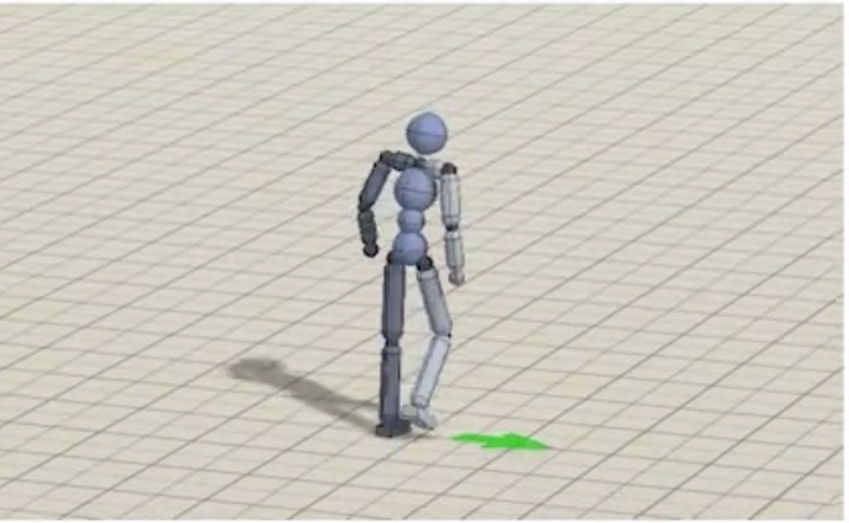
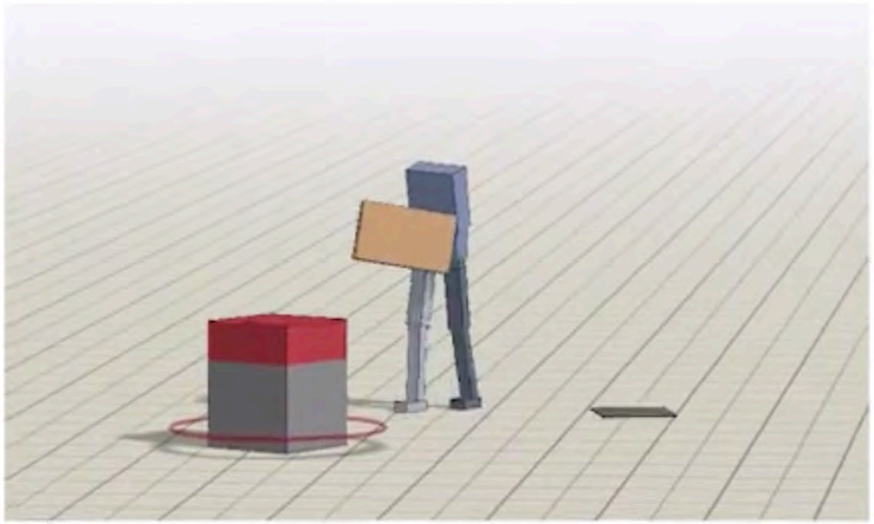
MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies

(with audio)


Xue Bin Peng, Michael Chang, Grace Zhang,
Pieter Abbeel, Sergey Levine

University of California
Berkeley

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\,\pi_i(a|s,g)$$

gating function

primitive

Standard hierarchical policies compose primitive skills by using a gating function,

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\,\pi_i(a|s,g)$$

gating function

primitive

which specifies the probability of activating each primitive in a given scenario.

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\,\pi_i(a|s,g)$$

gating function

primitive

One of the limitations of this model is that only one primitive can be activated at each timestep,

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\,\pi_i(a|s,g)$$

gating function

primitive

which can restrict the range of behaviors that can be produced by the composite policy.

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\pi_i(a|s,g)$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s,g)^{w_i(s,g)}$$

We propose combining primitives using a multiplicative composition scheme,

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\pi_i(a|s,g)$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s,g)^{w_i(s,g)}$$

which enables multiple primitives to be activated simultaneously, and contribute to the composite policy's action distribution.

Overview

Additive Model:

$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\pi_i(a|s,g)$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s,g)^{w_i(s,g)}$$

gating function

The weights from the gating function specify each primitive's influence on the composite distribution,

Overview

Additive Model:

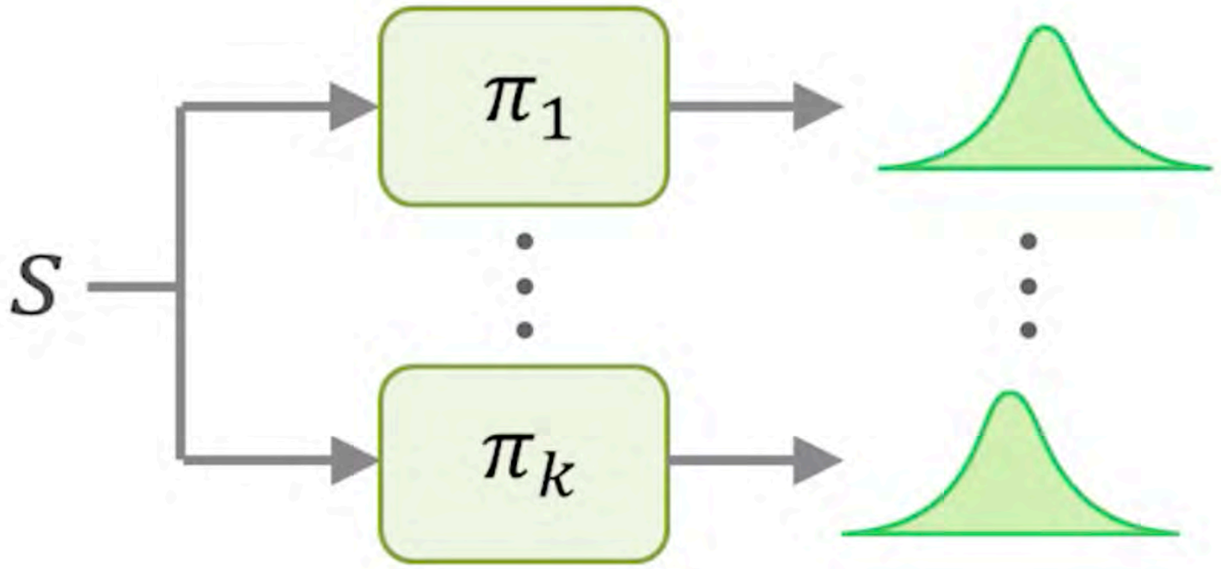
$$\pi(a|s,g) = \sum_{i=1}^{k} w_i(s,g)\pi_i(a|s,g)$$

$$\pi(a|s,g) = \frac{1}{Z(s,g)} \prod_{i=1}^{k} \pi_i(a|s,g)^{w_i(s,g)}$$

gating function

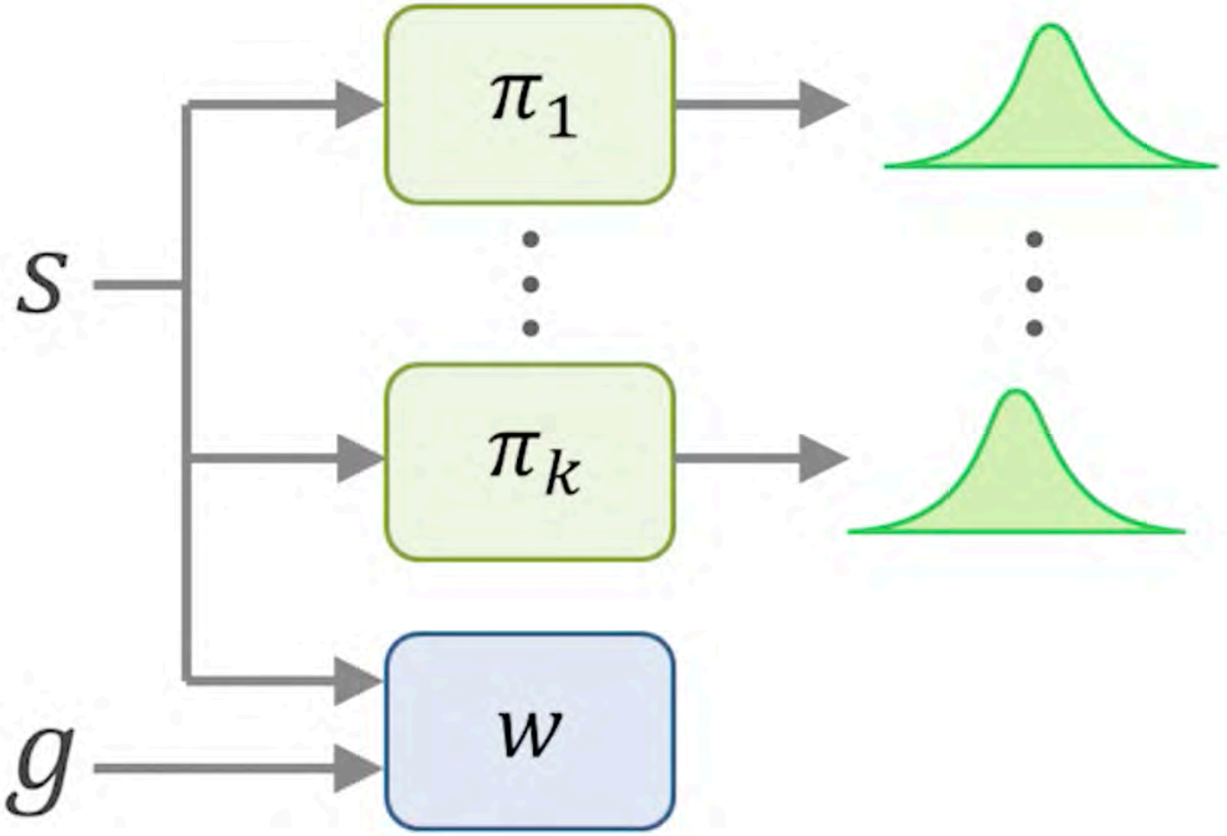
with a higher weight corresponding
to a larger influence.

Multiplicative Composition Policy (MCP)



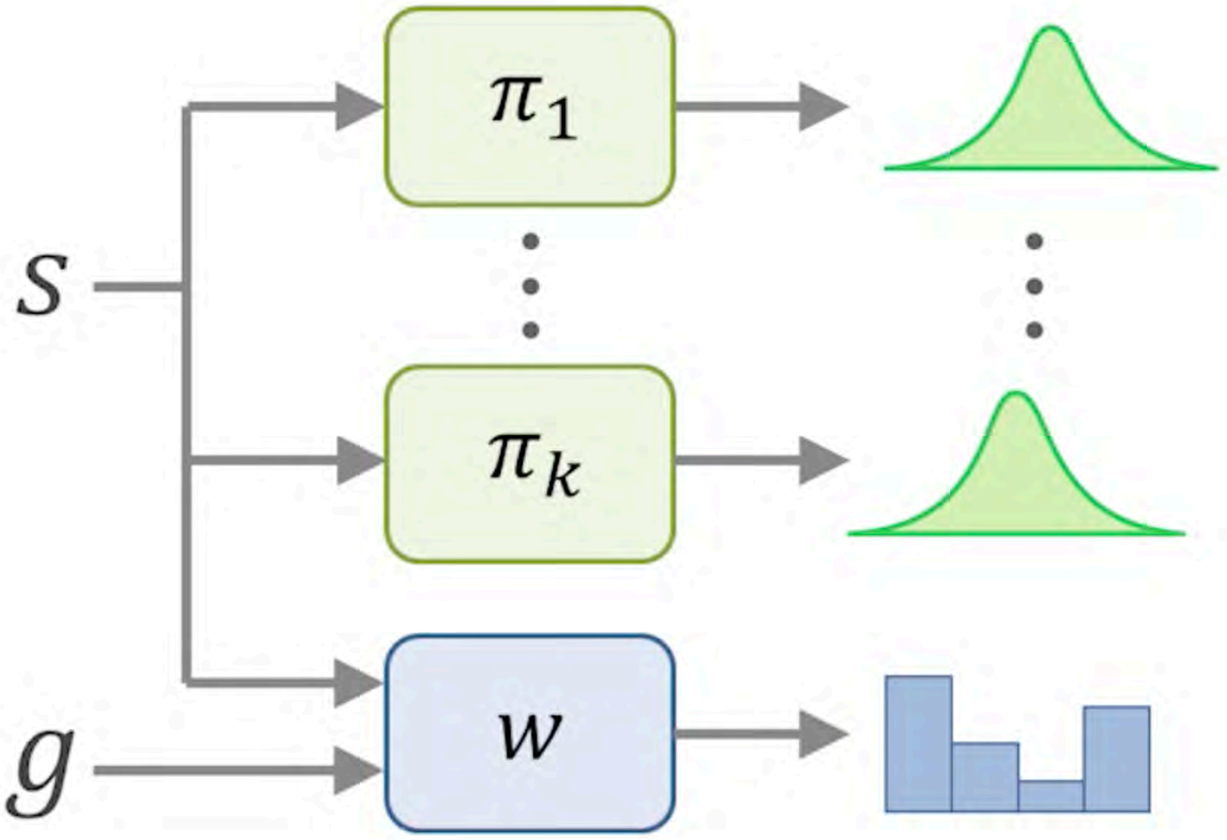
Given a state, each primitive proposes an action distribution in response to that state.

Multiplicative Composition Policy (MCP)



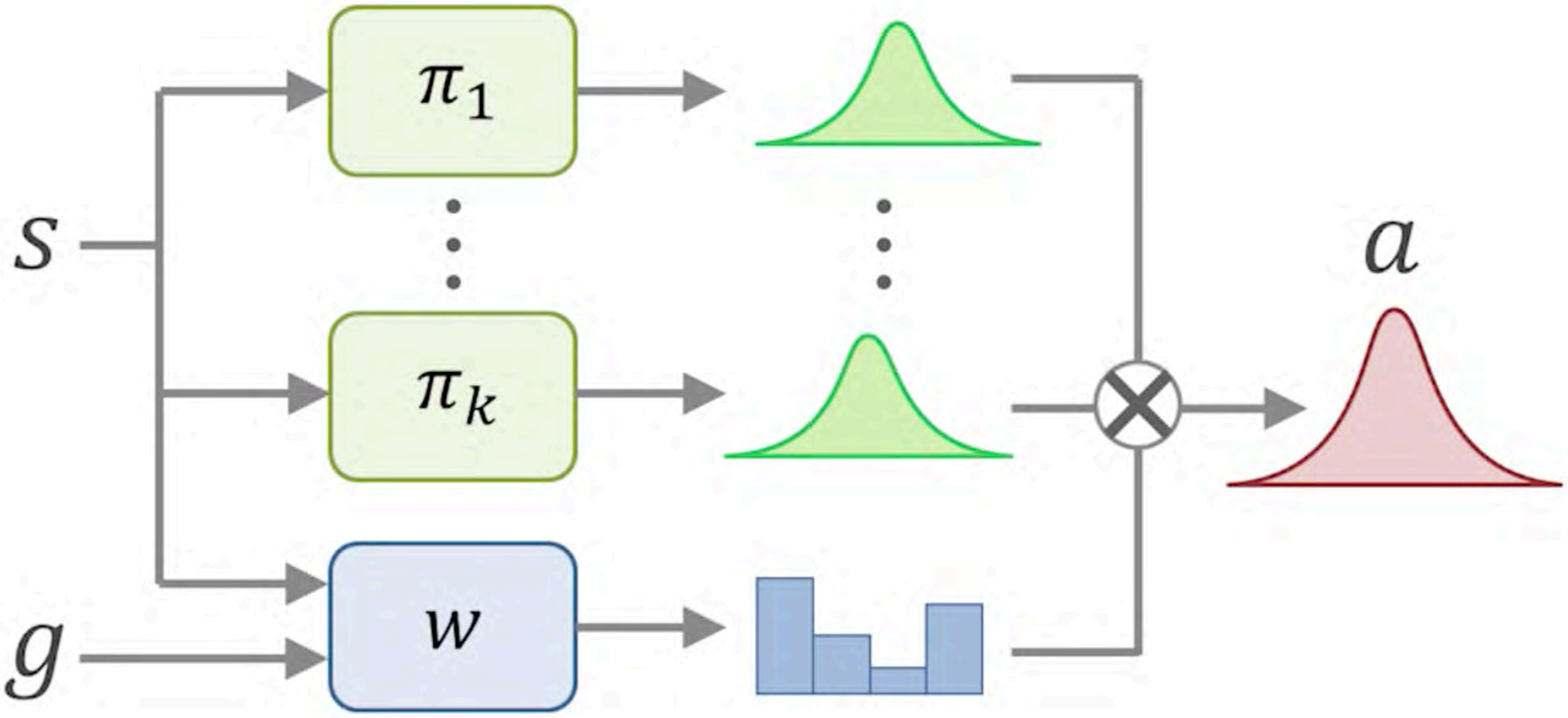
The gating function receives both the state and a task specific goal as input,

Multiplicative Composition Policy (MCP)



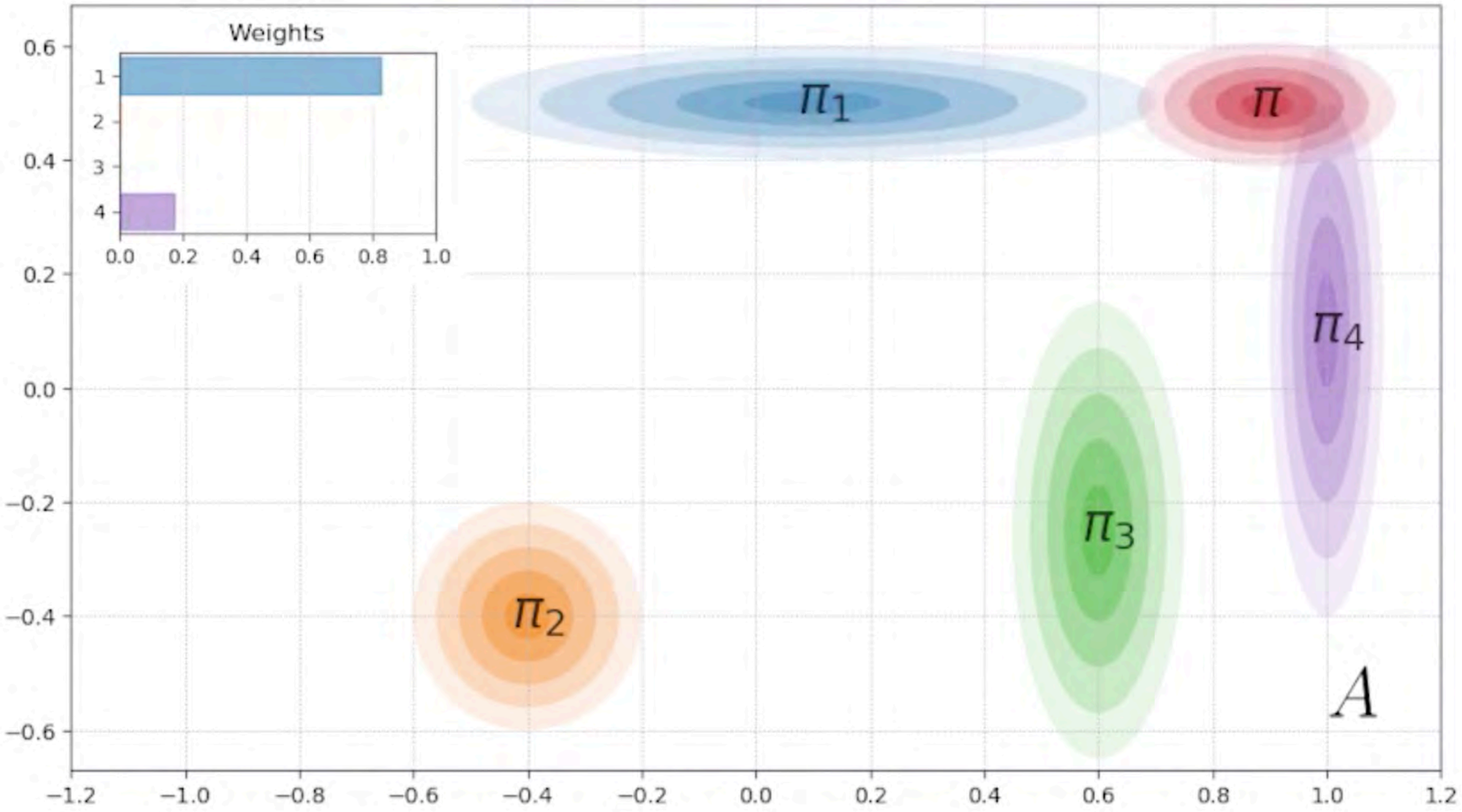
then outputs the weights for each primitive.

Multiplicative Composition Policy (MCP)



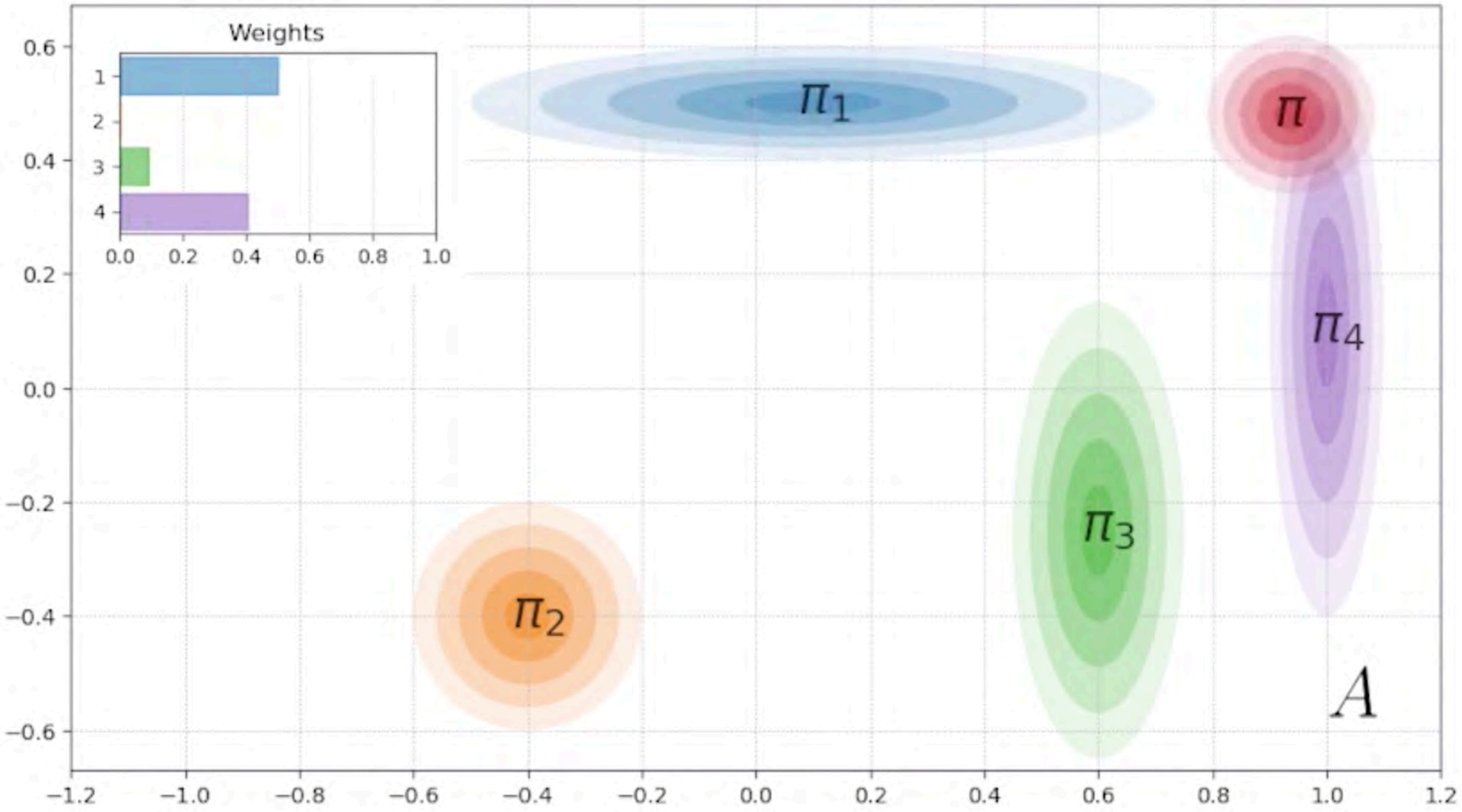
The distributions are then composed according to the weights to produce the composite action distribution.

Gaussian Primitives



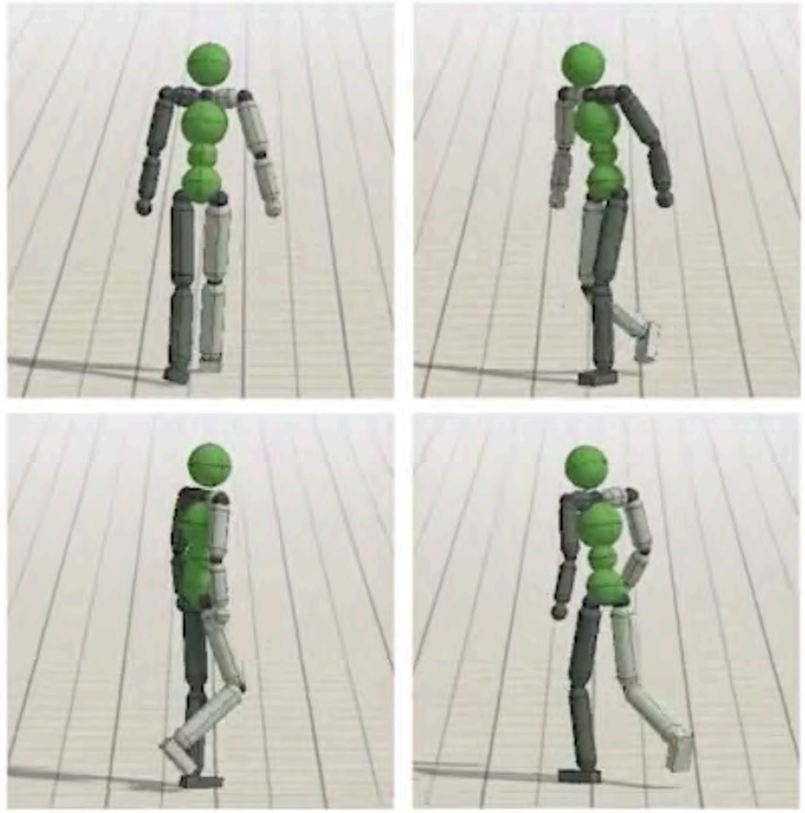
Each primitive's action distribution is modeled by a Gaussian.

Gaussian Primitives



Varying the weights produces different interpolations of the primitives' distributions.

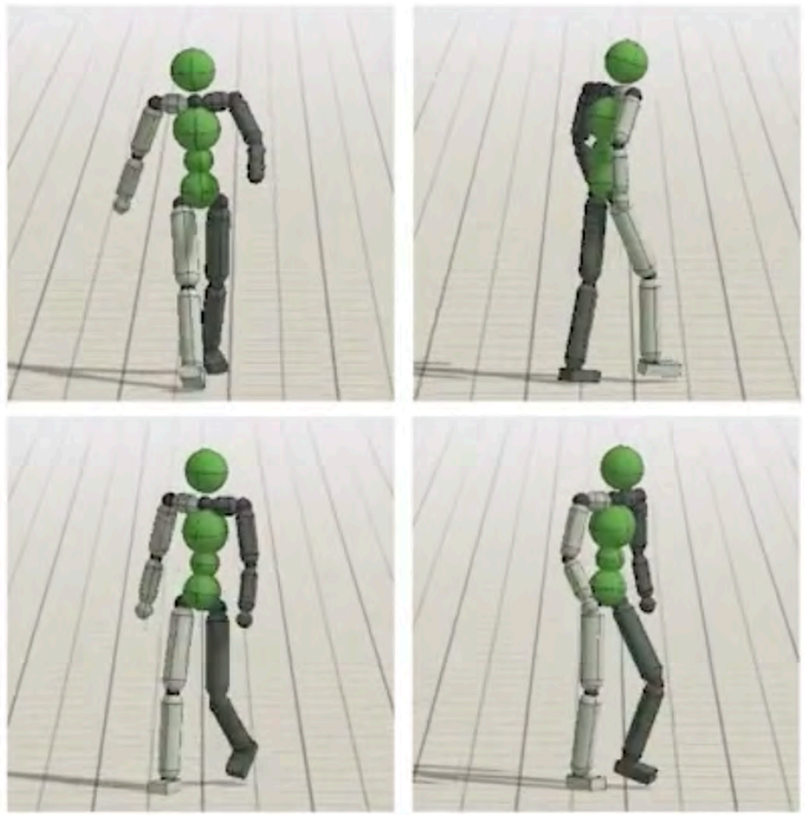
Transfer



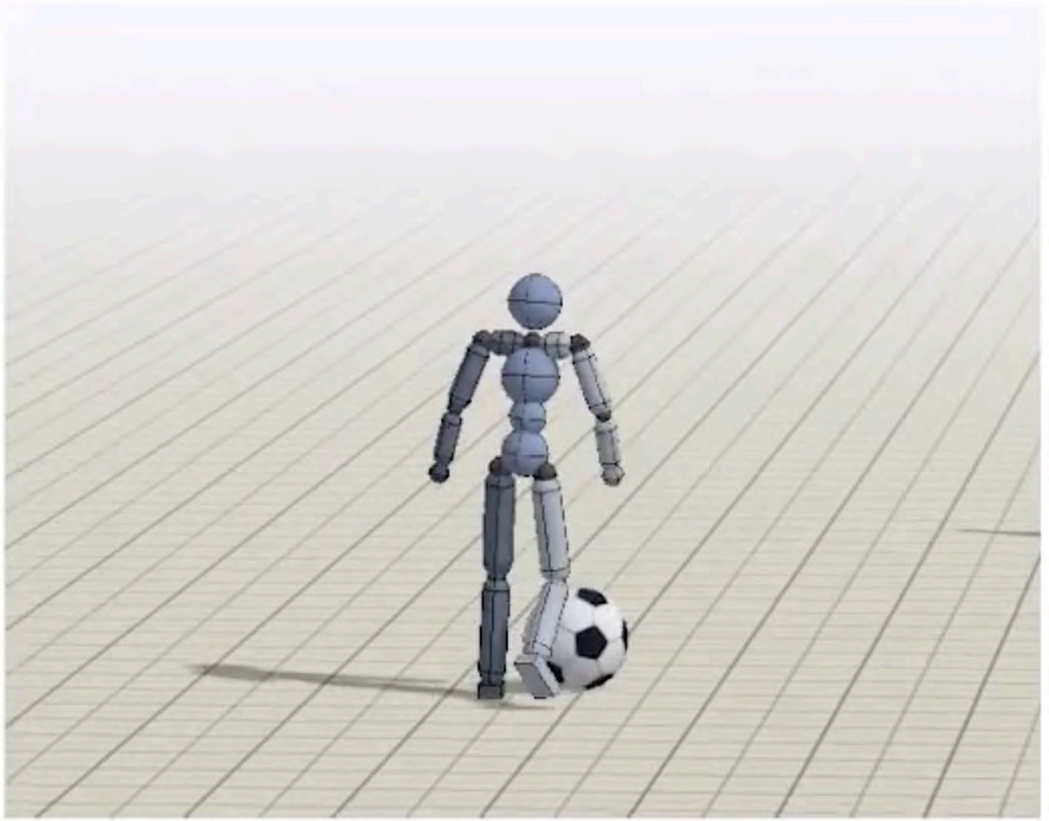
Pre-training

Primitives are learned through pre-training tasks that encourage the primitives to specialize in different skills.

Transfer



Pre-training

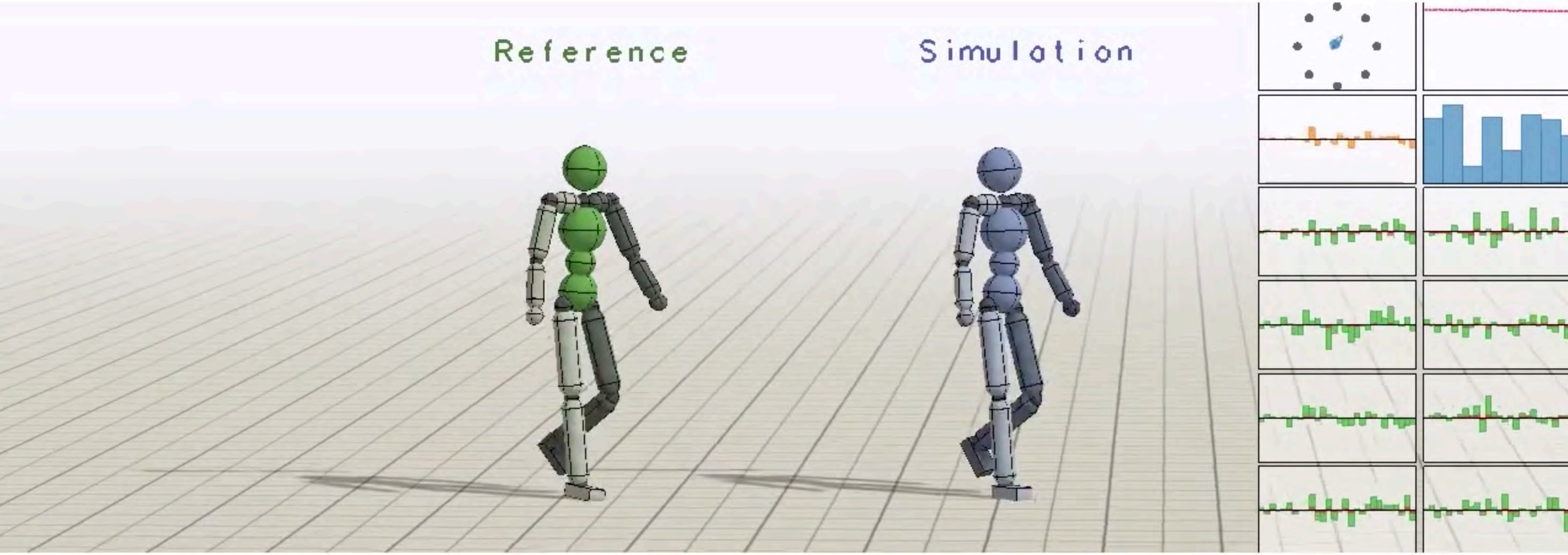


Transfer

When transfering the primitives to new tasks, a new gating function is trained to compose the primitives for the new task.

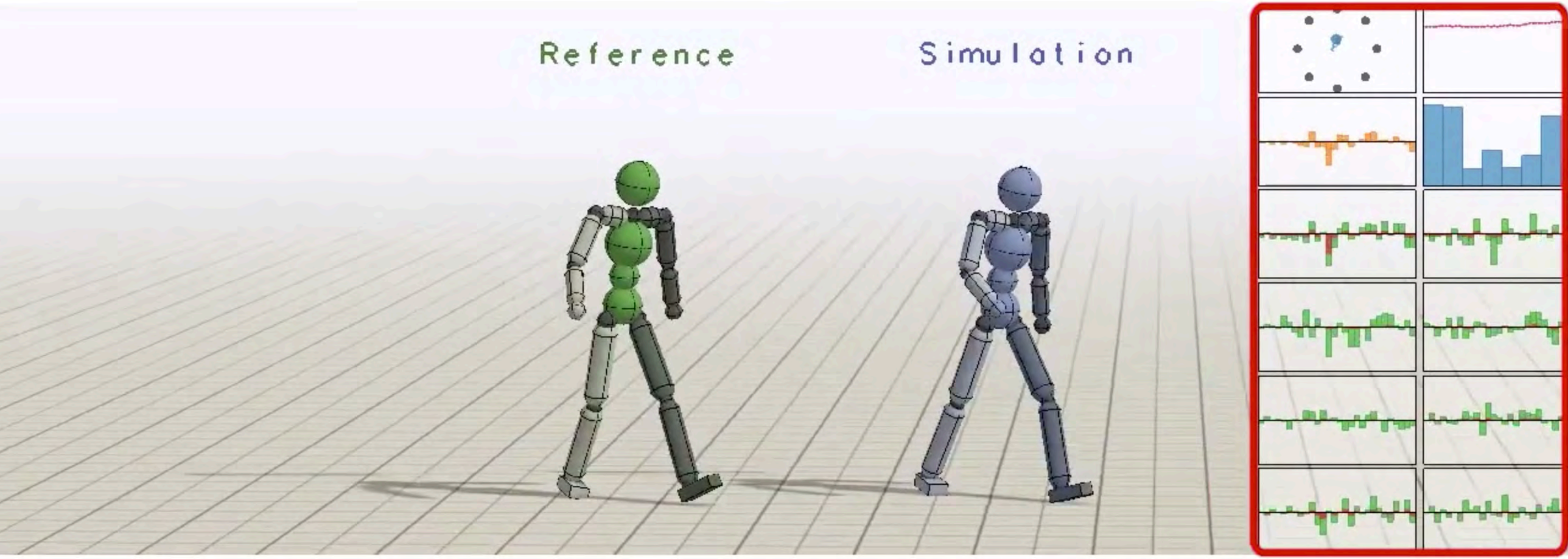
Pre-Training

Pre-Training: Humanoid



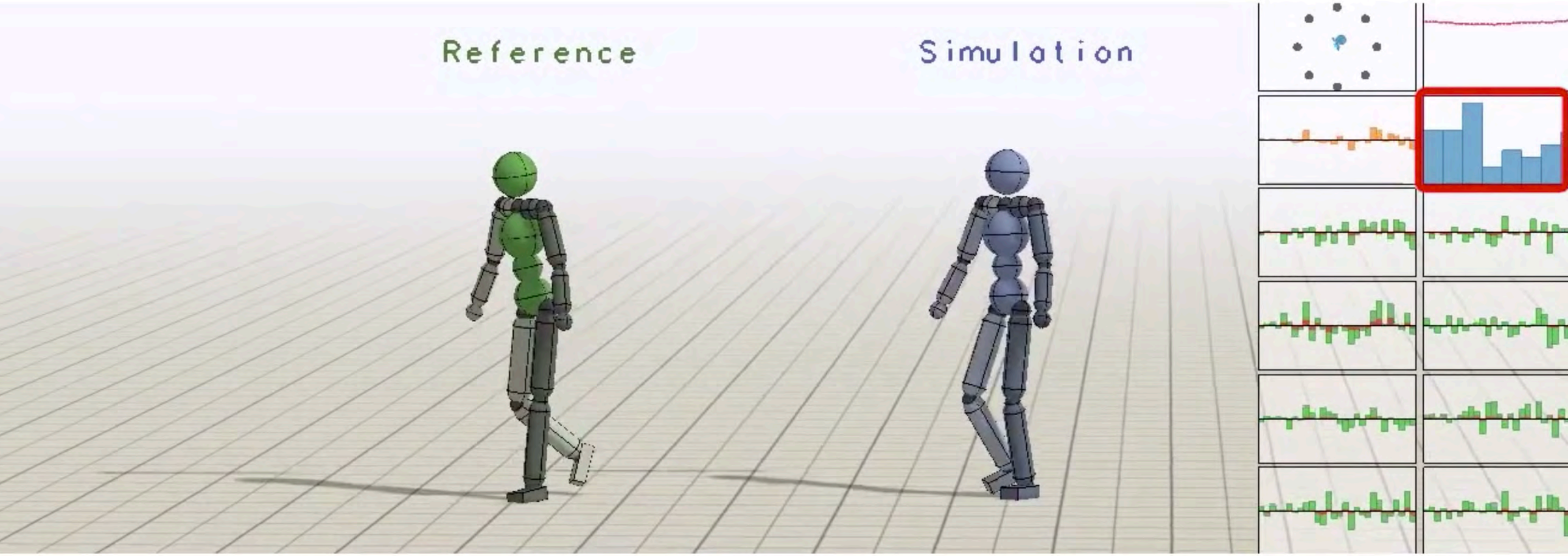
The primitives are trained by imitating reference motions, such as mocap clips recorded from human actors.

Pre-Training: Humanoid



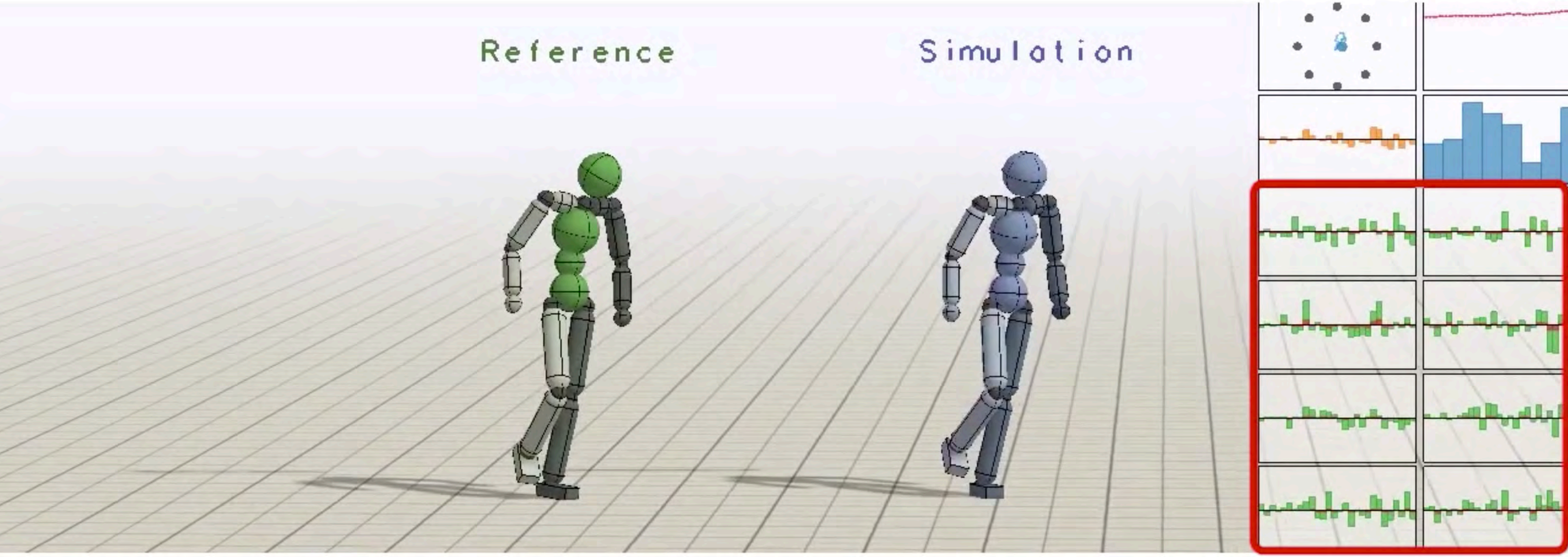
The plots visualize the behaviours of the different primitives.

Pre-Training: Humanoid



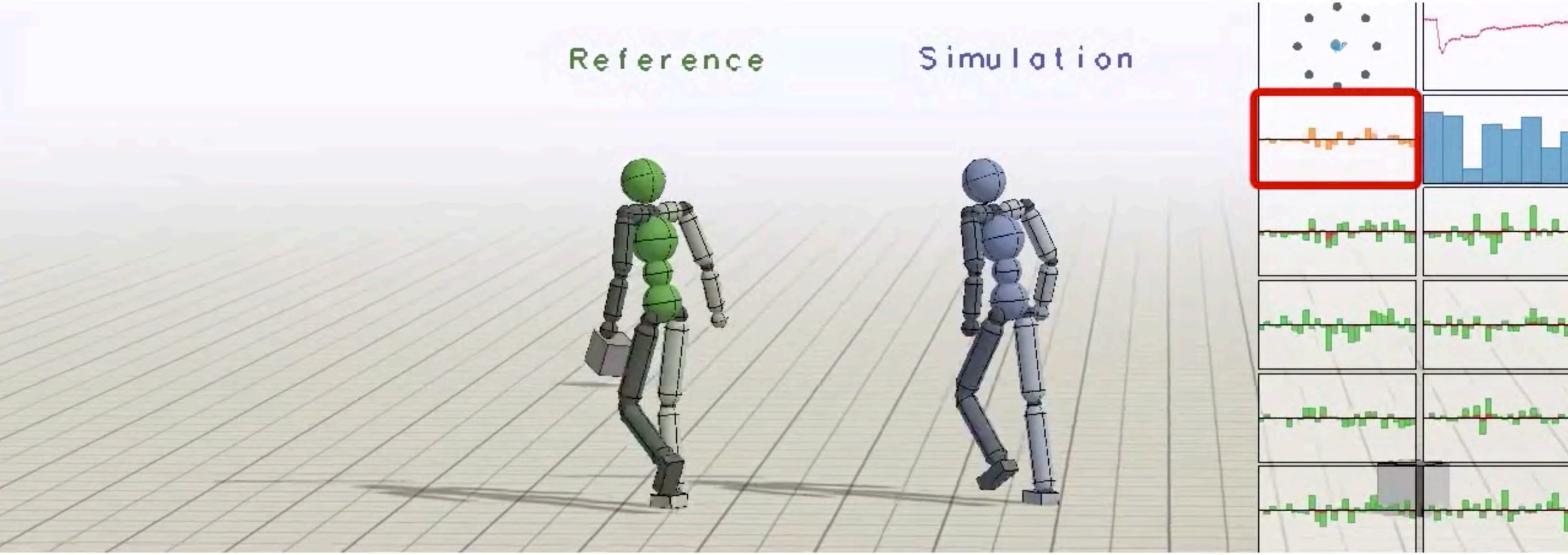
The blue plot shows the weights
for the primitives.

Pre-Training: Humanoid



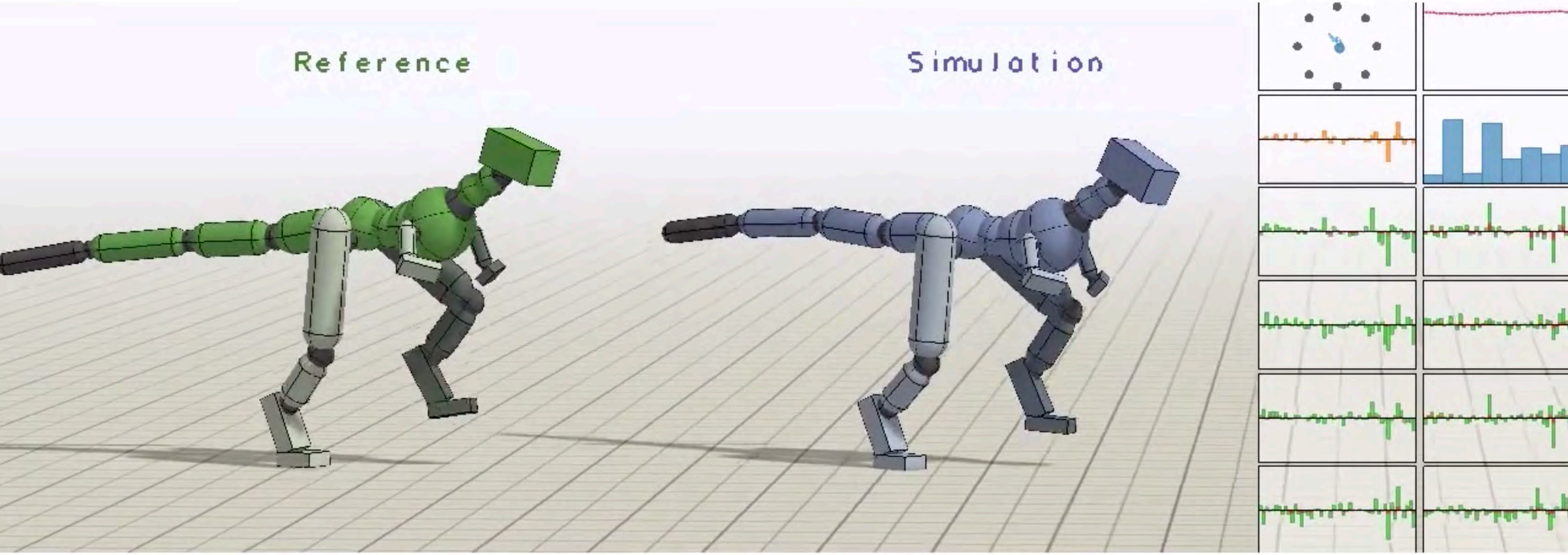
The green plots show the actions proposed by each primitive.

Pre-Training: Humanoid



The orange plot shows the action produced by composing the primitives.

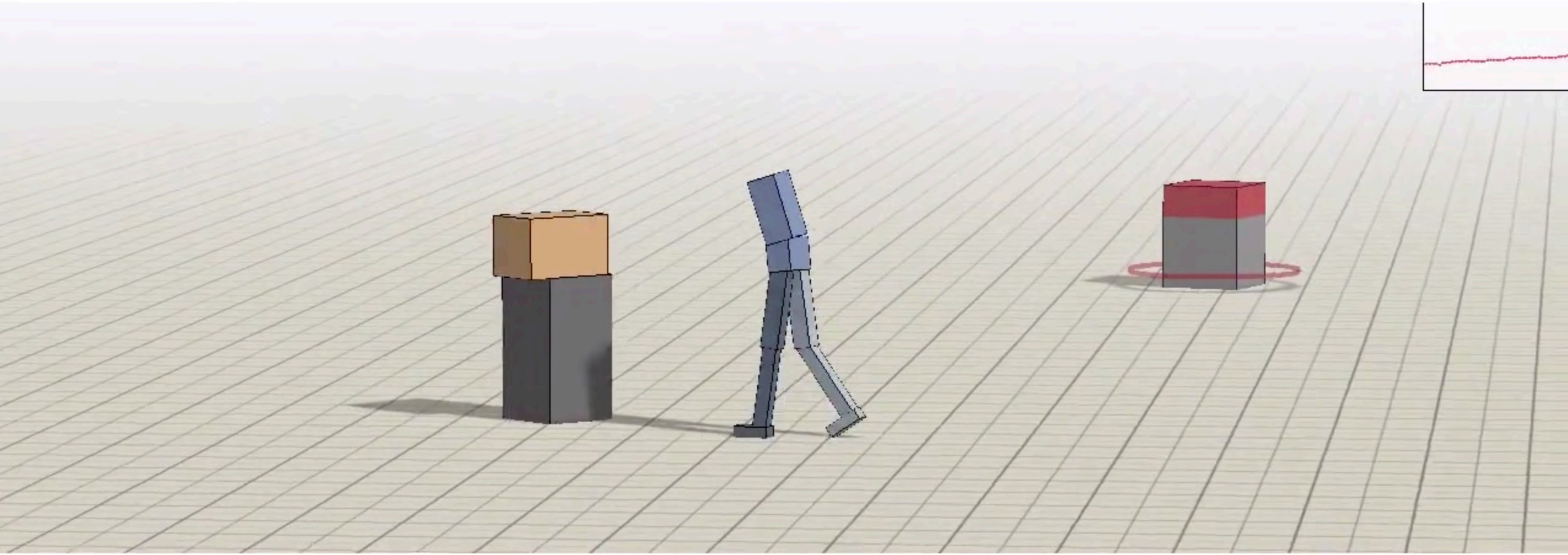
Pre-Training: T-Rex



We can also learn primitives for controlling a complex T-Rex character, with 55 degrees-of-freedom.

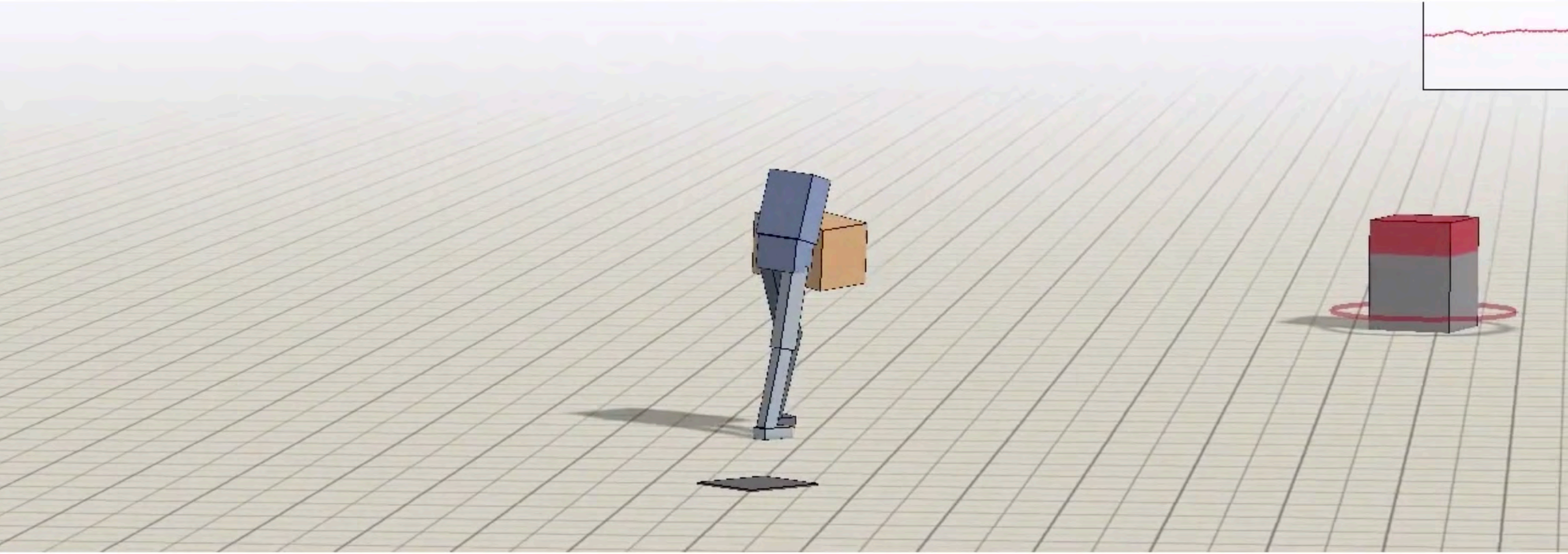
Transfer Tasks

Carry: Biped



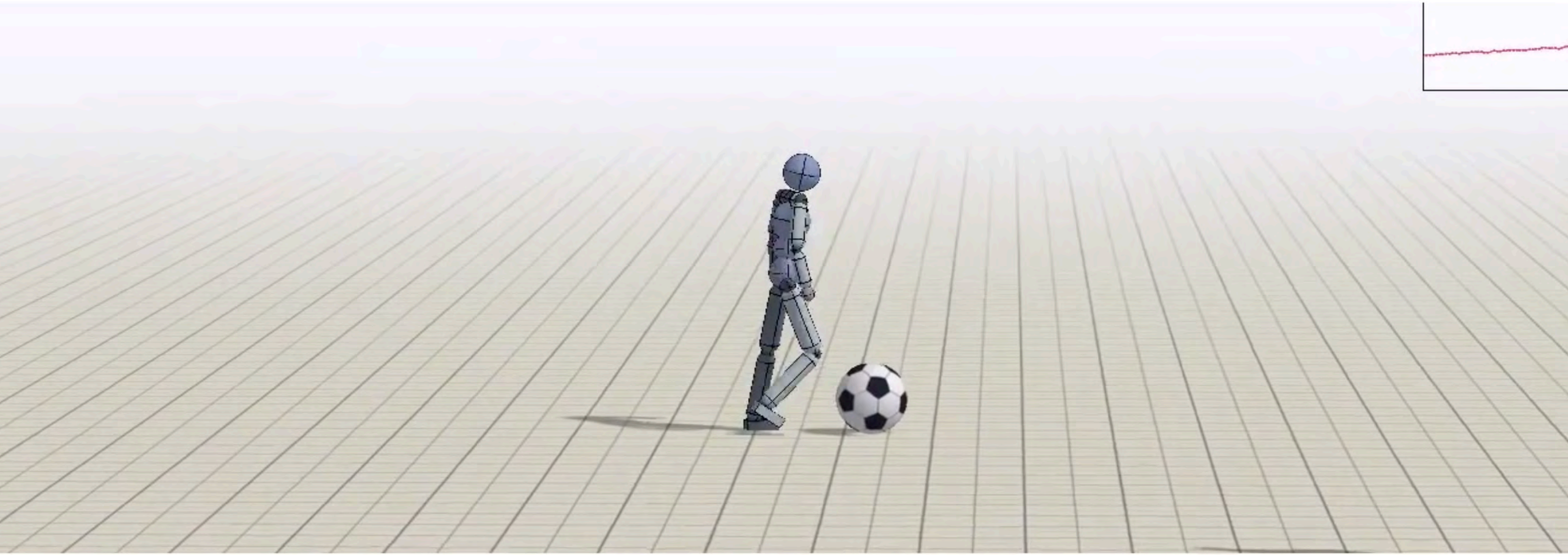
Once trained, the primitives can be transferred to challenging new tasks,

Carry: Biped



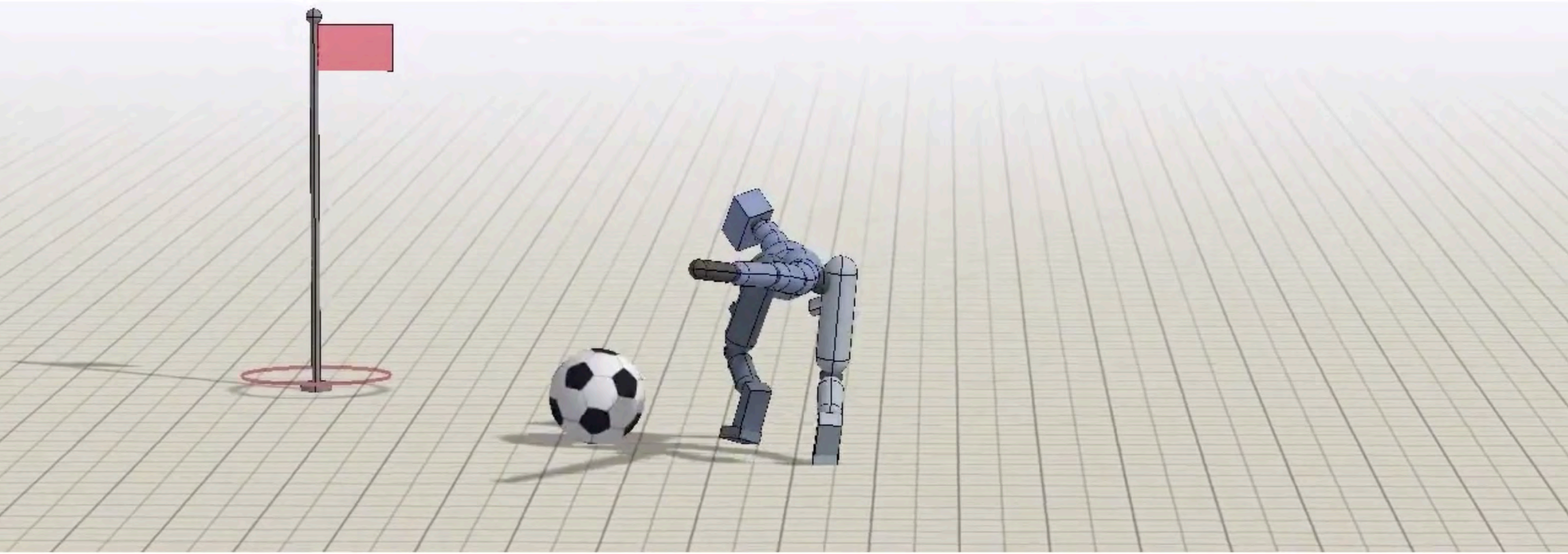
such as picking up an object and
carrying it to a target location.

Dribble: Humanoid

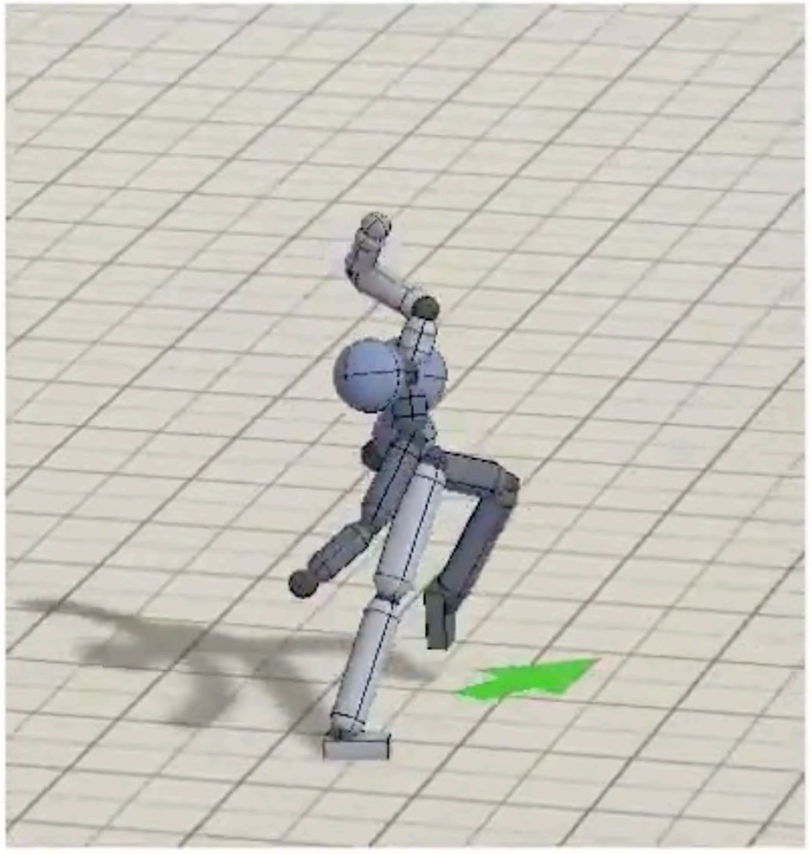


We can also train characters to dribble
a soccer ball to a goal.

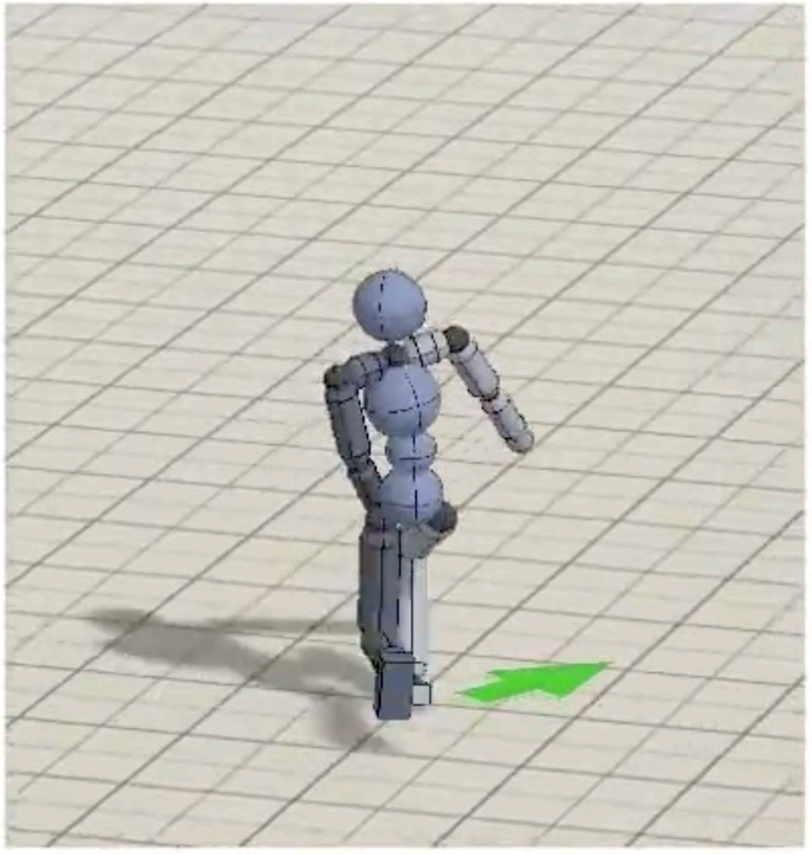
Dribble: T-Rex



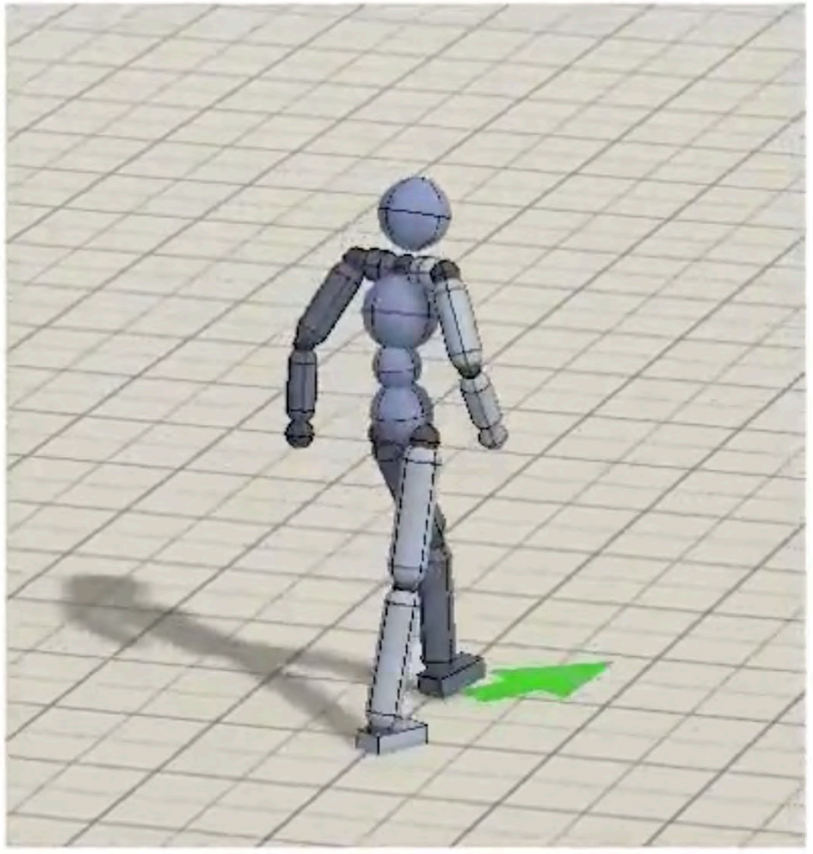
Heading: Humanoid



Scratch



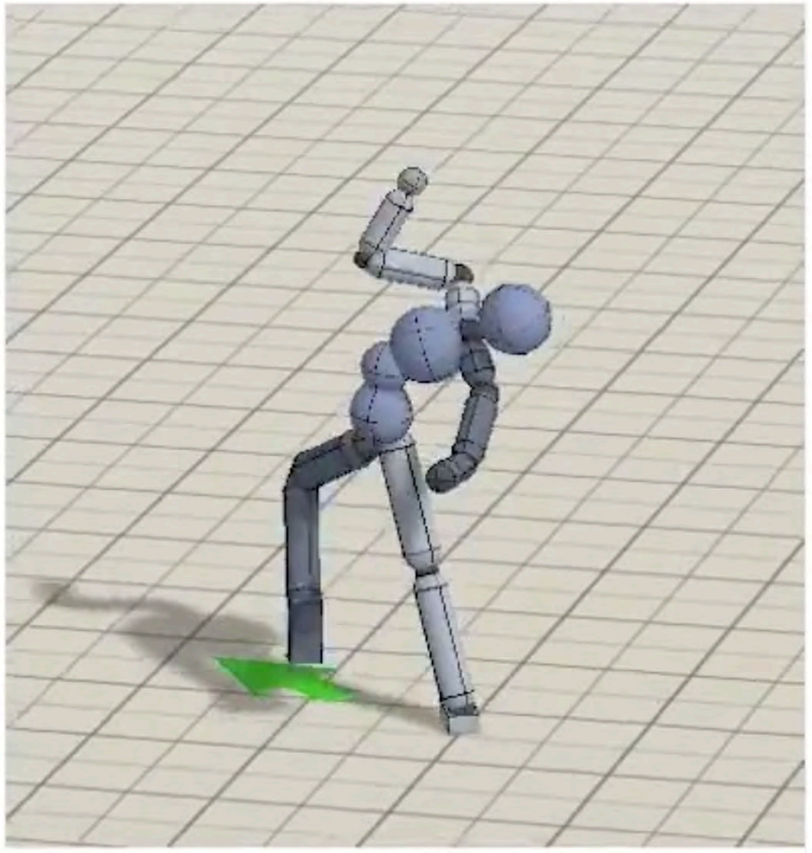
Finetune



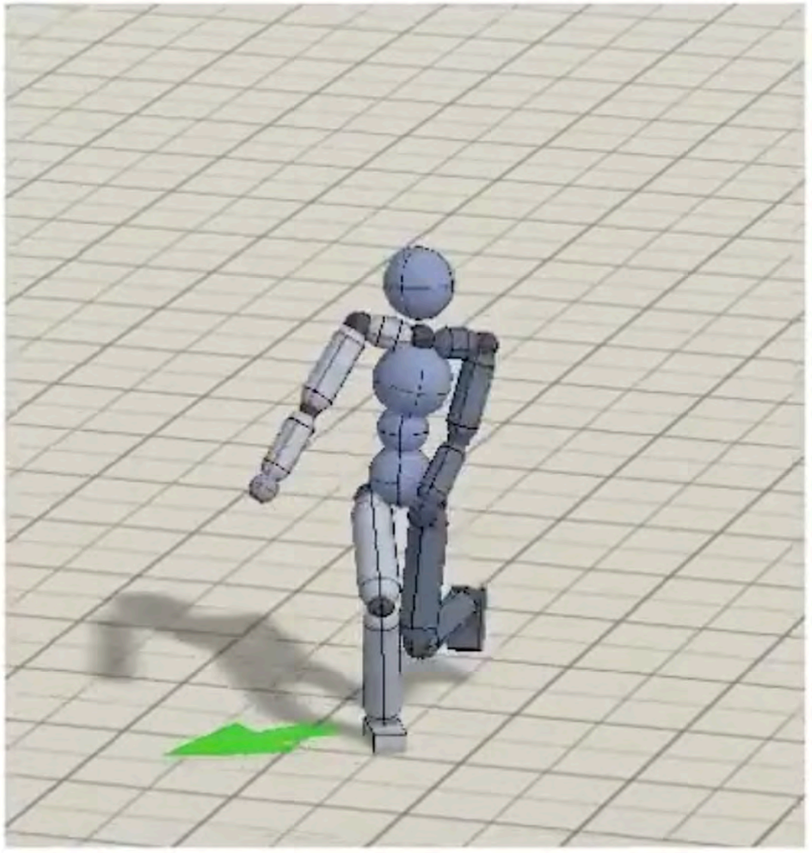
MCP (Ours)

We compare MCP to a number of prior methods.

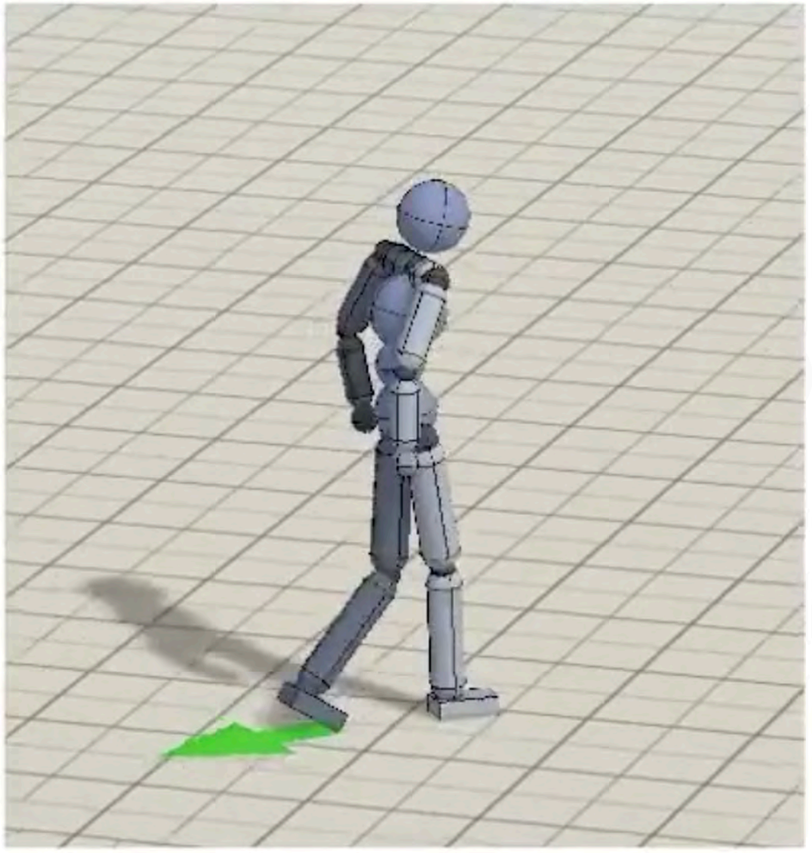
Heading: Humanoid



Scratch



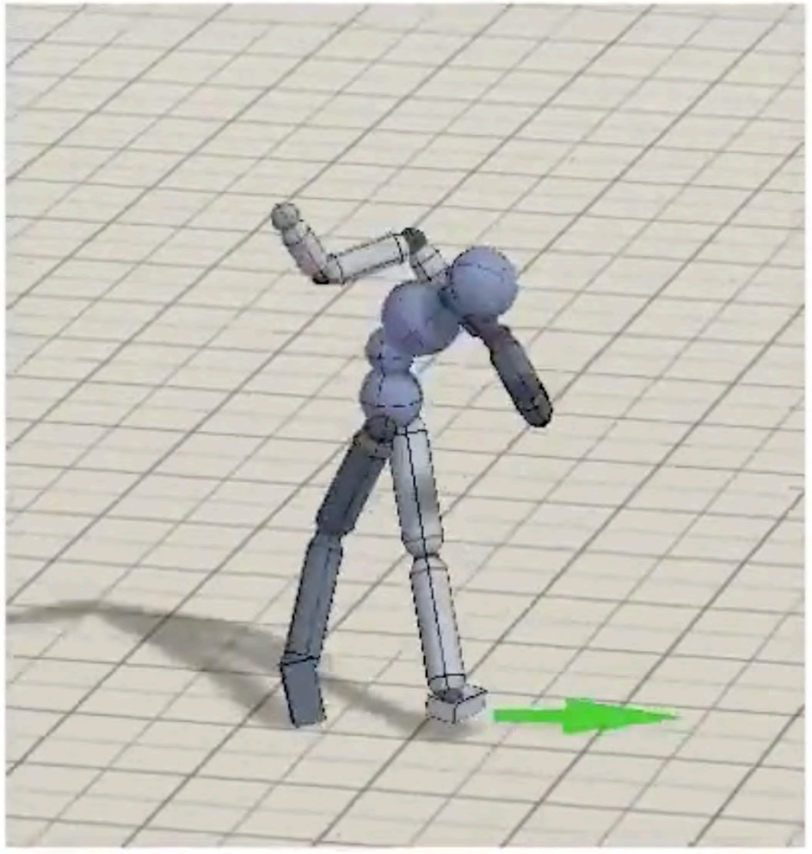
Finetune



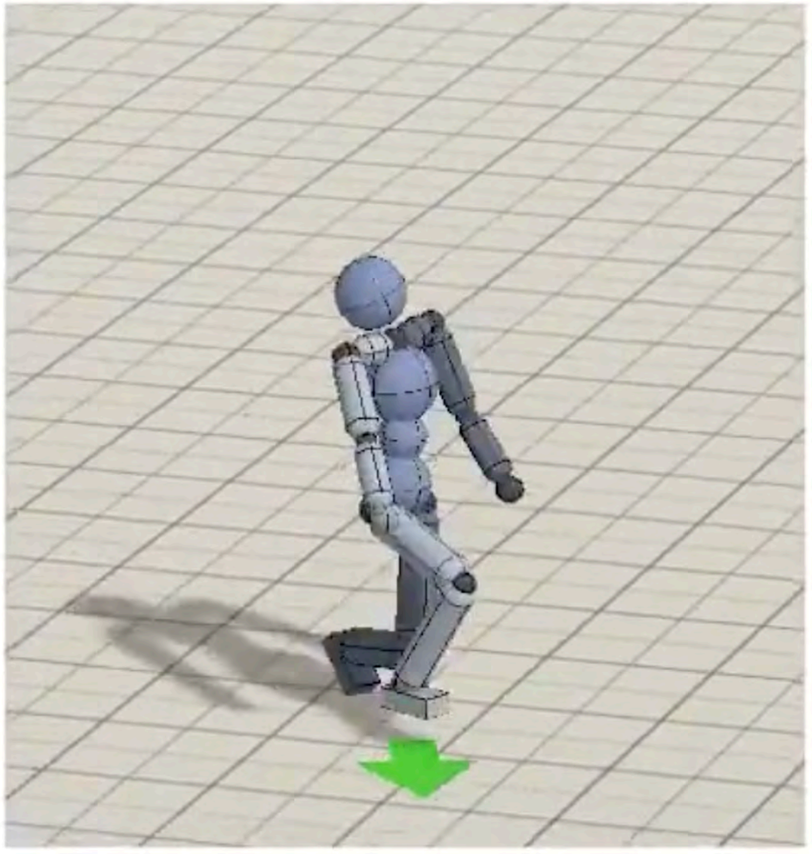
MCP (Ours)

Training from scratch tends to
produce unnatural behaviours.

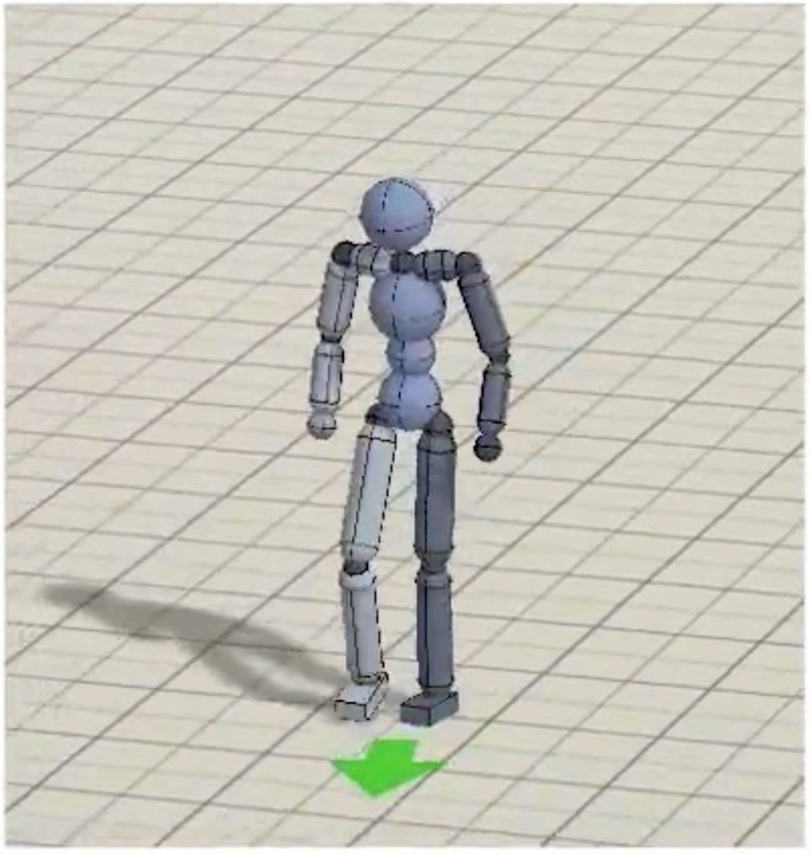
Heading: Humanoid



Scratch



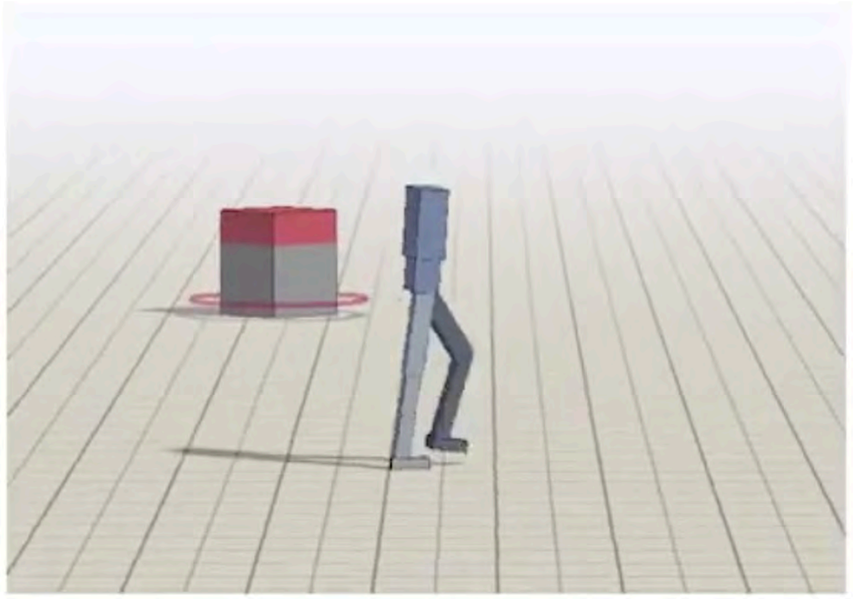
Finetune



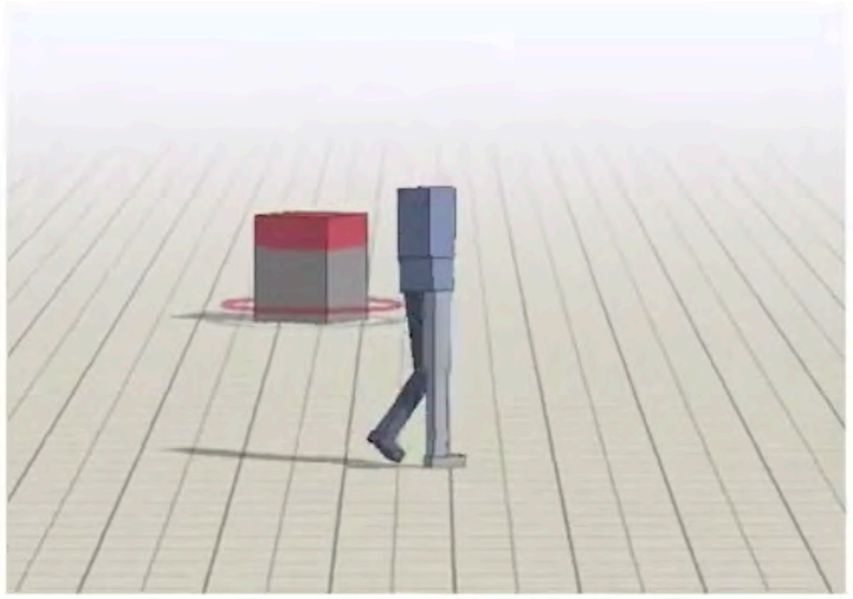
MCP (Ours)

MCP produces more natural motions.

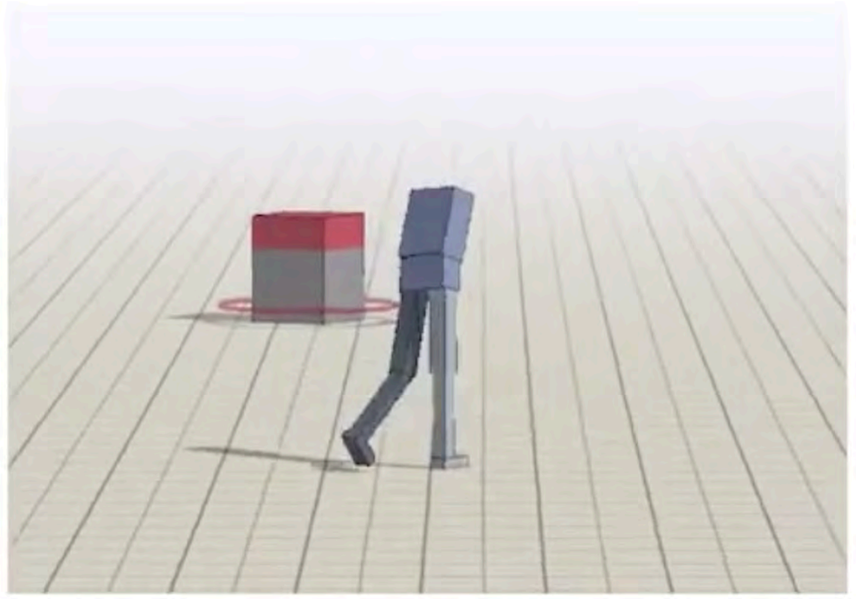
Carry: Biped



Hierarchical



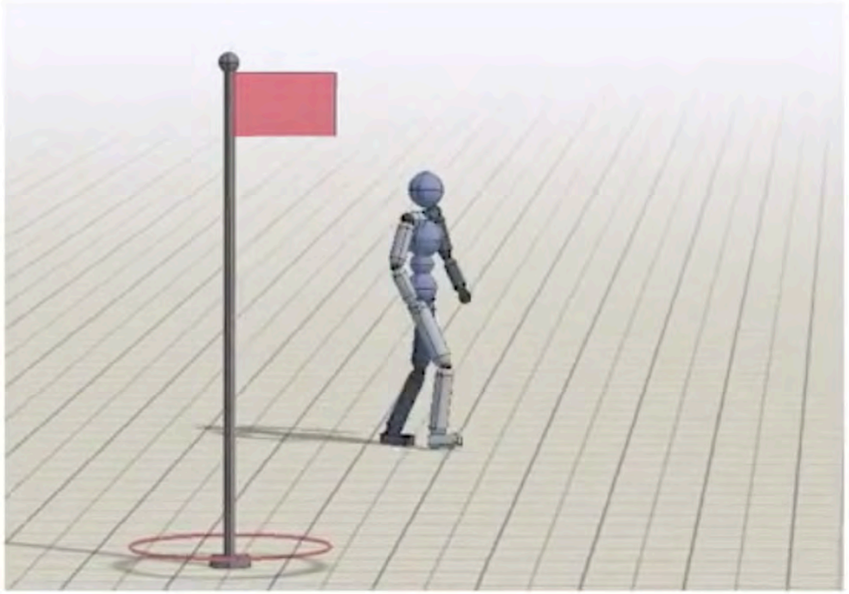
Mixture of Experts



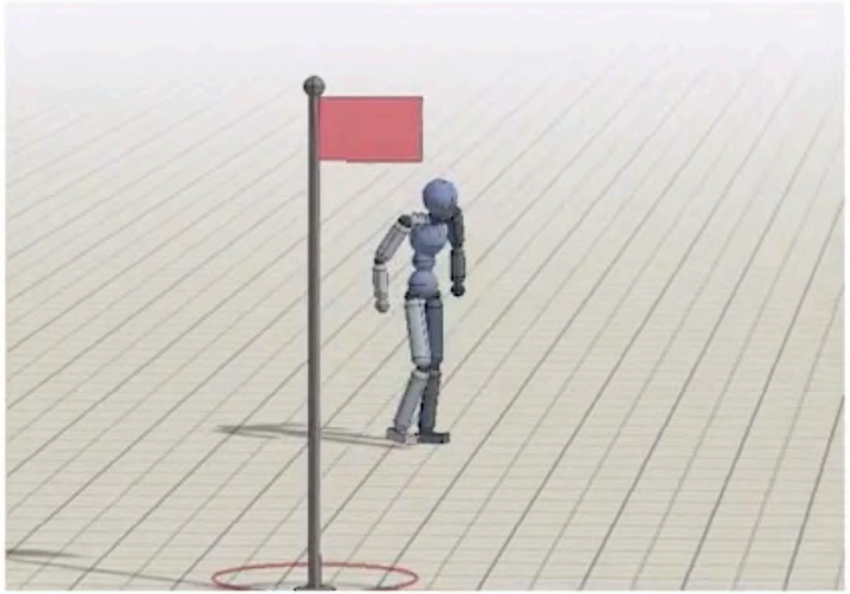
MCP (Ours)

Models that activate only a single primitive at a time can struggle with more complex tasks.

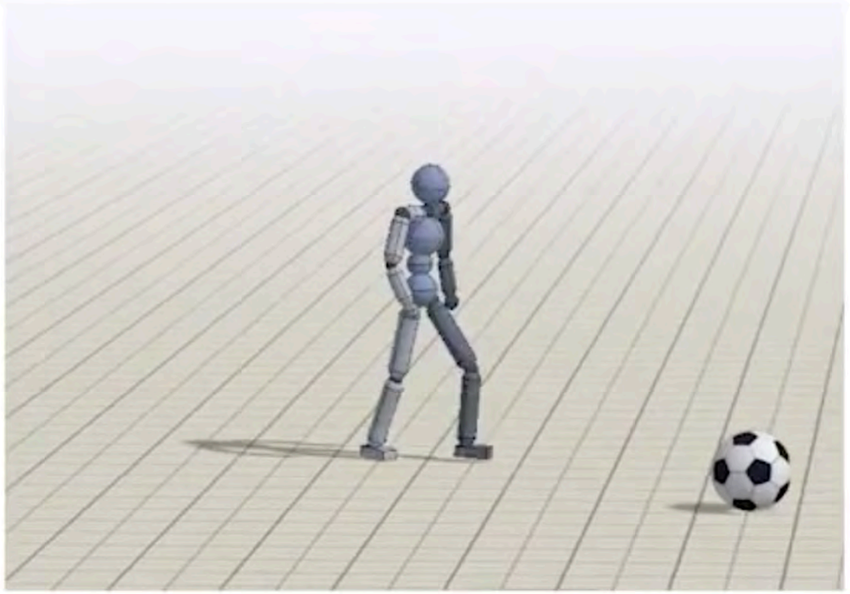
Dribble: Humanoid



Option-Critic
[Bacon et al., 2017]



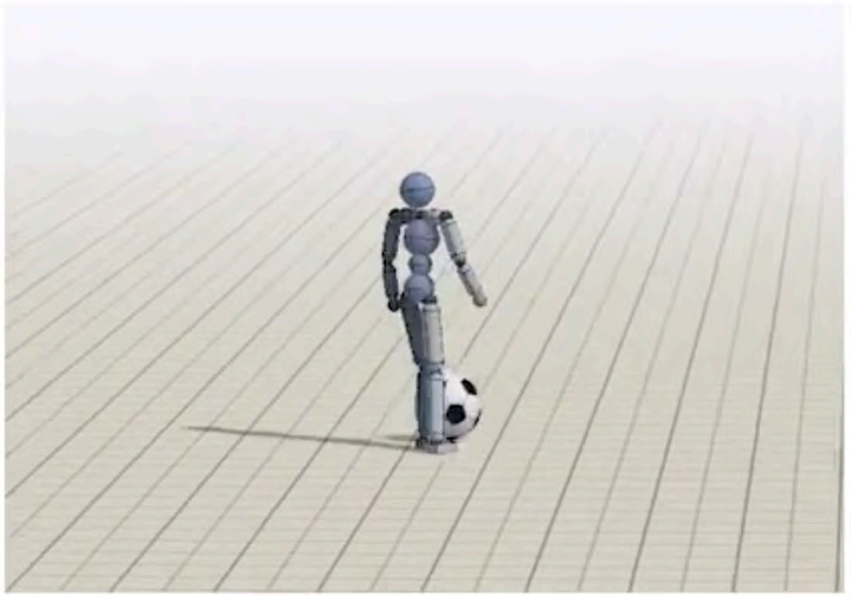
Latent Space
[Merel et al., 2018]



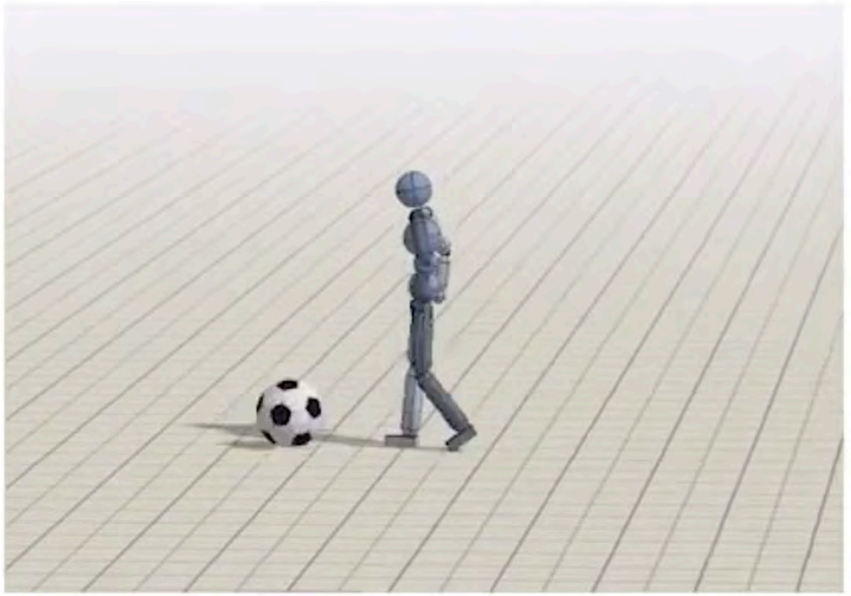
MCP (Ours)

MCP outperforms prior methods on
a number of challenging tasks.

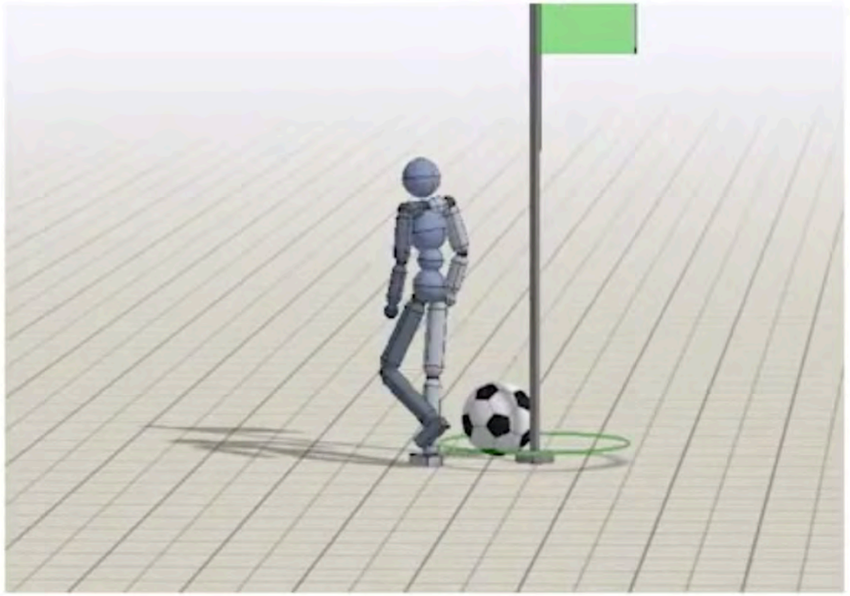
Dribble: Humanoid



Option-Critic
[Bacon et al., 2017]



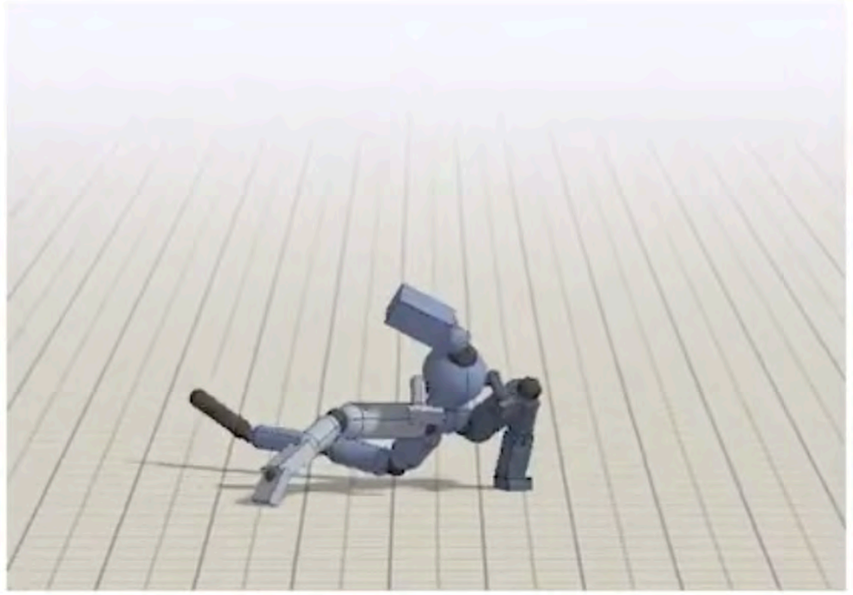
Latent Space
[Merel et al., 2018]



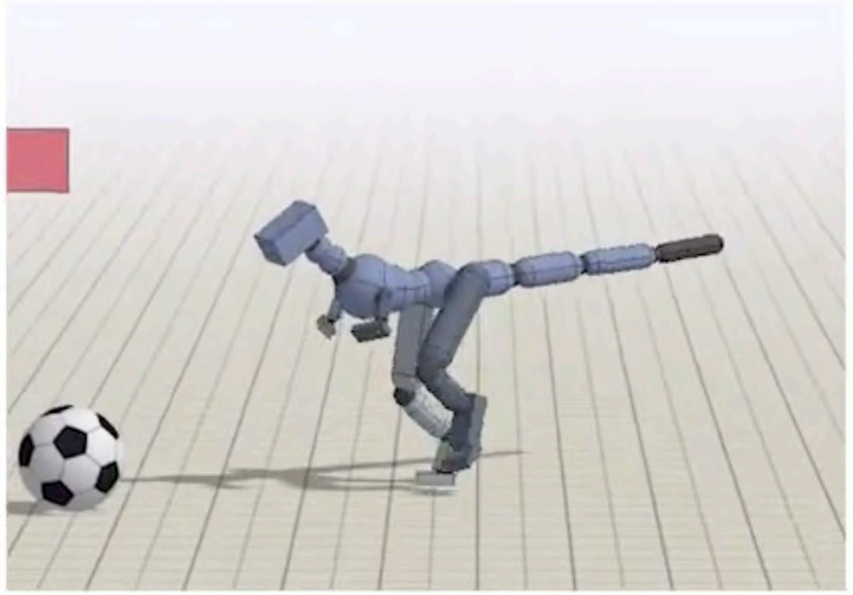
MCP (Ours)

MCP outperforms prior methods on
a number of challenging tasks.

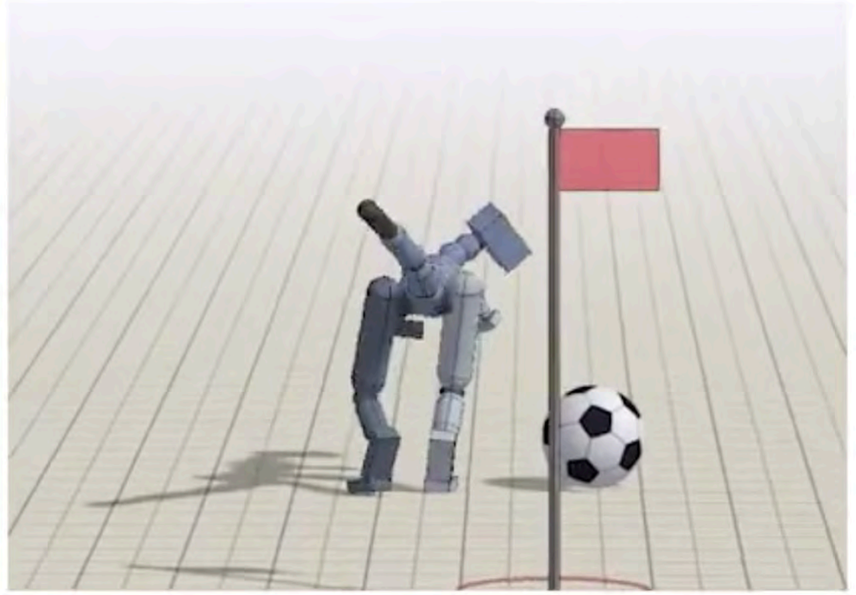
Dribble: T-Rex



Finetune

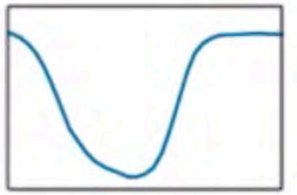


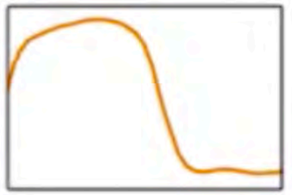
Latent Space
[Merel et al., 2018]

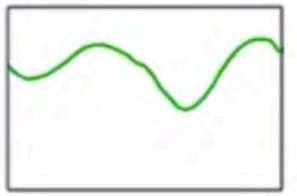


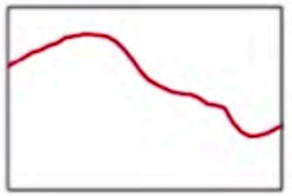
MCP (Ours)

Primitive Specialization







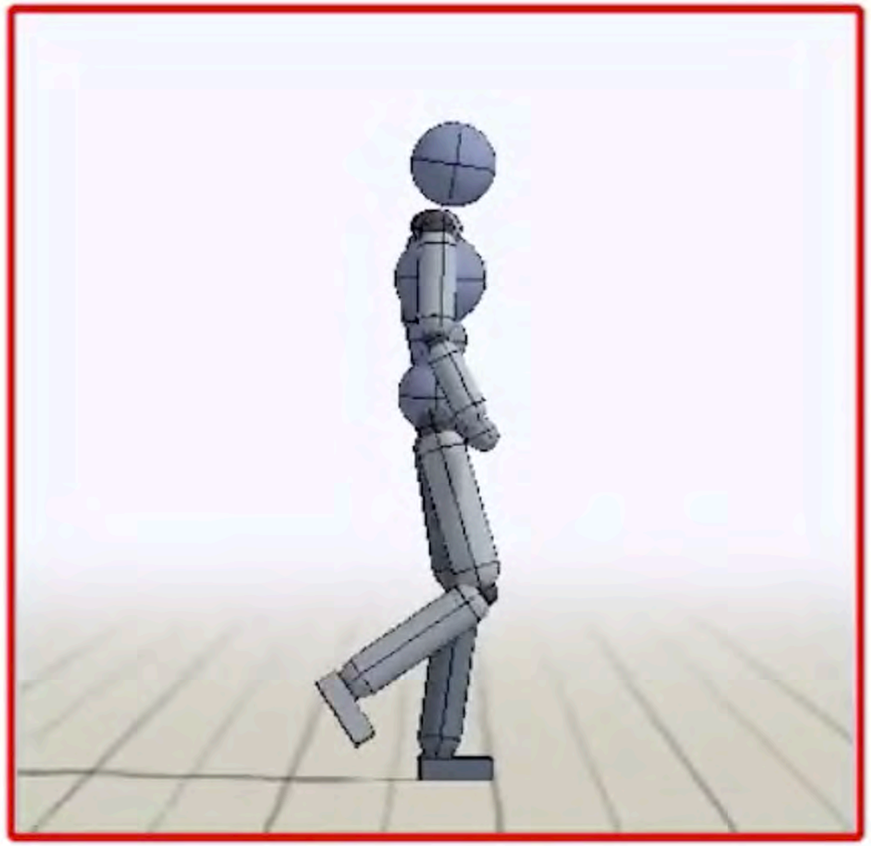


Composite

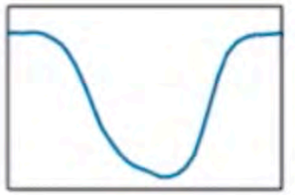
Primitives

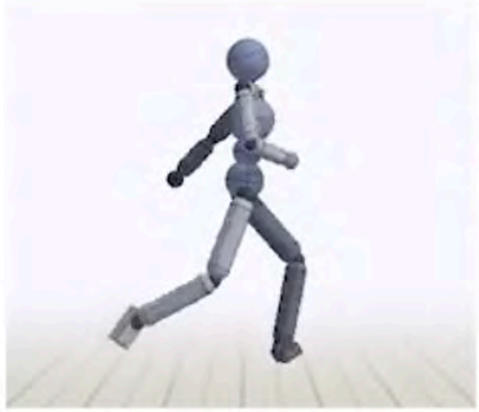
To analyze the specialization of the primitives,

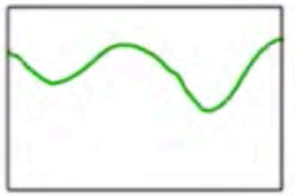
Primitive Specialization

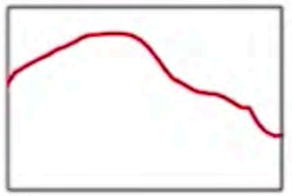


Composite





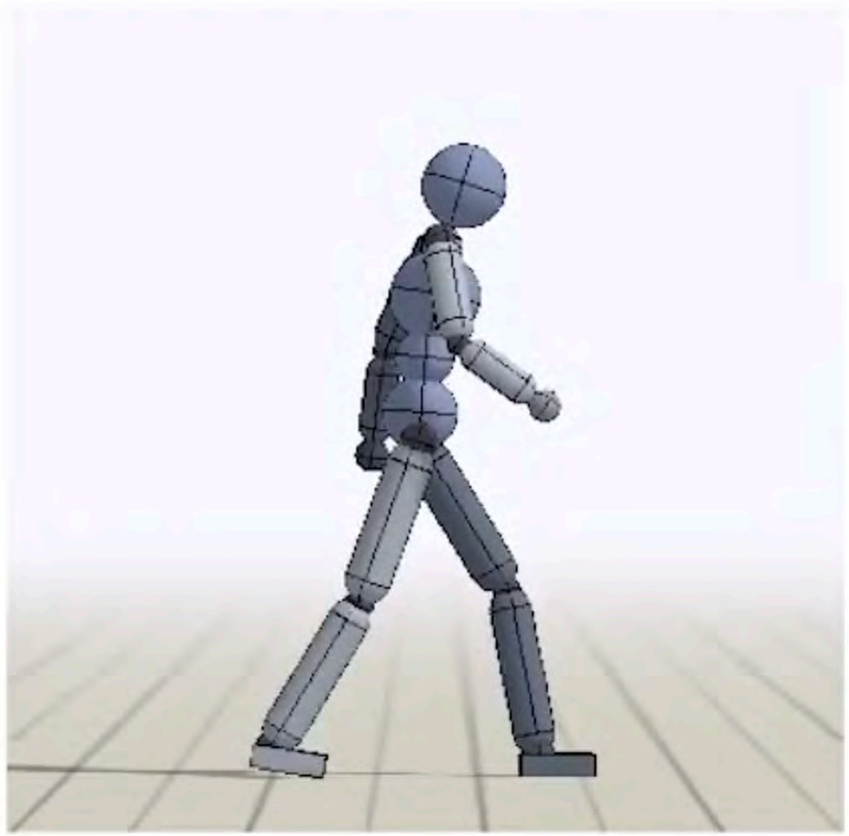




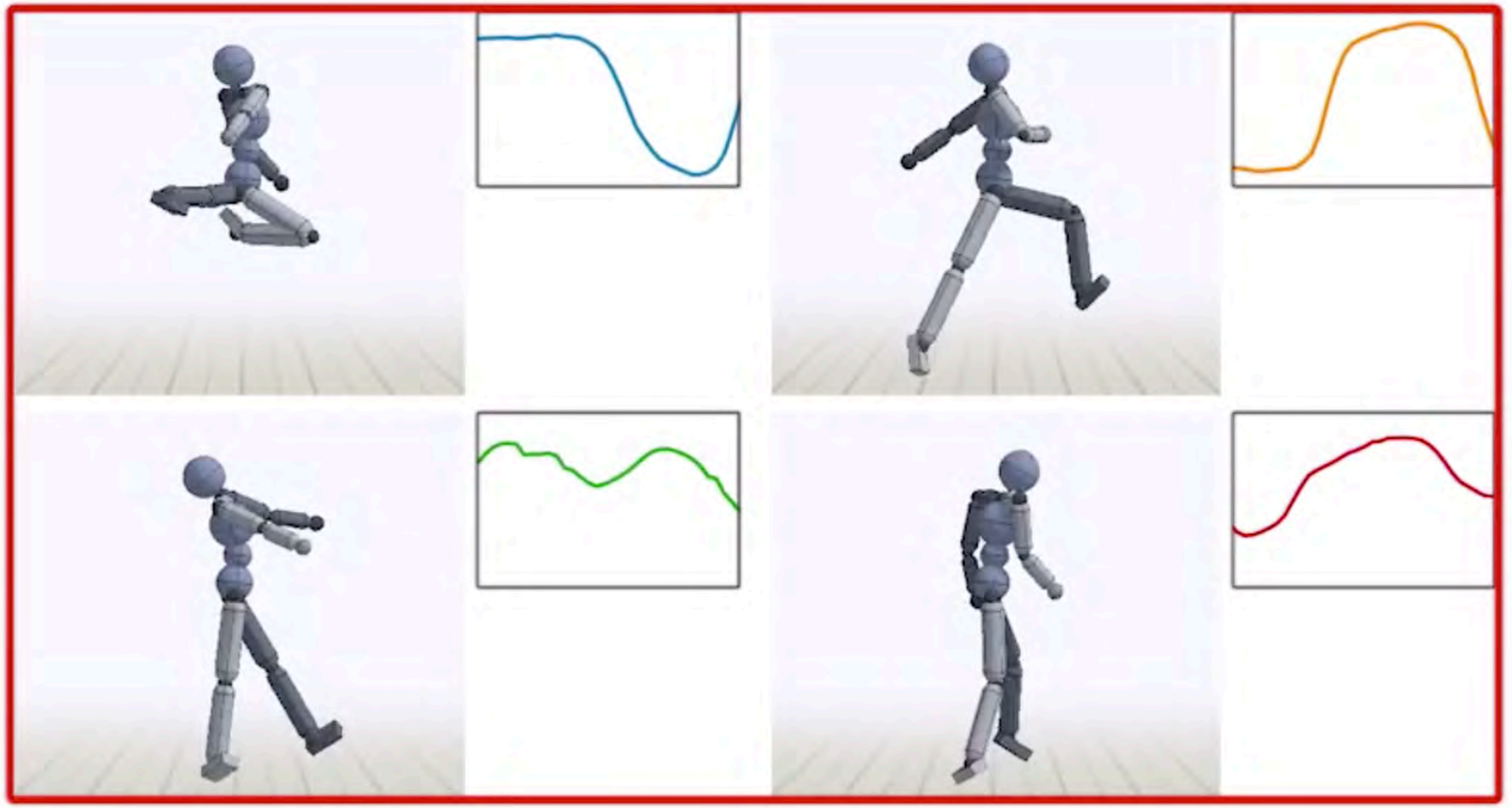
Primitives

we record the actions proposed by each primitive over the course of a walk cycle,

Primitive Specialization



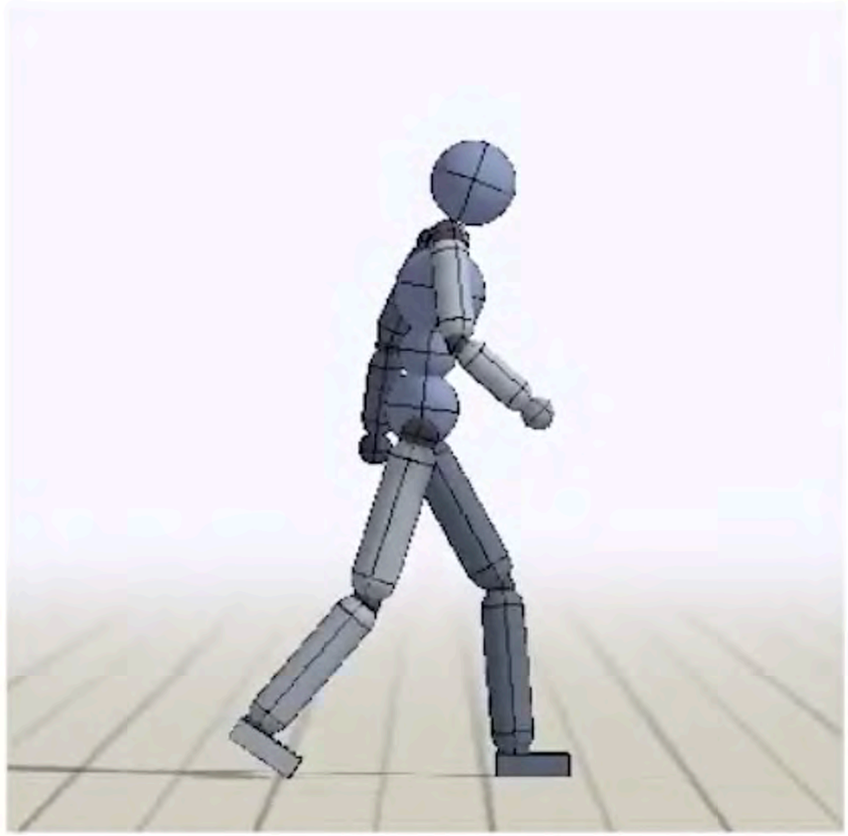
Composite



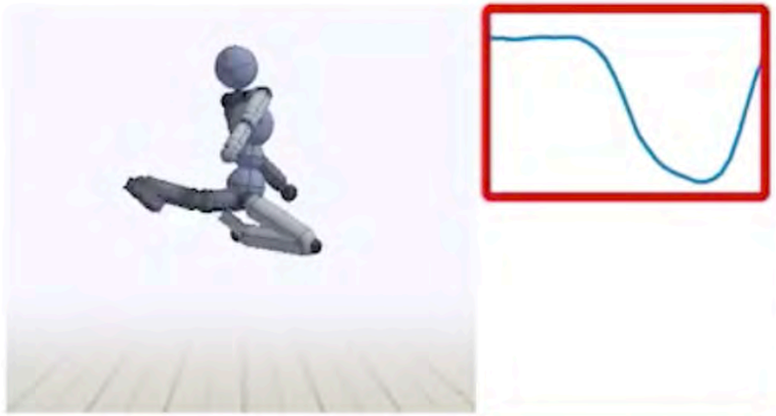
Primitives

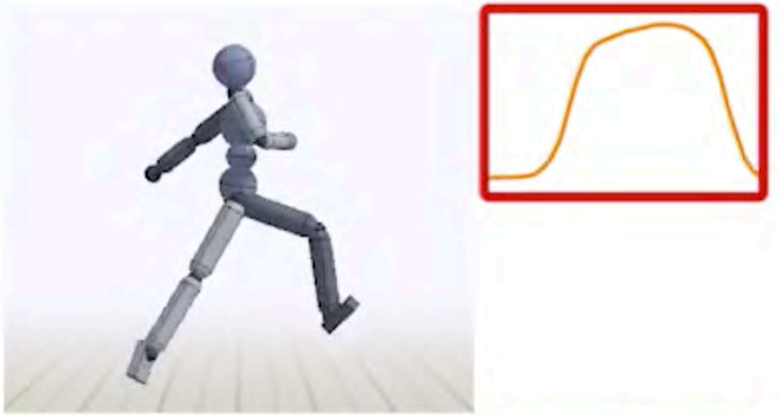
then replay the actions on the simulated characters on the right.

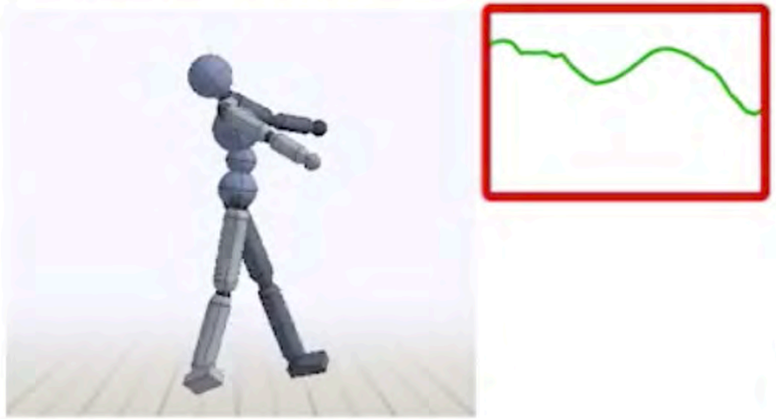
Primitive Specialization



Composite



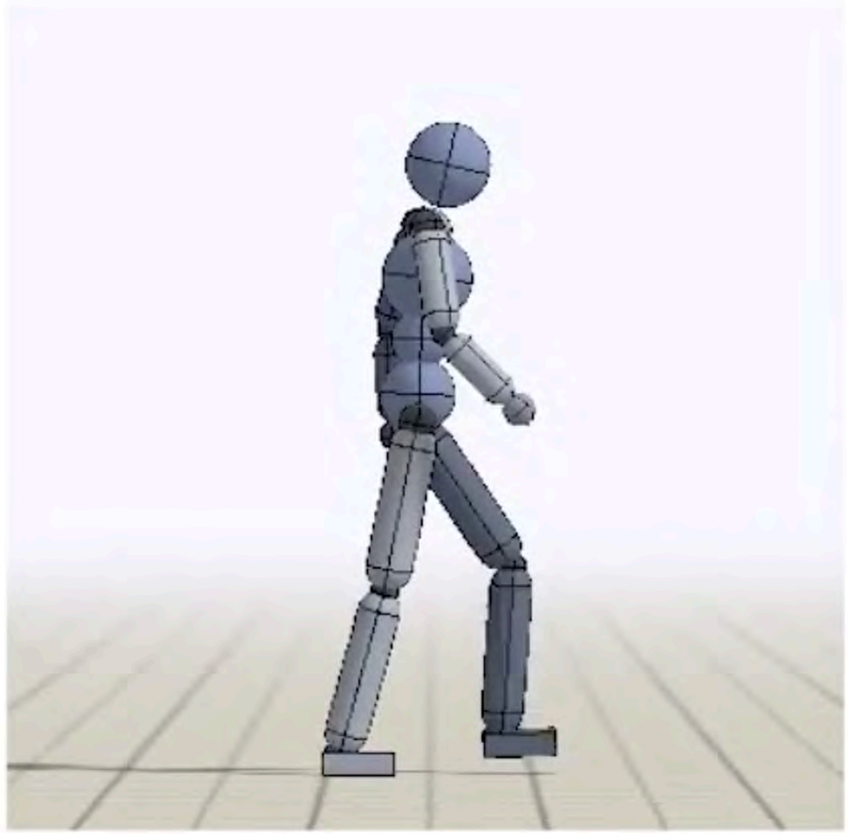




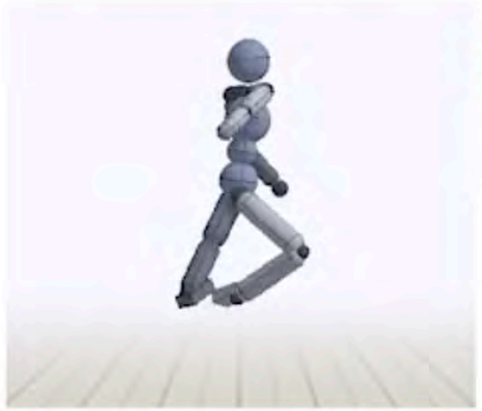
Primitives

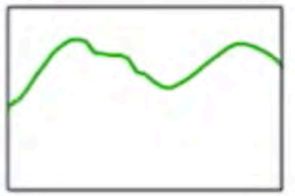
The graphs show the activations of the primitives over time.

Primitive Specialization



Composite

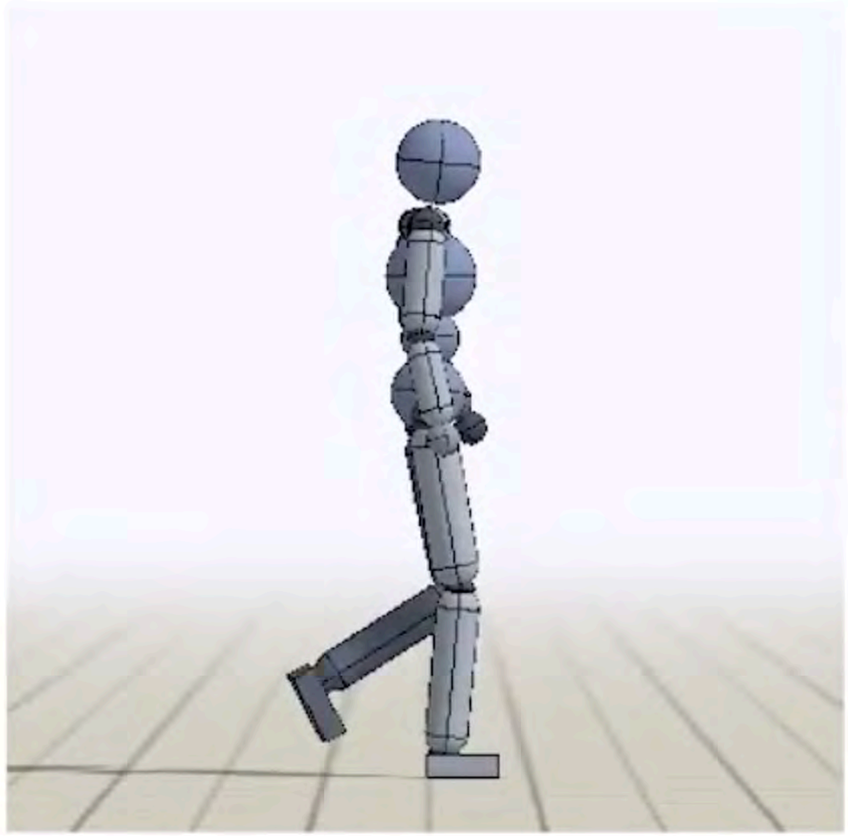




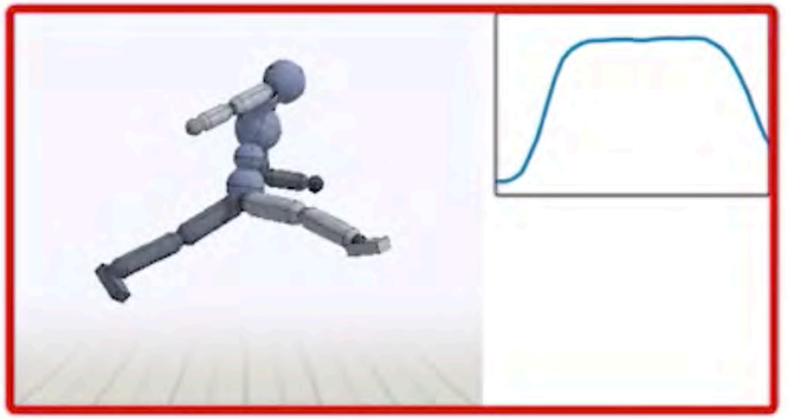
Primitives

The primitives learn to specialize in different phases of a walk cycle.

Primitive Specialization



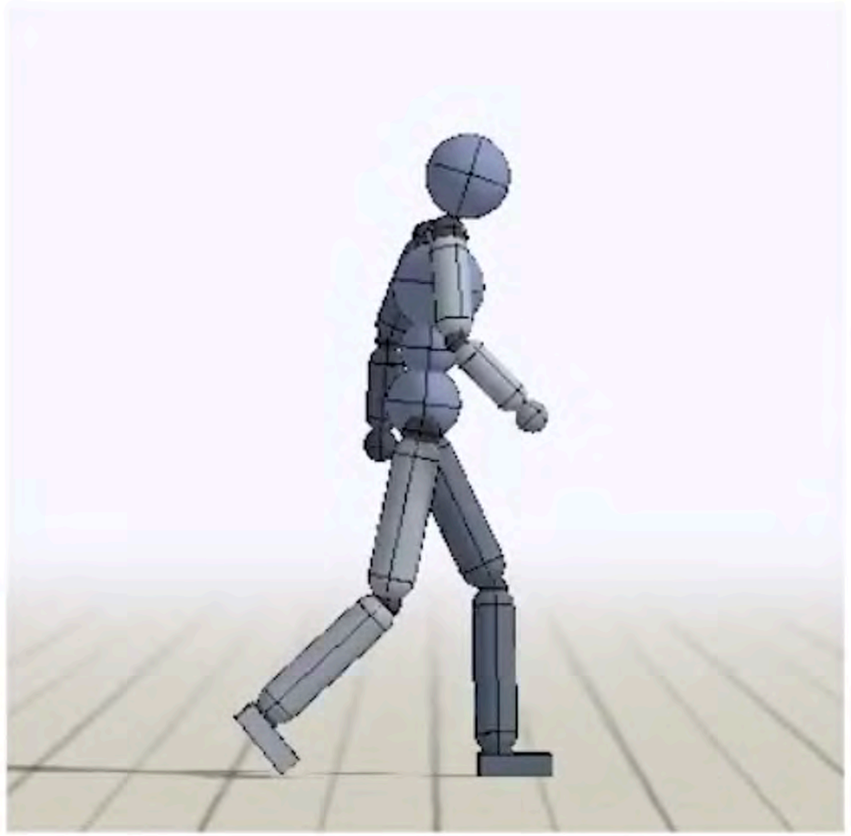
Composite



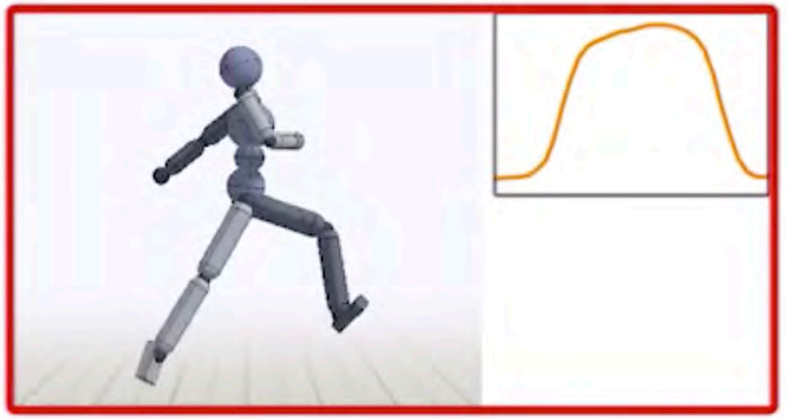
Primitives

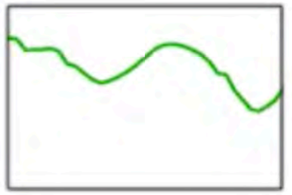
Primitive 1 is most active during left stance,
and less active during right stance.

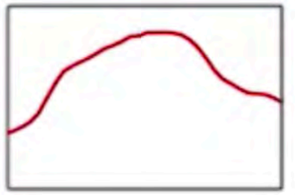
Primitive Specialization



Composite



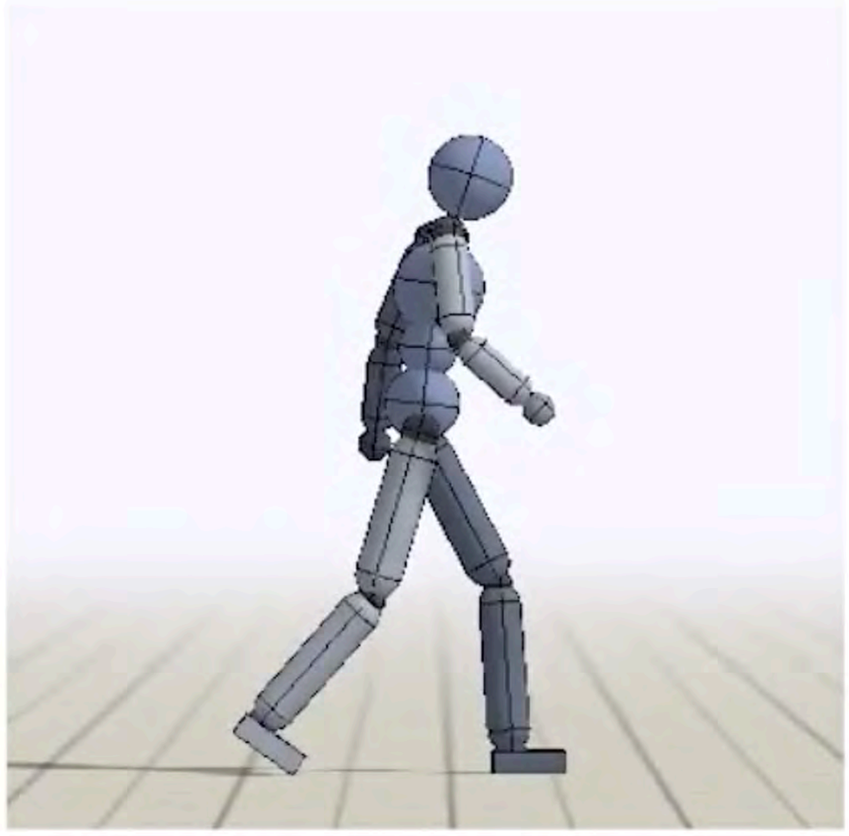




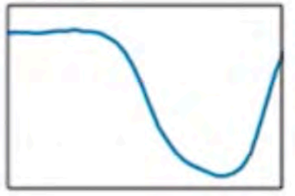
Primitives

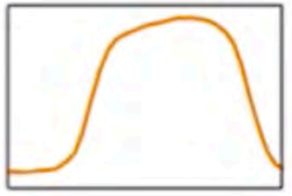
While primitive 2 exhibits the opposite behavior, becoming most active during right stance.

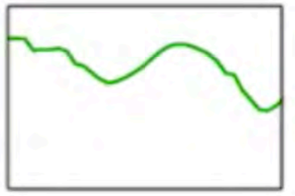
Primitive Specialization

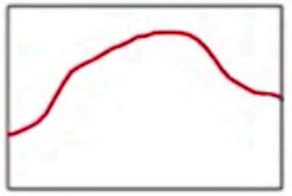


Composite





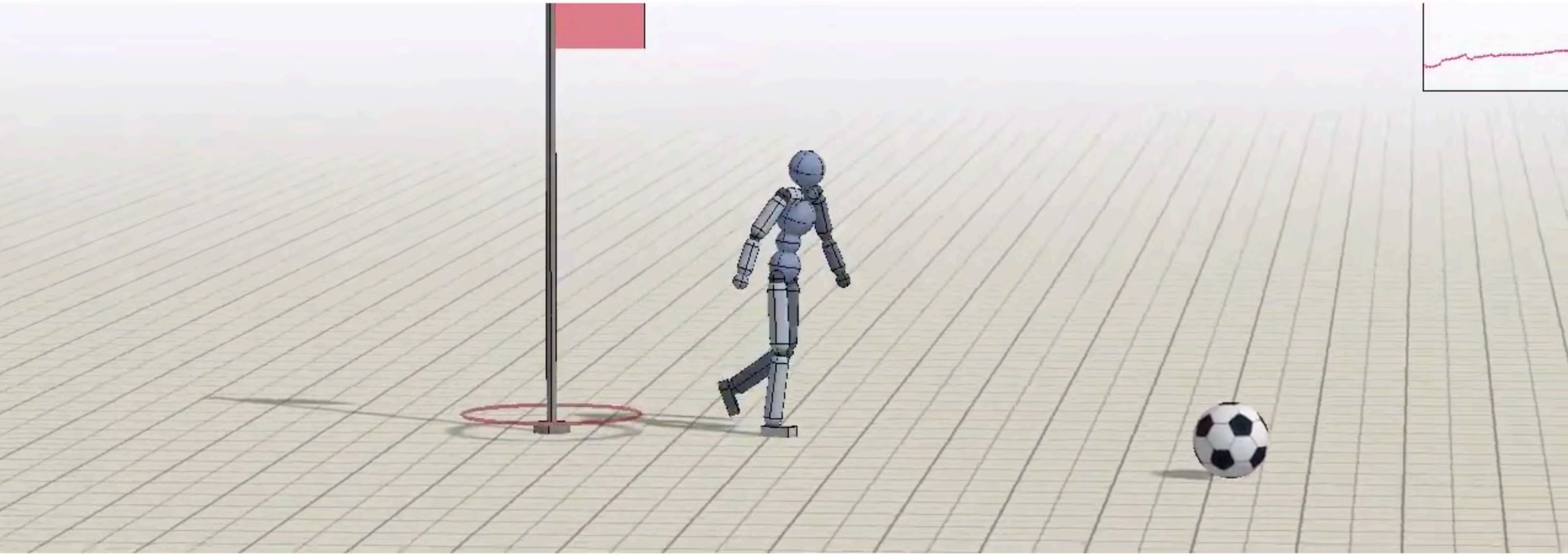




Primitives

Other primitives also exhibit distinct behaviors.

(3x Speedup)



Project page: xbpeng.github.io/projects/MCP/