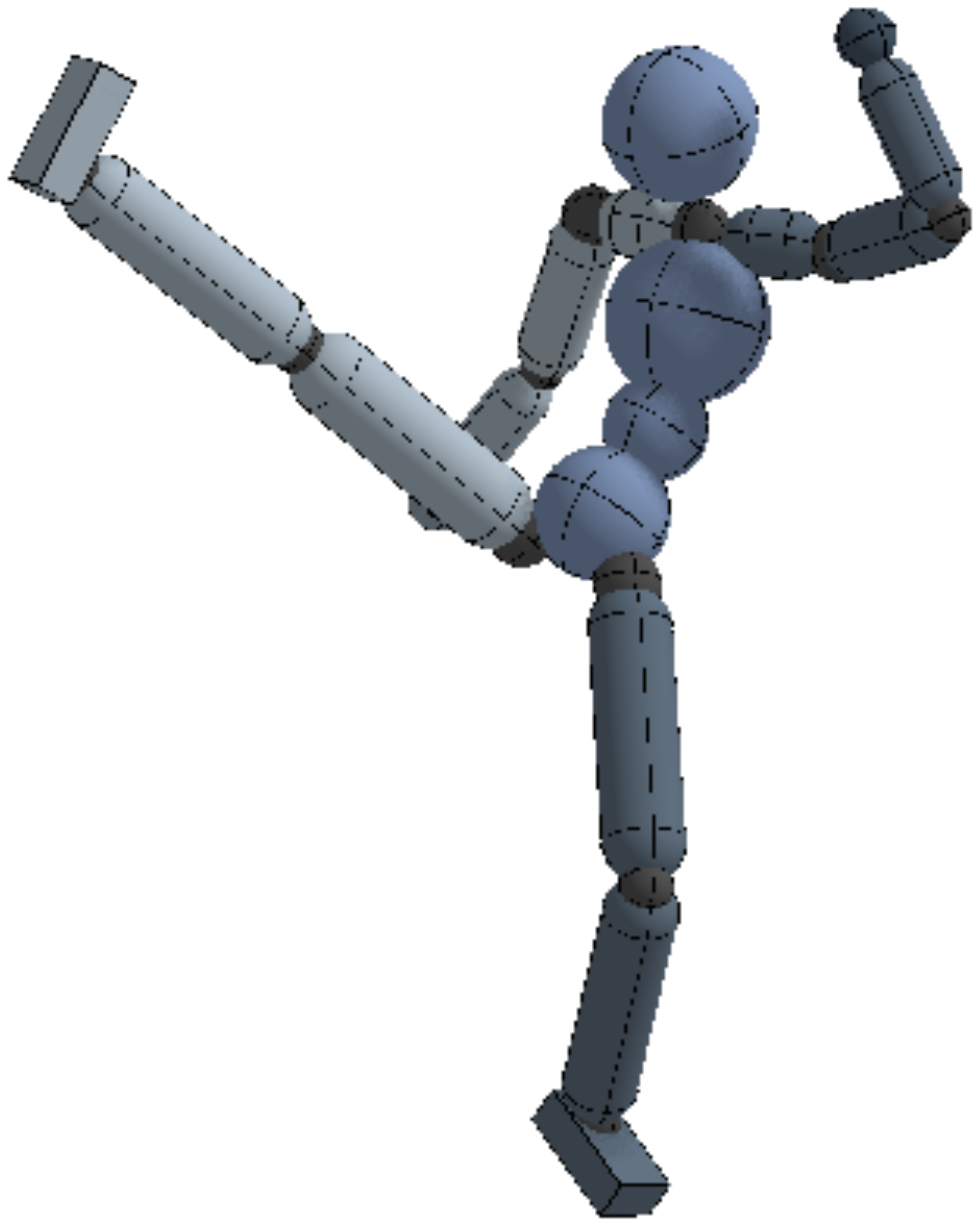


STATE + ACTION

State:

- link positions
- link velocities

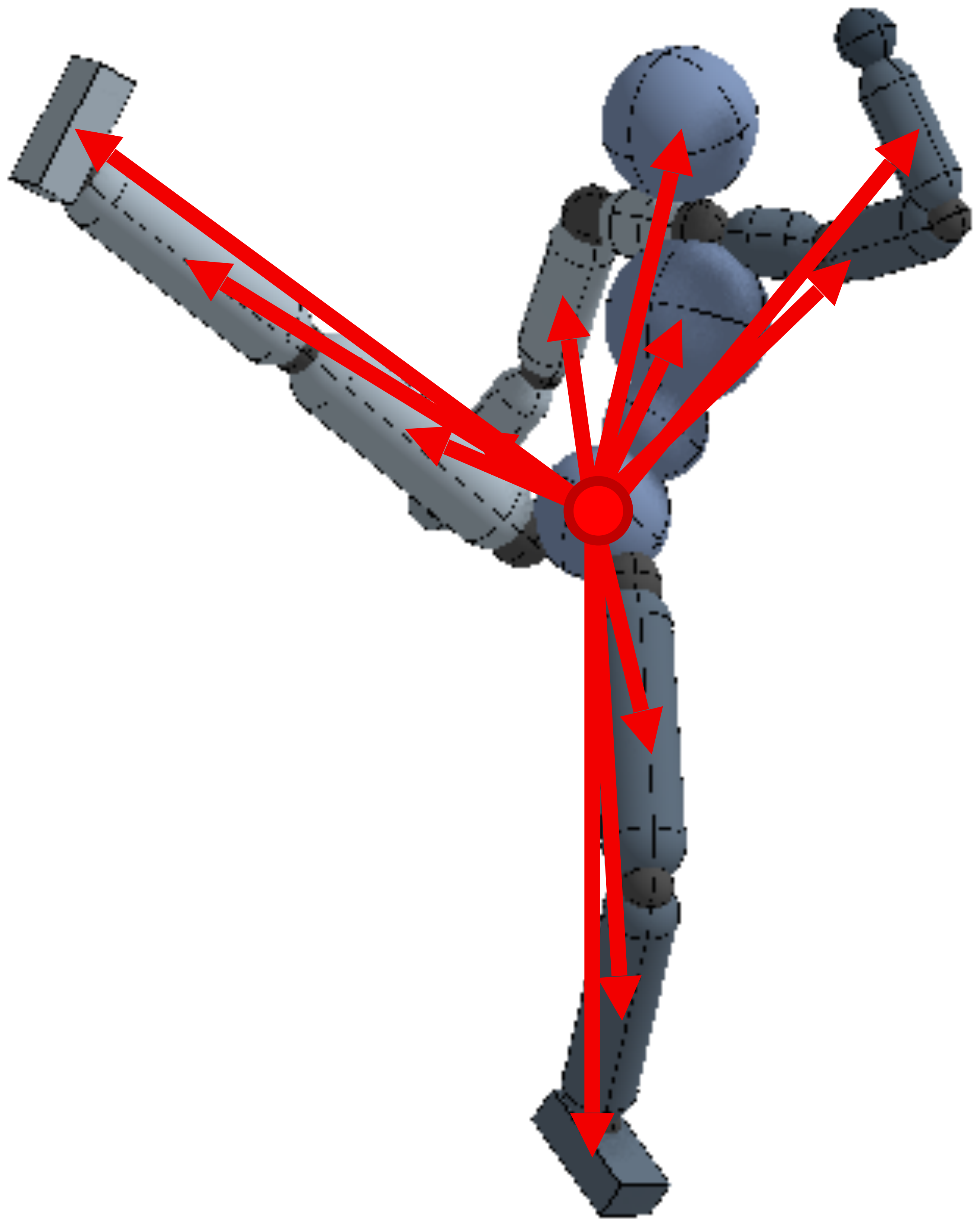
- terrain heights



STATE + ACTION

State: 197 D

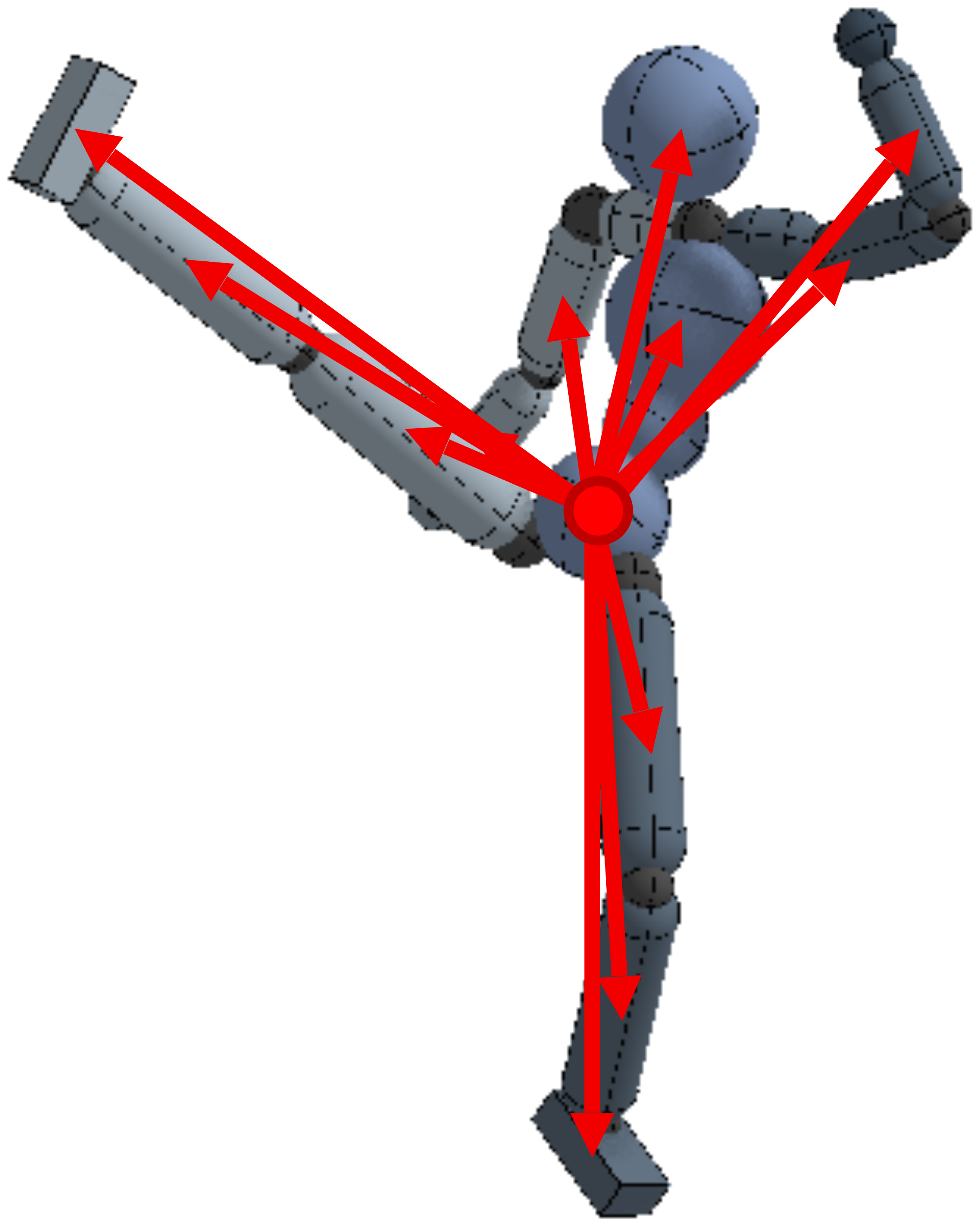
- link positions
- link velocities



STATE + ACTION

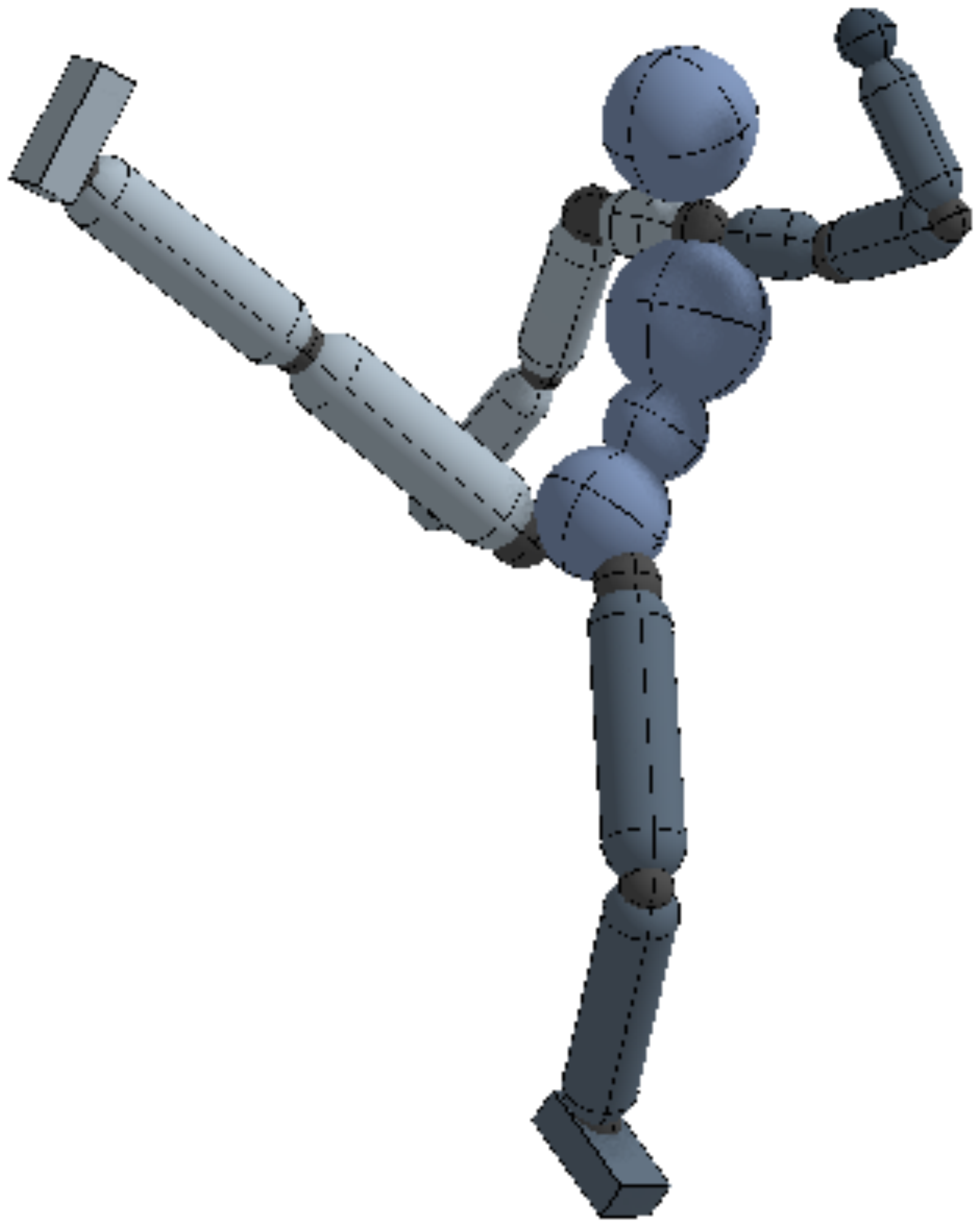
State: 197 D

- link positions
- link velocities



Action:

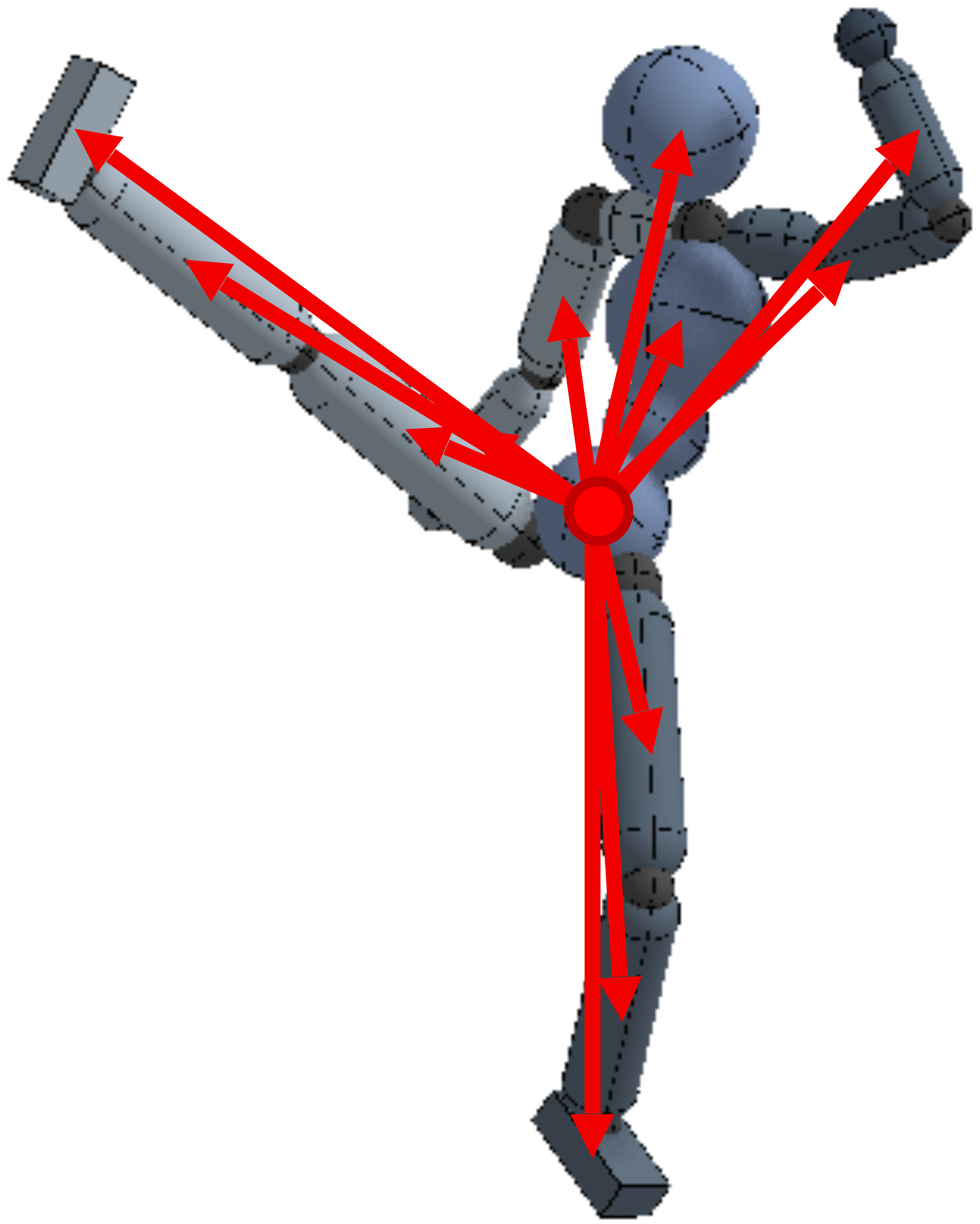
- PD targets



STATE + ACTION

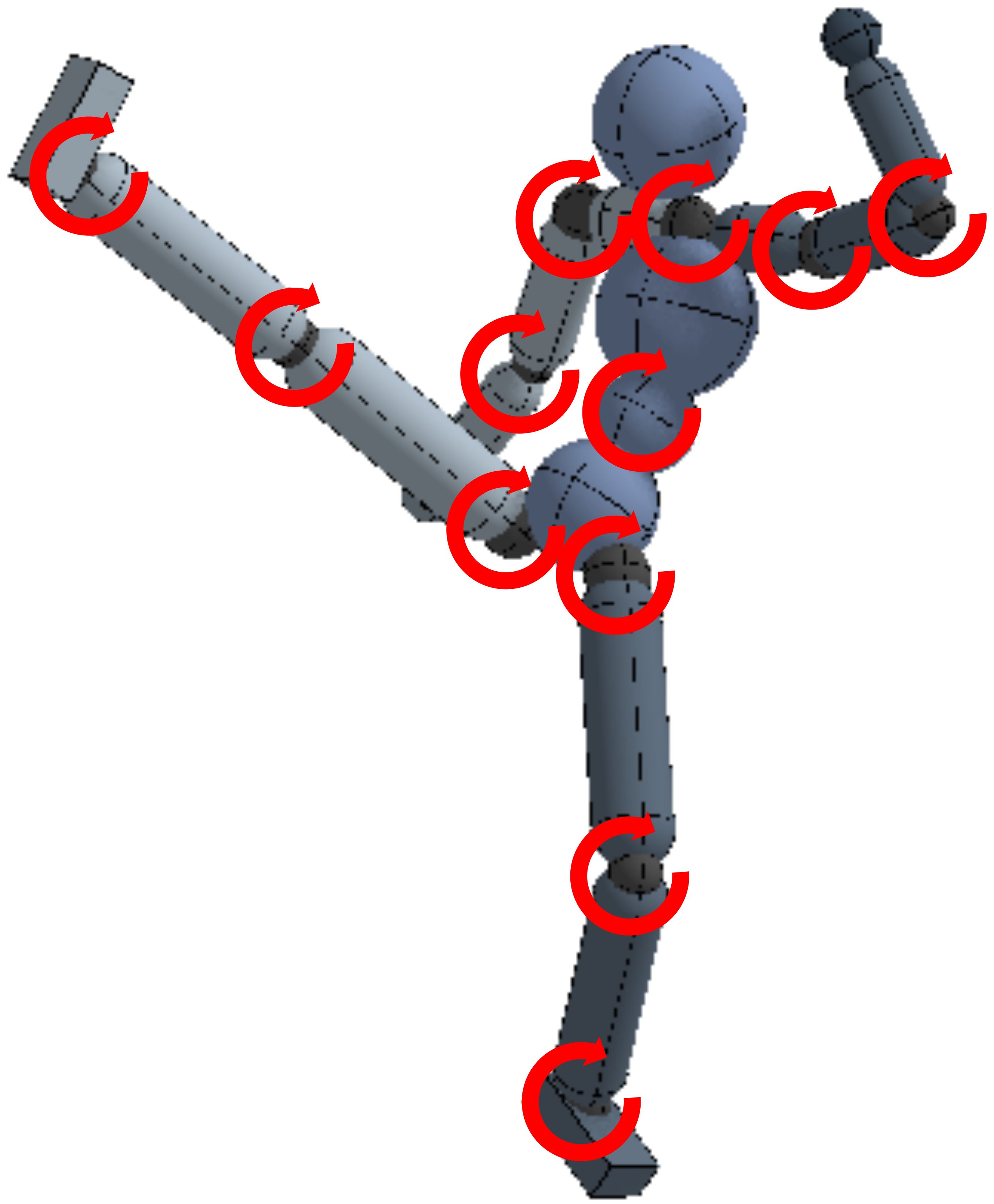
State: 197 D

- link positions
- link velocities



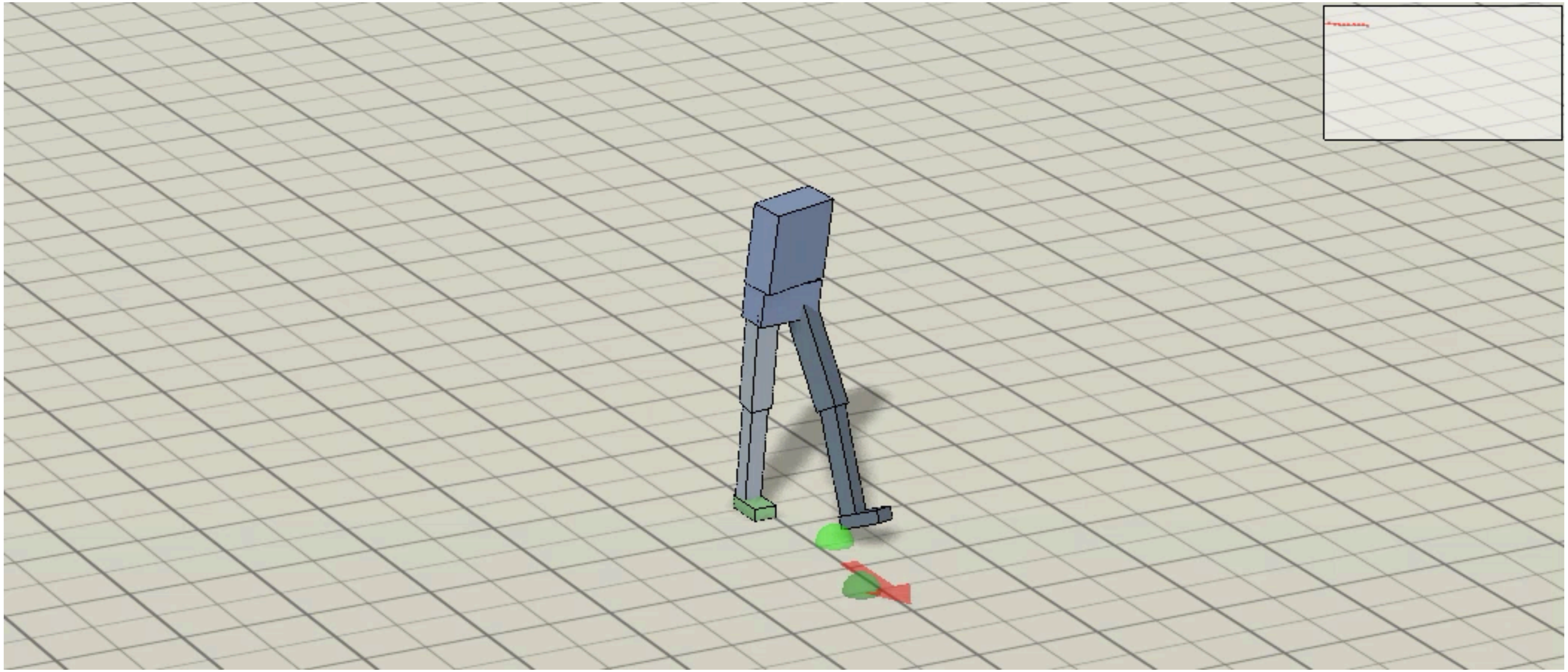
Action: 36 D

- PD targets



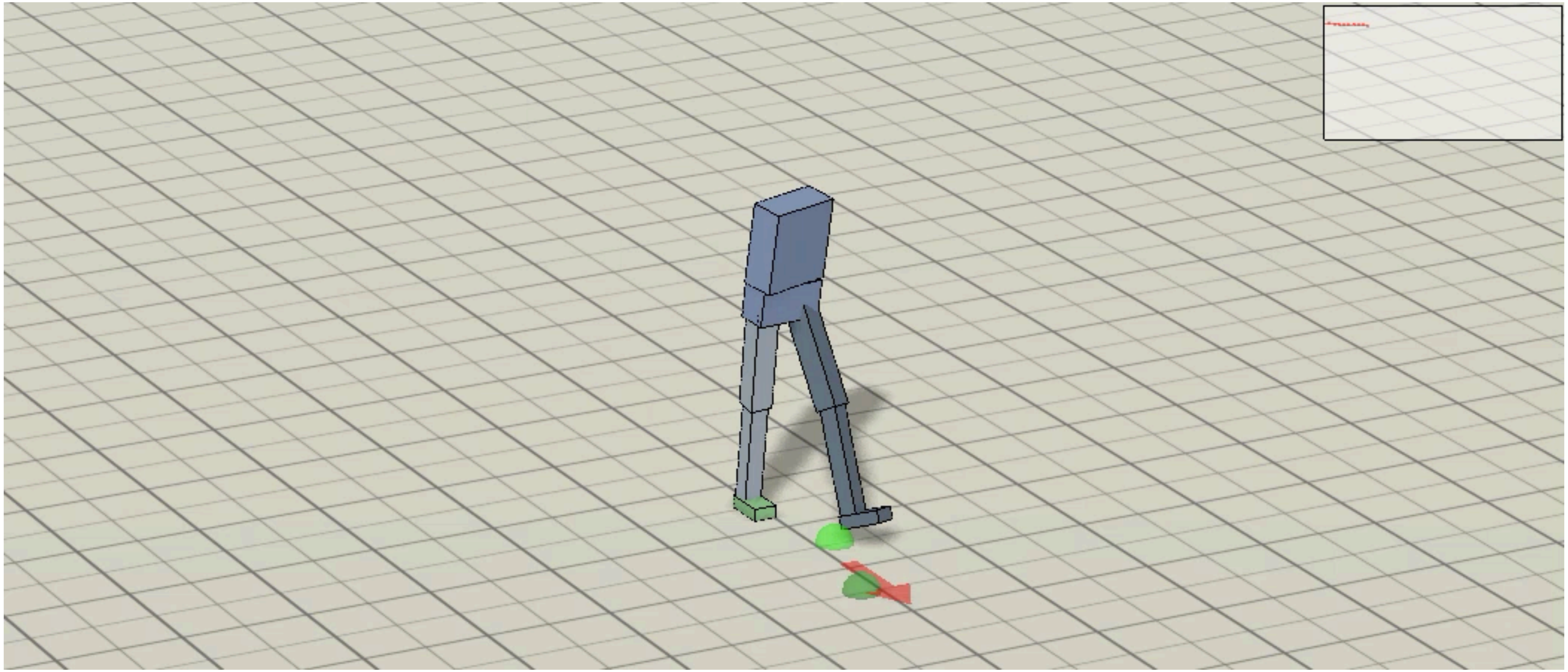
WALKING

[DeepLoco: SIGGRAPH 2017]



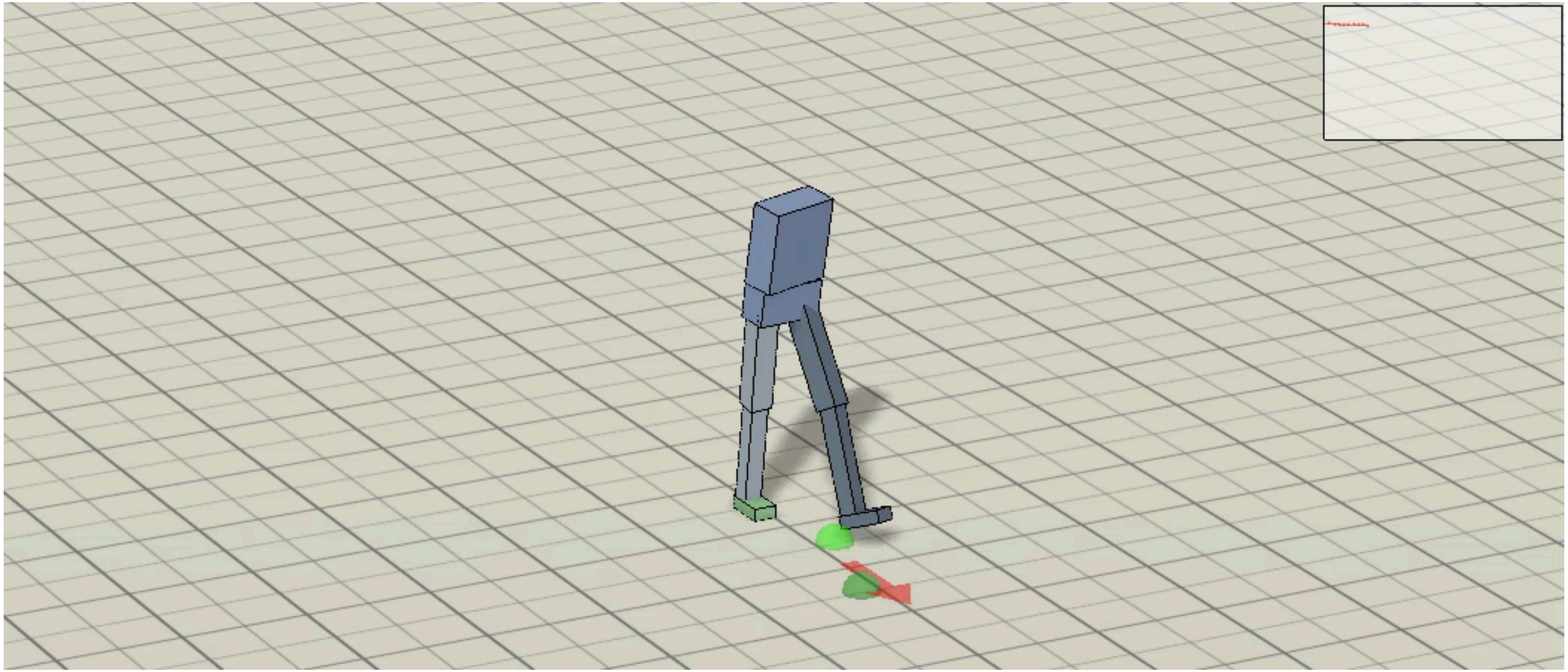
WALKING

[DeepLoco: SIGGRAPH 2017]



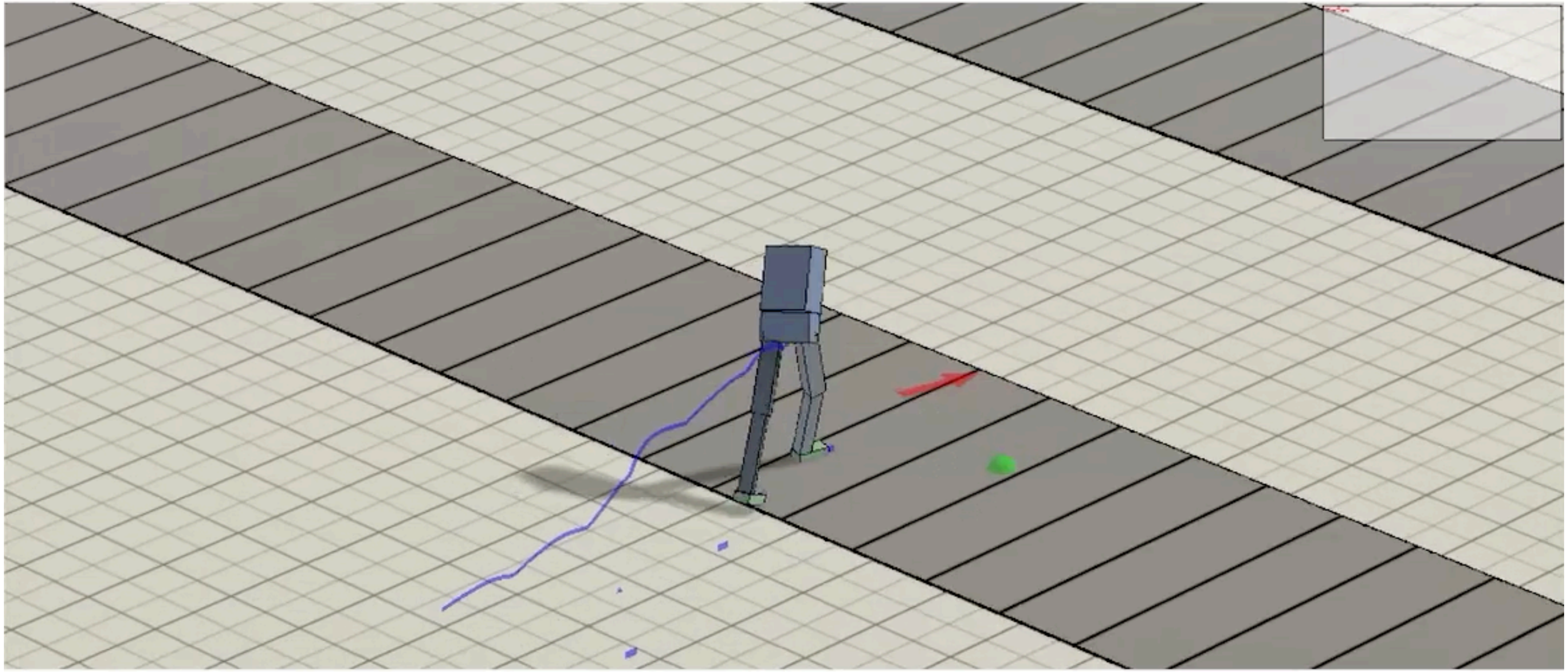
WALKING

[DeepLoco: SIGGRAPH 2017]



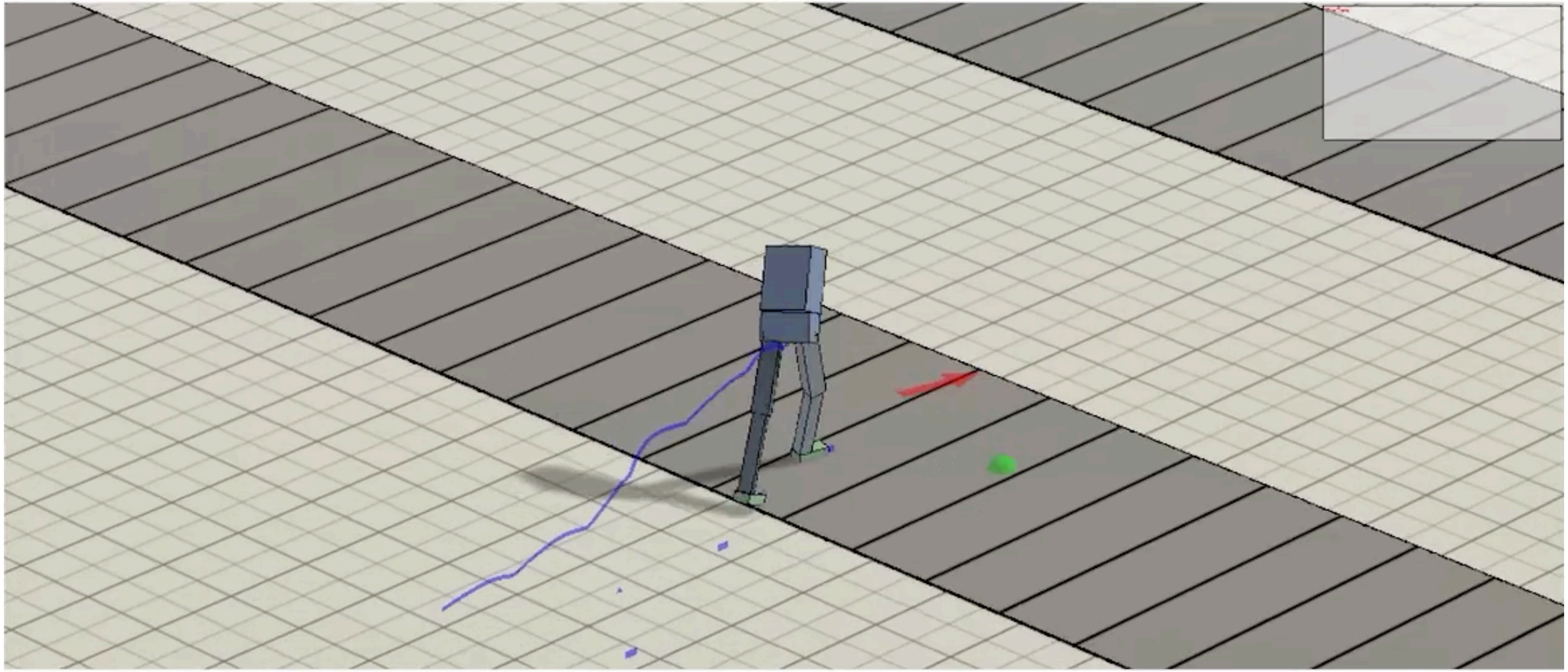
[DeepLoco: SIGGRAPH 2017]

Walking on Conveyor Belts



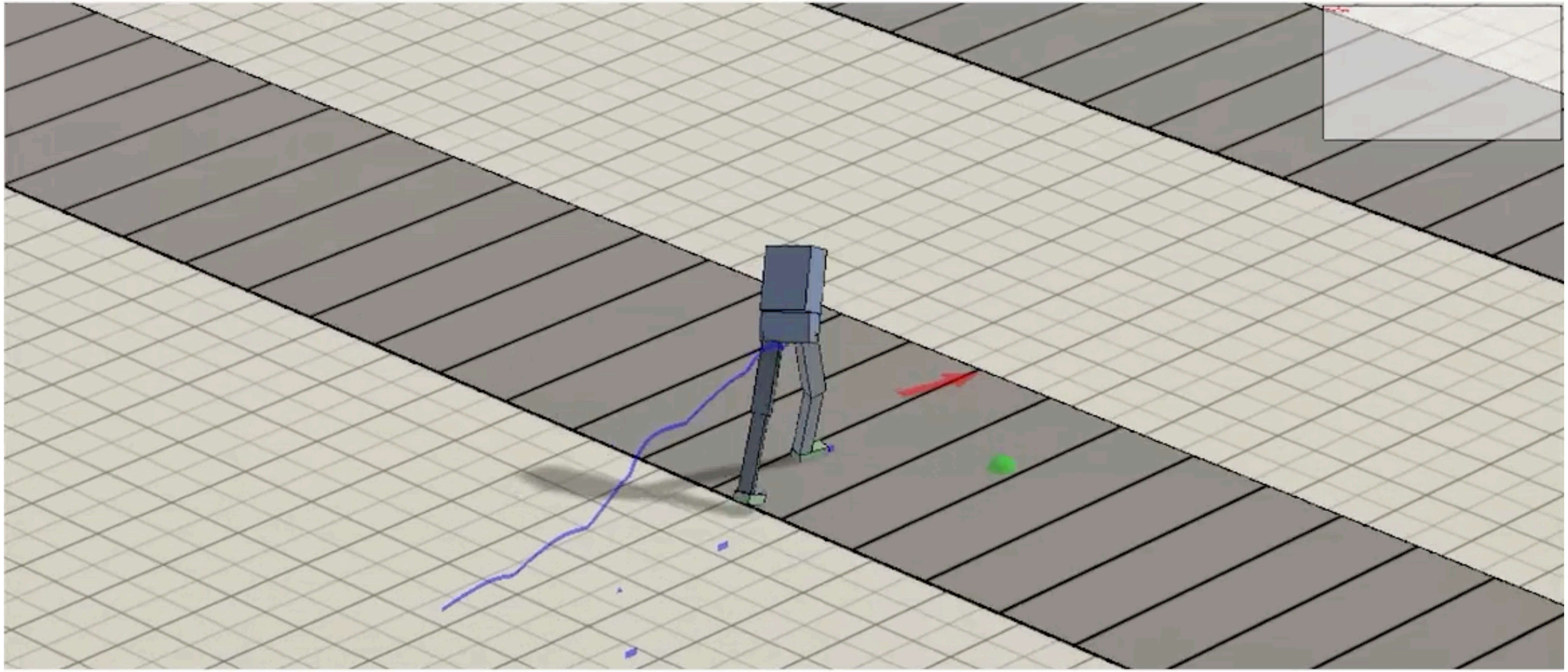
[DeepLoco: SIGGRAPH 2017]

Walking on Conveyor Belts



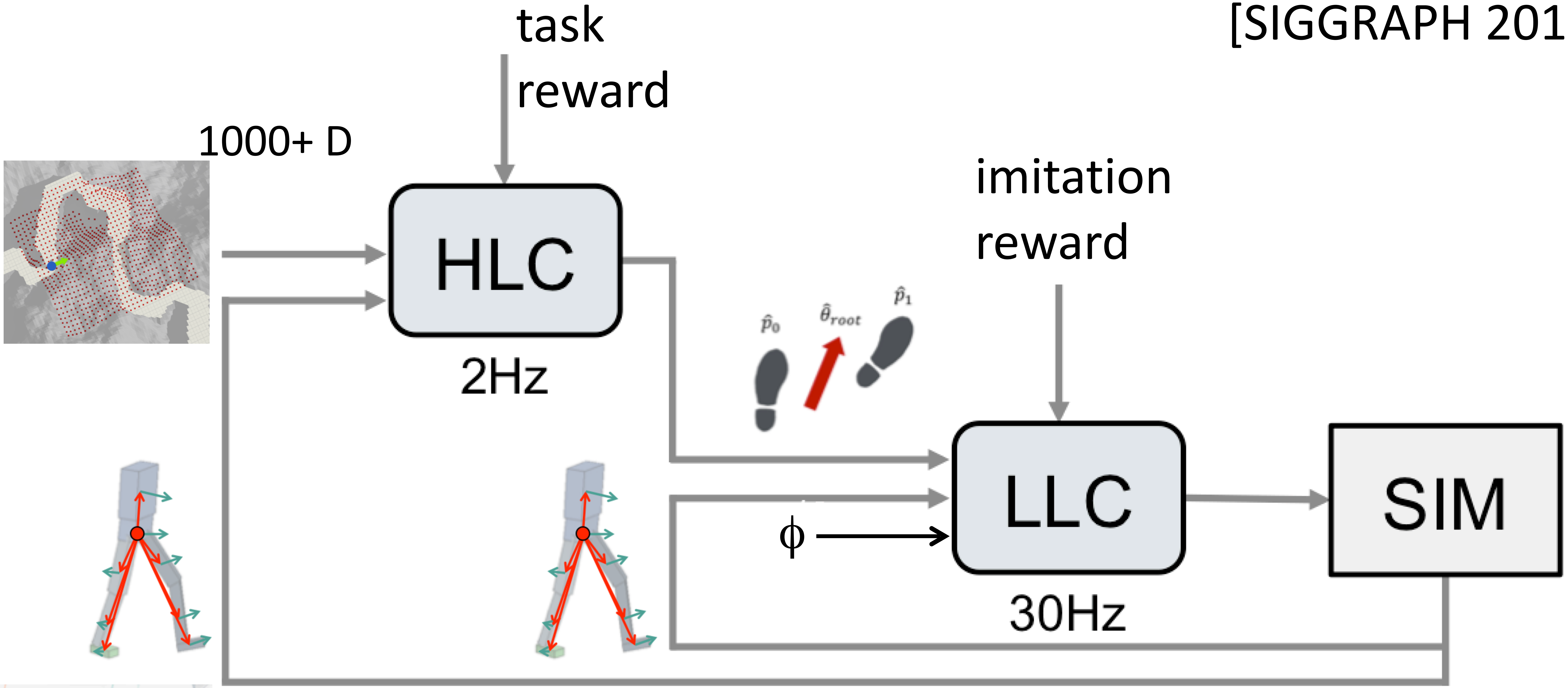
[DeepLoco: SIGGRAPH 2017]

Walking on Conveyor Belts



DEEPLOCO: HIERARCHICAL RL

[SIGGRAPH 2017]

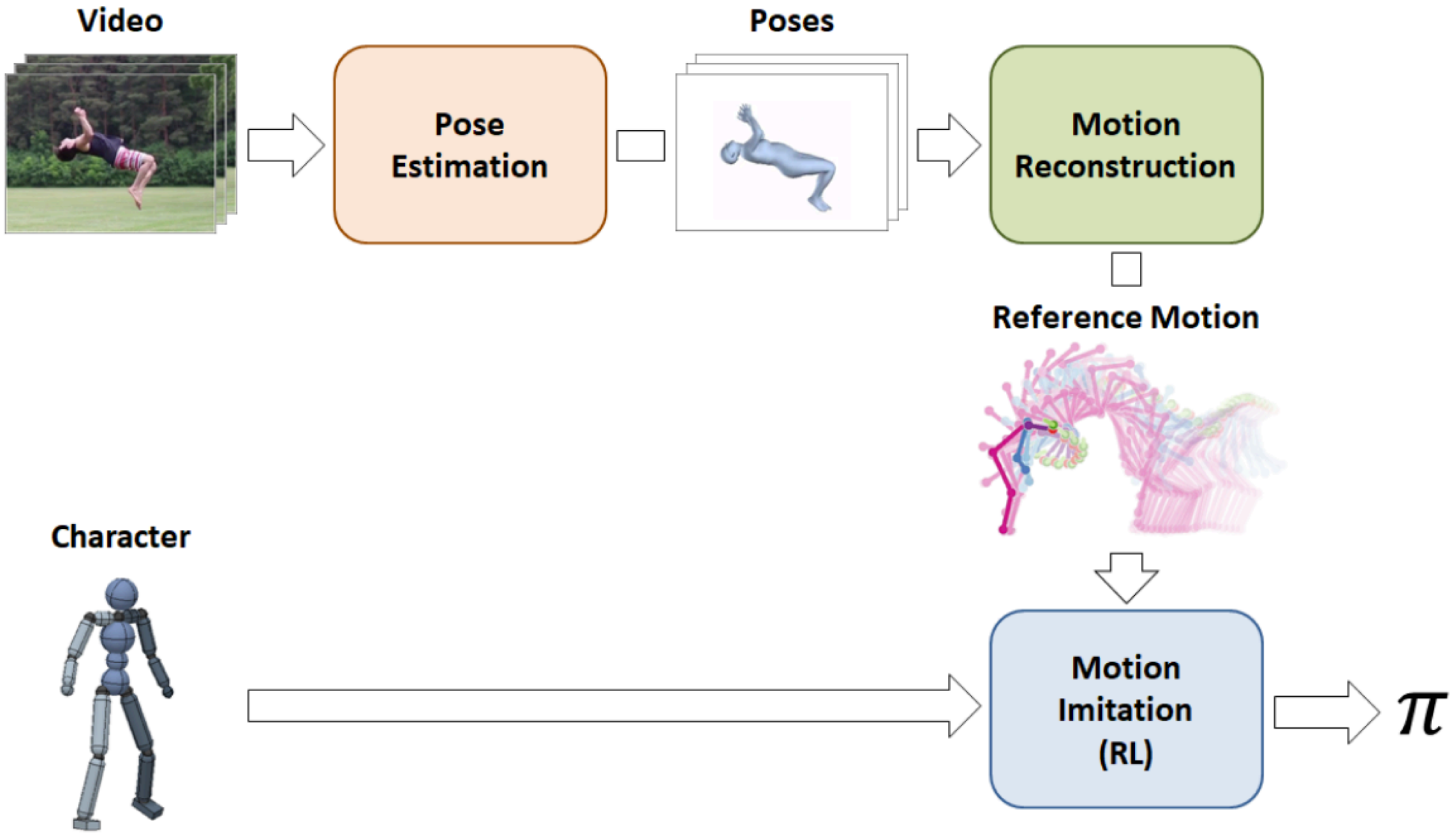


Skills From Video:
Reinforcement learning of physical skills from video

[SIGGRAPH ASIA 2018]

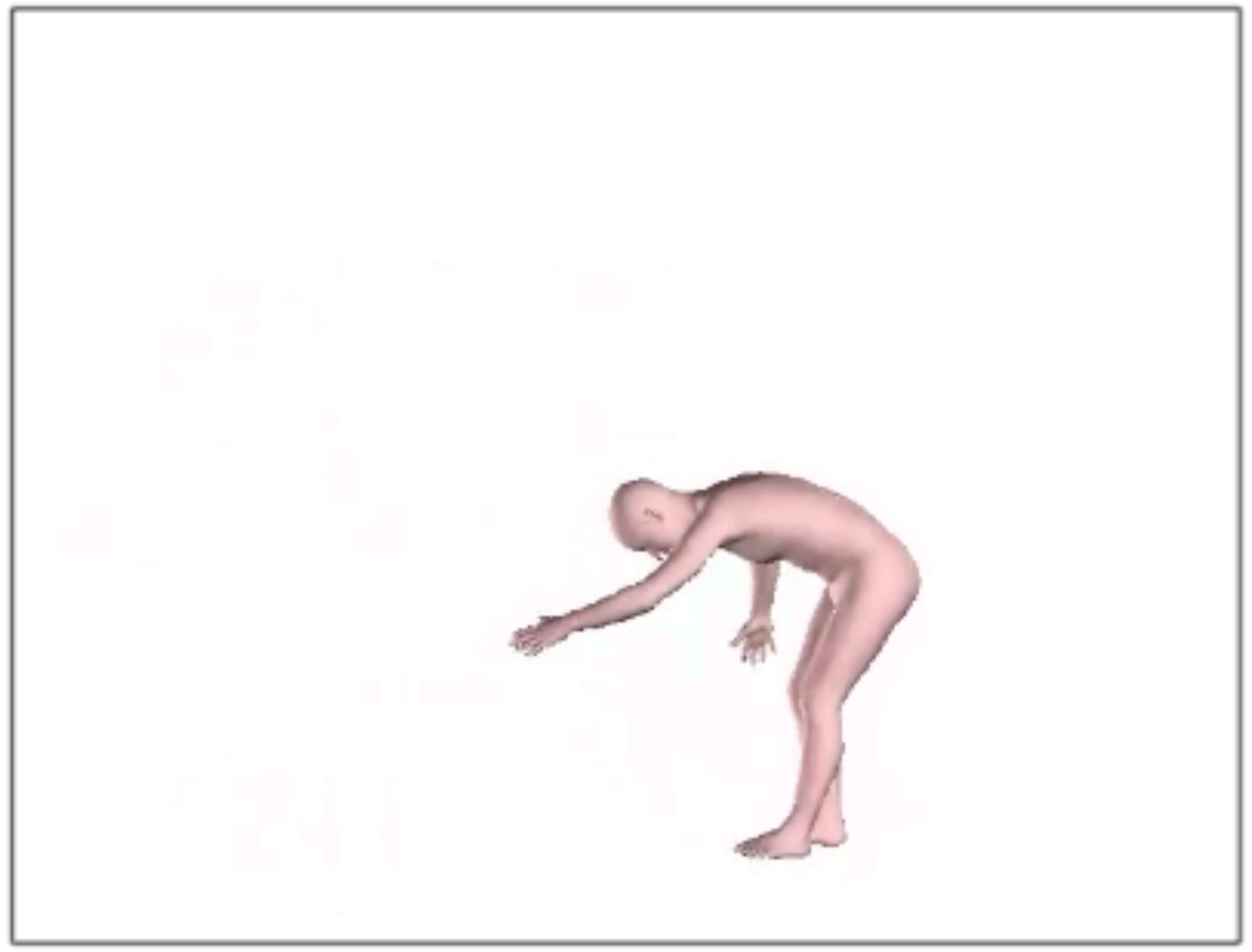
Transactions on Graphics (Proc. ACM SIGGRAPH Asia 2018)

Xue Bin Peng Angjoo Kanazawa Jitendra Malik Pieter Abbeel Sergey Levine
University of California, Berkeley

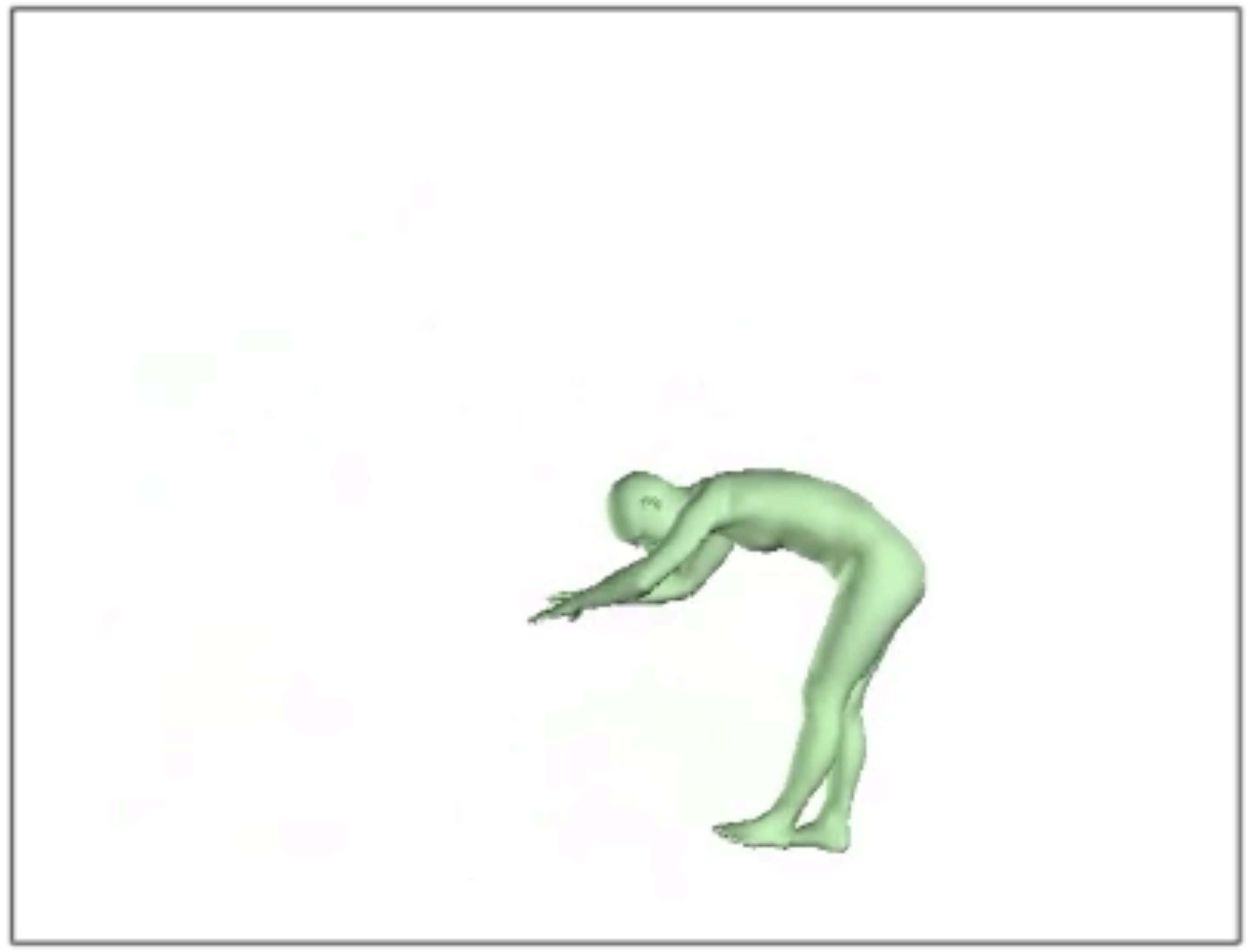

Video
Pose
Estimation
Poses
Motion
Reconstruction
Reference Motion
Character
Motion
Imitation
(RL)
π

Pose Estimation

Video: Handspring A



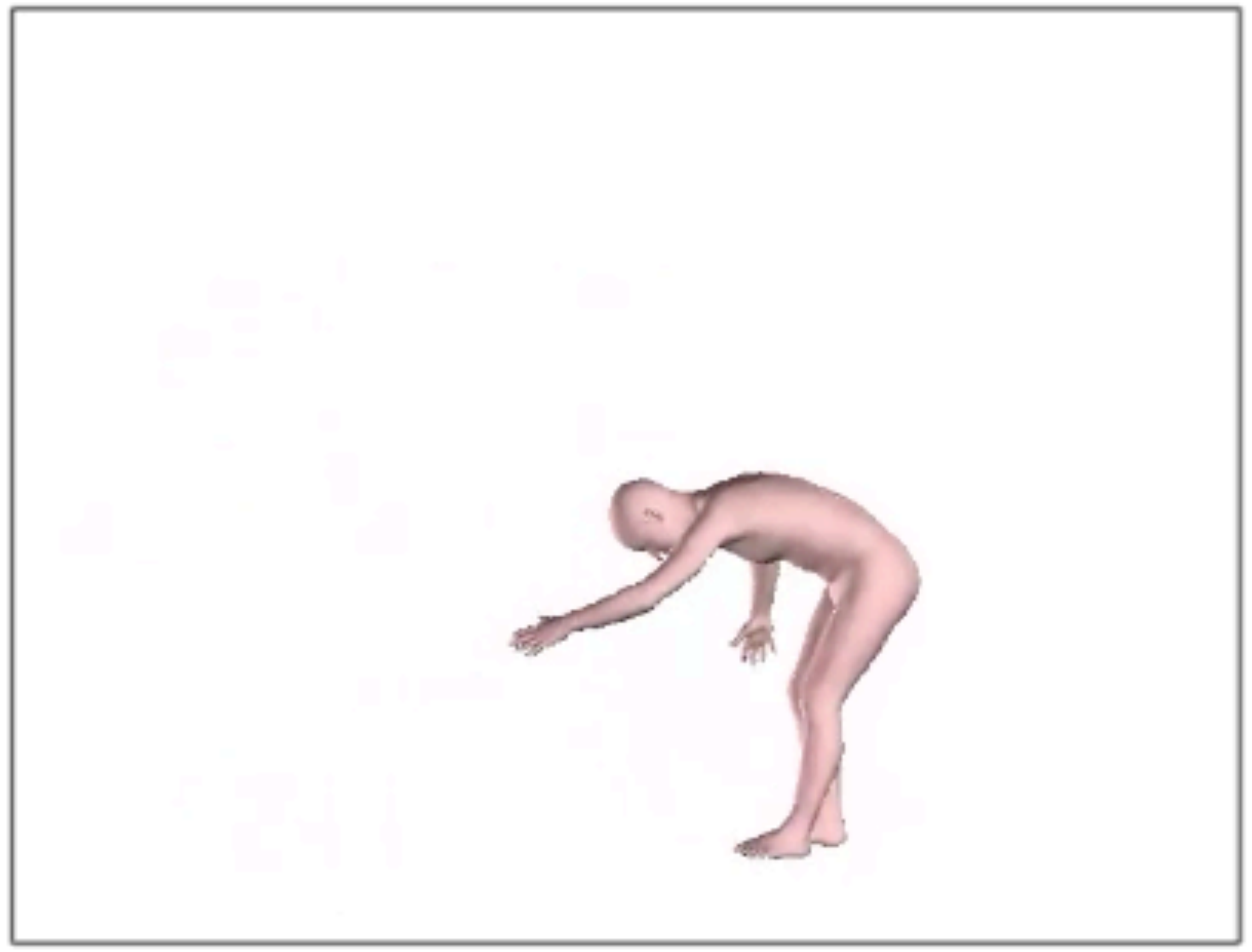
No Augmentation
[Kanazawa et al. 2018]



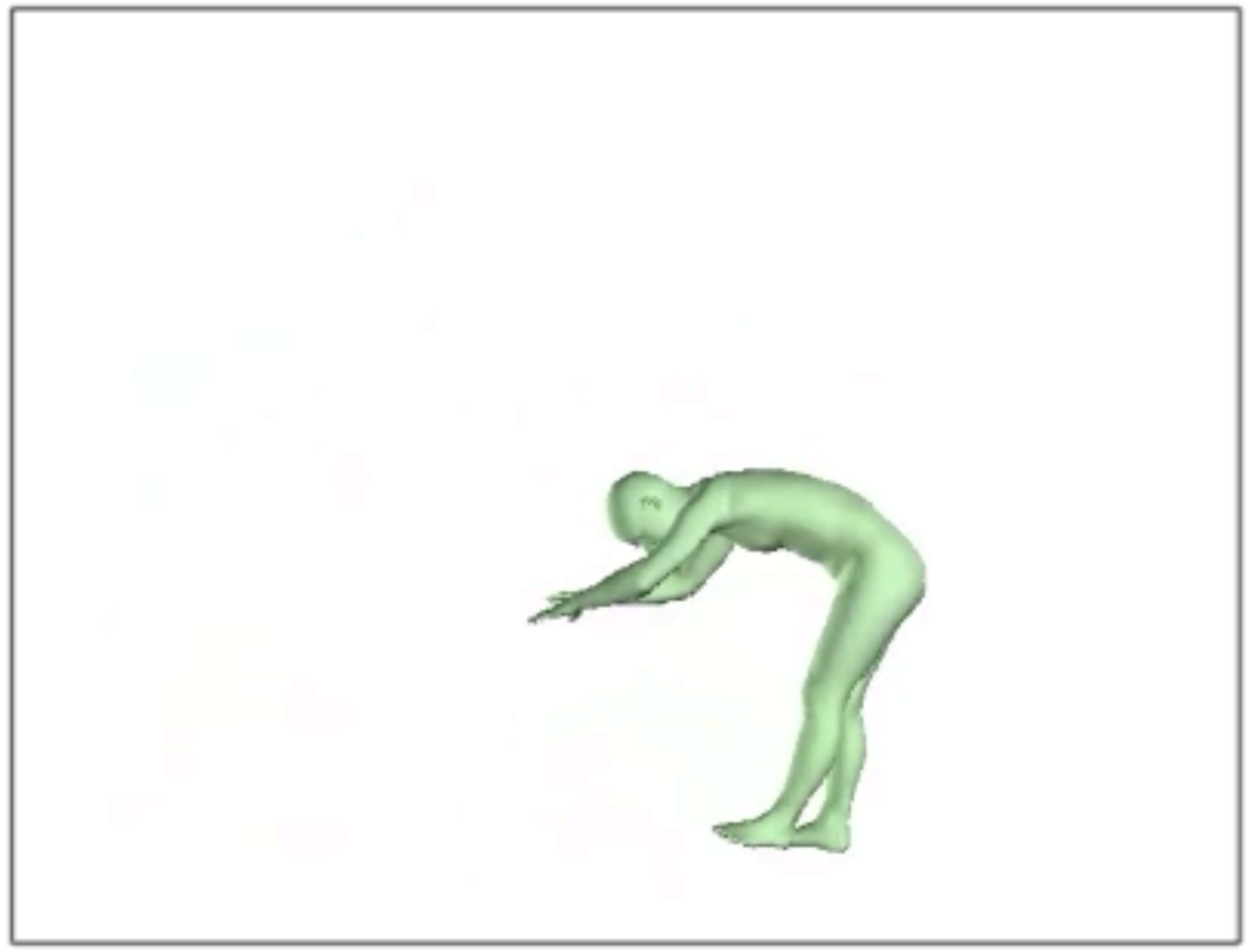
With Augmentation
(our work)

Pose Estimation

Video: Handspring A



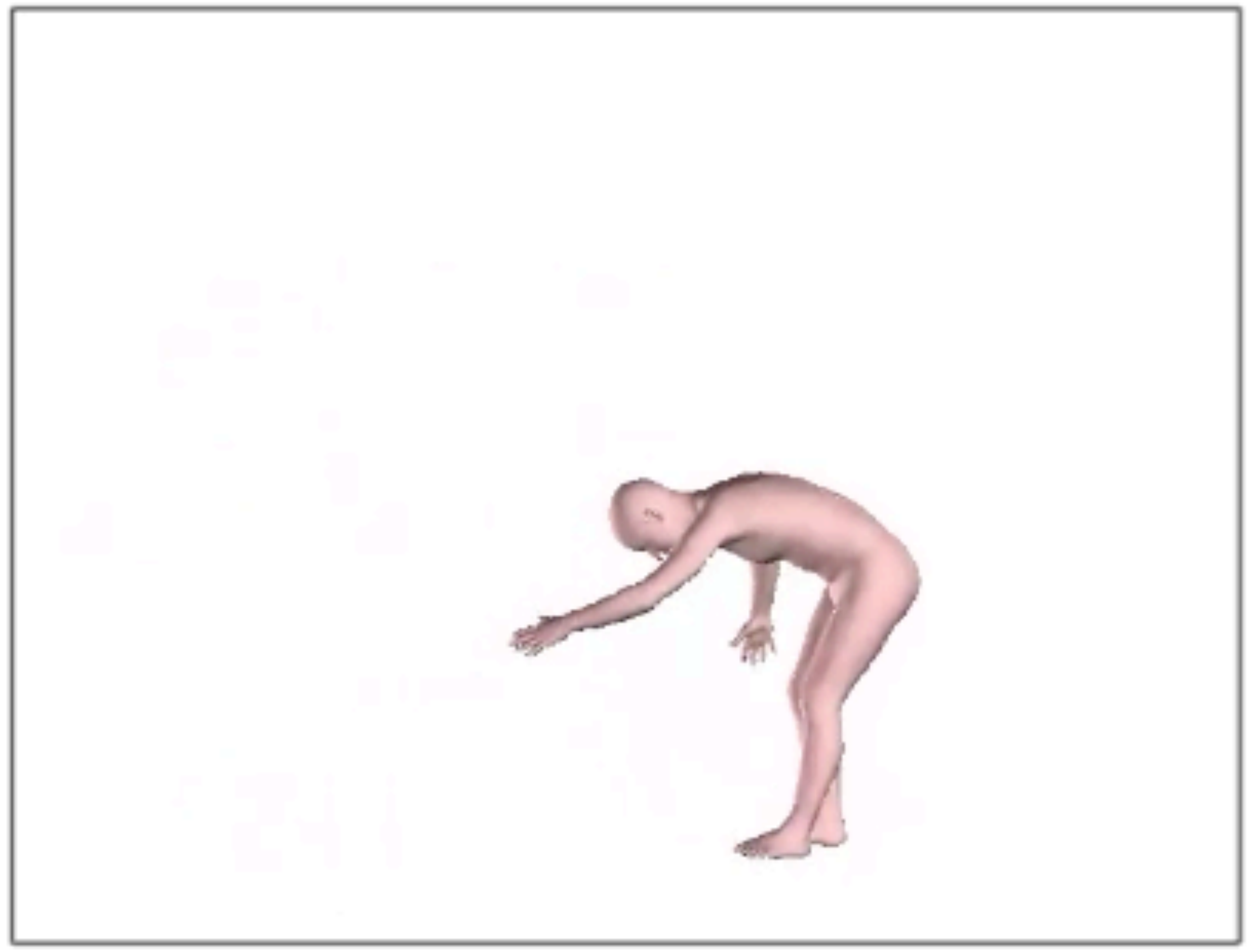
No Augmentation
[Kanazawa et al. 2018]



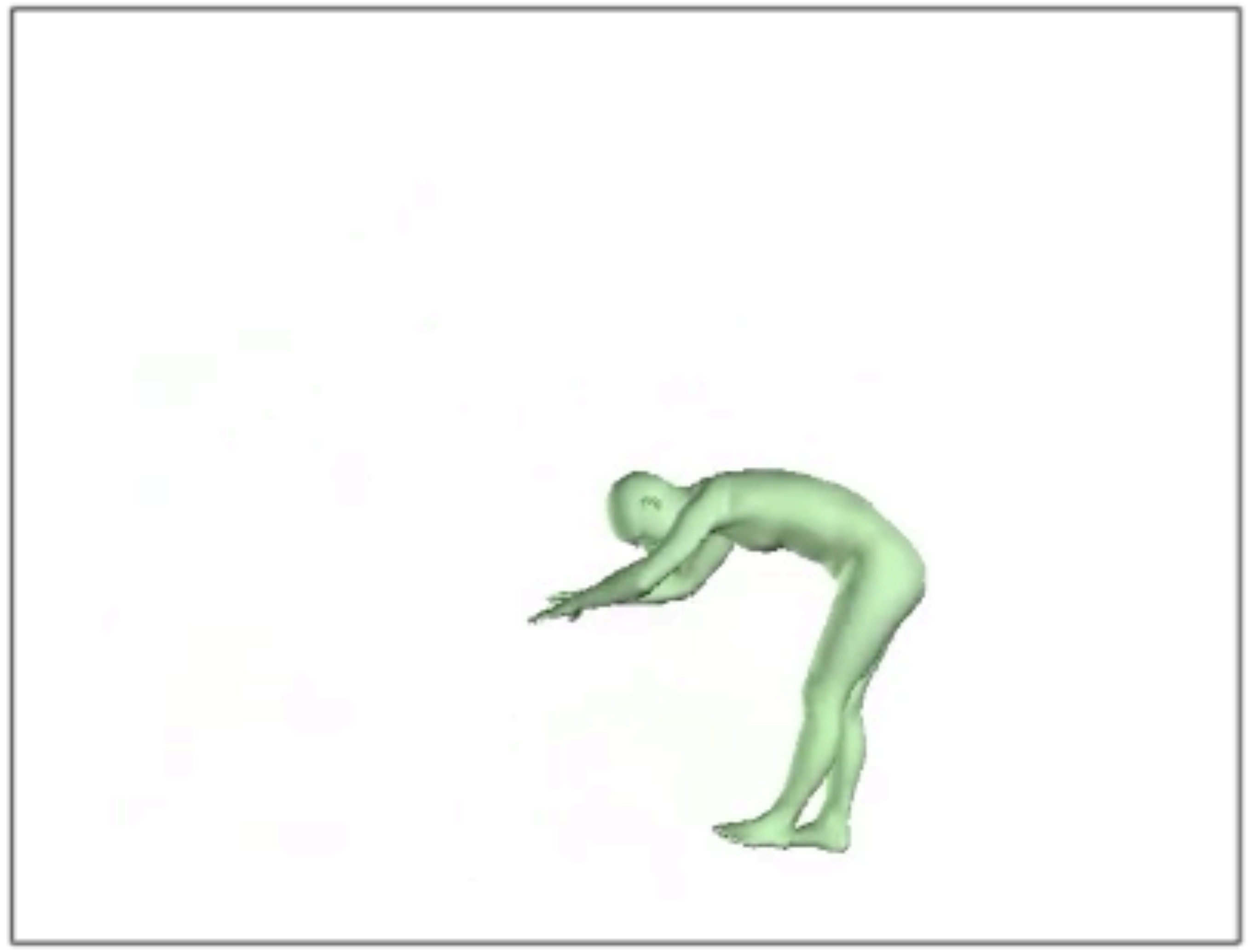
With Augmentation
(our work)

Pose Estimation

Video: Handspring A



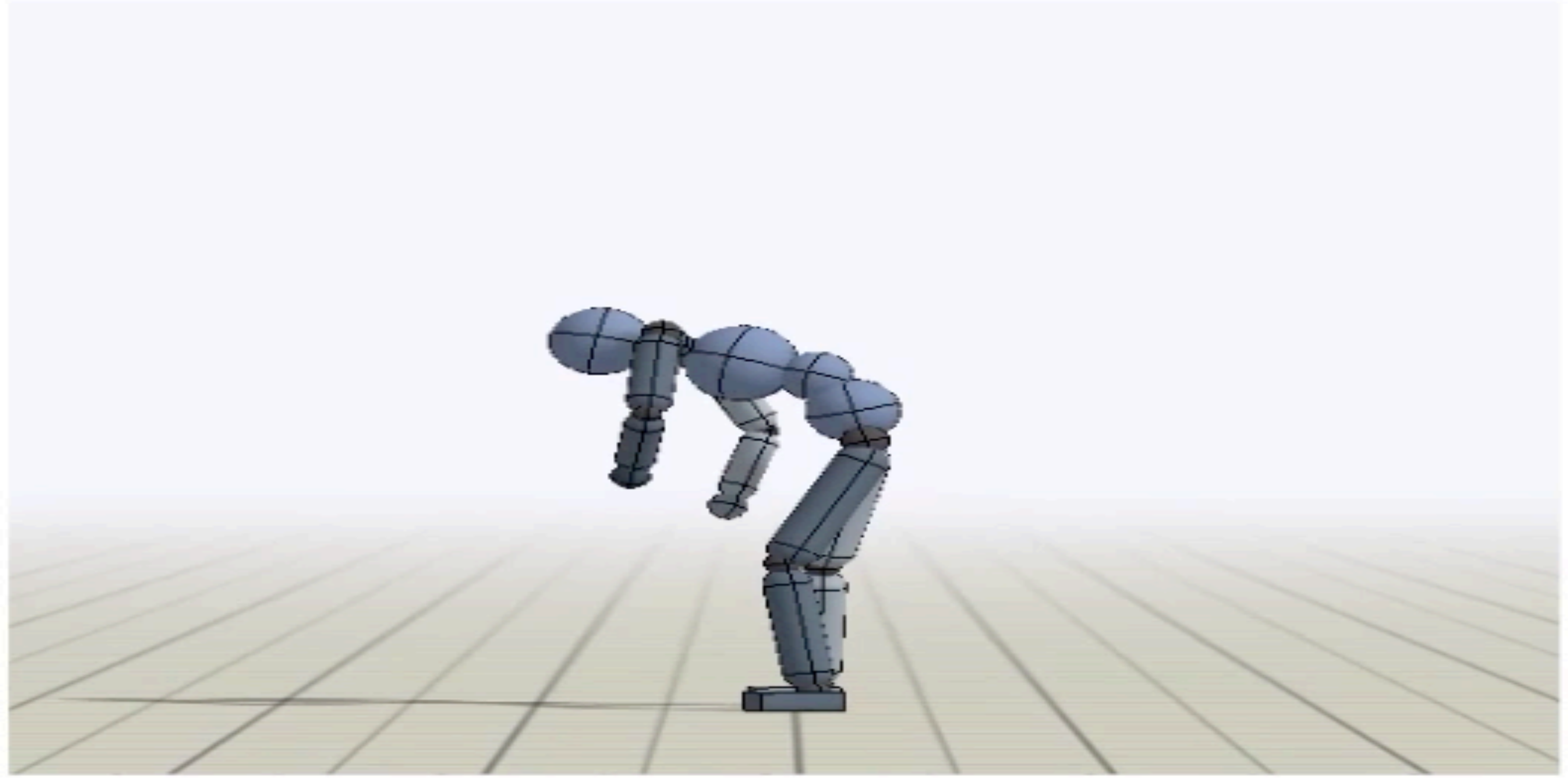
No Augmentation
[Kanazawa et al. 2018]



With Augmentation
(our work)

SKILLS FROM VIDEOS

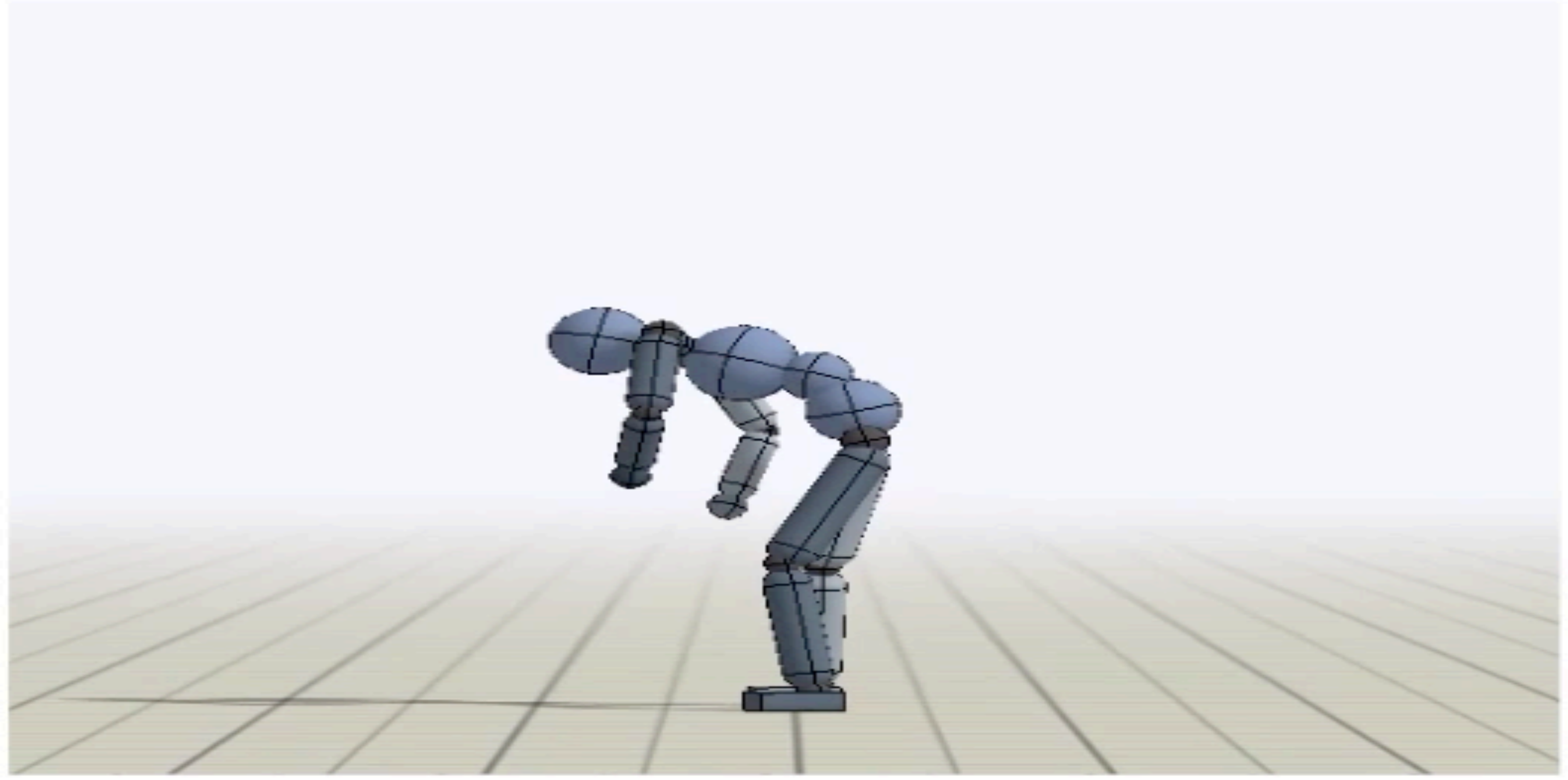
Video



Simulation

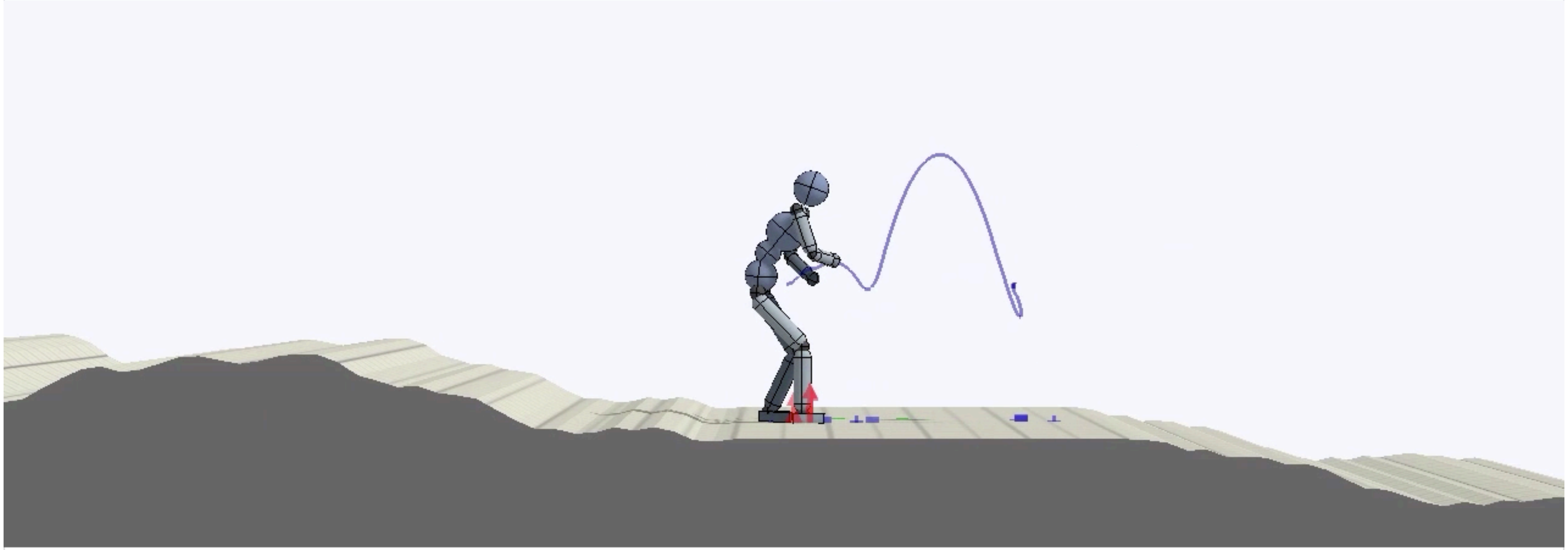
SKILLS FROM VIDEOS

Video



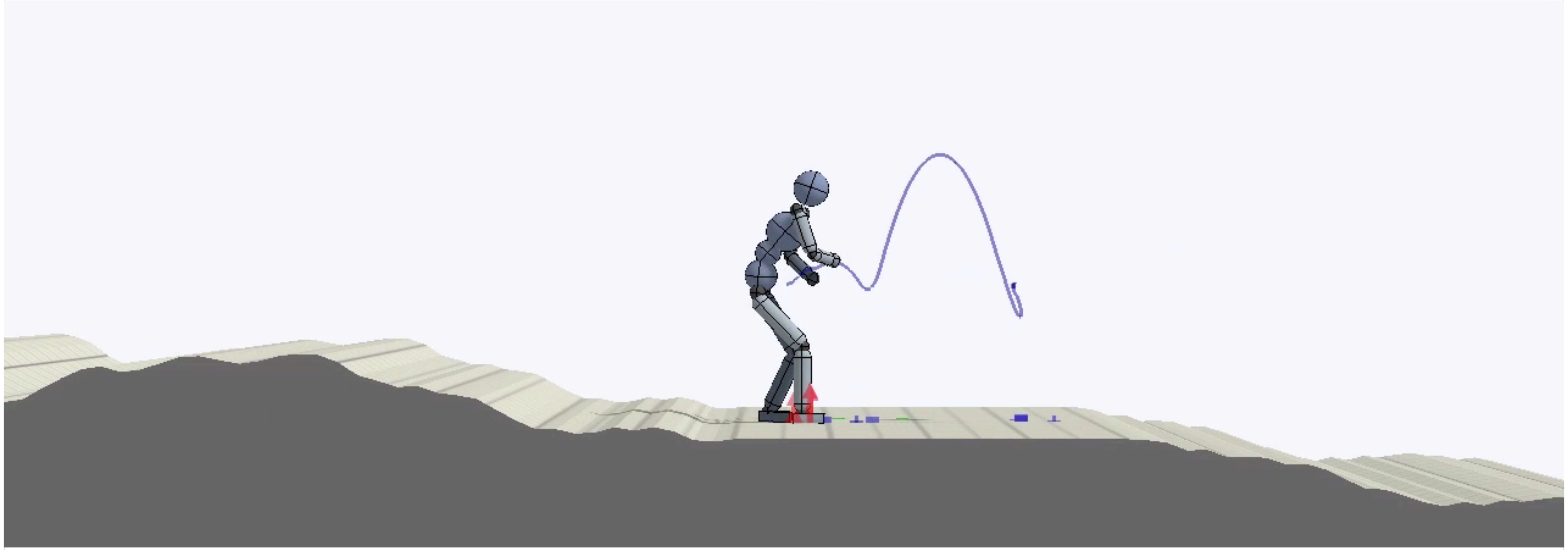
Simulation

Retargeting



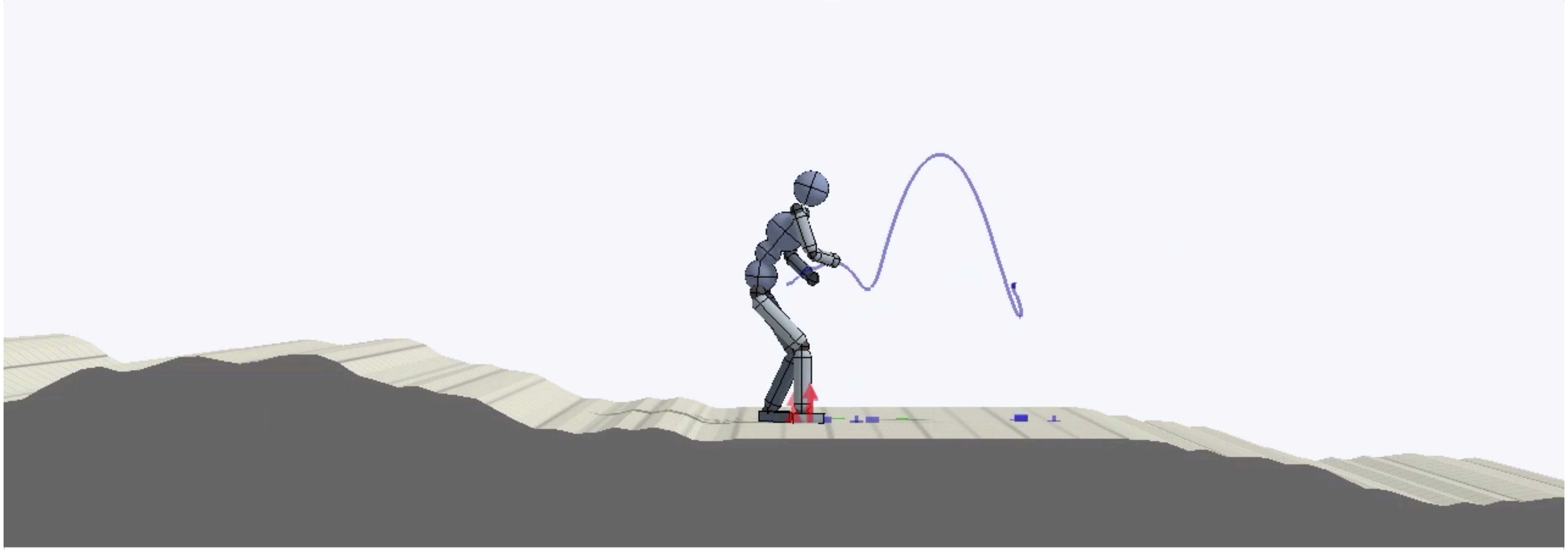
and retarget to different environments.

Retargeting



and retarget to different environments.

Retargeting

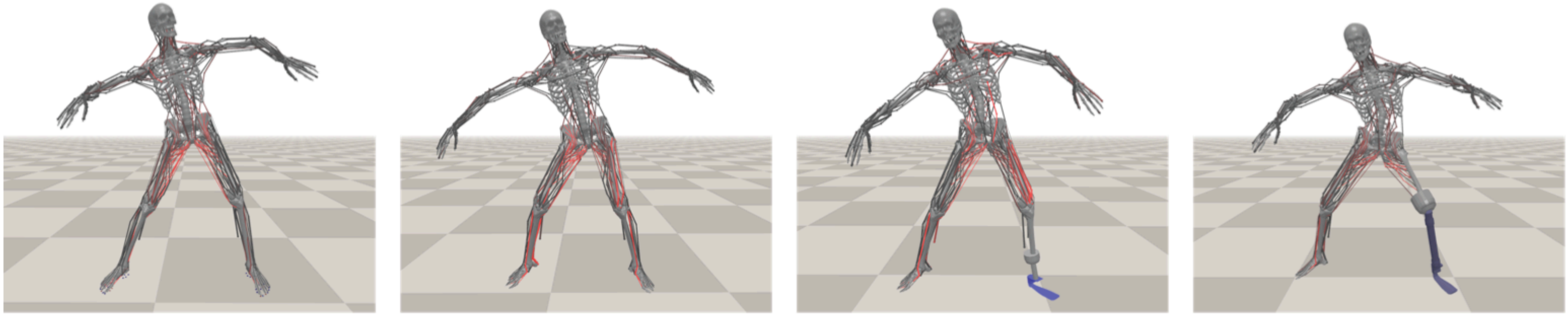


and retarget to different environments.

DEEP-MIMIC FOR BIOMECHANICAL MODELS

Scalable Muscle-Actuated Human Simulation and Control

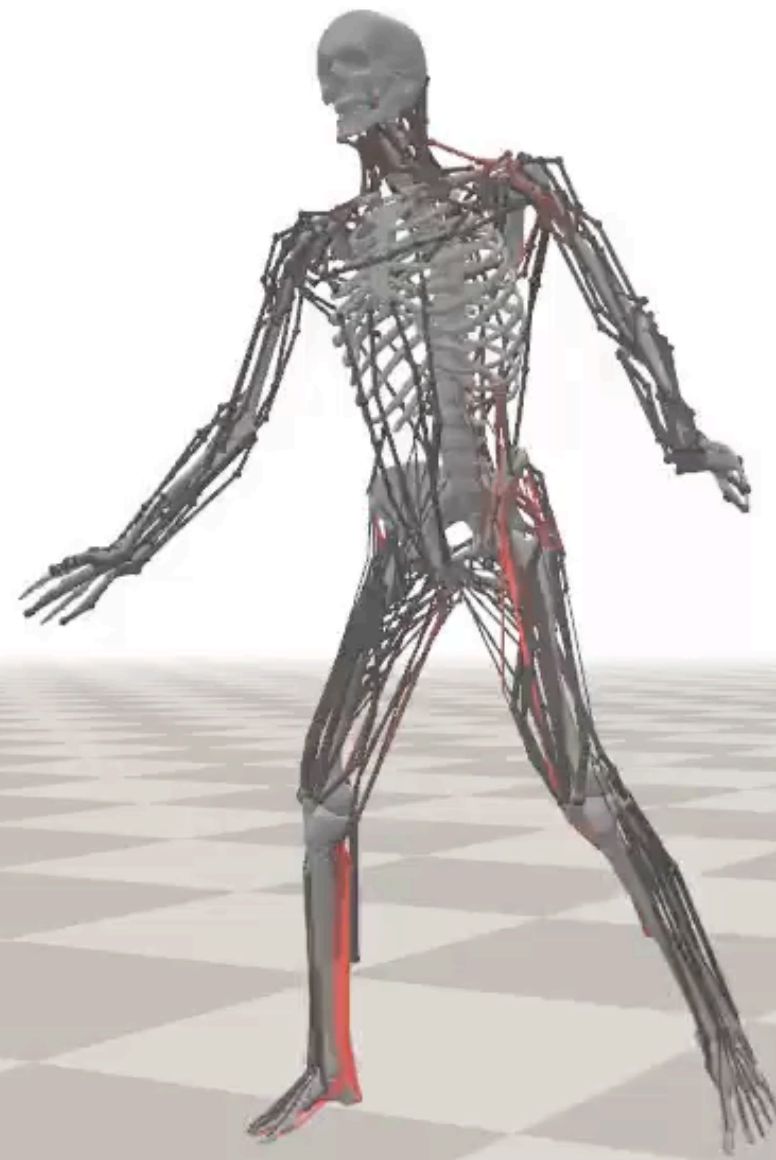
SIGGRAPH 2019 Conditional Accept, Seoul National University



Scalable Muscle-Actuated Human Simulation and Control

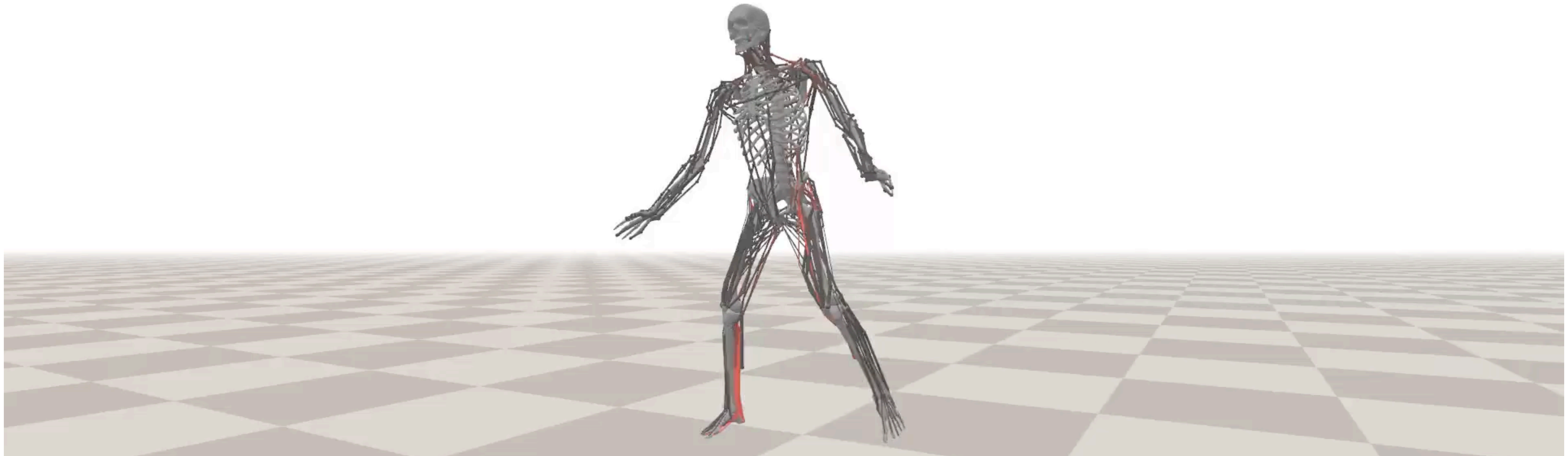
Seunghwan Lee[1], Kyoungmin Lee[2], Moonseok Park[2], and Jehee Lee[1]
Seoul National University[1], Seoul National University Bundang Hospital[2]

Scalable Muscle-Actuated Human Simulation and Control



Seunghwan Lee[1], Kyoungmin Lee[2], Moonseok Park[2], and Jehee Lee[1]
Seoul National University[1], Seoul National University Bundang Hospital[2]

Scalable Muscle-Actuated Human Simulation and Control



Seunghwan Lee(1), Kyoungmin Lee(2), Moonseok Park(2), and Jehee Lee(1)
Seoul National University(1), Seoul National University Bundang Hospital(2)

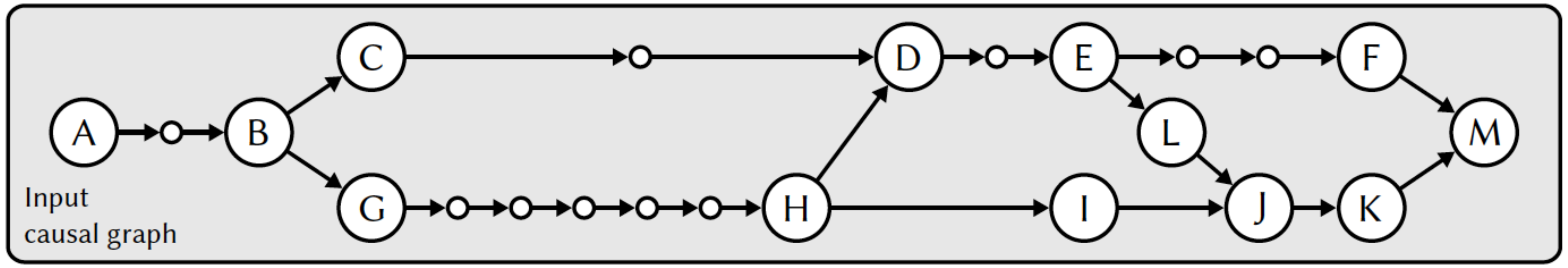
"World’s largest Rube Goldberg machine lights up Christmas tree"
https://www.youtube.com/watch?v=RBOqfLVCDv8

"World's largest Rube Goldberg machine lights up Christmas tree"
https://www.youtube.com/watch?v=RBOqfLVCDv8

Learning Robustness from Simulations

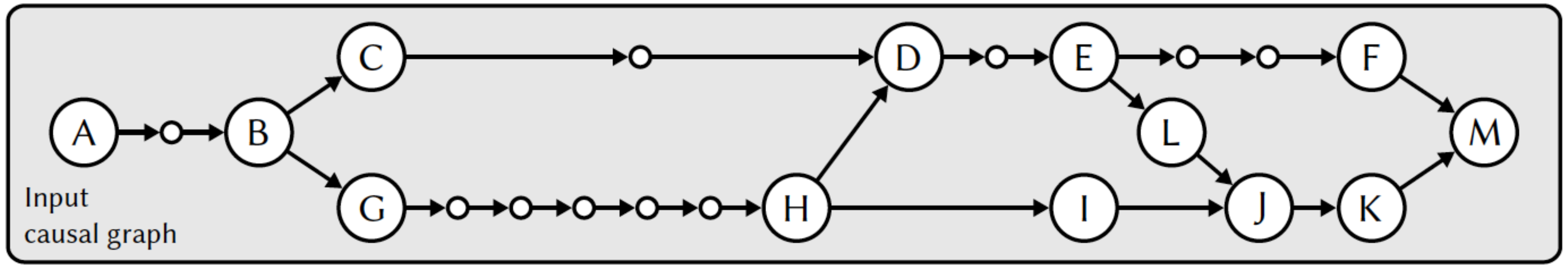
[Roussel, Cani, Leon, Mitra, Siggraph, 2019]

Learning **Robustness** from Simulations

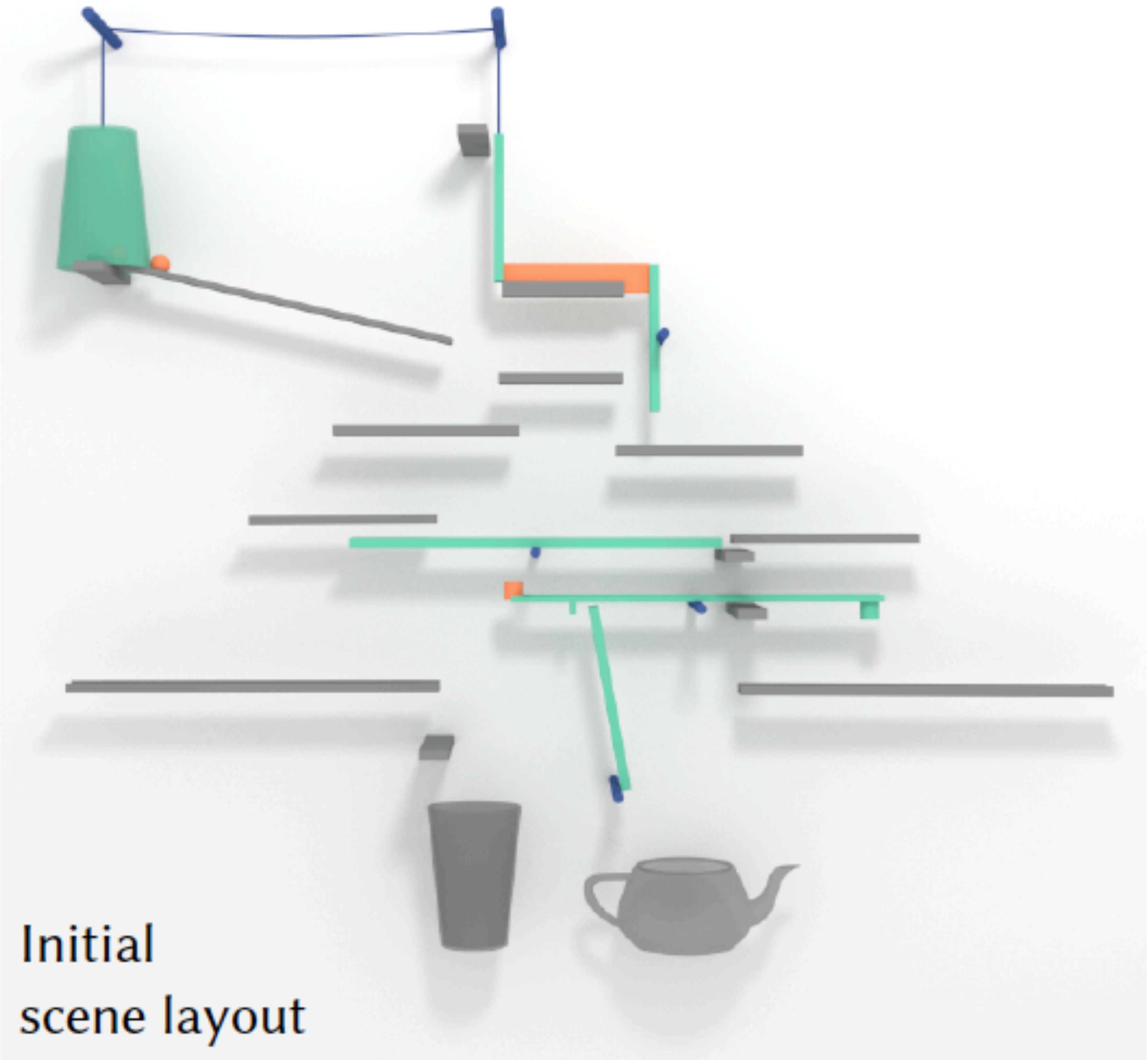


[Roussel, Cani, Leon, Mitra, Siggraph, 2019]

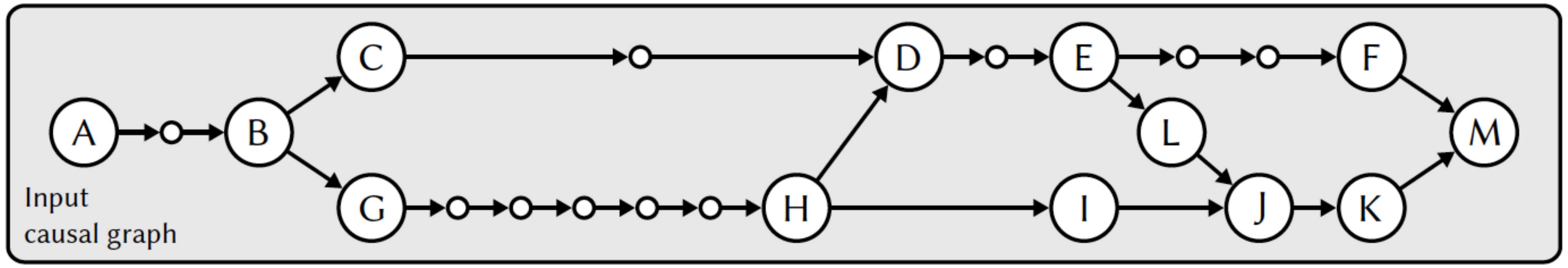
Learning **Robustness** from Simulations



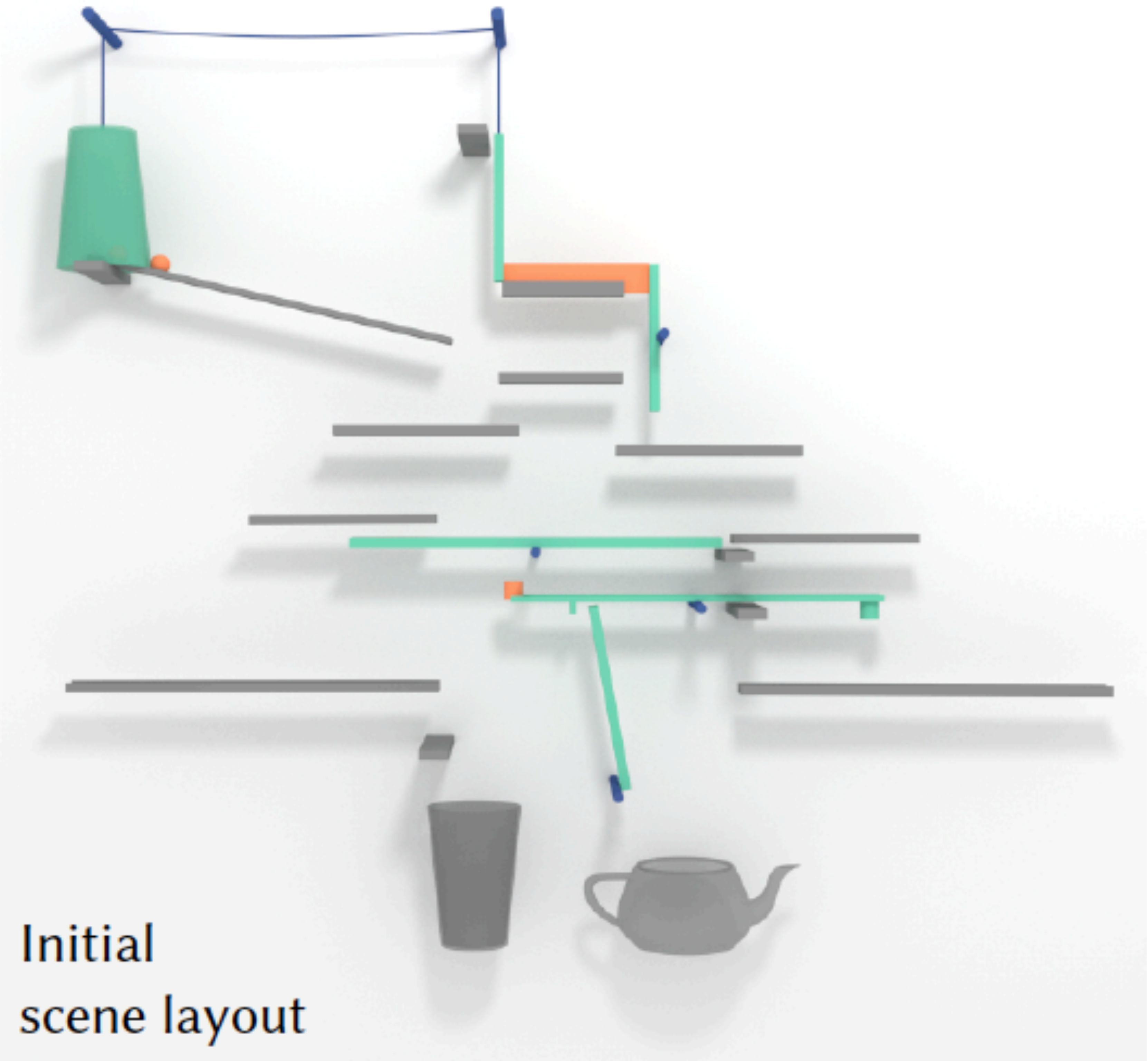
[Roussel, Cani, Leon, Mitra, Siggraph, 2019]

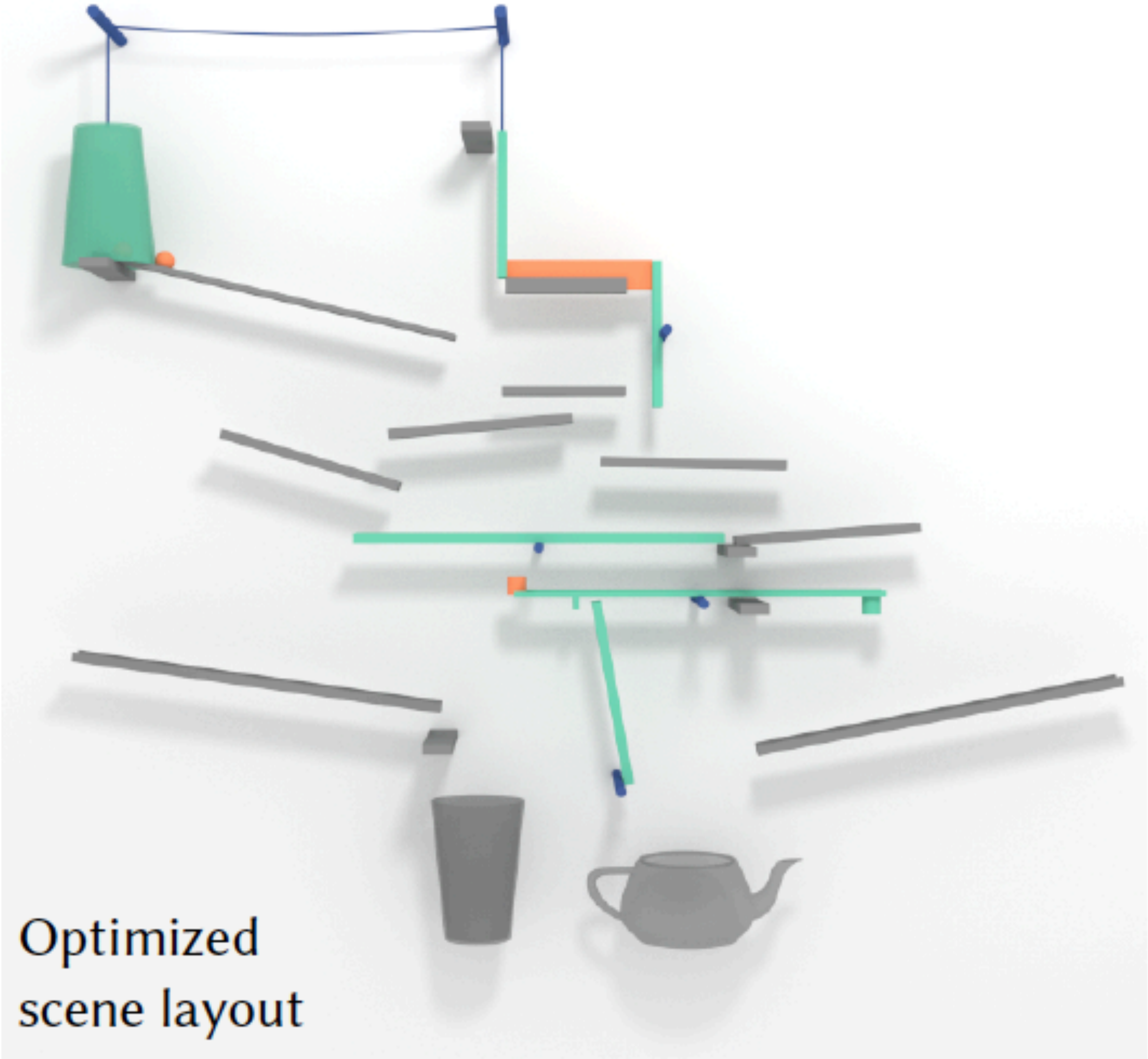


Learning Robustness from Simulations

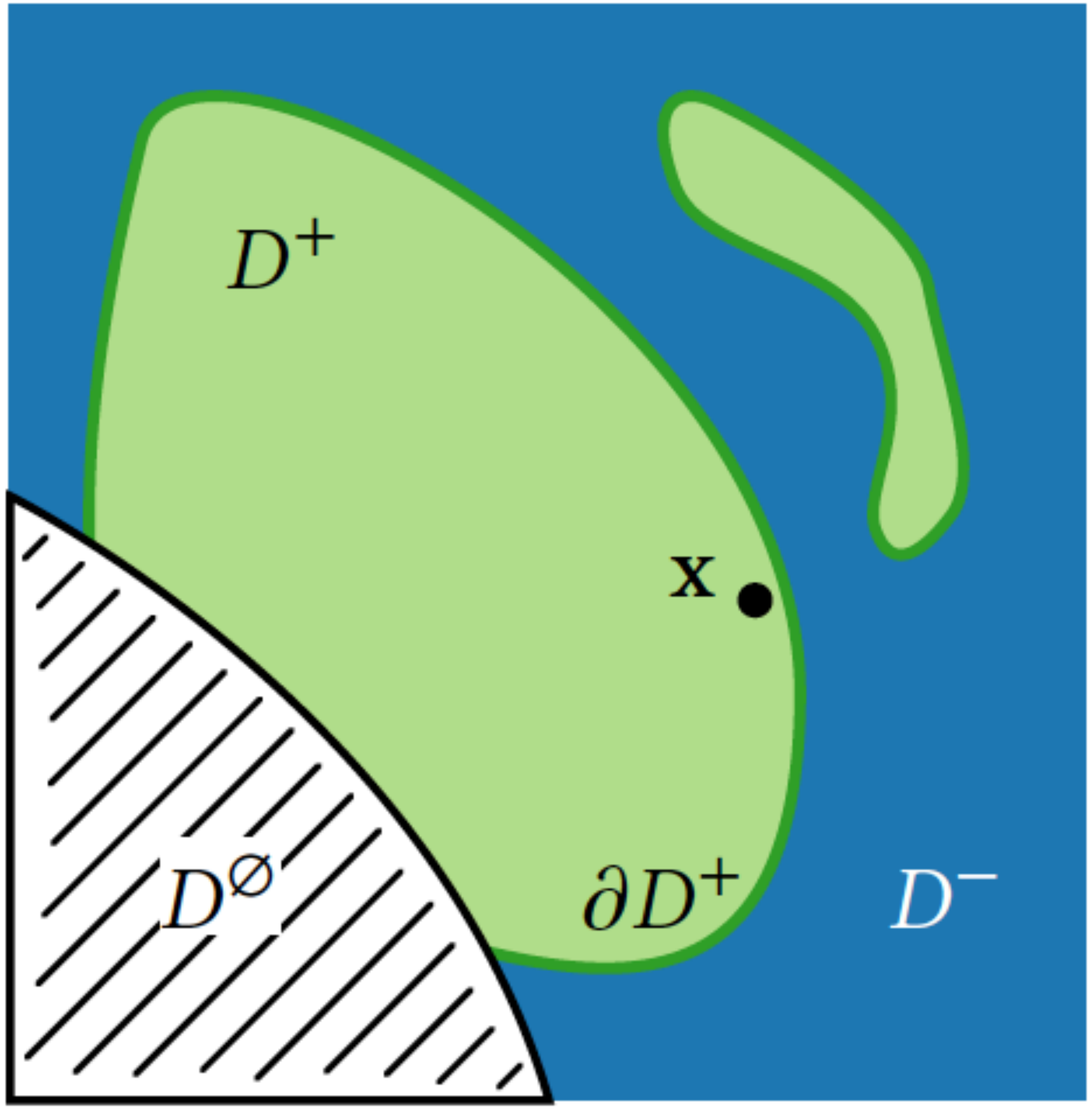


[Roussel, Cani, Leon, Mitra, Siggraph, 2019]

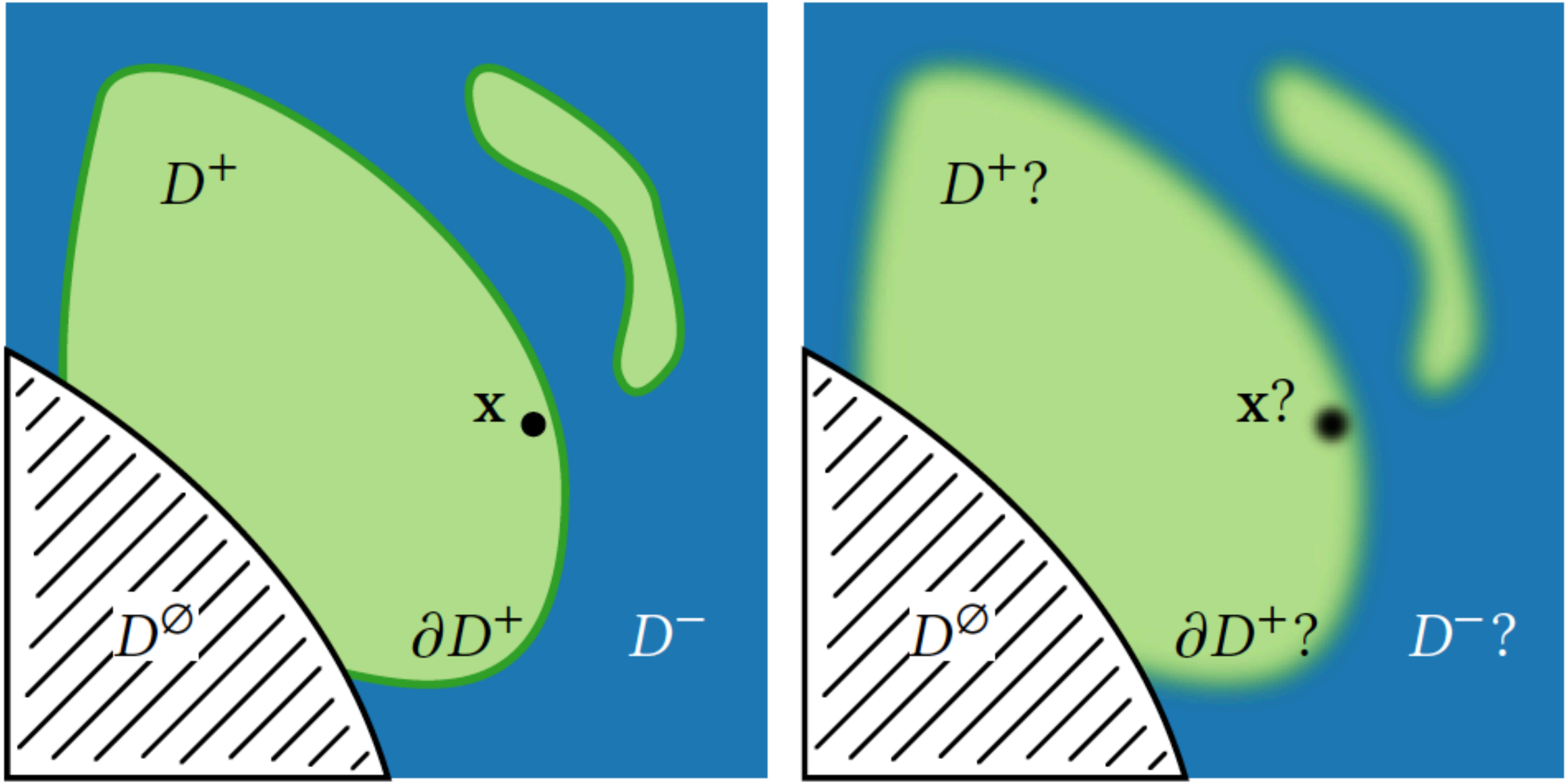




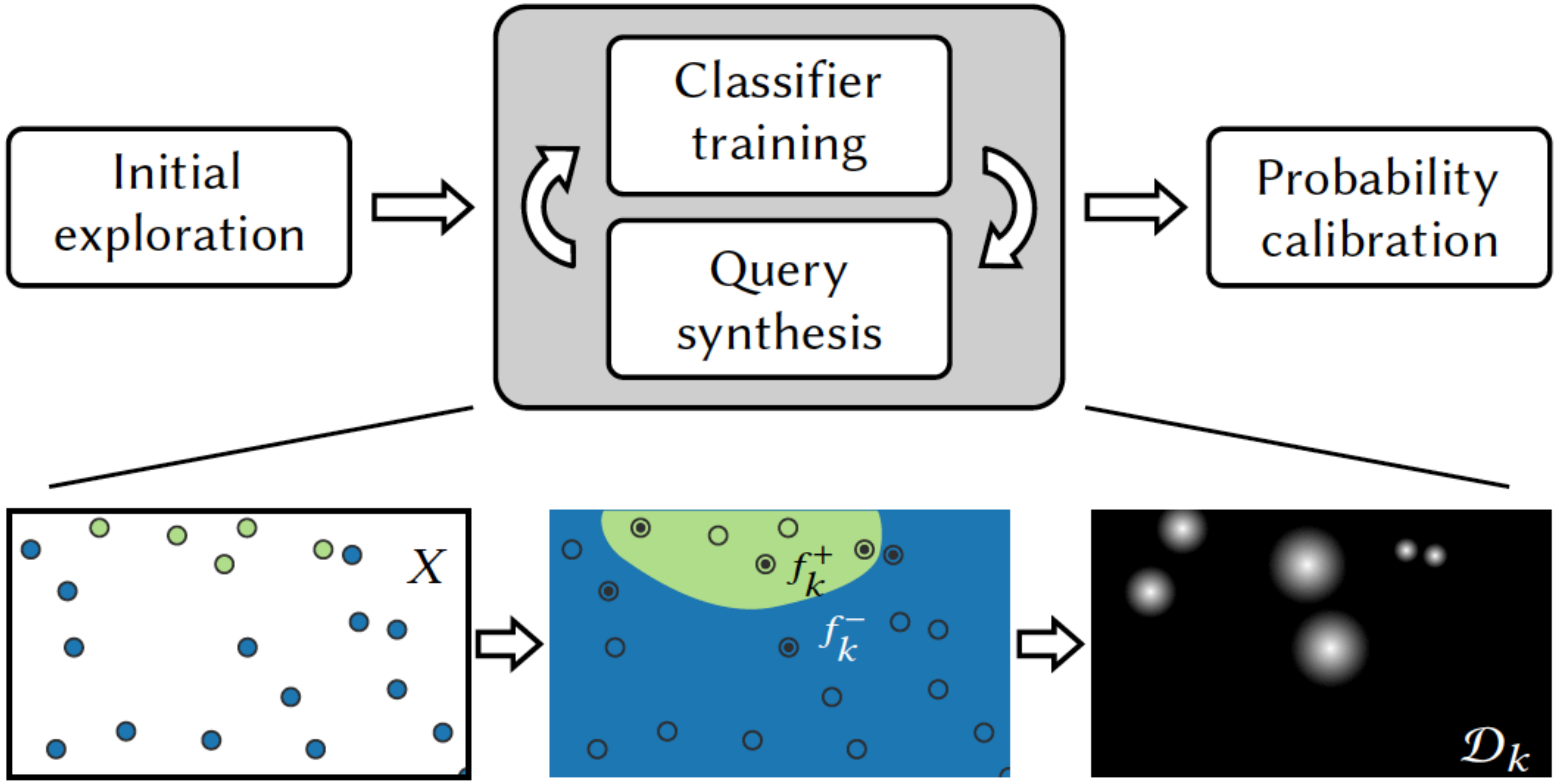
Modeling High-dimensional Design Spaces



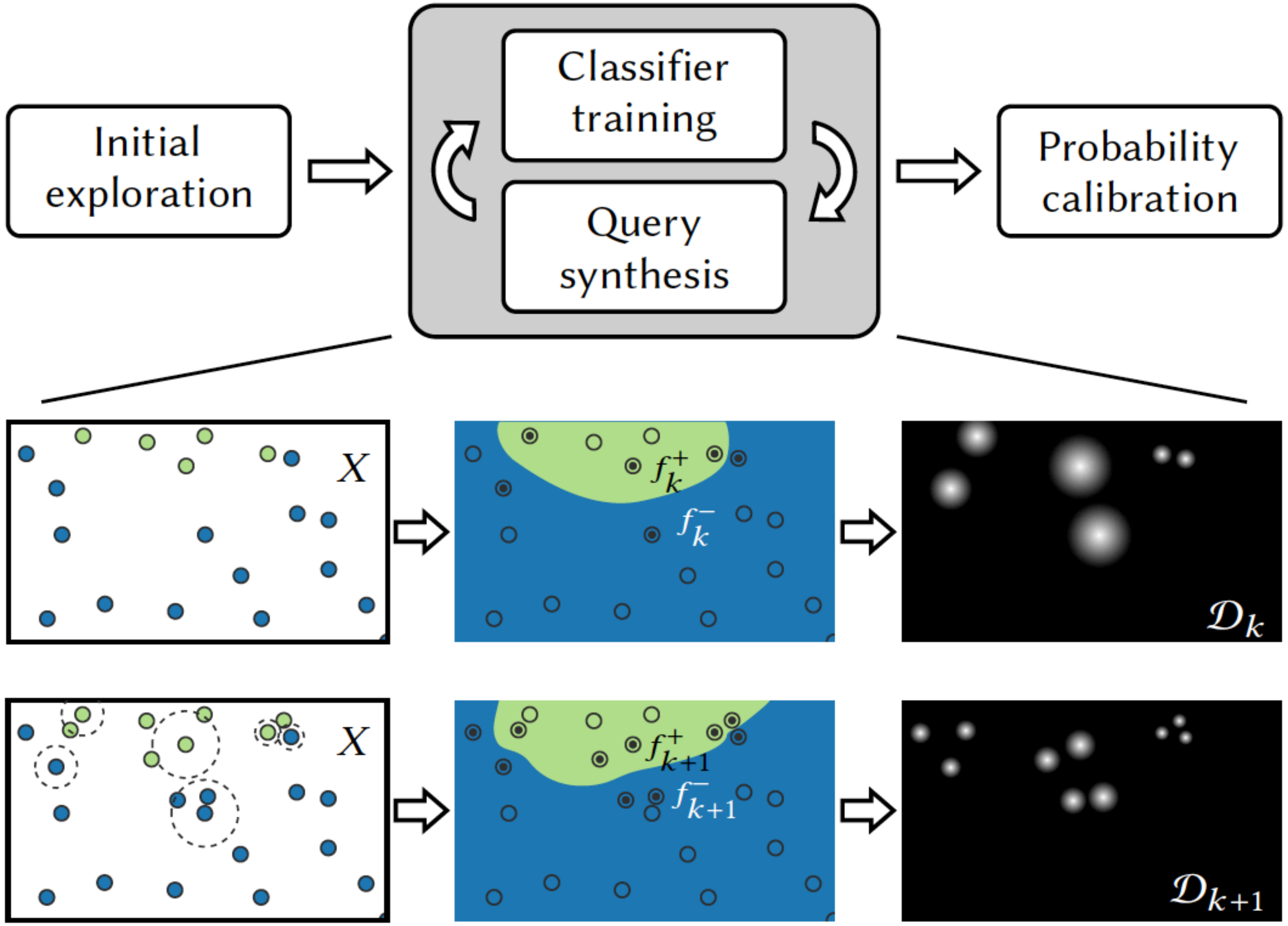
Modeling High-dimensional Design Spaces



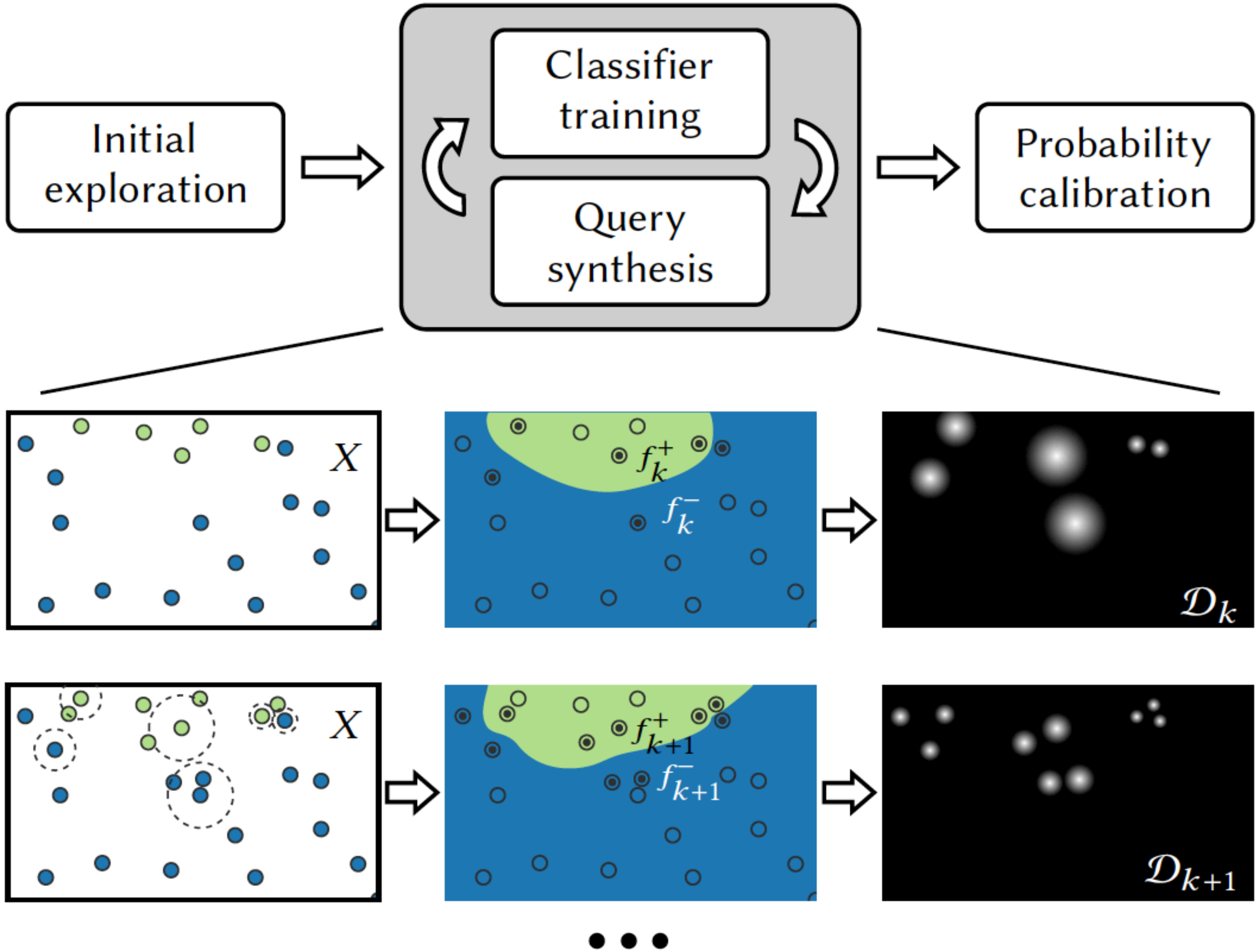
Online Modeling



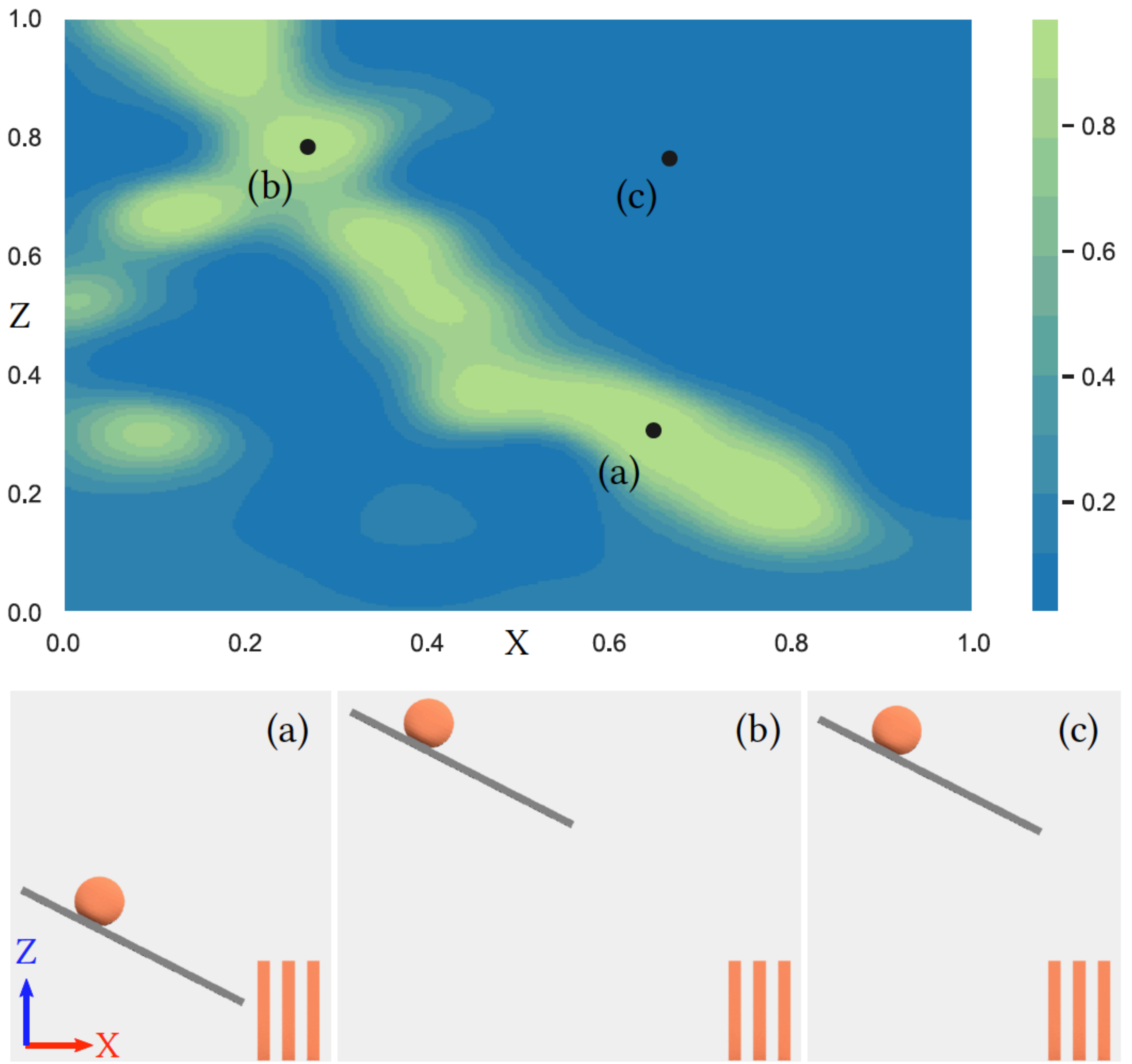
Online Modeling



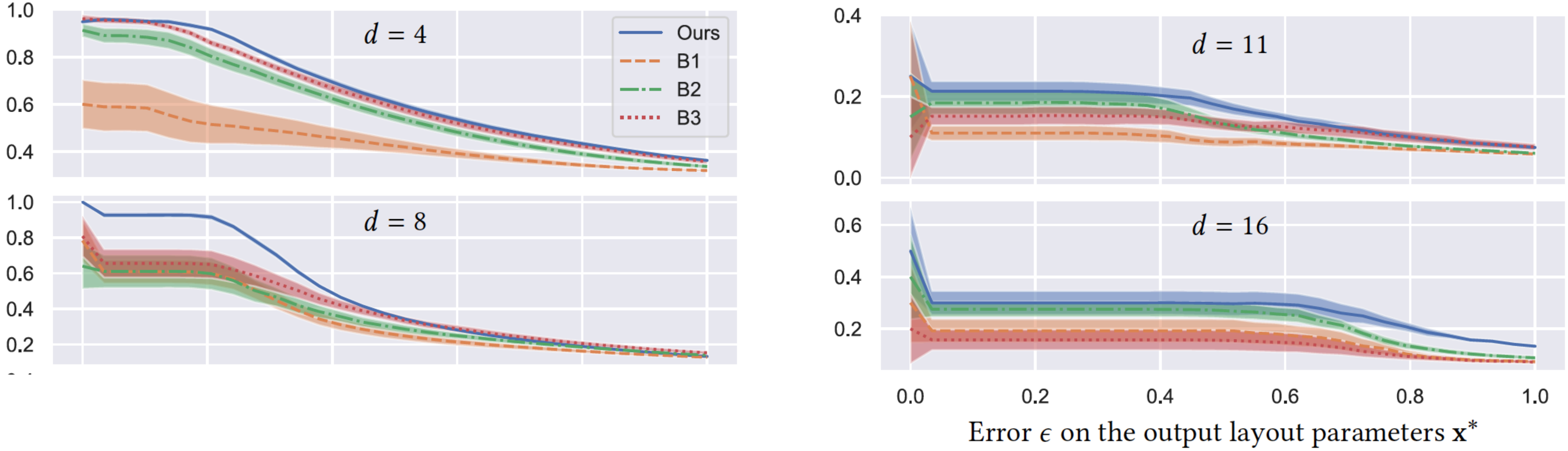
Online Modeling

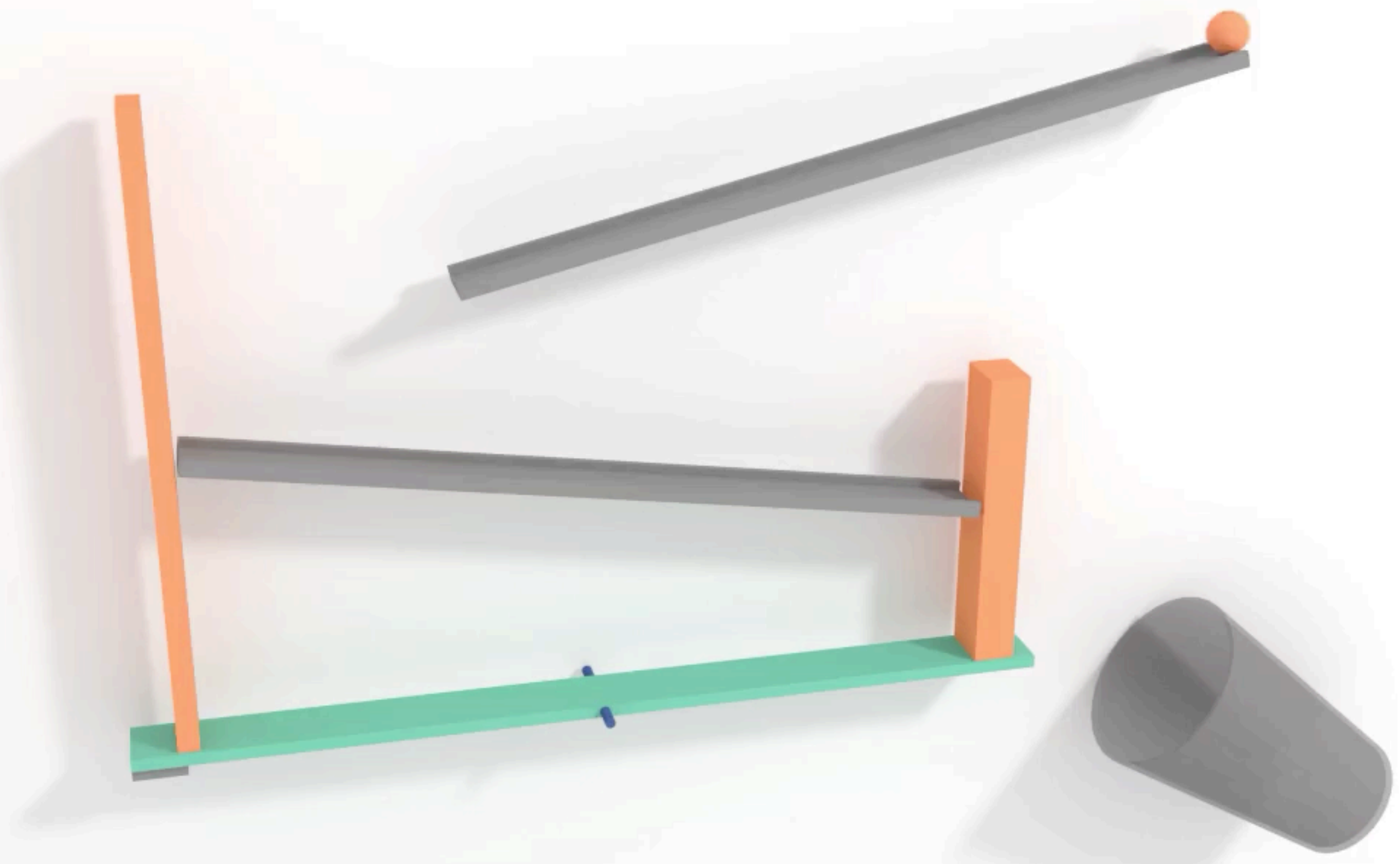


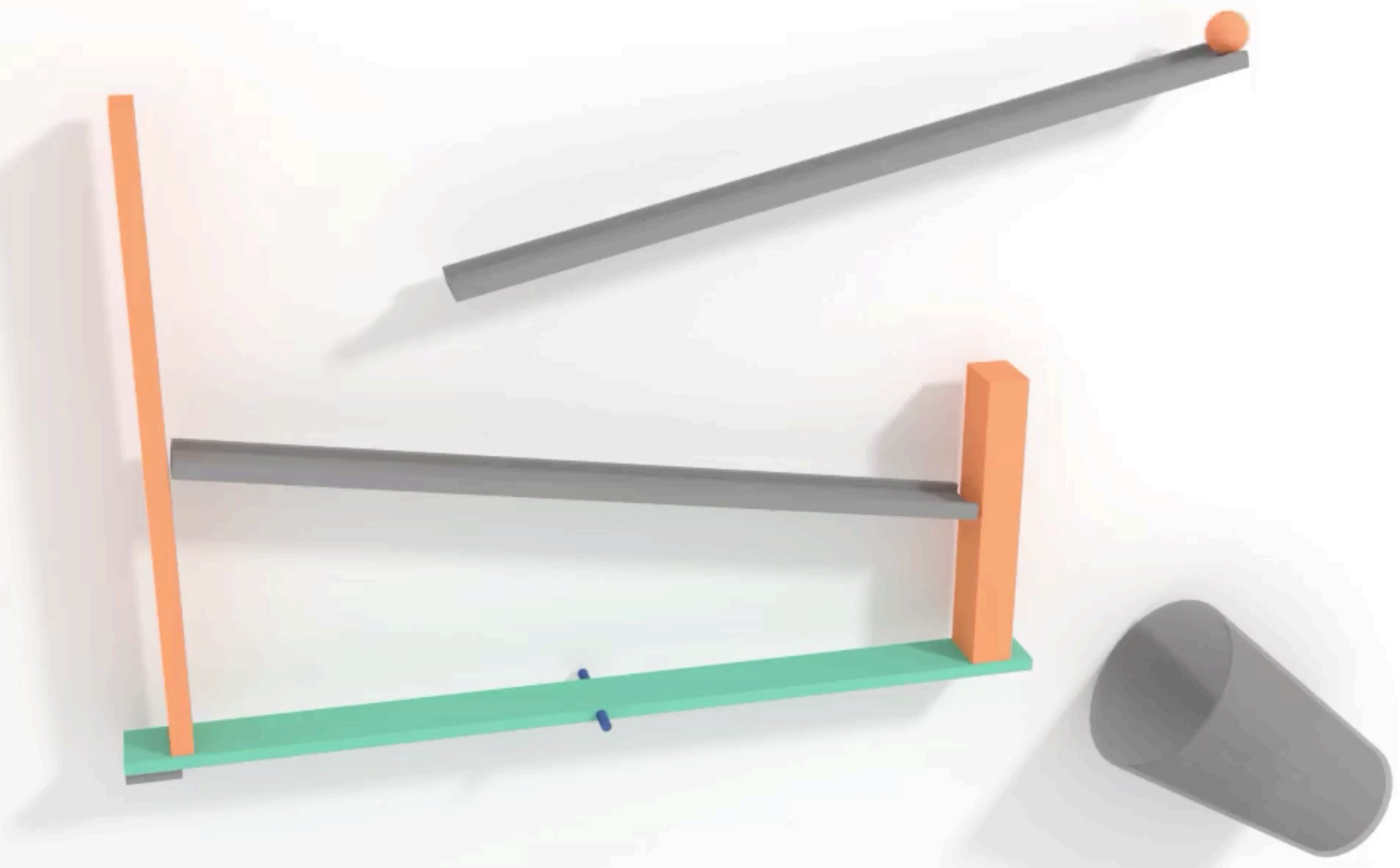
Simple Example



Average Robustness Estimates







Code Examples

PCA/SVD basis
Linear Regression
Polynomial Regression
Stochastic Gradient Descent vs. Gradient Descent
Multi-layer Perceptron
Edge Filter 'Network'
Convolutional Network
Filter Visualization
Weight Initialization Strategies
Colorization Network
Autoencoder
Variational Autoencoder
Generative Adversarial Network

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

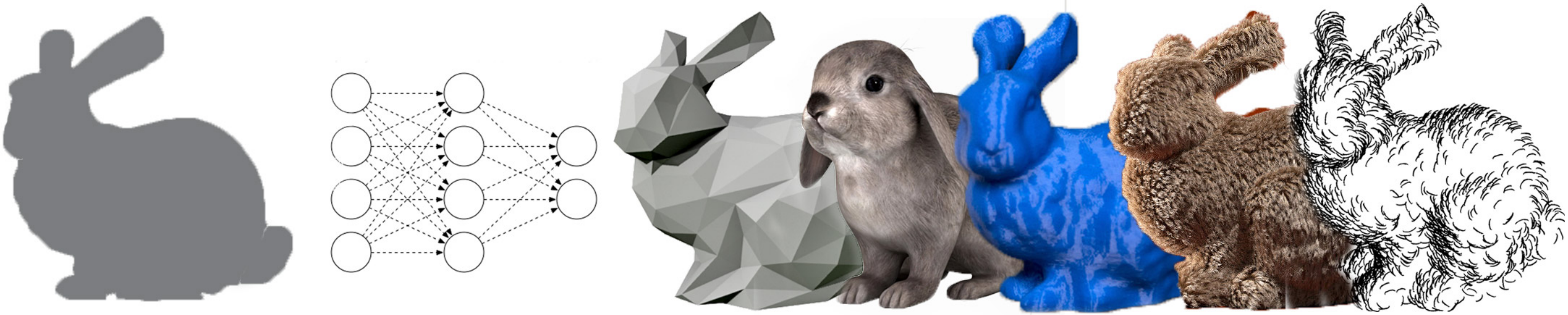
Scan me

Timetable

			Niloy	Federico	Iasonas	Emanuele
Theory/Basics	Introduction	9:00	X	X	X	X
	Machine Learning Basics	~ 9:05	X			
	Neural Network Basics	~ 9:35		X		
	Alternatives to Direct Supervision (GANs)	~11:00			X	
State of the Art	Image Domain	~11:45			X	
	3D Domains (extrinsic)	~13:30	X			
	3D Domains (intrinsic)	~ 14:15				X
	Physics and Animation	~ 16:00	X			
	Discussion	~ 16:45	X	X	X	X

Sessions: A. 9:00-10:30 (**coffee**) B. 11:00-12:30 [**LUNCH**] C. 13:30-15:00 (**coffee**) D. 15:30-17:00

Course Information (slides/code/comments)



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

Scan me