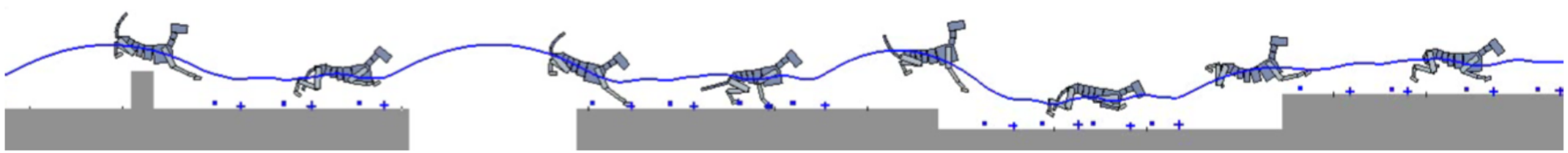
Learning Dynamic Locomotion Skills for Terrains with Obstacles

Xue Bin (Jason) Peng
Glen Berseth
Michiel van de Panne

Department of Computer Science
University of British Columbia



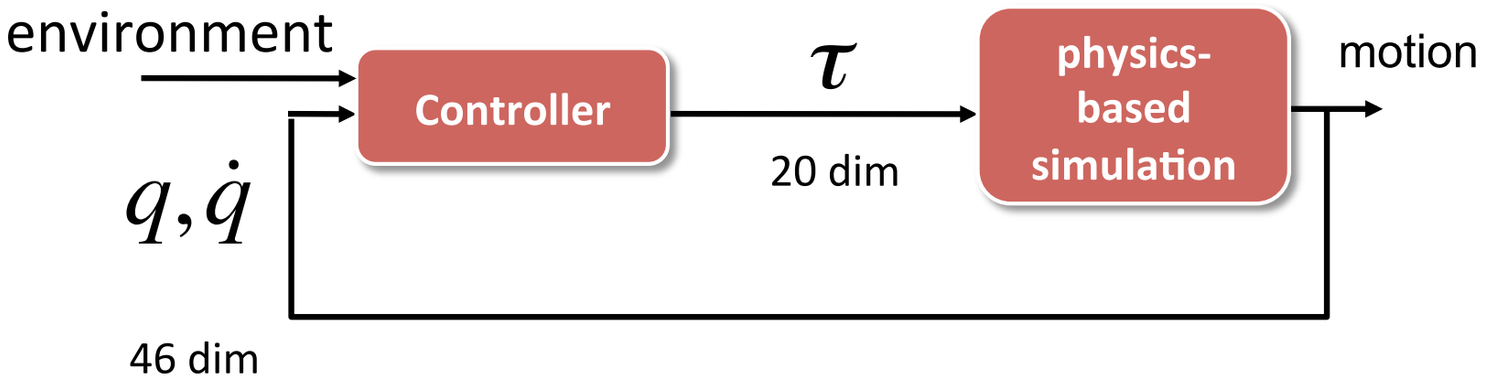
Movement skills

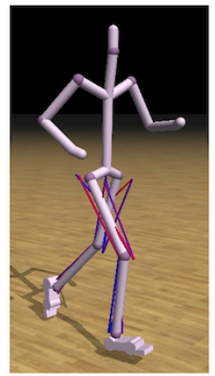
graphics & animation
robotics
motor learning

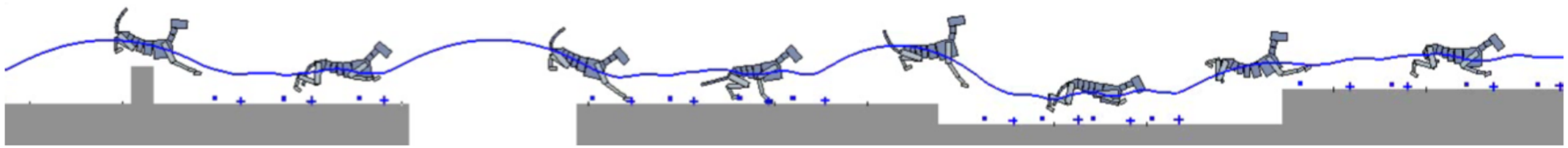
"Can you fly that thing?"

“Not yet …”

Physics-based simulation



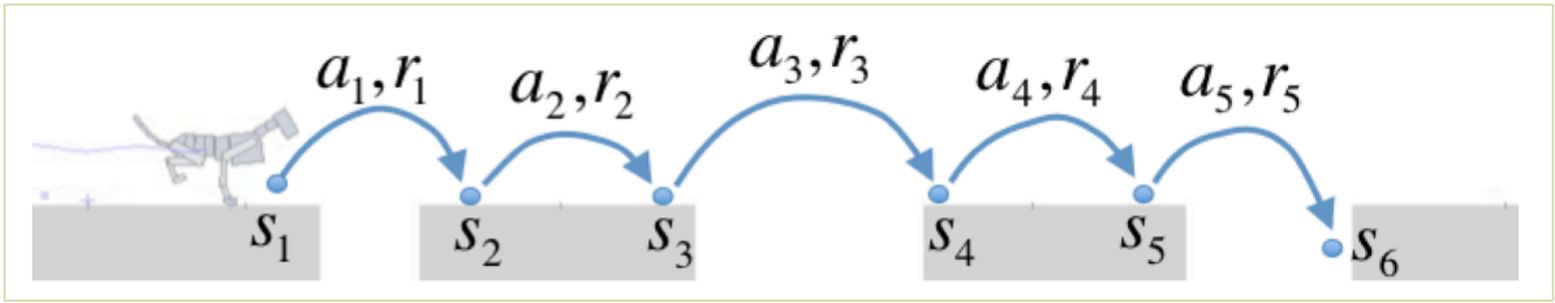




Results

21-link dog

realtime simulation



body + terrain

10 dim

s

High-level control policy π

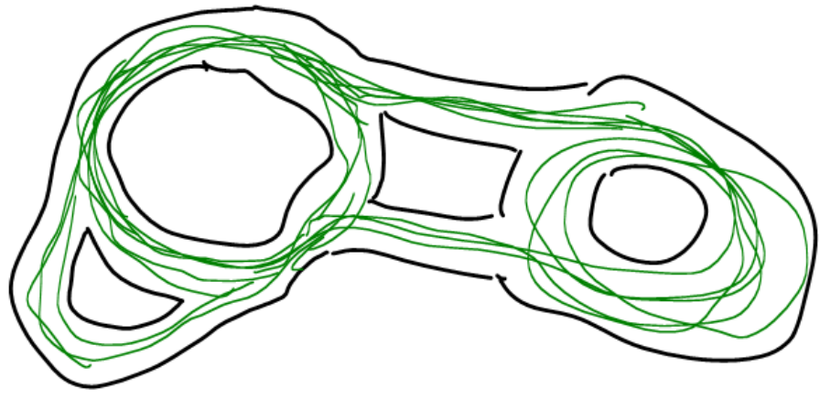
28 dim

a

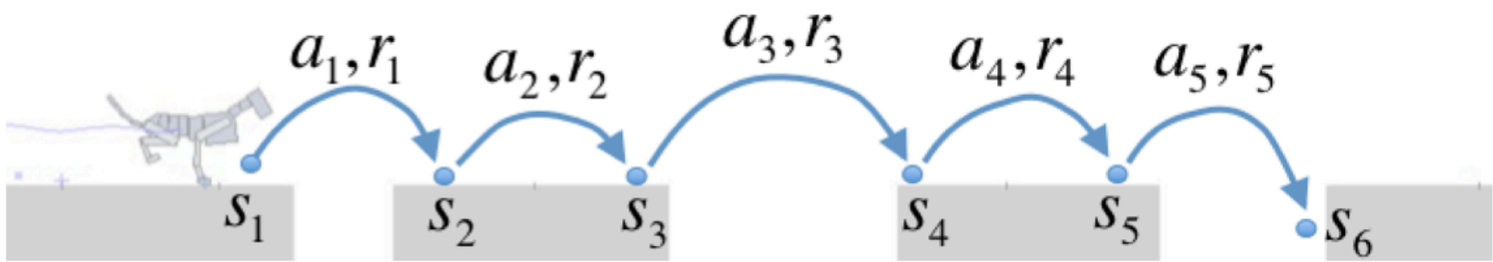
Low-level control

20 dim

τ



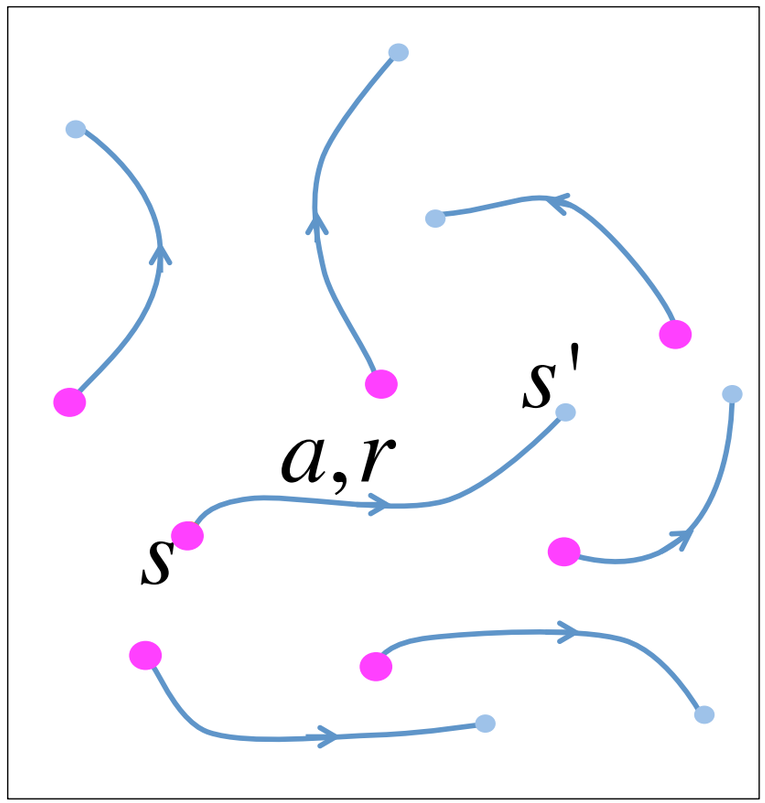
Experience Tuples



$$\mathcal{T} = \{(s_i, a_i, r_i, s'_i)\}$$

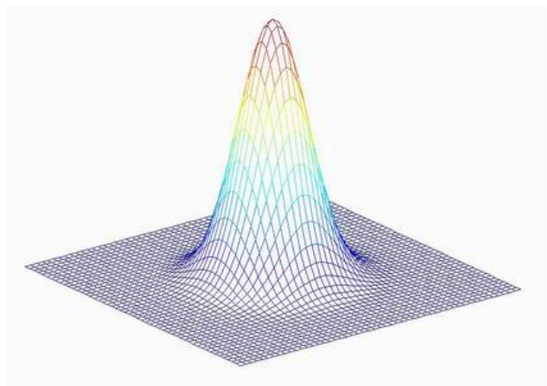
Conceptual View

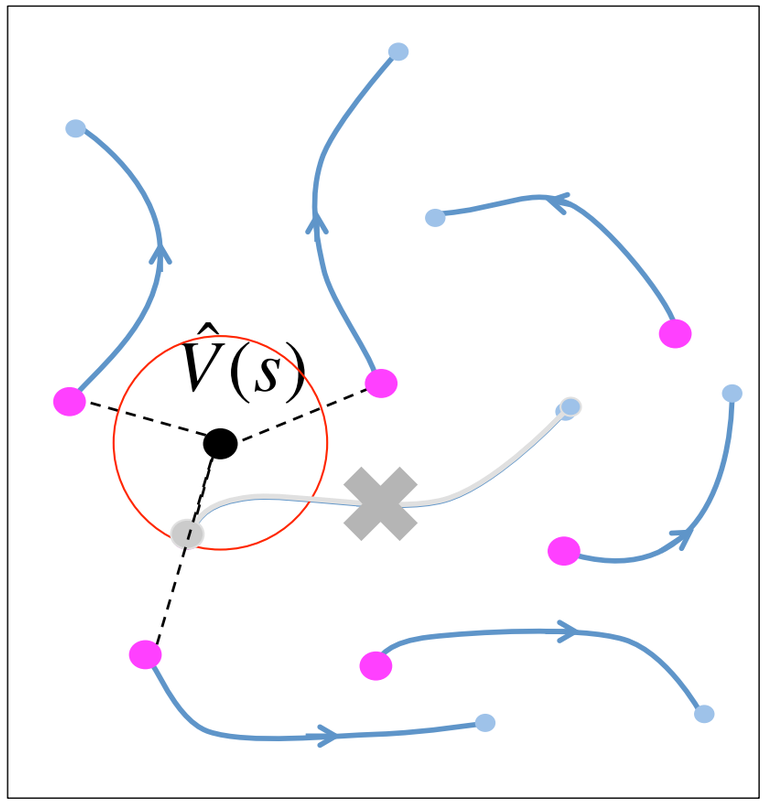
$$V^*(s) = \max_a \{r(s,a) + \gamma V^*(s')\}$$



Value function approximation

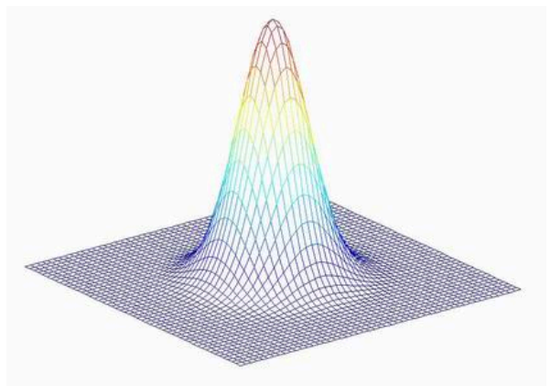
kNN interpolation for $\hat{V}(s)$

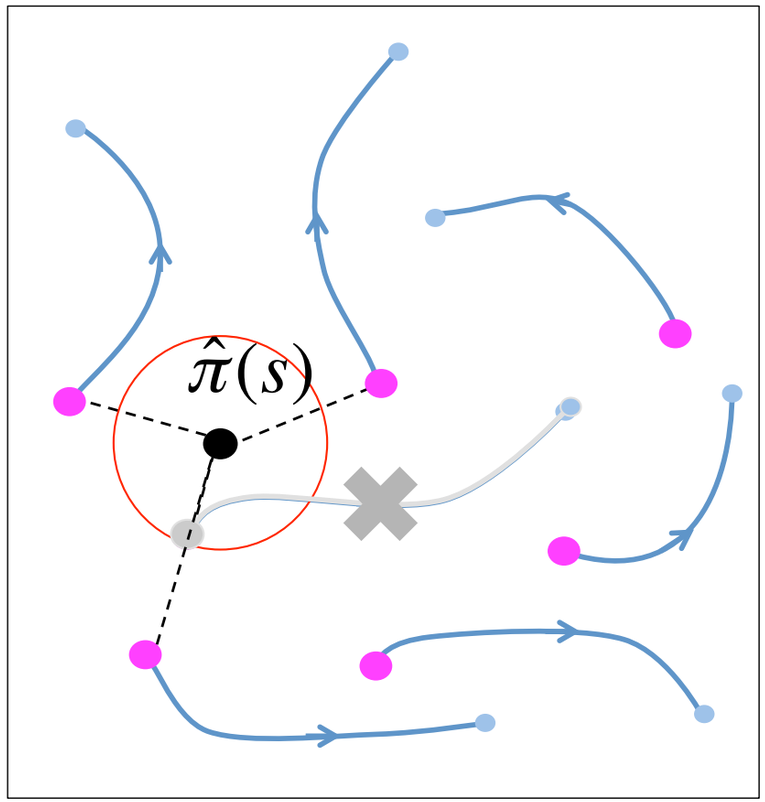




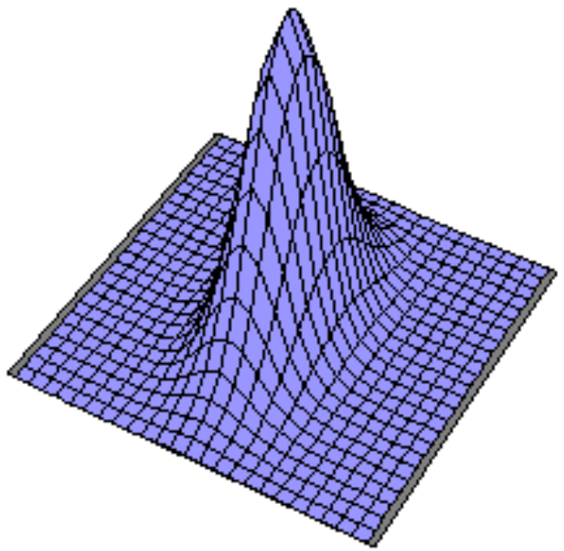
Policy approximation

kNN interpolation for $\hat{\pi}(s)$





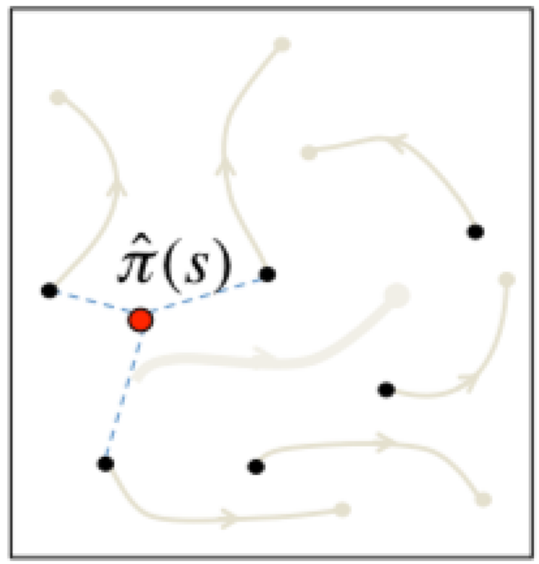
Distance Metric Learning

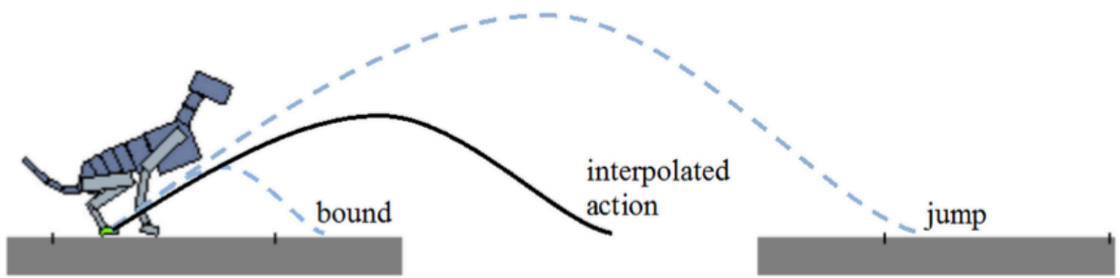


feature	uniform	gaps	steps	walls	mixed
d_0	0.1	0.168	0.270	0.198	0.192
h_0	0.1	0.068	0.263	0.017	0.068
d_1	0.1	0.163	0.004	0.165	0.163
h_1	0.1	0.156	0.006	0.118	0.274
d_2	0.1	0.026	0.058	0.002	0.068
$v_{\mathrm{com,x}}$	0.1	0.118	0.083	0.024	0.002
d_{front}	0.1	0.039	0.127	0.151	0.020
$p_{\mathrm{com,x}}$	0.1	0.083	0.081	0.132	0.074
$p_{\mathrm{com,y}}$	0.1	0.057	0.035	0.105	0.071
θ_{torso}	0.1	0.122	0.073	0.088	0.068
σ_λ^2	0.2	0.200	0.260	0.250	0.212

Table 4: *Distance metric weights computed for the dog scenarios.*

Outlier Removal

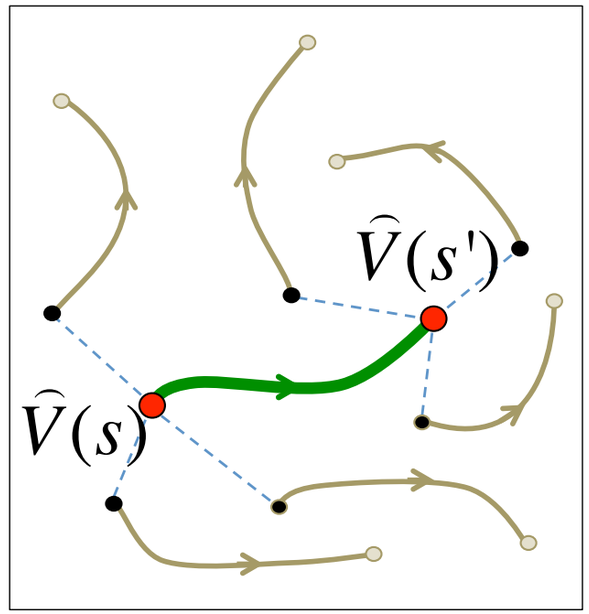




Value iteration using
positive temporal difference updates

Does the tuple contribute
to an improvement in V(s)?

$$\delta = r + \gamma \hat{V}(s') - \hat{V}(s)$$



Value iteration using
positive temporal difference updates

for each tuple:

$$\delta = r + \gamma \hat{V}(s') - \hat{V}(s)$$

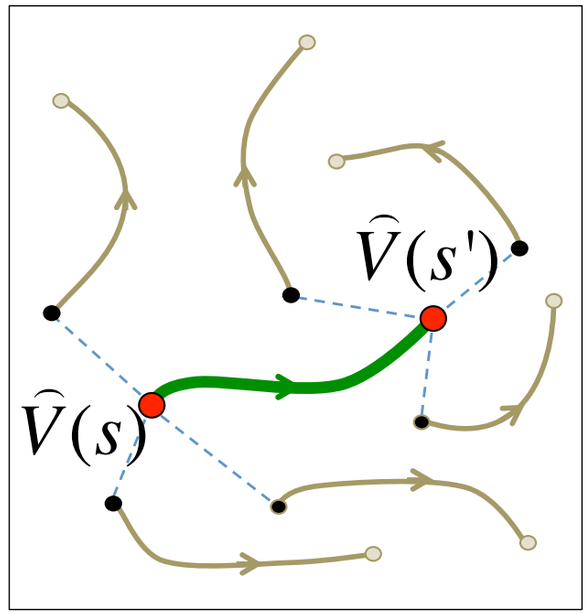
If $\delta > 0$

$$V(s) \leftarrow V(s) + \alpha\delta$$

tag as “winning” tuple

else

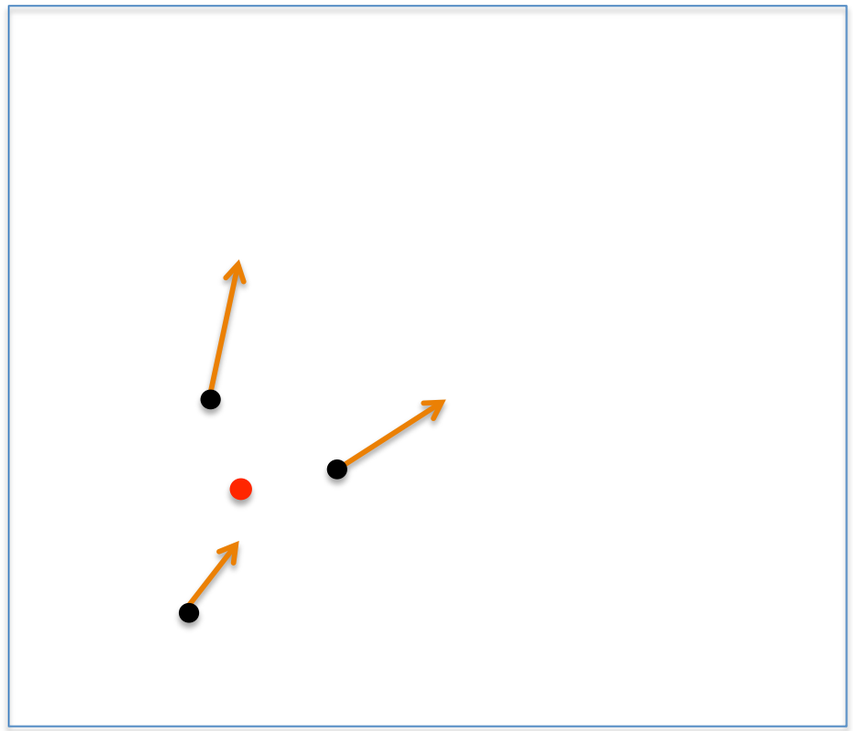
tag as “losing” tuple

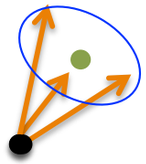


Local Exploration

$\mathcal{E}$-greedy

Probability	Action
$1-\epsilon$	exploit policy
$\epsilon(1-\alpha)$	local exploration
$\epsilon\alpha$	global exploration

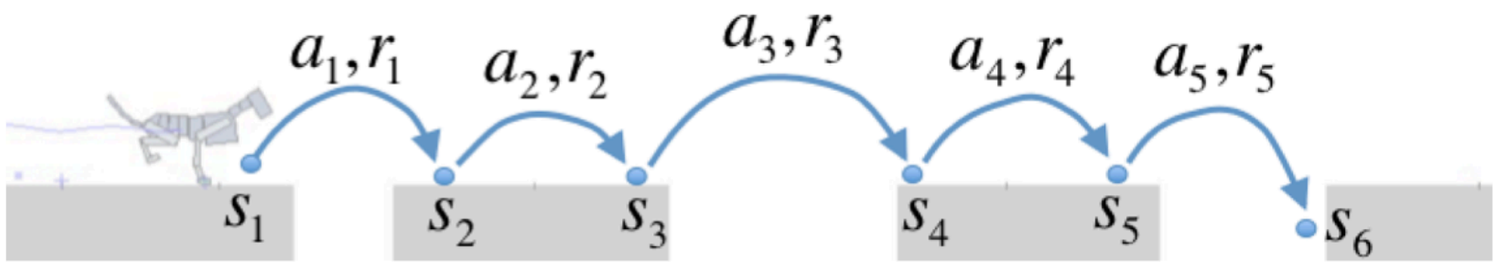




C

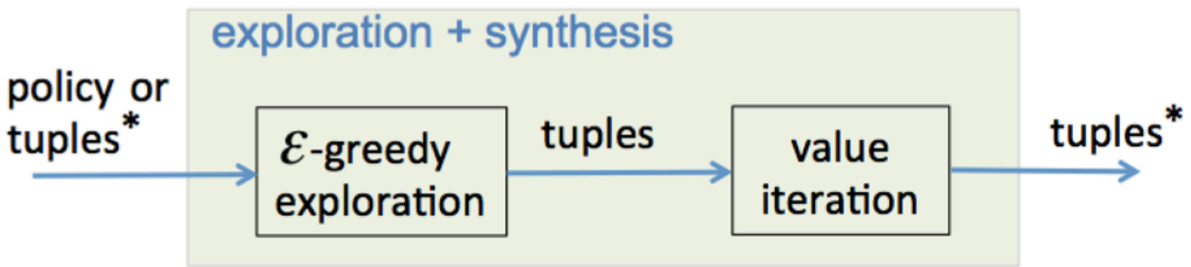
$$a' = a + D\,(C + \gamma I)\,N(0, \sigma)$$

Learning Pipeline Overview

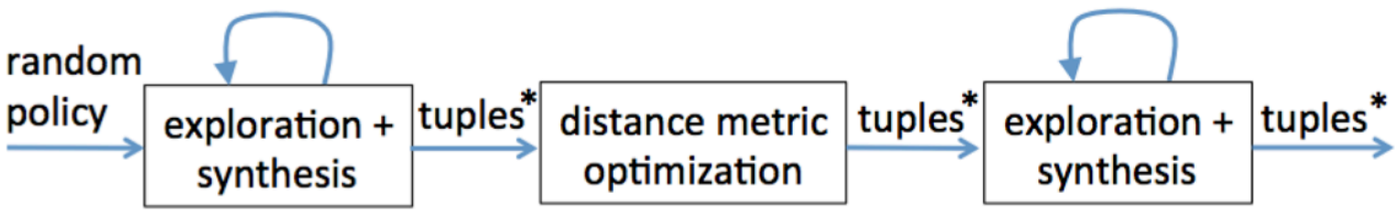


$$\mathcal{T} = \{(s_i, a_i, r_i, s_i')\}$$

1 M ⟶ 250 k



Learning Pipeline



Rewards

$$R_c = \begin{cases} 0 & : \text{ character falls} \\ 0.5 + w_v E_v + w_e E_e + w_s E_s & : \text{ otherwise} \end{cases}$$

States

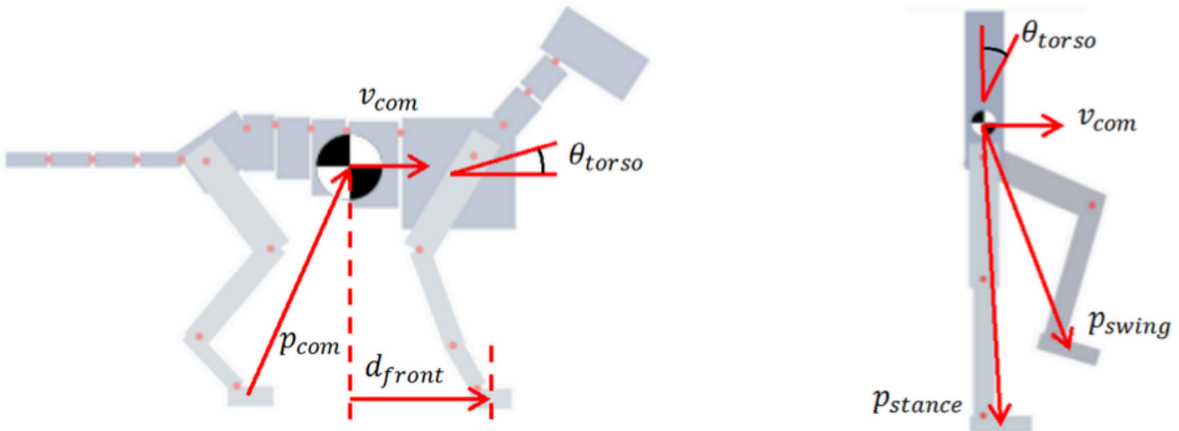


Figure 11: *State features for the dog and the biped.*

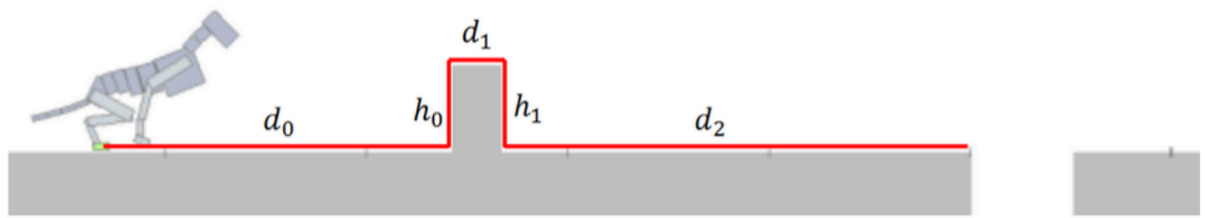
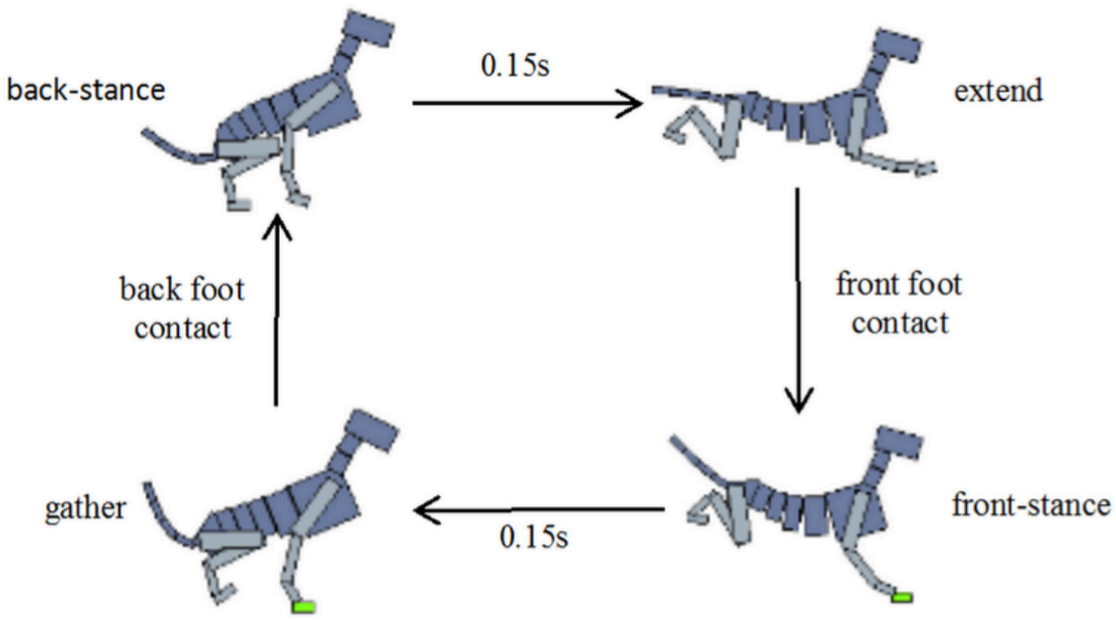


Figure 12: *Terrain features used in the state description.*

Actions


back-stance
0.15s
extend
front foot
contact
front-stance
0.15s
gather
back foot
contact

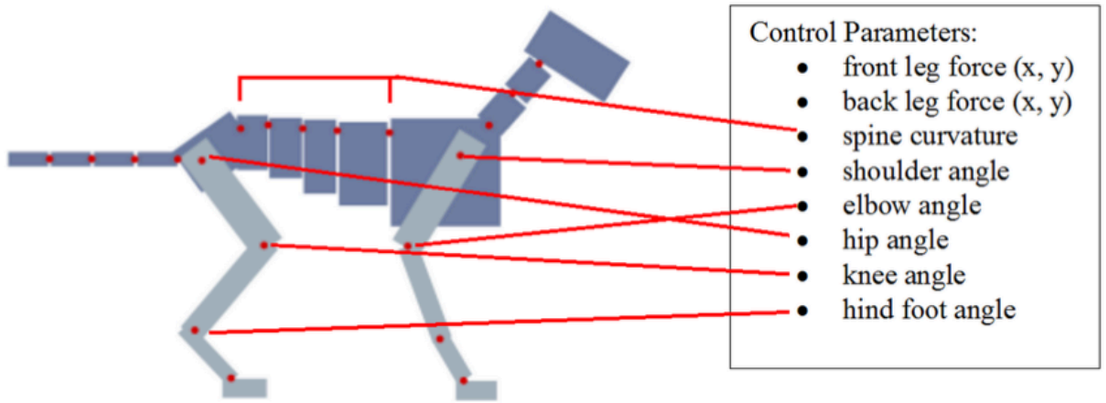
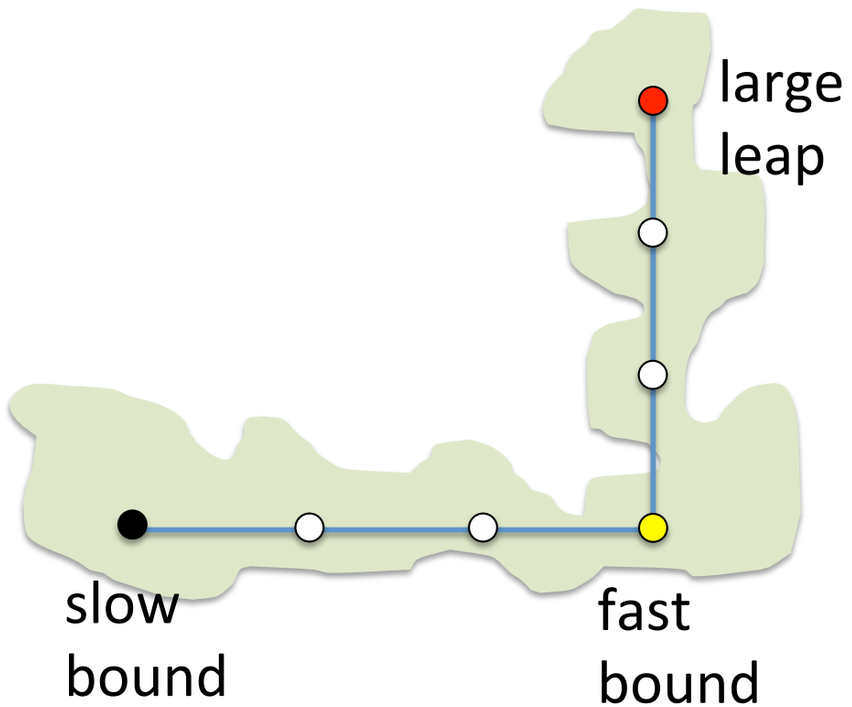


Figure 7: *Control parameters for the dog.*

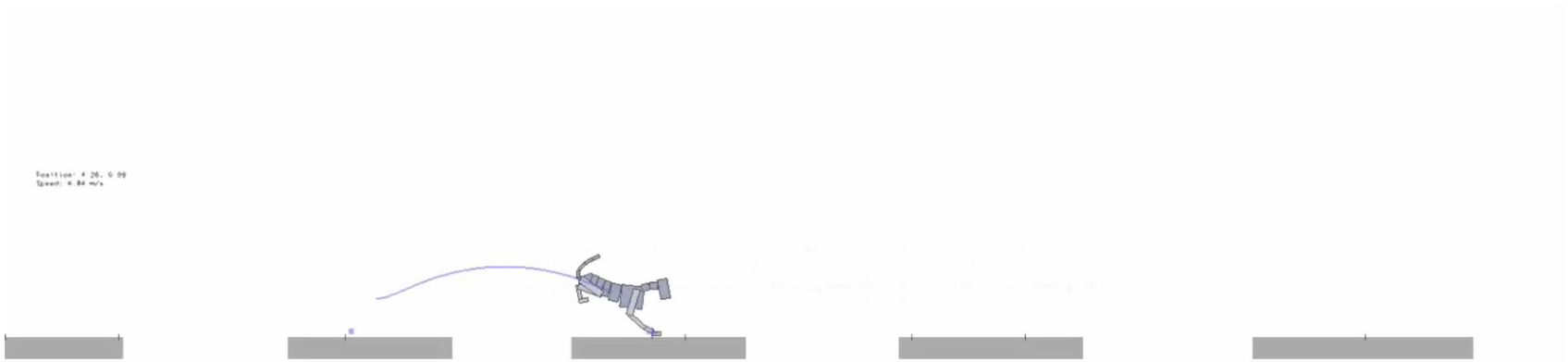
Action			
back-stance:	extend:	front-stance:	gather:
• back leg force (x, y) • spine curvature • shoulder angle • elbow angle • hip angle • hock angle • hind foot angle	• spine curvature • shoulder angle • elbow angle • hip angle • hock angle • hind foot angle	• front leg force (x, y) • spine curvature • shoulder angle • elbow angle • hip angle • hock angle • hind foot angle	• spine curvature • shoulder angle • elbow angle • hip angle • hock angle • hind foot angle

Figure 8: *Dog Action Parameters.*

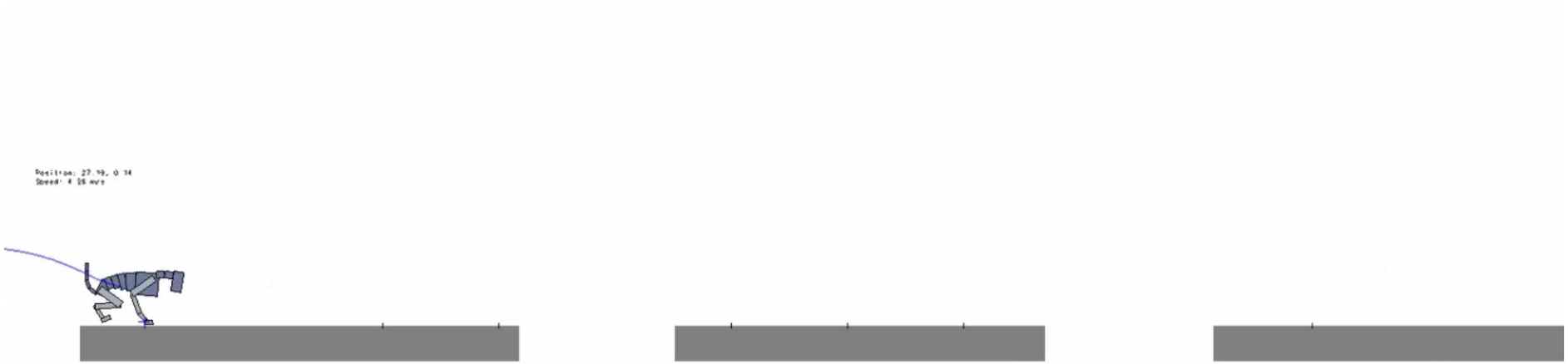
Actions



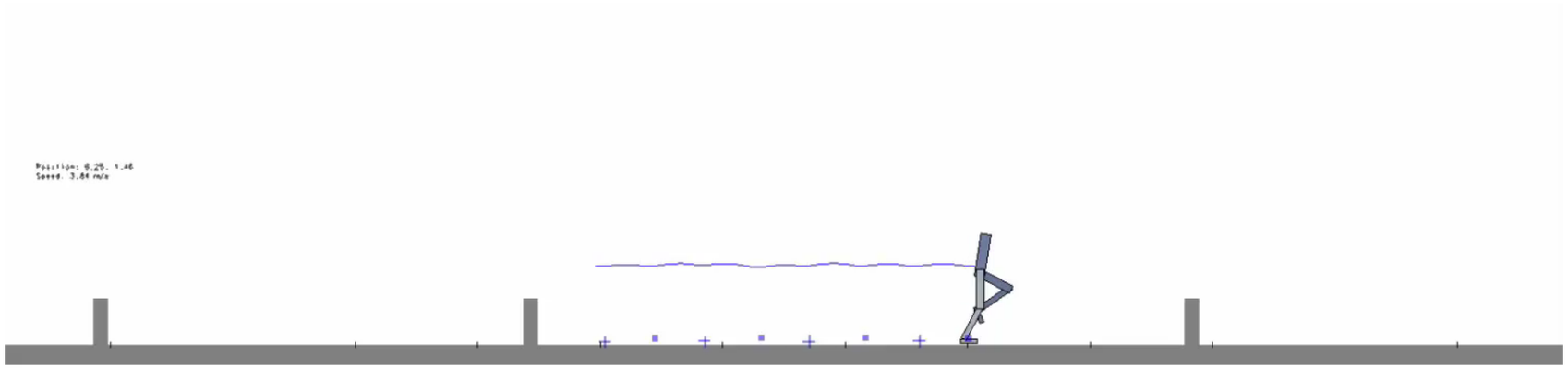
Initial random-action policy



tight gaps



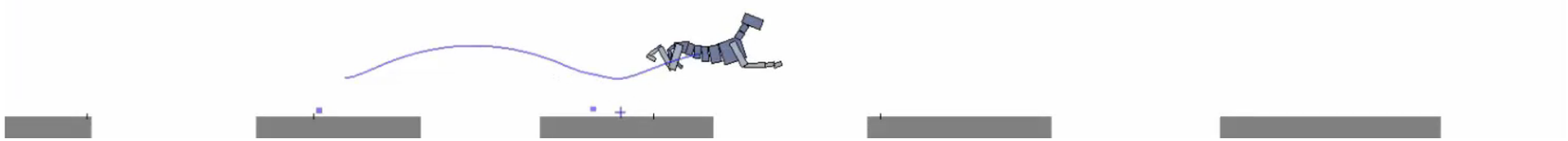
mixed



big walls

Continuous action space

Discrete action space



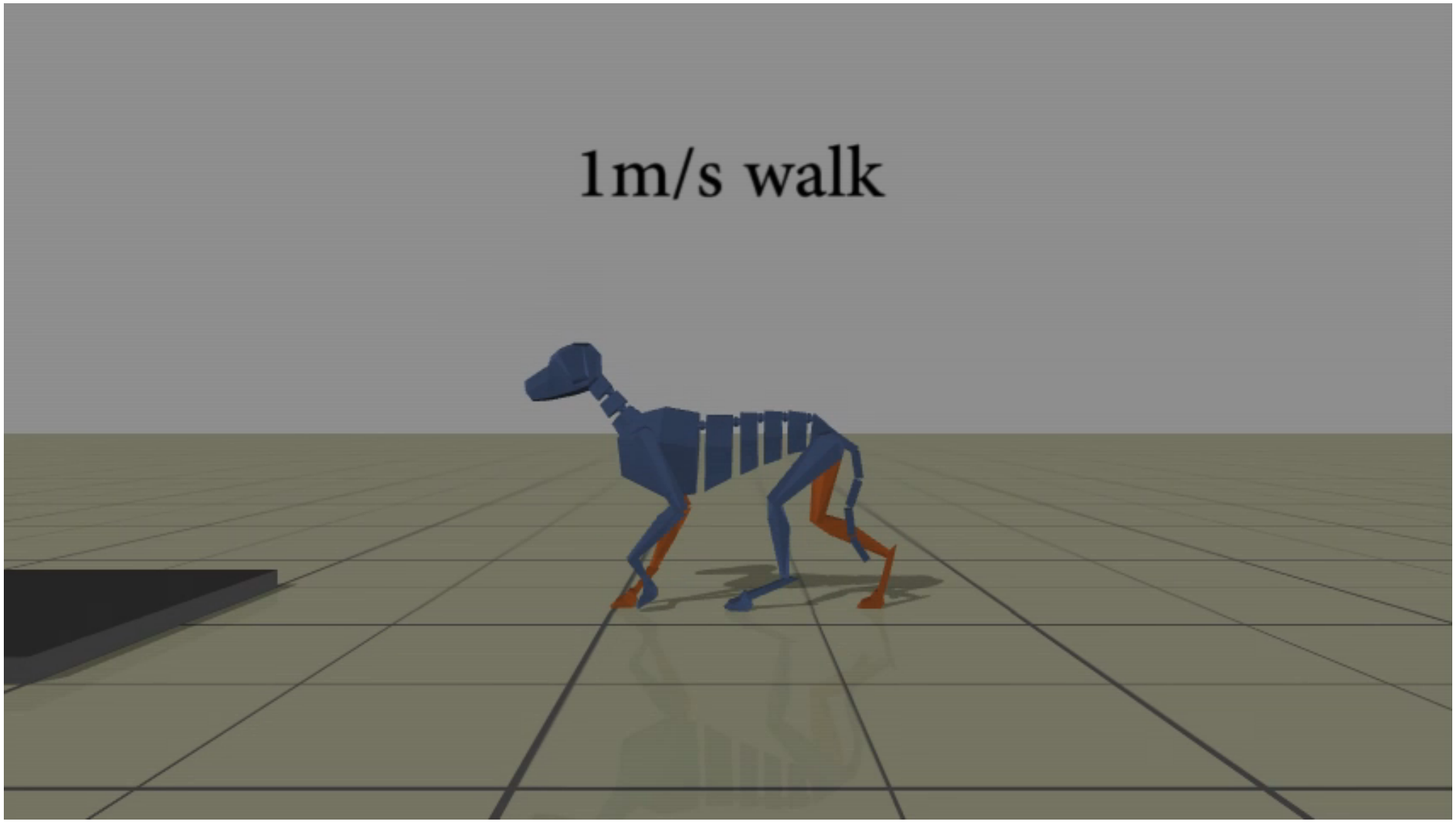
Comparison of action selection methods

regular algorithm

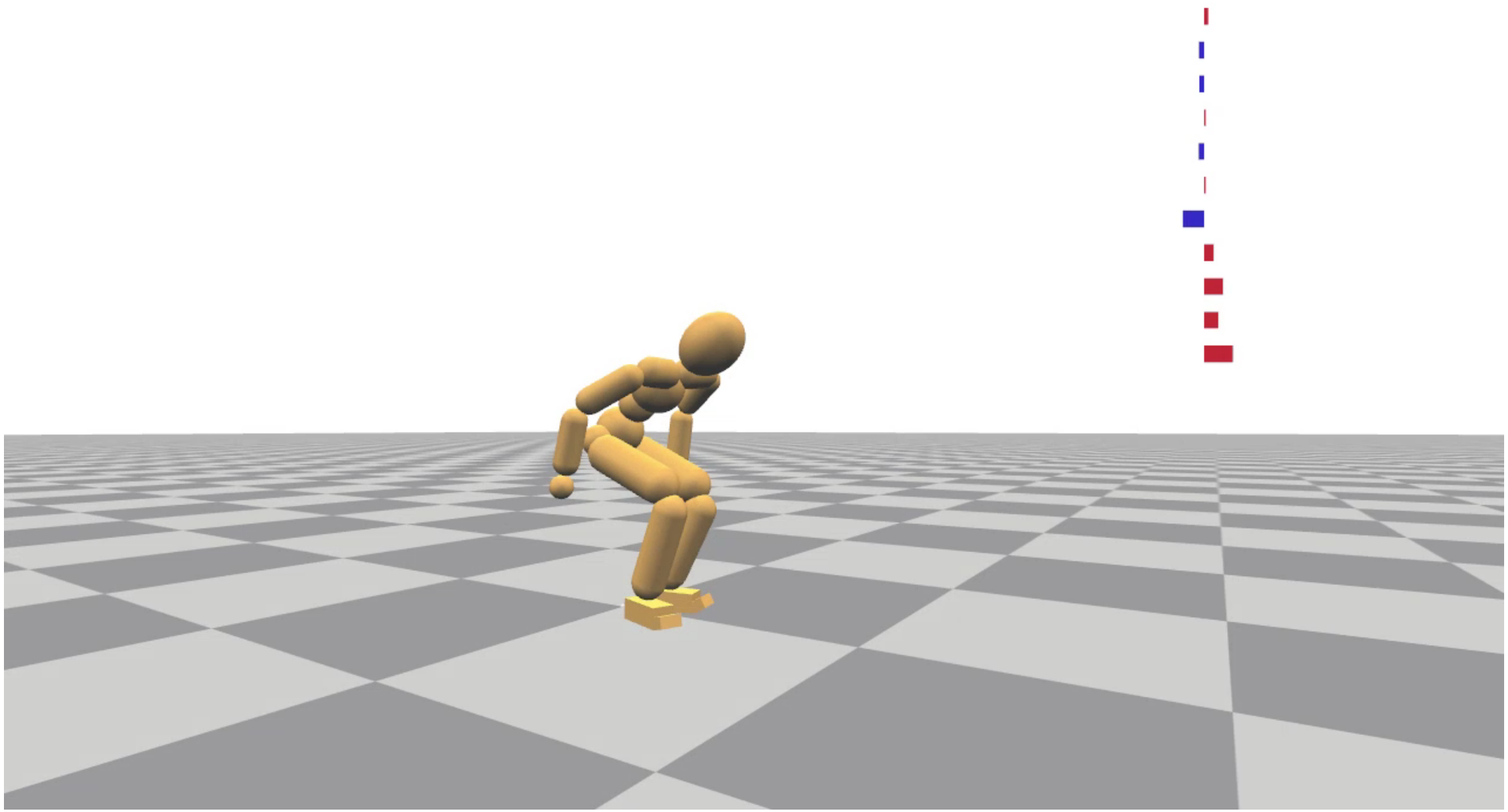
algorithm without outlier removal

algorithm only selects nearest neighbour

Towards 3D: Quadrupeds



Towards 3D: bipeds



Key Features

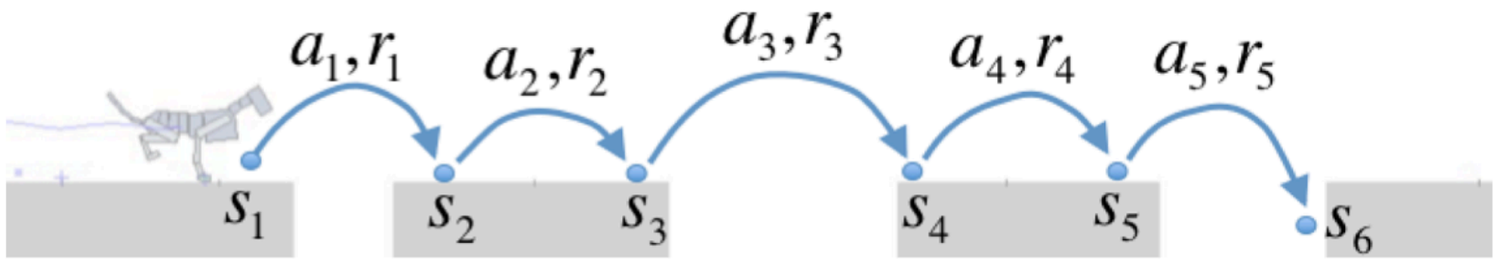
- "shallow" fn approximation; get the details right
  - non-parametric kernel-based approximation
  - outlier removal
  - distance metric optimization
- abstracted actions
- local action exploration
- value iteration using batched positive TD
- devote resources to what works

  Q(s,a) vs V(s), π(s) $\mathcal{T} = \{(s_i, a_i, r_i, s'_i)\}$

Future Work

- curriculum-based learning, transfer learning
- reacting vs online model-based planning
- defining task difficulty ?

Questions ?



- Poster M15
- *Dynamic Terrain Traversal Skills Using Reinforcement Learning*
  ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2015)