Reconnaissance d’objets et vision artificielle 2020

Learning visual representations for robotics

Ivan Laptev, INRIA

Announcements

- **Final Project Presentations**
  - **When**: Jan 14 (Thrs): 10:30-16:00
    Jan 15 (Fri): 10:30-16:00
  - **Where**: if possible: Inria Paris research center, 2 Rue Simone IFF, 75012, Paris. Otherwise: online.
  - **Schedule:** To be announced on the course web-page

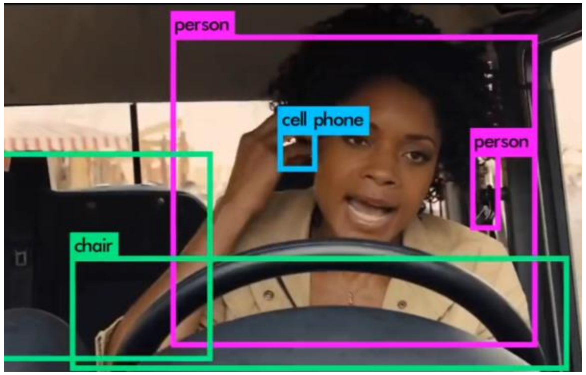
- **Final Projects**:
  - 110 FP proposals received
  - Google Cloud credits 50$ (+50$ on request)

- **Internship topics**:
  - Next lecture
  - Will appear on the calss web-page
  - Apply be emailing ivan.laptev@inria.fr attach CV + reference letters

This course so far…

Image classification

Object detection



Human pose estimation

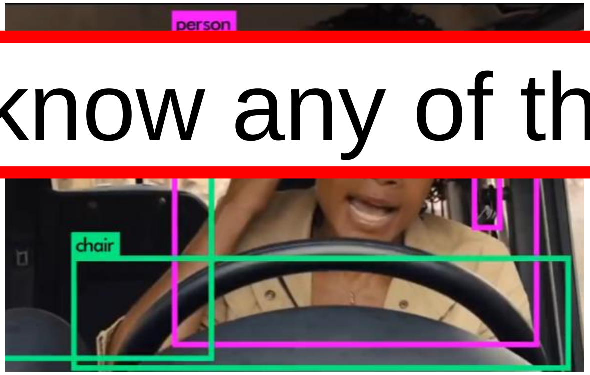
Reconstructing hands and objects

Learning tasks from instructional videos

This course so far…

Image classification

Object detection



Human pose estimation

How do we know any of this is useful?

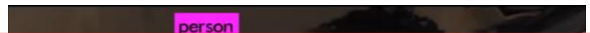
Reconstructing hands and objects

Learning tasks from instructional videos

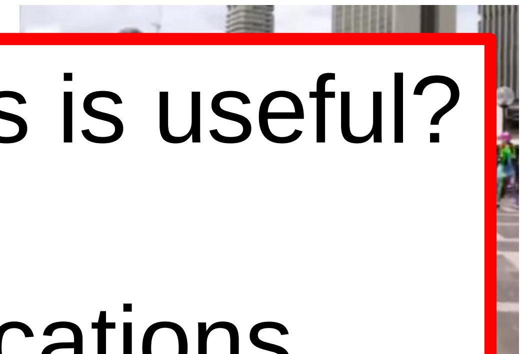
This course so far…

Image classification

Object detection



Human pose estimation

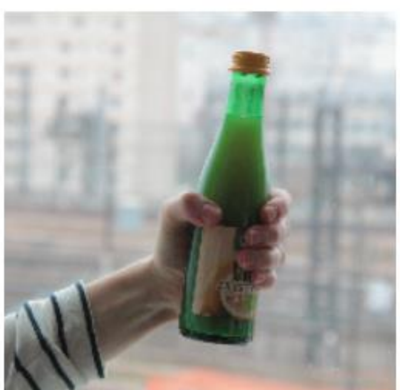


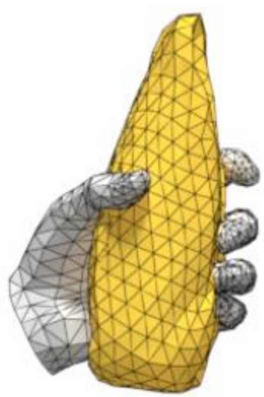
How do we know any of this is useful?

…

Let's connect it to applications

Reconstructing hands and objects



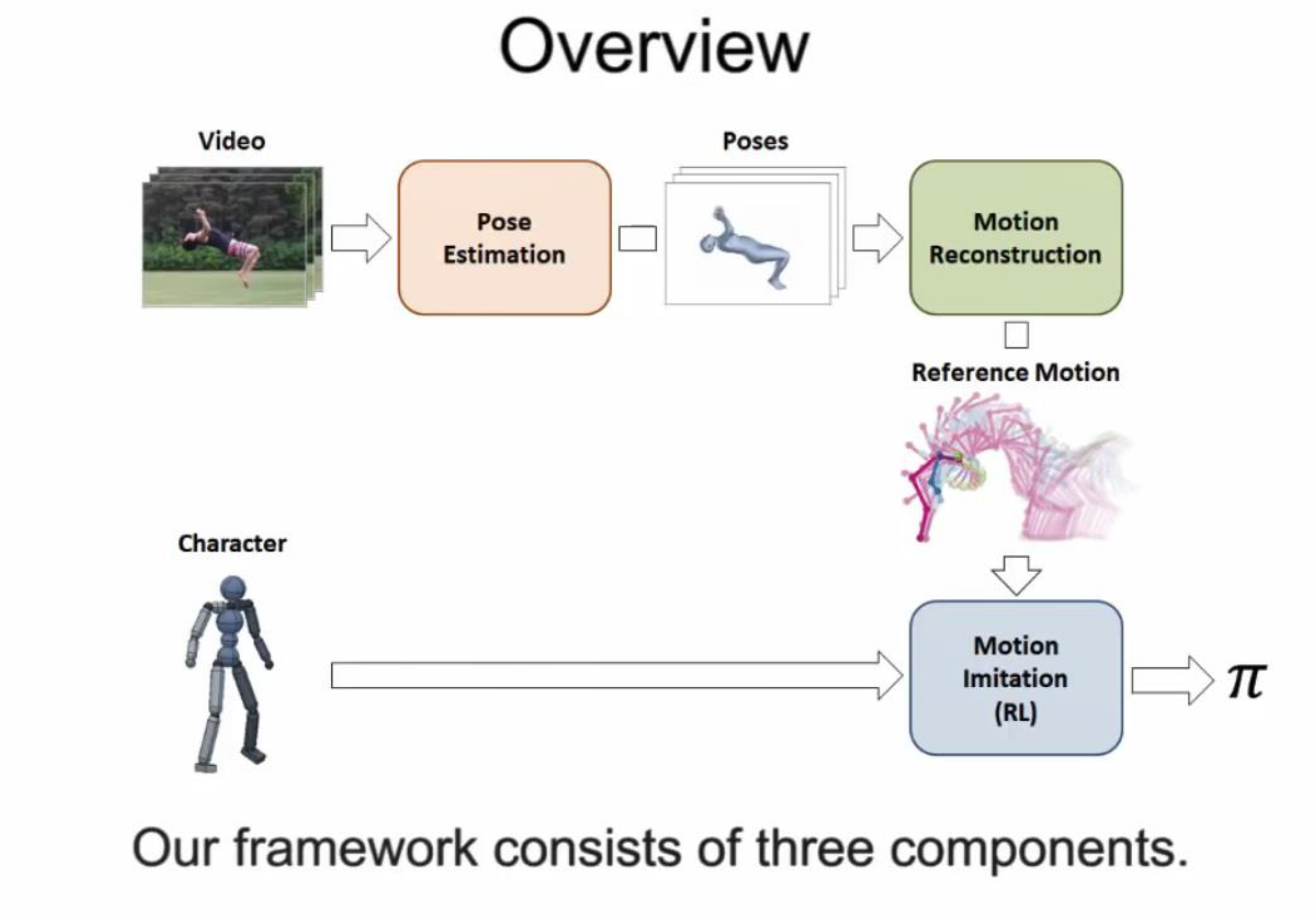


Learning tasks from instructional videos

Example: Learning skills from videos

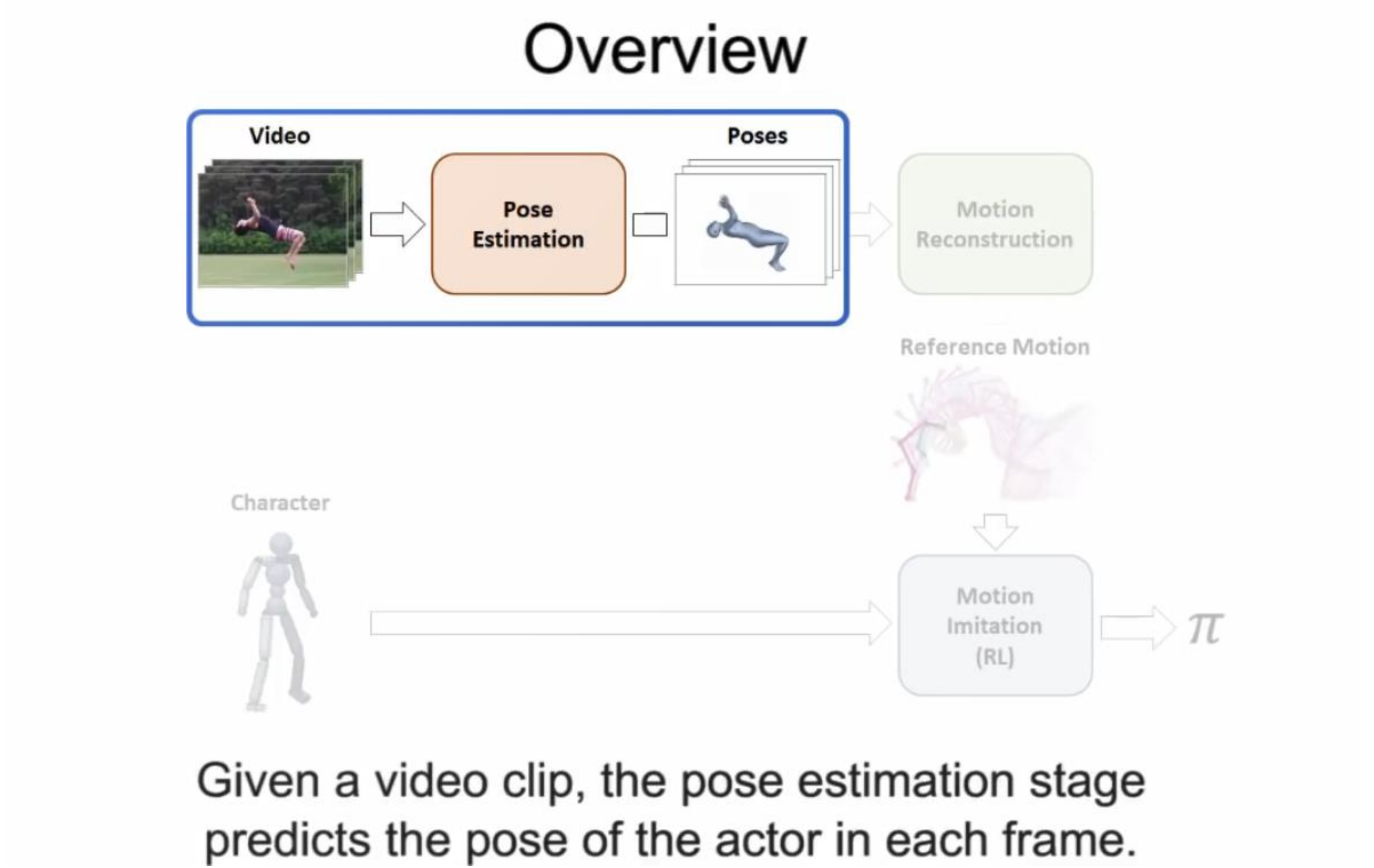
Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos



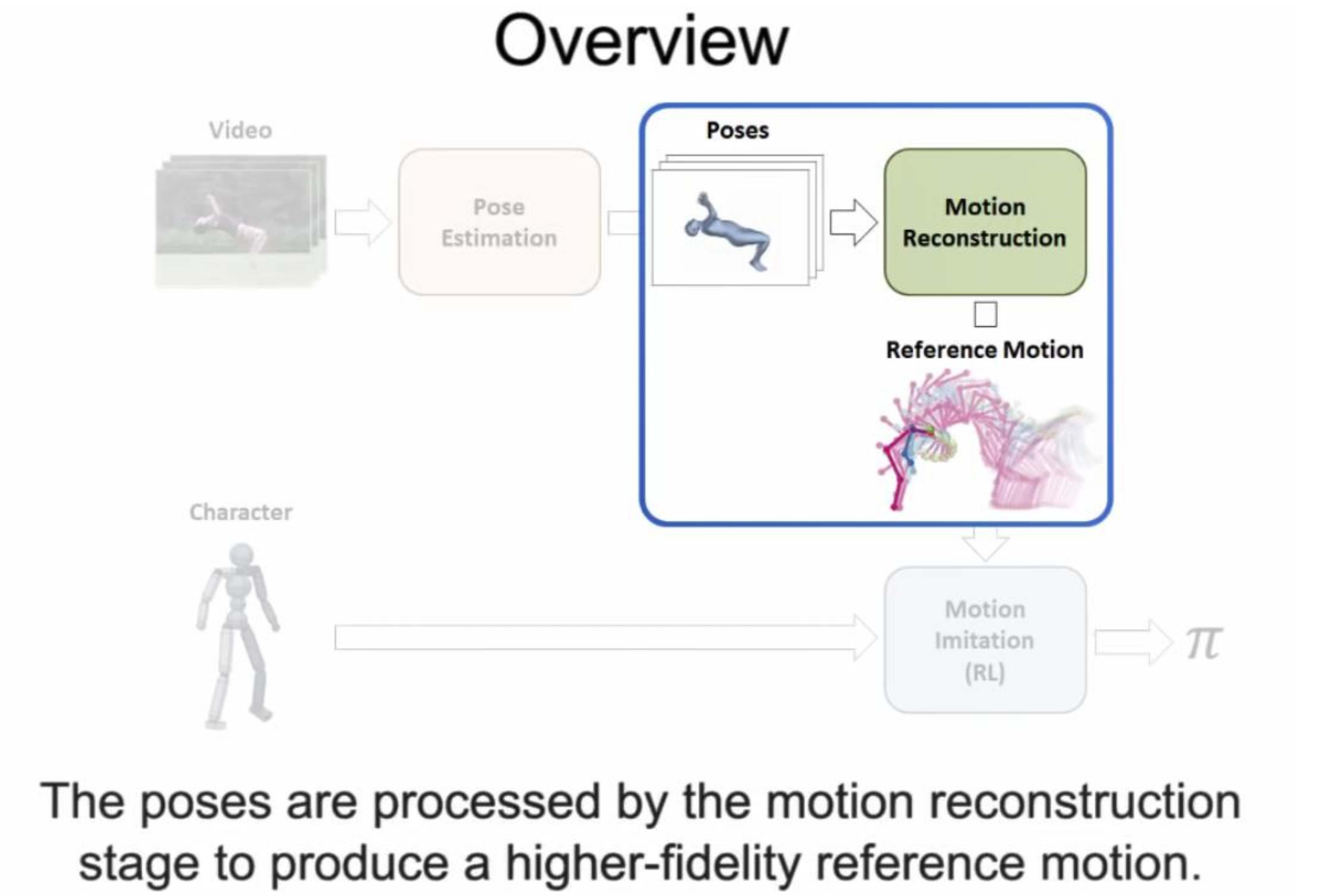
Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos



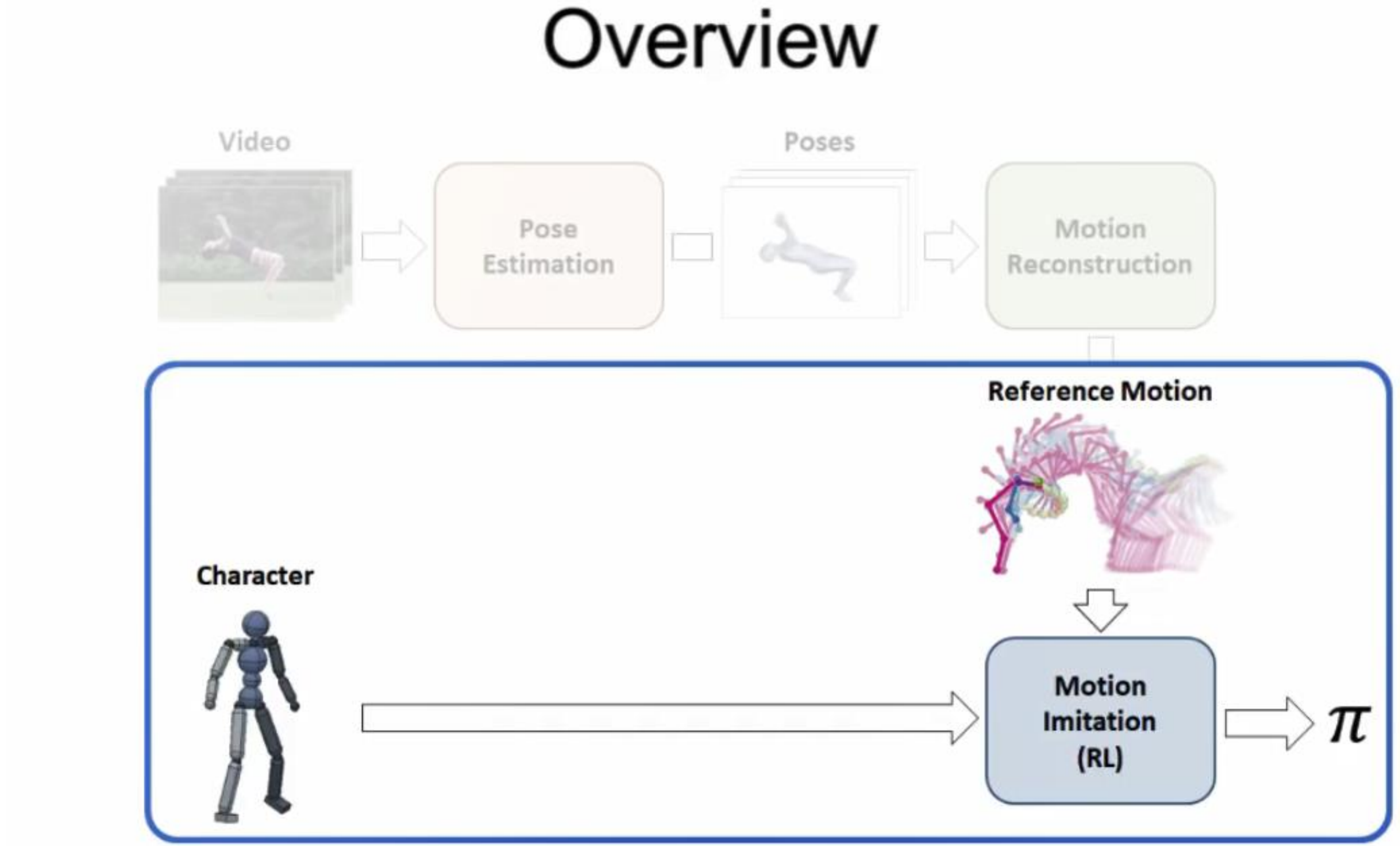
Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos



Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos

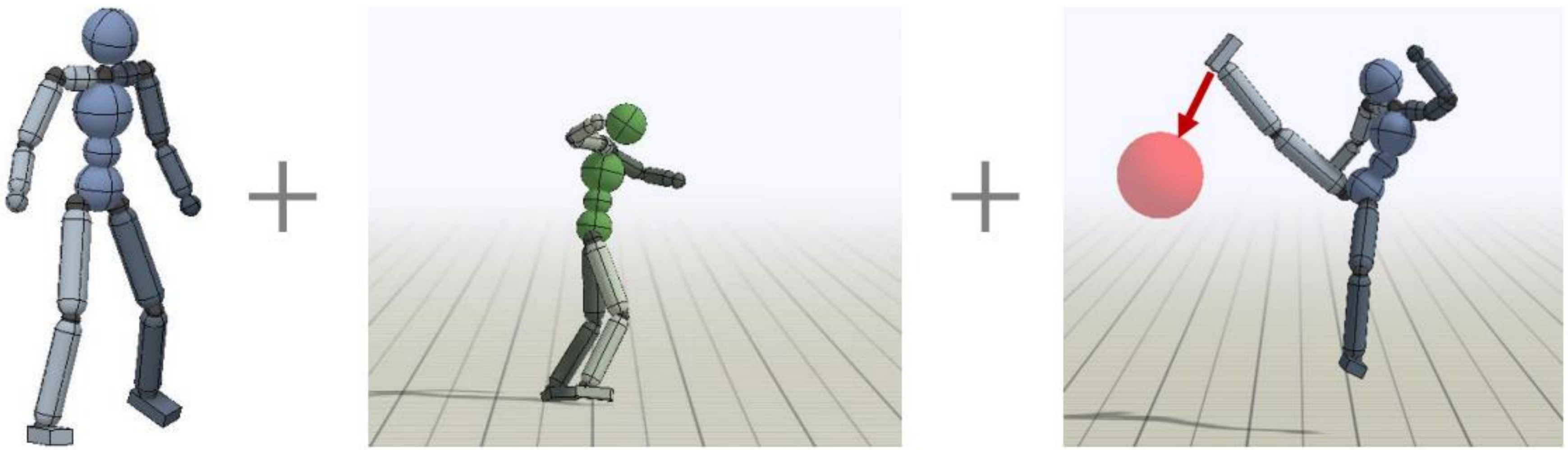


In the motion imitation stage, a policy is trained with reinforcement learning to imitate the reference motion.

Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos

Overview

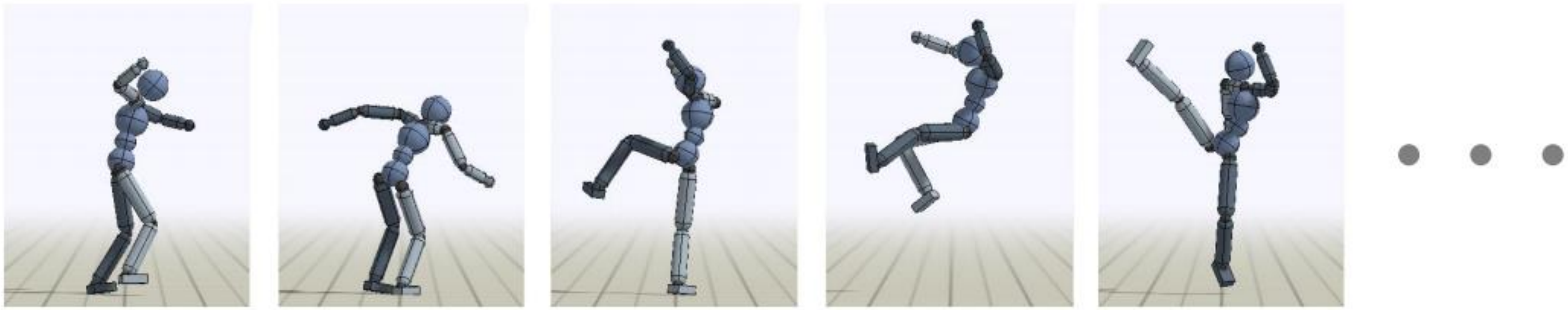


Character + Reference Motion + Task: Hit Target

Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos

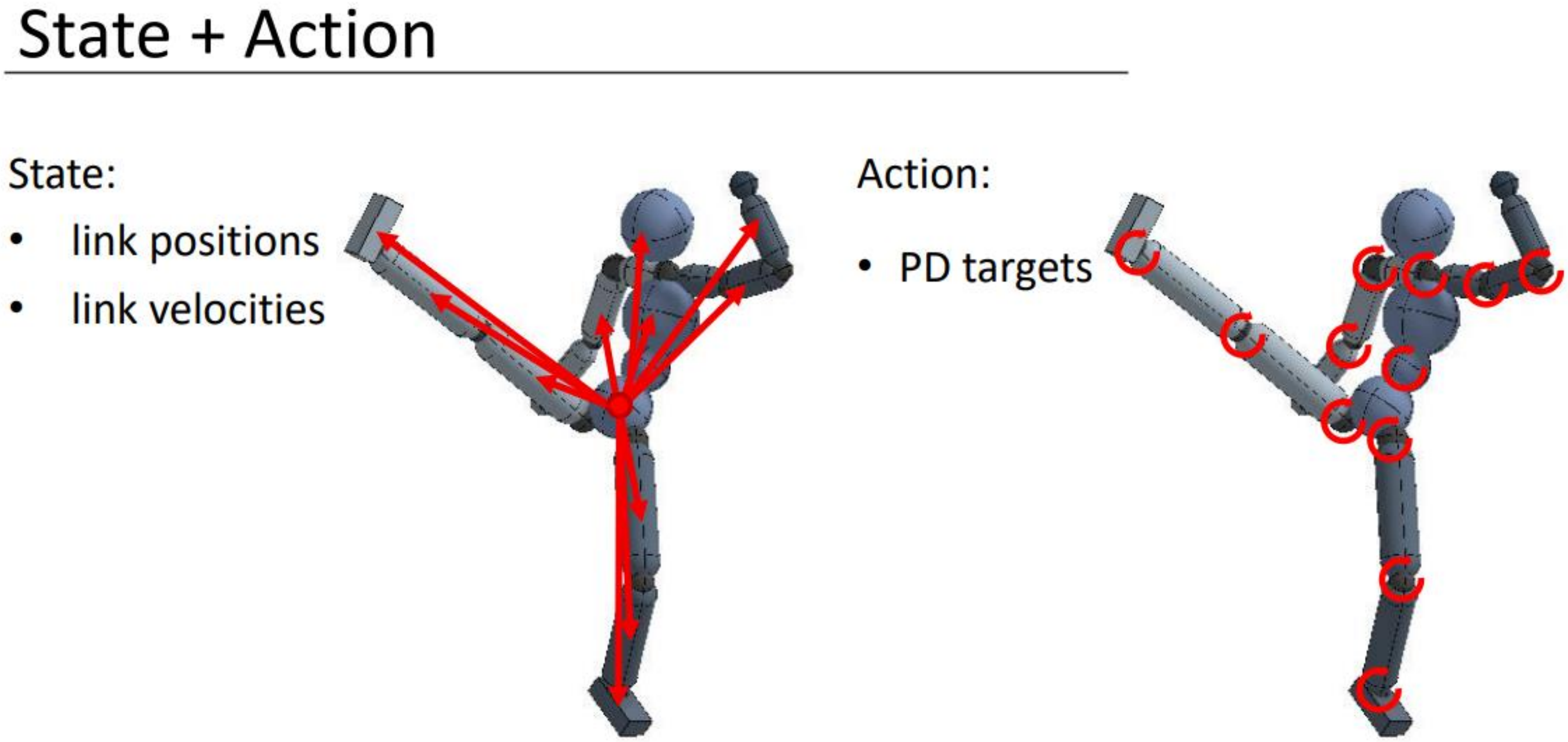
Reference Motion



$a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad \cdots$

Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos



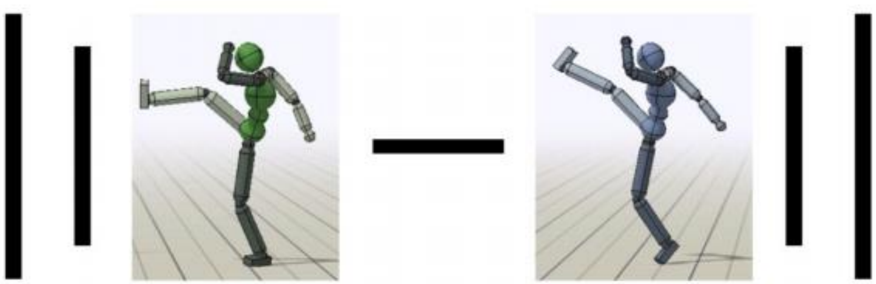
Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos

Reward

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

Imitation Objective

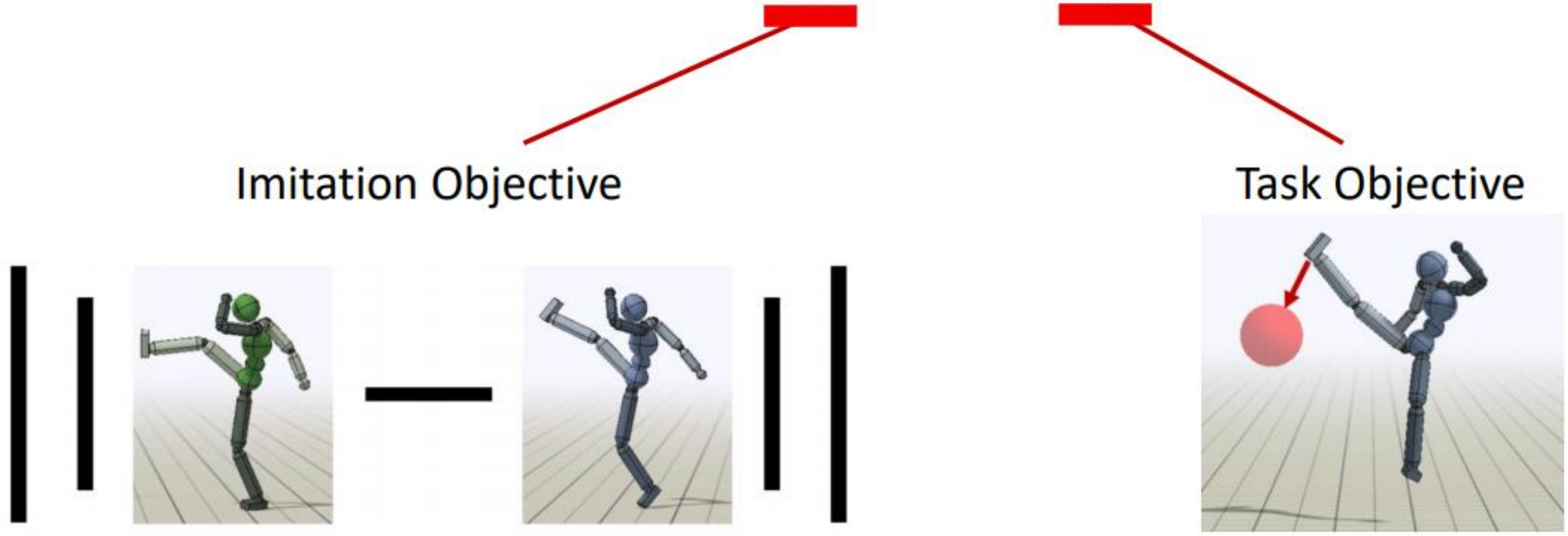


Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos

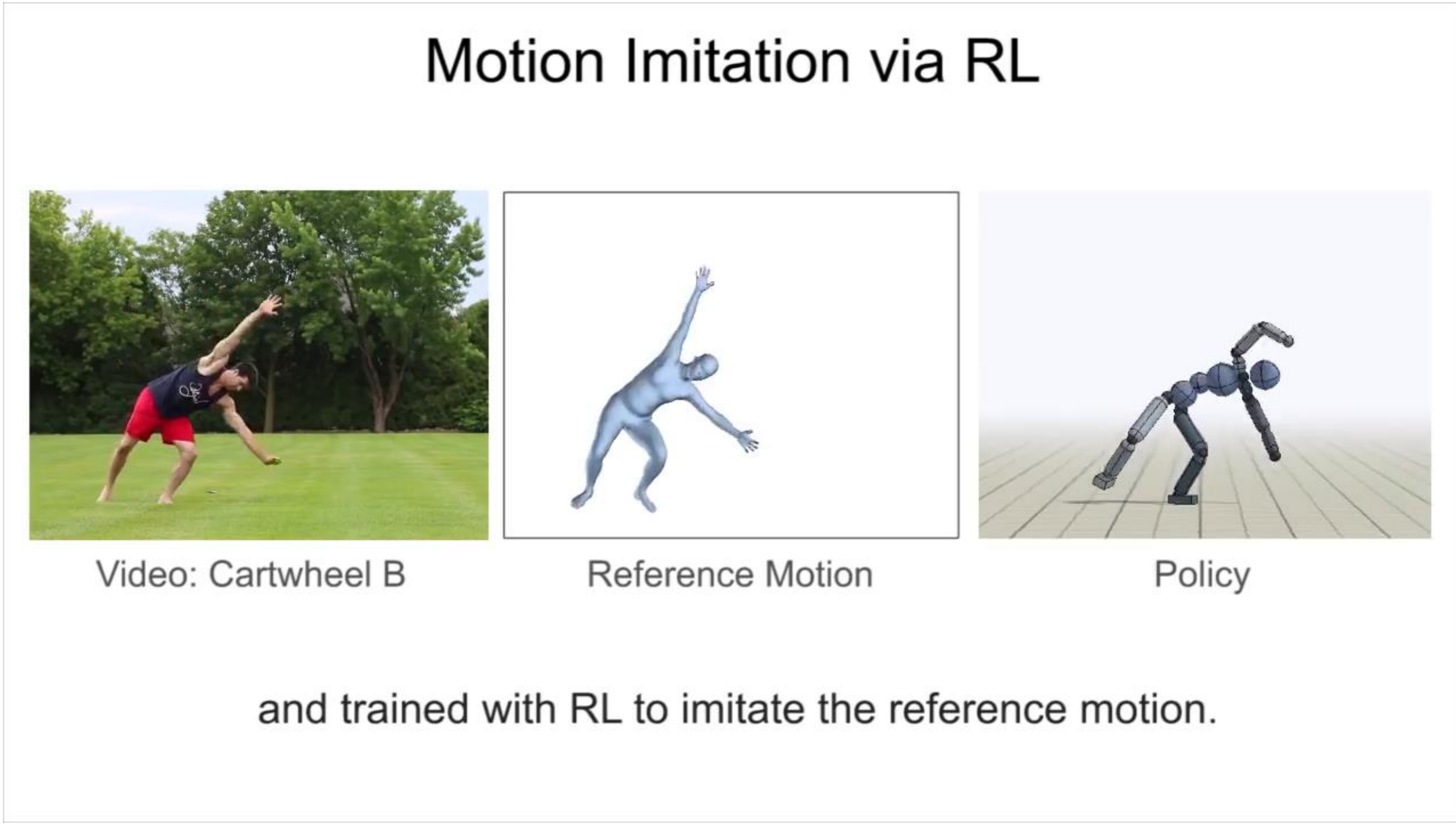
Reward

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



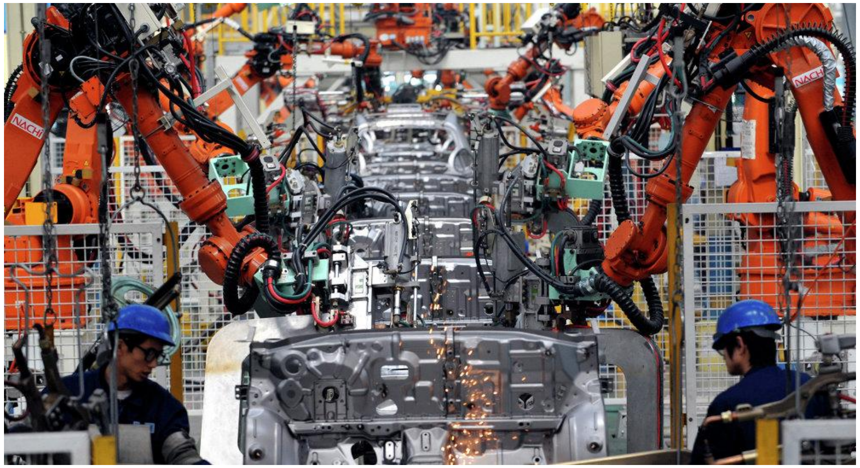
Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Example: Learning skills from videos



Peng XB, Kanazawa A, Malik J, Abbeel P, Levine S. SFV: Reinforcement learning of physical skills from videos. ACM TOG 2019

Robotics more generally…



Structured → **Unstructured**

Factory Robots:

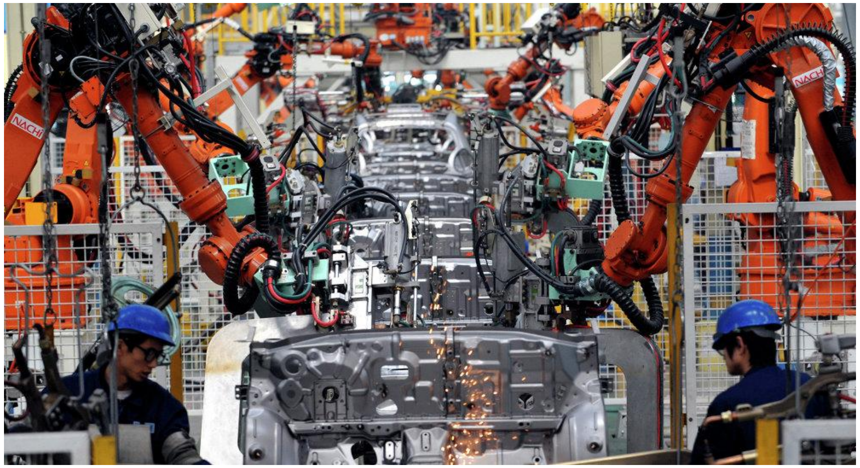
➔ Specialized, Task specific

➔ Very constrained factory environment where everything is predefined

Collaborative Robots:

➔ Open environment with varying conditions

➔ Needs to be generalist, handle multiple tasks, collaborate with people



Structured → **Unstructured**

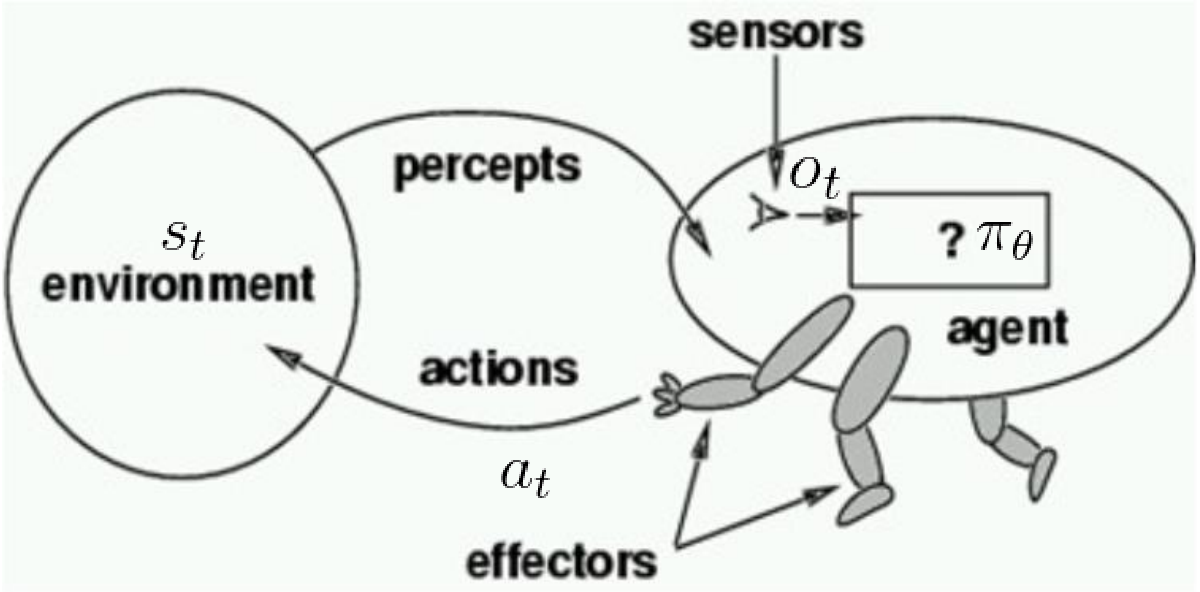
Perception

- Perception is a key element for a robot to work in an open world. One of the main component of Perception is Vision.

- Need for a robot with a good visual representation of its environment.

How to learn actions given raw sensory input?

Perception-Action cycle



s_t state
o_t observation
a_t action
π_θ policy

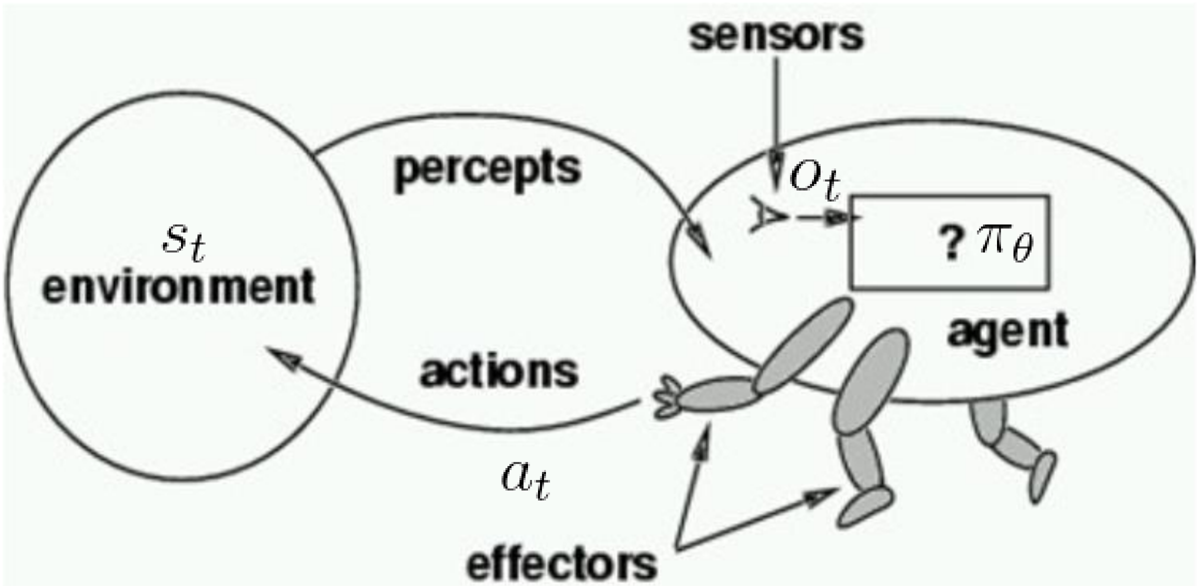
How to obtain $\pi_\theta(a_t|o_t)$?

Strategy 1: State-based

- estimate $\tilde{s}_t$ from o_t
- use Newtonian physics and explicit 3D geometry to derive a_t

estimating $\tilde{s}_t$ from o_t may be very hard

Perception-Action cycle



s_t state
o_t observation
a_t action
π_θ policy

How to obtain $\pi_\theta(a_t|o_t)$?

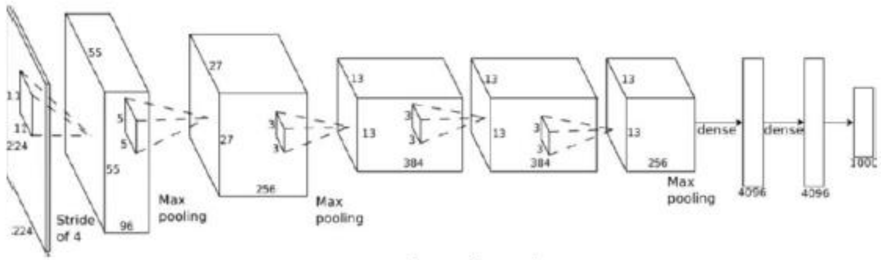
Strategy 1: State-based

- estimate $\tilde{s}_t$ from o_t
- use Newtonian physics and explicit 3D geometry to derive a_t

Strategy 2: sensor-based

- learn $\pi_\theta(a_t|o_t)$ from the data

Imitation Learning



$\mathbf{o}_t$

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{a}_t$

$\mathbf{o}_t$
$\mathbf{a}_t$

training data

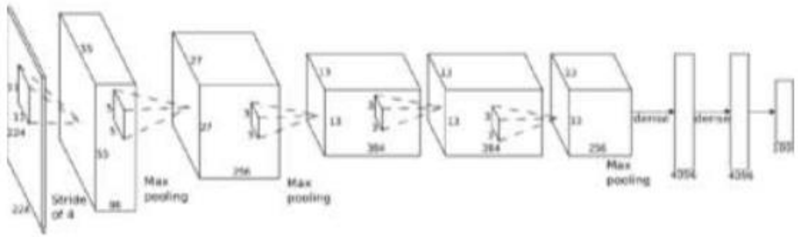
supervised learning

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

behavior cloning

Slide credit: S. Levine

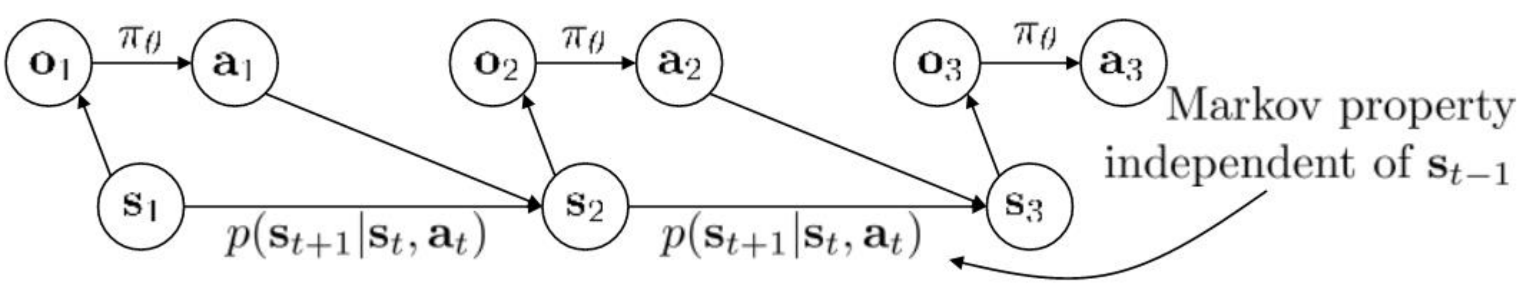
Imitation Learning



$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

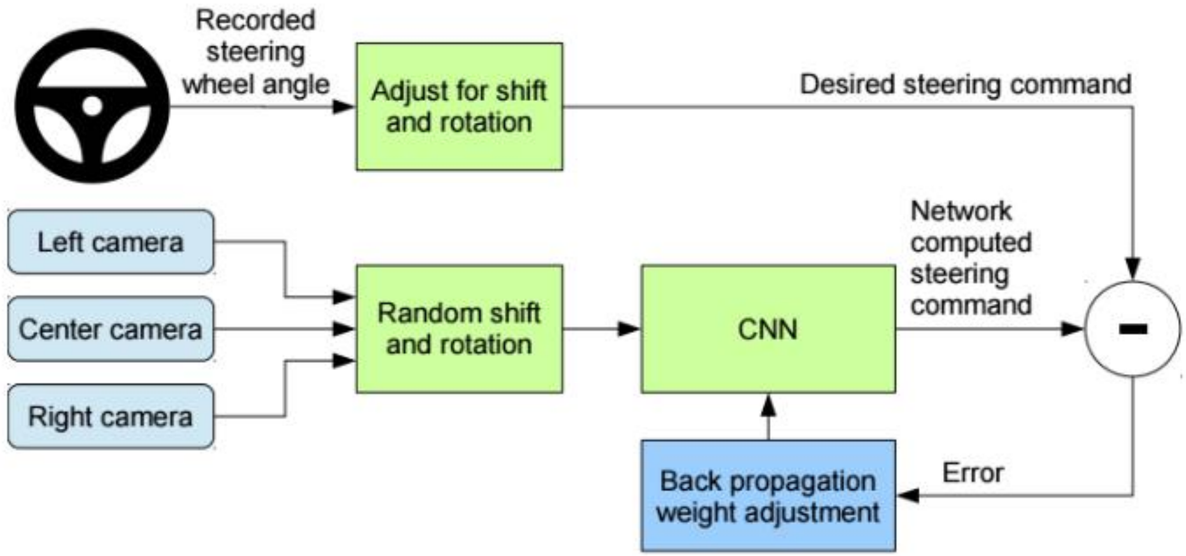
$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)



Slide credit: S. Levine

Imitation Learning



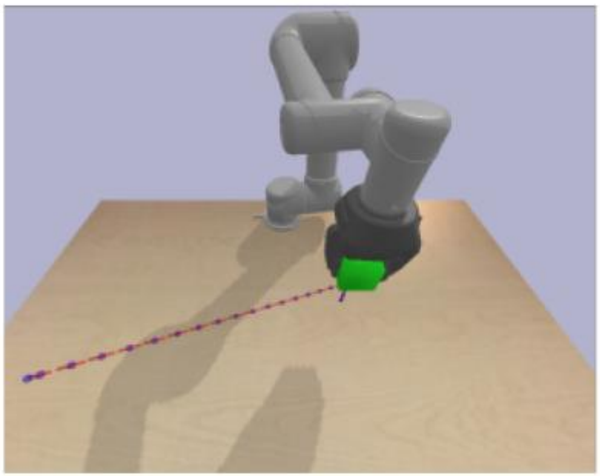
1. Collect a set of training data $\mathcal{D} = \{(o_i^*, a_i^*)\}_{i=1...n}$ where actions are performed by an expert agent.
2. Train a model π_θ to minimize

$$\mathcal{L}(\pi_\theta) = l(\pi_\theta(o_i^*), a_i^*)$$

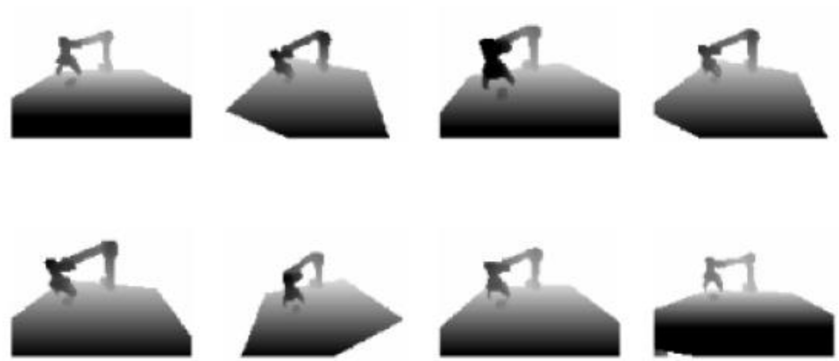
where l is any loss function.
For example : $\mathcal{L}(\pi_\theta) = \|\pi_\theta(o_i^*) - a_i^*\|_2^2$

Slide credit: S. Levine

Imitation Learning



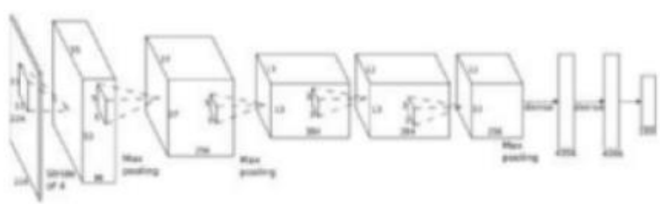
$\boldsymbol{\tau}^{\text{demo}} = [\boldsymbol{y}_0, \boldsymbol{u}_0, ..., \boldsymbol{y}_T, \boldsymbol{u}_T]$ ⇨ Demonstration by experts $\mathcal{D} = \{\boldsymbol{\tau}_i^{\text{demo}}\}_{i=1}^{N}$



Record Depth Images $\mathbf{D}_t$ from Multiple Viewpoints.

$\mathbf{o}_t = (\mathbf{D}_{t-2}, \mathbf{D}_{t-1}, \mathbf{D}_t)$

Input of the network



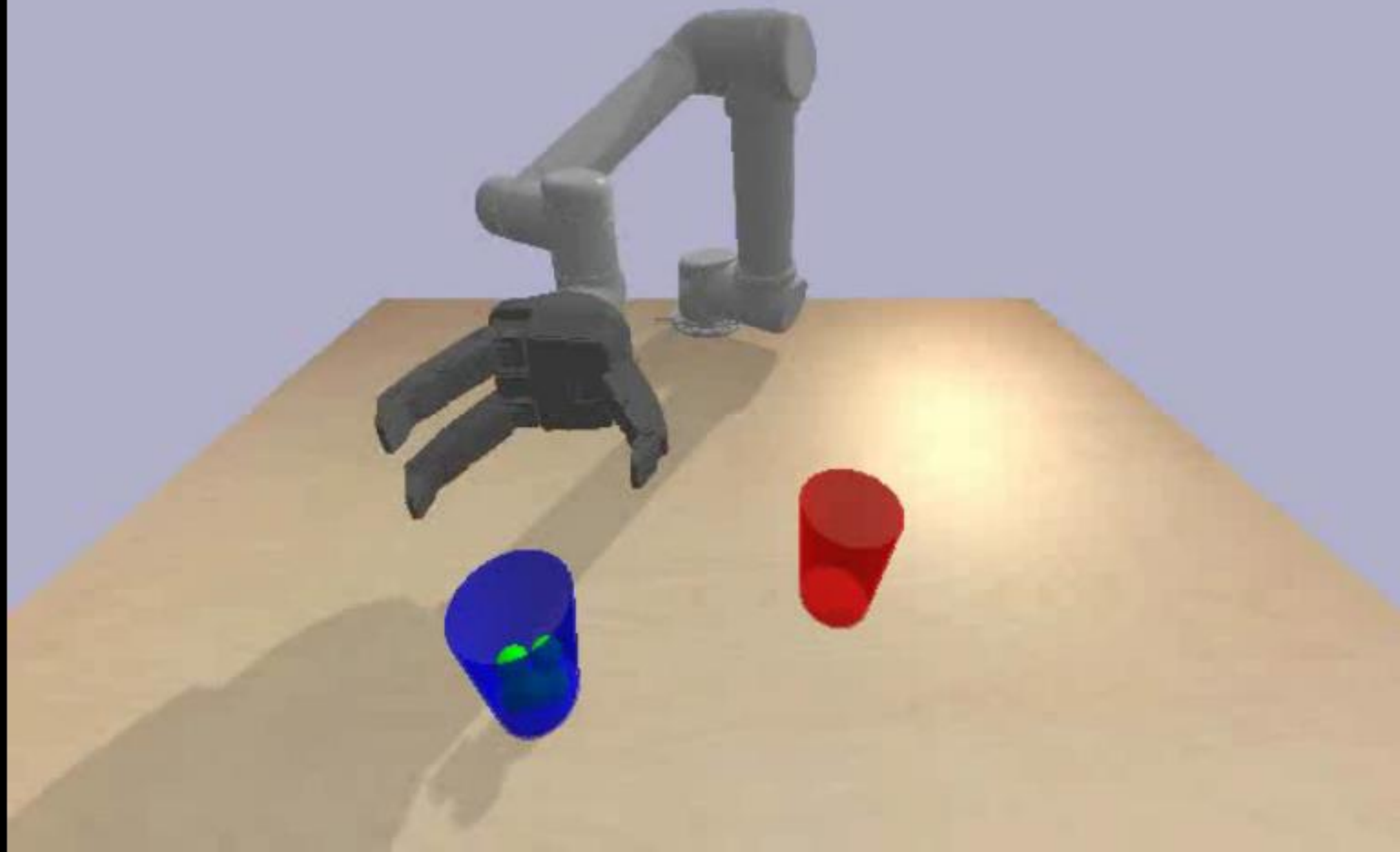
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$

$\mathbf{v}_t$ linear velocity $\mathbb{R}^3$

$\boldsymbol{\omega}_t$ angular velocity $\mathbb{R}^3$

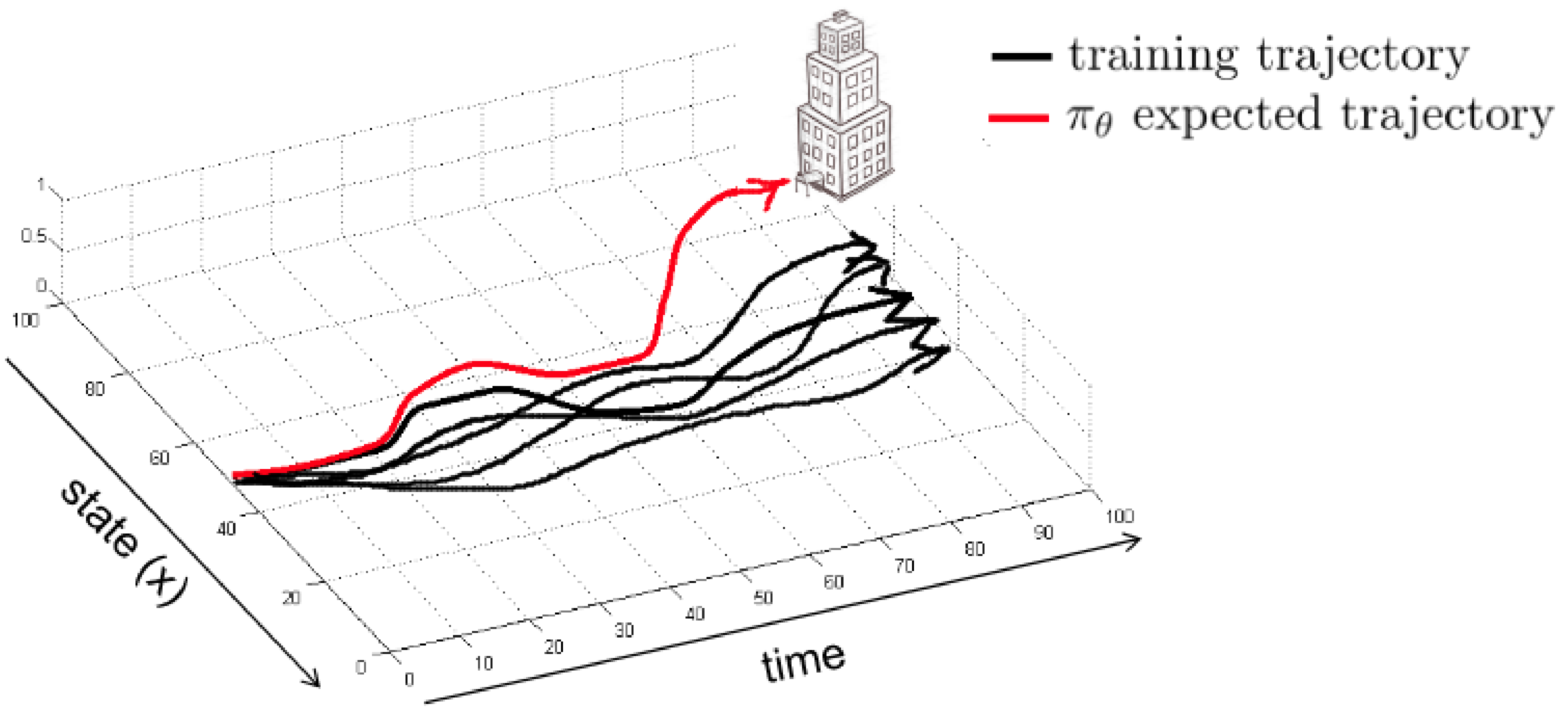
$\mathbf{g}_t$ gripper velocity $\mathbb{R}$

UR5-Pour

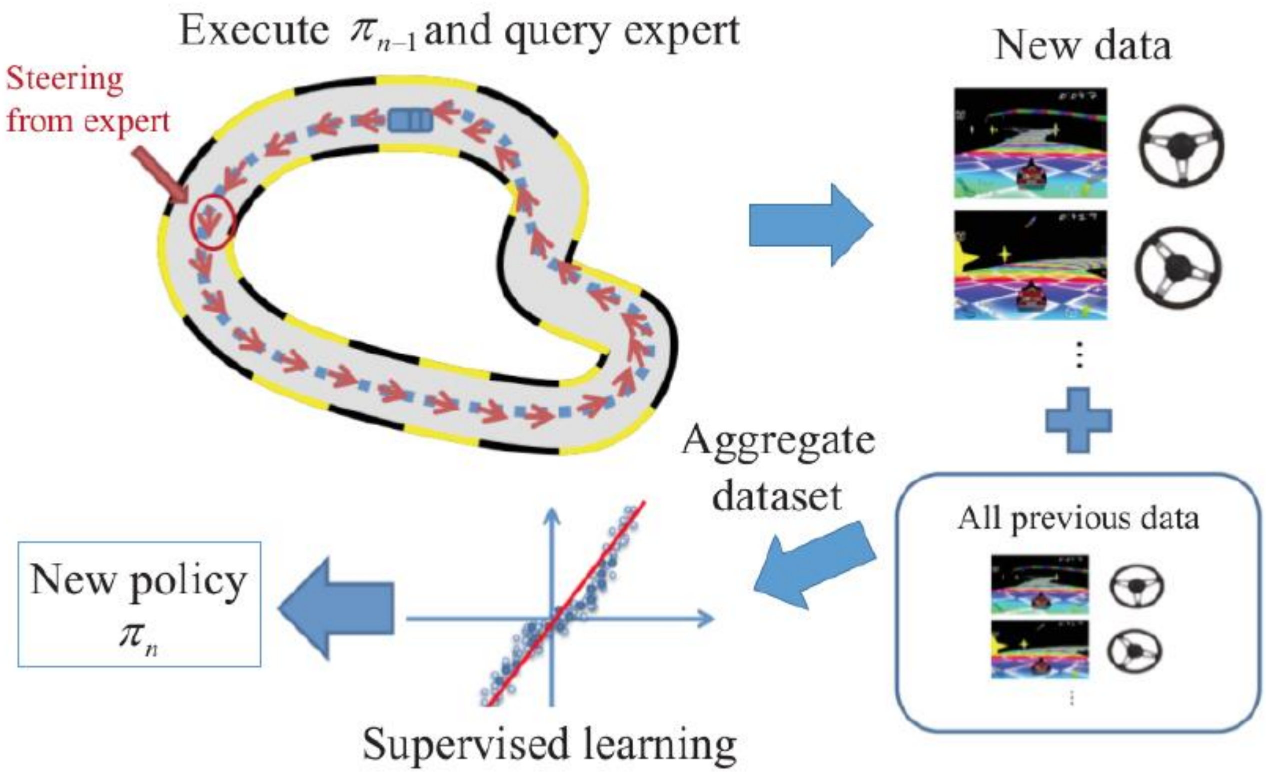


Imitation Learning: Limitations

Only optimize a local problem, a state-action distribution only describes the short term behavior.
➔ lead to mismatch with long-term behavior.



Imitation Learning: DAgger



In each iteration, DAgger generates new examples using the current policy with corrections provided by the $\boxed{\text{expert}}$, then adds the new demonstrations to the dataset and optimize over it.

Expert may have access to the simulator state during training

Learning vision-guided robotic manipulation

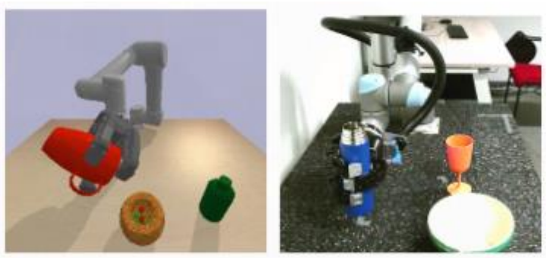
[Robin Strudel Alexander Pashevich, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid, 2019]

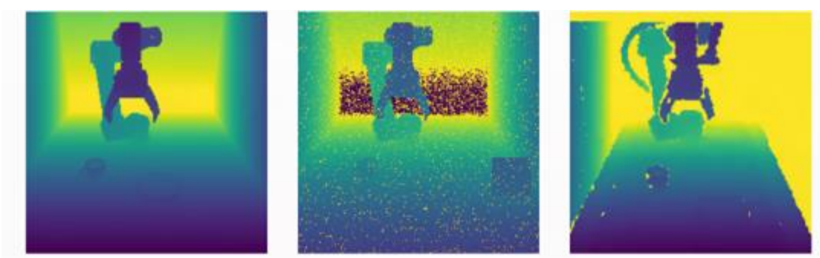
Learning Implicit Representations

- Sample efficient approach: Combine Imitation Learning and Reinforcement Learning
- Policy transfer: Train in simulation, deploy on a real robot

Learning to combine primitive skills

Sim2Real Policy Transfer





[1] Learning to combine primitive skills: A step towards versatile robotic manipulation, Robin Strudel*, Alexander Pashevich*, Igor Kalevatykh, Ivan Laptev, Josef Sivic, Cordelia Schmid, 2019. Under review.

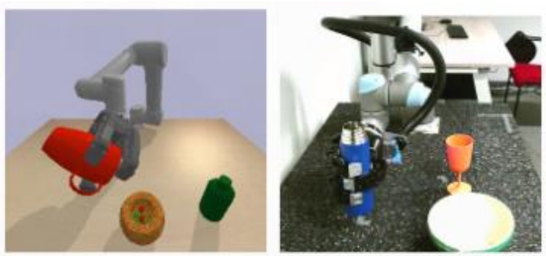
[2] Learning to Augment Synthetic Images for Sim2Real Policy Transfer, Alexander Pashevich*, Robin Strudel*, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid. IROS 2019.

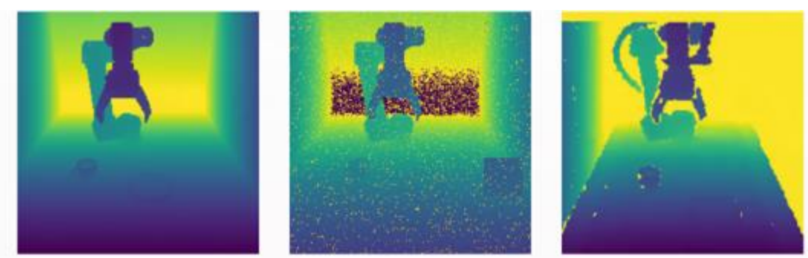
Learning Implicit Representations

- Sample efficient approach: Combine Imitation Learning and Reinforcement Learning
- Policy transfer: Train in simulation, deploy on a real robot

Learning to combine primitive skills

Sim2Real Policy Transfer





[1] Learning to combine primitive skills: A step towards versatile robotic manipulation, Robin Strudel*, Alexander Pashevich*, Igor Kalevatykh, Ivan Laptev, Josef Sivic, Cordelia Schmid, 2019. Under review.

[2] Learning to Augment Synthetic Images for Sim2Real Policy Transfer, Alexander Pashevich*, Robin Strudel*, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid. IROS 2019.

Motivation

- Imitation learning (IL)
  + is good at solving short tasks
  - requires a lot of demonstrations
  - can not go beyond demonstrations

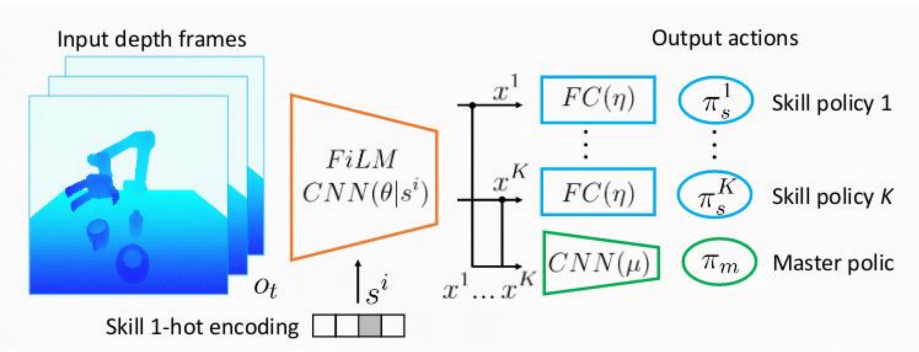
- Reinforcement learning (RL)
  - can solve novel problems using only the reward function
  - requires task-specific reward engineering
  - is sample inefficient

- Our approach: RL selecting skills learnt with BC (RLBC)

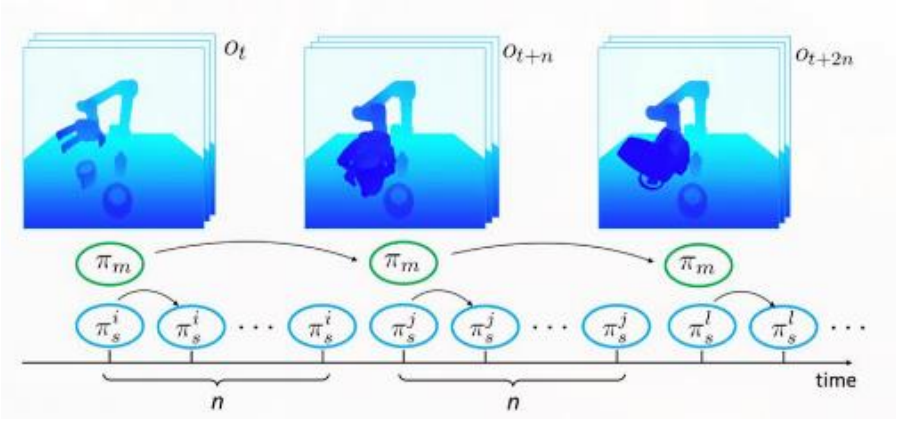
Motivation

- Sample inefficient $\rightarrow$ shorten the horizon of the RL by using discrete actions
- Reward shaping is difficult $\rightarrow$ use a single sparse reward for RL

Learn a vocabulary of BC skills π_s^i

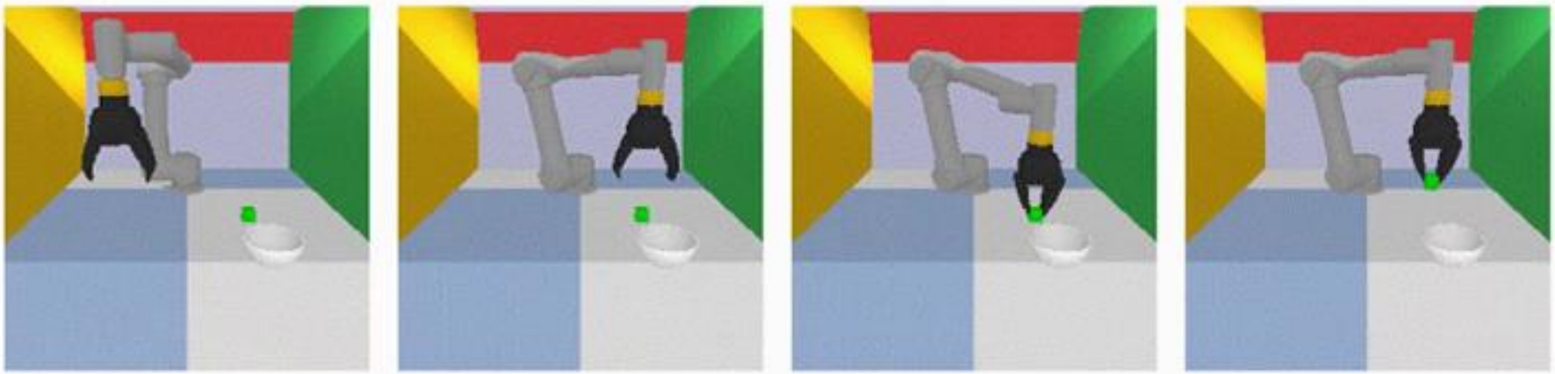


Train a master policy π_m combining skills with RL



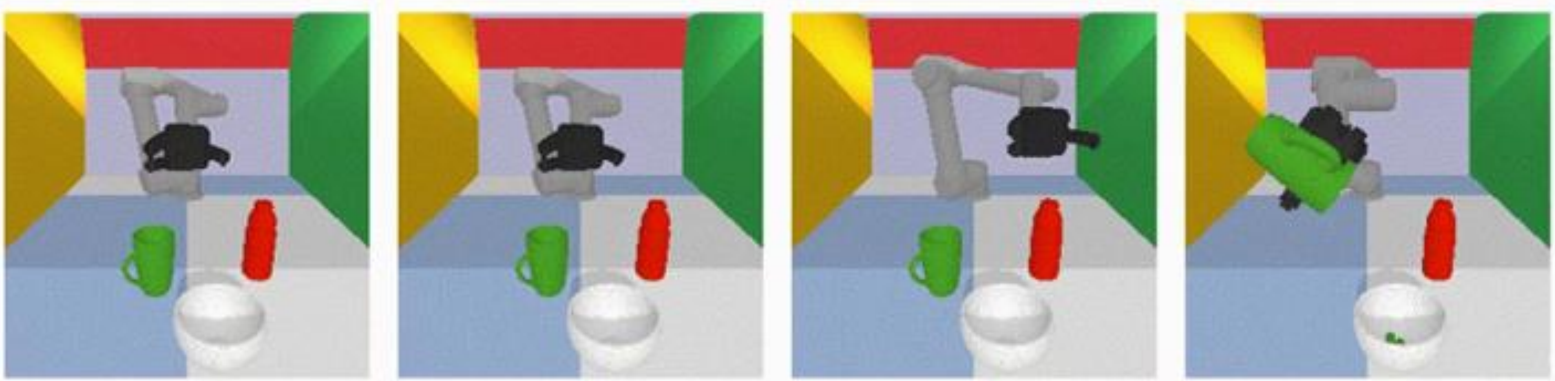
Skills trained with Behavioral Cloning

Task: UR5 - Bowl



go to the cube | go down and grasp | lift the cube | go to the bowl

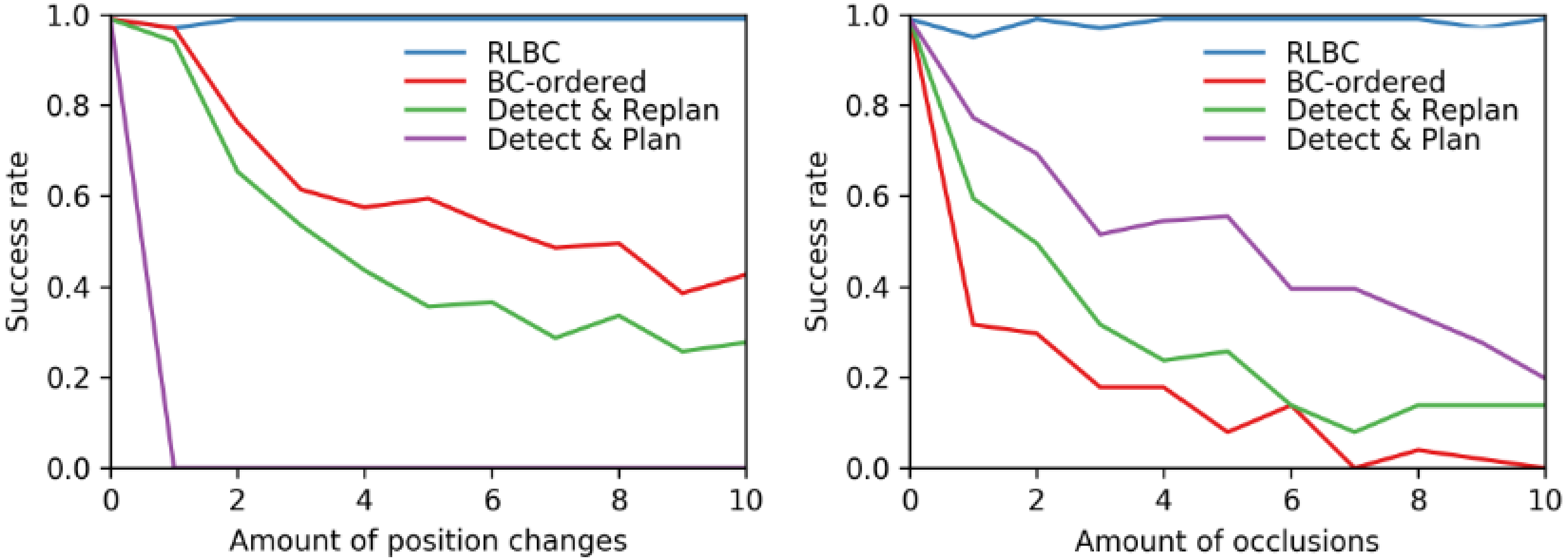
Task: UR5 - Breakfast



go to the cup | go to the bottle | grasp an object and pour | release an object


1. Go to the cube

Robustness to perturbations



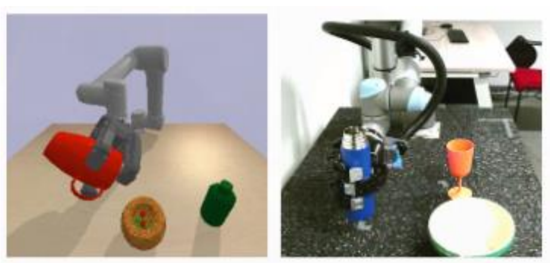
How to get a good sim2real transfer?

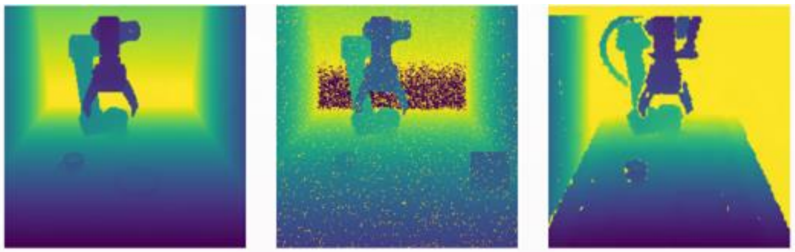
Learning Implicit Representations

- Sample efficient approach: Combine Imitation Learning and Reinforcement Learning
- Policy transfer: Train in simulation, deploy on a real robot

Learning to combine primitive skills

Sim2Real Policy Transfer





[1] Learning to combine primitive skills: A step towards versatile robotic manipulation, Robin Strudel*, Alexander Pashevich*, Igor Kalevatykh, Ivan Laptev, Josef Sivic, Cordelia Schmid, 2019. Under review.

[2] Learning to Augment Synthetic Images for Sim2Real Policy Transfer, Alexander Pashevich*, Robin Strudel*, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid. IROS 2019.

How to obtain training data?

- Physics simulators &
  realistic computer graphics

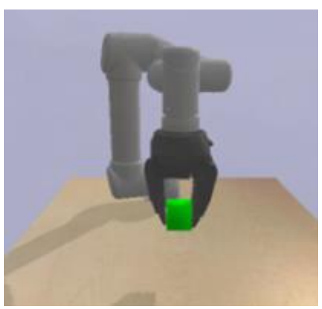
  + Fast and scalable: Can generate millions of images
  + Full-state knowledge: Can use standard planning tools to generate demonstrations

How to obtain training data?

- Physics simulators &
  realistic computer graphics

  + Fast and scalable: Can generate millions of images
  + Full-state knowledge: Can use standard planning tools to generate demonstrations
  - Domain gap: trained policies do not transfer well to the real world

Synthetic

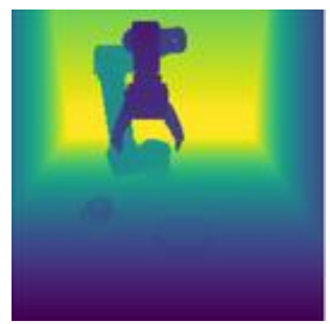


$\neq$

Real

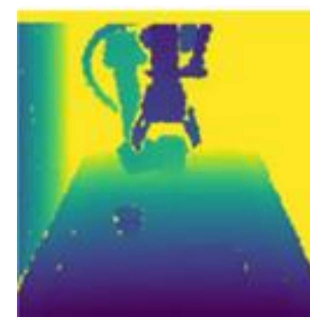
RGB

Synthetic



$\neq$

Real



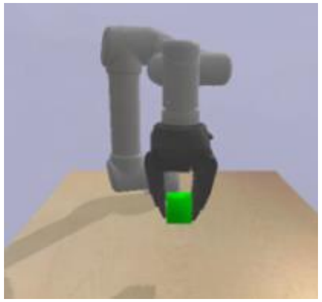
Depth

How to obtain training data?

- Physics simulators &
  realistic computer graphics
  - + Fast and scalable: Can generate millions of images
  - + Full-state knowledge: Can use standard planning tools to generate demonstrations
  - - Domain gap: trained policies do not transfer well to the real world

This work

Synthetic

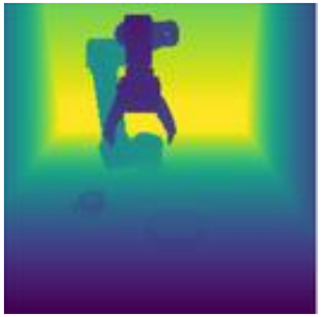


$\neq$

Real

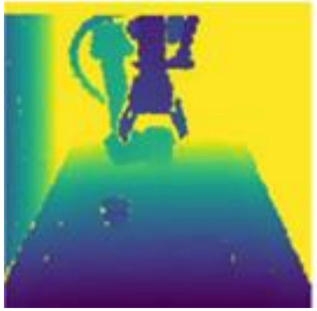
RGB

Synthetic



$\neq$

Real

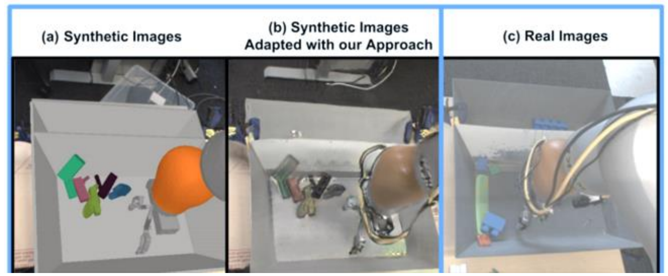


Depth

How to handle domain gap?

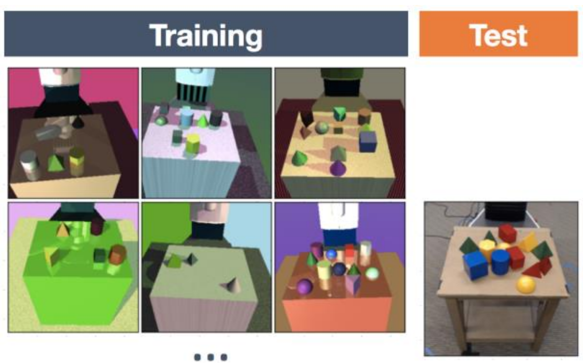
- Domain adaptation

  Bousmalis et al., Using simulation and domain adaptation to improve efficiency of deep robotic grasping. ICRA 2018



- Domain randomization

  Tobin et al., Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, IROS 2017

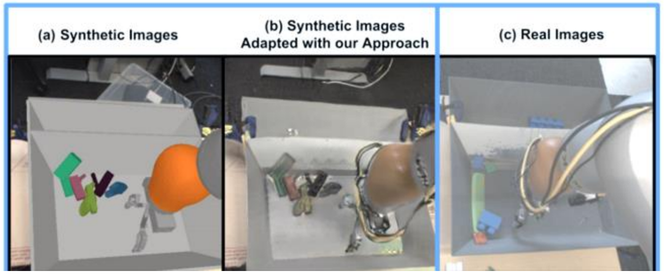


Cubuk et al., AutoAugment: Learning Augmentation Policies from Data, CVPR 2019

How to handle domain gap?

- Domain adaptation

  Bousmalis et al., Using simulation and domain adaptation to improve efficiency of deep robotic grasping. ICRA 2018

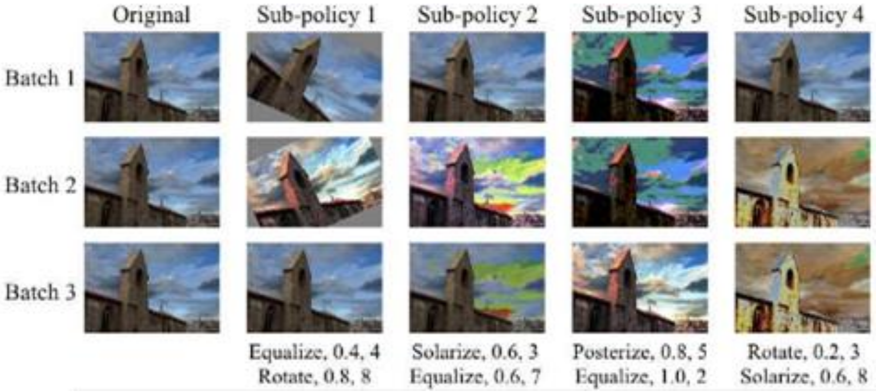


This work

- Domain randomization

  Tobin et al., Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World, IROS 2017

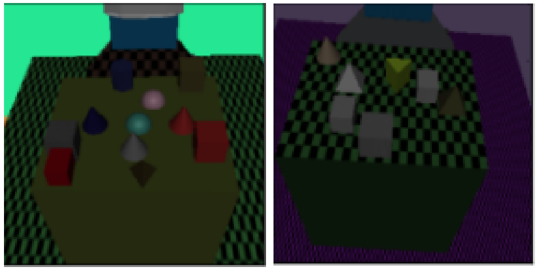
Cubuk et al., AutoAugment: Learning Augmentation Policies from Data, CVPR 2019



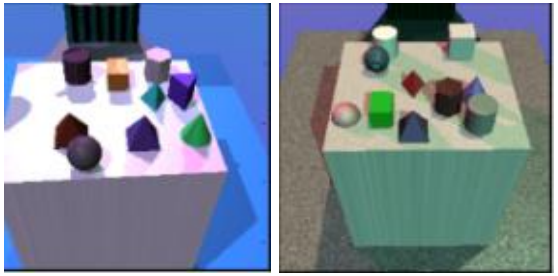
What randomization?

- Previous work: manual selection of random transformations

Texture

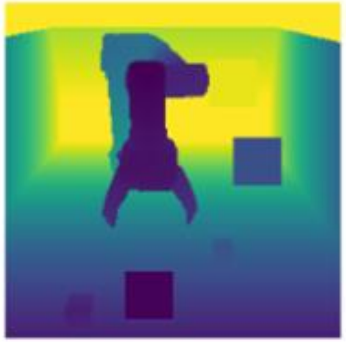


Lightning

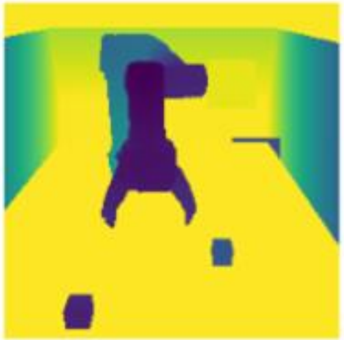


- Our contributions: Optimize the sequence of random transformations for manipulation tasks

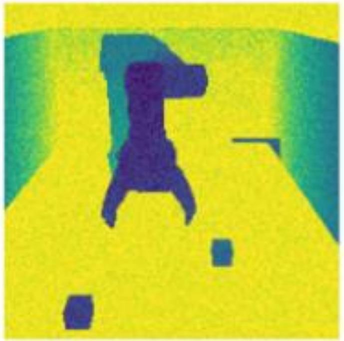
1) Cutout



2) EraseObject



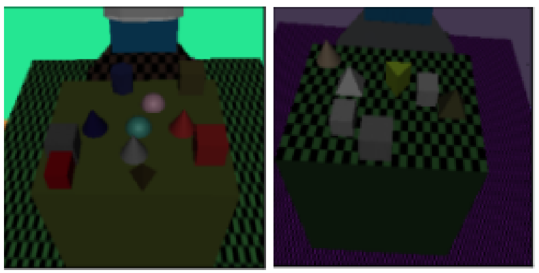
3) WhiteNoise



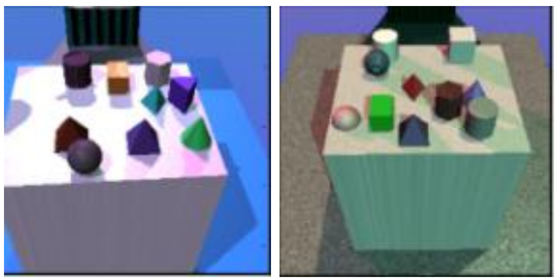
What randomization?

- Previous work: manual selection of random transformations

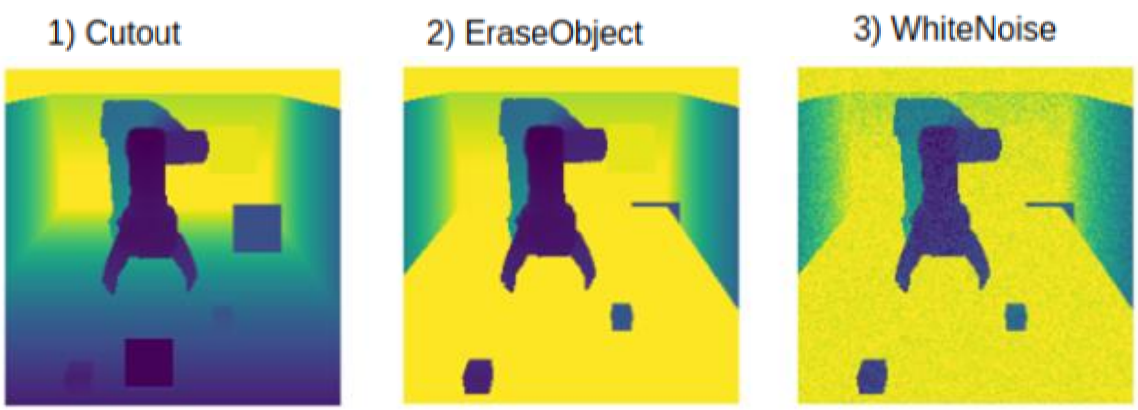
Texture



Lightning



- Our contributions: Optimize the sequence of random transformations for manipulation tasks

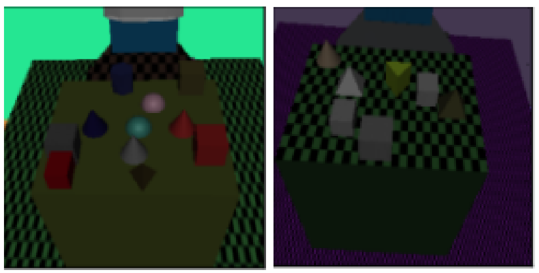


Q: How to choose suitable augmentations?

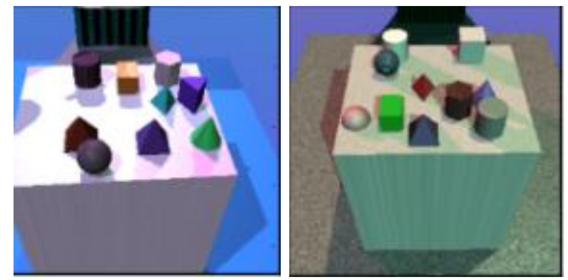
What randomization?

- Previous work: manual selection of random transformations

Texture

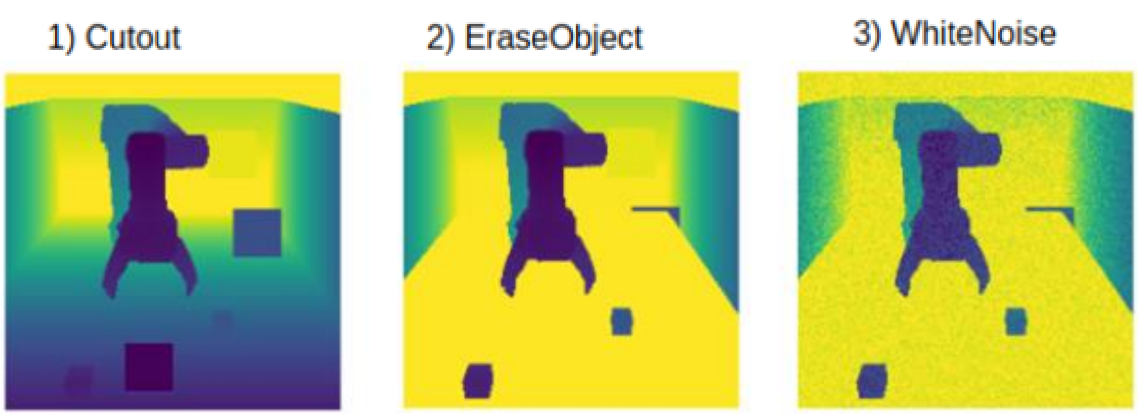


Lightning



- Our contributions: Optimize the sequence of random transformations for manipulation tasks

Q: How to choose suitable augmentations?

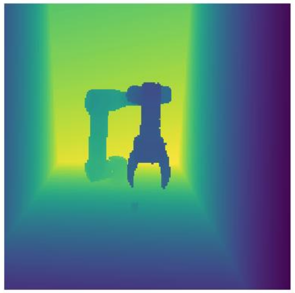


... → Sim2Real Score

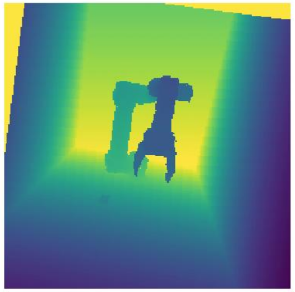
Augmentation function space

Augmentation function is a sequence of N primitive transformations. We consider 11 primitive transformations, for each of them we select magnitude and probability.

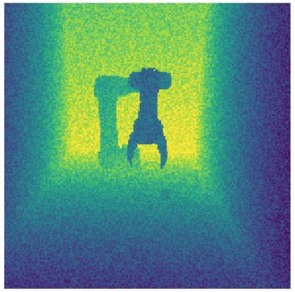
Examples of primitive transformations:



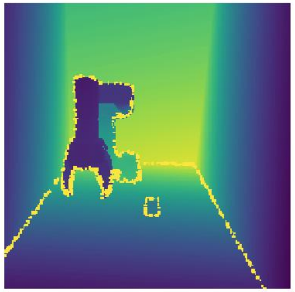
original image



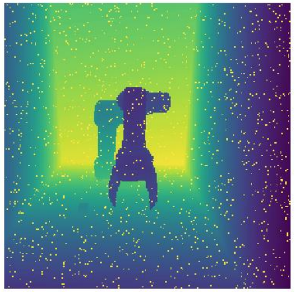
affine



gaussian noise



edge noise

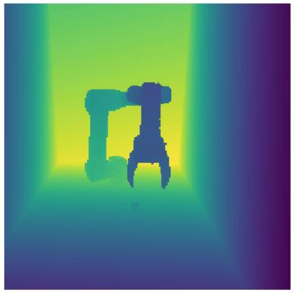


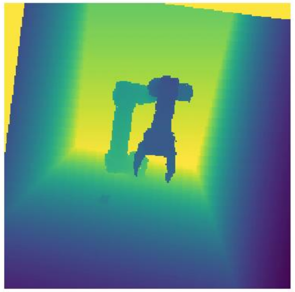
bernoulli noise

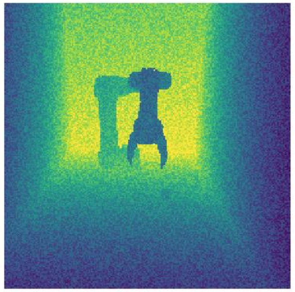
Augmentation function space

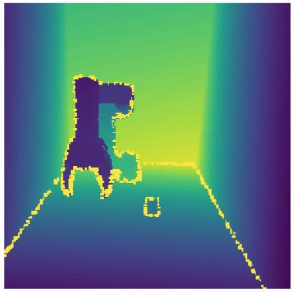
Augmentation function is a sequence of N primitive transformations. We consider 11 primitive transformations, for each of them we select magnitude and probability.

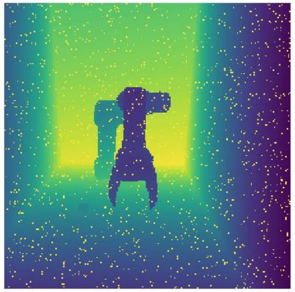
Examples of primitive transformations:


original image


affine

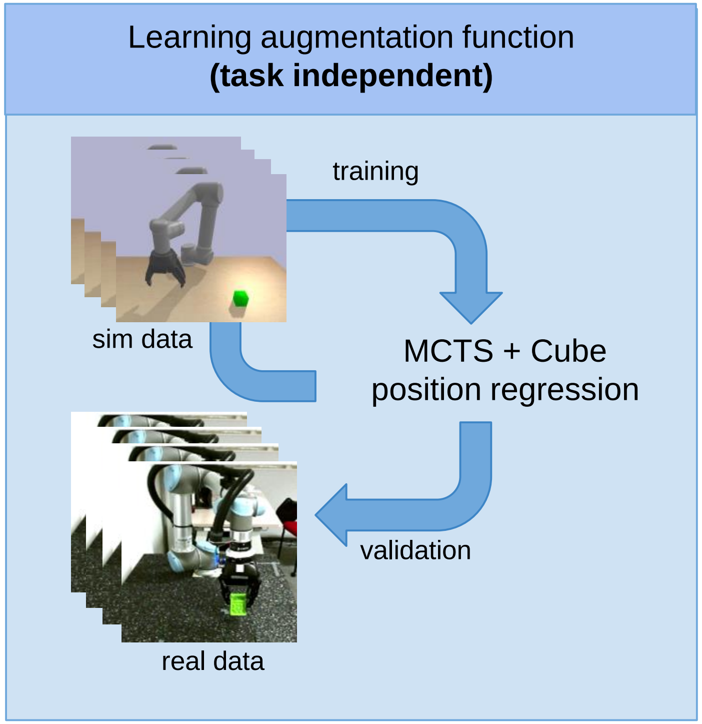

gaussian noise


edge noise


bernoulli noise

For N=8, there are ~10^{14} possible augmentation functions.

Cube position regression as a proxy task



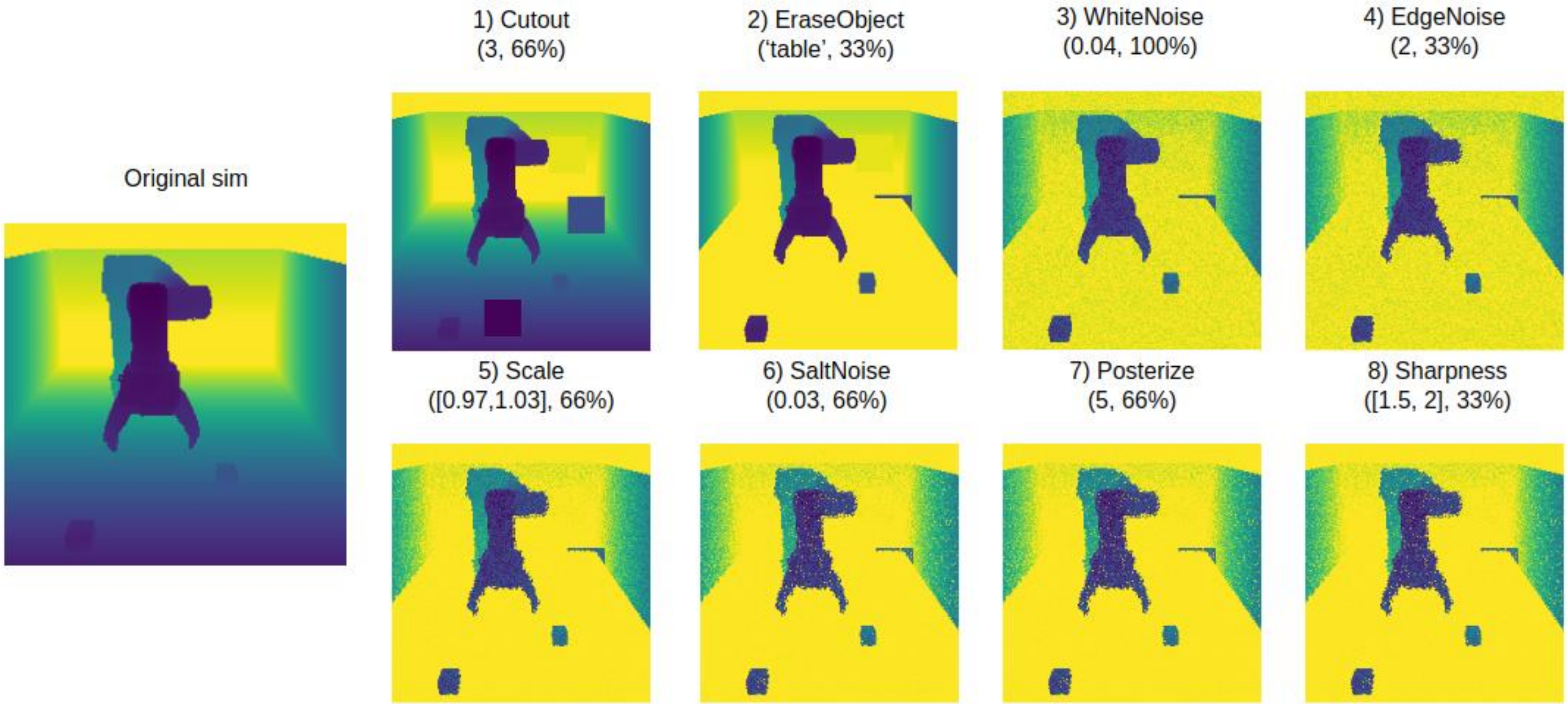
1. Training network to in simulation to regress cube position (ResNet-18, 20000 depth images + cube position);

2. Evaluate network on real images (200 validation images, predict cube position);

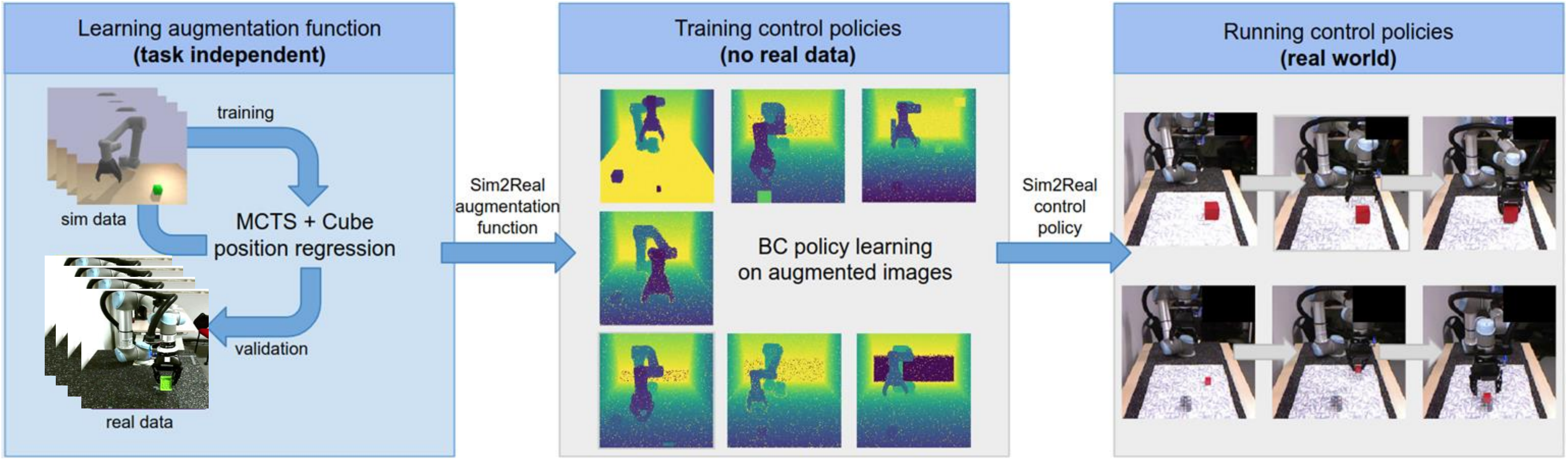
3. Use prediction error to find optimal sequence of augmentations with Monte-Carlo Tree Search*.

* Rémi Coulom, Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. ICCG 2006

Example of a learned augmentation function



Method pipeline



1. Collect demonstrations in simulation
2. Augment demonstrations with learned function
3. Train policy on augmented data

Quantitative results: Proxy task

	Augmentation	Error in sim	Error in real
i	None	**0.63 ± 0.50**	6.52 ± 5.04
ii	Random (8 operations)	6.56 ± 4.05	5.77 ± 3.12
iii	Handcrafted (4 operations)	0.99 ± 0.68	2.35 ± 1.36
iv	Learned (1 operation)	1.19 ± 0.87	1.86 ± 2.45
v	Learned (4 operations)	1.21 ± 0.78	1.17 ± 0.71
vi	Learned (8 operations)	1.31 ± 0.90	**1.09 ± 0.73**

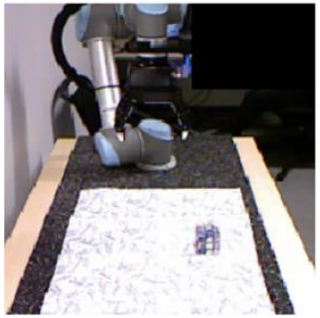
Cube prediction error (in cm) on synthetic and real depth images using different types of depth data augmentation.

Cube position regression

Quantitative results: Control tasks

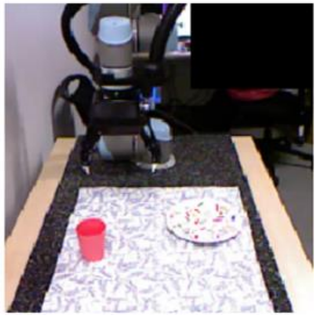
Augmentation	Pick	Stack	Cup Placing
None	3/20	1/20	0/20
Handcrafted (4 operations)	9/20	2/20	6/20
Learned (1 operation)	8/20	1/20	1/20
Learned (8 operations)	**19/20**	**18/20**	**15/20**

Success rates for control policies executed on a real robot (20 trials per experiment).



Cube picking

Cubes stacking



Cup placing

Learning to build 3D categories

[Alexander Pashevich, Igor Kalevatykh, Ivan Laptev and Cordelia Schmid IROS 2020]

Actions change the state of the world

before

after

Actions change the state of the world

before

after

Actions change the state of the world

before

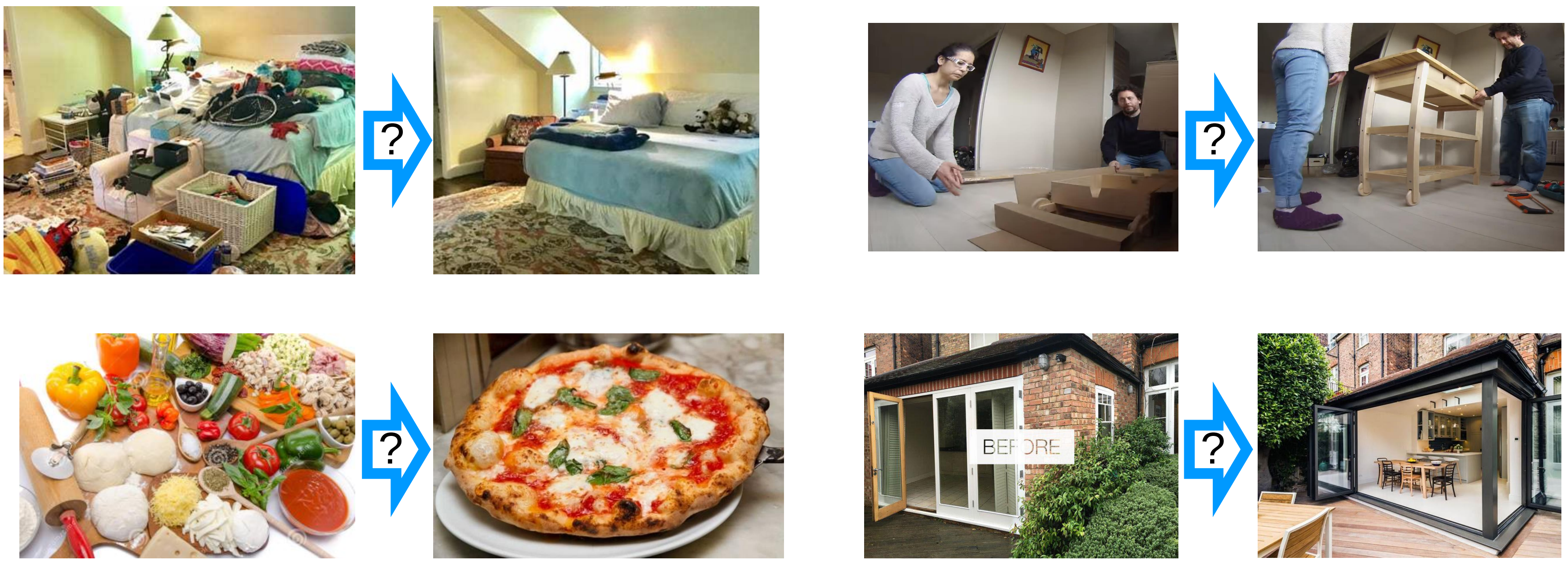
after

Actions change the state of the world

before

after

What actions are needed to change the world?



What actions are needed to change the world?

We do not have an answer, but:

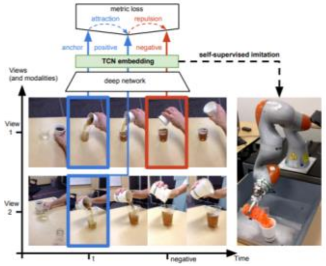
- We want to deal with diverse and complex scenes
- Assumption of a known state is unrealistic
- Vision and learning are likely to be key ingredients in the solution

Recent work on learning, vision and robotics

P. Agrawal et al. Learning to Poke by Poking: Experiential Learning of Intuitive Physics. NIPS 2016

L. Pinto and A. Gupta. Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours. ICRA 2016.

D. Gandhi et al., Learning to Fly by Crashing. ICRA 2017



P. Sermanet et al., Time-Constrastive Networks: Self-Supervised Learning from Multi-View Observation ICRA 2018

Great, but:

- Most work is looking at learning primitive actions
- We want to learn the full task

This work: Learning to build a 3D category

This work: Learning to build a 3D category

… let’s start simpler

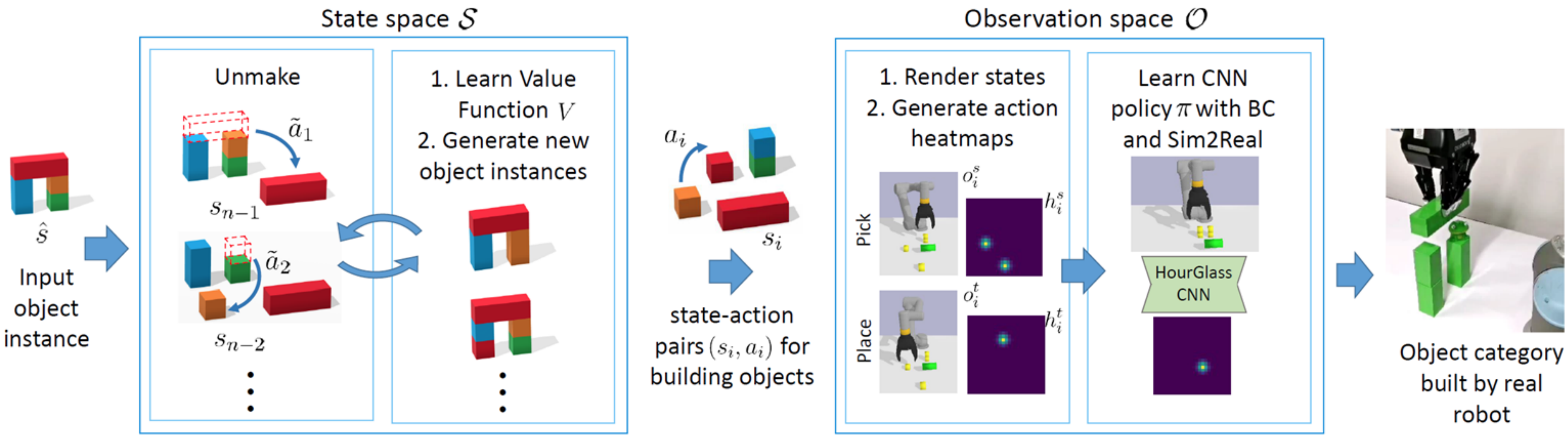
Given:
- 3D instance 
- Category classifier

?

before

after

Method overview

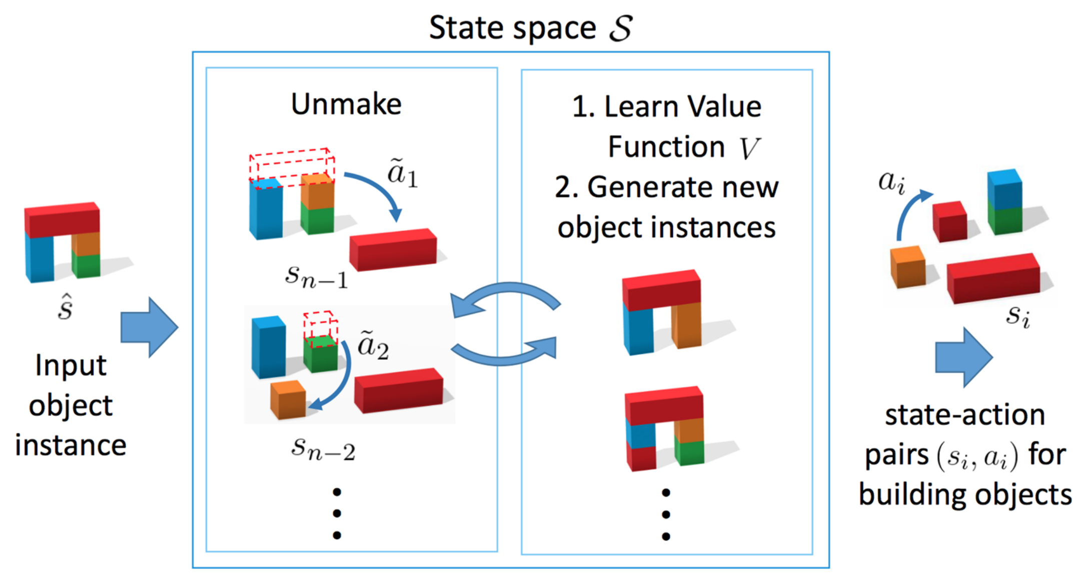


- Discover action plans in *state space*
- Learn visual policies in the *observation space*

State space: Make by unmaking

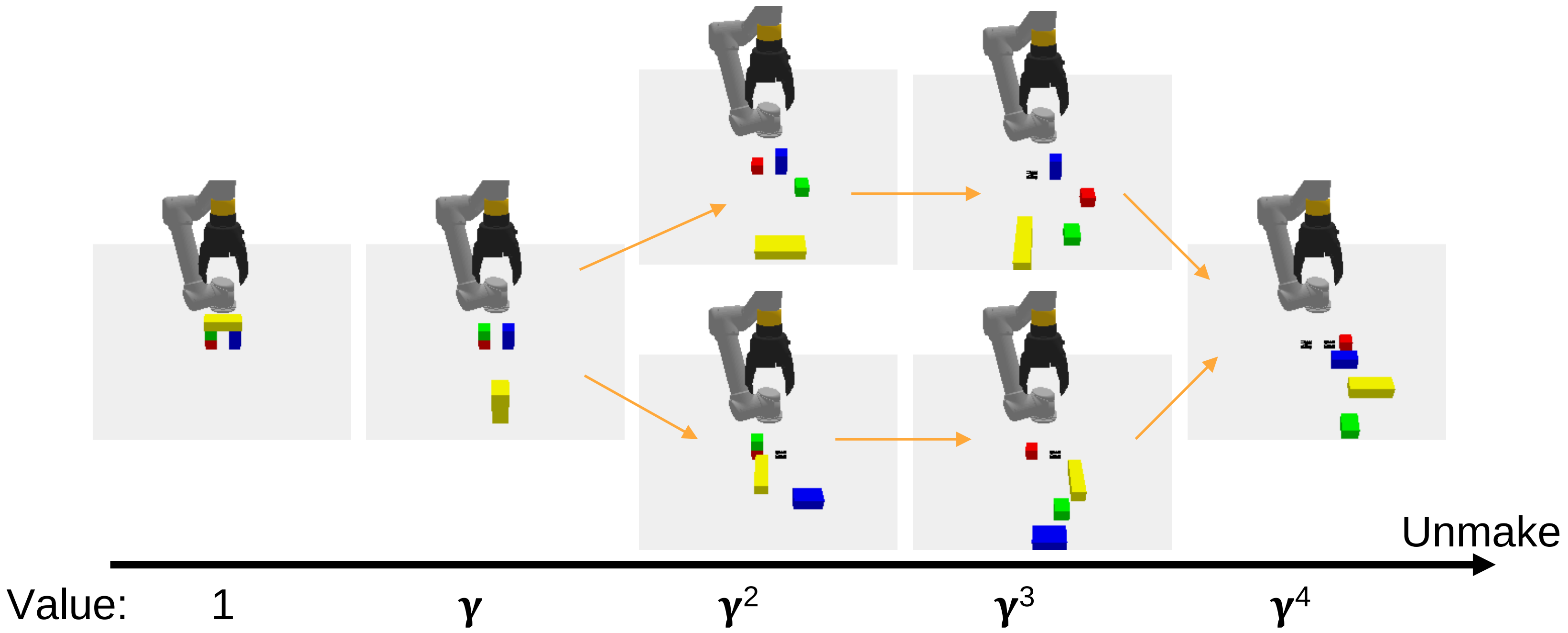
Goals:
- Find new instances
- Generate state-action pairs

1. Disassemble a given instance N times;
2. Learn a value function NN from N disassemblies;
3. Use the value function to generate new instances.



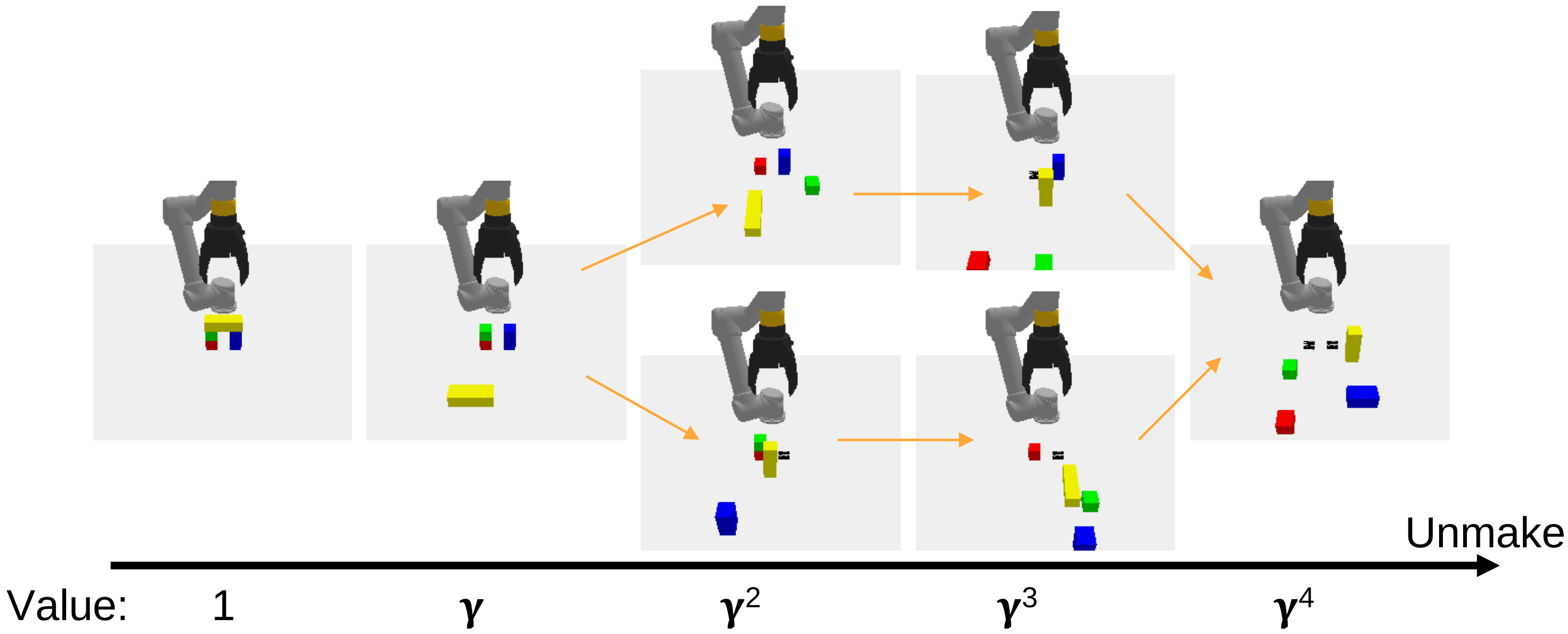
State space: Unmake

$$V : \mathcal{S} \mapsto \mathbb{R} \qquad \hat{V}(s_i) = \gamma^{n-i}$$

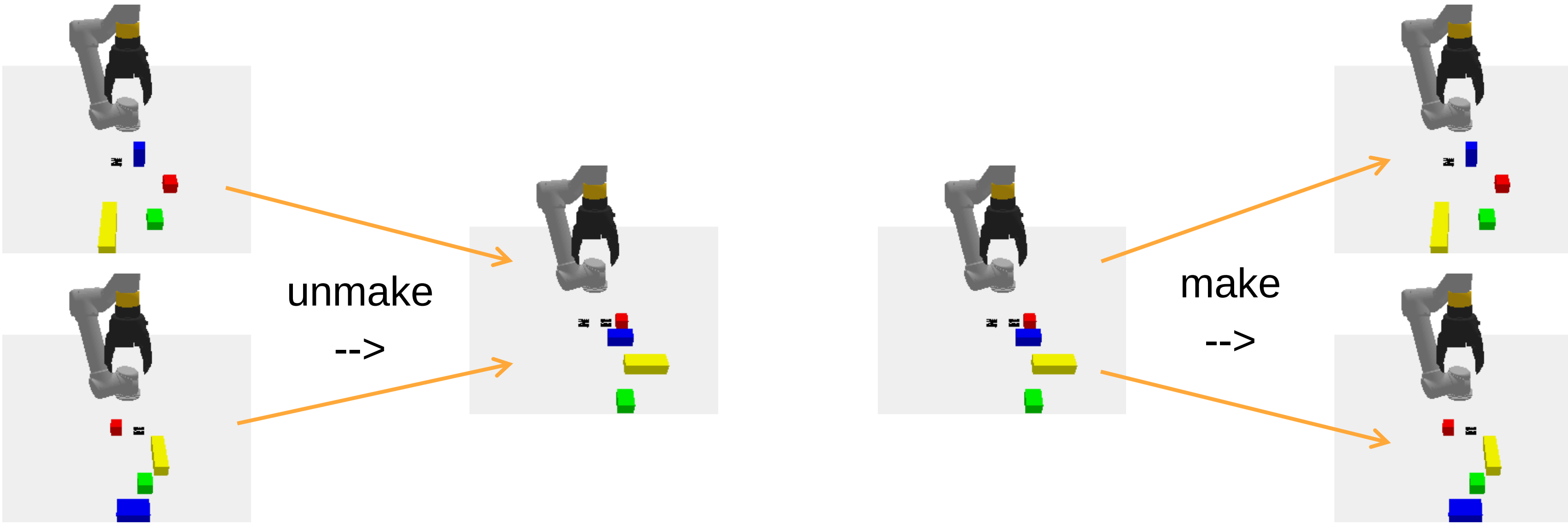


State space: Unmake

$$V : \mathcal{S} \mapsto \mathbb{R} \qquad \hat{V}(s_i) = \gamma^{n-i}$$



State space: Unmake



Learn a value function NN from N disassemblies:

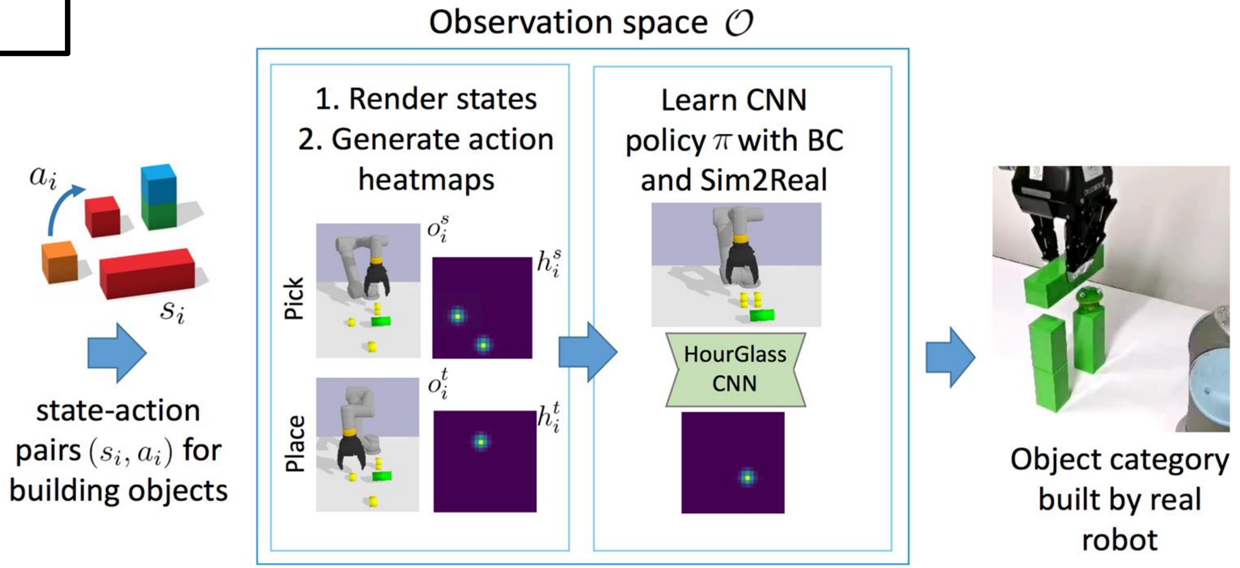
$$\hat{\eta} = \arg\min_{\eta} \mathrm{MSE}(V_{\eta}(s_i), \hat{V}(s_i))$$

Generate state-actions pairs

Learning in observation space

Goal:
- Learn visual policy

1. Render observations from states;
2. Generate heatmaps from actions;
3. Train a hourglass CNN with Behaviour Cloning and sim2real [1].



[1] A. Pashevich, R. Strudel, I. Kalevatykh, I. Laptev, and C. Schmid, “Learning to augment synthetic images for sim2real policy transfer,” IROS 2019.

Building arches with blocks
Height: 3U

Building arches with new primitives
(unseen during training)

Learning Obstacle Representations for Neural Motion Planning

Robin Strudel Ricardo Garcia Justin Carpentier Jean-Paul Laumond

Ivan Laptev Cordelia Schmid

Inria ENS PSL

Conference on Robot Learning

References

- Berkeley Deep RL Course CS 294 by Sergey Levine: http://rail.eecs.berkeley.edu/deeprlcourse-fa17/

- An Algorithmic Perspective on Imitation Learning, 2018
  T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, Y. Peters.

- Allesandro Lazaric RL Course: http://chercheurs.lille.inria.fr/~lazaric/

How to give a talk and write a paper

Slides by Bill Freeman, MIT:

http://www.di.ens.fr/willow/teaching/recvis12/slides/lecture23TalksAndPapers.pdf
http://billf.mit.edu/sites/default/files/documents/cvprPapers.pdf

Lecture notes by Bill Freeman, MIT:

http://www.di.ens.fr/willow/teaching/recvis12/slides/slideNotes23TalksPapers.pdf

Other sources:

http://www.cs.berkeley.edu/~messer/Bad_talk.html
http://www-psych.stanford.edu/~lera/talk.html

No free lunch

The more you work on a talk, the better it gets: if you work on it for 3 hours, the talk you give will be better than if you had only worked on it for 2 hours. If you work on it for 5 hours, it will be better still. 7 hours, better yet…

All talks are important

There are no unimportant talks.

There are no big or small audiences.

Prepare each talk with the same enthusiasm.

How to give a talk

Delivering:

Look at the audience! Try not to talk to your laptop or to the screen. Instead, look at the other humans in the room.

You have to believe in what you present, be confident… even if it only lasts for the time of your presentation.

Do not be afraid to acknowledge limitations of whatever you are presenting. Limitations are good. They leave job for the people to come. Trying to hide the problems in your work will make the preparation of the talk a lot harder and your self confidence will be hurt.

Slides by A. Torralba

The different kinds of talks you'll have to give as a researcher

- 2-5 minute talks
- 20 -30 minute conference presentations
- 30-60 minute colloquia

Very short talks

- Rehearse it.
- Cut things out that aren't essential. You can refer to them at a high level.
- You might focus on answering just a few questions, eg: what is the problem? Why is it interesting? Why is it hard?
- Typically these talks are just little advertisements for a poster or for some other (longer) talk. So you just need to show people that the problem is interesting and that you're fun to talk with.
- These talks can convey important info--note popularity of SIGGRAPH fast forward session.

http://groups.csail.mit.edu/vision/courses/6.869/lectures/lecture23TalksAndPapers.pdf

Slides by W. Freeman

In your talk try answering the following questions

- What problem did you address?
- Why is it interesting?
- Why is it hard?
- What was the key to your approach?
- How well did it work?

http://www.di.ens.fr/willow/teaching/recvis12/slides/lecture23TalksAndPapers.pdf

Slides by W. Freeman

See more at:

Writing papers and giving talks

Bill Freeman

MIT CSAIL

May 2, 2011

http://www.di.ens.fr/willow/teaching/recvis12/slides/lecture23TalksAndPapers.pdf

Sources on writing technical papers

- How to Get Your SIGGRAPH Paper Rejected, Jim Kajiya, SIGGRAPH 1993 Papers Chair, http://www.siggraph.org/publications/instructions/rejected.html
- Ted Adelson's Informal guidelines for writing a paper, 1991. http://www.ai.mit.edu/courses/6.899/papers/ted.htm
- Notes on technical writing, Don Knuth, 1989.

  http://www.ai.mit.edu/courses/6.899/papers/knuthAll.pdf

- What's wrong with these equations, David Mermin, Physics Today, Oct., 1989. http://www.ai.mit.edu/courses/6.899/papers/mermin.pdf
- Ten Simple Rules for Mathematical Writing, Dimitri P. Bertsekas http://www.mit.edu:8001/people/dimitrib/Ten_Rules.html

Slides by A. Torralba