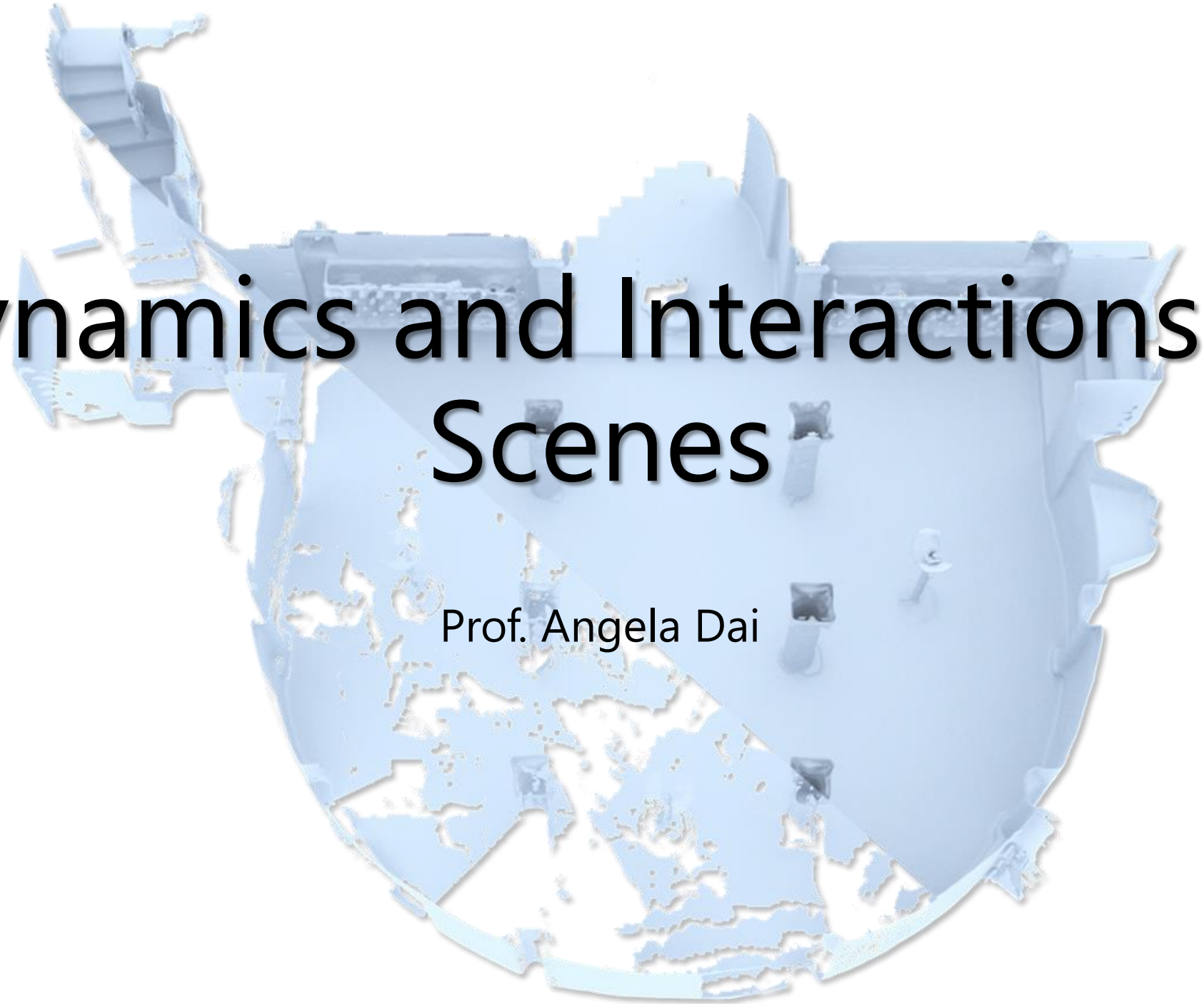
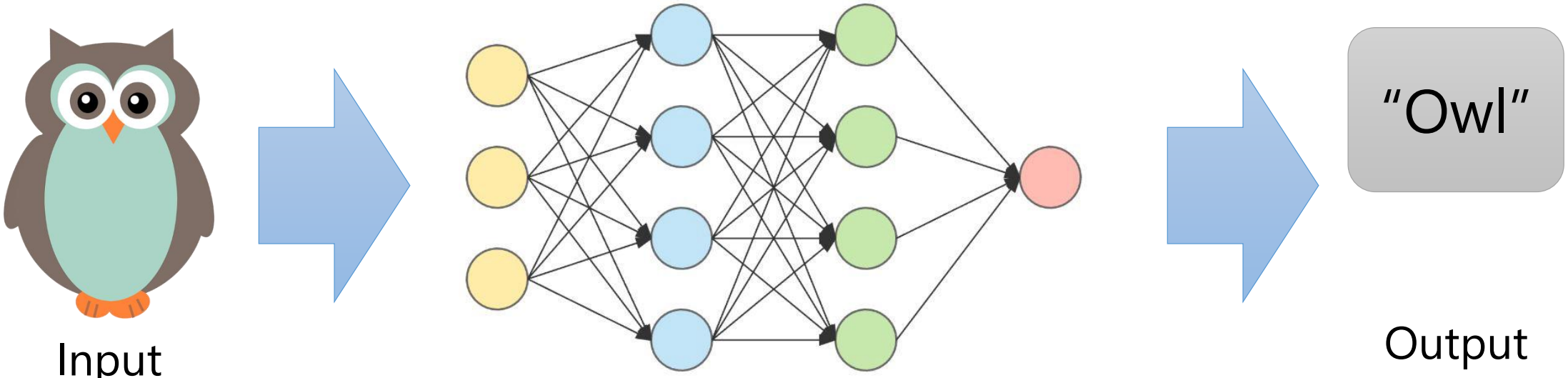
Dynamics and Interactions in Scenes



Prof. Angela Dai

Brief Recap

Deep Learning



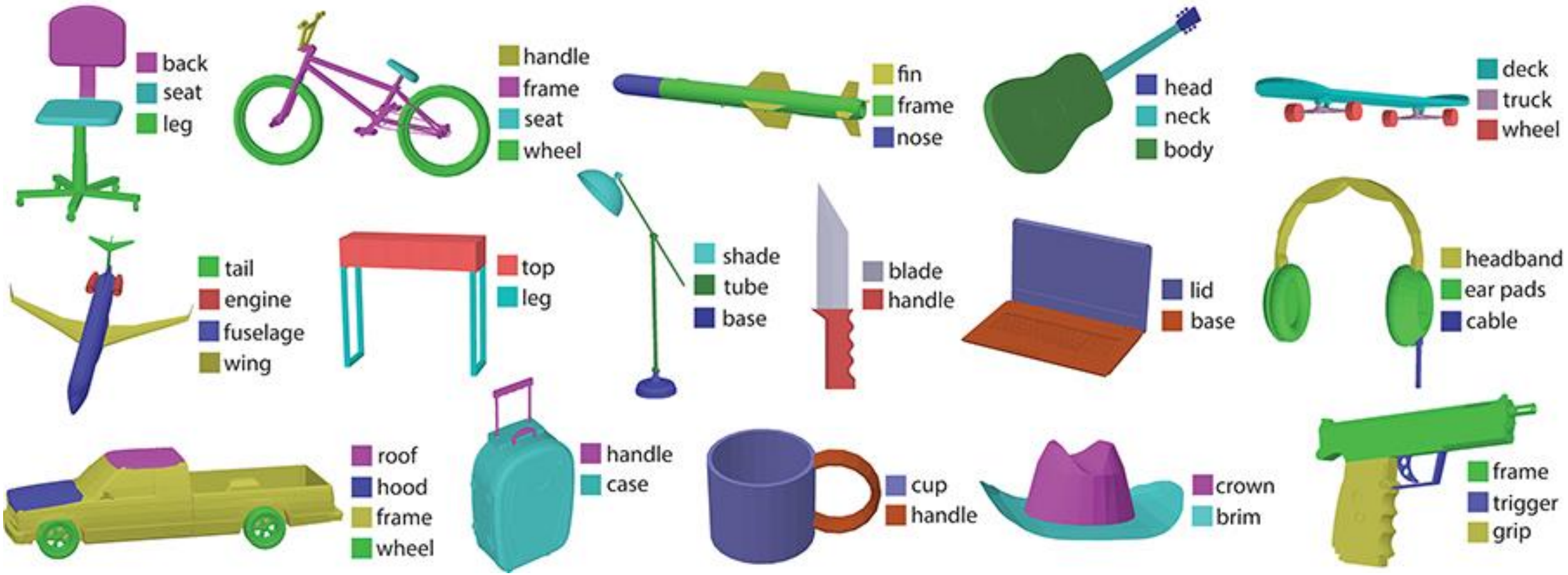
Input

Learned Feature Extraction
and Classification

Output

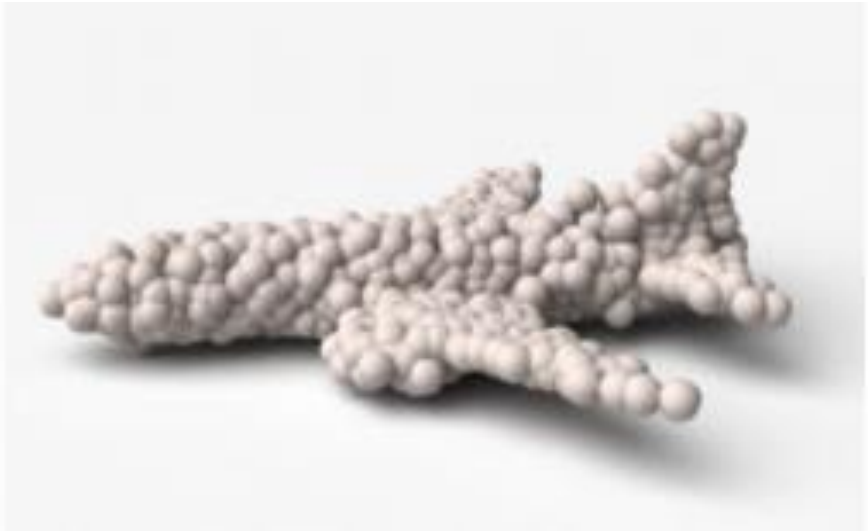
Want to automatically learn good feature representations for the task

Shape segmentation into parts

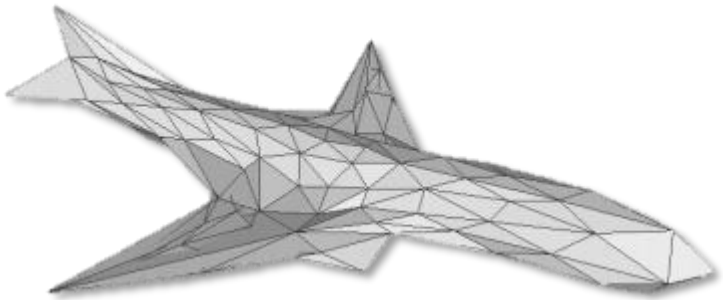


Generating Shapes

Signed Distance Fields

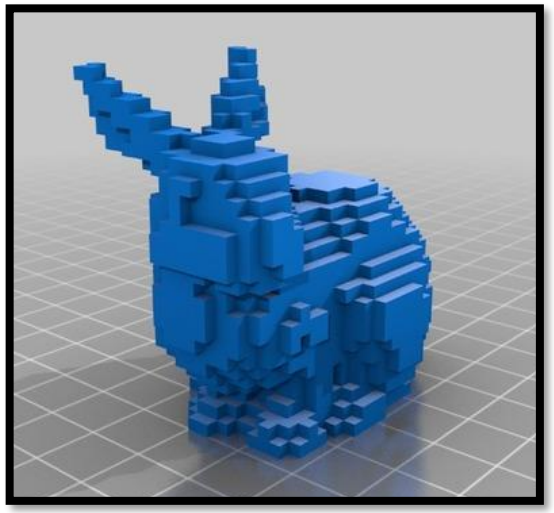


Point Clouds

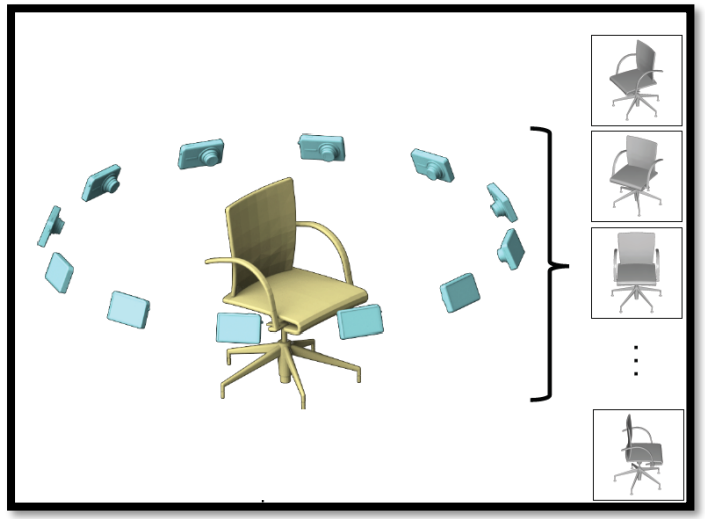


Meshes

3D Deep Learning by Representations

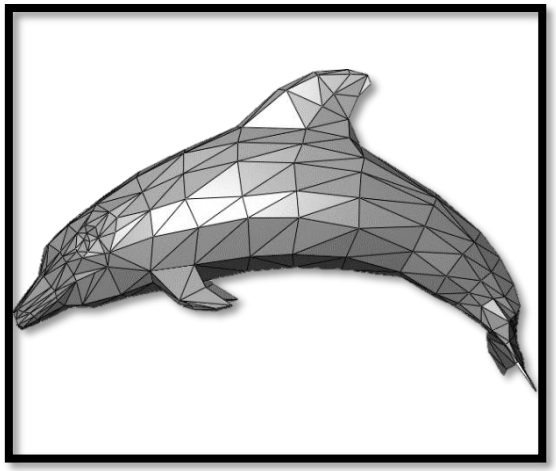


Volumetric
3D CNNs: Dense, Hierarchical, Sparse



Multi-View
(also: multi-view + volumetric/point/mesh)

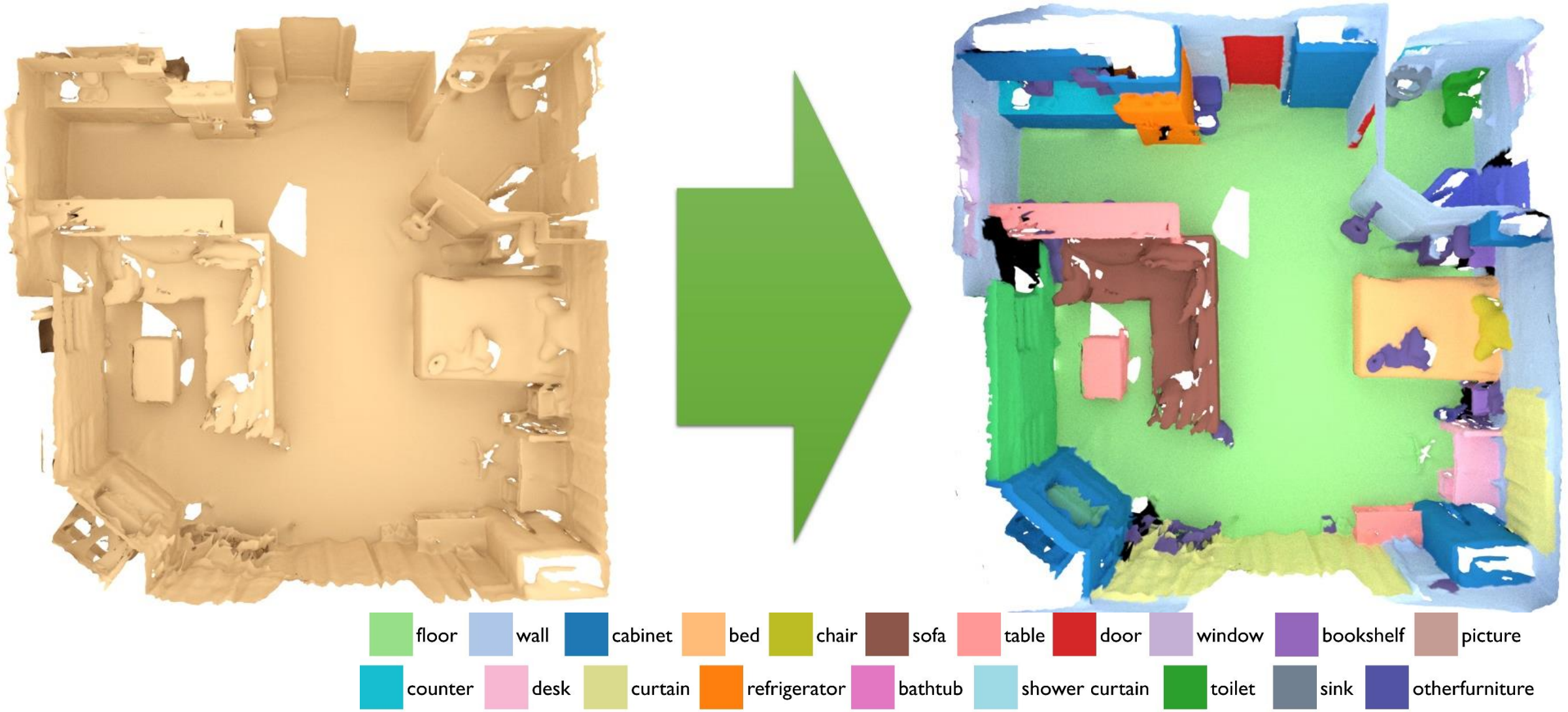
Point Cloud



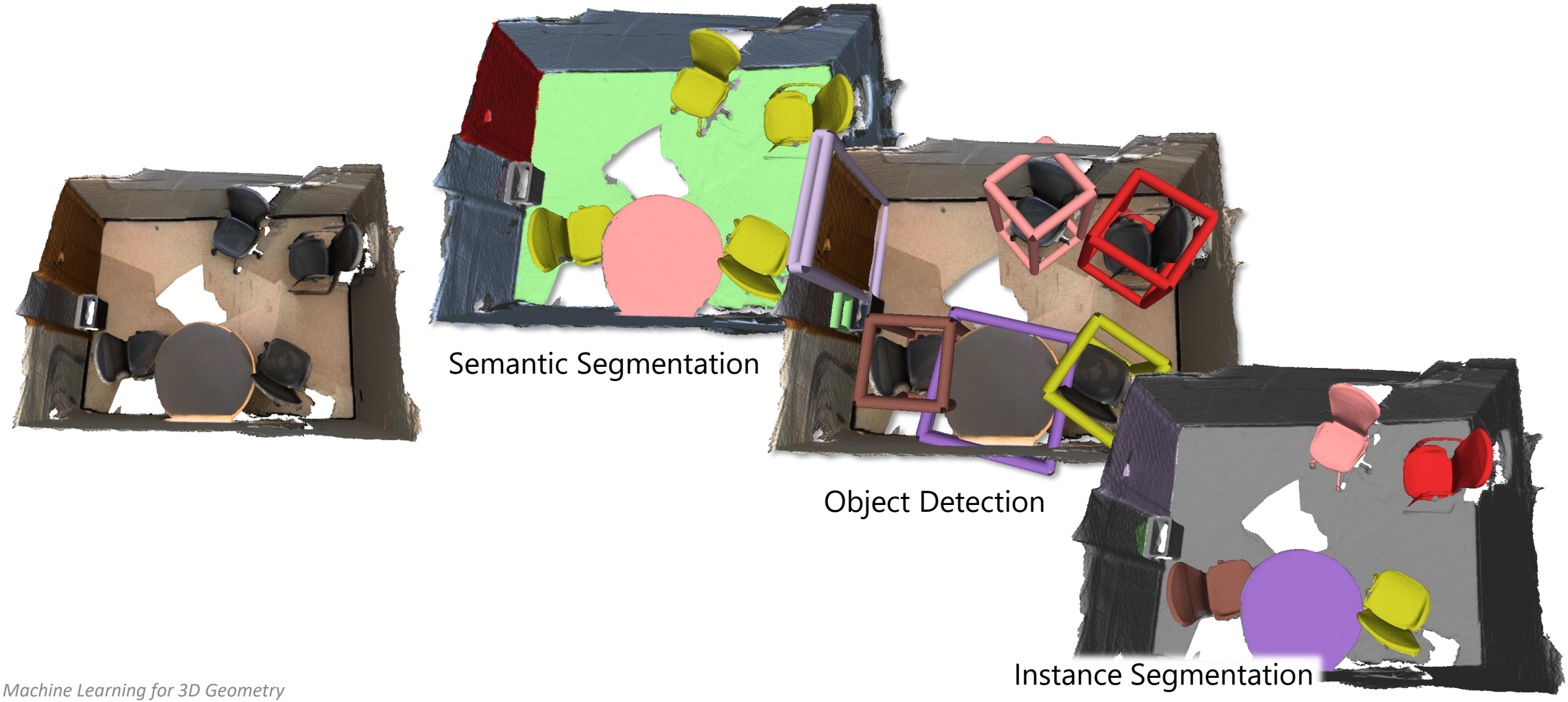
Mesh
Graph Neural Networks

and more!

3D Semantic Segmentation



Understanding object-ness



Generating 3D Scenes

Modeling Dynamics

- Consider non-rigid movement over time
- Applications: character animation, motion tracking

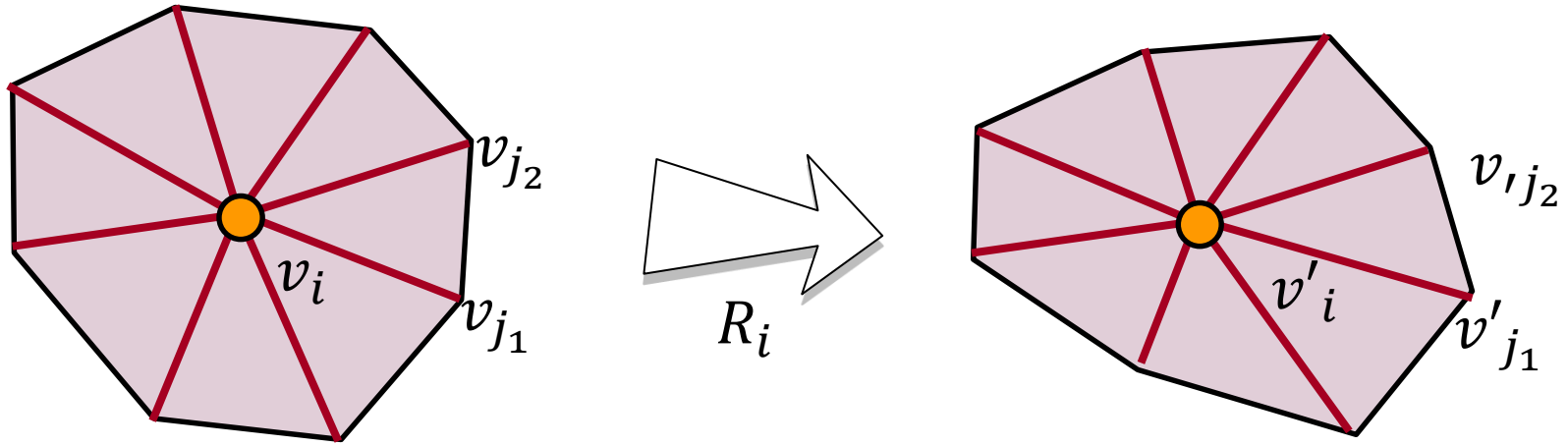
How to characterize motion?

- Move vertices of meshes

How to characterize motion?

- Moving one vertex should also move others
  - Most of the time, the neighboring vertices

- Recall: ARAP
  - maximize local rigidity



How to characterize motion?

- Moving one vertex should also move others
  - Most of the time, the neighboring vertices

- Recall: ARAP
  - Minimize deviation from rigidity

$$\min \sum_{i=1}^{n} \sum_{j \in N(i)} \left\| \left(v_i' - v_j' \right) - R_i \left(v_i - v_j \right) \right\|^2$$

Can we learn domain-specific priors?

- Consider one object type (e.g., human bodies, faces, hands, 4-legged animals, etc.)
- Movement is not free-form, limited by object physics (e.g., skeleton kinematics)
- Physics is very complex to model – learn dynamic priors

Domain-Specific Data-Driven Dynamics

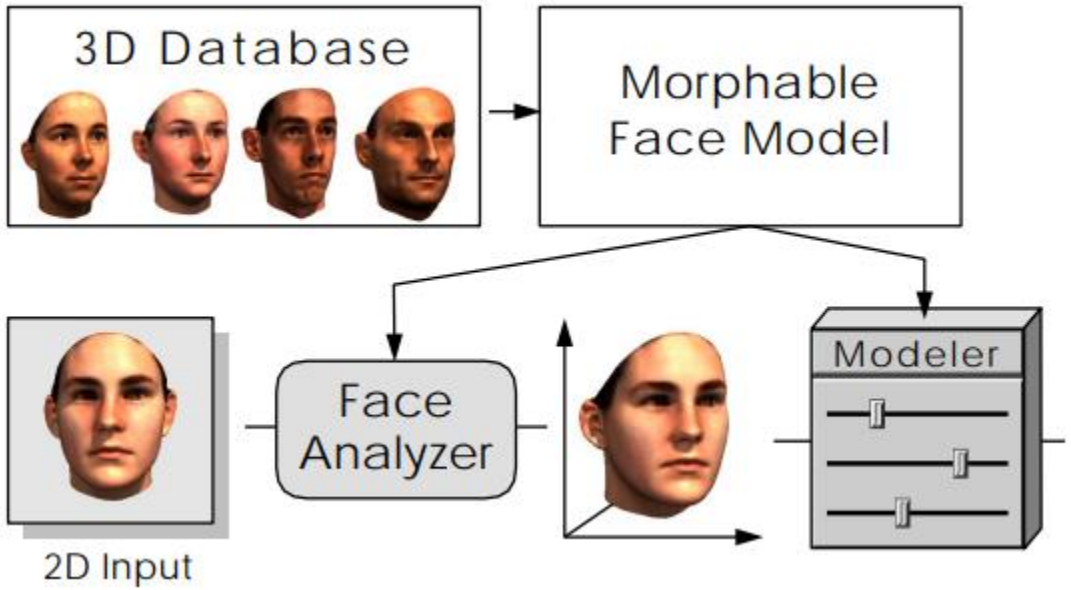
- Construct dynamic model from large set of objects of a class category

- Learn *parametric model*
  - Summarize data through set of fixed-size paramters
  - Parameters are independent of #train instances

  - Construction of parametric model:
    - Choose function form
    - Learn function parameters

Parametric Face Model

- 3D Morphable Face Model (Blanz and Vetter, 1999)
  - Collected 200 laser scans of faces (color + geometry)
  - Canonicalization of data collection:
    - All subjects wore bathing caps, no makeup or accessories, no facial hair



Parametric Face Model

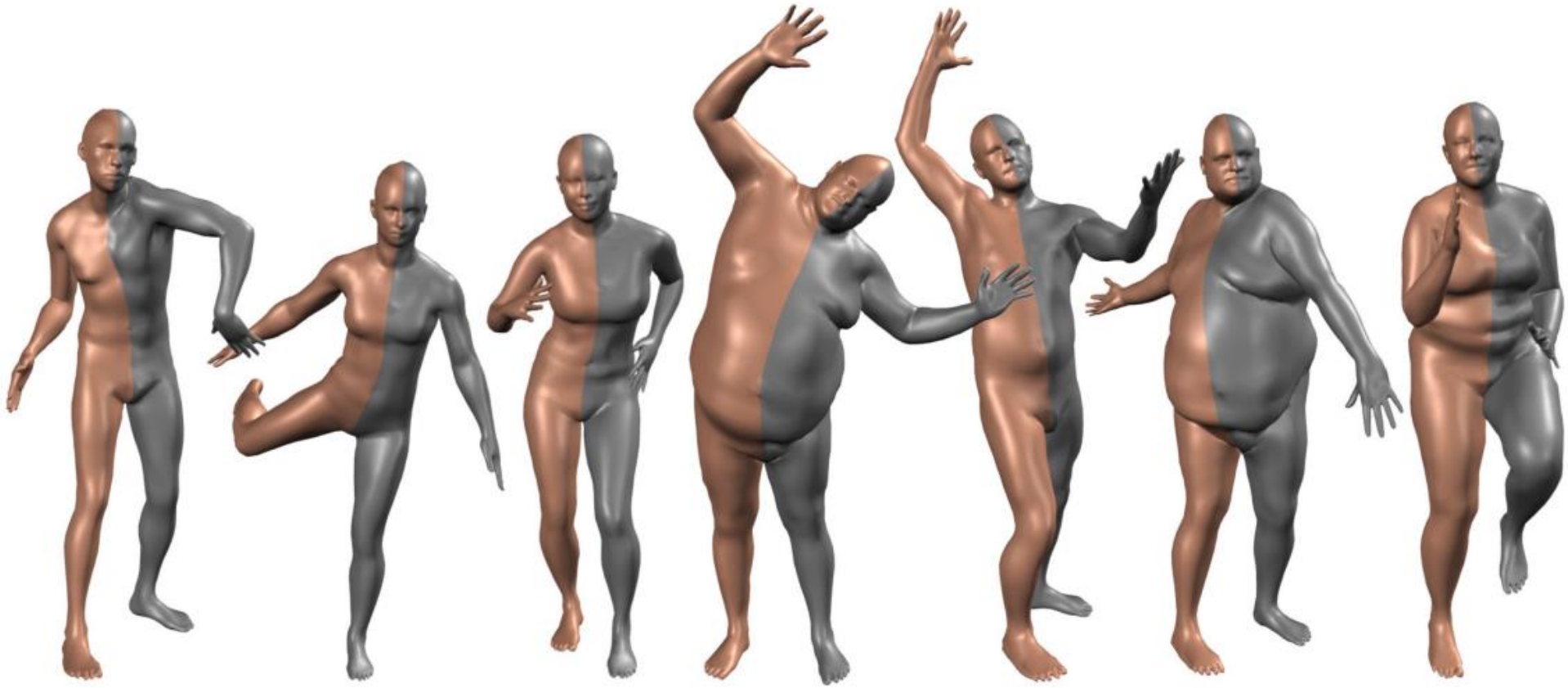
- 3D Morphable Face Model (Blanz and Vetter, 1999)
  - Collected 200 laser scans of faces (color + geometry)
  - Canonicalization of data collection:
    - All subjects wore bathing caps, no makeup or accessories, no facial hair
  - Align all data together (full correspondence)
  - Compute a linear generator function with PCA

$$c(w) = \bar{c} + Ew$$

  - $\bar{c}$: mean over train instances
  - $E \in \mathbb{R}^{3n \times d}$, d most dominant eigenvectors
  - w: low-dimensional parameter vector

Parametric Body Model

- SMPL (Loper et. al., 2015)

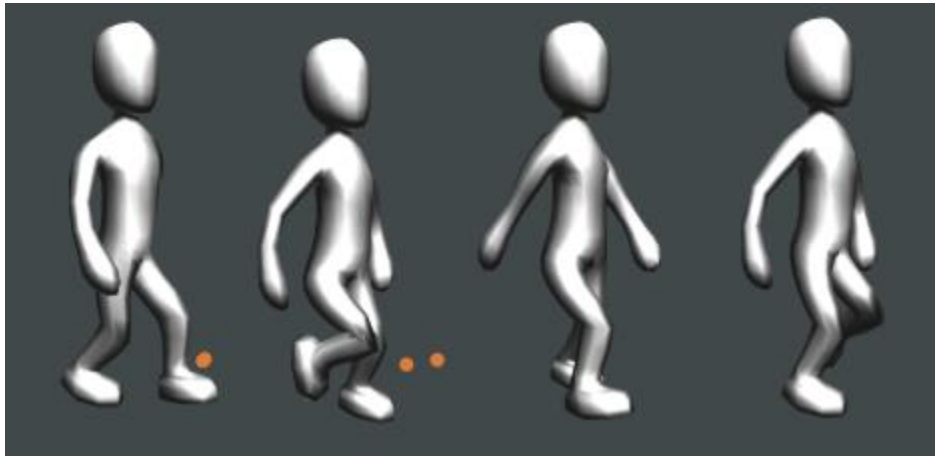


Parametric Body Model

- SMPL (Loper et. al., 2015)
  - Decompose body into identity-dependent shape and non-rigid pose-dependent shape
  - Data: ~1800 registrations of 40 individuals
  - Shape: PCA decomposition
  - Pose: how to parameterize?

How do artists animate 3D?

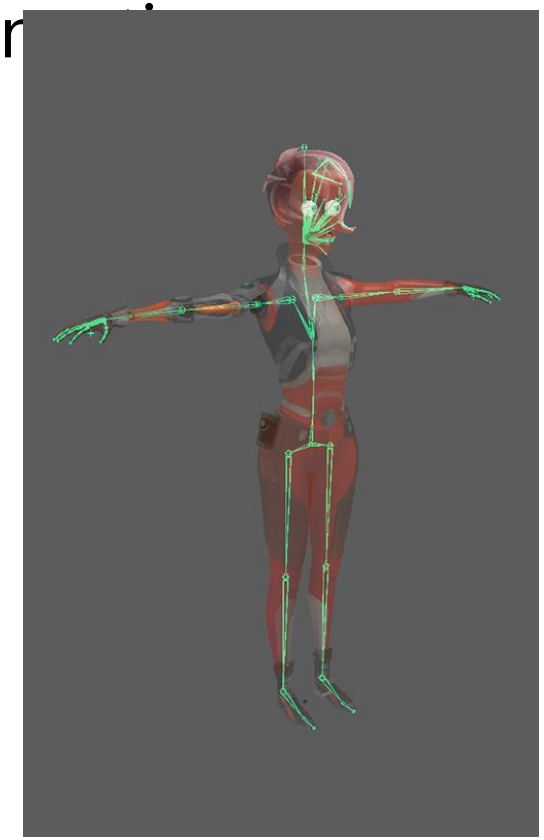
- Character moves based on its skeleton
- Define a skeleton and attach it to a character



https://tech.metail.com/introduction-to-skinning-and-3d-animation/

Creating a skeleton: rigging

- Artist defines a *rig*
- Rig: hierarchy of joints (locations) and their conn
- Hierarchy: joint inherits transforms of parents



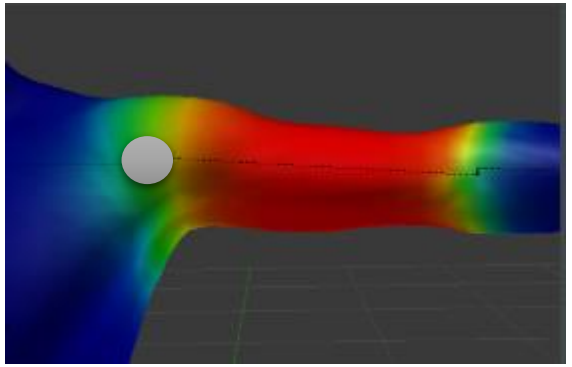
- Can also contain other attributes, e.g., orientation

- No standard single way to rig an object

https://3d-ace.com/blog/character-rigging-how-why-and-where-you-can-use-it/

Influence of skeleton: weight painting

- Describe how much each joint affects each mesh vertex
- Artist paints each vertex of the mesh with a weight $w_i \in [0,1]$ per joint
- Sum of weights per vertex is 1



https://tech.metail.com/introduction-to-skinning-and-3d-animation/

Transforming mesh by rig: skinning

- Transform mesh vertex positions according to a defined rig
- Various skinning methods
- Common: linear blending

$$v_i' = \sum_{k=1}^{K} w_{k,i} B_k \, v_i$$

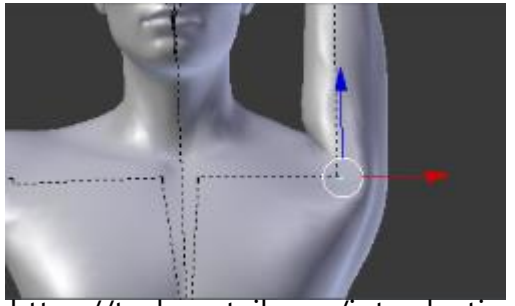
transformed vertex

weight for vertex i and bone k

bone transform

original vertex

- Note: no preservation of volume



https://tech.metail.com/introduction-to-skinning-and-3d-animation/

Parametric Body Model

- Shape: linear function B_S

$$B_S(\beta; S) = \sum_{n=1}^{M} \beta_n S_n \qquad \beta \in \mathbb{R}^M, S_n \in \mathbb{R}^{3N}$$

  - β: linear shape coefficients
  - S_n: PCA components of shape displacements

  - Compute S_n from registered training meshes
  - User can set β to specify desired shape

Parametric Body Model: SMPL

- Pose: map pose vector θ to vector of part relative rotation matrices, $R: \mathbb{R}^D \rightarrow \mathbb{R}^{9K}$
  - K: number of joints
  - D: dimension of pose vector
  - Non-linear mapping due to application of rotations (sin, cos)
  - Linear in $R^*(\theta) = R(\theta) - R(\theta^*), \theta^*$ is the rest pose

$$B_P(\theta; P) = \sum_{n=1}^{9K} R_n(\theta) - R_n(\theta^*) P_n$$

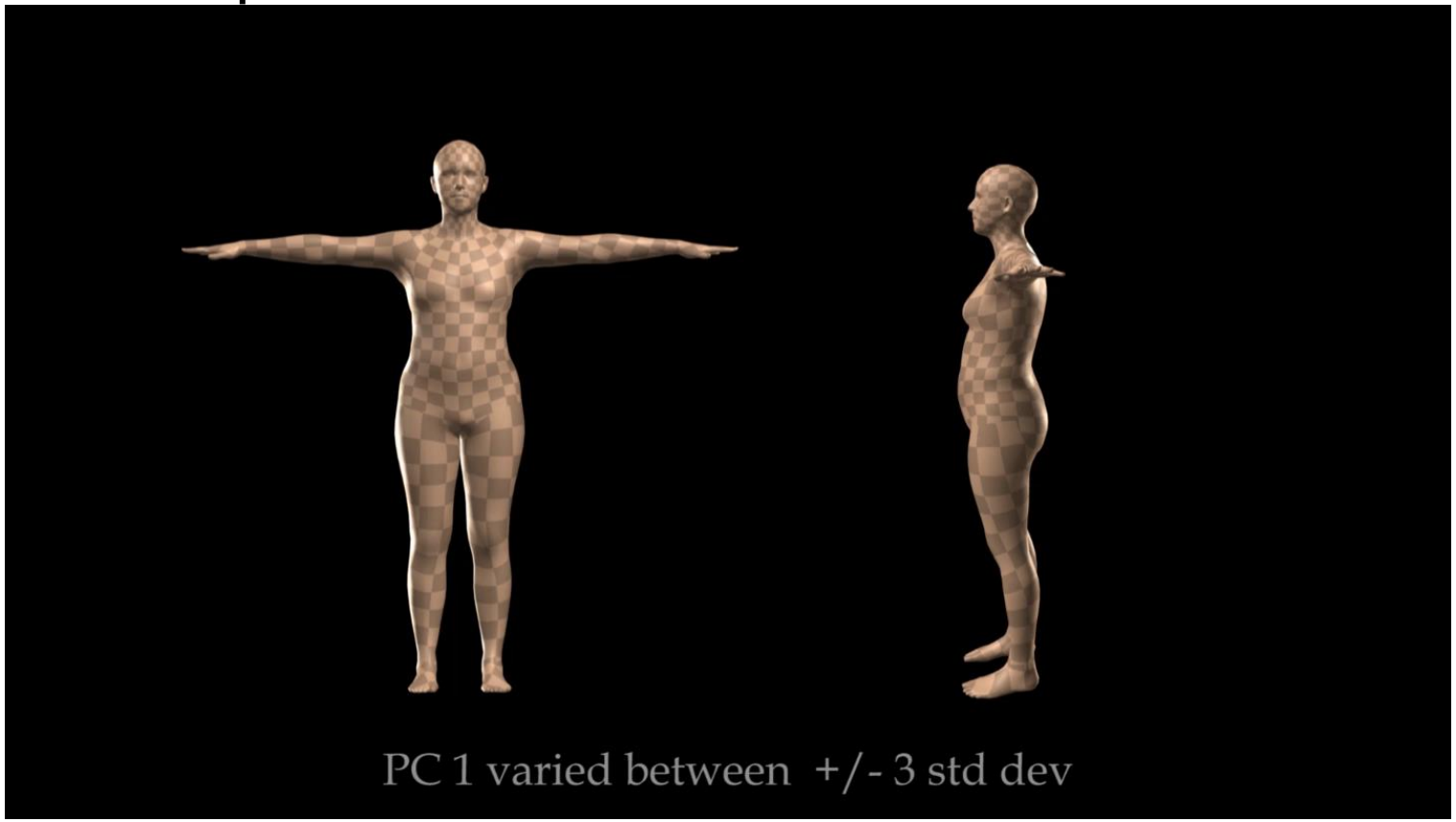
  - $P_n \in \mathbb{R}^{3N}$: vectors of vertex displacements

Parametric Body Model: SMPL

- Optimize over train meshes
- Energy: mean squared distance to train mesh vertices

Parametric Body Model: SMPL

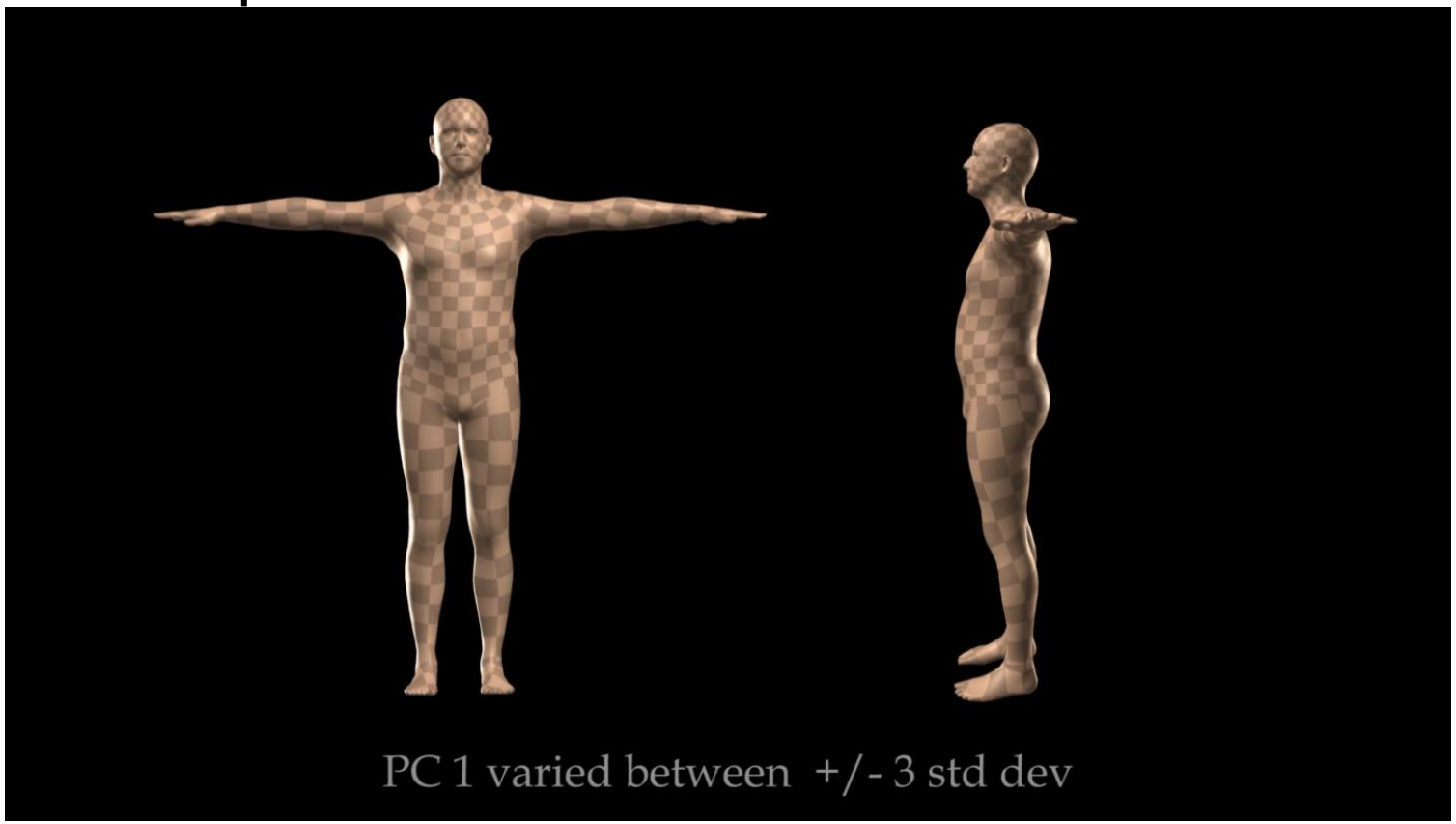
- Optimize over train meshes
- Energy: mean squared distance to train mesh vertices



Shape Space: Female

Parametric Body Model: SMPL

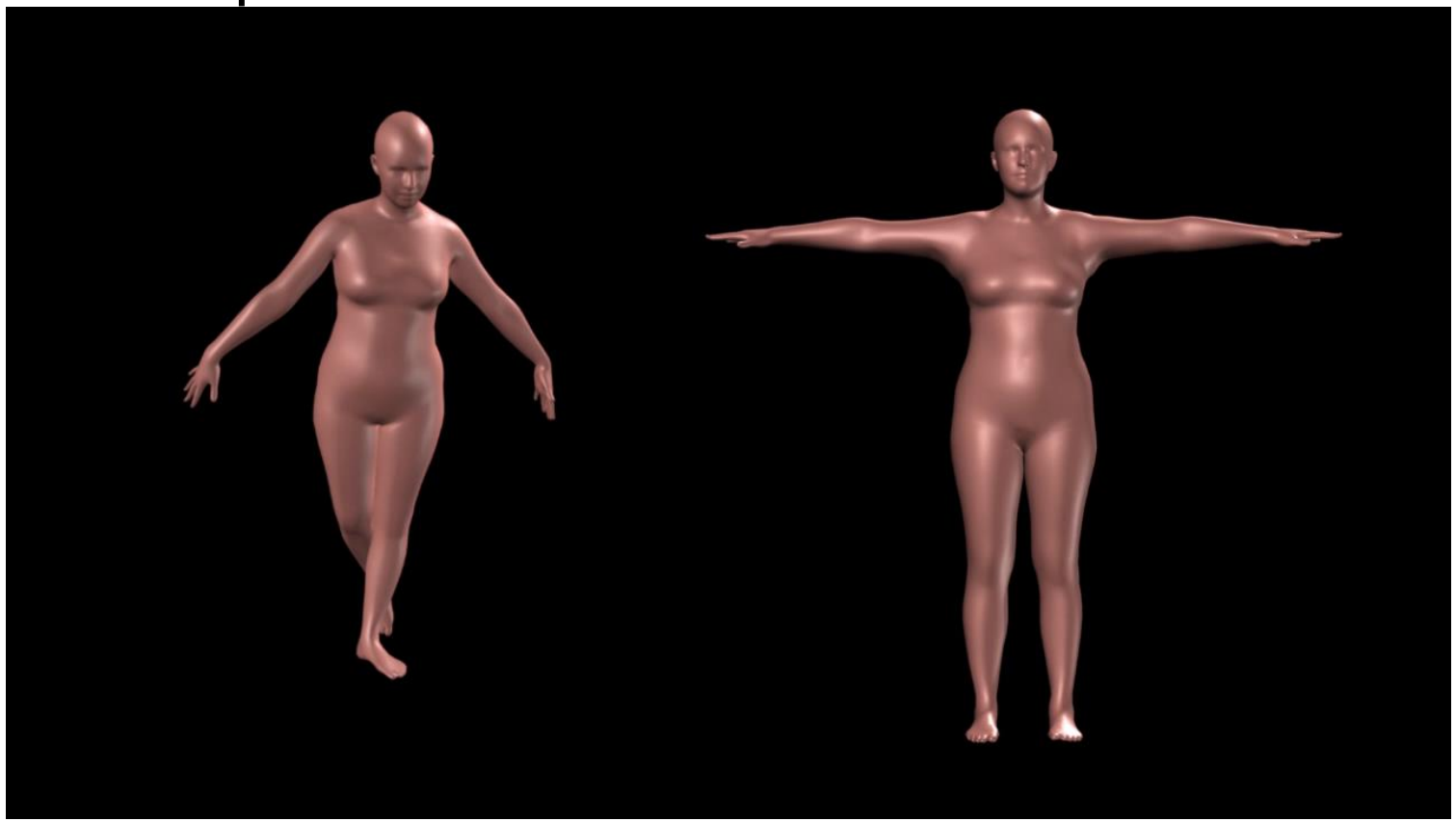
- Optimize over train meshes
- Energy: mean squared distance to train mesh vertices



Shape Space: Male

Parametric Body Model: SMPL

- Optimize over train meshes
- Energy: mean squared distance to train mesh vertices



Pose Space

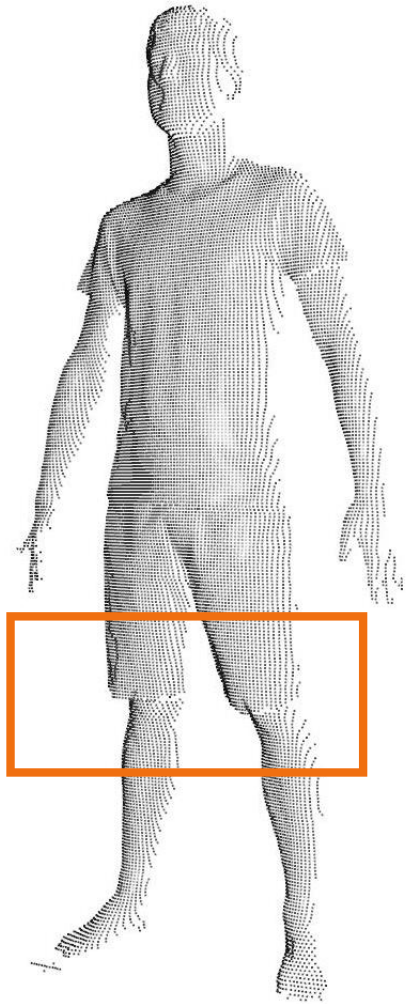
Traditional Parametric Models

- Enable significant progress in modeling deformable objects
- Compatible with modern graphics pipelines

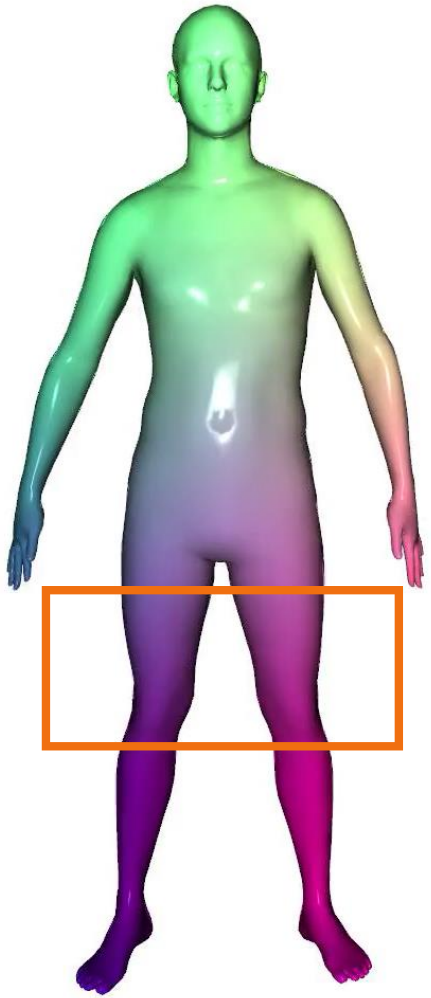
- Complex construction process for data, annotations, correspondence

- Difficult to represent complex, fine-scale details (e.g., clothing, hair, etc.)

Traditional Parametric Models

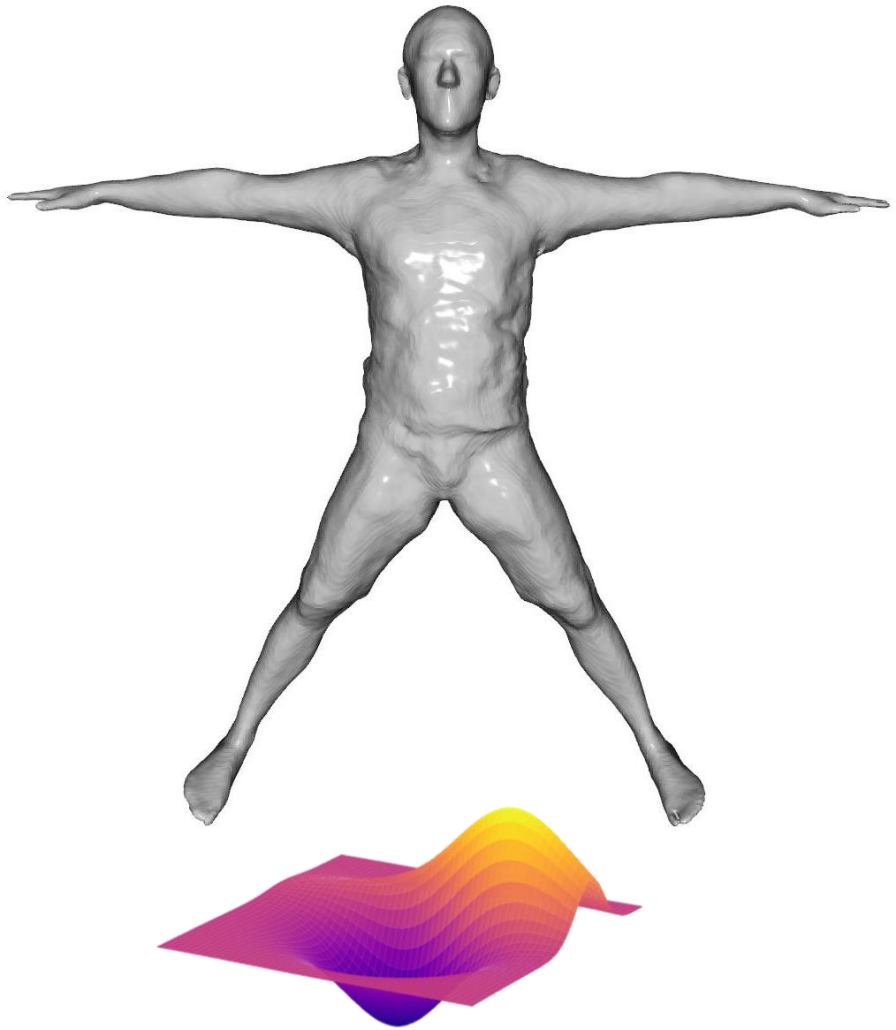


Depth Input



SMPL
[Loper et al. 2015]

Neural Parametric Models

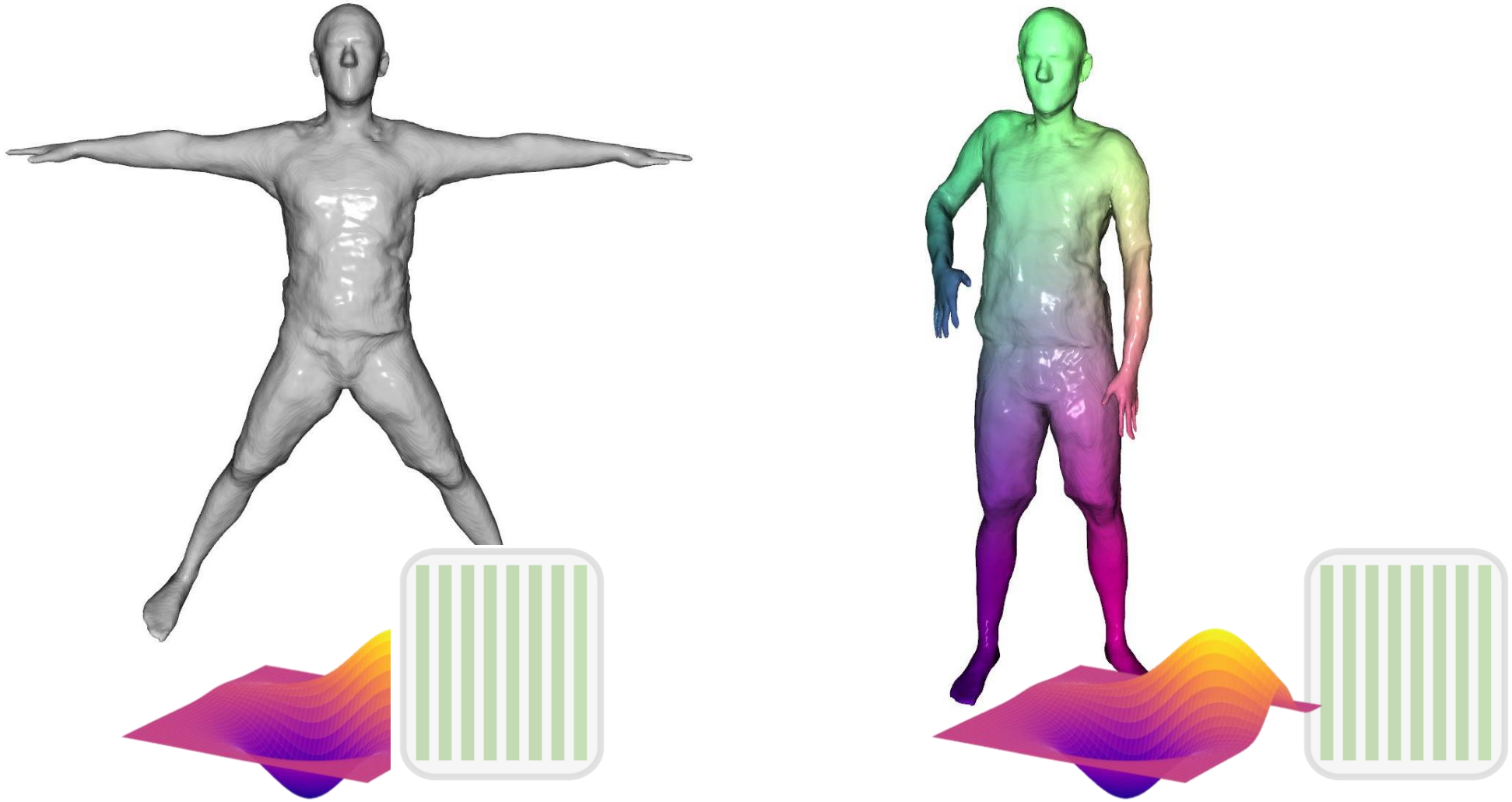


Shape Space

Pose Space

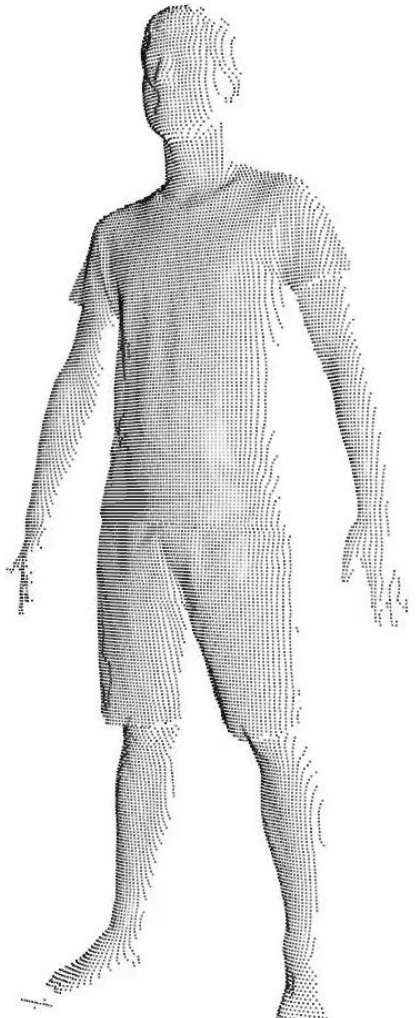
[Palafox et al. ‘21] NPMs

Neural Parametric Models

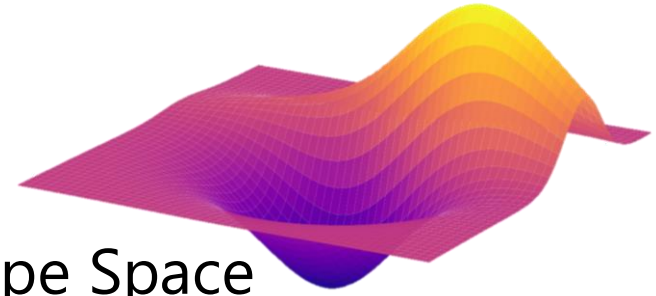


[Palafox et al. ‘21] NPMs

Neural Parametric Models



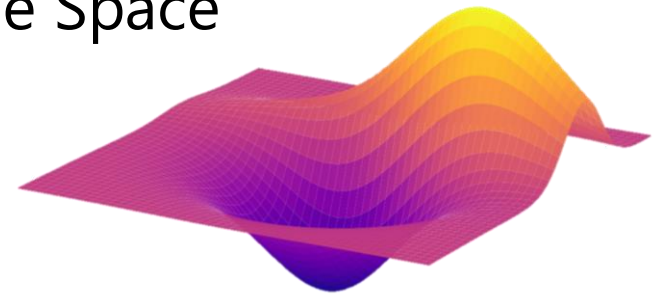
Input Depth Sequence

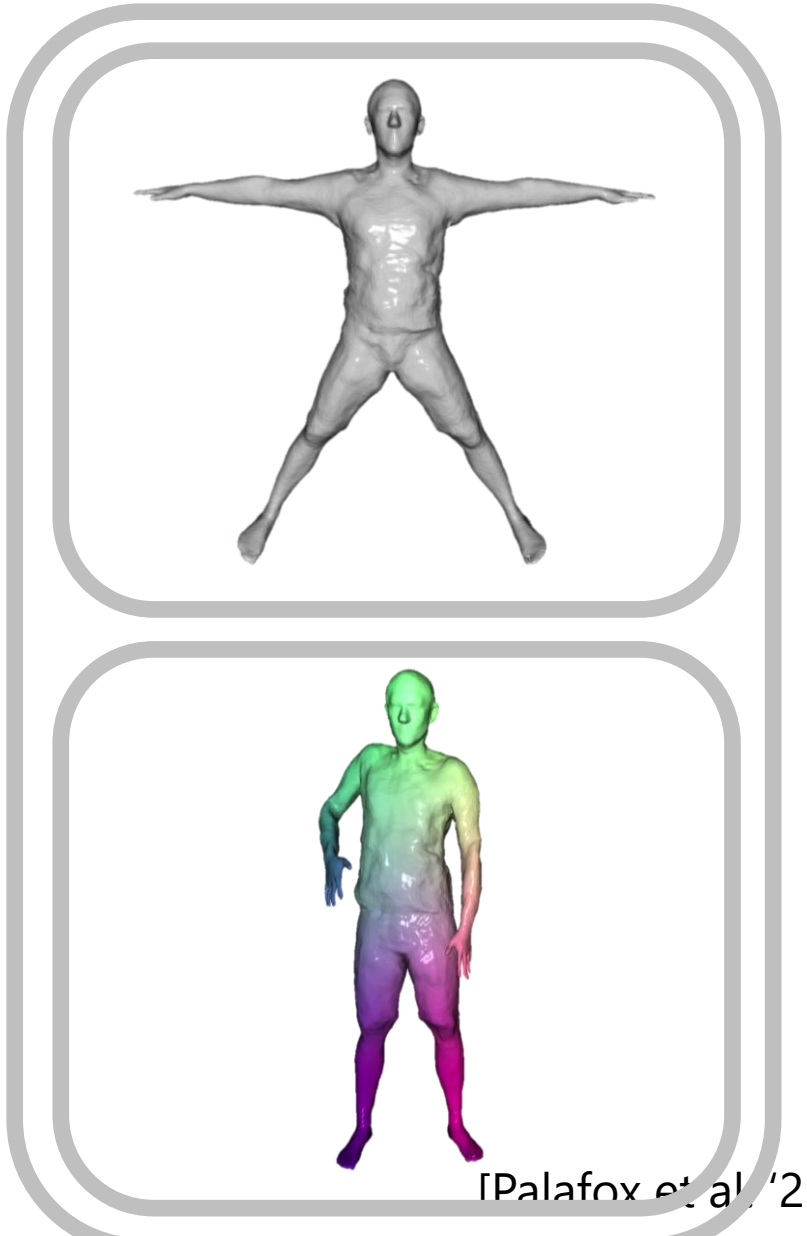


Shape Space

Latent Code Optimization

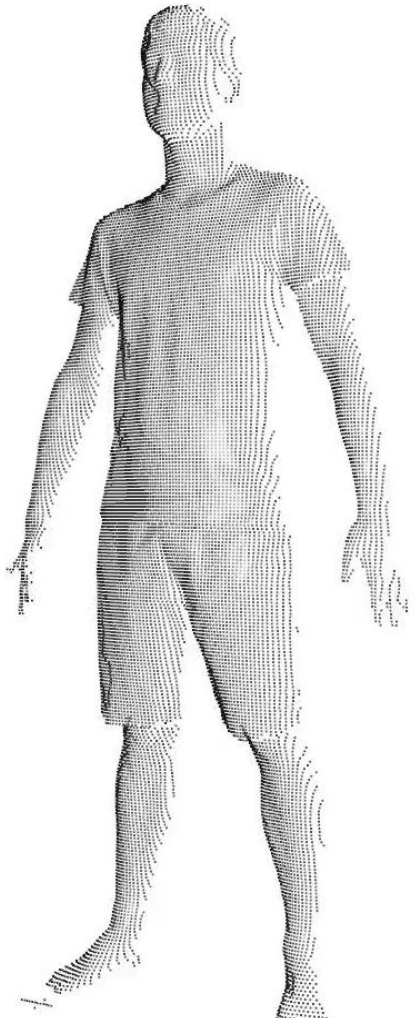
Pose Space



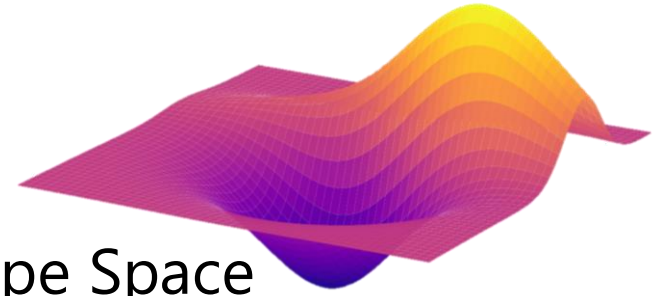


[Palafox et al. '21] NPMs

Neural Parametric Models



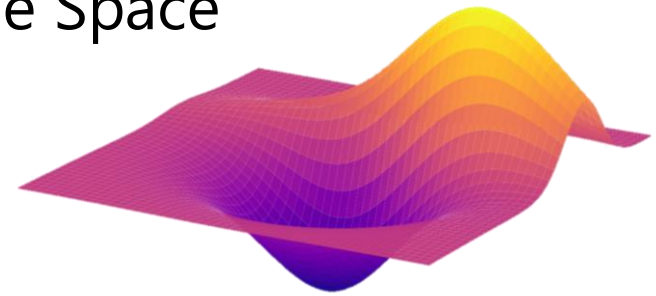
Input Depth Sequence

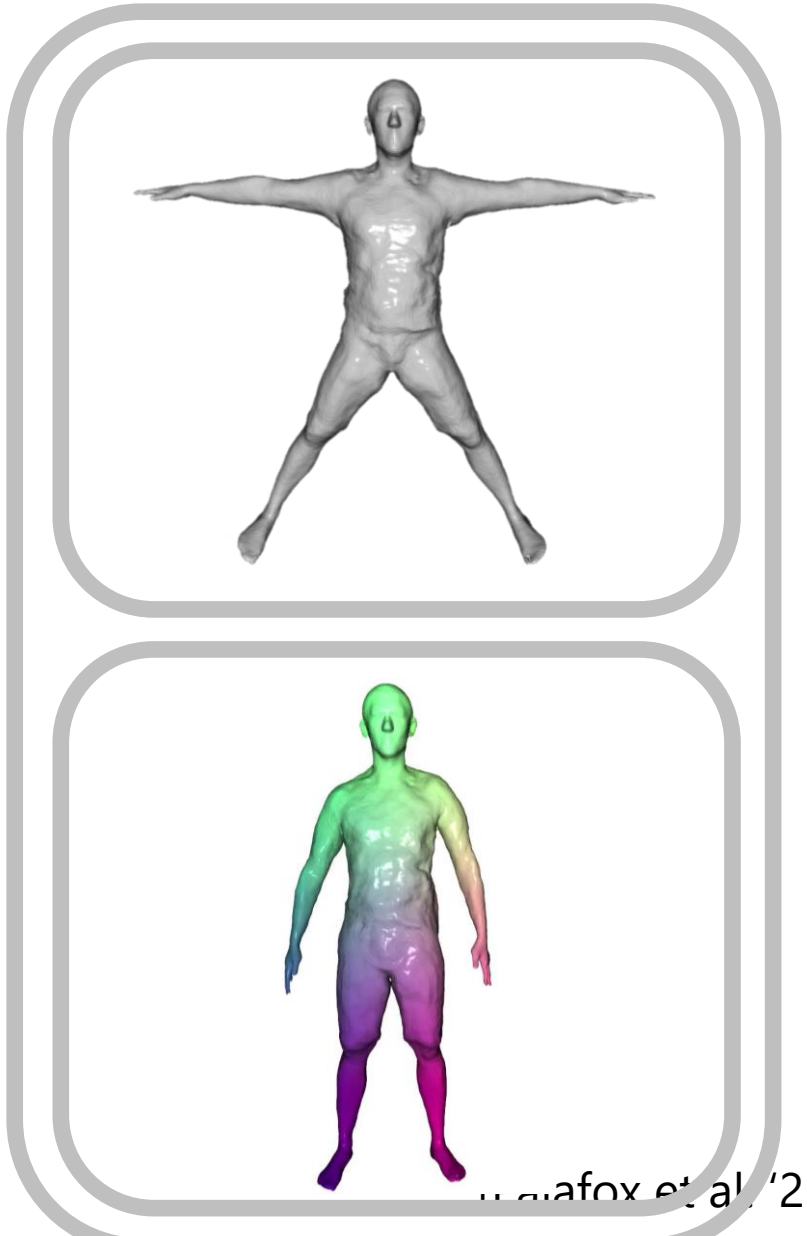


Shape Space

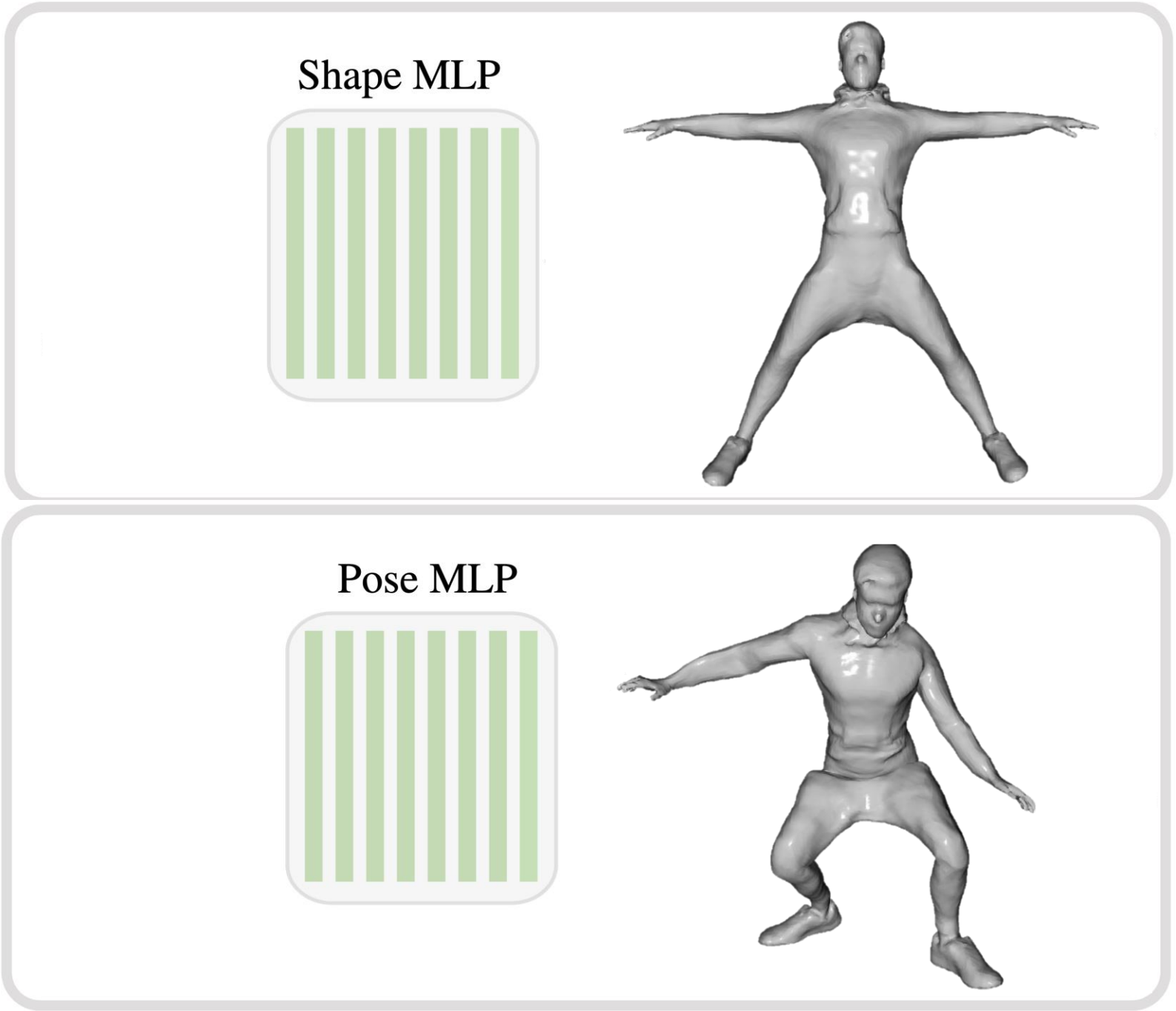
Latent Code Optimization

Pose Space





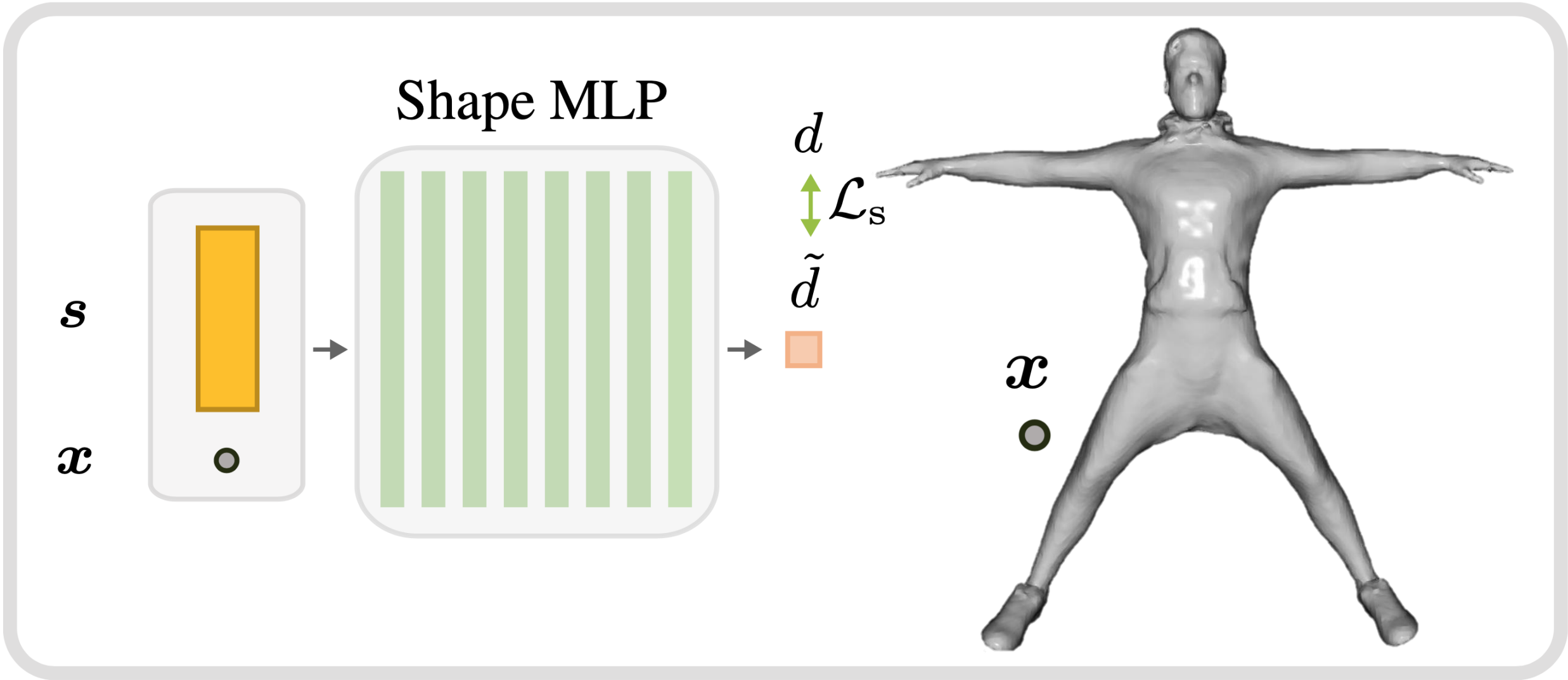
[Yifan et al. '21] NPMs



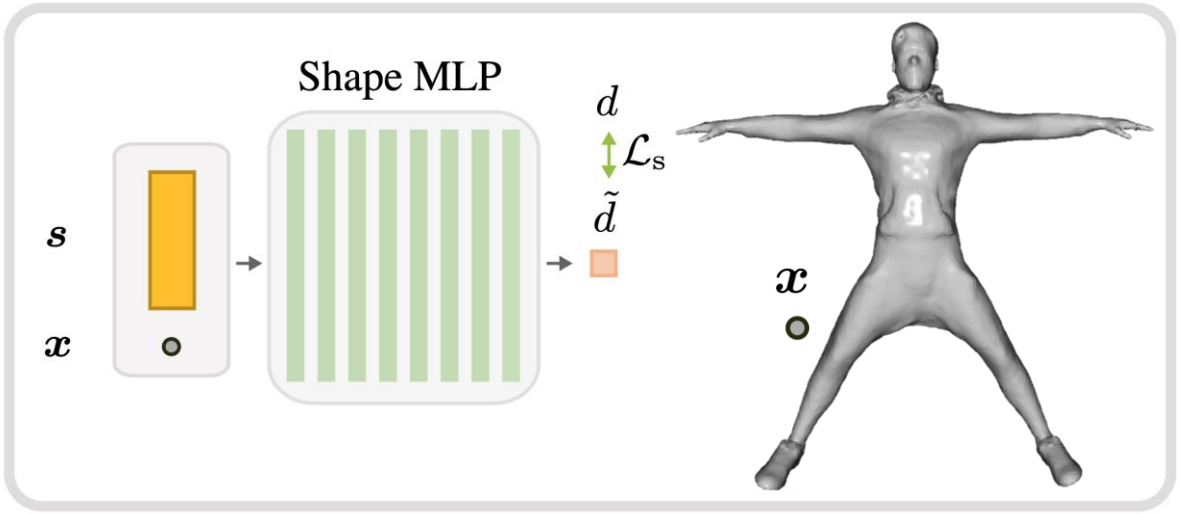
[Palafox et al. ‘21] NPMs

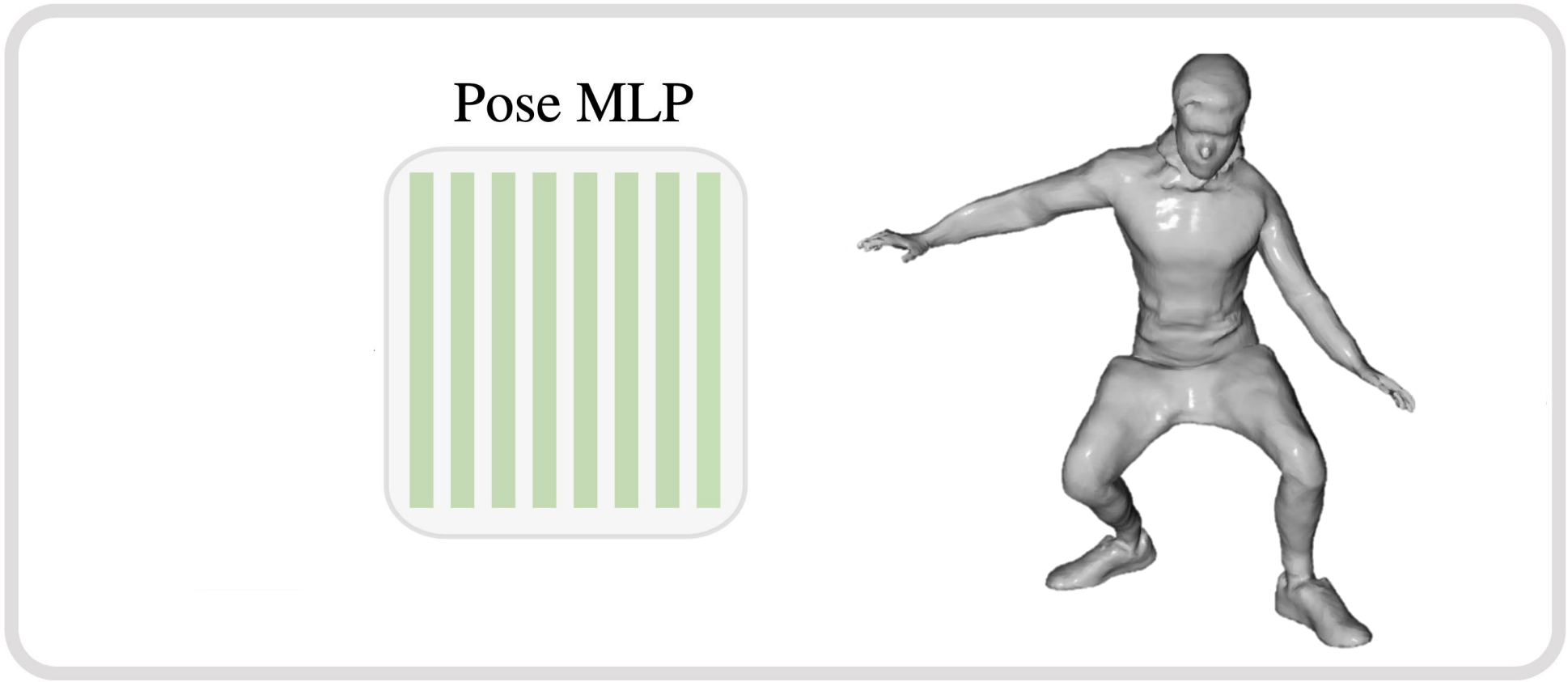
Shape MLP

[Palafox et al. ‘21] NPMs

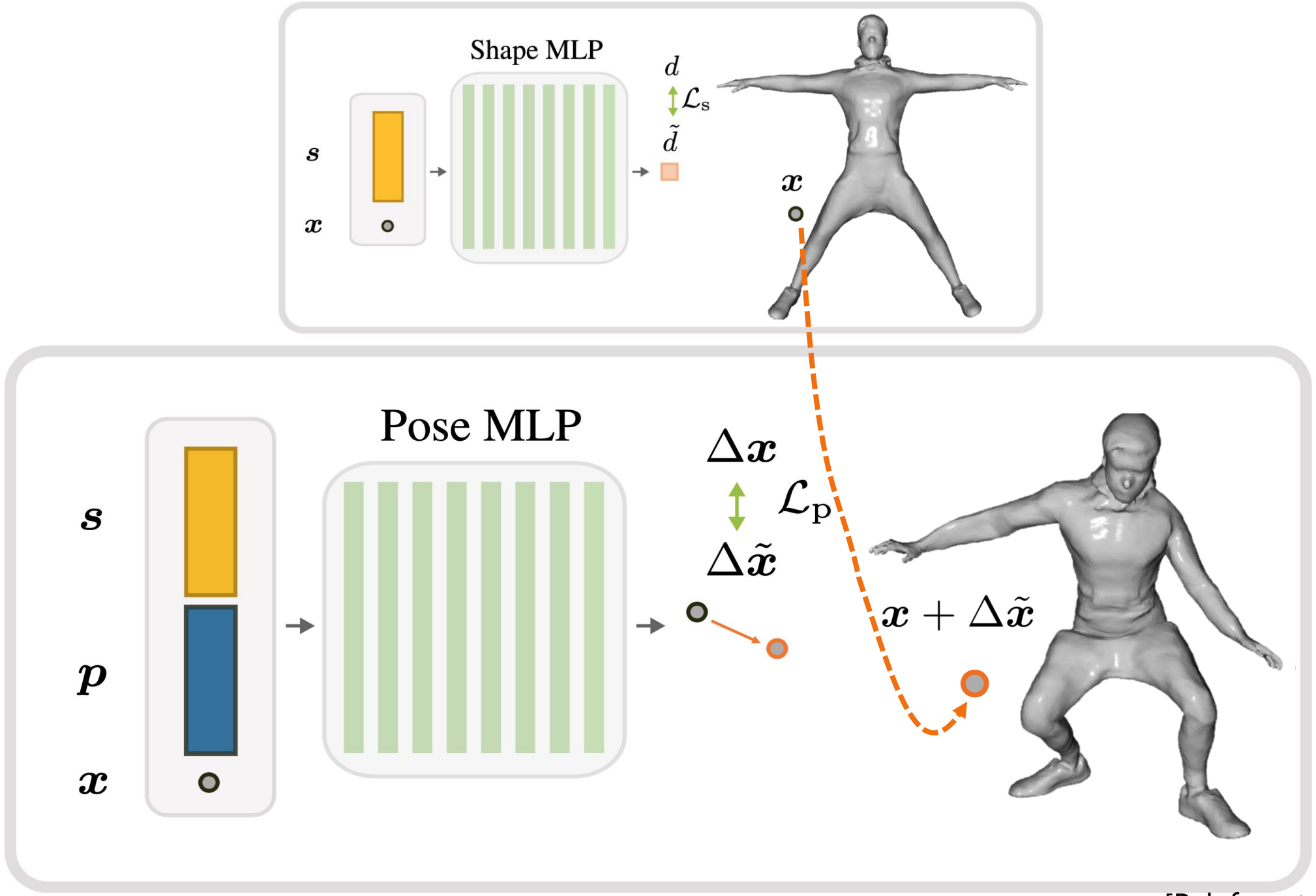


[Palafox et al. '21] NPMs

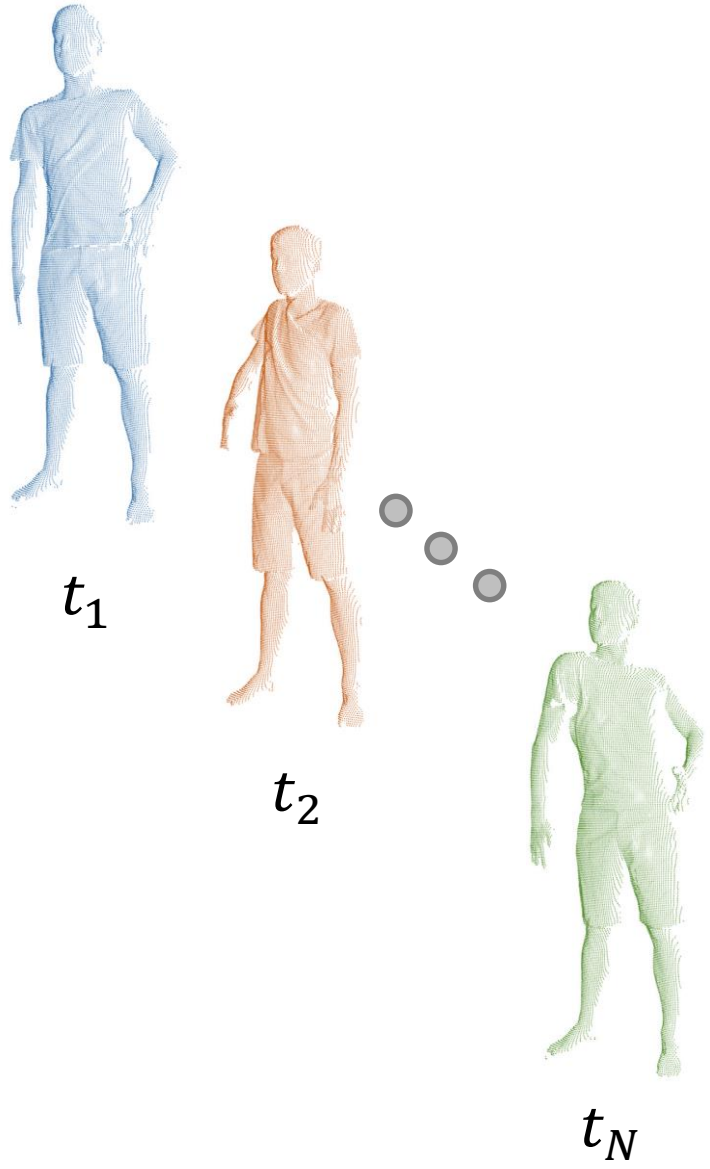




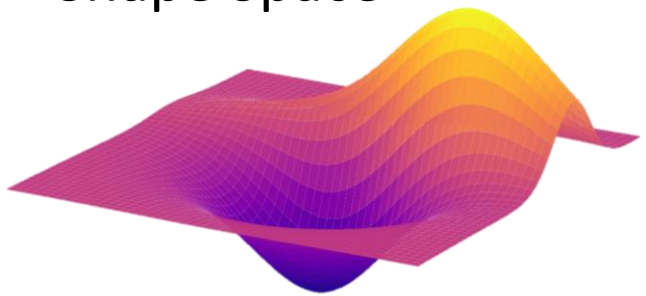
[Palafox et al. '21] NPMs



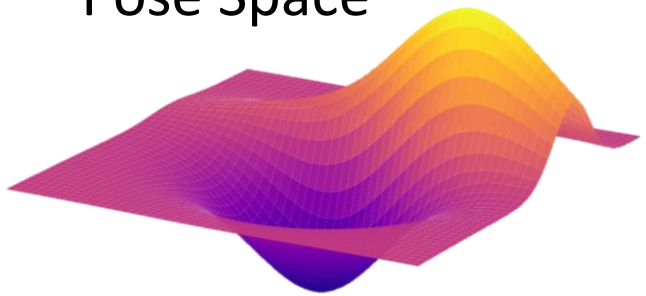
[Palafox et al. '21] NPMs

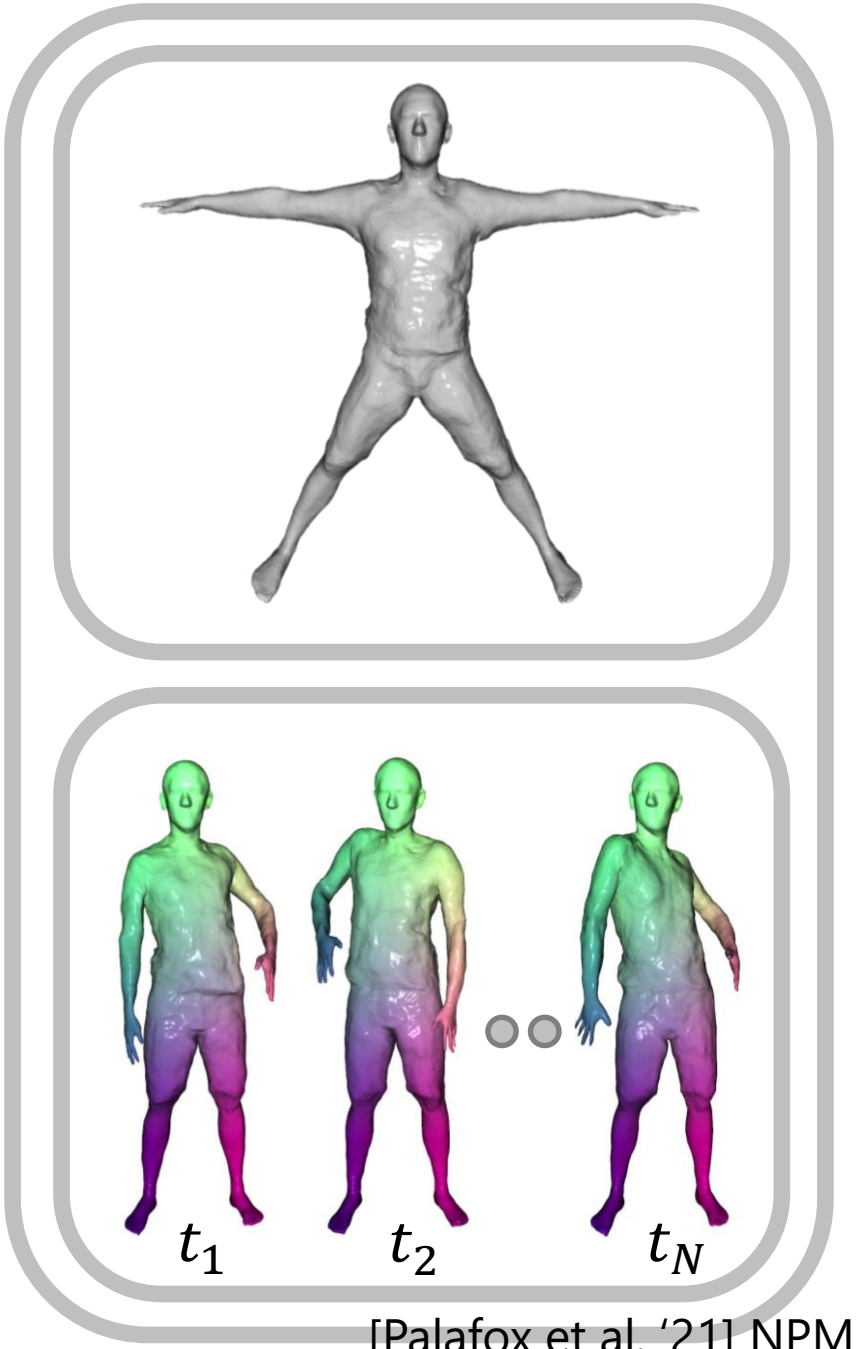


Shape Space

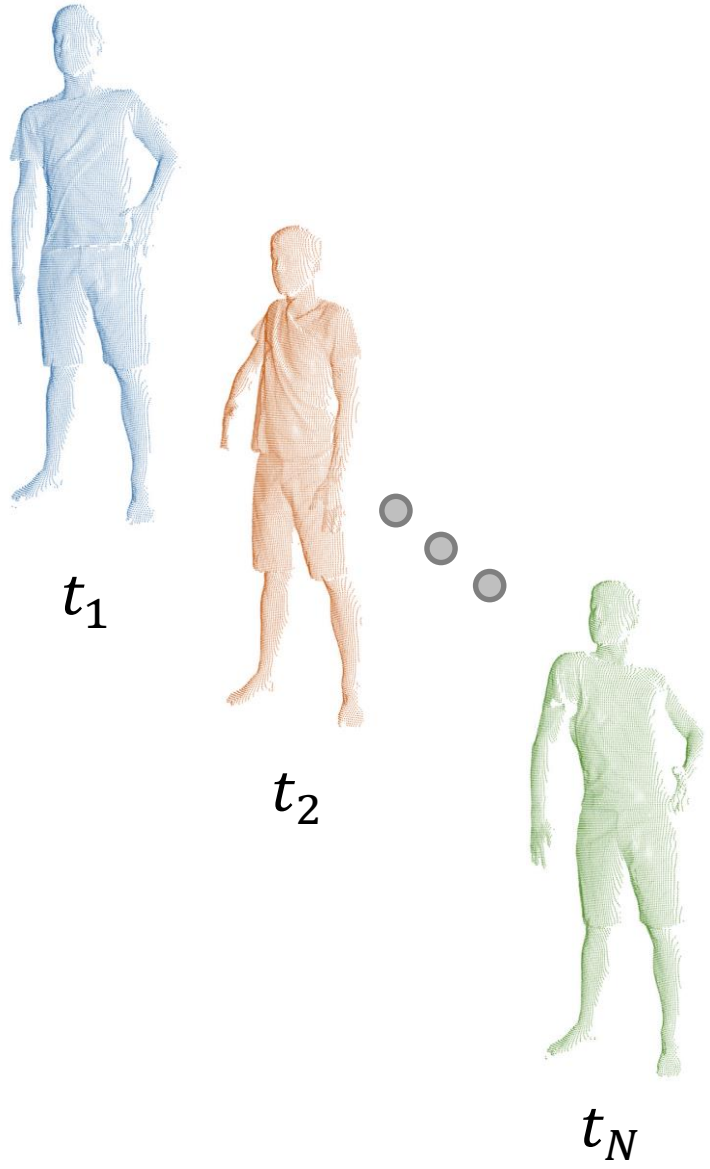


Pose Space

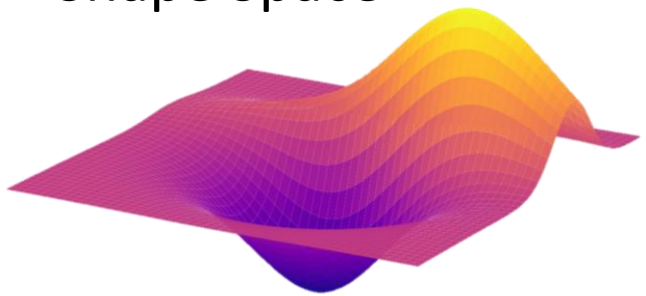




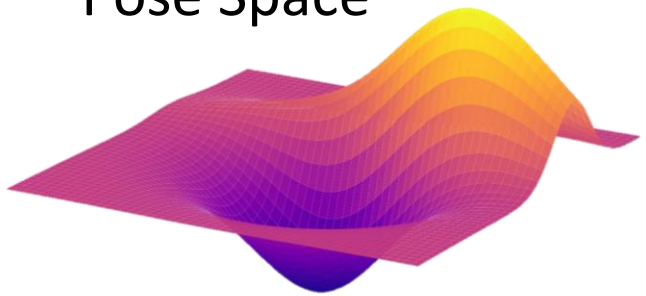
[Palafox et al. '21] NPMs



Shape Space



Pose Space



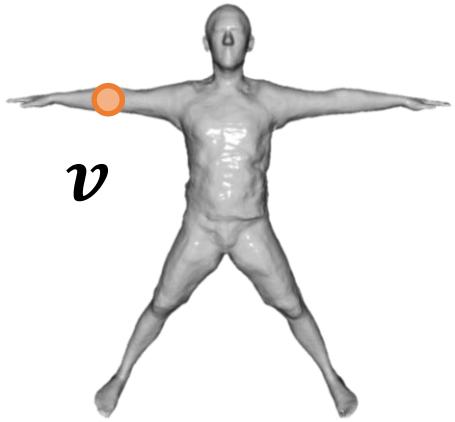
Optimized **Shape** Code

$\boldsymbol{s}$

Optimized **Pose** Codes

$\boldsymbol{p}_1$ $\boldsymbol{p}_2$ $\boldsymbol{p}_N$

[Palafox et al. '21] NPMs



$\boldsymbol{v}$

$\boldsymbol{s}$

$\boldsymbol{p}_1$

$\boldsymbol{p}_2$

Pose MLP

[Palafox et al. ‘21] NPMs

$\boldsymbol{s}$

$\boldsymbol{p}_2$

$\boldsymbol{v}$

$\boldsymbol{p}_N$

Pose MLP

[Palafox et al. ‘21] NPMs

$\boldsymbol{s}$

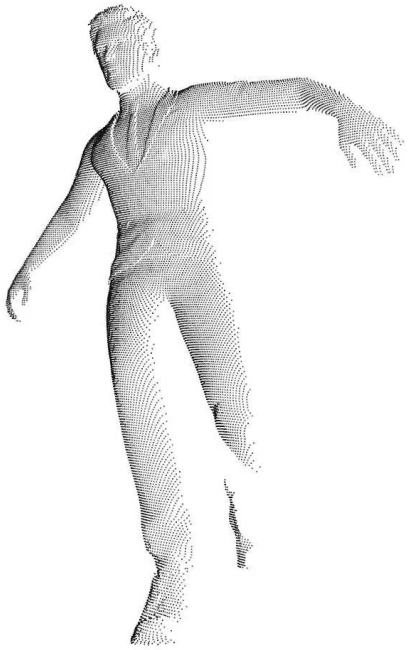
$\boldsymbol{p}_N$

$\boldsymbol{v}$

Pose MLP

[Palafox et al. '21] NPMs

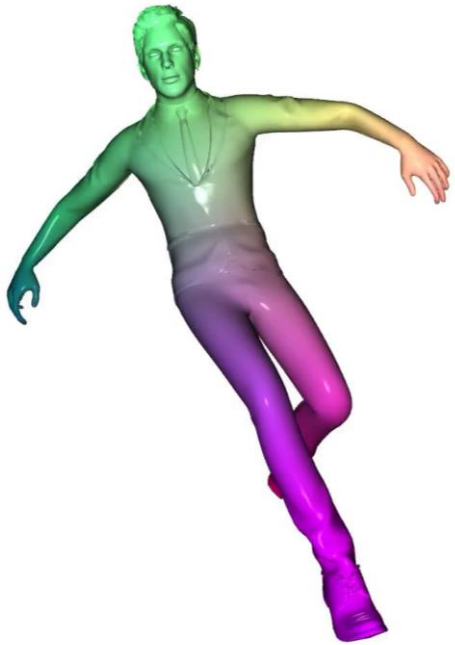
Neural Parametric Models



Depth Input

OpenPose + SMPL
[Cao et al. 2019]
[Loper et al. 2015]

NPMs



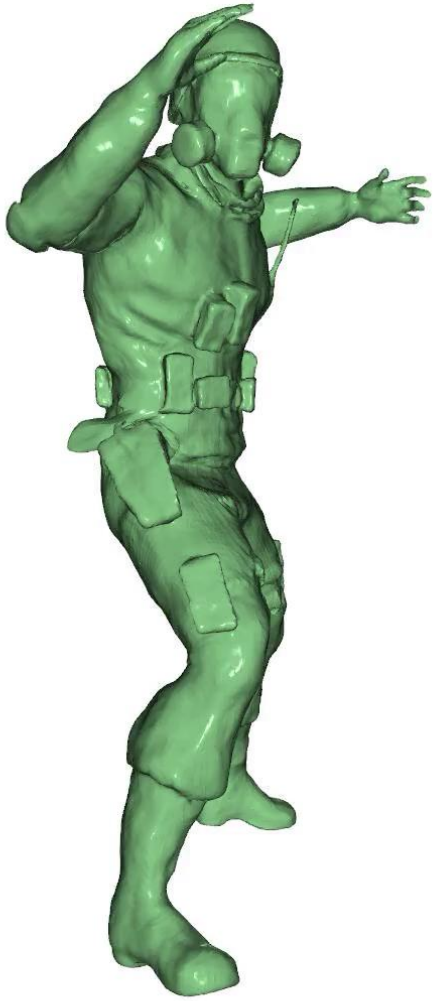
GT Scan

[Palafox et al. '21] NPMs

Neural Parametric Models: Interpolation

[Palafox et al. ‘21] NPMs

Neural Parametric Models: Interpolation



[Palafox et al. ‘21] NPMs

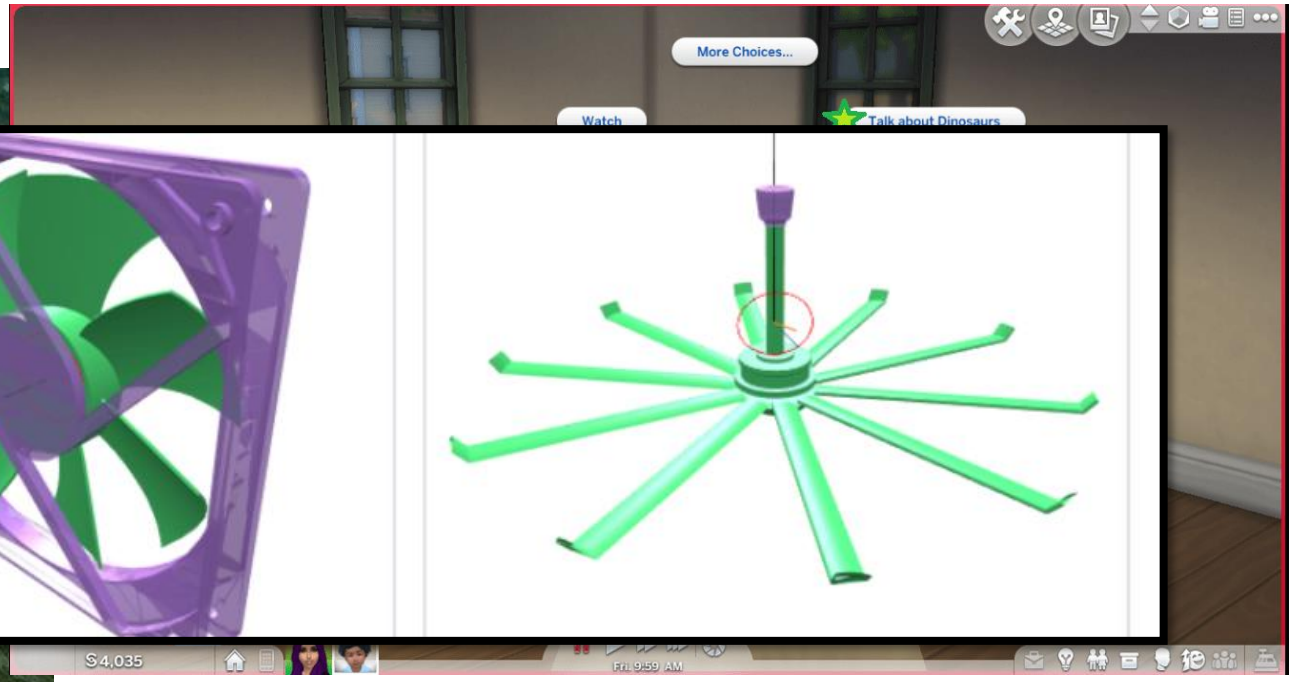
Understanding interactions

Divinity: Original Sin

Sims 4

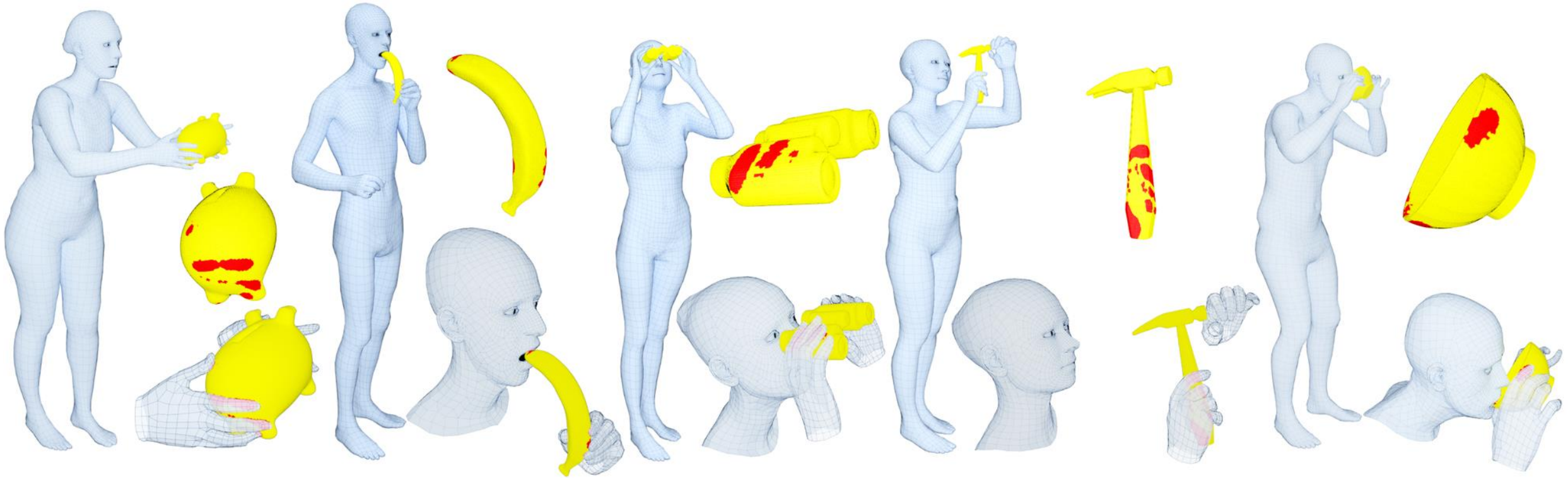
Understanding interactions

Divinity: Original Sin



Sims 4

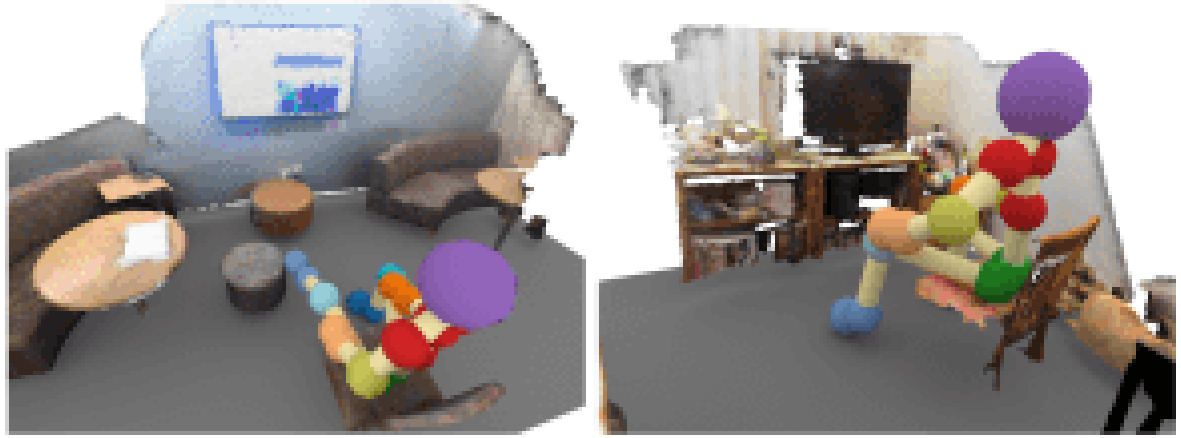
Understanding interactions



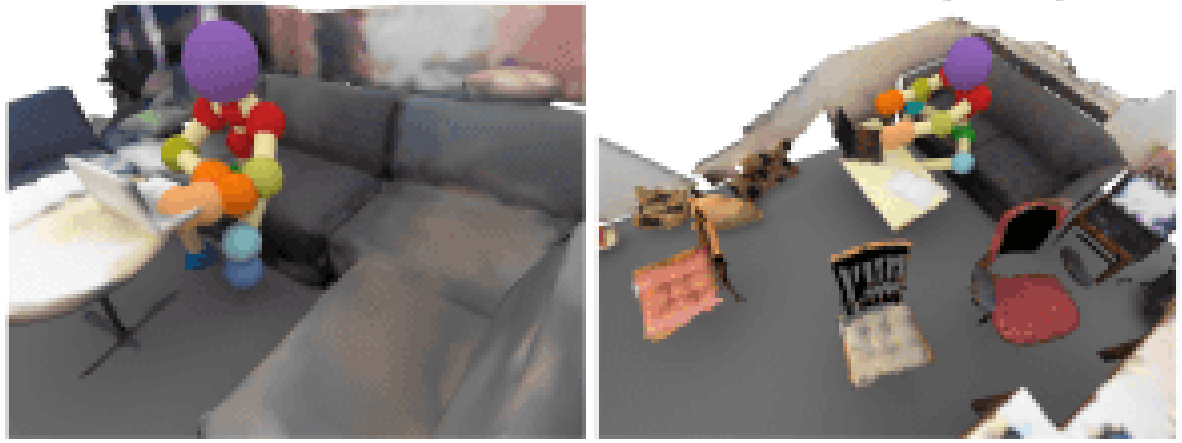
[Taheri et al. ’20]

Predicting interactions

"Sit on a chair and watch TV"



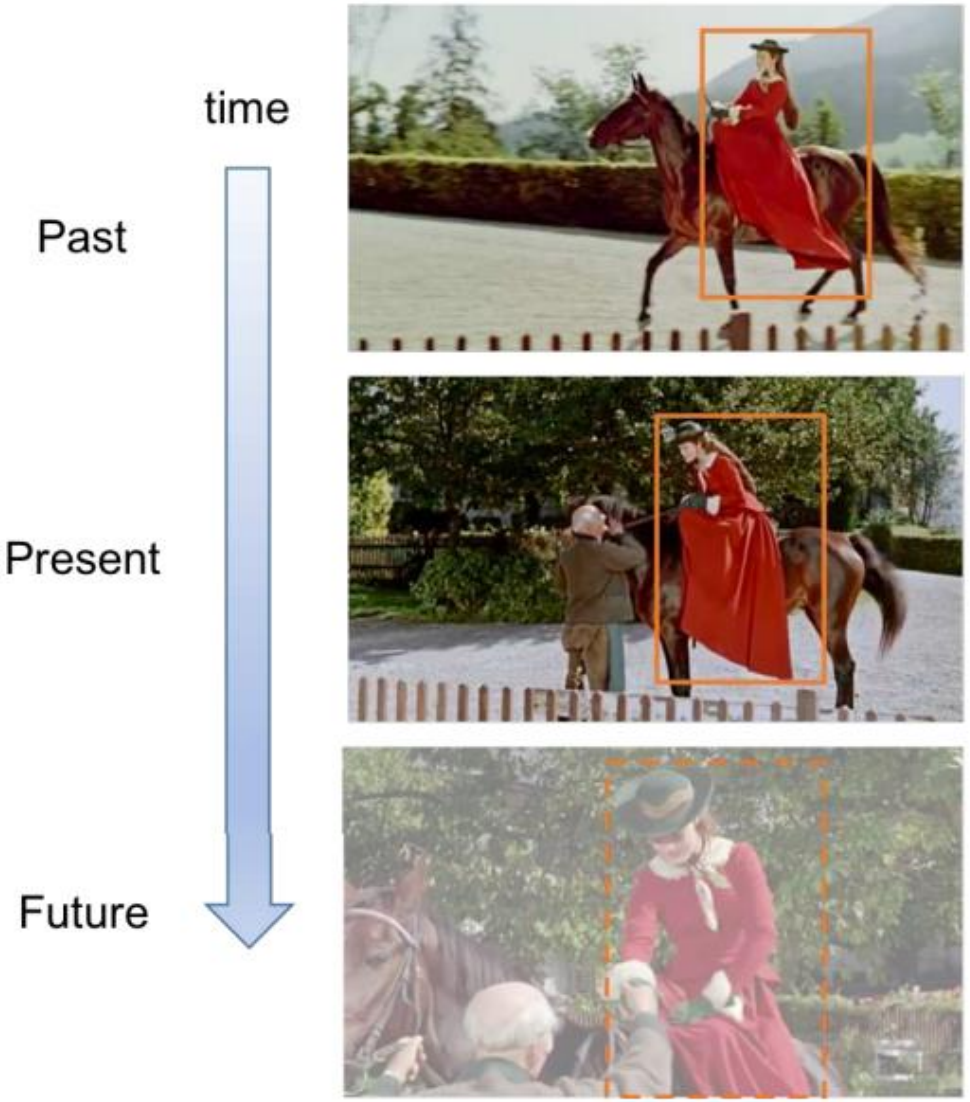
"Sit on a couch and use a laptop"



"Write on a whiteboard"

[Savva et al. '16]

Predicting interactions

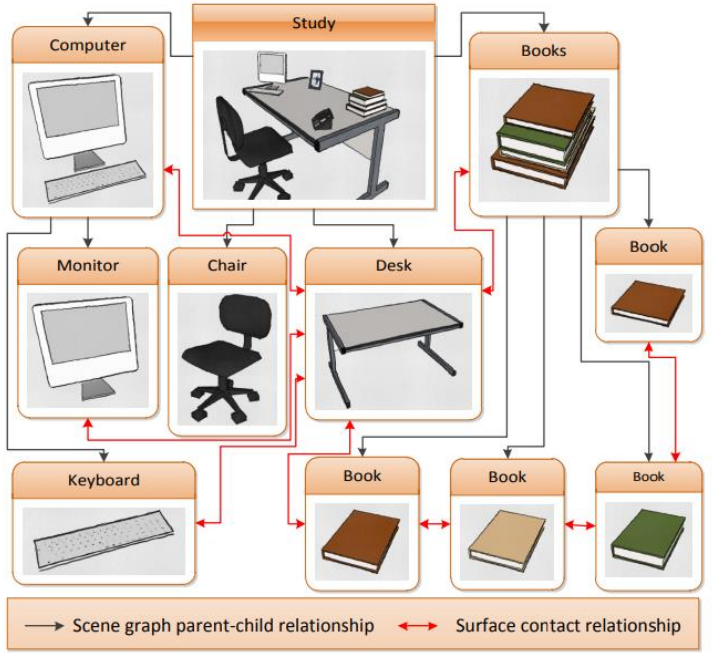


[Sun et al. ’19]

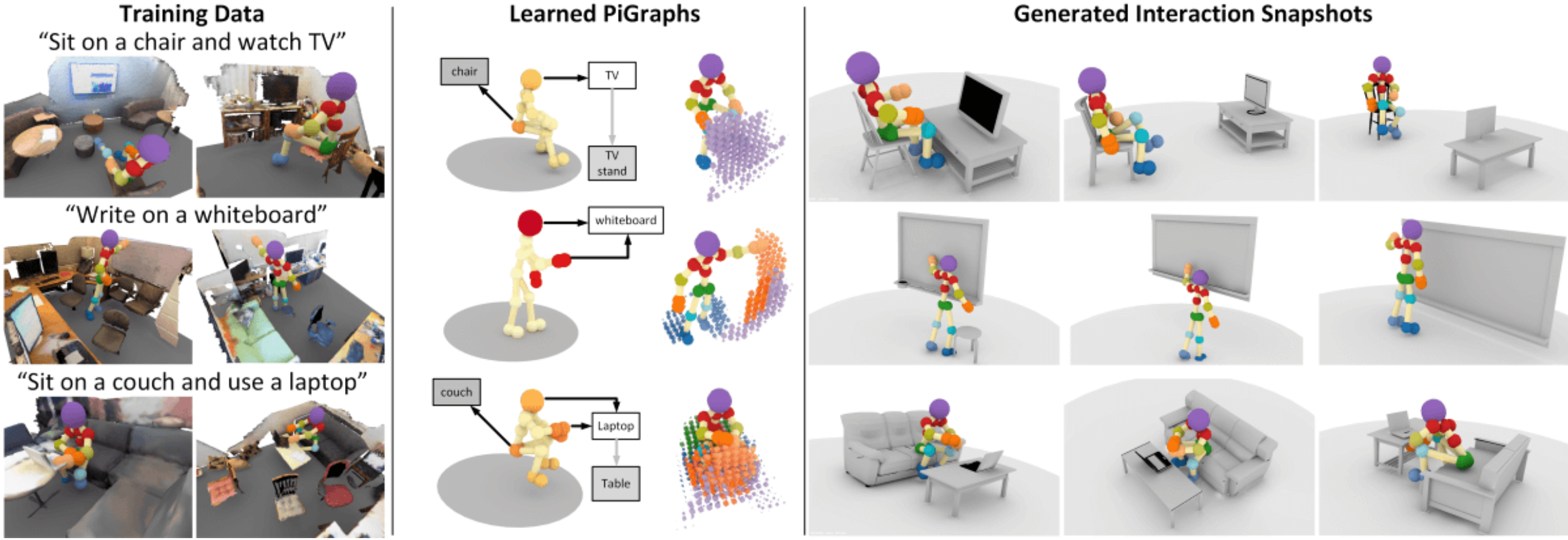
Challenges

- Data – acquisition and annotation for supervision? Diversity?

- What representation for objects/scene/person?
  - Efficiently representing dynamics or deformations?



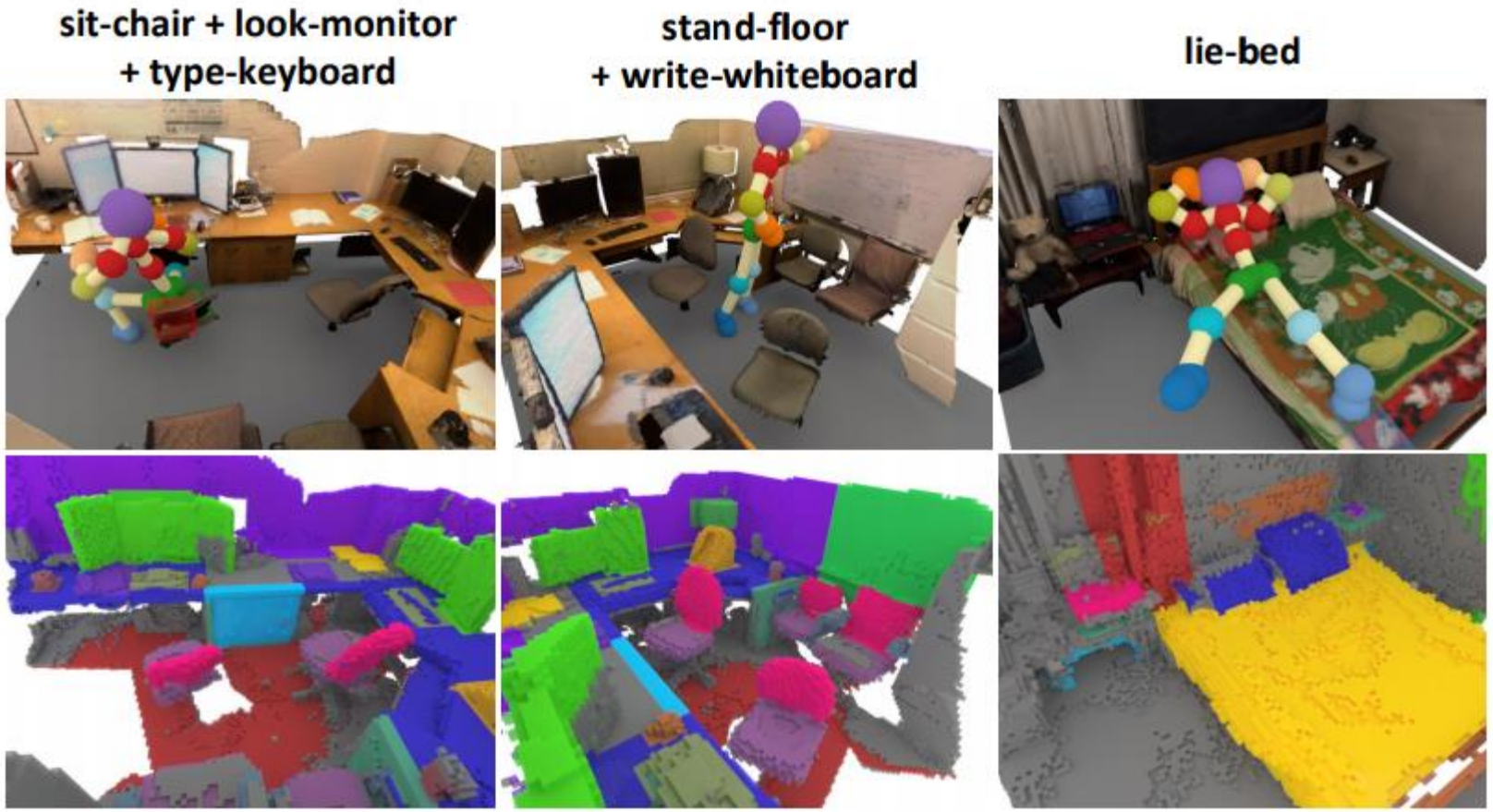
PiGraphs: Learning Interaction Snapshots



[Savva et al. '16]

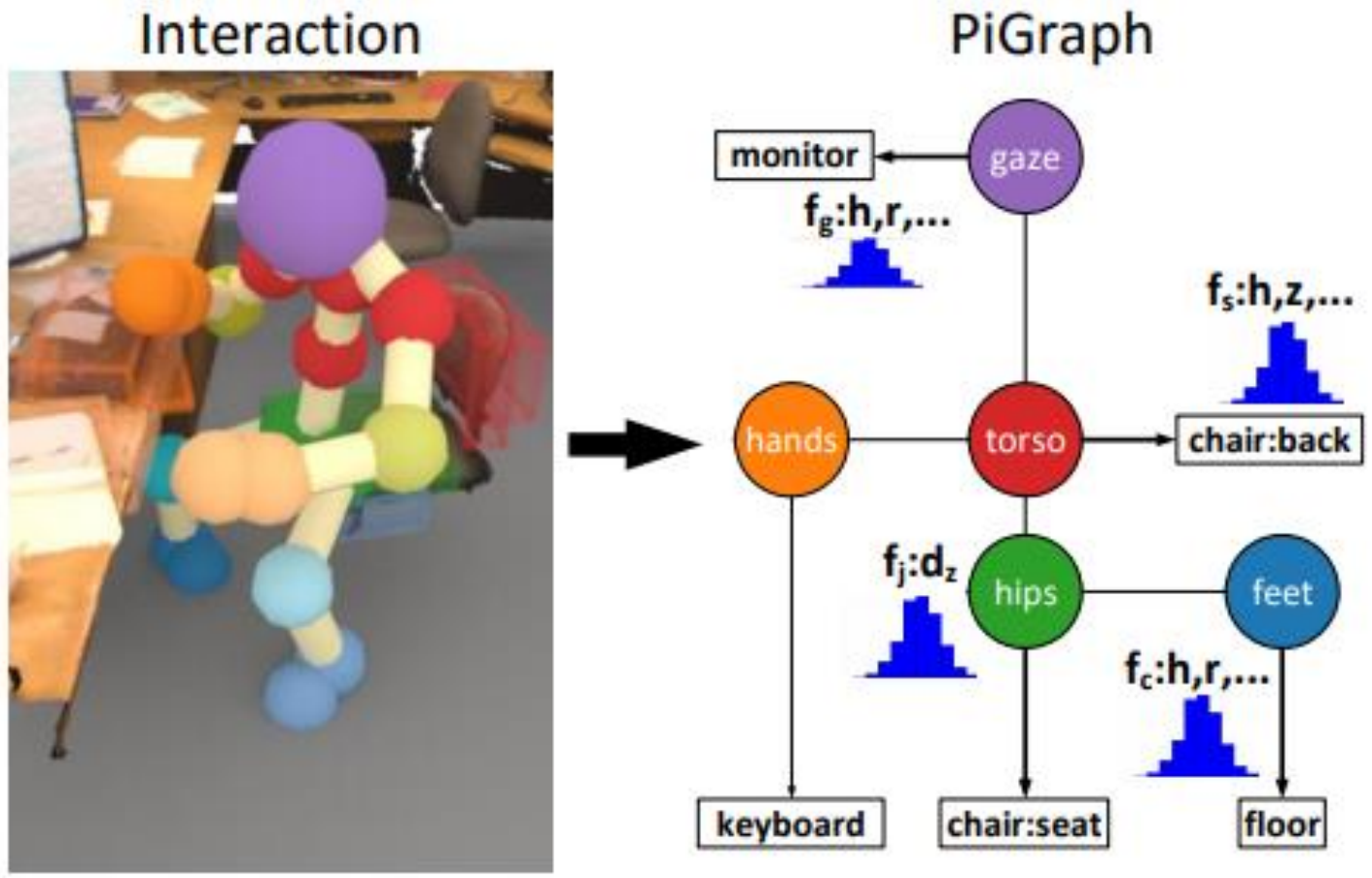
PiGraphs: Learning Interaction Snapshots

- Data collection and annotation



[Savva et al. ‘16]

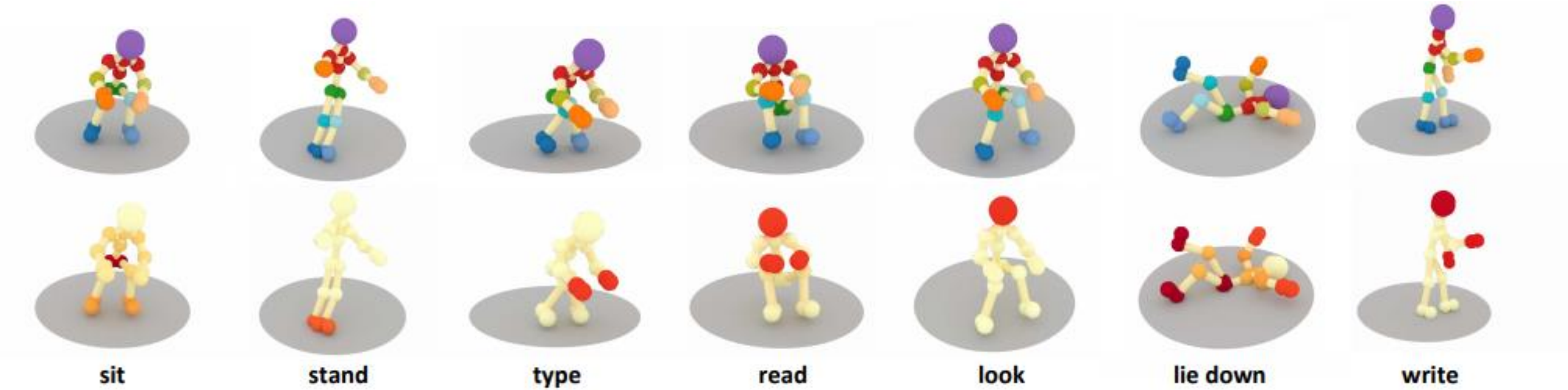
PiGraphs: Learning Interaction Snapshots



[Savva et al. '16]

PiGraphs: Learning Interaction Snapshots

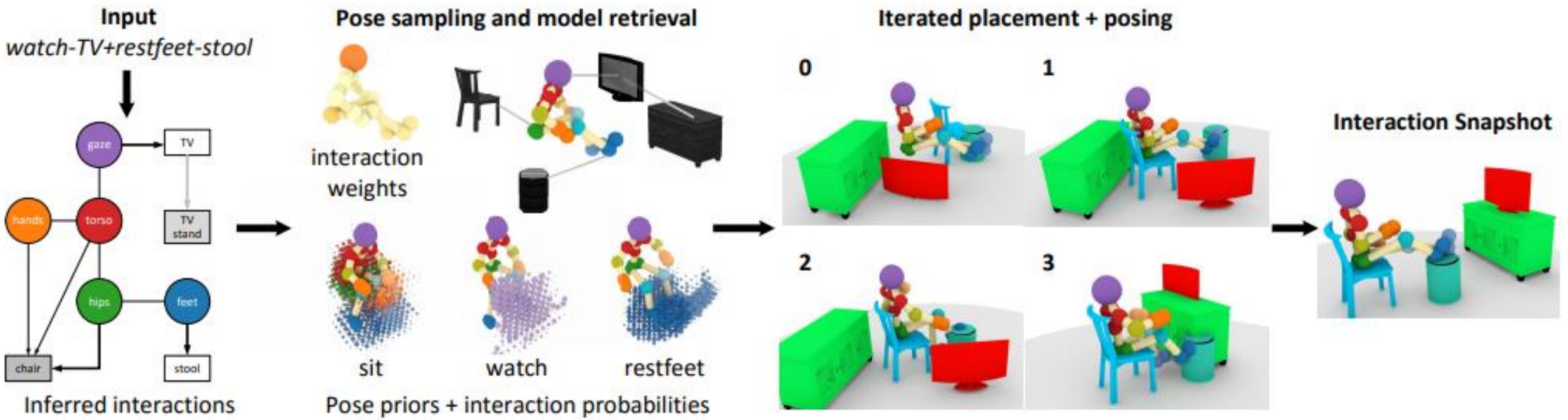
Maximum likelihood poses for aggregated skeleton distributions of some action verbs



Conditional probabilities of body part interaction with objects during each action

[Savva et al. ‘16]

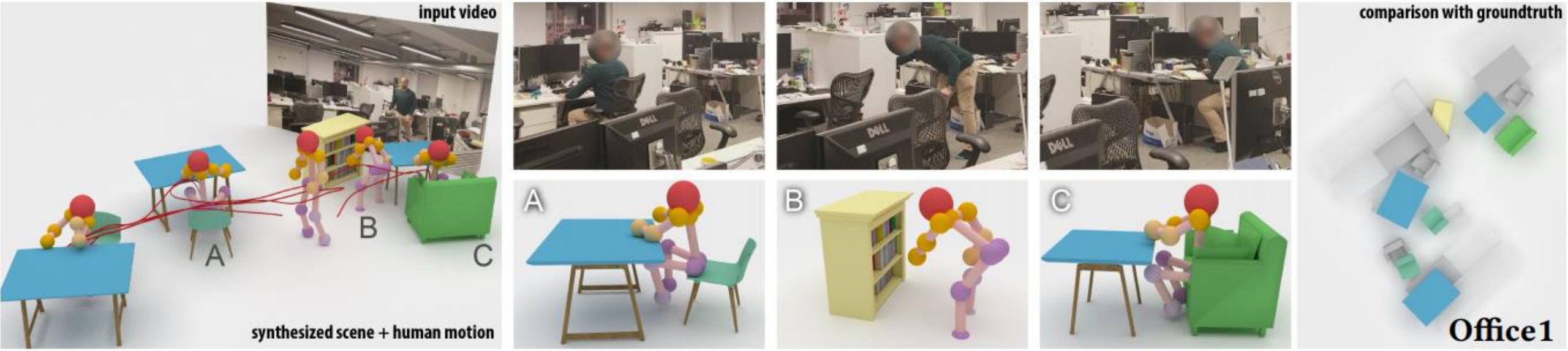
PiGraphs: Learning Interaction Snapshots



[Savva et al. '16]

iMapper: Interaction-guided Scene Mapping from Monocular Videos

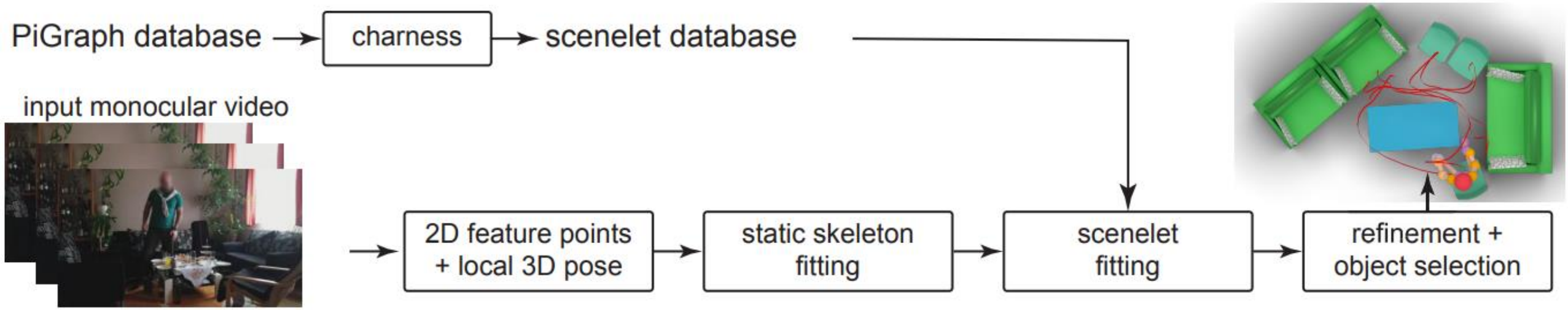
- Input: monocular video
- Output: scene object arrangement and human motion



[Monspart et al. ‘18]

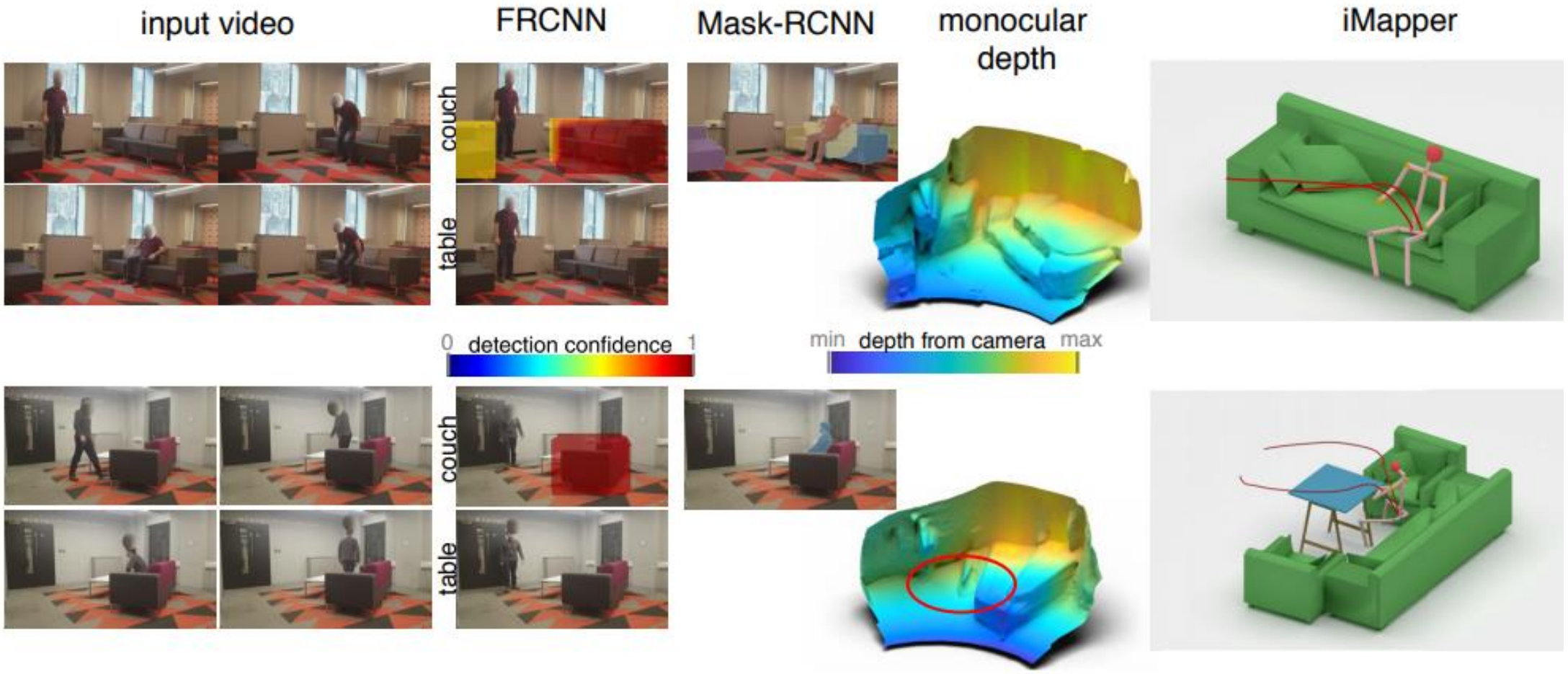
iMapper: Interaction-guided Scene Mapping from Monocular Videos

- Database of short interactions (scenelets) capturing relationship between human motion and objects (from PiGraph data)
- Synthesize output by fitting scenelets to video



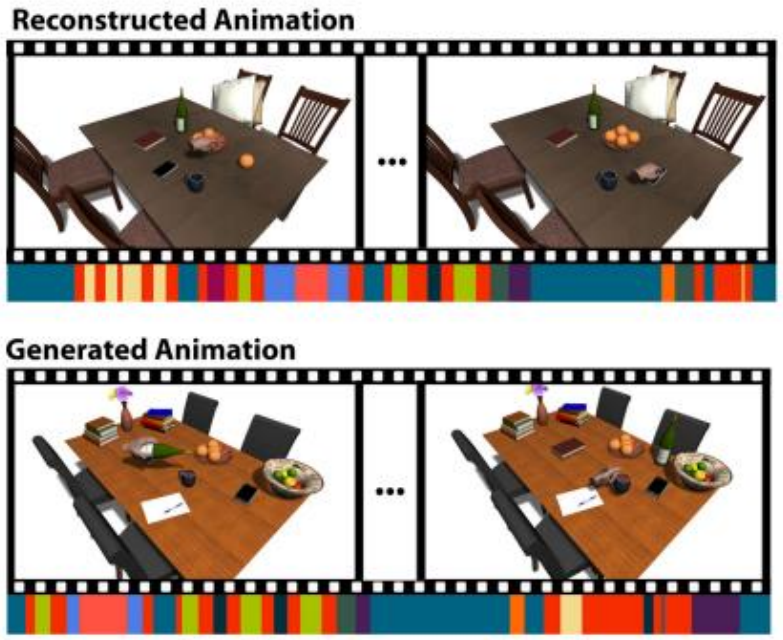
[Monspart et al. '18]

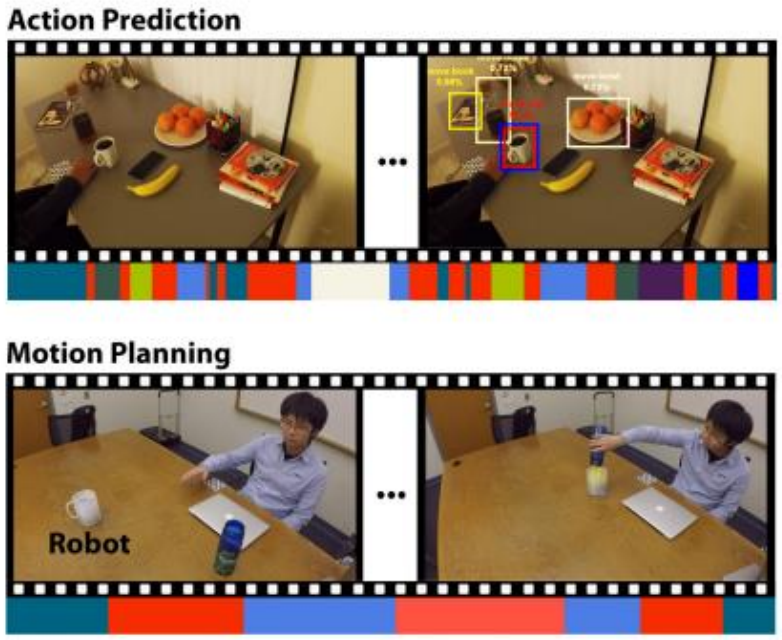
iMapper: Interaction-guided Scene Mapping from Monocular Videos



[Monspart et al. '18]

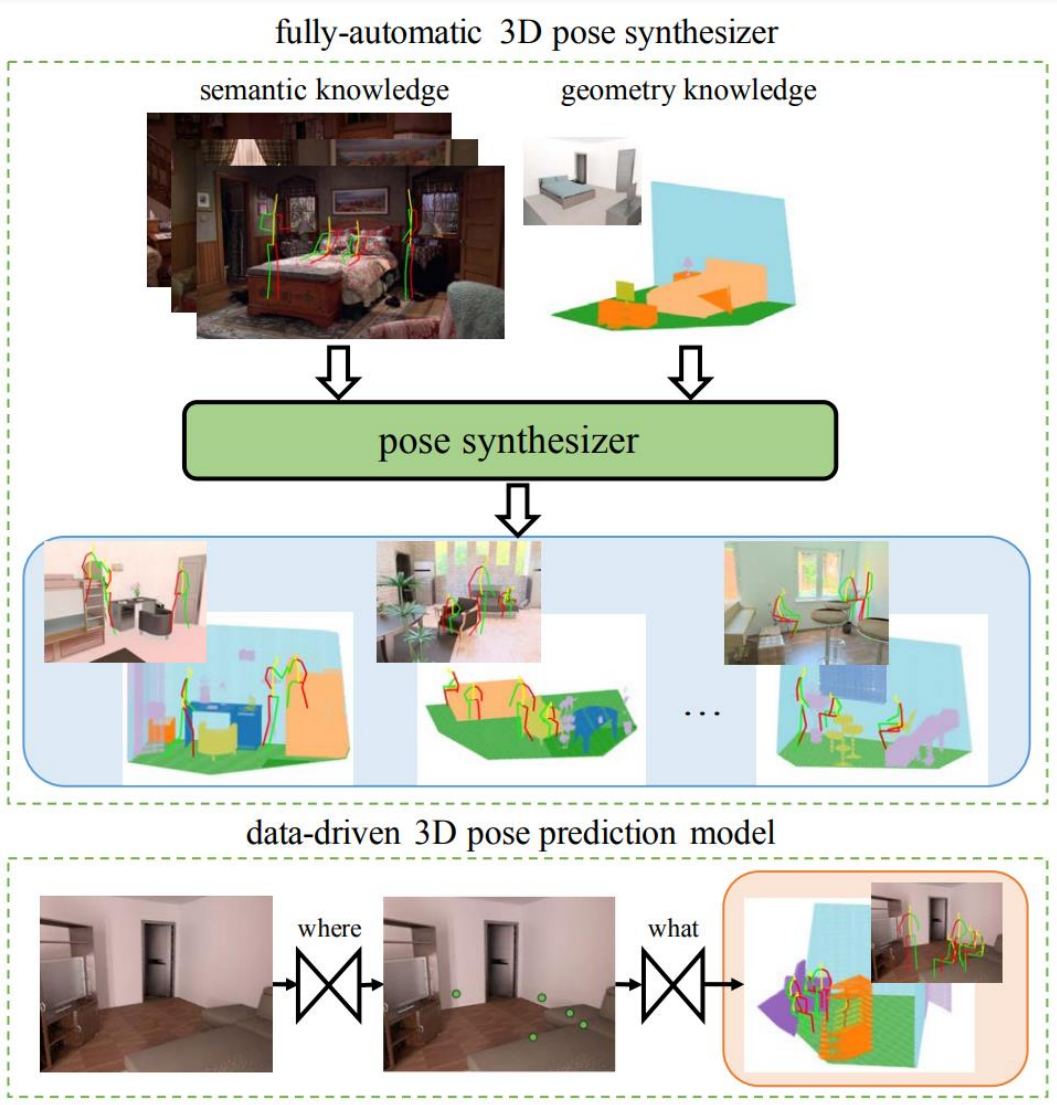
Learning a Generative Model for Multi-Step Human-Object Interactions from Videos





[Wang et al. '19]

Putting Humans in a Scene: Learning Affordance in 3D Indoor Environments

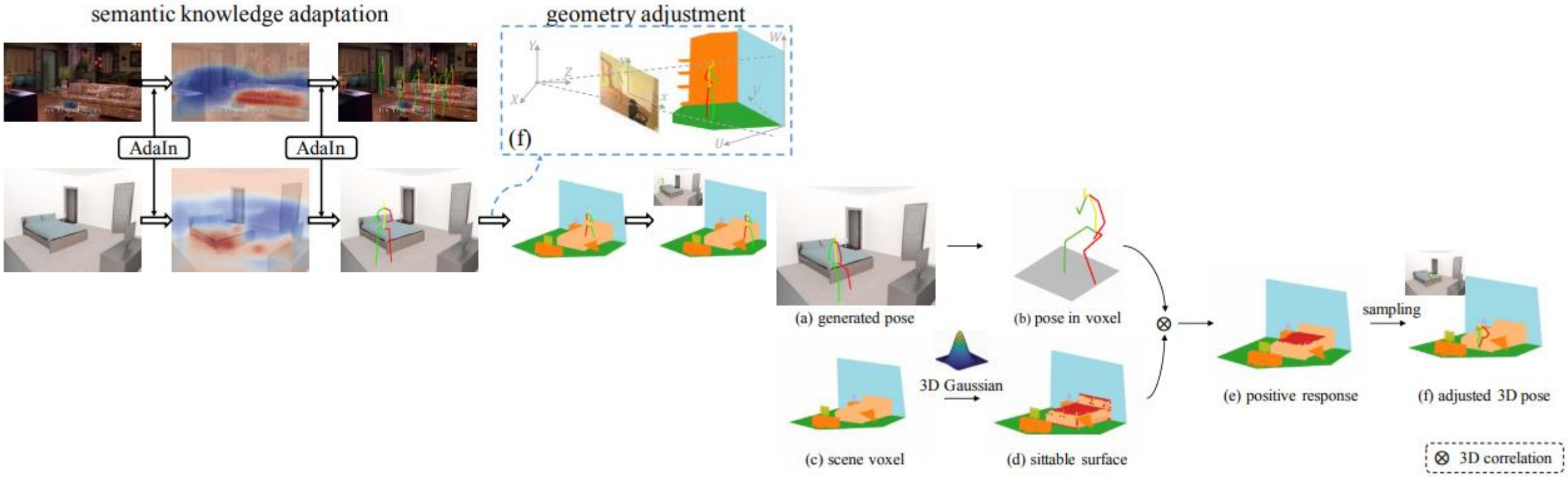


Predict what human poses are afforded by a 3D indoor scene based on learning from 2D poses

[Li et al. '19]

Putting Humans in a Scene: Learning Affordance in 3D Indoor Environments

- Synthesize 3D poses: first generate poses in 2D images, then project to 3D



[Li et al. '19]

Generating 3D People in Scenes

[Zhang et al. ‘20]

Generating 3D People in Scenes

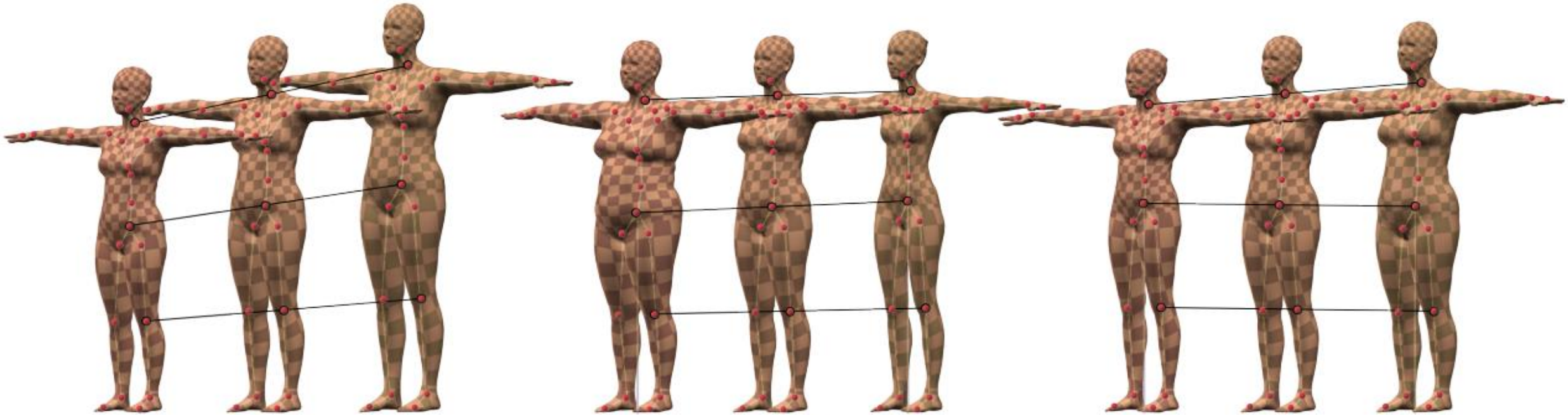
- Input: scene depth and semantics
- Conditional variational autoencoder for body; use 3D body representation (SMPL-X) rather than skeleton joints only
- Fit 3D human body to the scene geometry to avoid floating/collisions

- Dataset (PROX-E) of 3D body meshes placed in RGB-D data
- Virtual rendering (≈ 70k frames) of scenes

[Zhang et al. '20]

Human body models

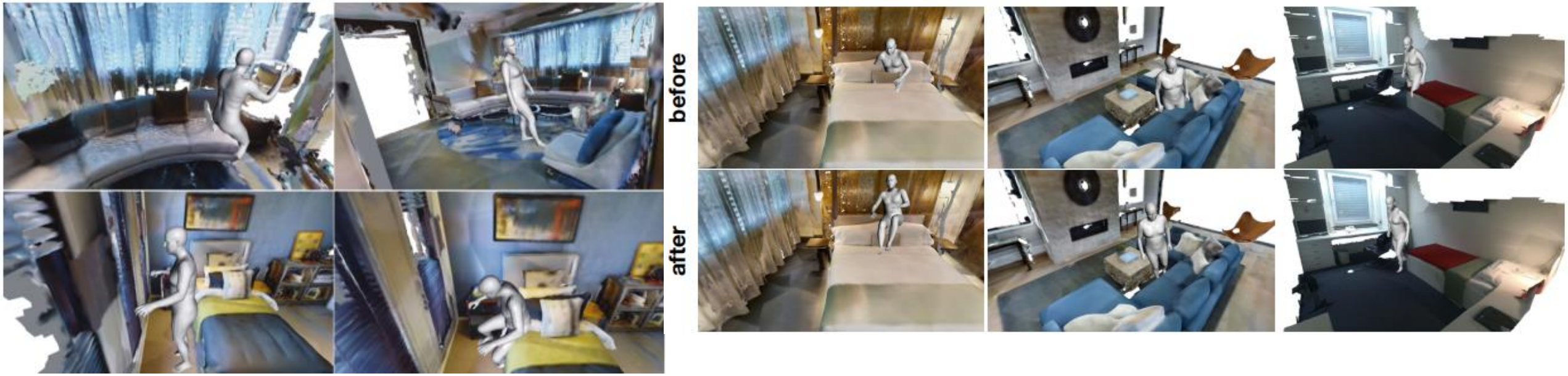
- SMPL, SMPL-X
- Parametric 3D models for human body shape



(a) *PC 1* (b) *PC 2* (c) *PC 3*

[Loper et al. ‘15; Pavlakos et al. ‘19]

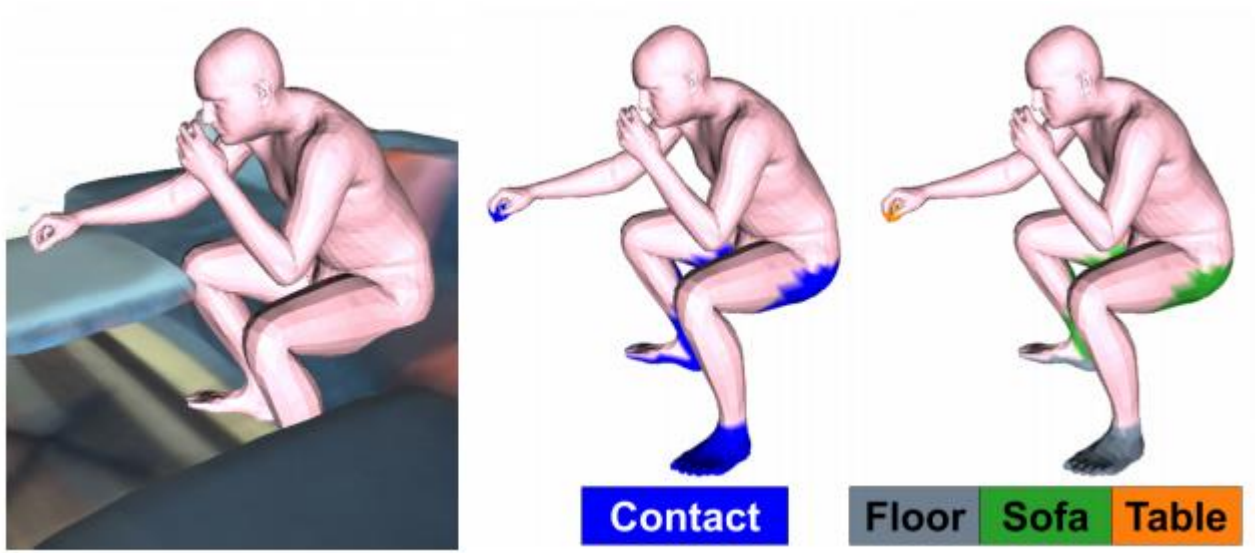
Generating 3D People in Scenes

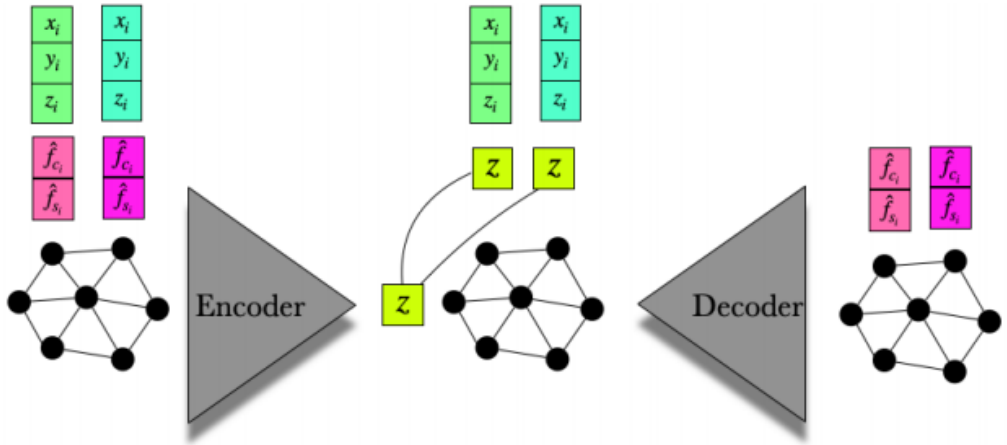


[Zhang et al. ‘20]

Populating 3D Scenes

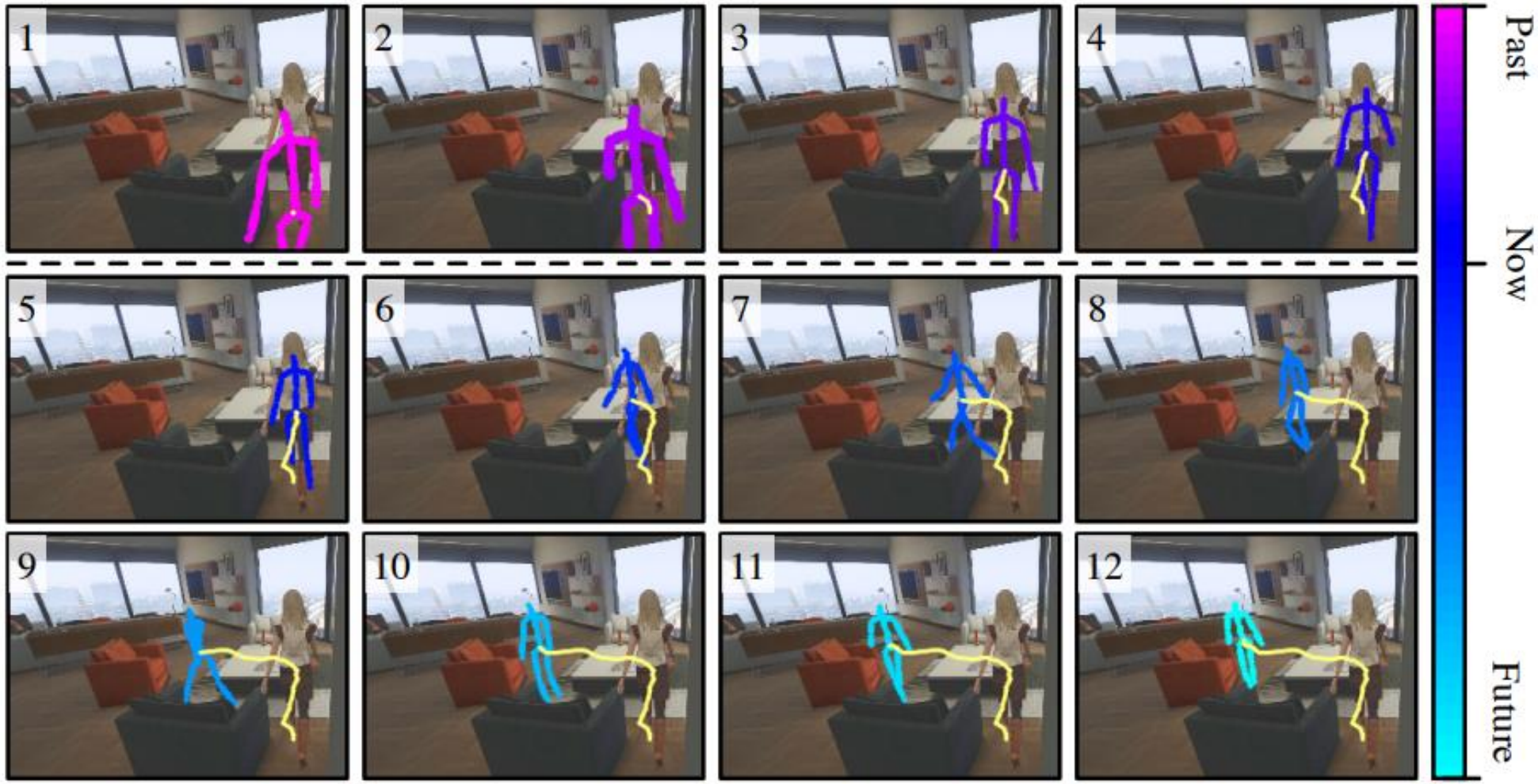
- Model of human-scene interaction with SMPL-X
- Encoder on vertices of SMPL-X the contact/semantic label for object contacts in an interaction





[Hassan et al. '21]

Human motion prediction with scene context



[Cao et al. ‘20]

Human motion prediction with scene context

- Input: image and 2D pose history
- Sample multiple possible 2D destinations
- Predict 3D path to each destination

(a) predicted goals (b) planned paths (c) final poses

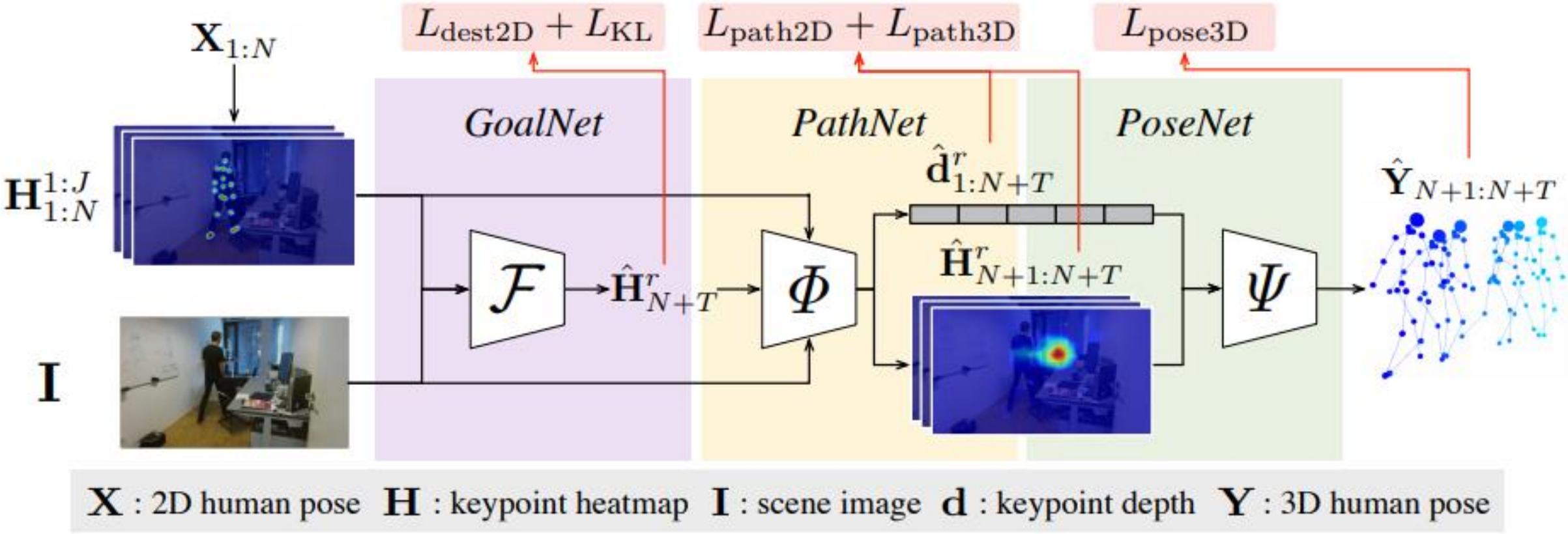
[Cao et al. ‘20]

Human motion prediction with scene context

- Synthetic dataset from GTA of person moving in scene

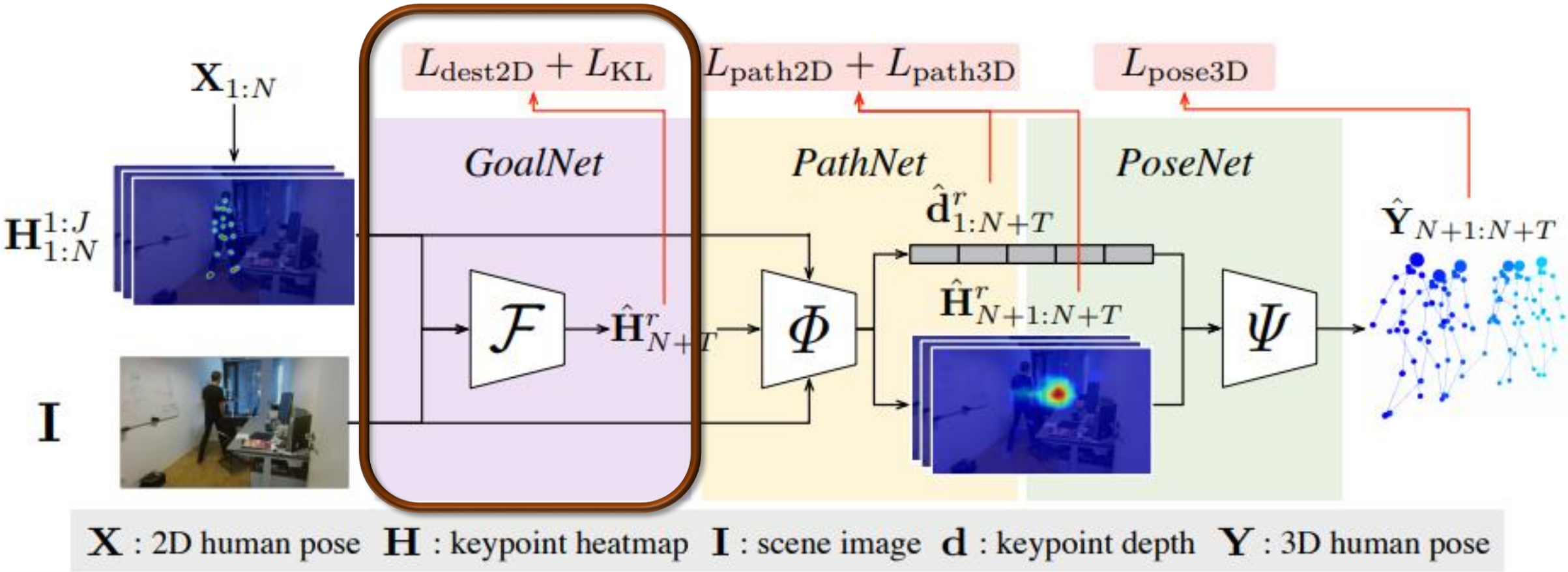
[Cao et al. ‘20]

Human motion prediction with scene context



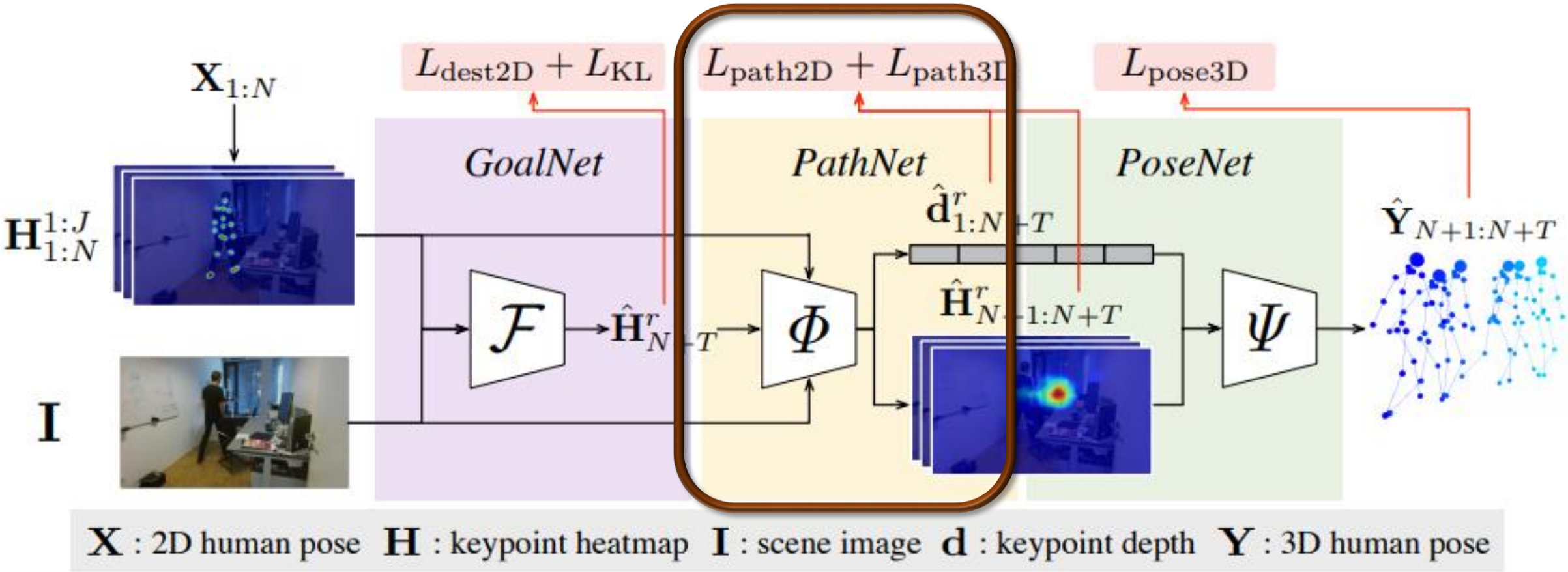
[Cao et al. '20]

Human motion prediction with scene context



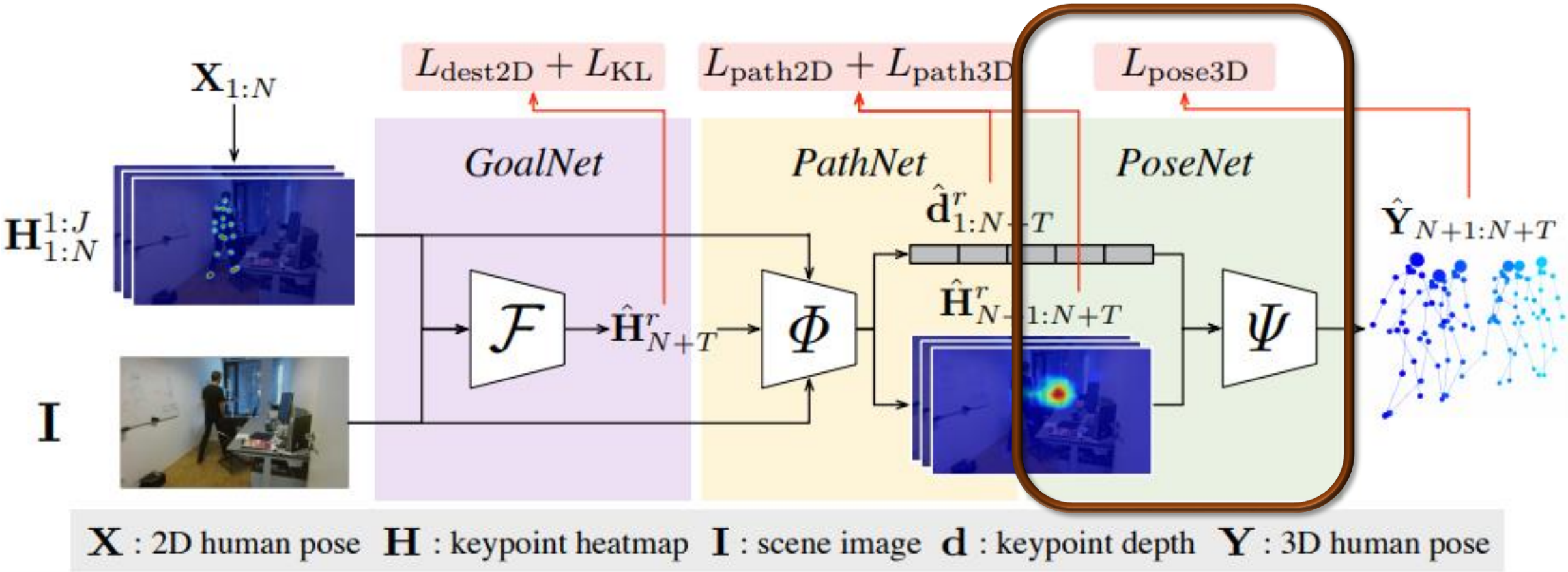
[Cao et al. '20]

Human motion prediction with scene context



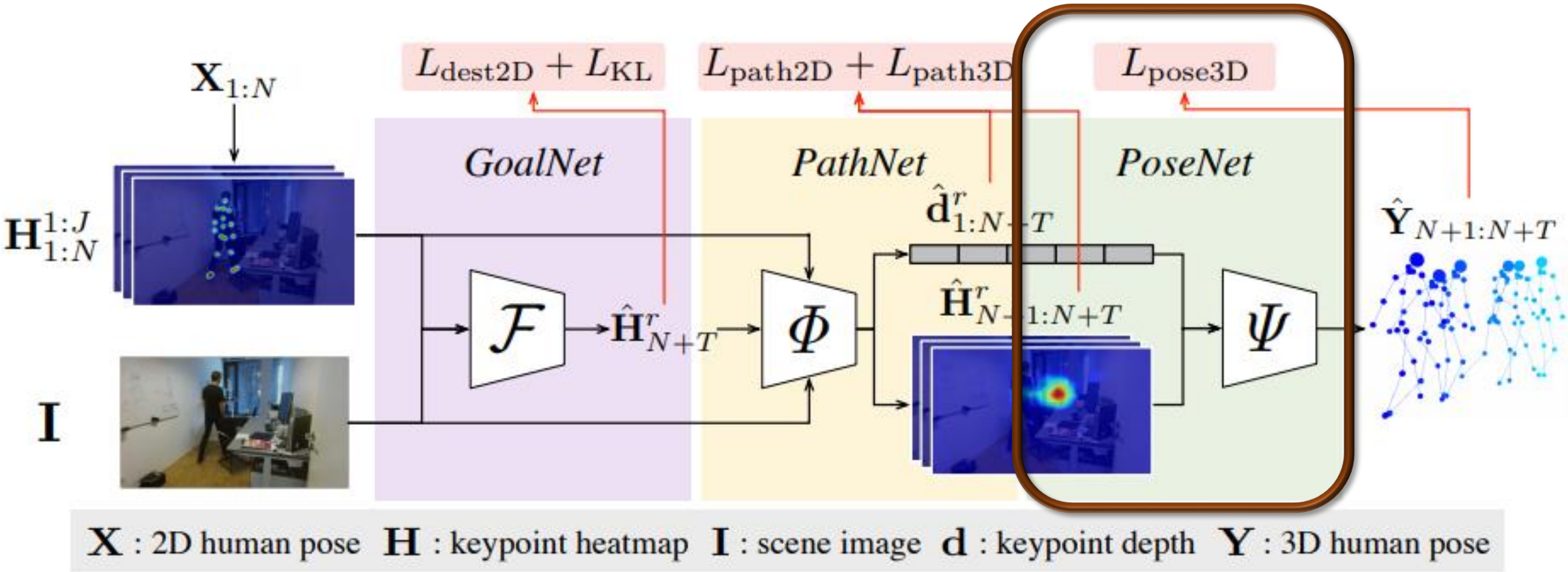
[Cao et al. '20]

Human motion prediction with scene context



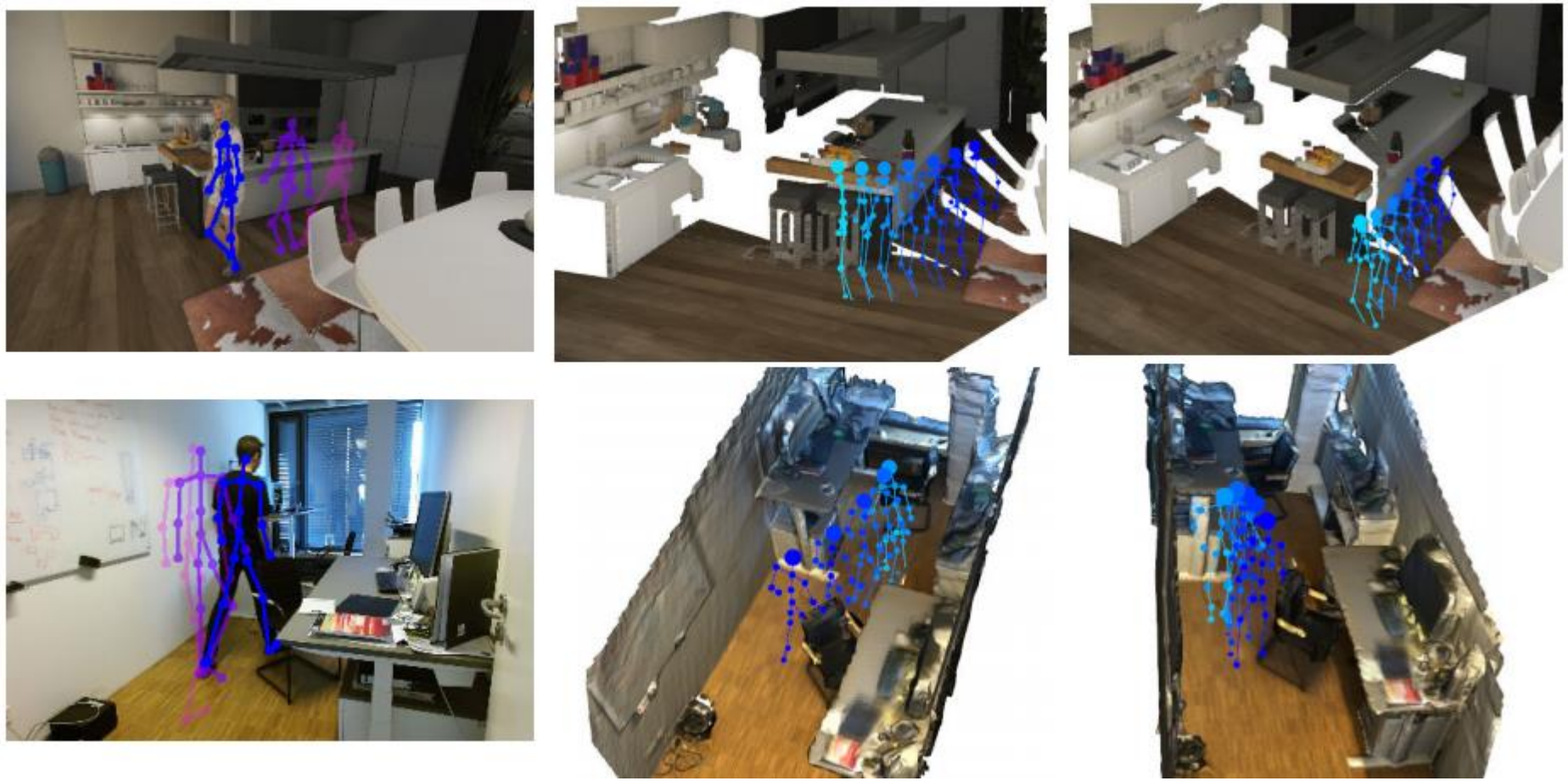
[Cao et al. '20]

Human motion prediction with scene context



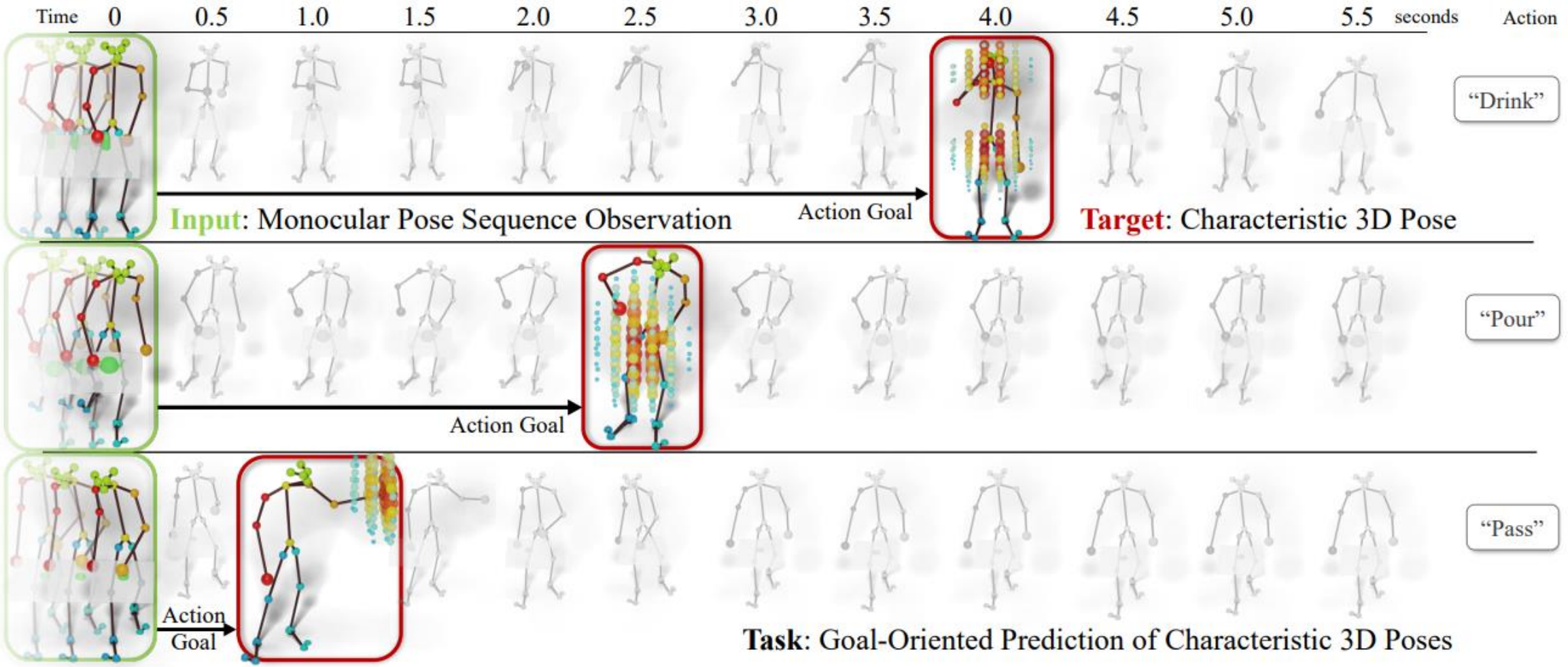
[Cao et al. '20]

Human motion prediction with scene context



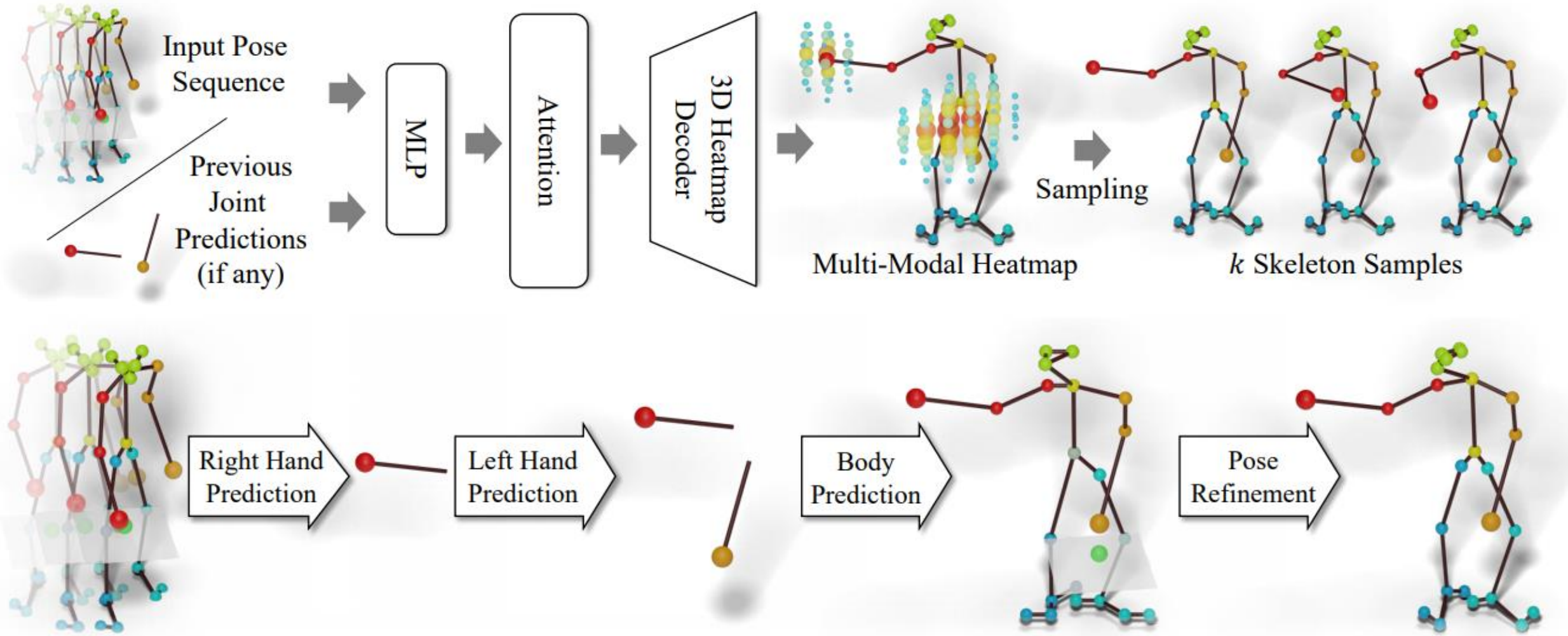
[Cao et al. ‘20]

Forecasting Characteristic Poses



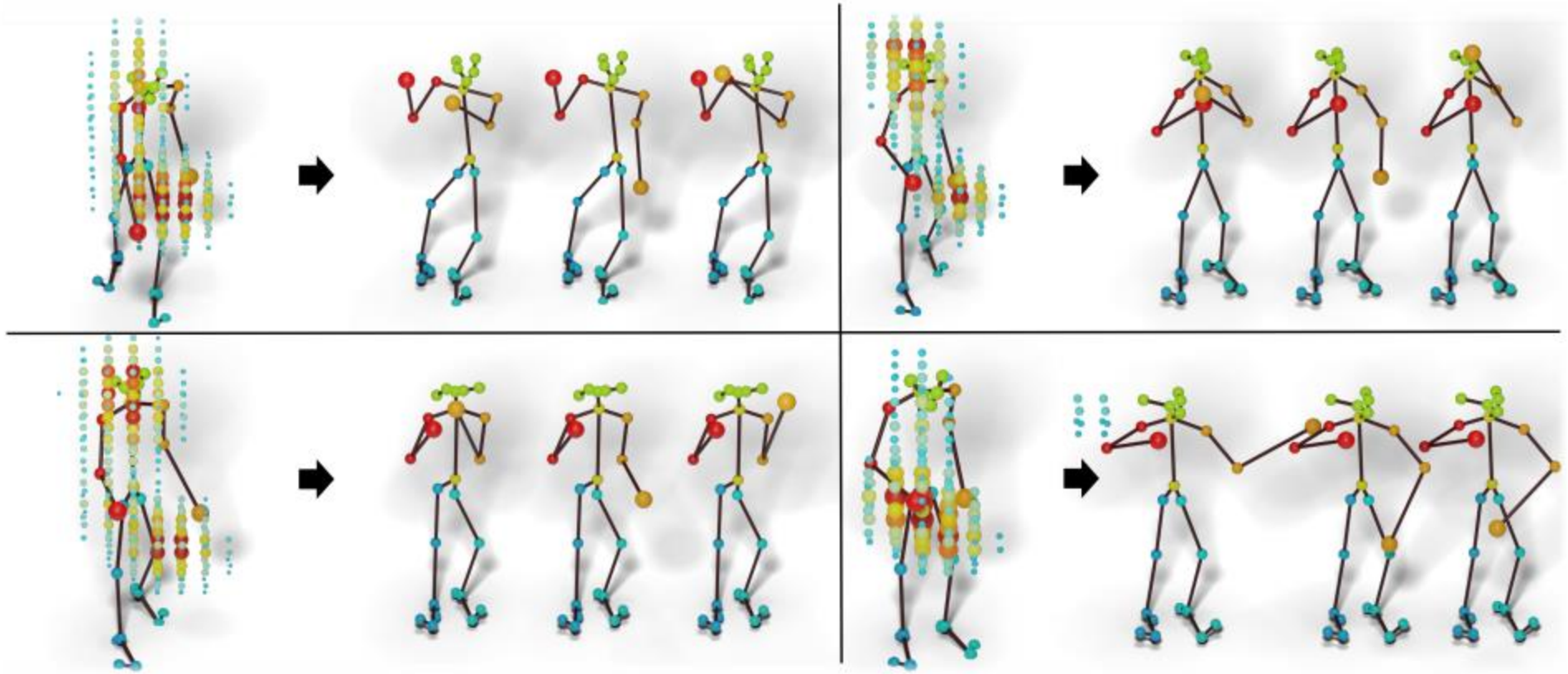
[Diller et al. '21]

Forecasting Characteristic Poses



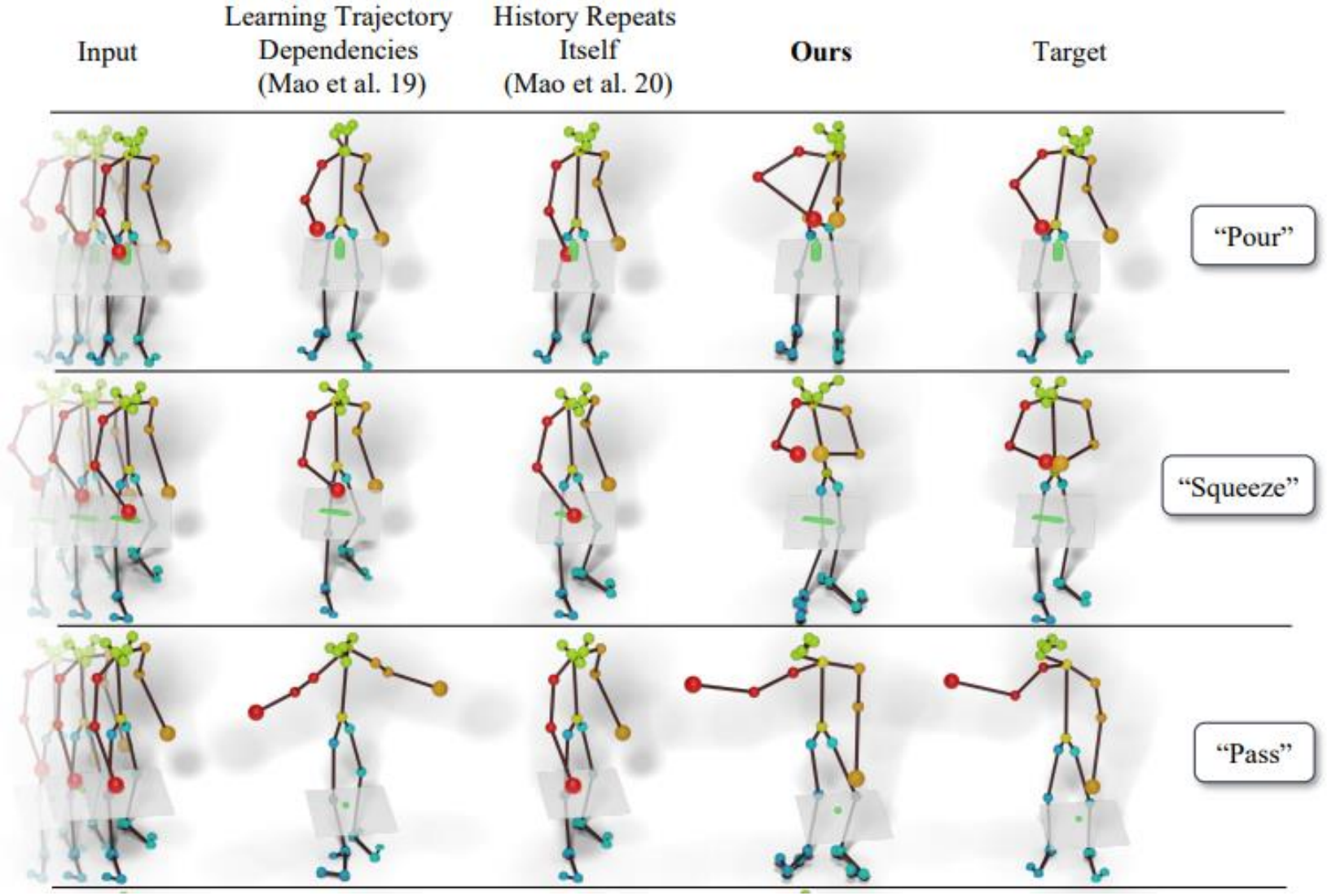
[Diller et al. '21]

Forecasting Characteristic Poses



[Diller et al. ‘21]

Forecasting Characteristic Poses



[Diller et al. '21]

Forecasting 3D Humans from Video

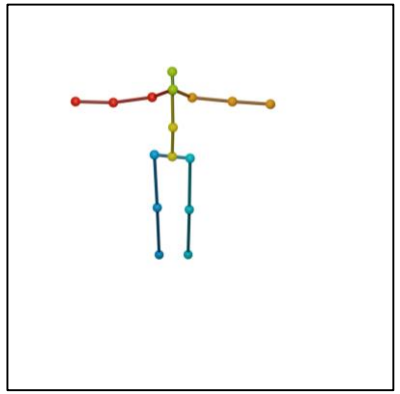
2D Action Sequences

- Take
- Wash
- Take
- Take
- Take
- Close
- Take
- Take
- Peel
- Throw in Garbage
- Cut
- Add
- Throw in Garbage

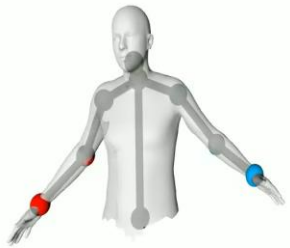
MPI Cooking II [Rohrbach et al. 16]

AMASS
[Mahmood et al. 19]

GRAB
[Taheri et al. 20]



Human3.6m
[Ionescu et al. 13]

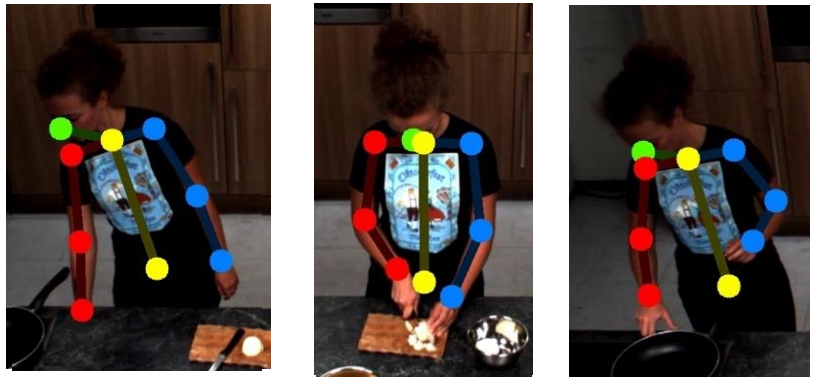


No correspondence

Joint 3D Pose and Action Forecasting

Input Sequence

2D Human Pose Observations



Actions

"take" "cut" "change temperature"

Objects

drawer, frying pan, hand | knife, onion | hand, stove

Time

Pose History Encoder

Action Encoder

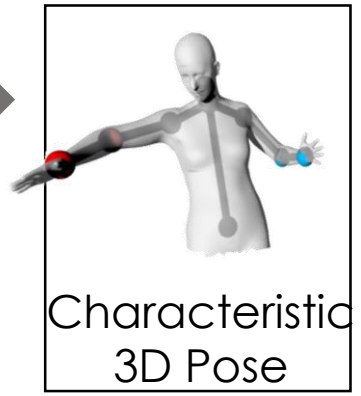
Object Encoder

MLP Decoder

Forecasted 3D Pose + Action

"add"

Action CE Loss



Characteristic 3D Pose

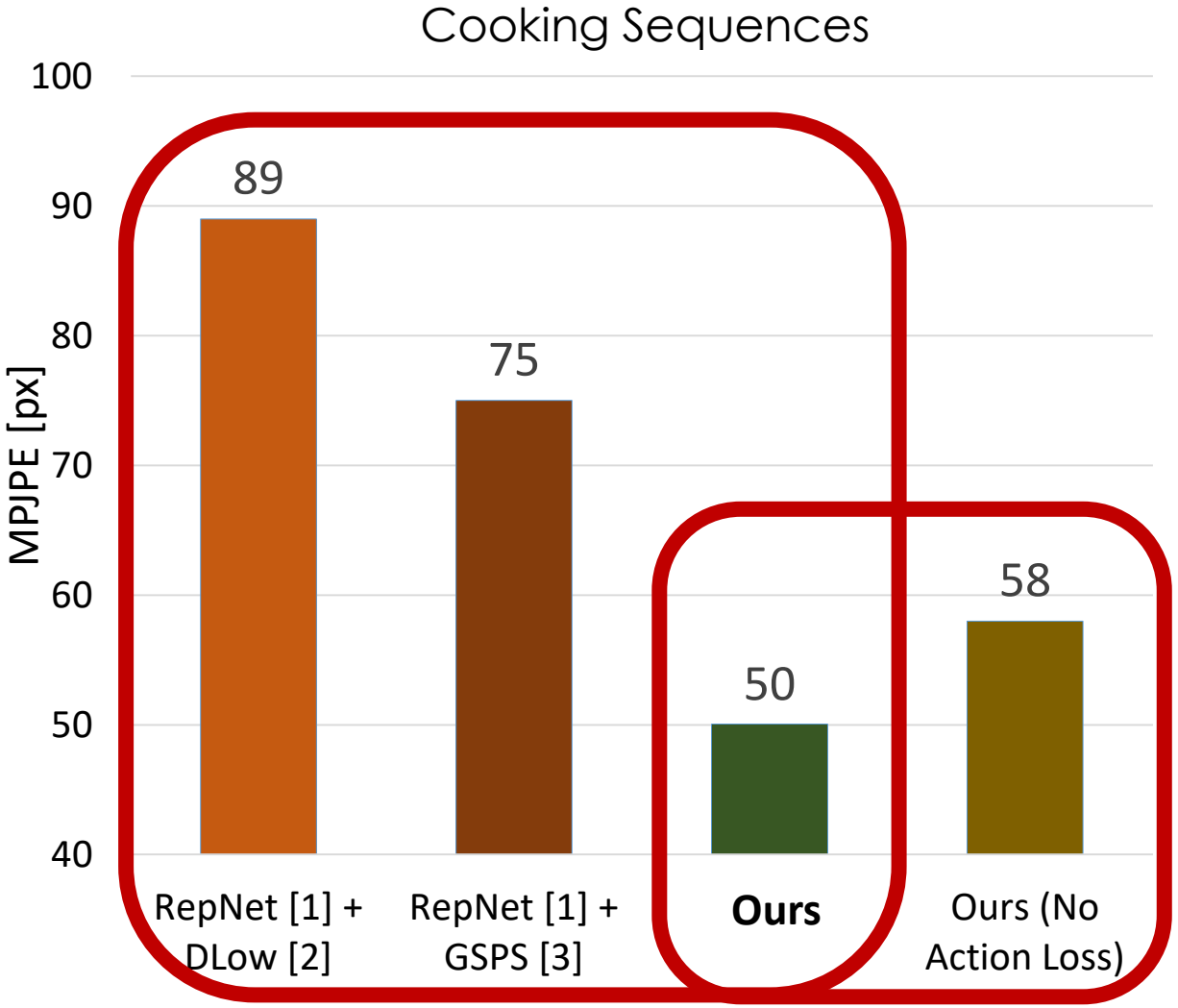
2D Proj.

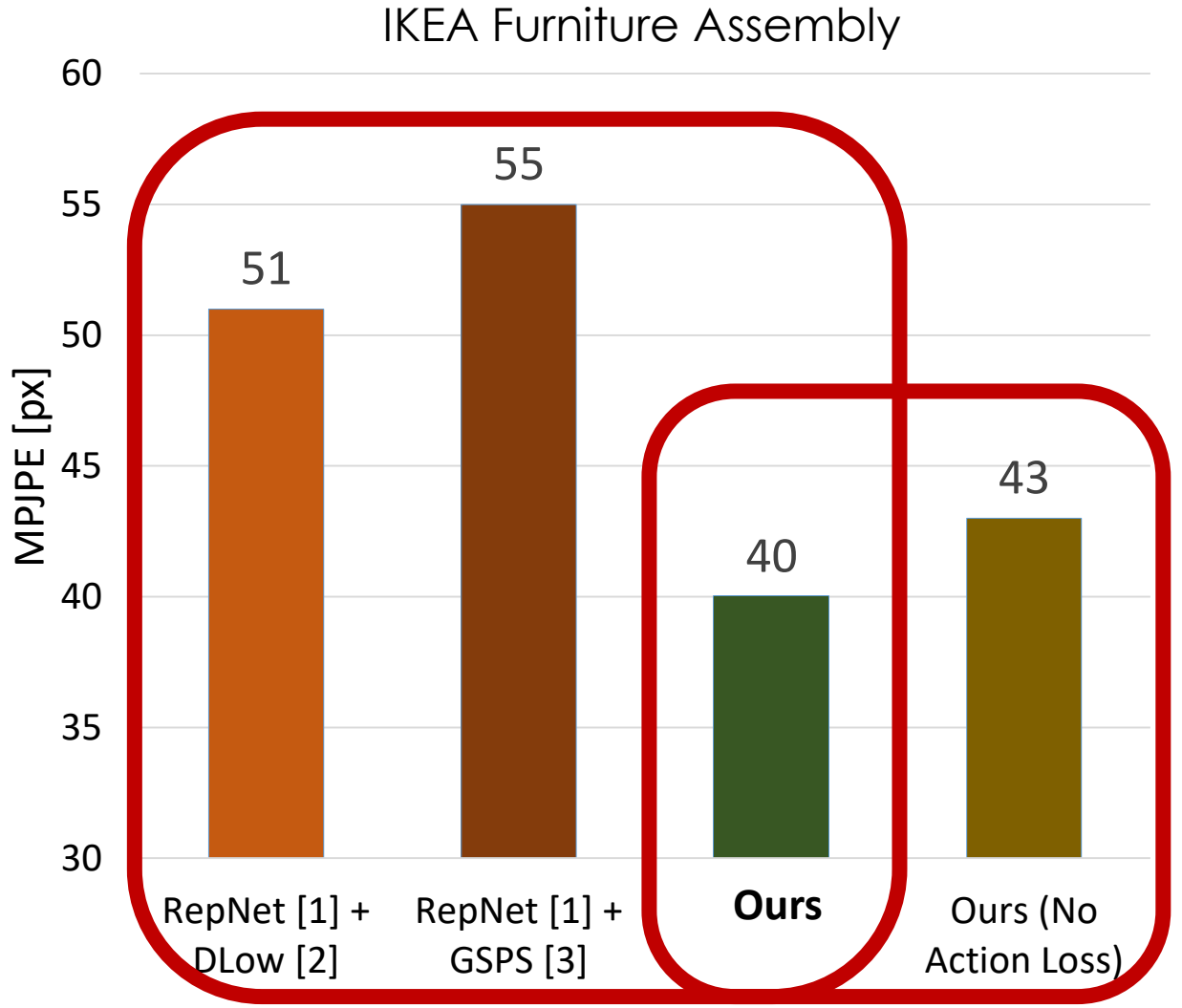
3D Adversarial Loss

2D Joint Loss

[Diller et al.]: Forecasting 3D

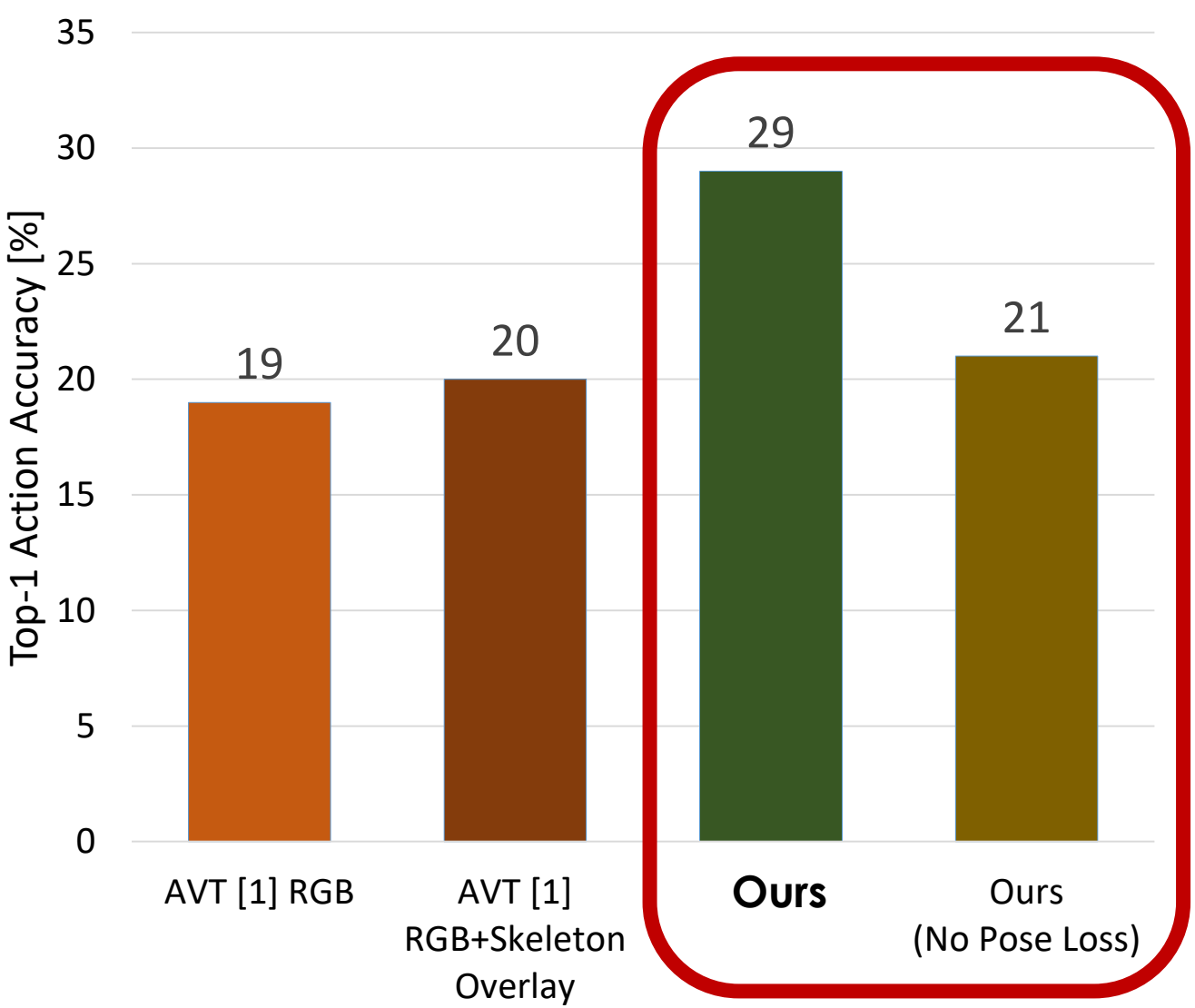
Pose Forecasting: 2D Joint Error ↓

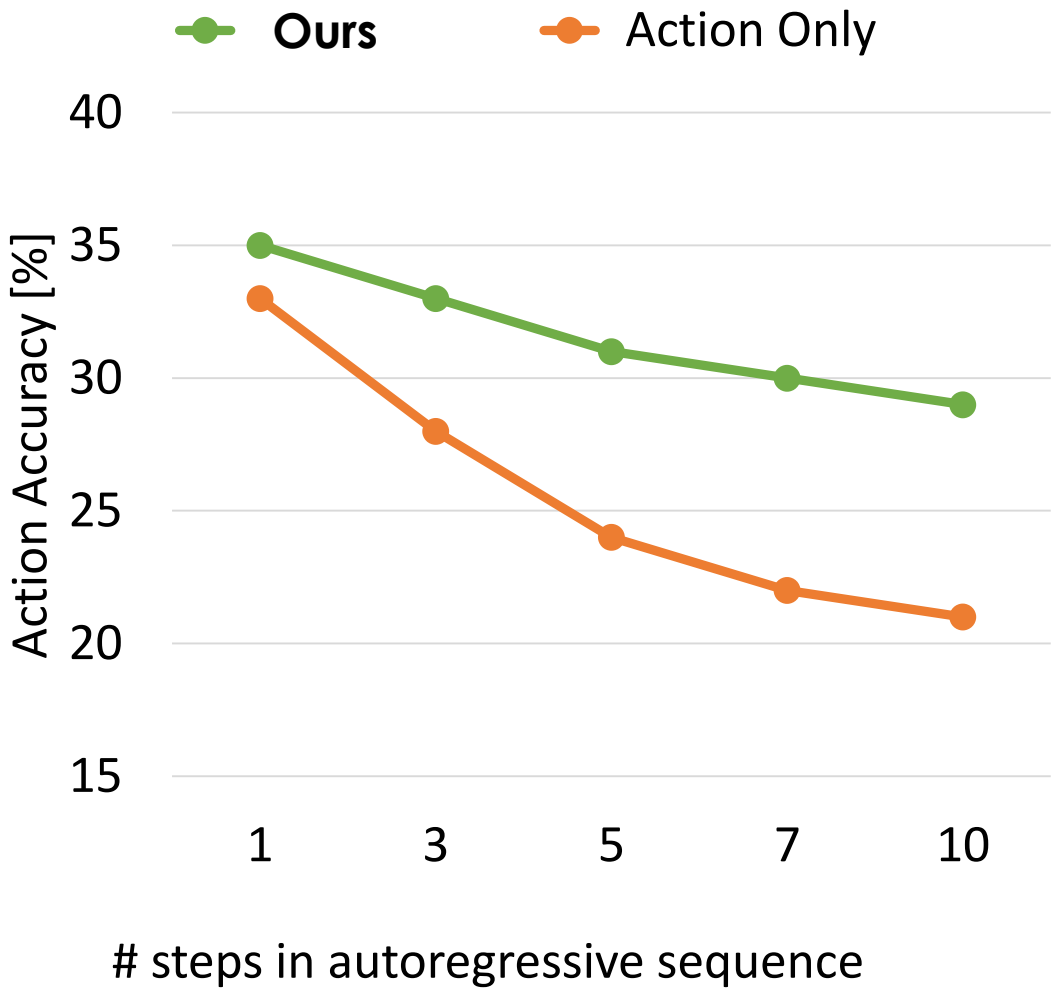




[Diller et al.]: Forecasting 3D

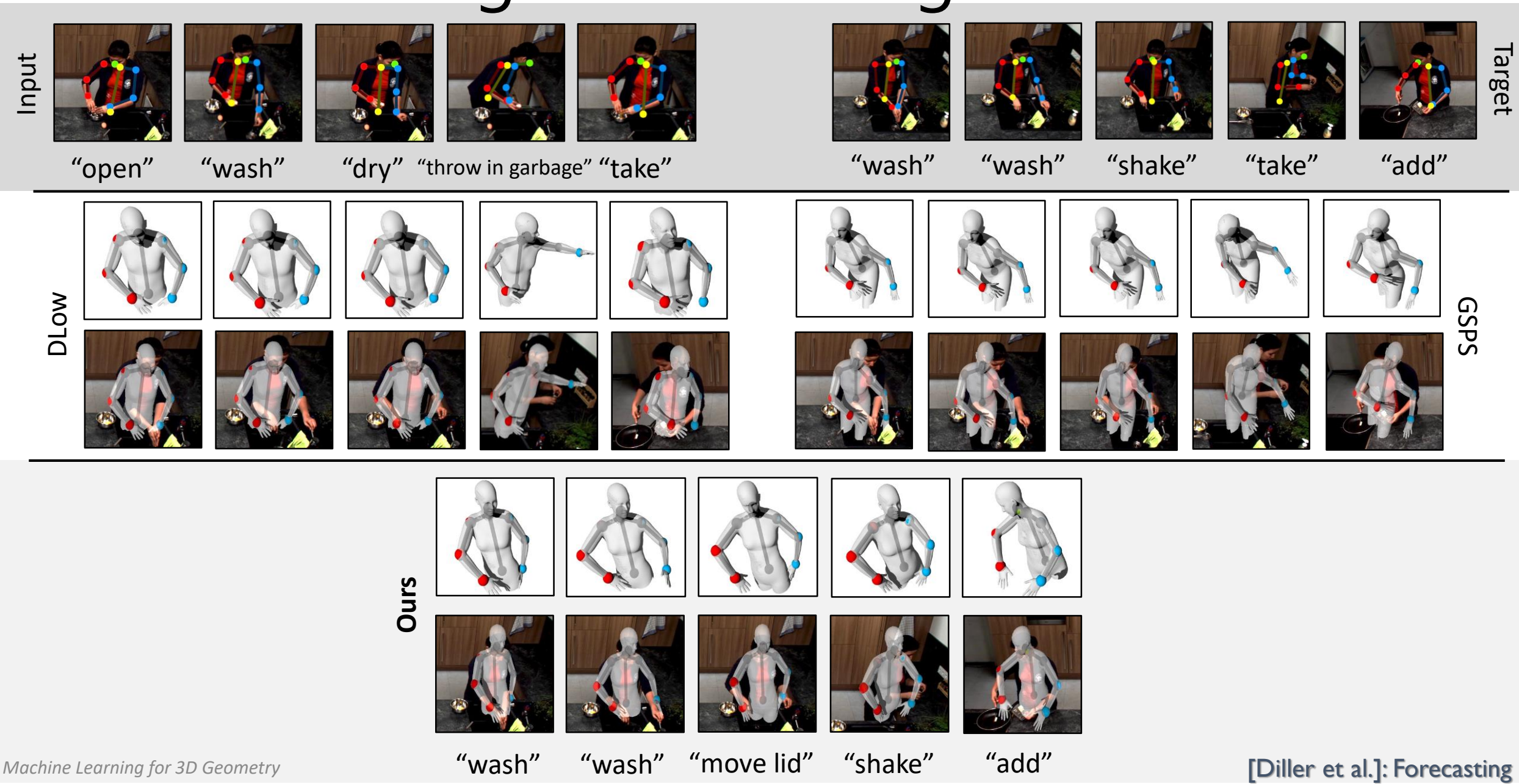
Action Forecasting: Action Accuracy ↑





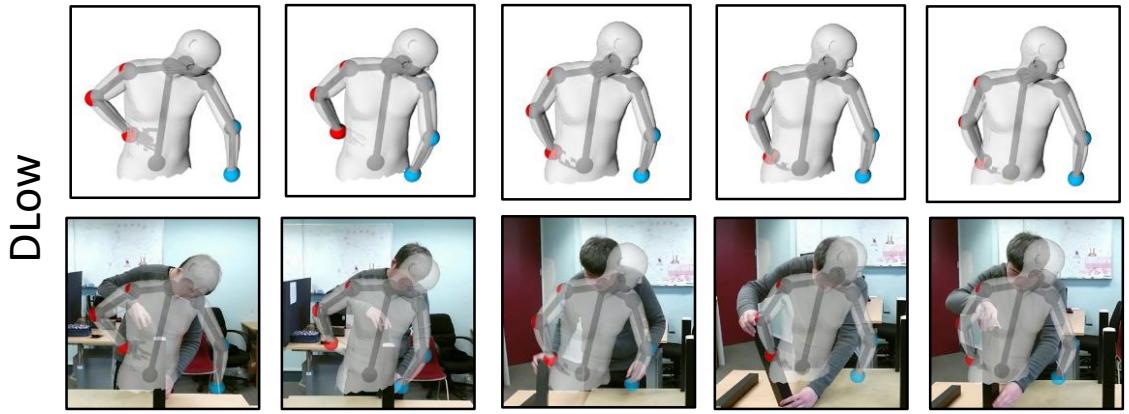
[Diller et al.]: Forecasting 3D

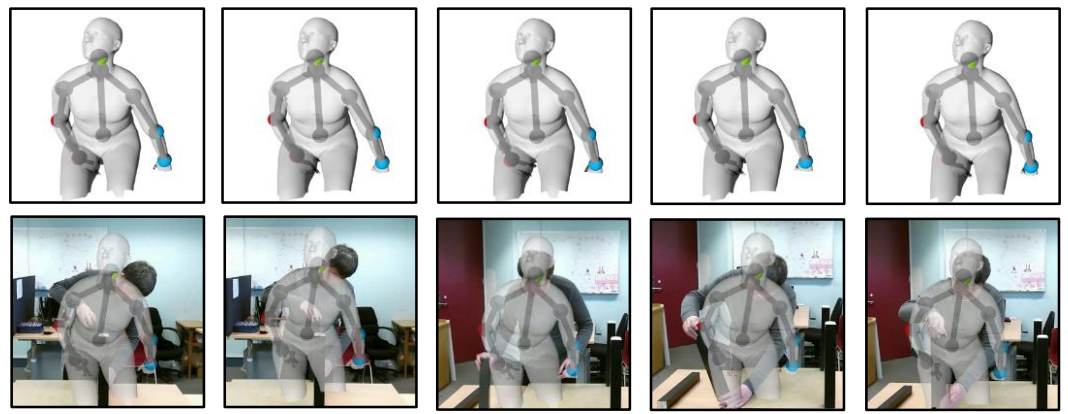
MPI Cooking II Forecasting

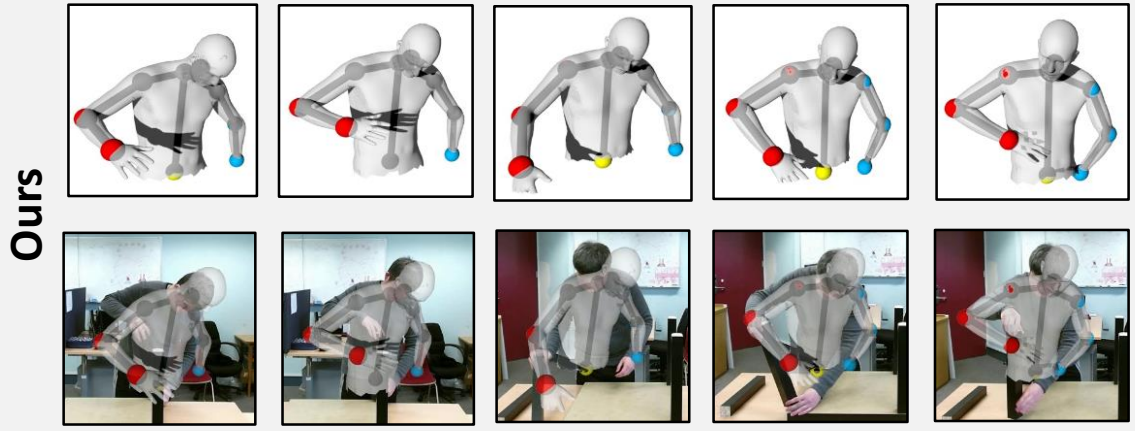


[Diller et al.]: Forecasting 3D

IKEA Assembly Forecasting





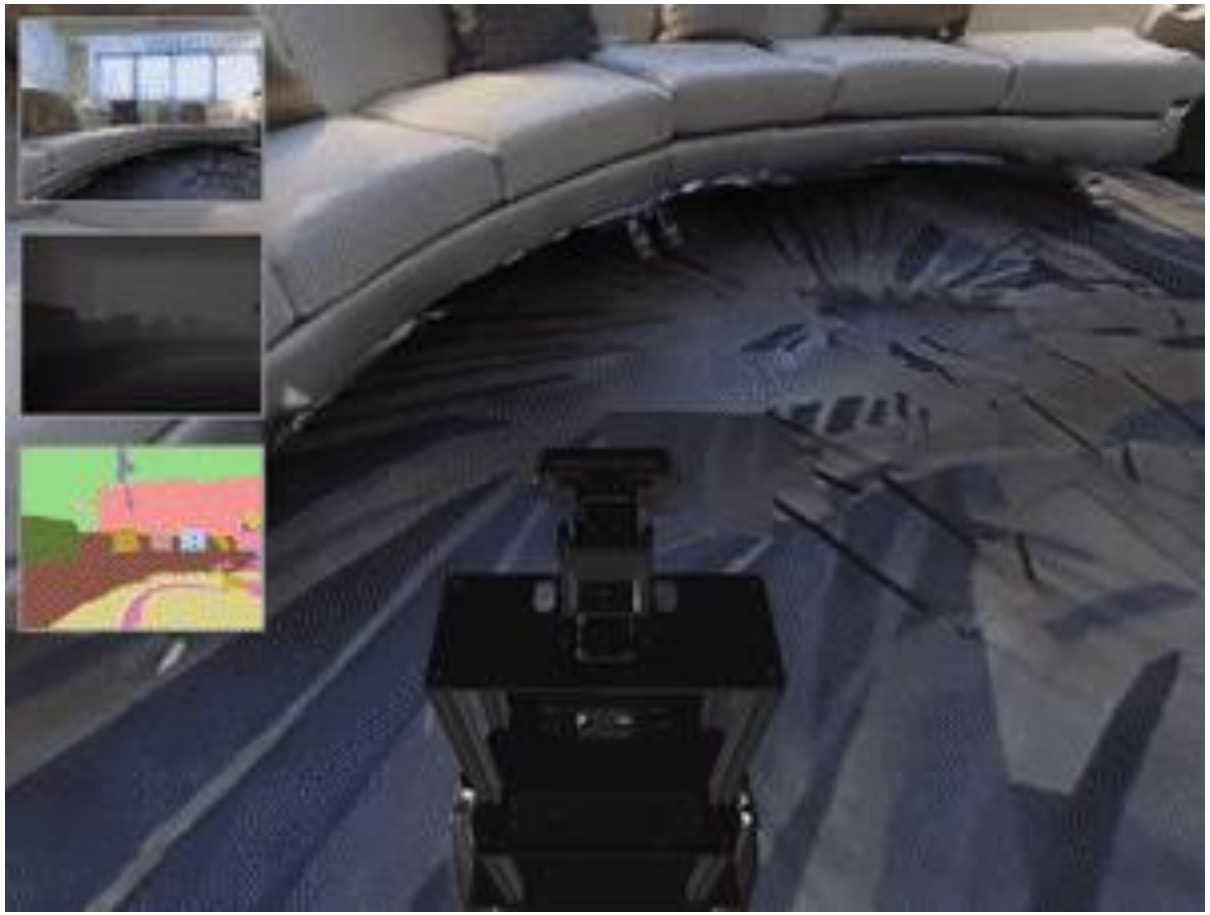


[Diller et al.]: Forecasting 3D

Simulation Environments

- Virtual 3D environment to support training/testing AI agents
- Don’t need to rely on time speed of real-world data collection
- Can simulate more rare or dangerous scenarios
- No real-world material cost
- Easier to reproduce scenarios

Habitat



[Savva et al. ‘19]

Habitat

- 3D simulator for agents, sensors, efficient rendering
- Operates on existing 3D datasets (Matterport3D, Replica, etc)
- Benchmark challenge tasks:
  - PointNav: for an agent at a random start position and orientation in unseen environment, navigate to target specified relative to start position
    - RGB-D camera
  - ObjectNav: for an agent at a random start position and orientation in unseen environment, find an instance of an object category by navigating to it
    - RGB-D camera and GPS+compass sensor

[Savva et al. ‘19]

Habitat

Tasks





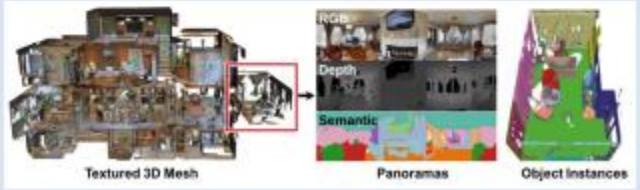

EmbodiedQA (Das et al., 2018) **Language grounding** (Hill et al., 2017) **Interactive QA** (Gordon et al., 2018) **Vision-Language Navigation** (Anderson et al., 2018) **Visual Navigation** (Zhu et al., 2017, Gupta et al., 2017)

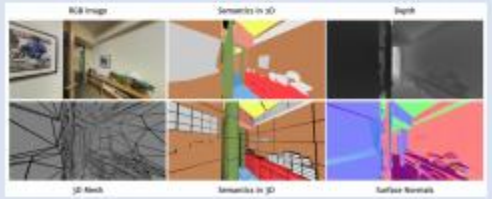
Simulators




House3D (Wu et al., 2017) **AI2-THOR** (Kolve et al., 2017) **MINOS** (Savva et al., 2017) **Gibson** (Zamir et al., 2018) **CHALET** (Yan et al., 2018)

Datasets



Replica (Straub et al., 2019) **Matterport3D** (Chang et al., 2017) **2D-3D-S** (Armeni et al., 2017)

Habitat Platform

Habitat API

Habitat Sim

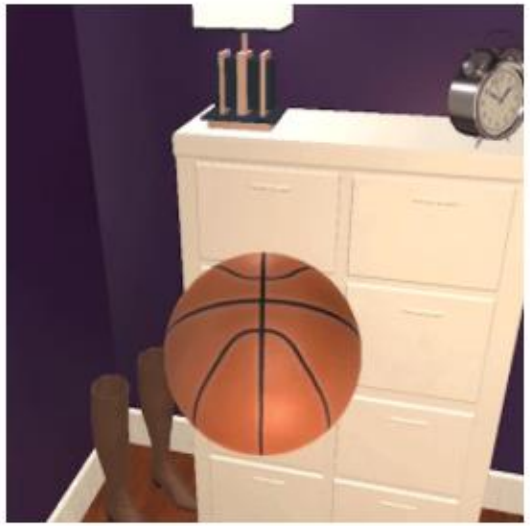
Generic Dataset Support

[Savva et al. '19]

AI2Thor

- Synthetic simulation environment with basic physics, different object states

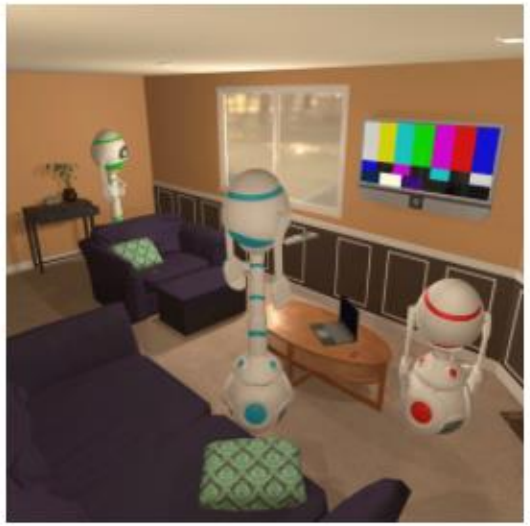
Physics



Object Manipulation

State Changes

Multi-Agent



[Kolve et al. ‘17; Deitke et al. ‘20]

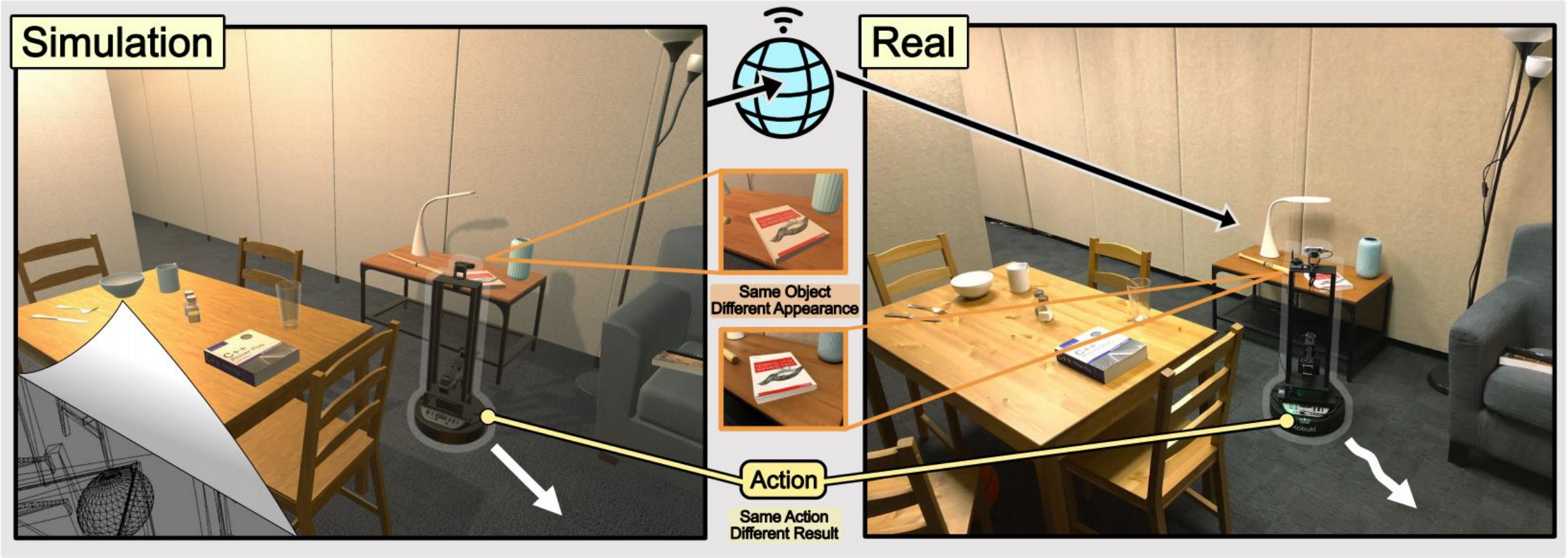
AI2Thor

Environment	Navigable	3D Scene Scans	3D Asset Library	Physics-Based Interaction	Object States	Object Specific Reactions	Dynamic Lighting	Multiple Agents	Real Counterpart
AI2-THOR	✓		✓	✓	✓	✓	✓	✓	✓
iGibson	✓	✓		✓				✓	
Habitat	✓	✓		Collisions					
Matterport3D	✓	✓							
Minos	✓	✓							

[Kolve et al. '17; Deitke et al. '20]

AI2Thor: RoboTHOR

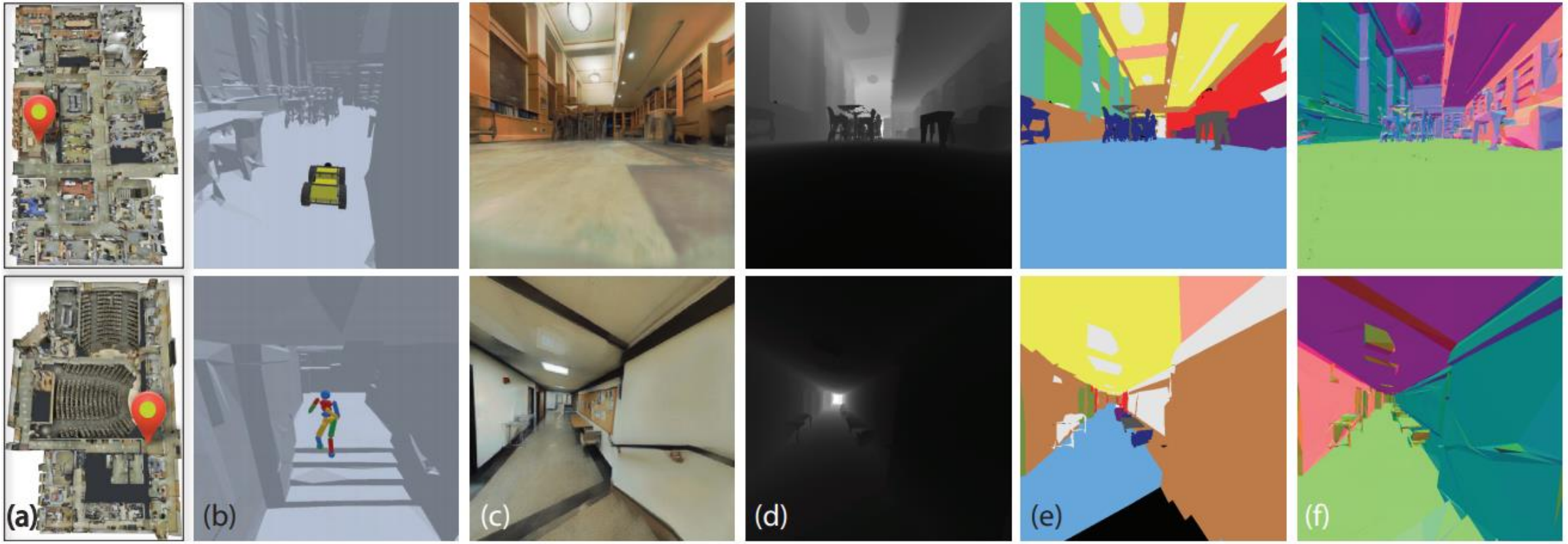
- Synthetic-real correspondences



[Kolve et al. '17; Deitke et al. '20]

Gibson

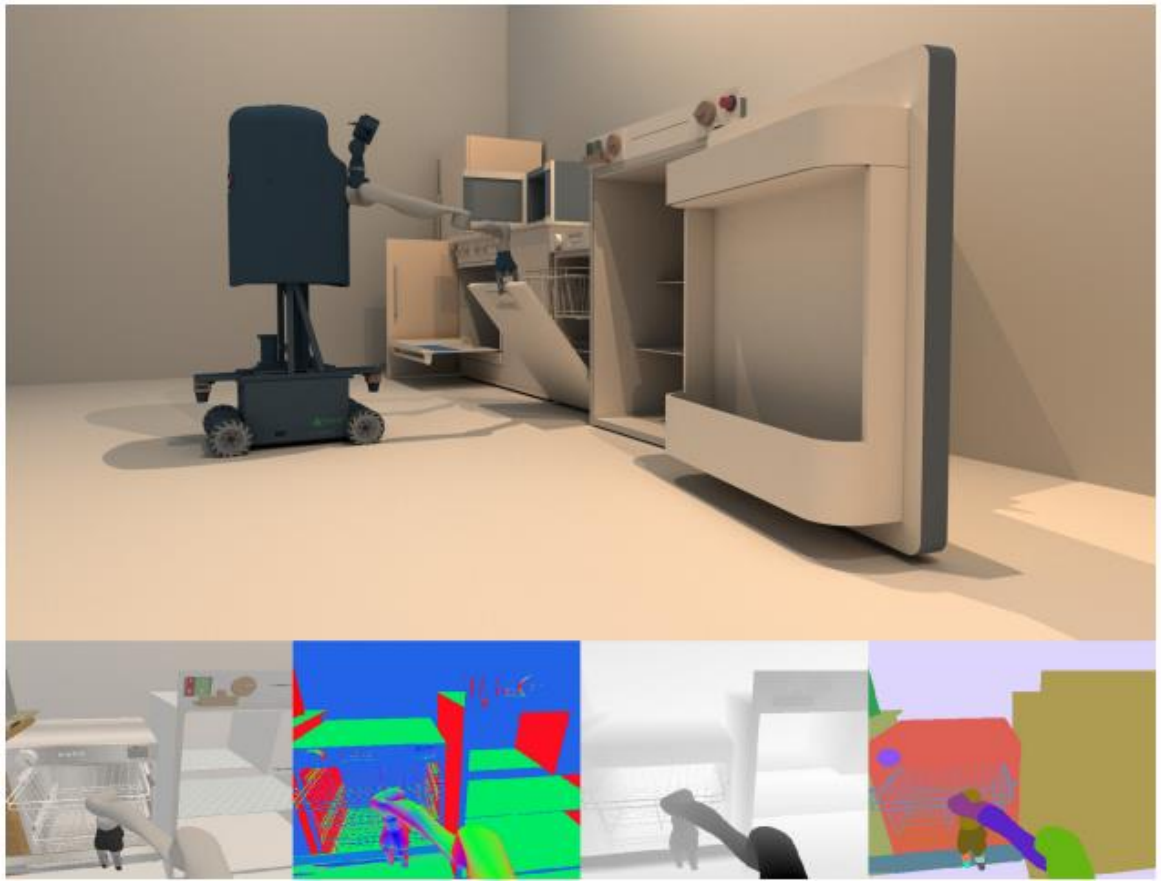
- Simulation environment based on real-world scanned data



[Xia et al. ‘18]

SAPIEN

- A SimulAted Part-based Interactive ENvironment



- Focus on simulation with articulated objects
- Annotation of PartNet with articulations
- Tasks: detect movable parts, estimate motion attributes, open doors and drawers

[Xiang et al. ‘20]