

# Motion Retargeting

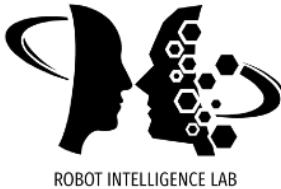
From Optimization-based to Learning-based

Sungjoon Choi, Korea University

# Why Robot Motions?

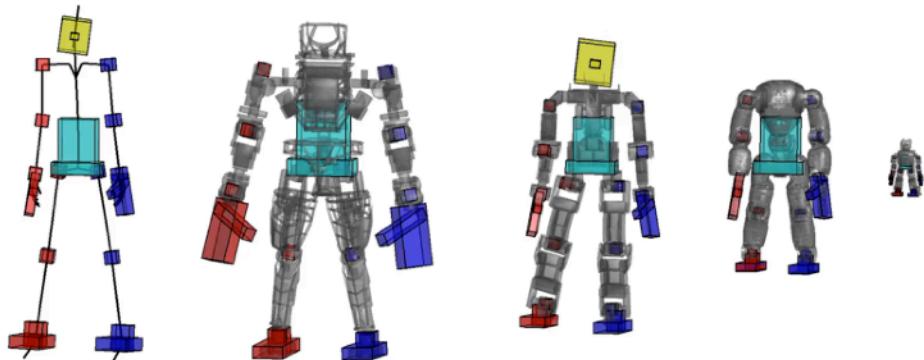


# Motion Retargeting

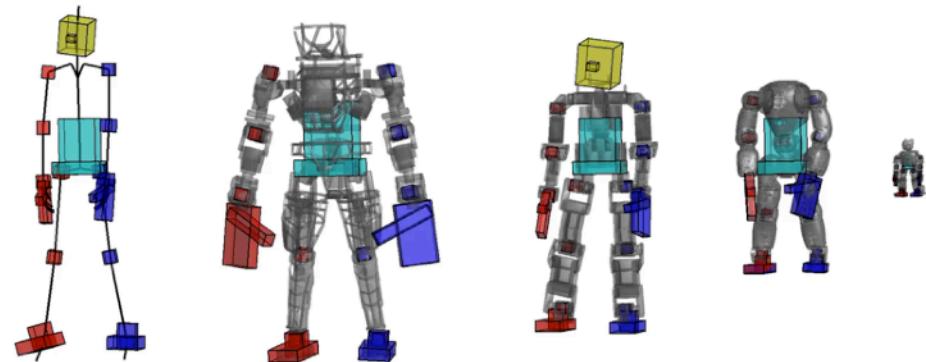


Angry Motion

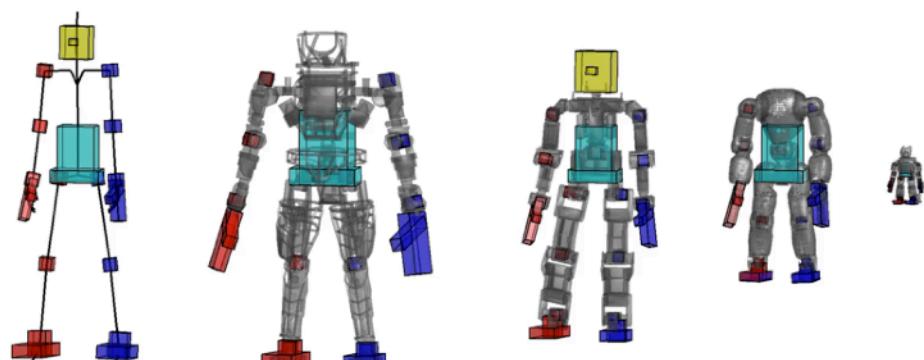
MoCap      Atlas      THORMANG      COMAN      MINI



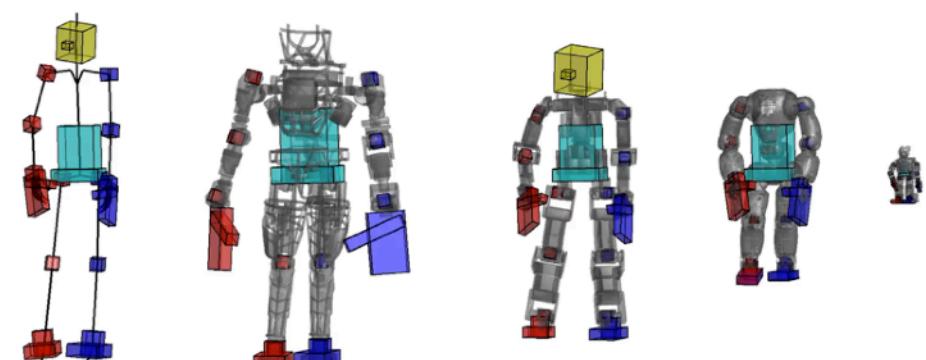
Sad Motion



Happy Motion



Surprise Motion



# Motion Retargeting Tetralogy



ROBOT INTELLIGENCE LAB

## Towards a Natural Motion Generator: a Pipeline to Control a Humanoid based on Motion Data

Sangjoo Choi and Jokyung Kim

**Abstract**—Retargeting the source body motion of human demonstrators or animated characters to humanoid robots is studied in this paper. We propose a pipeline for motion retargeting from source motion data to target humanoid robot. The pipeline consists of two main stages: motion capture and motion transfer. In the first stage, we capture a sequence of source motion data from the target humanoid robot. To this end, we employ a specialized motion capture system which is set up in a smaller scale structure, such as a basement and there is no need to build a large-scale motion capture system. The captured motion data is used to generate a target humanoid motion. In the second stage, we transfer the motion to the target humanoid robot. To this end, we successfully transfer the motion to four different humanoid robots (i.e., Honda Asimo, Toyota i-unit, Mitsubishi MEISTeR, and THIRDMAN) and Alita. Furthermore, OSAMAN and THIRDMAN are selected to show that the proposed method can be applied to physical robots.

### I. INTRODUCTION

Since the first animation human figure of Abraham Lincoln made in famous silent-movie in 1914, the Walt Disney company has been developing a number of famous characters. In addition, motion capture of humanoid robots are common in many amusement parks, such as in smaller scale structures, such as basements and there is no need to build a large-scale motion capture system. The captured motion data is used to generate the retargeted motion field here have a collaboration together to implement realistic shapes and natural motions of actual people and make them look more natural.

Advancement in measurements and graphics technologies, including a motion capture (MCap) system, have enabled us to retarget the motion of one character to another because of robots. Still, however, a related motion generation for a humanoid robot requires a number of choices and limitations. Therefore, it is difficult to find a general and universal methods to handle a large number motion database for robots with different anthropomorphism [1].

In this paper, we propose a pipeline to generate humanogenic movements of a humanoid robot using 2-D motion data acquired from human demonstrators or animated characters. This pipeline consists of two main stages: motion capture and motion transfer. In the first stage, we propose a method for generating motion for a humanoid robot, we can this problem as a motion retargeting problem where the goal is to find a motion that is similar to the target motion but is generated by target robot hardware. This motion retargeting often requires a considerable amount of iterative optimization resulting the robustness of motion retargeting and the efficiency of optimization processes. For example, a real-time motion retargeting

Yangmin Choi and Jokyung Kim are with Disney Research, 300 Park Ave., Suite 1000, New York, NY, USA. (Email: yangmin.choi, jokyung.kim@disneyresearch.com)

## Cross-Domain Motion Transfer via Safety-Aware Shared Latent Space Modeling

Sangjoo Choi<sup>1</sup> and Jokyung Kim<sup>2</sup>

**Abstract**—This paper proposes a data-driven motion retargeting method for cross-domain motion transfer between humanoid robots and human demonstrators or animated characters. We propose a pipeline for motion retargeting from source motion data to the target humanoid robot. To this end, we employ a specialized motion capture system which is set up in a smaller scale structure, such as a basement and there is no need to build a large-scale motion capture system. The captured motion data is used to generate a target humanoid motion. In the second stage, we transfer the motion to the target humanoid robot. To this end, we successfully transfer the motion to four different humanoid robots (i.e., Honda Asimo, Toyota i-unit, Mitsubishi MEISTeR, and THIRDMAN) and Alita. Furthermore, OSAMAN and THIRDMAN are selected to show that the proposed method can be applied to physical robots.

**Index Terms**—Deep Learning in Robotics and Autonomous Motion and Path Planning, Collision Avoidance

### I. INTRODUCTION

GENERATING expressive and lifelike motion for humanoid robots is becoming more and more important considering both the kinematic structure of 3-D motion data and the task-space functioning. Thus we present a cost effective motion retargeting method for humanoid robots to use a global optimization method that considers the similarities between the source and target motion as well as the constraints of target motion. This problem is often referred to as motion retargeting.

A large portion of motion retargeting studies in robotics have focused on motion retargeting from a single domain to another domain, such as motion retargeting from source motion capture (MCap) data and target configuration with additional constraint functions such as a balancing constraint [1–10]. However, motion retargeting studies have difficulties in flexibility and scalability, such as incorporating different target domains, as it is based on specific domain expertise.

On the other hand, data-driven methods have been used for motion retargeting problems to quickly generate motion such as Gaussian process latent variable models (GP-LVM) [11] or deep latent variable models with variational autoencoders (VAE) [12]. These methods have been widely used for motion retargeting, a shallow learning [13]. However, following studies [14–16] conducted translating motion to shared latent space, which is a common space shared by source and target motion. This shared latent space is generated by combining shared latent space between different domains with Gaussian interpolation [11, 12]. While learning-based motion retargeting methods have been proposed, they are based methods with respect to stability and flexibility, safety constraints, and so on.

Motion retargeting was originally presented in [17] based on a mapping function between two different domains with shared latent space modeling [18]. Specifically, the encoder and decoder networks of both source and robot pose domains are learned, which are trained to map source domain data to target domain data. Once the mapping from each domain (source domain and target robot domain) to the shared latent space is

learned, motion retargeting is performed by generating target motion by solving a constrained optimization problem [17].

Motion retargeting has been studied in various fields, such as game engines, robotics, and computer graphics.

The authors would like to thank the Associate Editor and Reviewers' comments.

This work was partially supported by grants from the National Research Foundation of Korea (NRF) funded by the Ministry of Science and Technology.

Address correspondence to S. Choi (e-mail: s.choi@kaist.ac.kr).

© 2019 IEEE. All rights reserved. This material may be reproduced by or given to others, and given to other institutions, without prior permission or written agreement from the copyright holders, provided that:

(1) full acknowledgement of the source is made; and

(2) a copy of the material is not offered for sale in any form.

Manuscript received September 10, 2018; revised December 9, 2018; accepted January 6, 2020.

This work was partially supported by grants from the National Research Foundation of Korea (NRF) funded by the Ministry of Science and Technology.

Address correspondence to S. Choi (e-mail: s.choi@kaist.ac.kr).

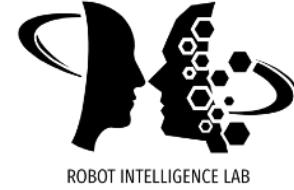
Address correspondence to J. Kim (e-mail: jokyung.kim@kaist.ac.kr).



# Towards a Natural Motion Generator

"Towards a Natural Motion Generator: a Pipeline to Control a Humanoid based on Motion Data", IROS 2019

# Key Question



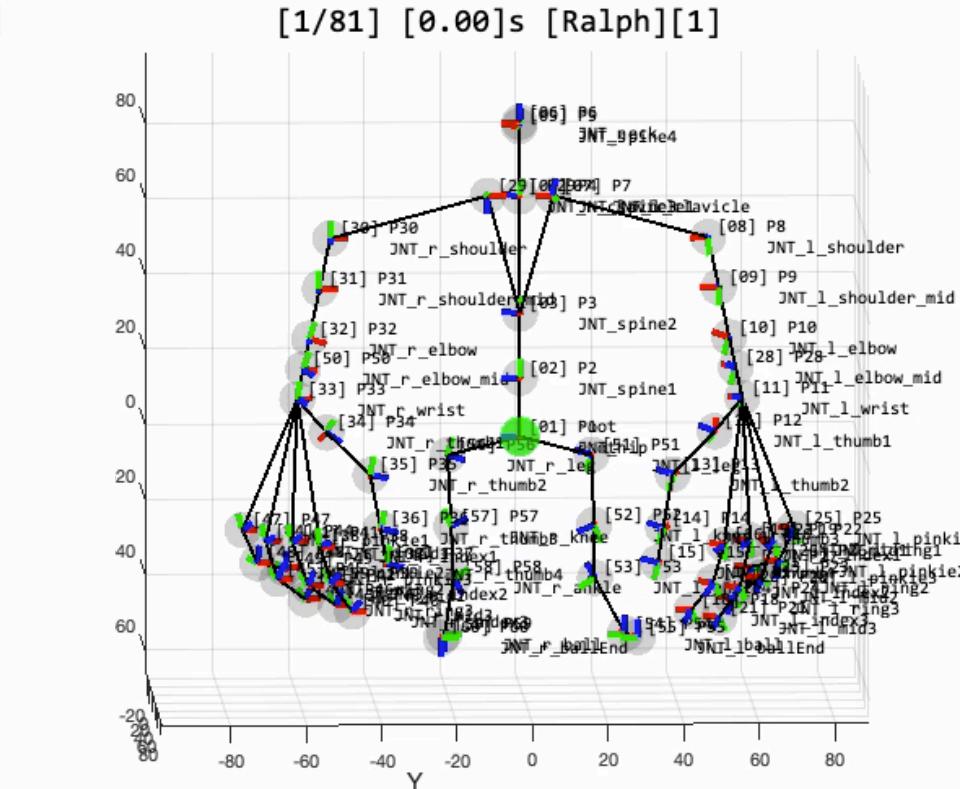
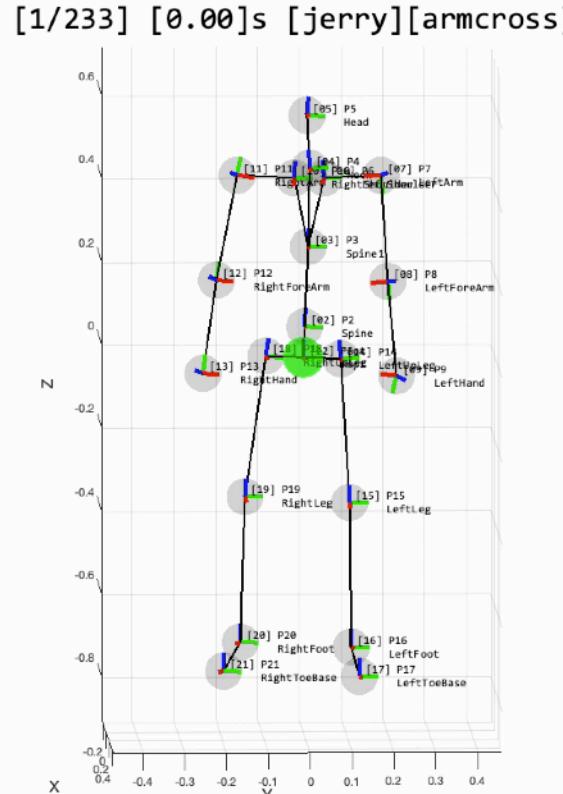
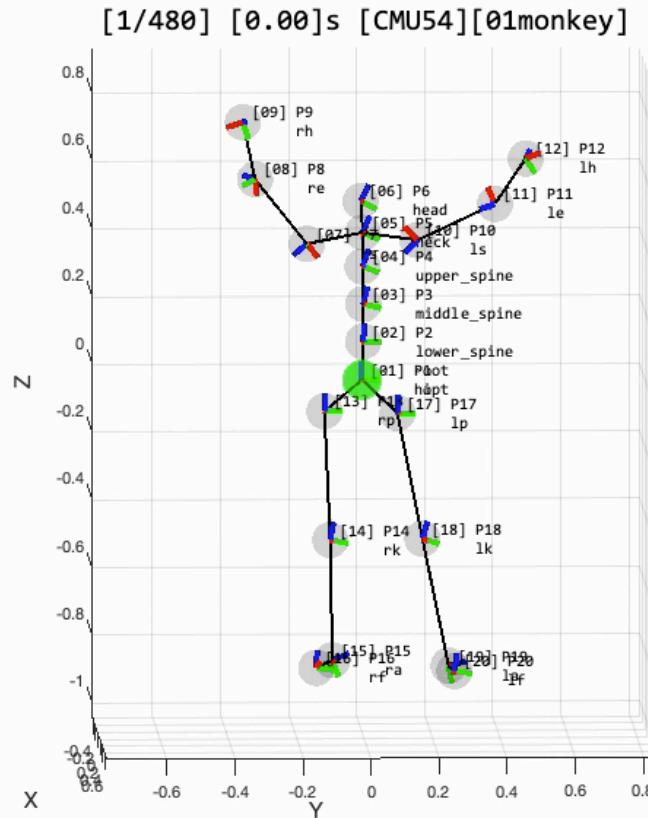
How can we generate **robotics motions** from the **motions of humans or animation characters?**

# Problem 1



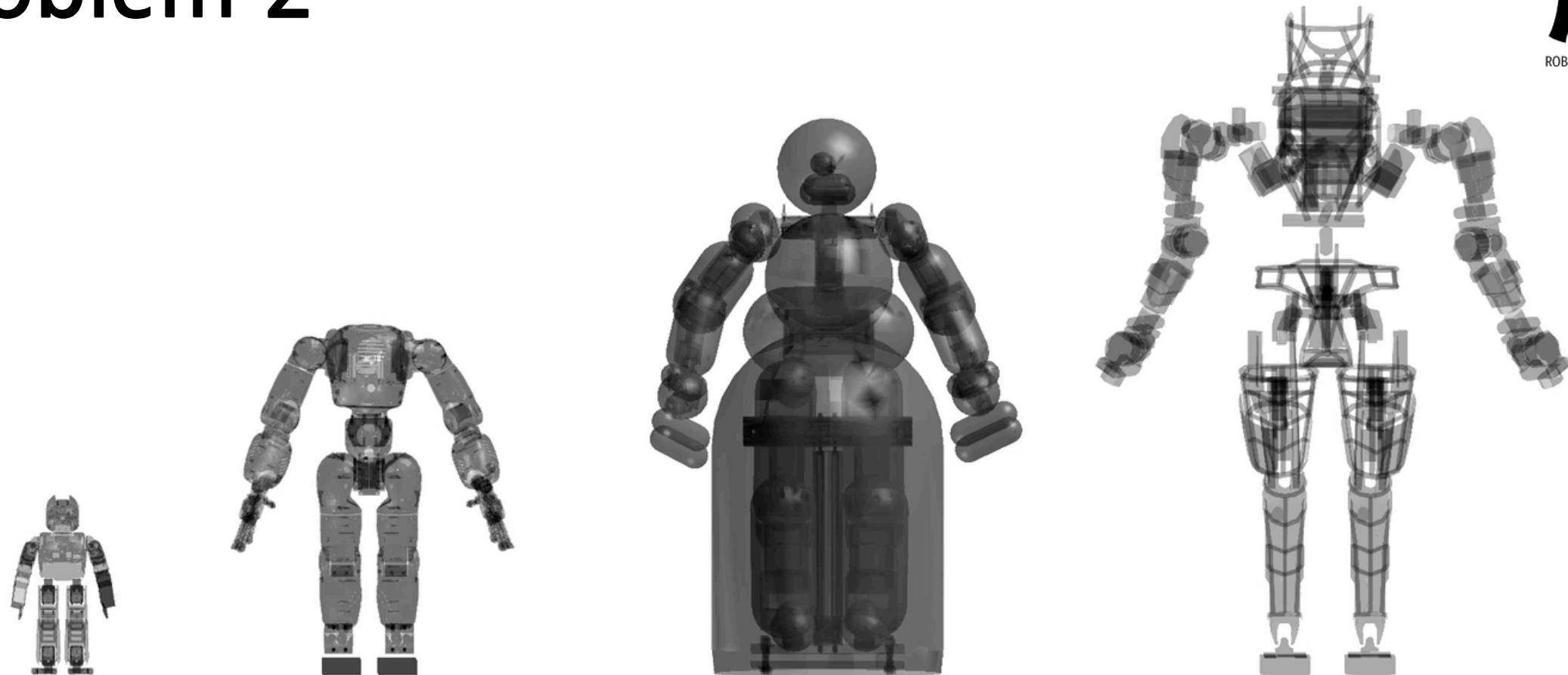
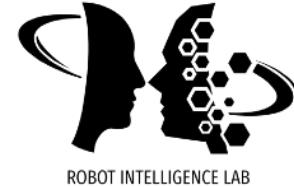
- **Problem 1:** Different **skeletons** from different motion data have significantly different **shapes** and **morphologies**.

# Problem 1



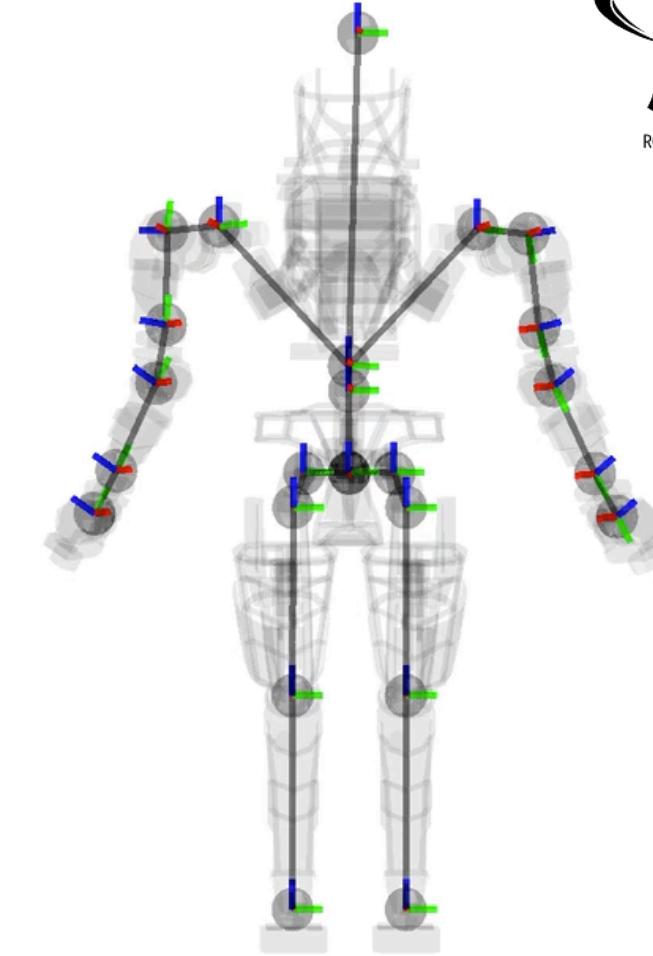
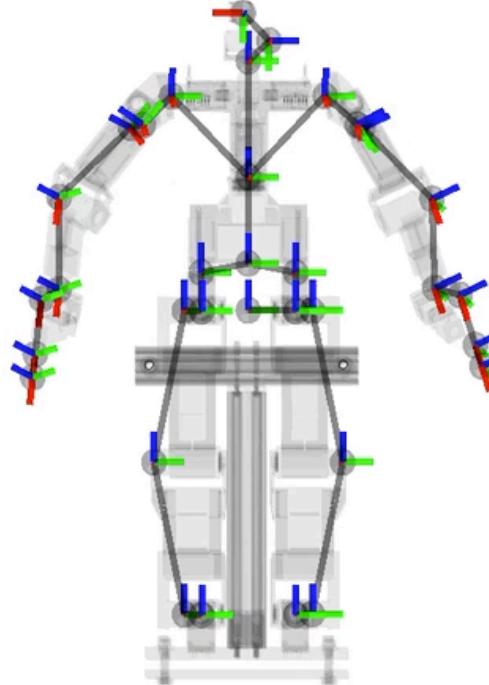
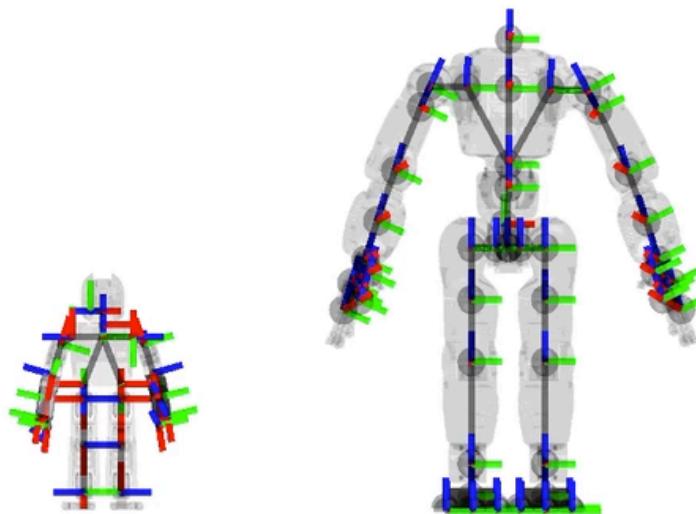
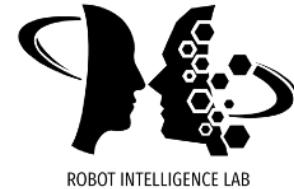
- **Problem 1:** Different **skeletons** from different motion data have significantly different **shapes** and **morphologies**.

# Problem 2



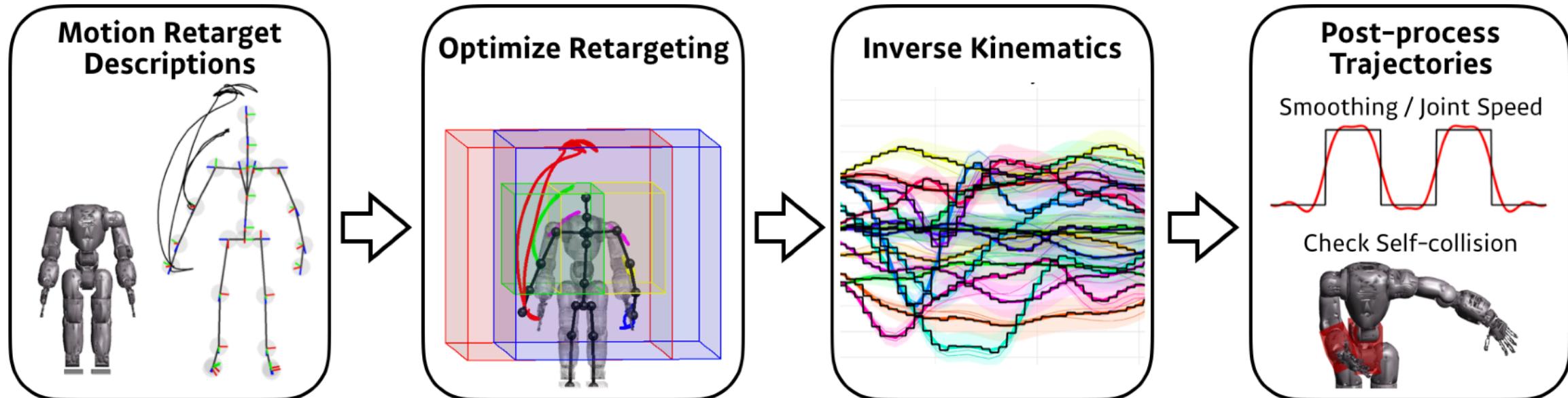
- **Problem 2:** Different **humanoid robots** have significantly **different shapes** and **morphologies**.

# Problem 2



- **Problem 2:** Different **humanoid robots** have significantly **different shapes** and **morphologies**.

# Motion Retargeting

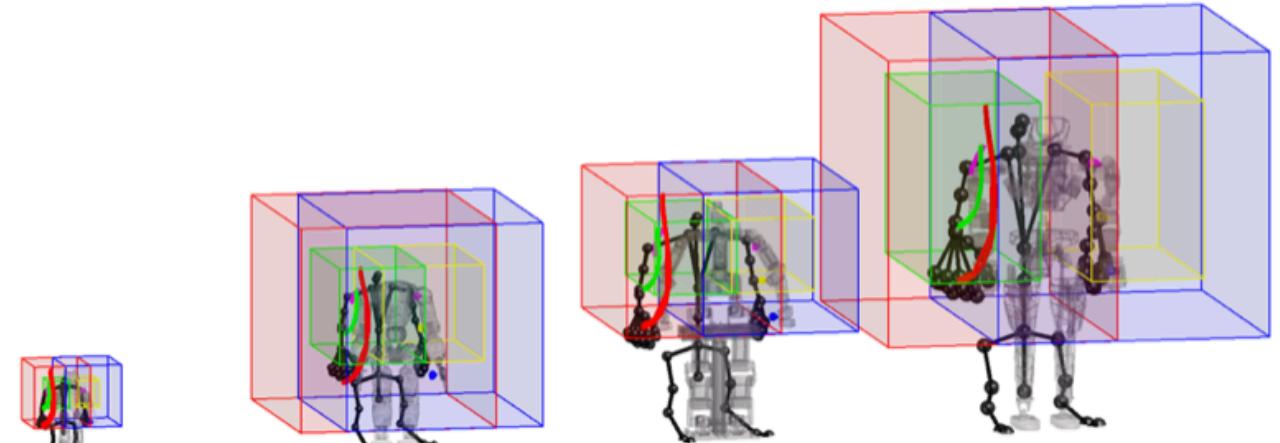
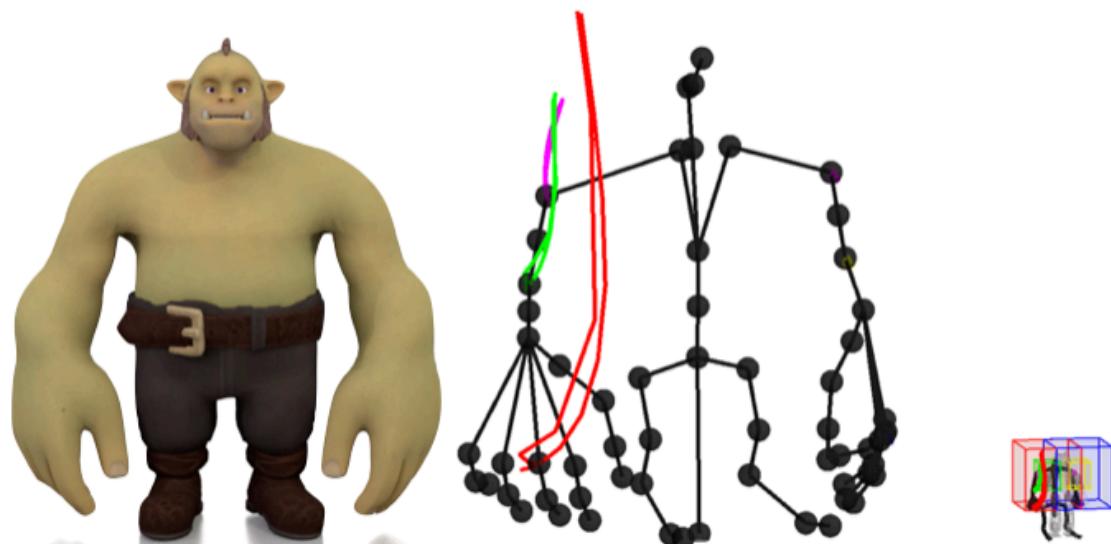
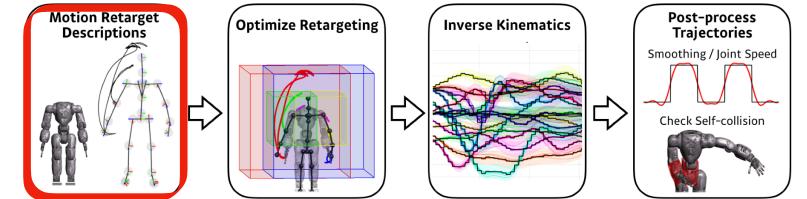


- We present an **optimization-based** motion retargeting method that generates robotic motions from motion capture or animation data.

# Motion Retargeting



ROBOT INTELLIGENCE LAB

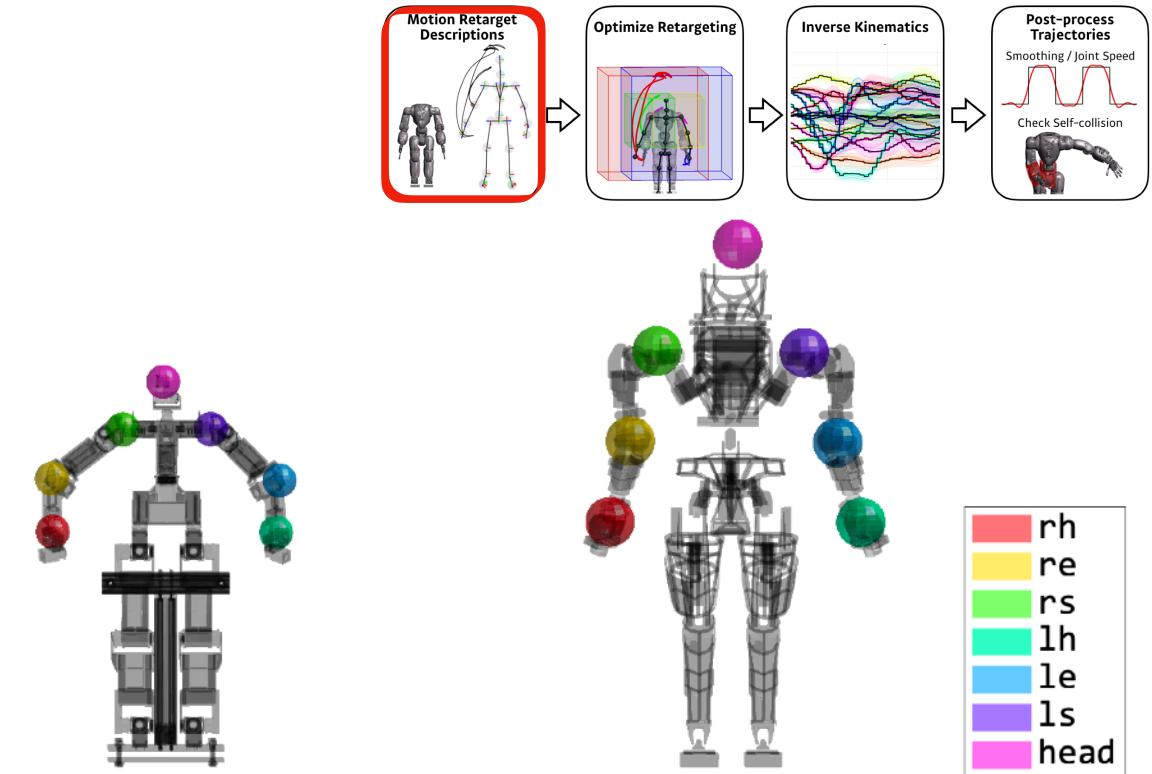
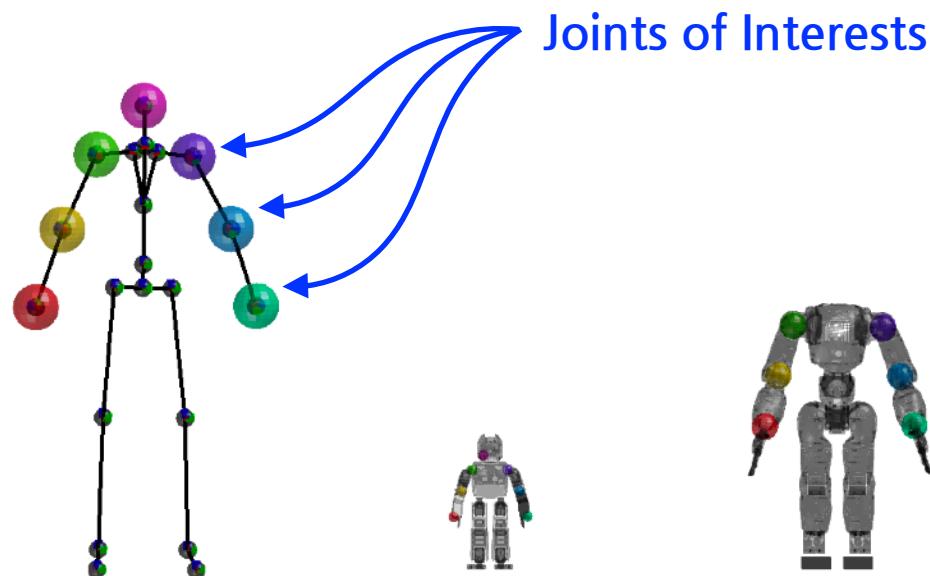


- The cornerstone of the proposed method is a gradient-free optimization-based motion retargeting that adjusts source motion skeleton to fit the target robot.

# Joints of Interest



ROBOT INTELLIGENCE LAB

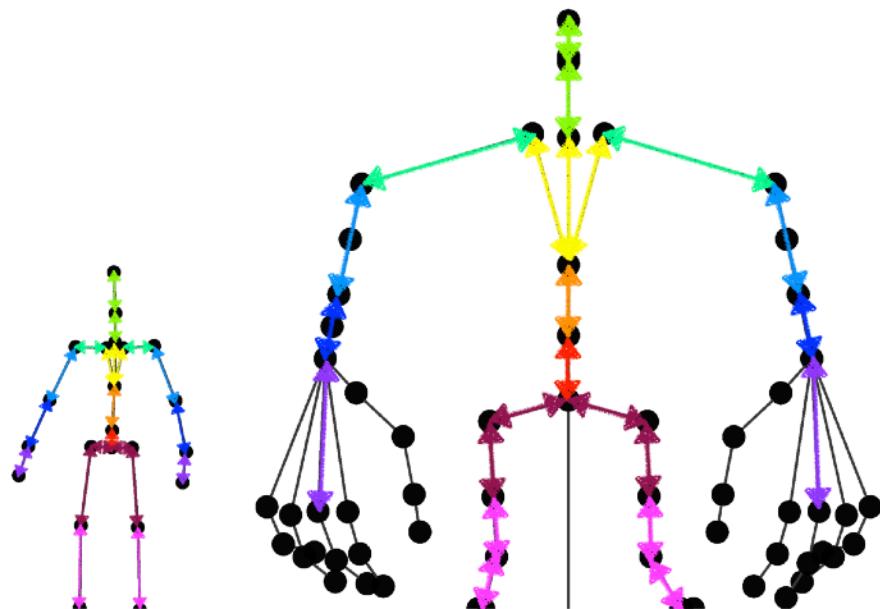


- We first define **joints of interest** of which to transfer from a source skeleton to target humanoid robots.

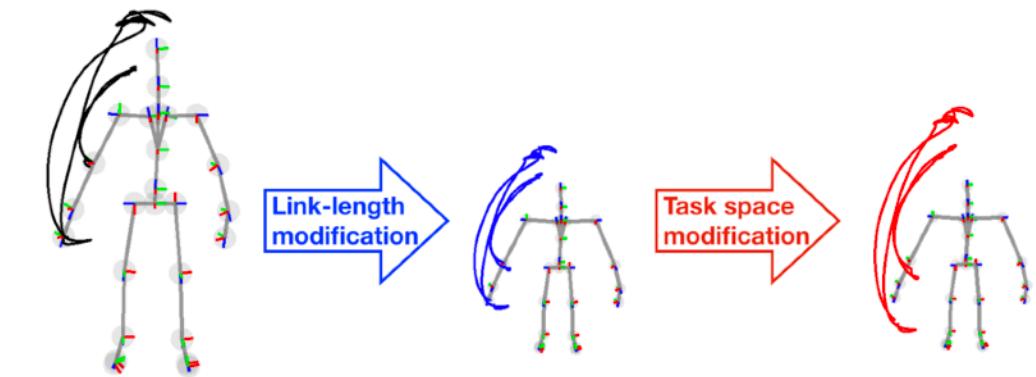
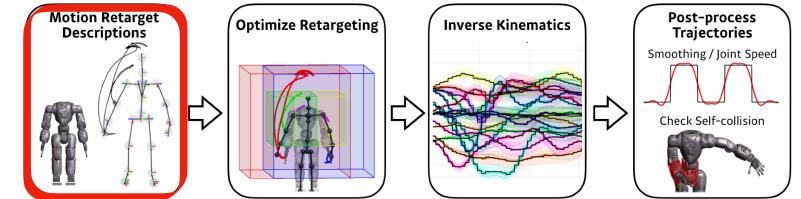
# Motion Transfer Parametrization



ROBOT INTELLIGENCE LAB



- ↔ red → hip to lower spine
- ↔ orange → lower spine to mid spine
- ↔ yellow → mid spine to neck
- ↔ light green → neck to head
- ↔ green → neck to shoulder
- ↔ blue → shoulder to elbow
- ↔ dark blue → elbow to wrist
- ↔ purple → wrist to hand
- ↔ pink → knee to foot
- ↔ magenta → pelvis to knee
- ↔ black → global rate

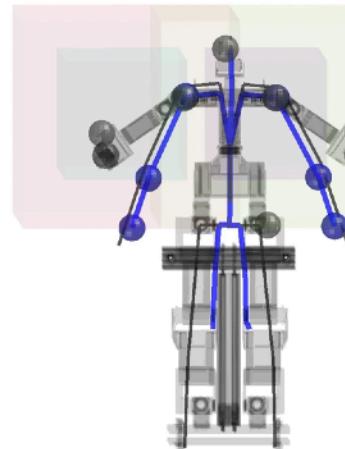
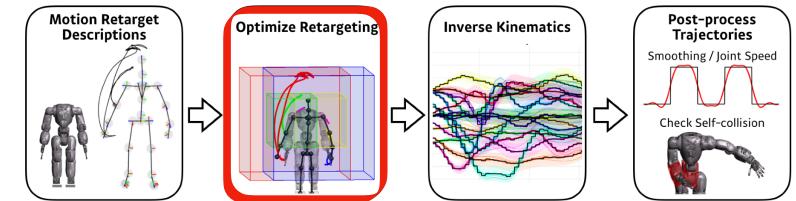
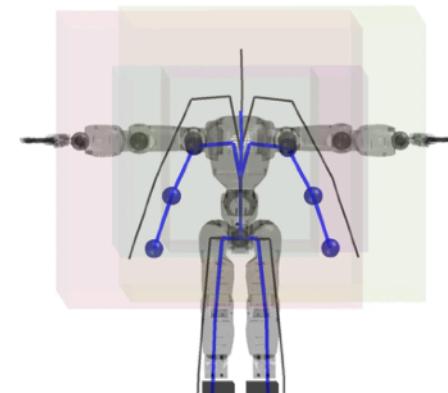
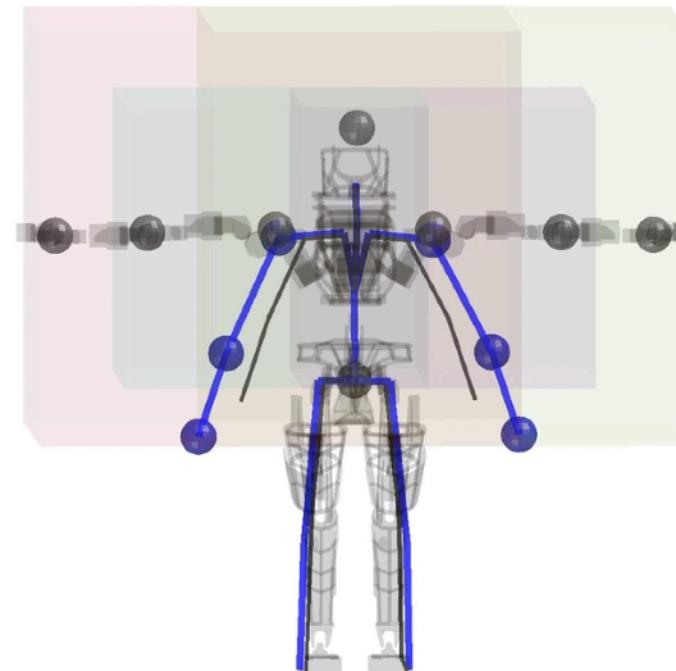


- The motion transfer is parametrized by **link-length modification** followed by **task space modification**.

# Cost Function



ROBOT INTELLIGENCE LAB



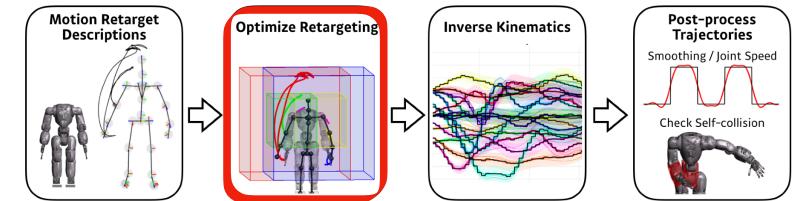
- The **cost function** is composed of four different sub-costs:

1. Location of joints at the idle pose (A-pose).
2. Link lengths between joints of interest.
3. Workspaces of both hands.
4. Relative trajectories between the original and adjusted source motions.

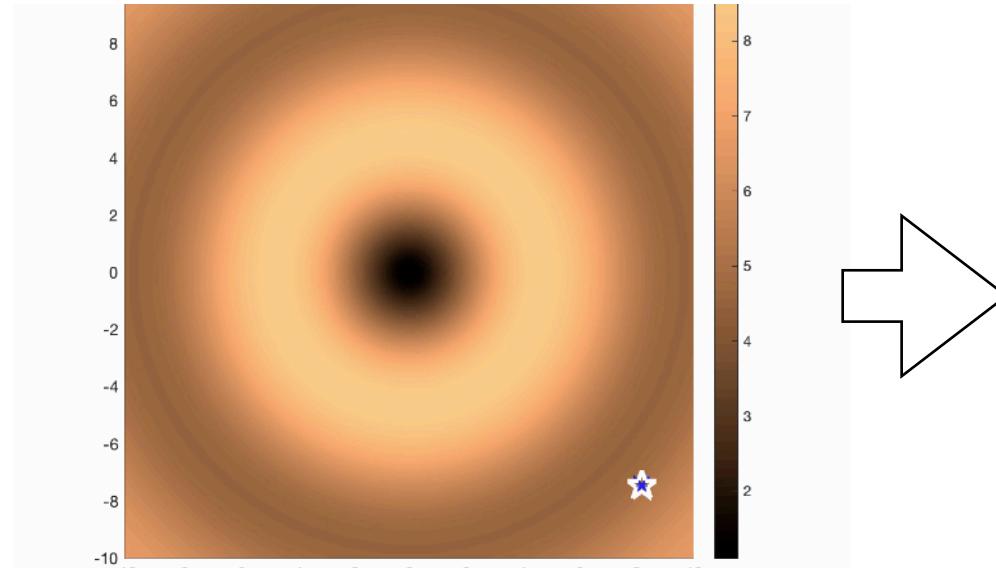
# Black-box Optimization



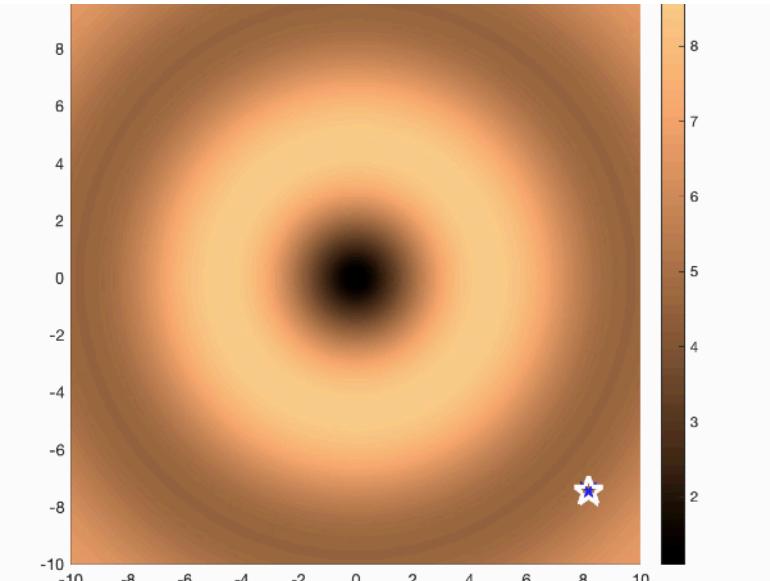
ROBOT INTELLIGENCE LAB



Original Bayesian Optimization



BO + Coordinate Descent

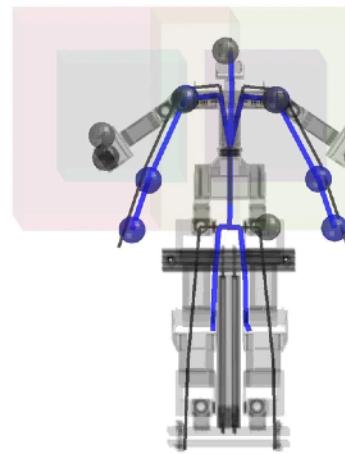
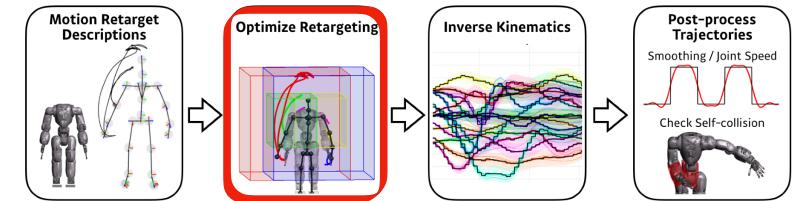
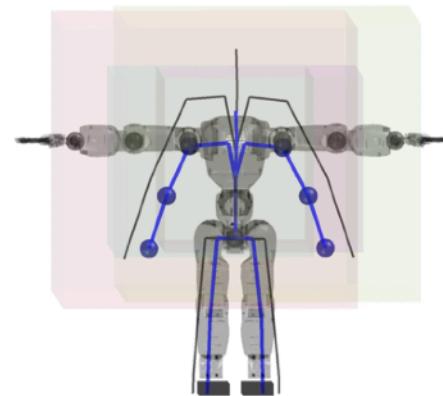
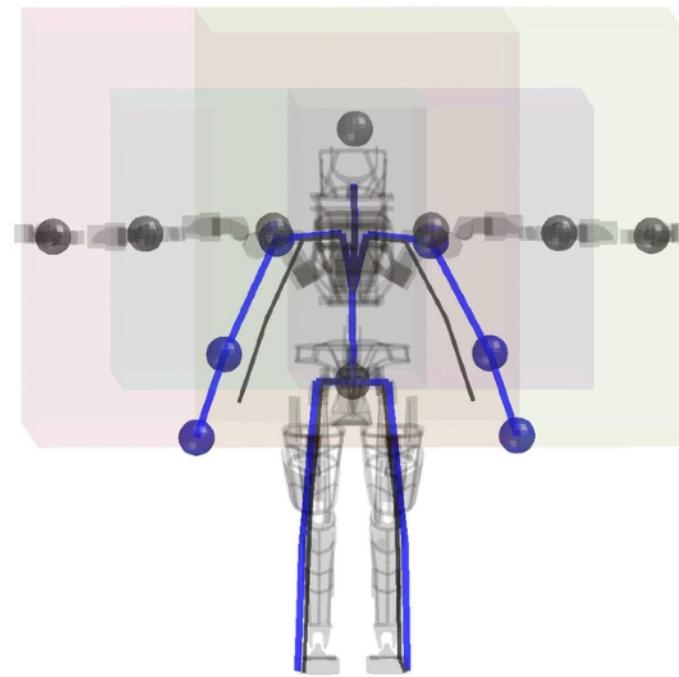


- Once we have the motion retargeting parametrization and the cost function to optimize, a gradient-free optimization method combining **Bayesian optimization with coordinate descent** is used for motion retargeting.

# Inverse Kinematics



ROBOT INTELLIGENCE LAB

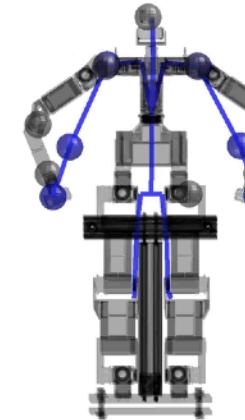
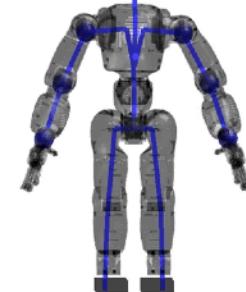
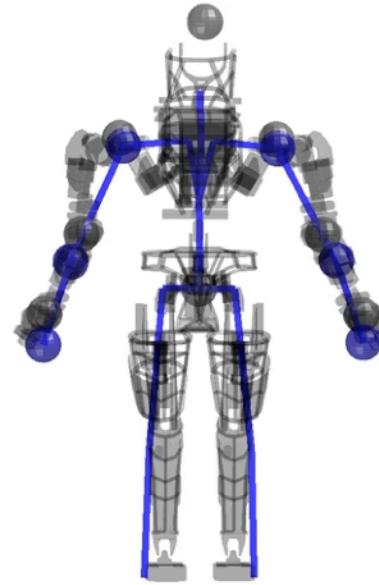
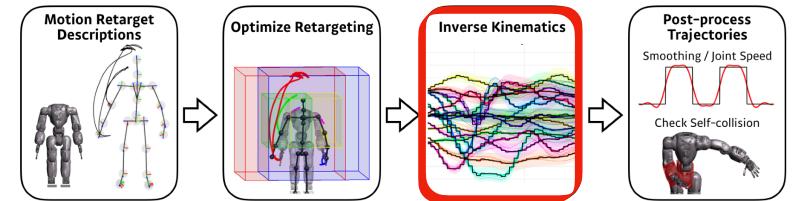


- The joint trajectories of target robots are computed by solving inverse kinematics with multiple target joints using an augmented Jacobian method.

# Inverse Kinematics



ROBOT INTELLIGENCE LAB

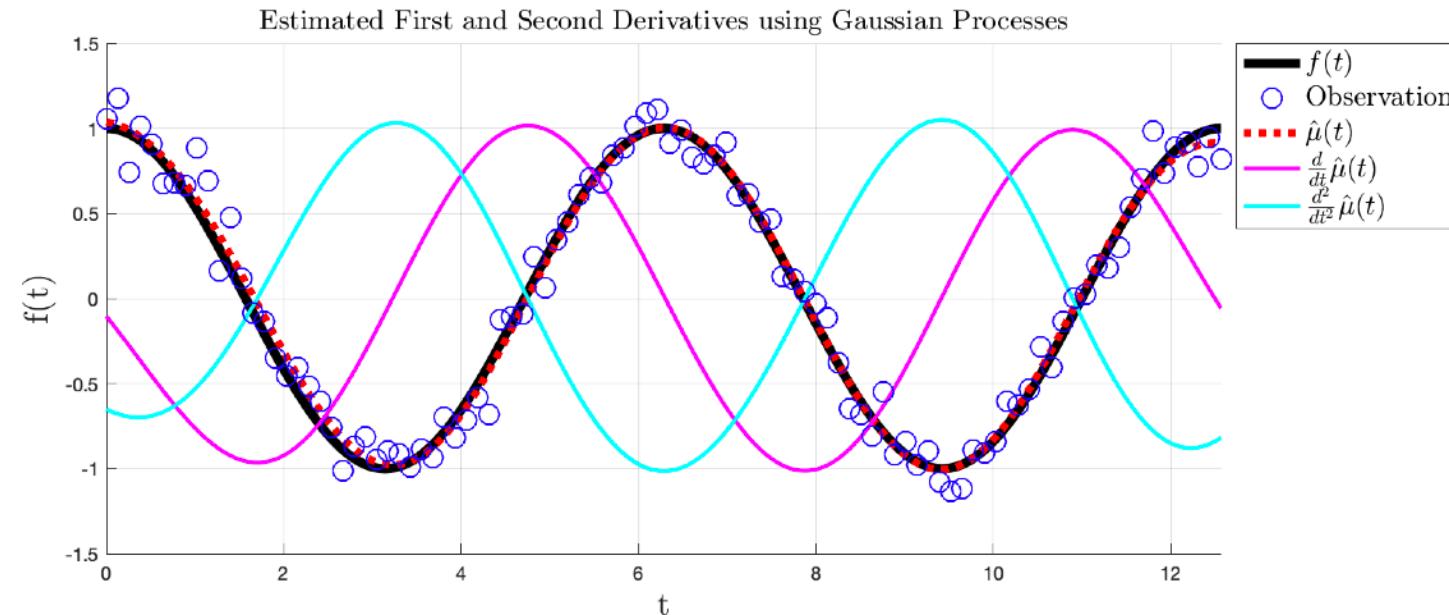
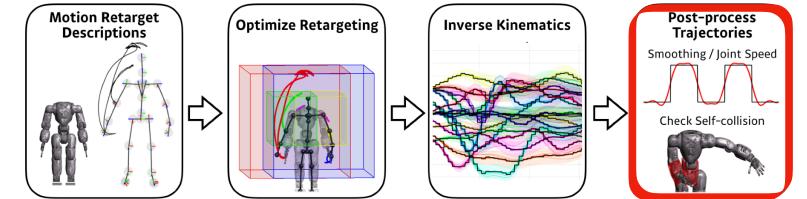


- The joint trajectories of target robots are computed by solving inverse kinematics with multiple target joints using an augmented Jacobian method.

# Post Processing



ROBOT INTELLIGENCE LAB



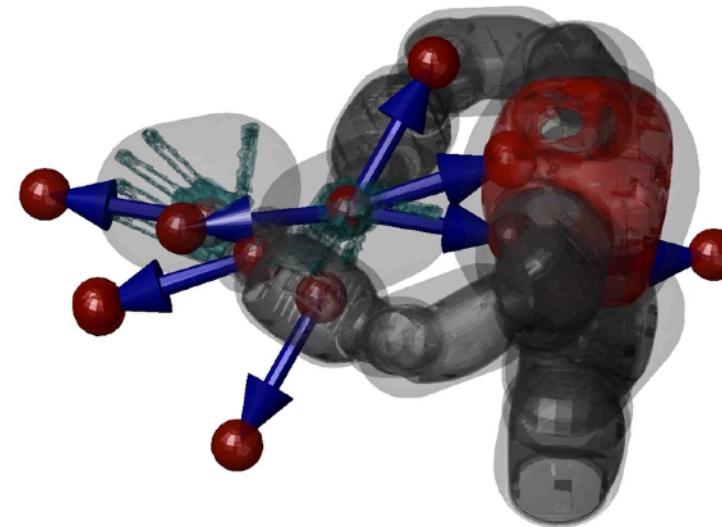
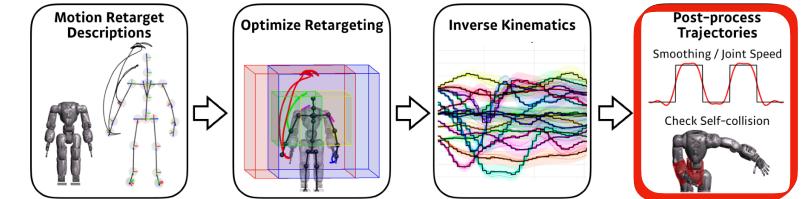
- Additional **trajectory smoothing** using Gaussian process regression as well as **self-collision handling** is processed.

# Post Processing



ROBOT INTELLIGENCE LAB

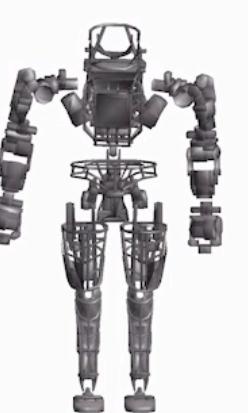
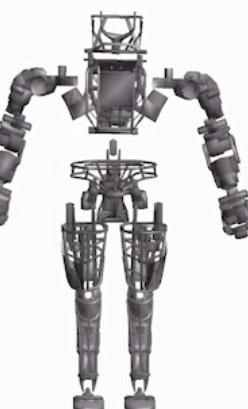
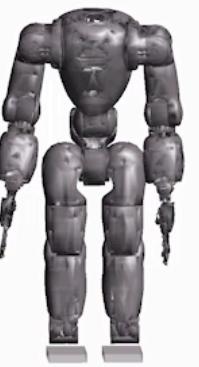
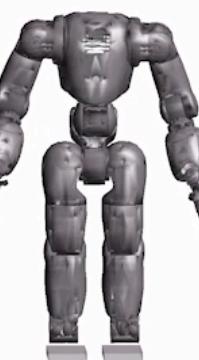
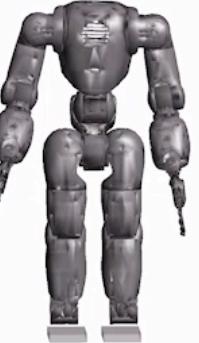
[01] Sum of Signed Distances:[-0.187]



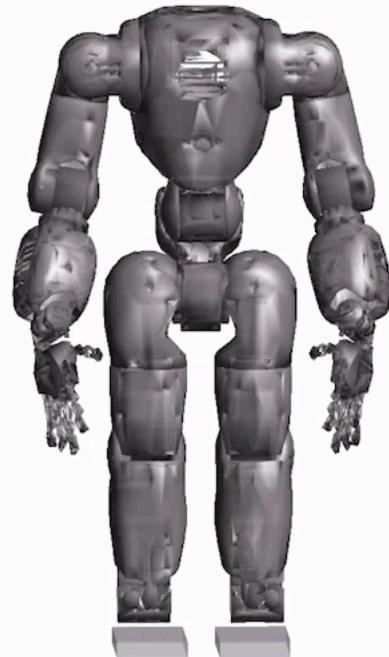
- Additional **trajectory smoothing** using Gaussian process regression as well as **self-collision handling** is processed.



# BIG POINT

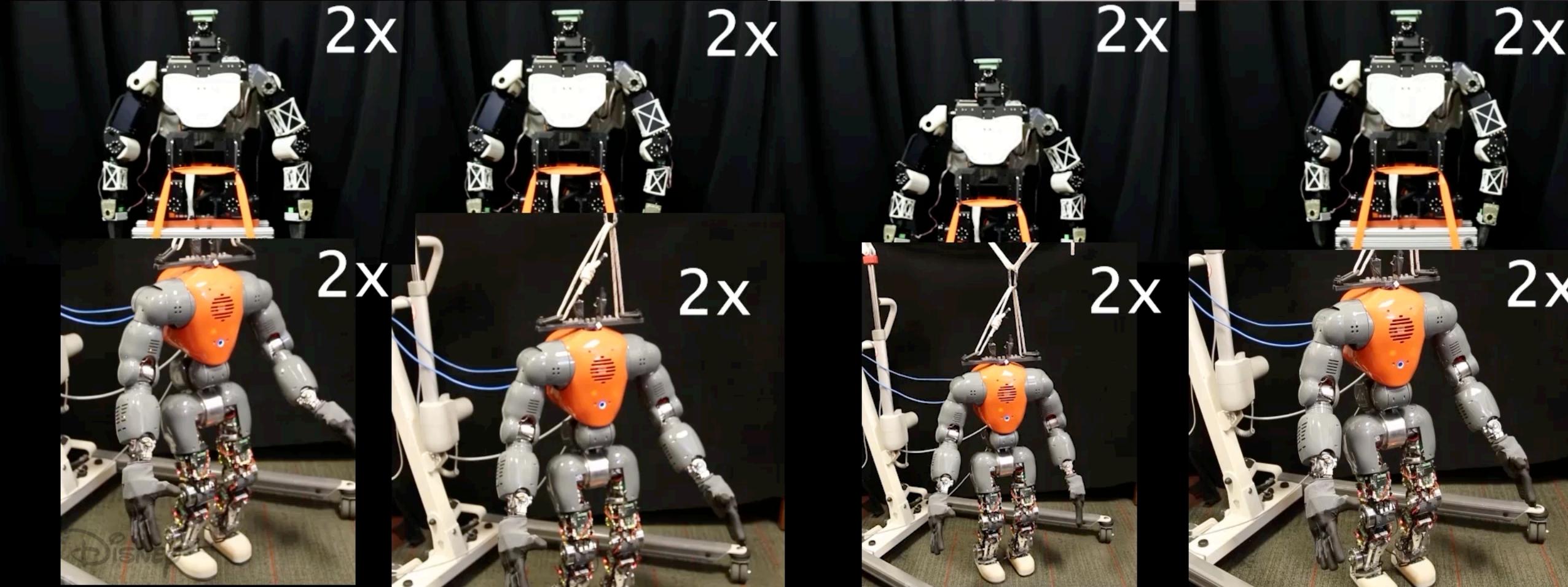


**GO ON**





2x



2x

2x

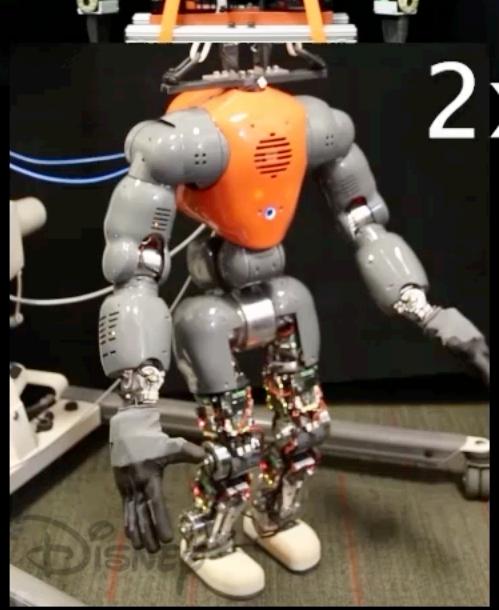
2x

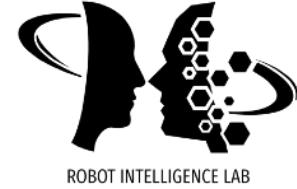
2x

2x

2x

2x

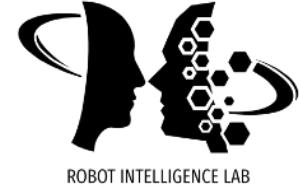




# Cross-Domain Motion Transfer

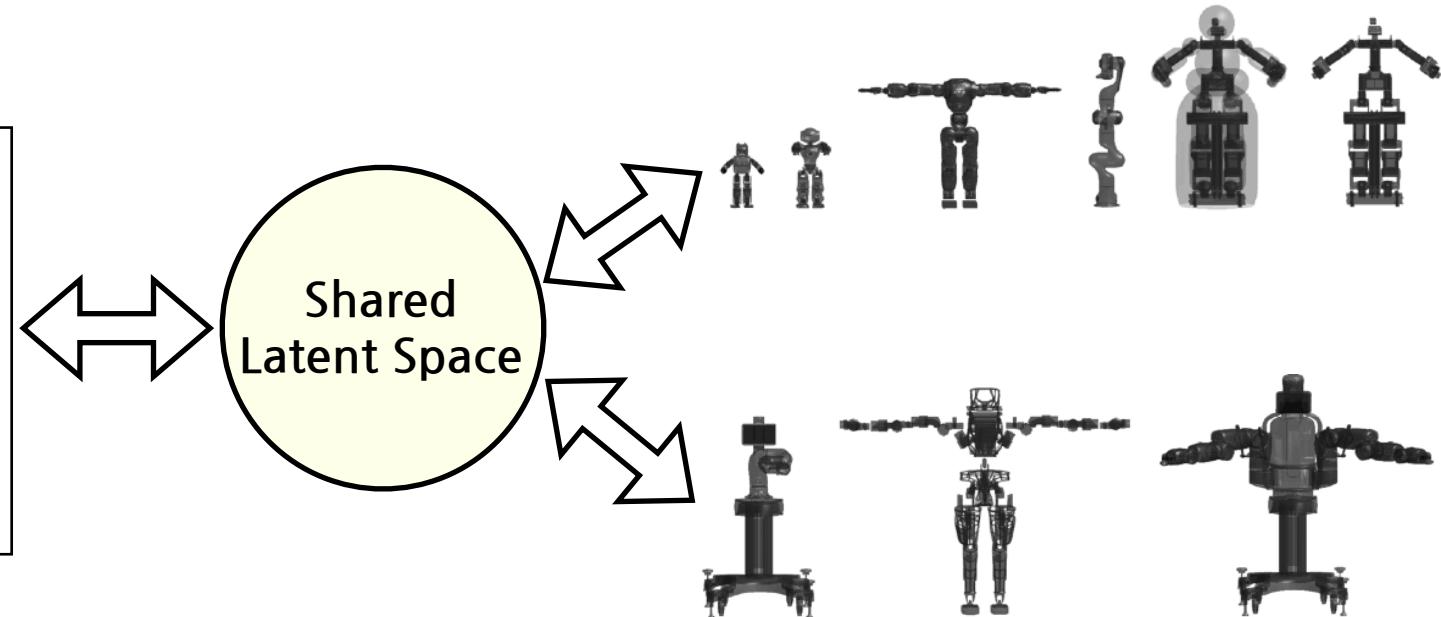
"Cross-Domain Motion Transfer via Safety-Aware Shared Latent Space Modeling", RA-L 2020

# Key Question



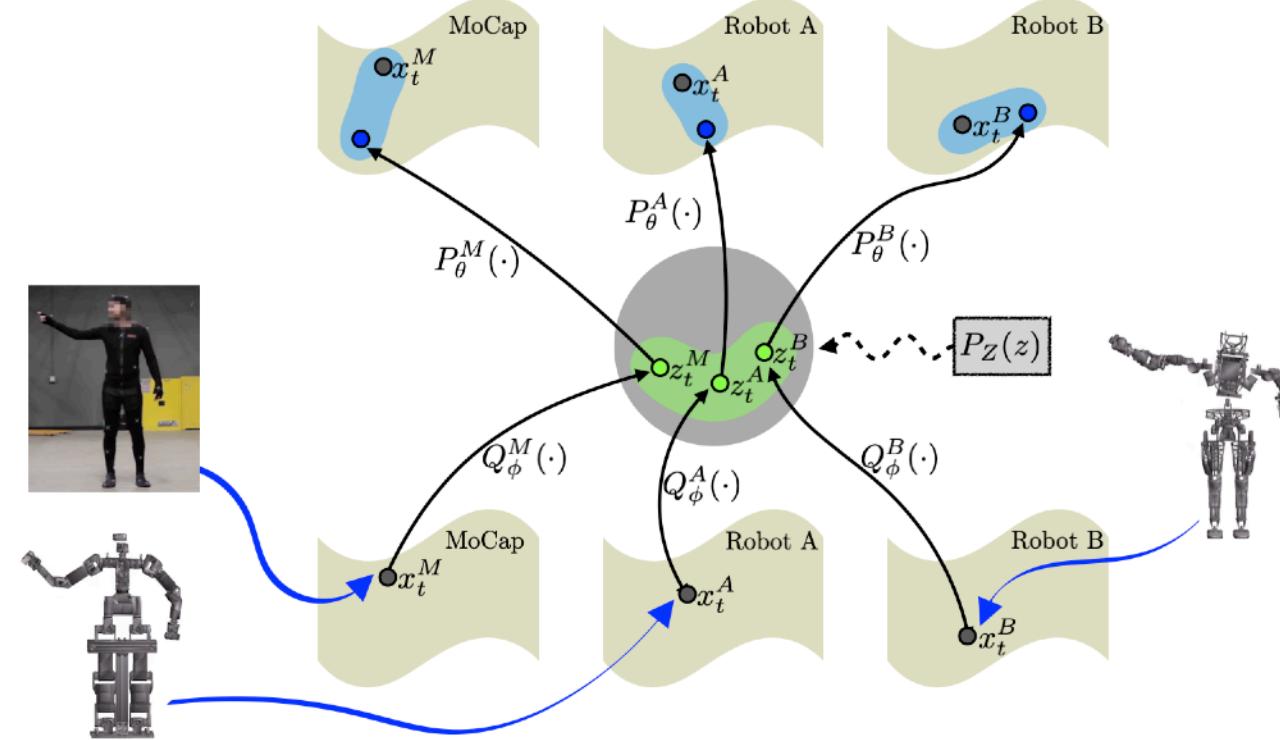
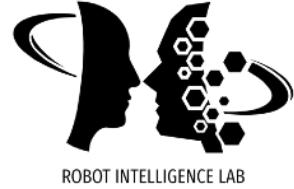
How can we **speed-up** the motion retargeting while considering the **feasibility** of the motion?

# Latent Motion Retargeting



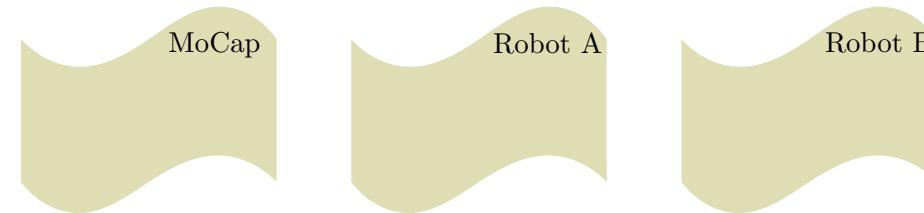
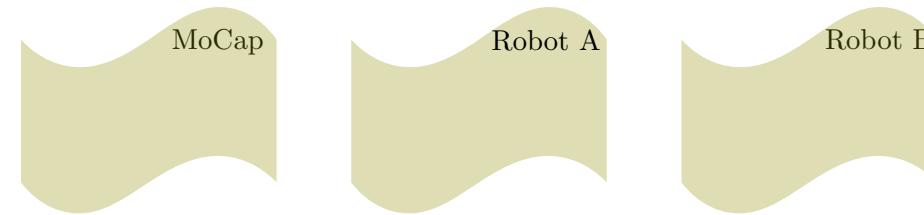
- One way of solving motion retargeting problems is to construct a **shared latent space** between different modalities (different mocap skeletons or different humanoid robot poses).

# Shared Latent Space

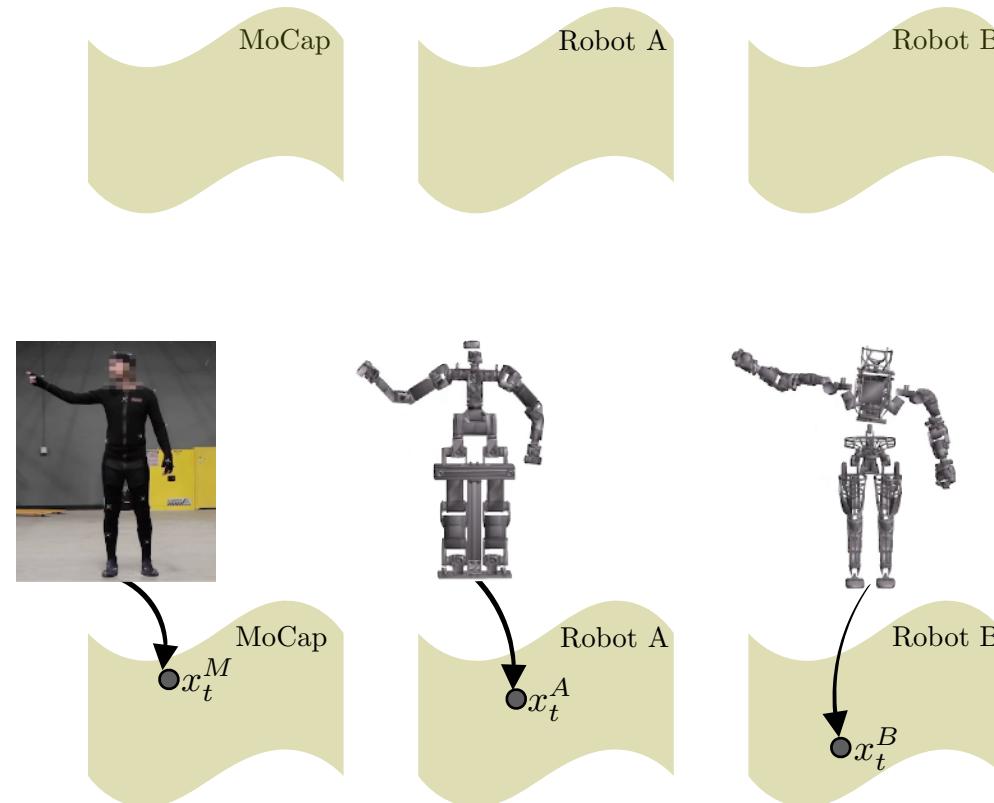


- However, an additional approach is needed to guarantee the feasibility of the transferred results (target humanoid pose).

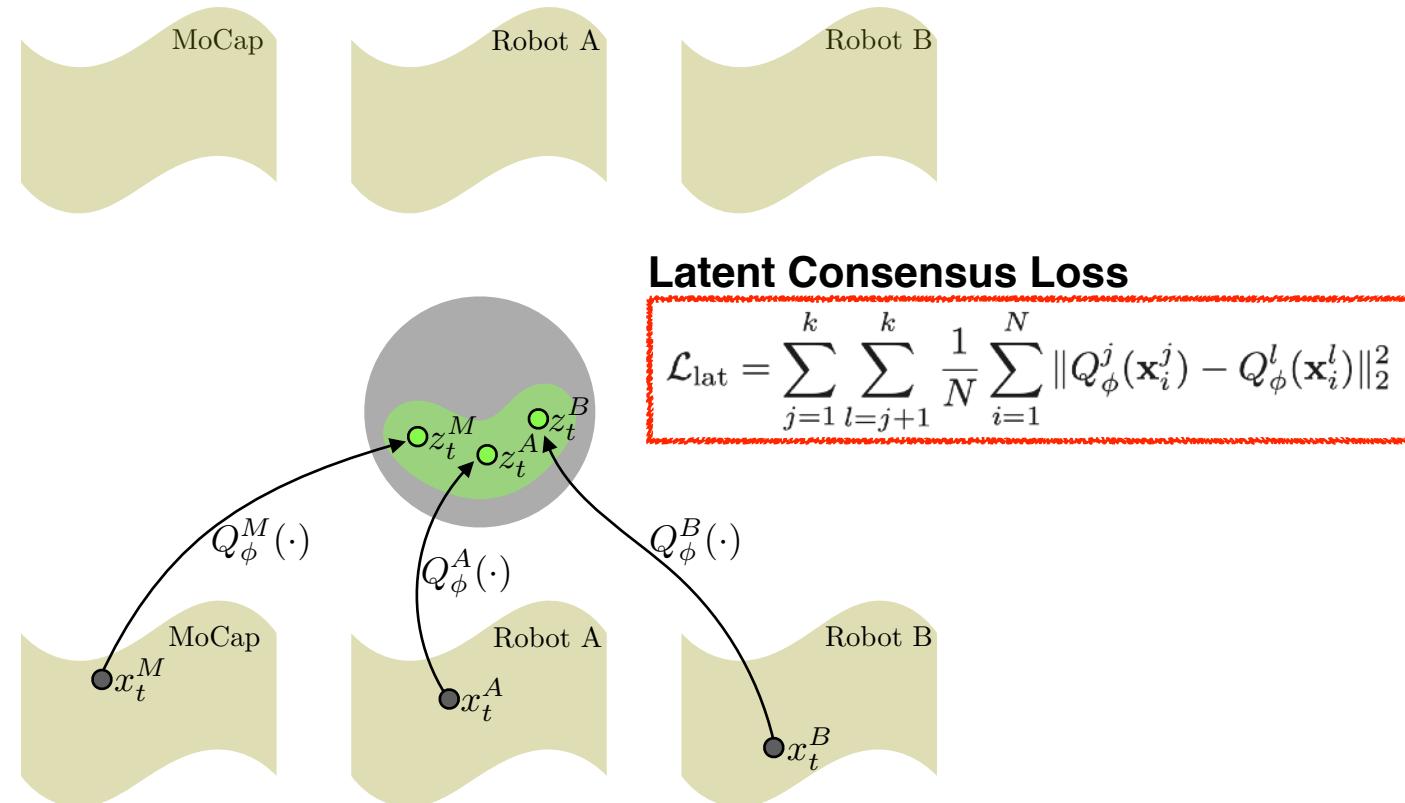
# Learning the Shared Latent Space



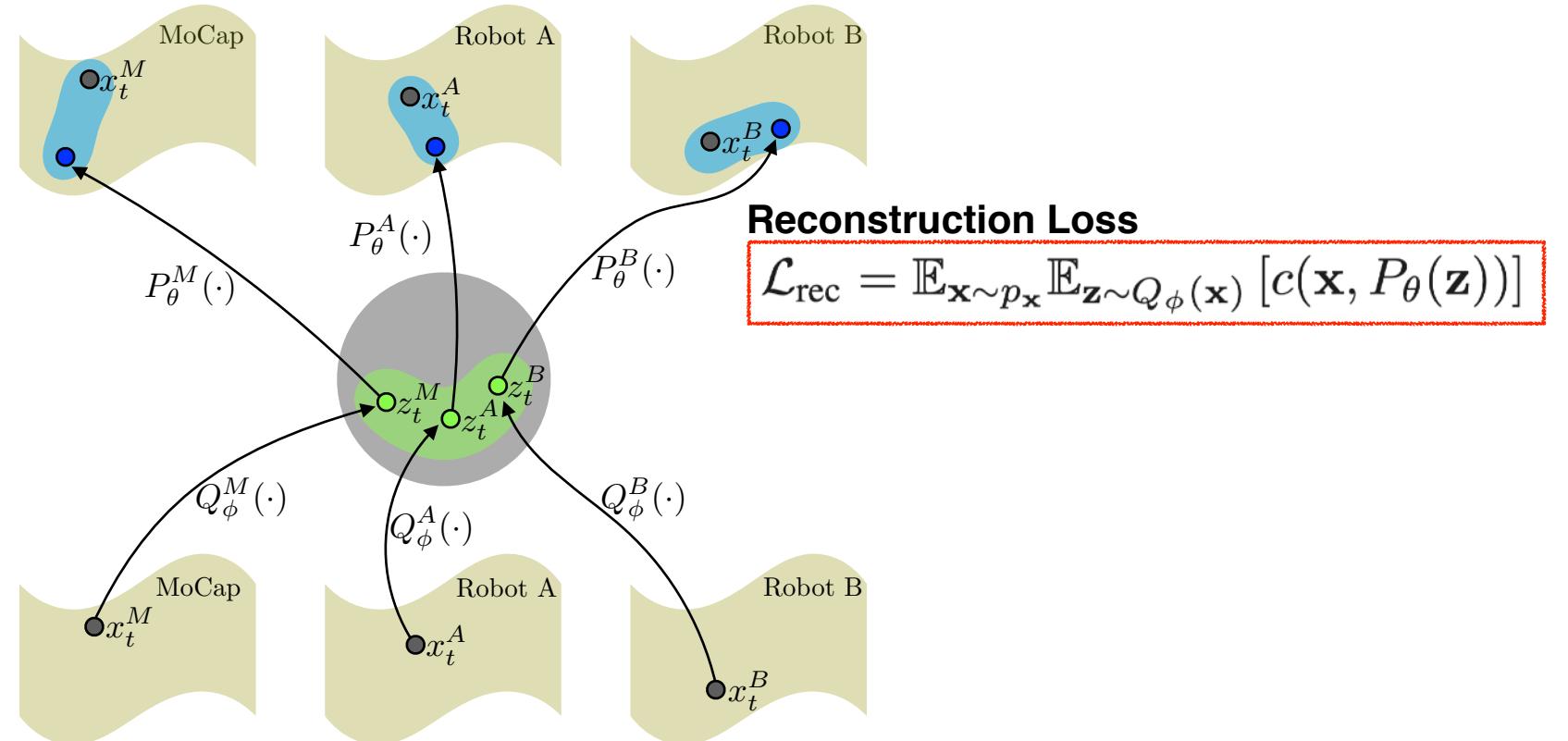
# Learning the Shared Latent Space



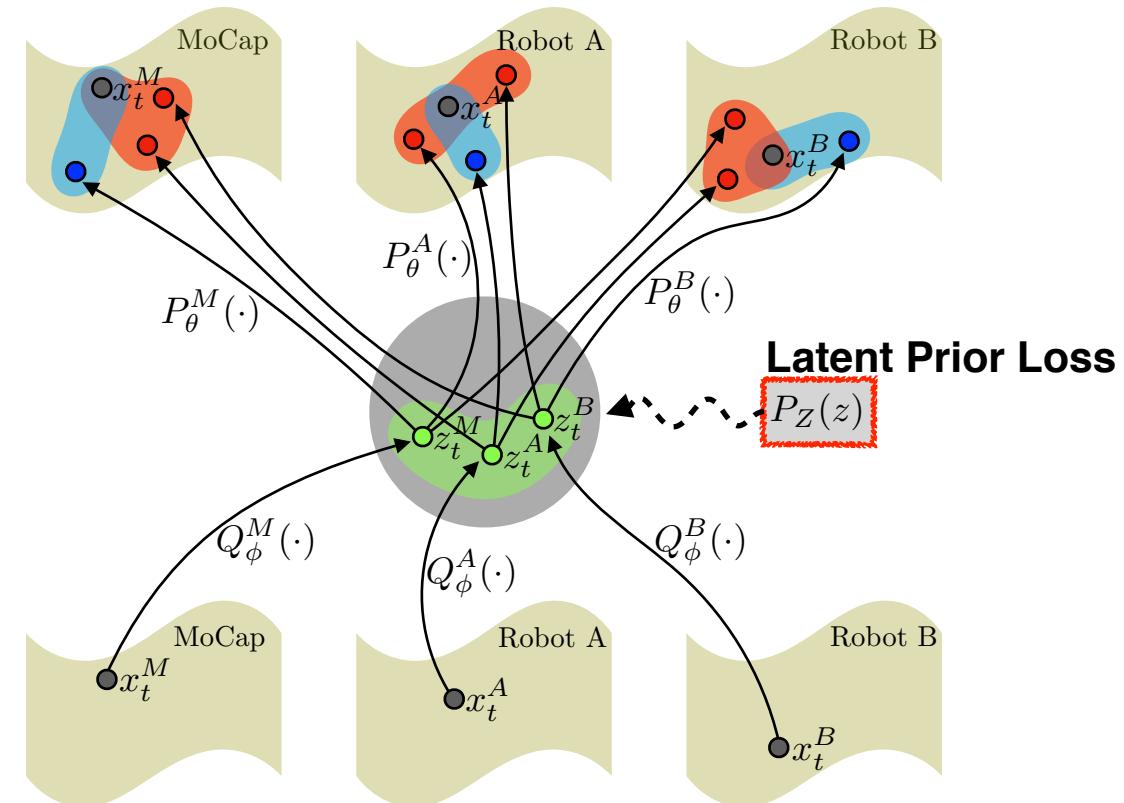
# Learning the Shared Latent Space



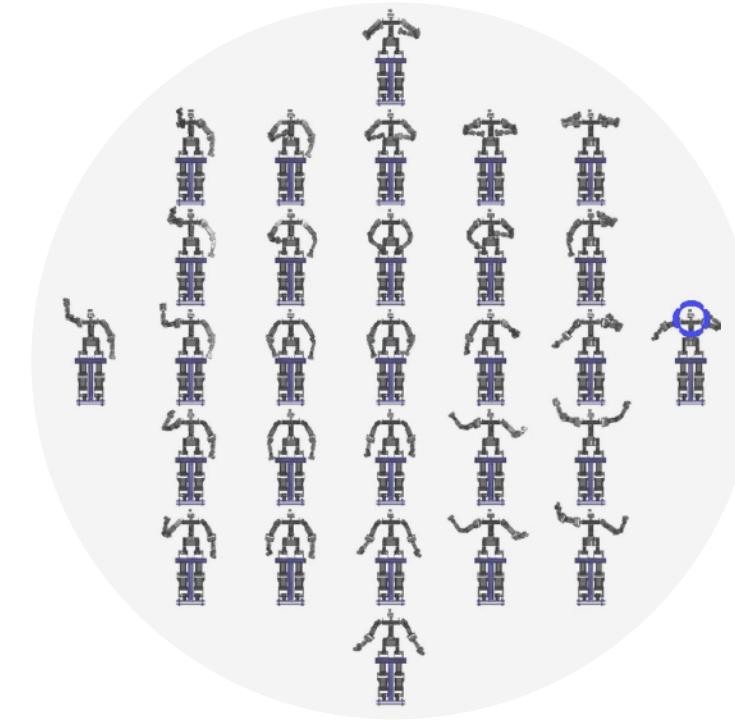
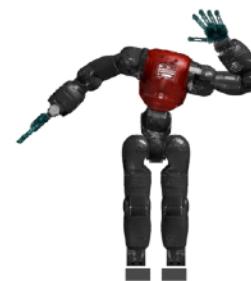
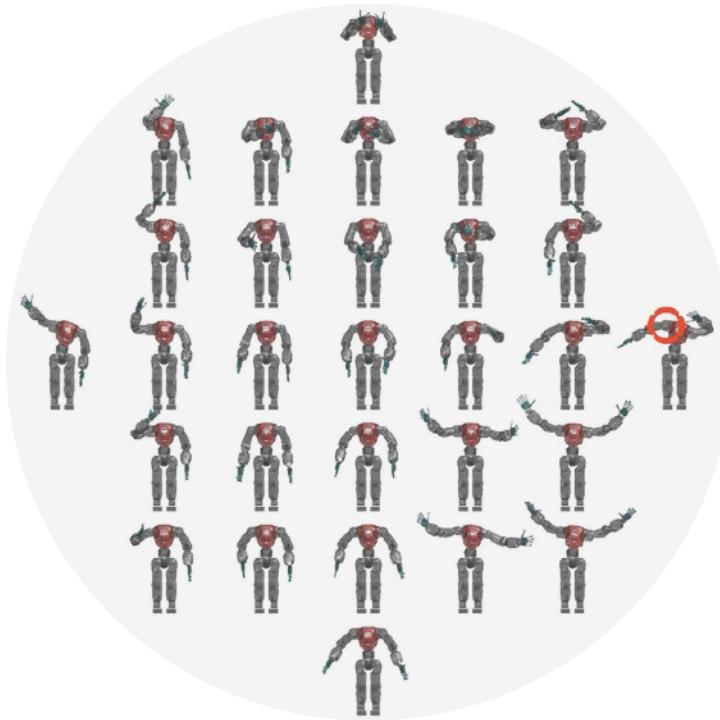
# Learning the Shared Latent Space



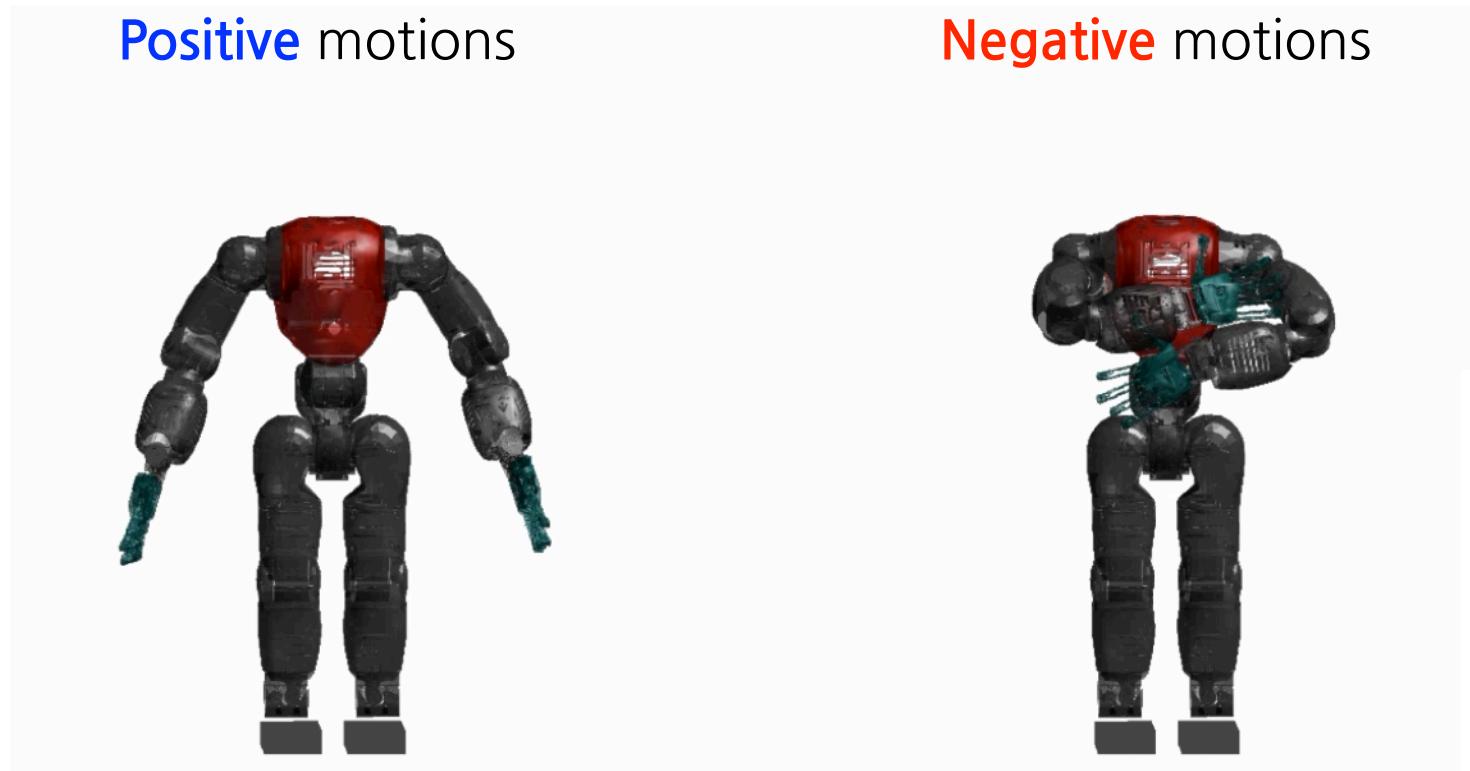
# Learning the Shared Latent Space



# Constructed Shared Latent Space

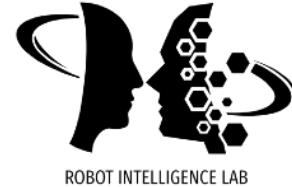


# Safety-aware Learning



- Both **positive** and **negative** motions are leveraged in deep latent space modeling to achieve safety-aware motion retargeting.
- **Negative** data give explicit knowledge of what not to do.

# Results



Baseline



Proposed Method



- By utilizing both positive and negative data, the proposed method greatly reduces the self-collision rates.

# Results



Arm Scratch Motion



Bow Motion



Big Point Motion



Hand High Motion

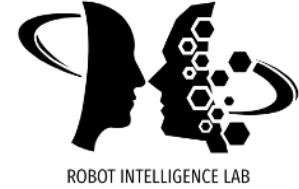




# Nonparametric Motion Retargeting

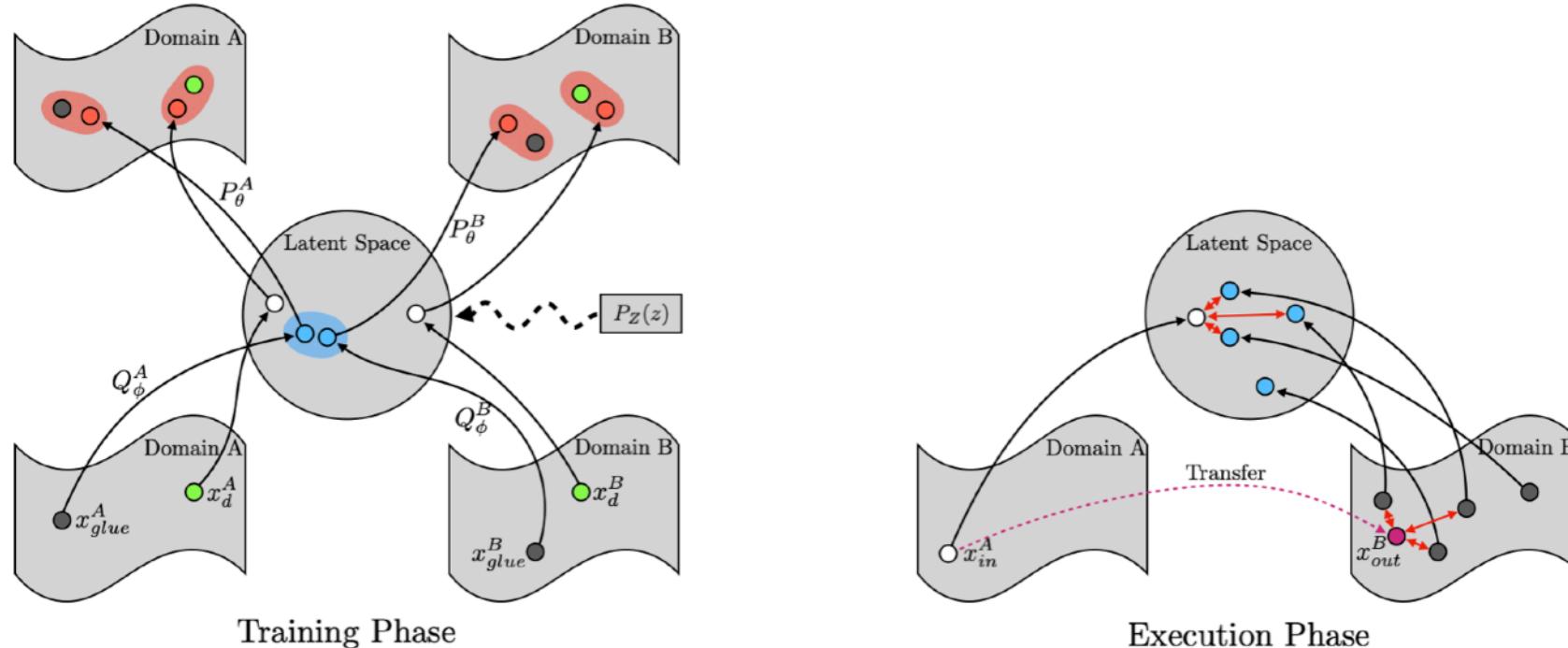
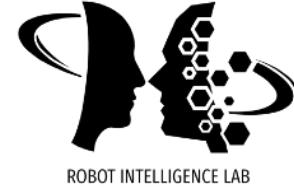
"Nonparametric Motion Retargeting for Humanoid Robots on Shared Latent Space", RSS 2020

# Key Question



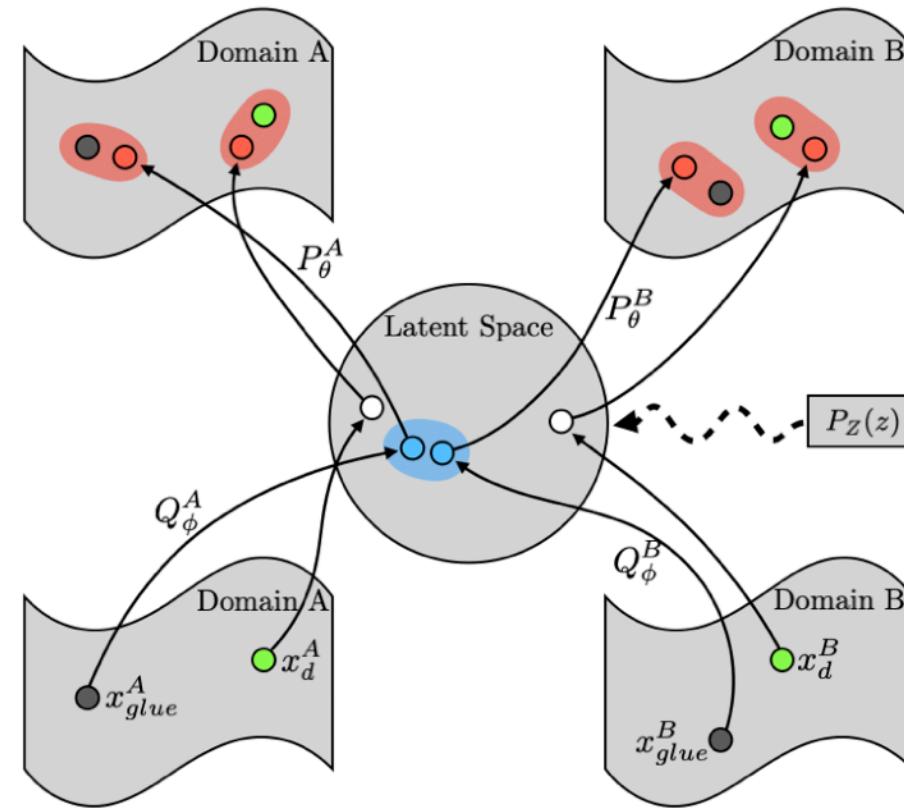
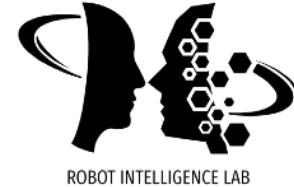
How can we always **guarantee** the feasibility of the retargeted motion?

# Nonparametric Motion Retargeting



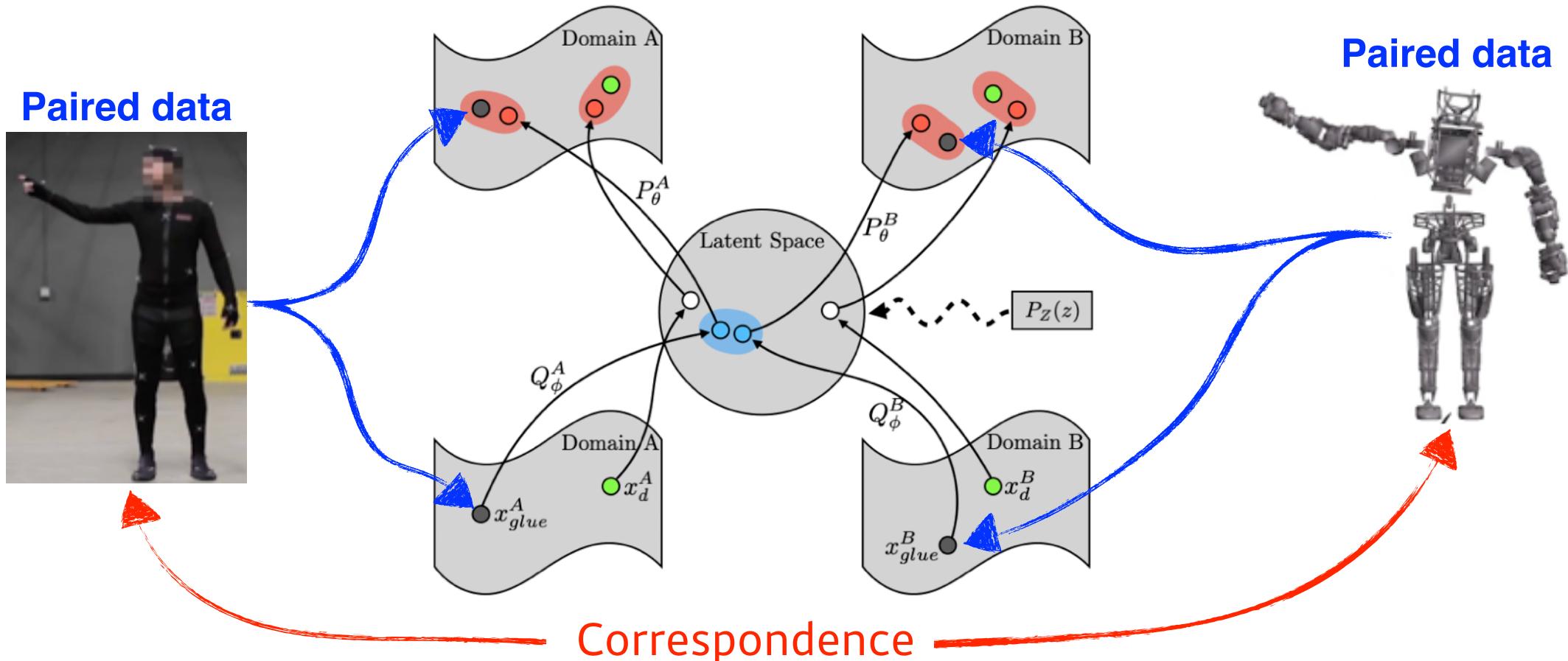
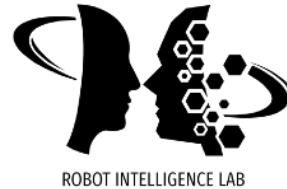
- In general, it is extremely difficult to **guarantee** the feasibility of the output of deep neural networks.
- In this work, we combine a nonparametric regression method with shared latent space modeling to present **nonparametric motion retargeting**.

# Training Phase

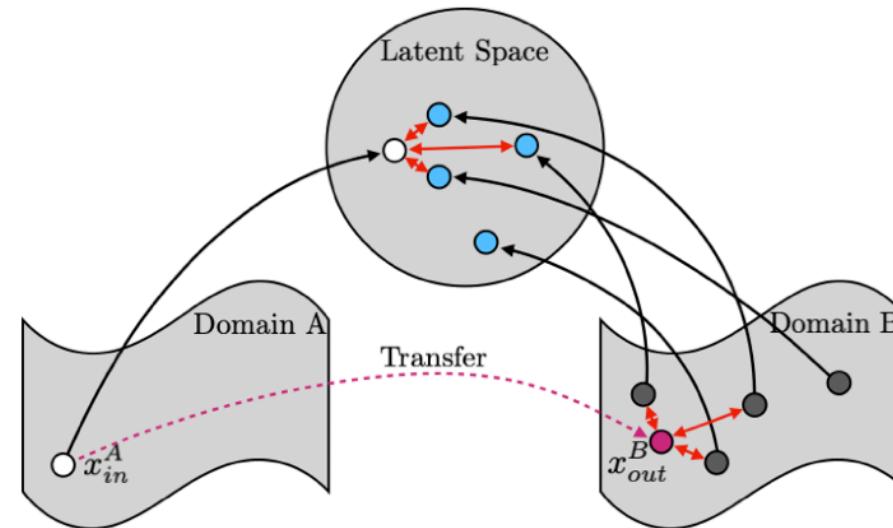


- Training phase is similar to our previous method.

# Training Phase

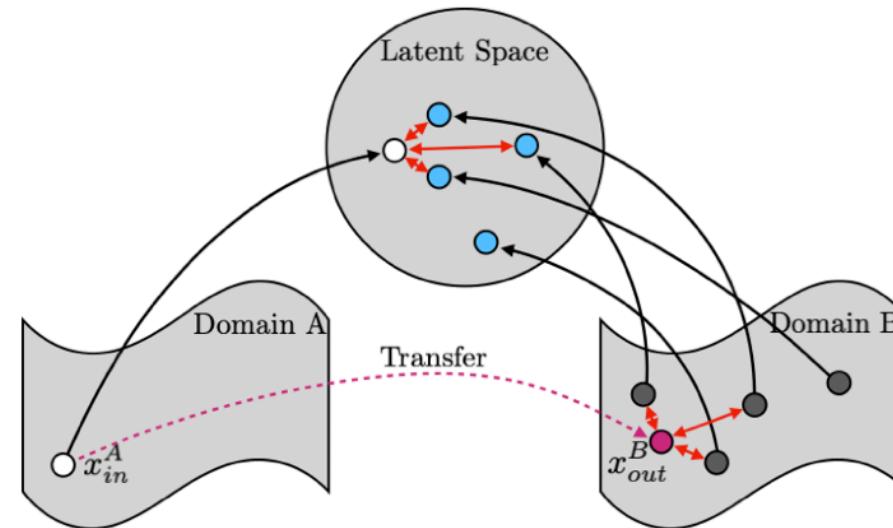


# Execution Phase



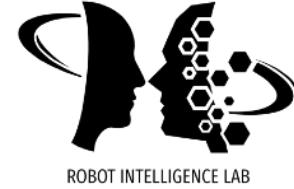
- On the **Execution phase**, we utilized nonparametric regression (or the nearest neighbor algorithm) on the shared latent space.

# Execution Phase

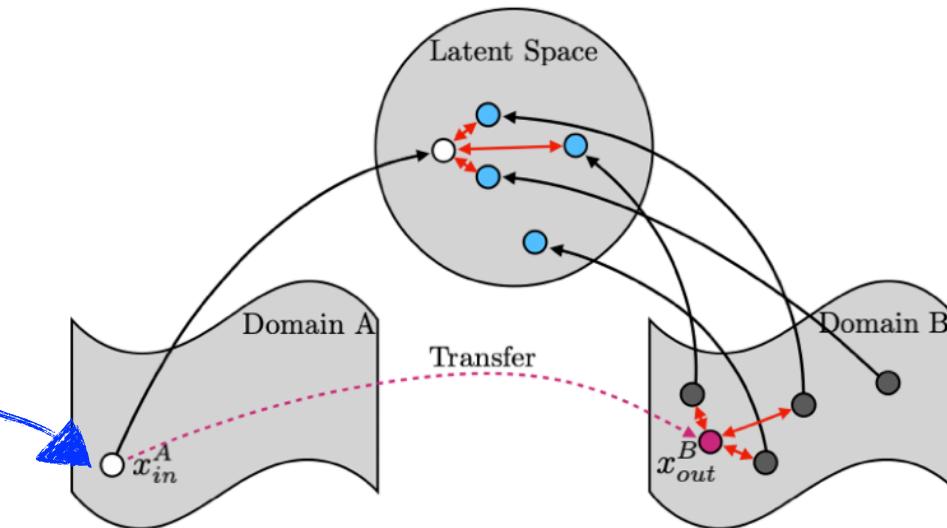


- On the **Execution phase**, we utilized nonparametric regression (or the nearest neighbor algorithm) on the shared latent space.

# Execution Phase

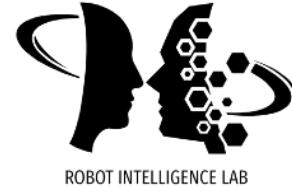


Given input

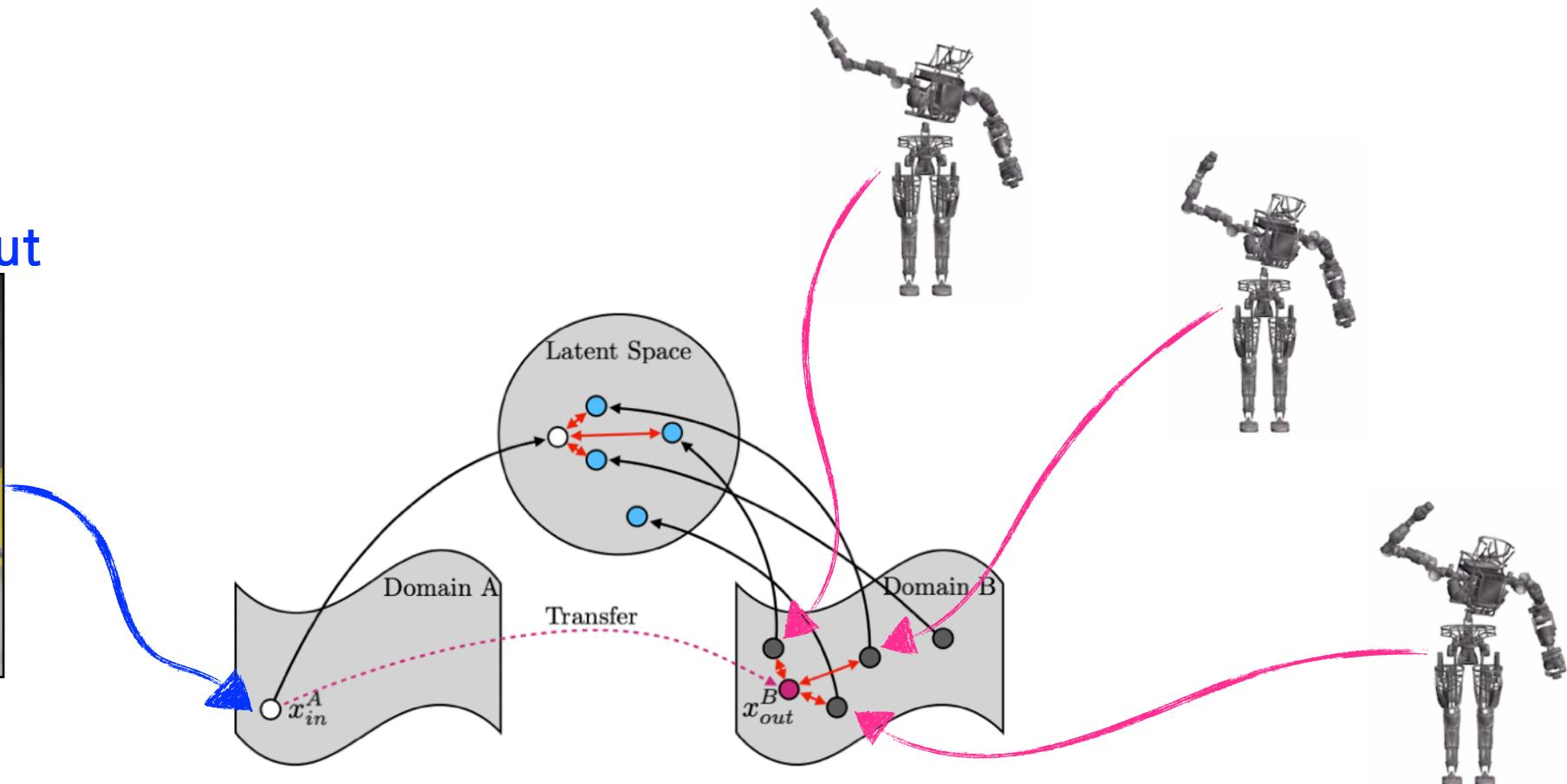


- Once a mocap pose is given as an input.

# Execution Phase



Given input



- The corresponding robot pose is computed (or selected) by **locally weighted regression** on the shared latent space.

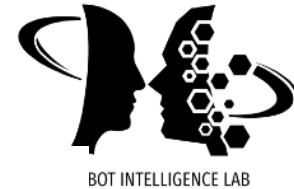
# Latent Nonparametric Method



- What is the benefit of leveraging **shared latent space** with **nonparametric methods**?

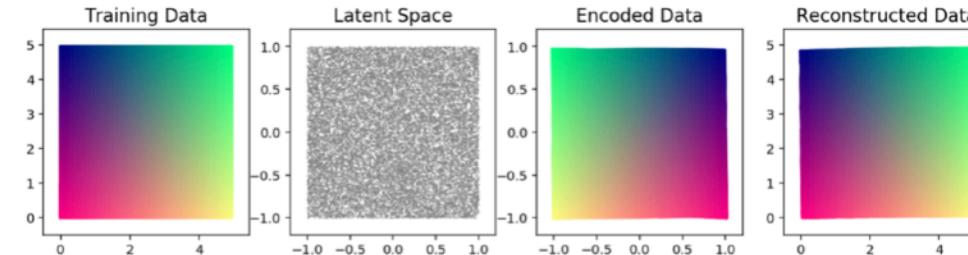
	Feasibility Guarantee	Paired Data Requirements	Inference Time
Nonparametric Method (e.g., Lookup Table)	YES	YES	$O(N)$
Function approximators (e.g., Multi-layer Perceptron)	NO	YES	$O(1)$
Shared Latent Space (e.g., Cross-Domain Motion Transfer)	NO	NO	$O(1)$
<b>Proposed Method</b> (e.g., Latent Nonparametric Method)	YES	NO	$O(N)$

# Efficient Sub-Sampling

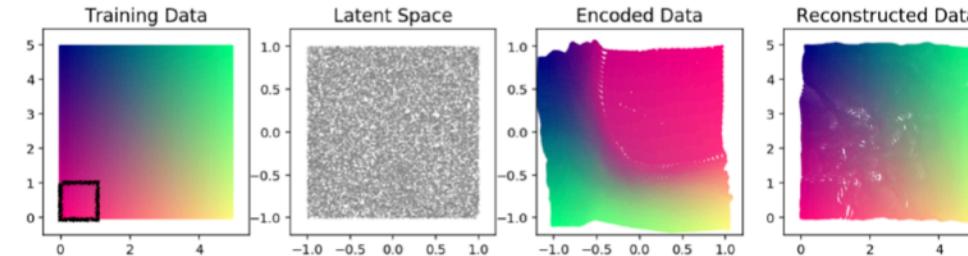


$$\begin{aligned}
 & \text{Original DPP} \\
 & \mathbf{x}_* = \arg \max_{\mathbf{x} \in \mathcal{X}_{\setminus m}} (1 - \mathbf{k}^T K^{-1} \mathbf{k}) \det(K) \\
 & \approx \arg \max_{\mathbf{x} \in \mathcal{X}_{\setminus m}} (1 - \mathbf{k}^T \mathbf{k}) \\
 & = \arg \min_{\mathbf{x} \in \mathcal{X}_{\setminus m}} \|\mathbf{k}\|_2 \\
 & = \arg \min_{\mathbf{x} \in \mathcal{X}_{\setminus m}} \|\mathbf{k}\|_1 \\
 & = \arg \min_{\mathbf{x} \in \mathcal{X}_{\setminus m}} \sum_{i=1}^m k(\mathbf{x}, \mathbf{x}_i).
 \end{aligned}$$

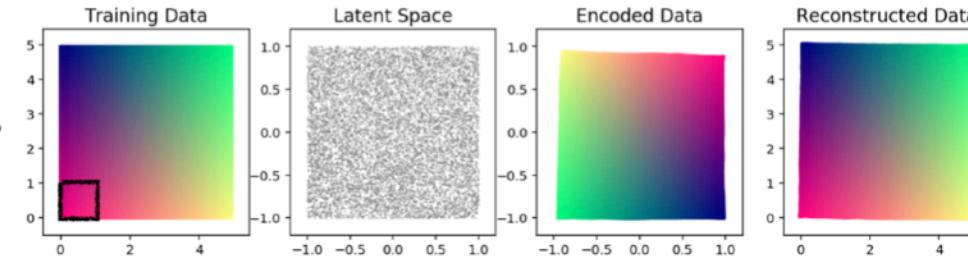
Bal.+Unif.



Imb.+Unif.

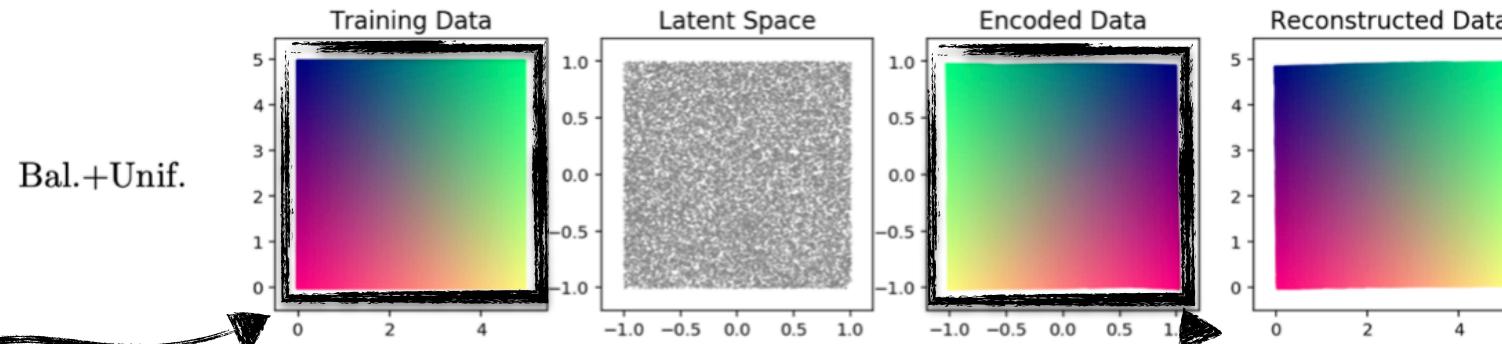


Imb.+LA-DPP



- An efficient subset sampling method is presented to handle the possible imbalance in the training dataset while constructing the latent space.

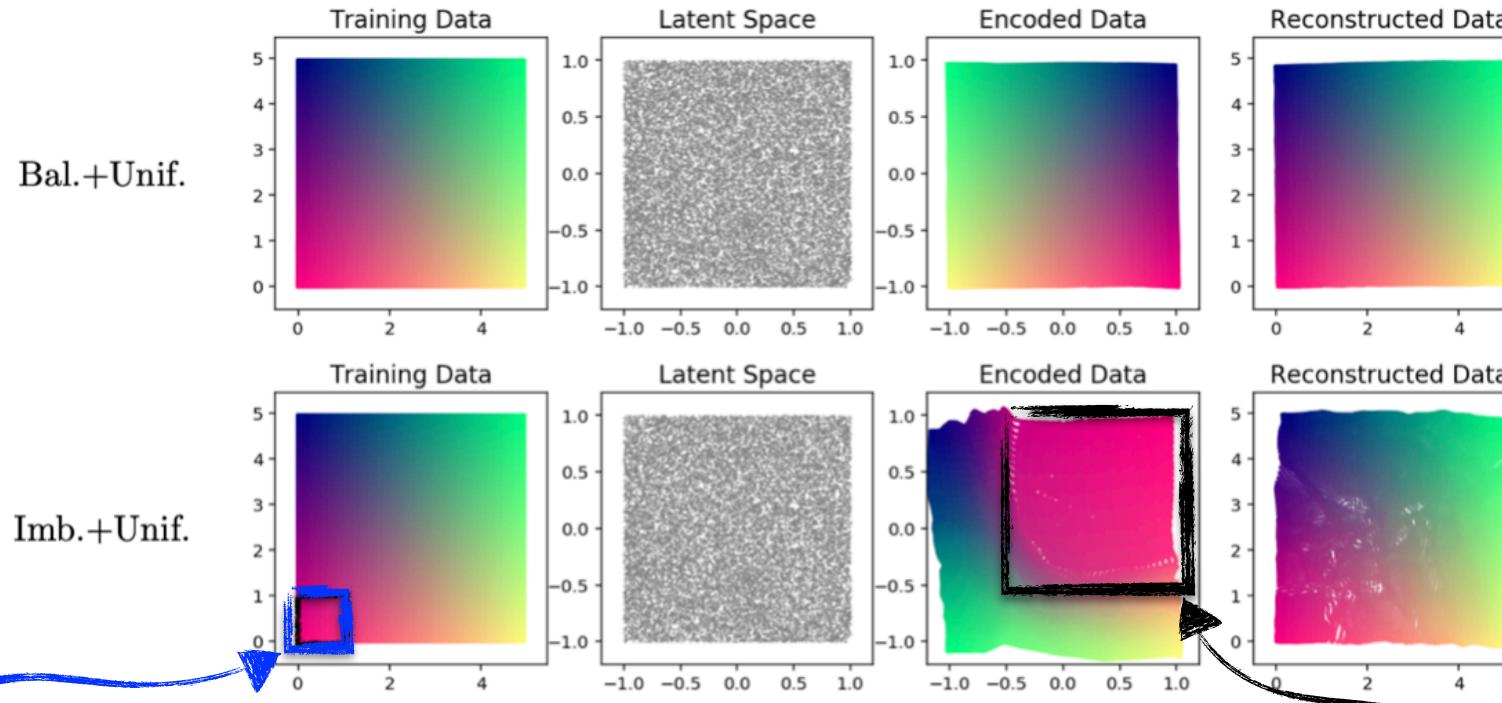
# Efficient Sub-Sampling



If the training data are uniformly distributed within the domain  
(i.e., a **balanced dataset**)

Then, the resulting latent space is also **uniformly covered** with the training data.

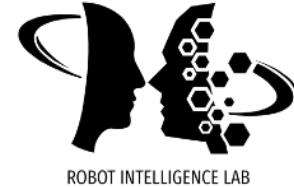
# Efficient Sub-Sampling



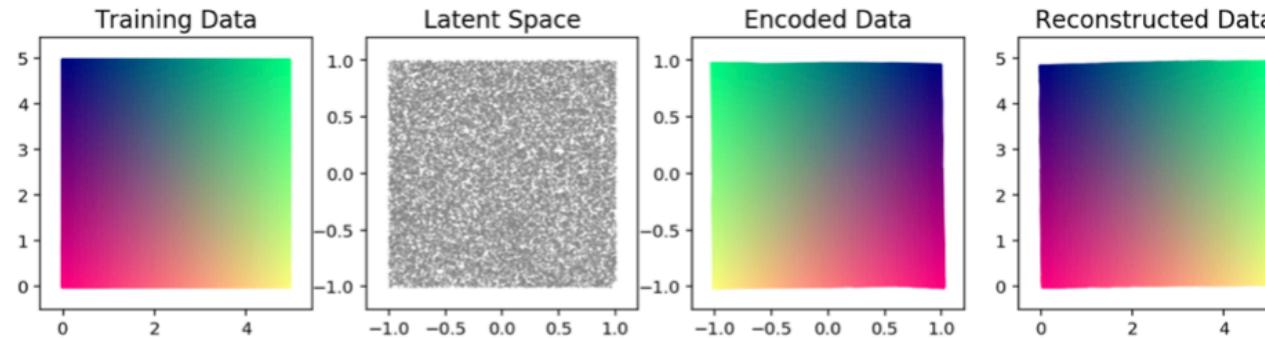
If the half of training data are located within the **small blue box** (i.e., an imbalanced dataset)

Then, about the **50%** of the latent space is covered with the data from the **small imbalanced region** in the dataset.

# Efficient Sub-Sampling

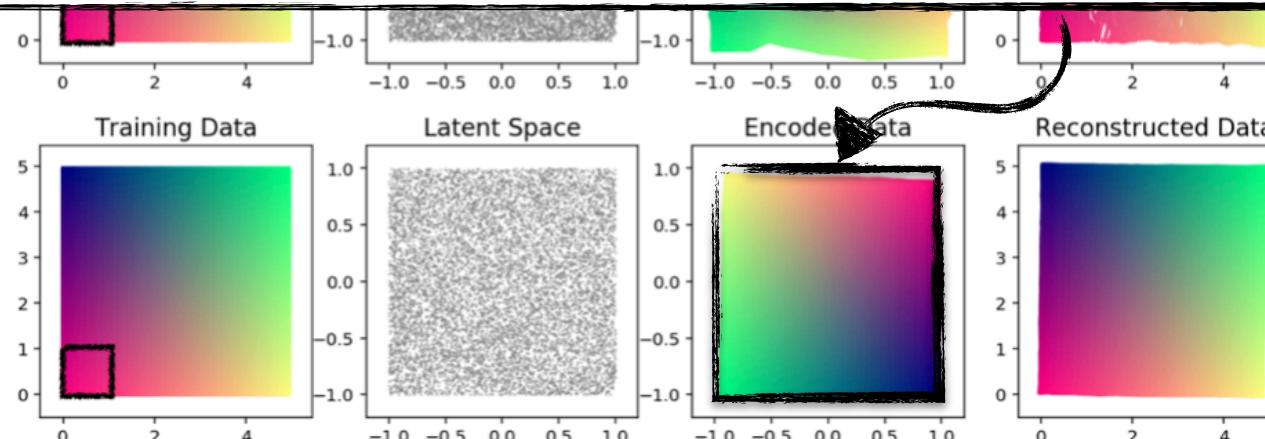


Bal.+Unif.



When the mini-batches are sampled with the **proposed subset sampling method**, the latent space is uniformly covered with the training data even with the **imbalanced dataset**.

Imb.+LA-DPP





# Results

## ArmCross



MoCap



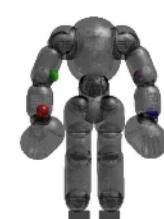
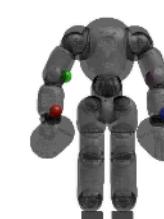
Proposed Method



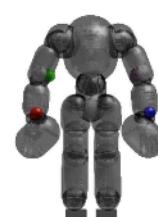
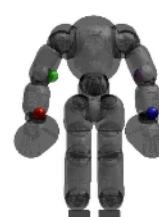
Baseline

Red color indicates  
self-collision is occurred.

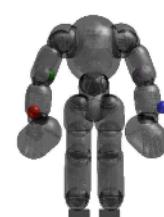
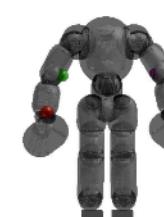
## BigPoint



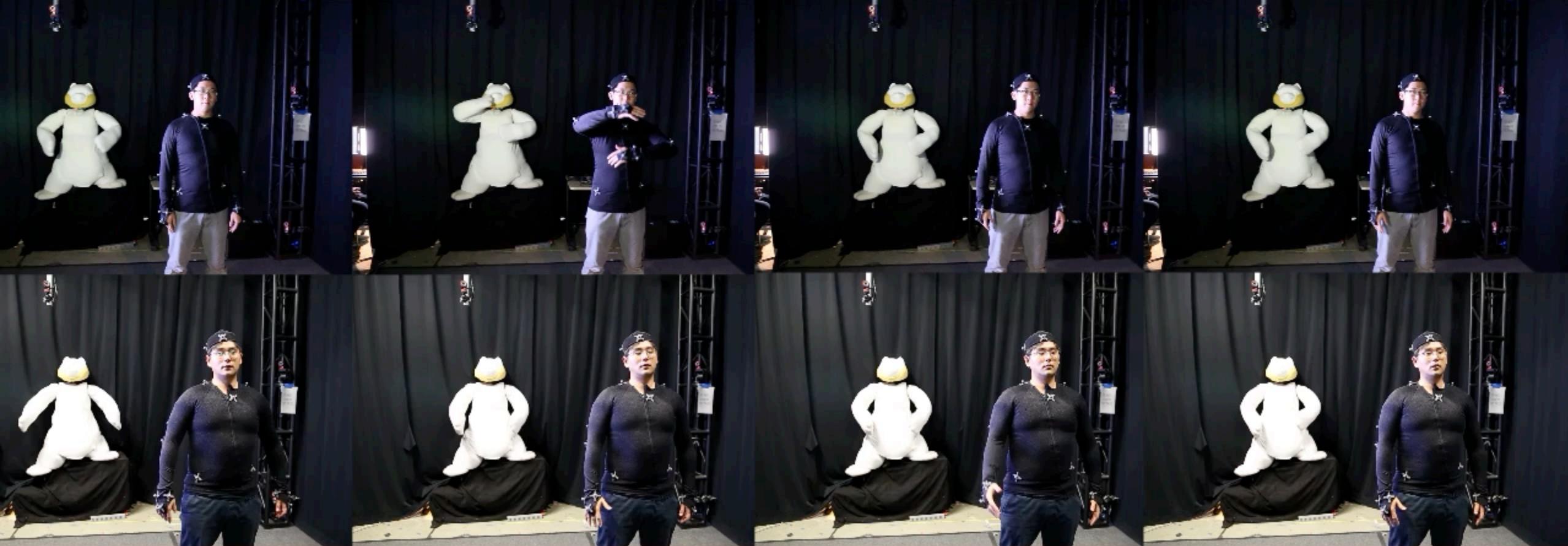
## BigWave



## BowBeg



# Results



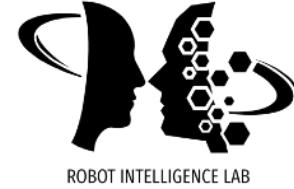
- There is no additional self-collision handling routine besides our proposed nonparametric motion retargeting method.



# Self-Supervised Motion Retargeting

"Self-Supervised Motion Retargeting with Safety Guarantee", ICRA 2021

# Key Questions

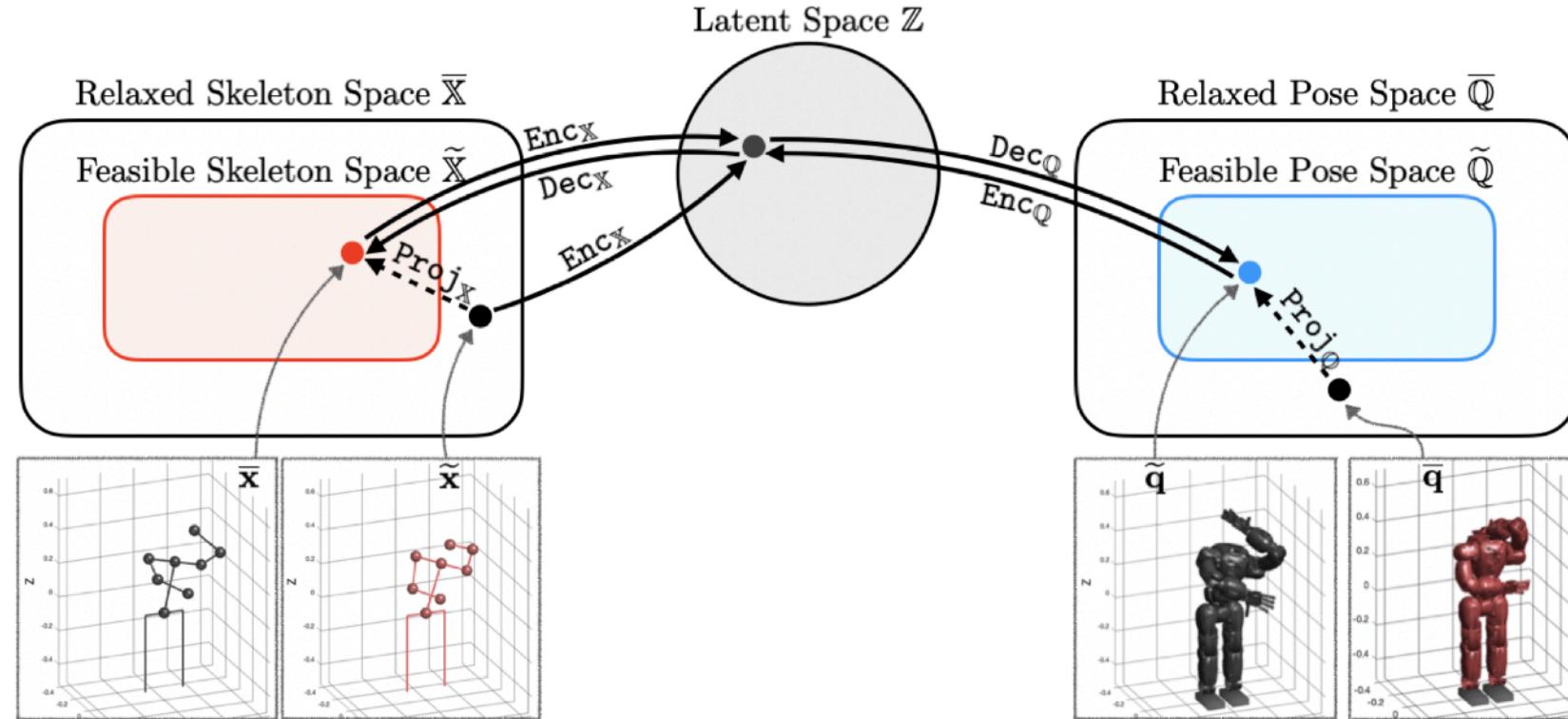
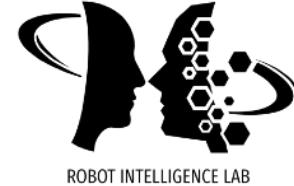


How can we consider **different expressivities** of human skeletons and humanoid robots.

+

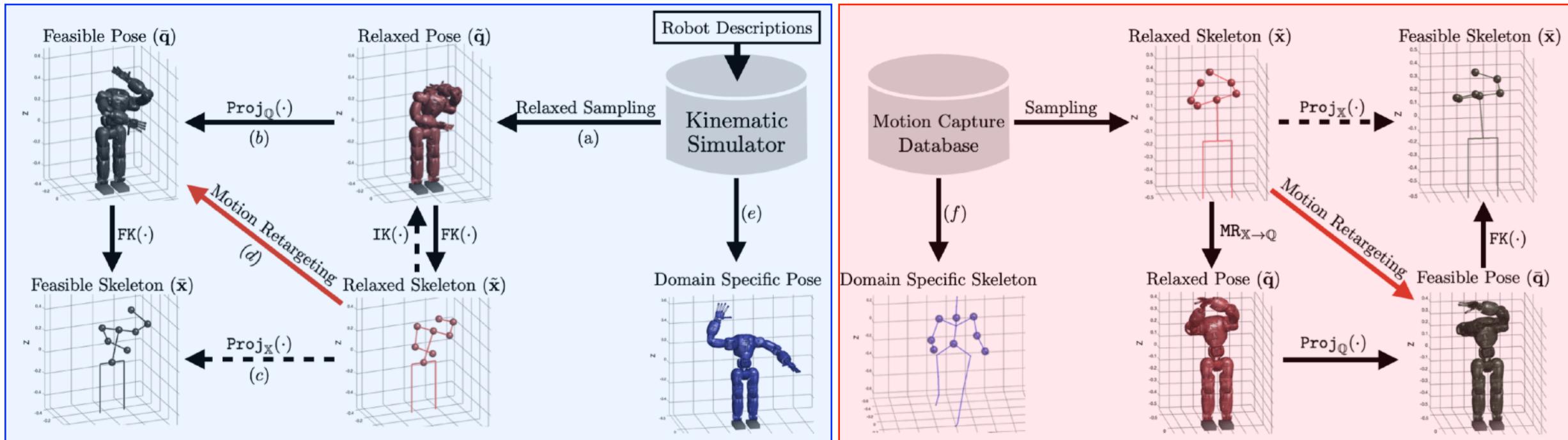
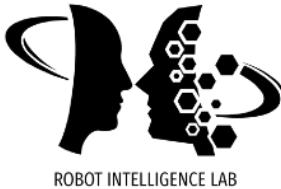
Can we train a learning-based motion retargeting method in a **self-supervised manner**?

# Projection-invariant Mapping



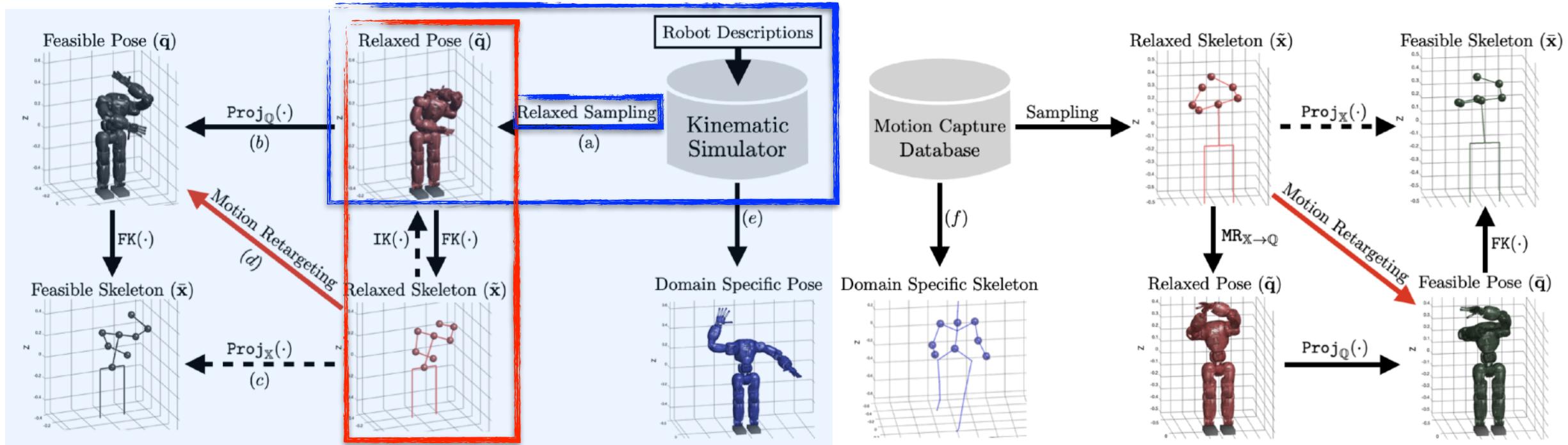
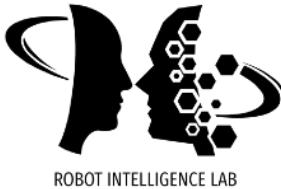
- Due to the limited expressivity of a humanoid robot, **multiple robot poses** may correspond to a **single human pose**. To this end, we propose a self-supervised motion retargeting by constructing a shared latent space between different domains.

# Automated Sampling Pipelines



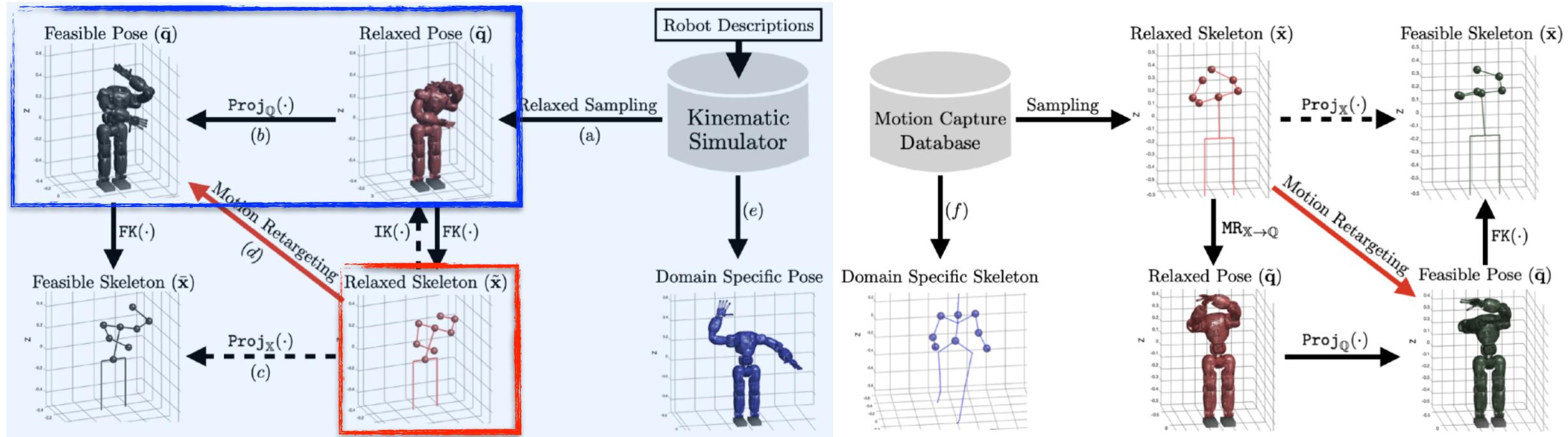
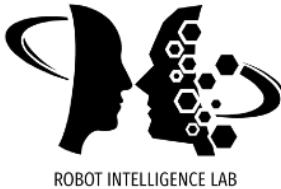
- There are two paths of automated training data sampling pipelines: 1) **from the robot descriptions** and 2) **from the motion capture data**.

# Automated Sampling Pipelines

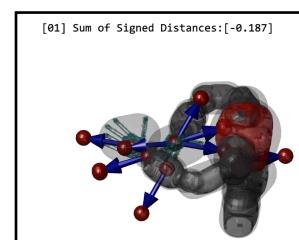


- From the robot descriptions (i.e. URDF), (a) we randomly **sample** a relaxed robot pose by sampling joint positions from the **relaxed joint limits** and get the corresponding robot pose by solving **forward kinematics**.

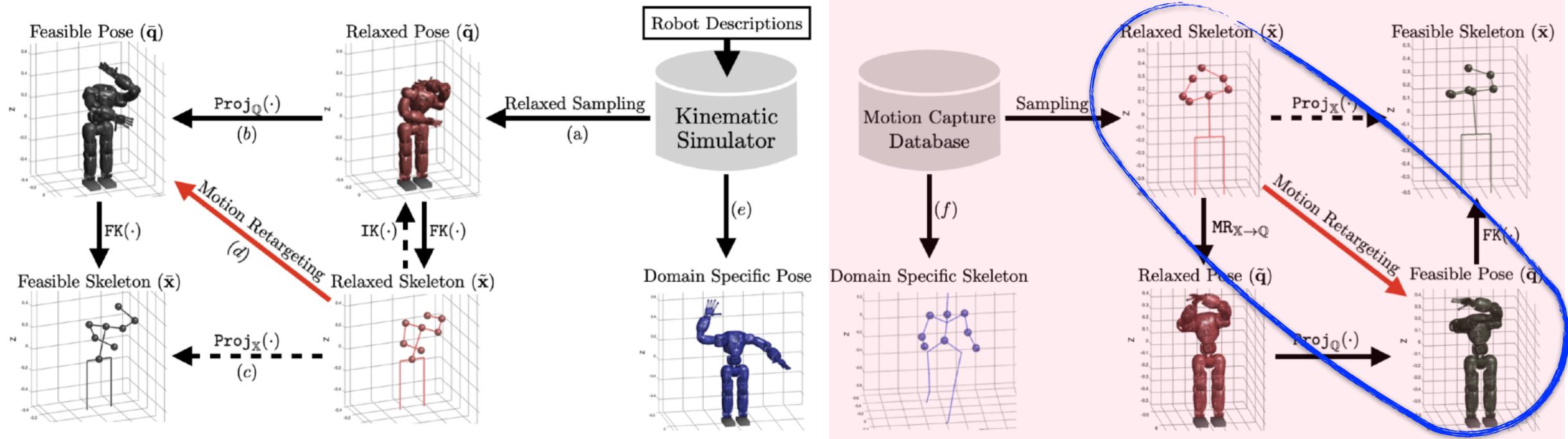
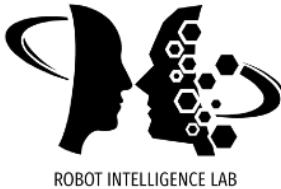
# Automated Sampling Pipelines



- Then, we find the corresponding **feasible pose** from the relaxed pose using the projection mapping (i.e., handling joint limit violations and self-collisions).



# Automated Sampling Pipelines

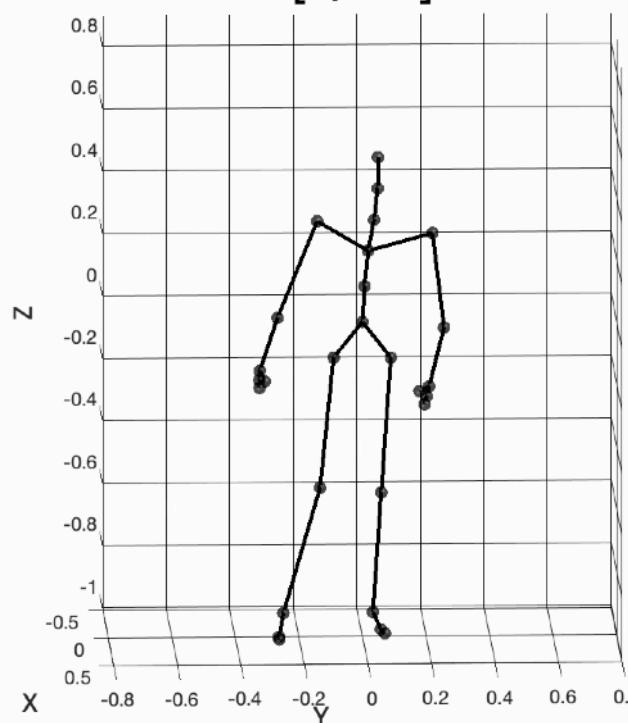


- Similarly, we utilize the **optimization-based motion retargeting** method with the projection mapping the collect training data.

# Results

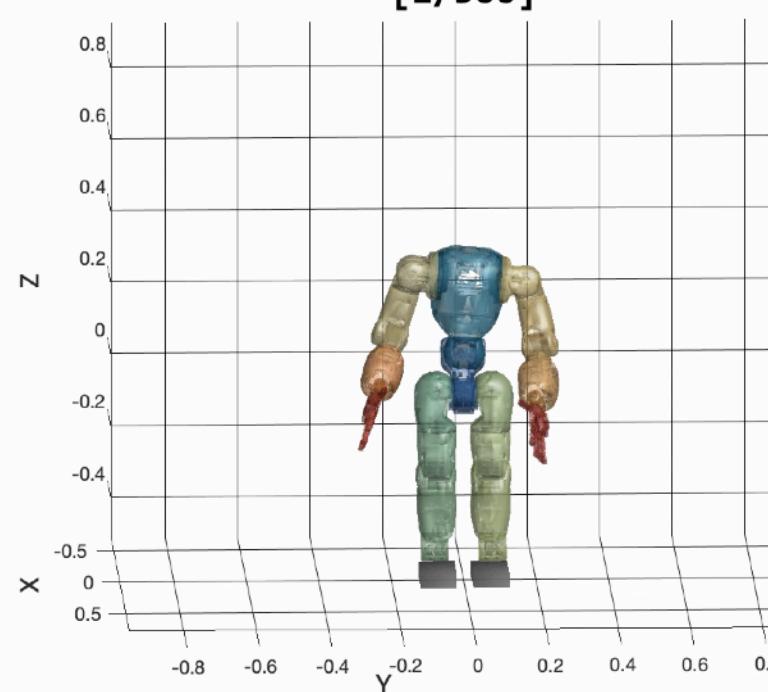
Source Motion

[2/500]



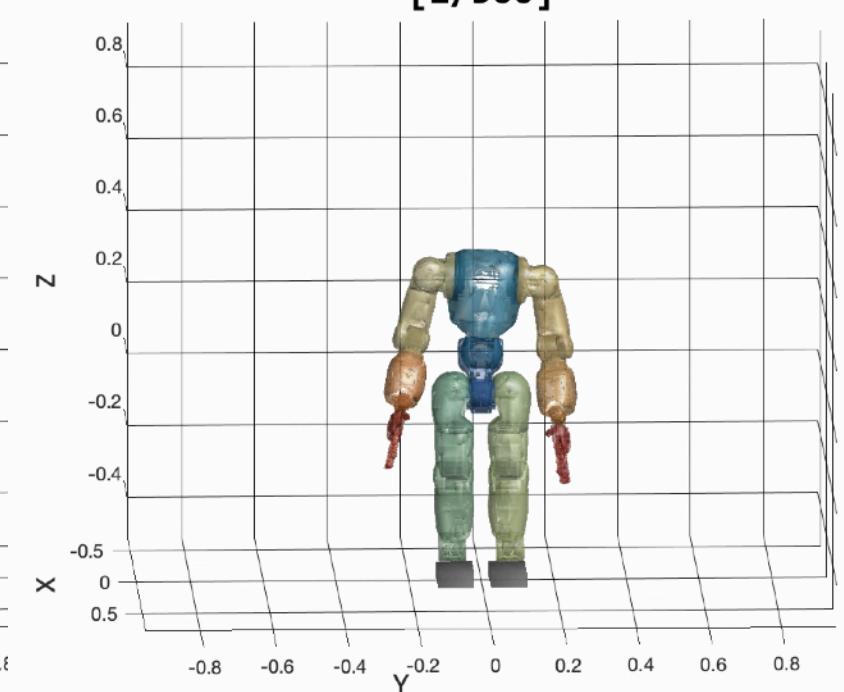
Proposed Method

[2/500]



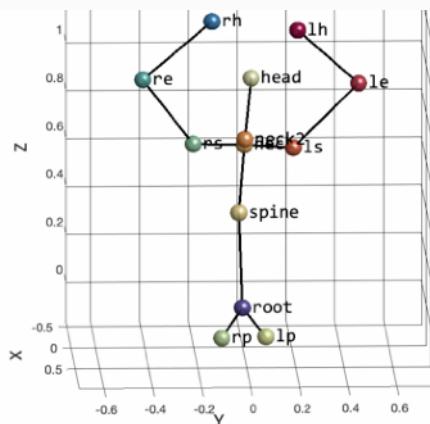
Base Line

[2/500]

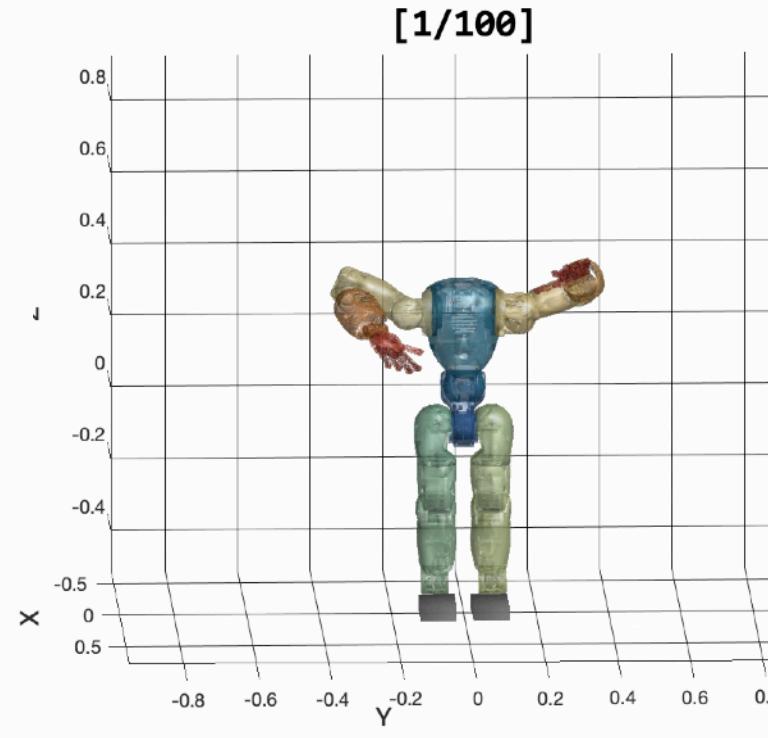


# Results

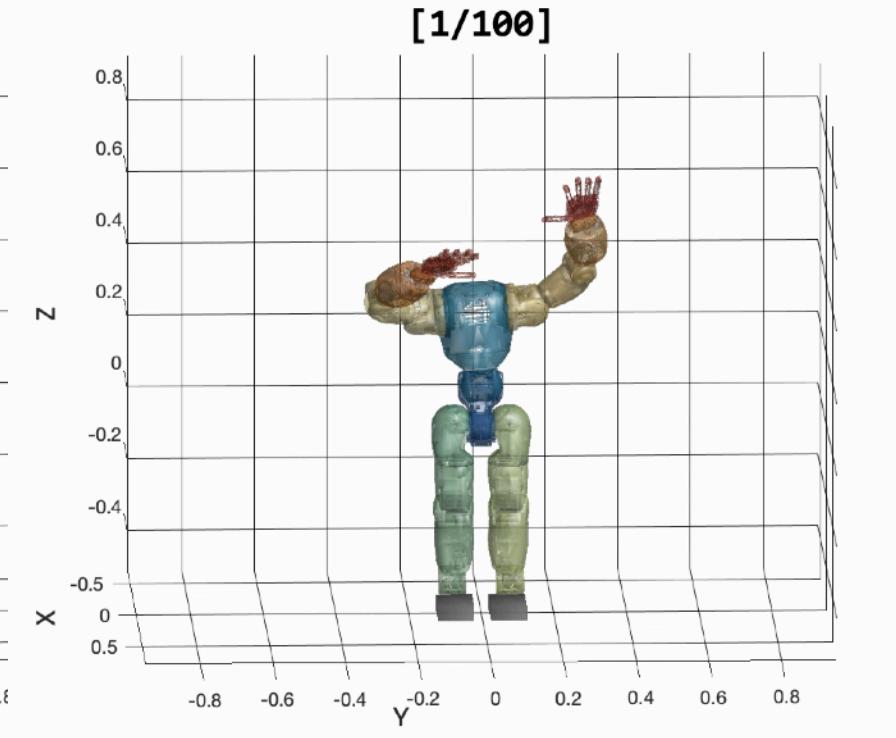
Source Motion



Proposed Method



Base Line



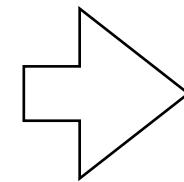
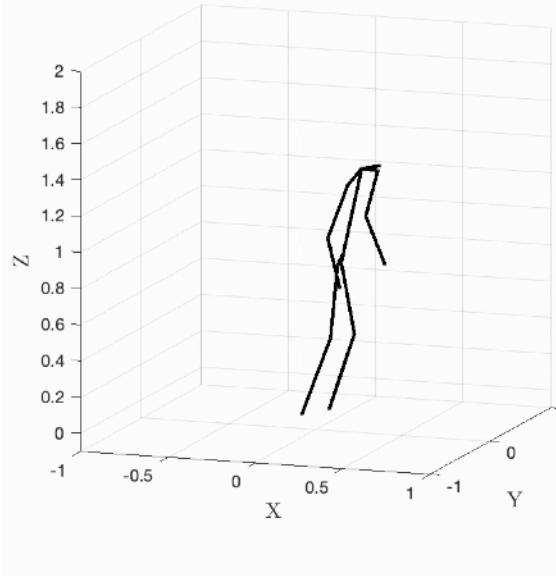


# Robust Motion Retargeting

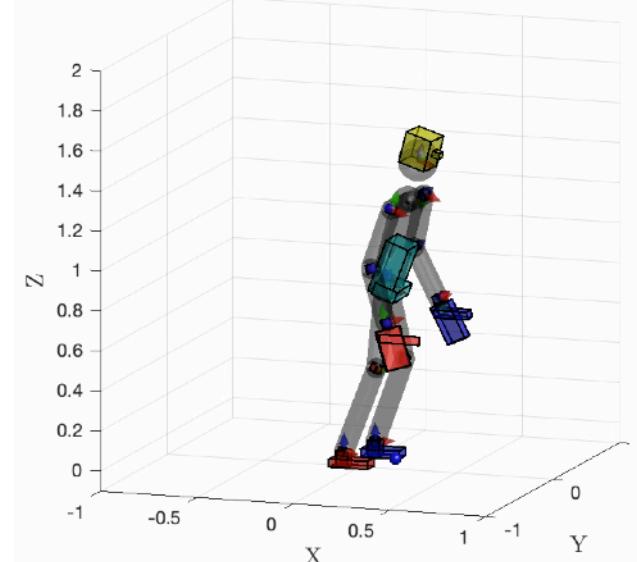
# MoCap Handling



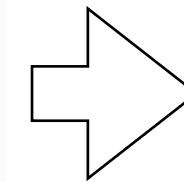
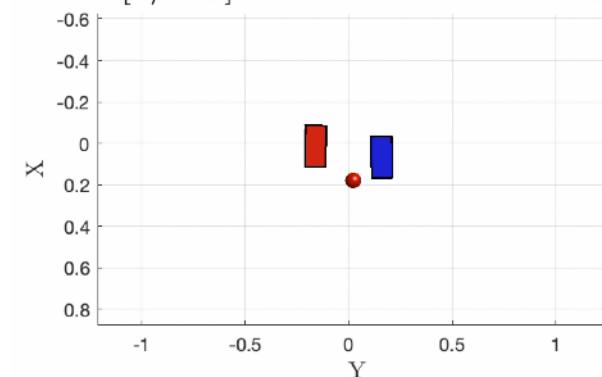
[1/500] MoCap Skeleton



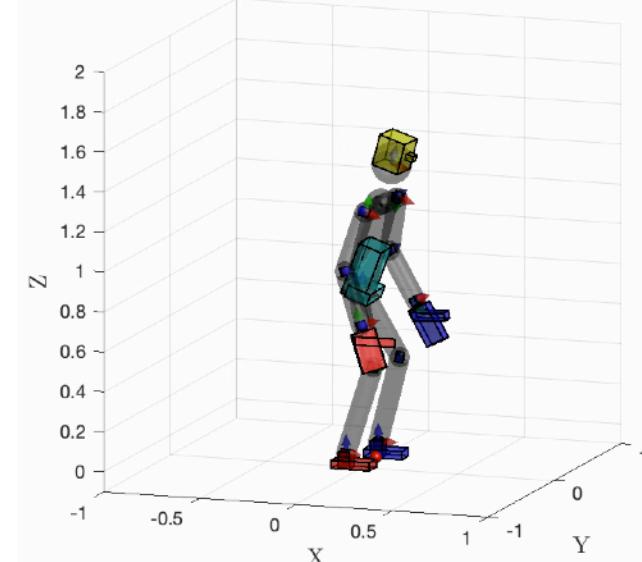
[1/500] Pre-Humanoidization



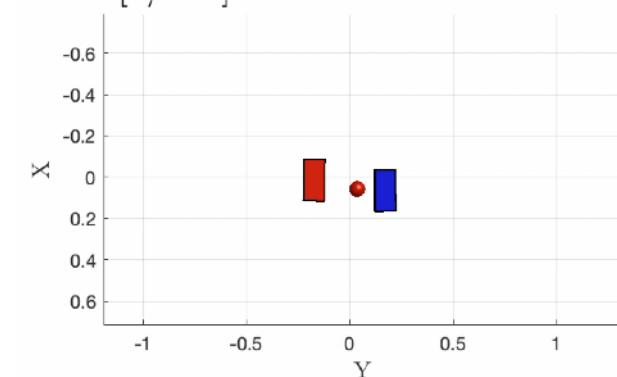
[1/500] Pre-Humanoidization



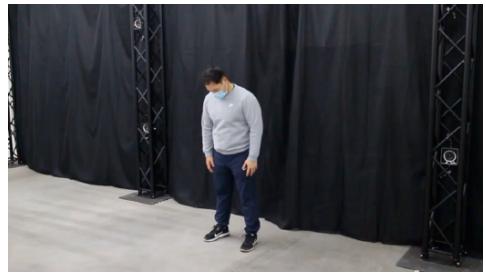
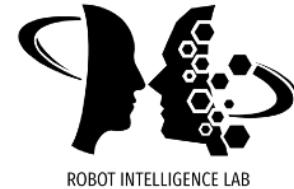
[1/500] Post-Humanoidization



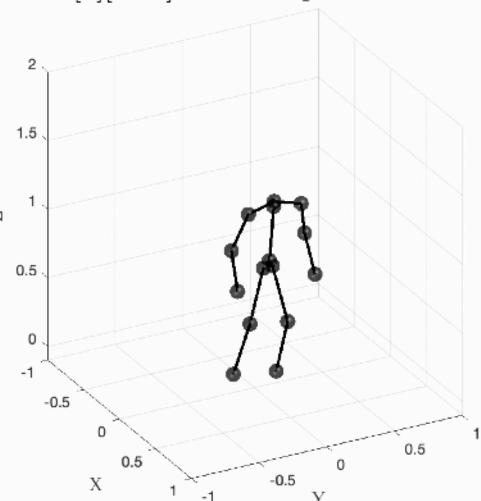
[1/500] Post-Humanoidization



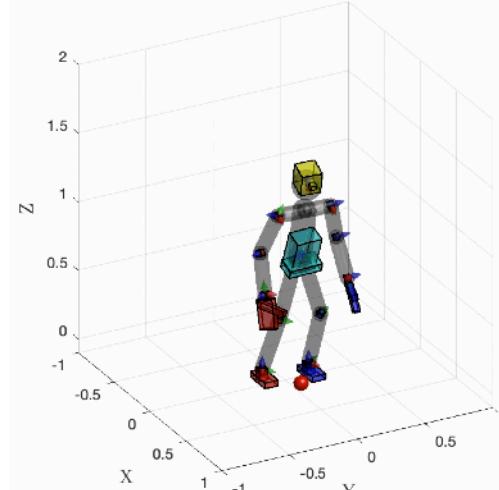
# Robust Motion Retargeting



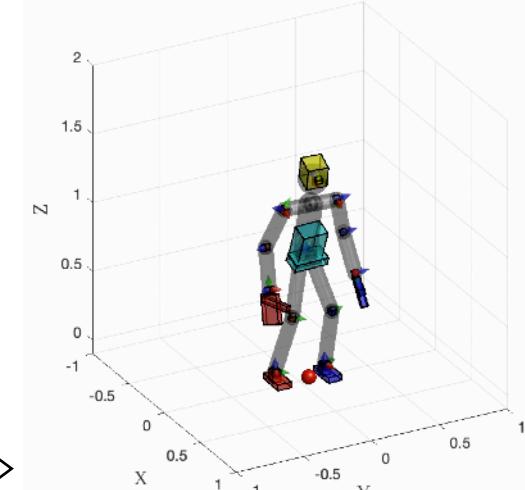
[1][0.10]sec MoCap Skeleton



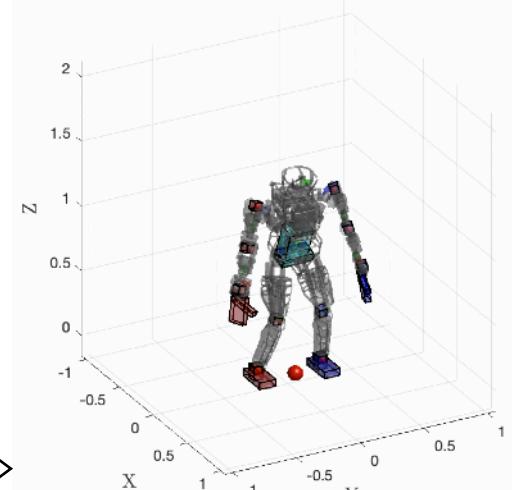
[1][0.10]sec Pre-Humanoidize



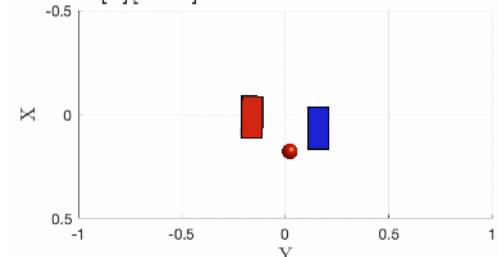
[1][0.10]sec Post-Humanoidize



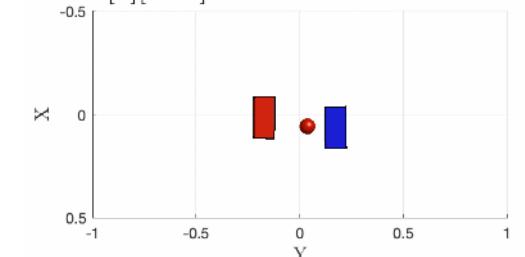
[1][0.10]sec [atlas]



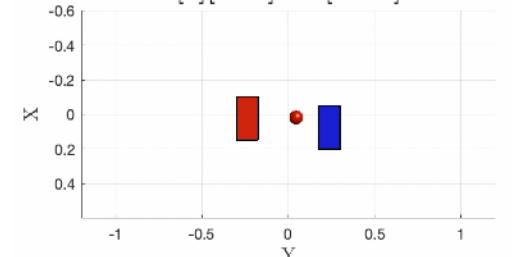
[1][0.10]sec Pre-Humanoidize



[1][0.10]sec Post-Humanoidize



[1][0.10]sec [atlas]



# Thank You



ROBOT INTELLIGENCE LAB



# Interactive Gaze

Realistic and Interactive Robot Gaze, IROS 2020

# REALISTIC AND INTERACTIVE ROBOT GAZE

WITH AUDIO

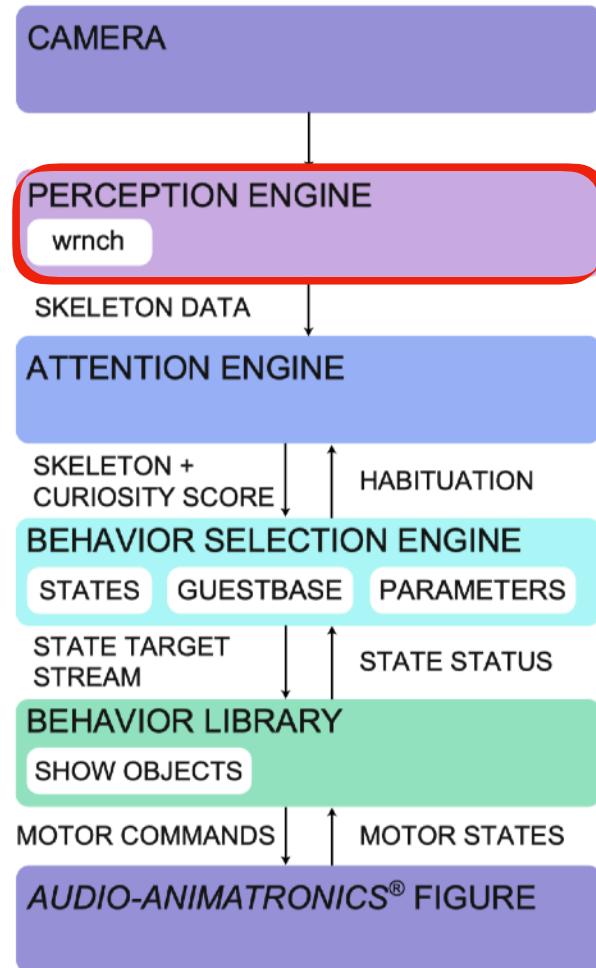
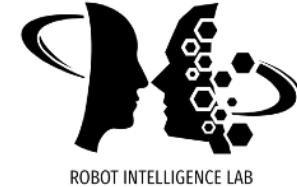
VIDEO SUBMISSION OF PAPER 1439 TO IROS 2020



Disney Research

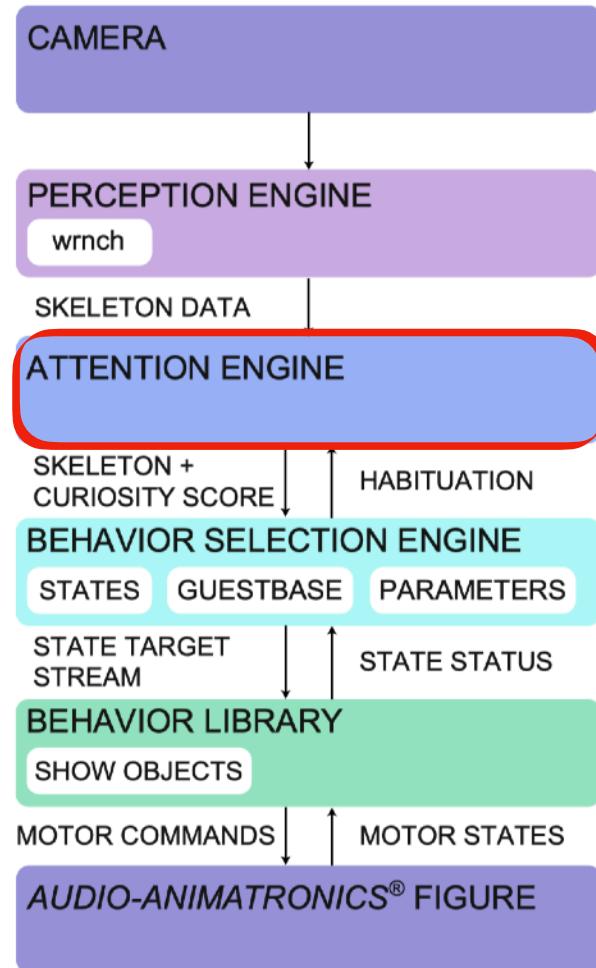
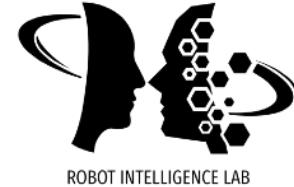
© Disney

# Interactive Gaze



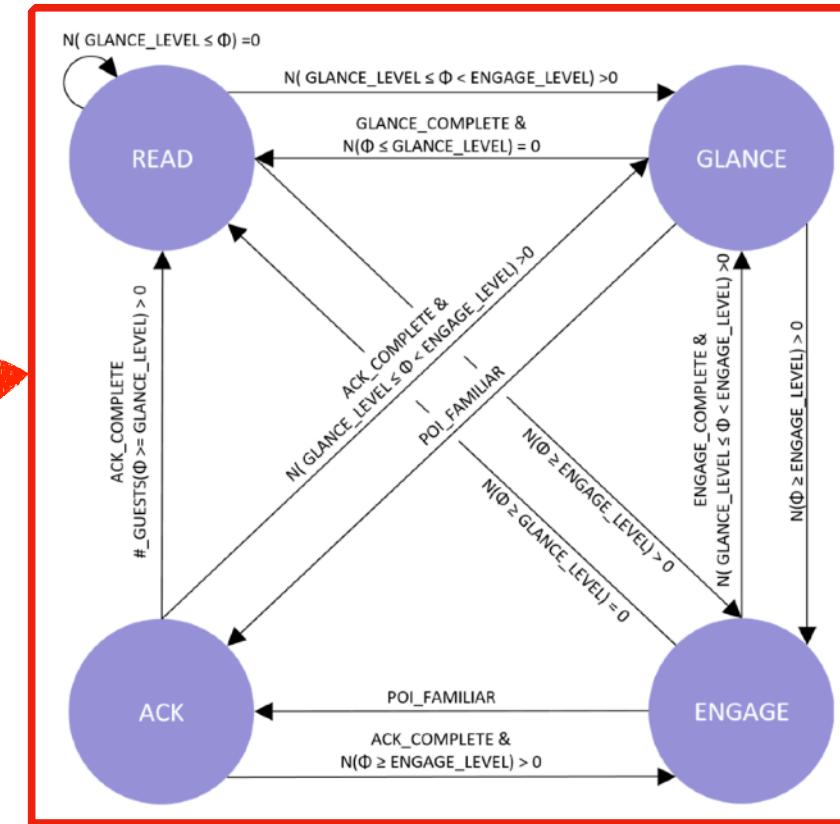
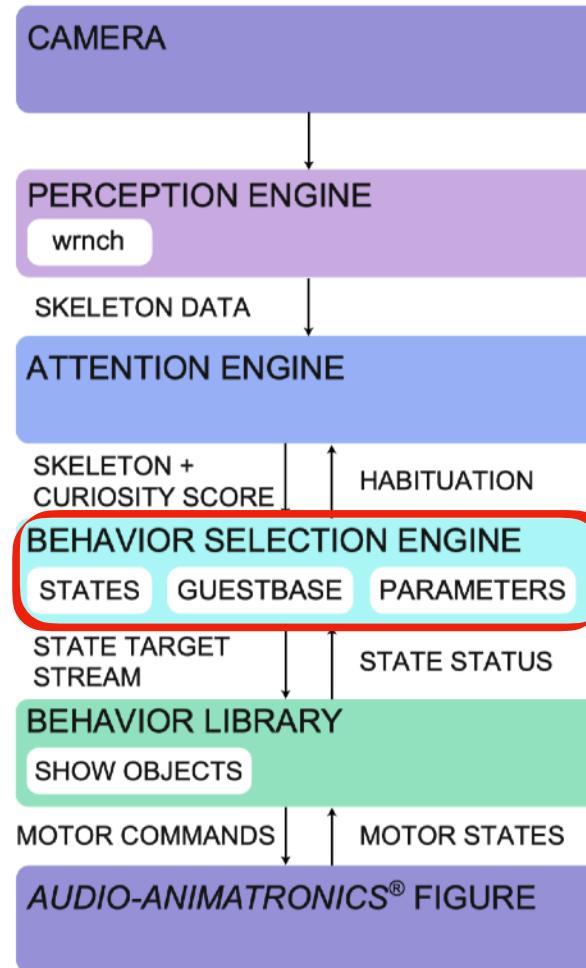
**Perception engine** infers useful information from visual observation such as skeleton poses on the scene.

# Interactive Gaze

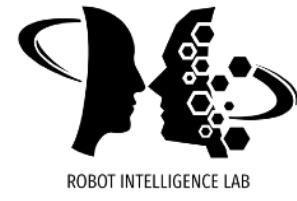


**Attention engine** computes the **curiosity level** of each person from the sequence of skeleton data.

# Interactive Gaze



**Behavior selection engine** decides which behavior the robot should select based on the curiosity levels of each guest.



ROBOT INTELLIGENCE LAB