Learning Skills from Demonstrations



Prof. Vincent Sitzmann

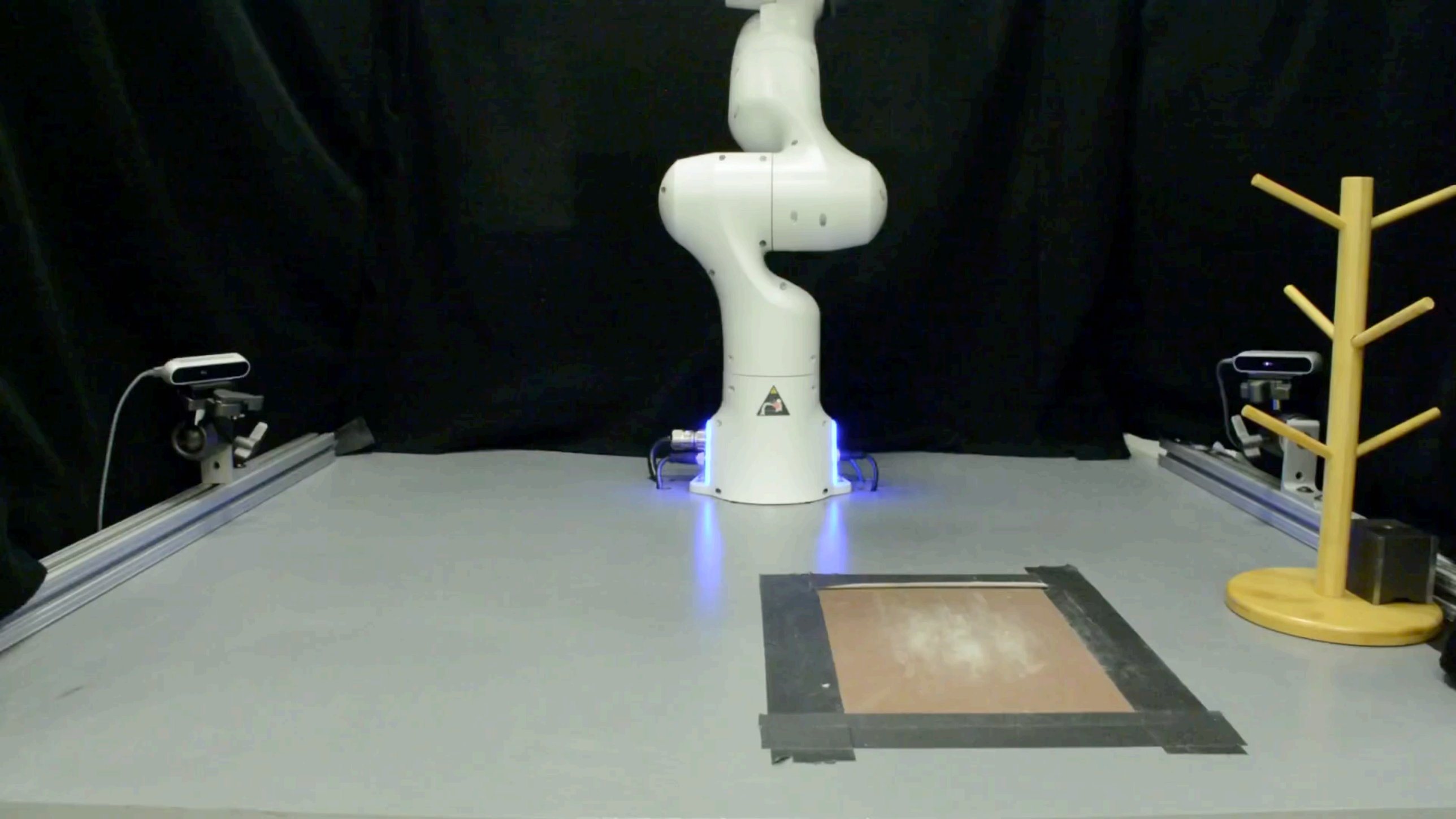
Please write a course evaluation!

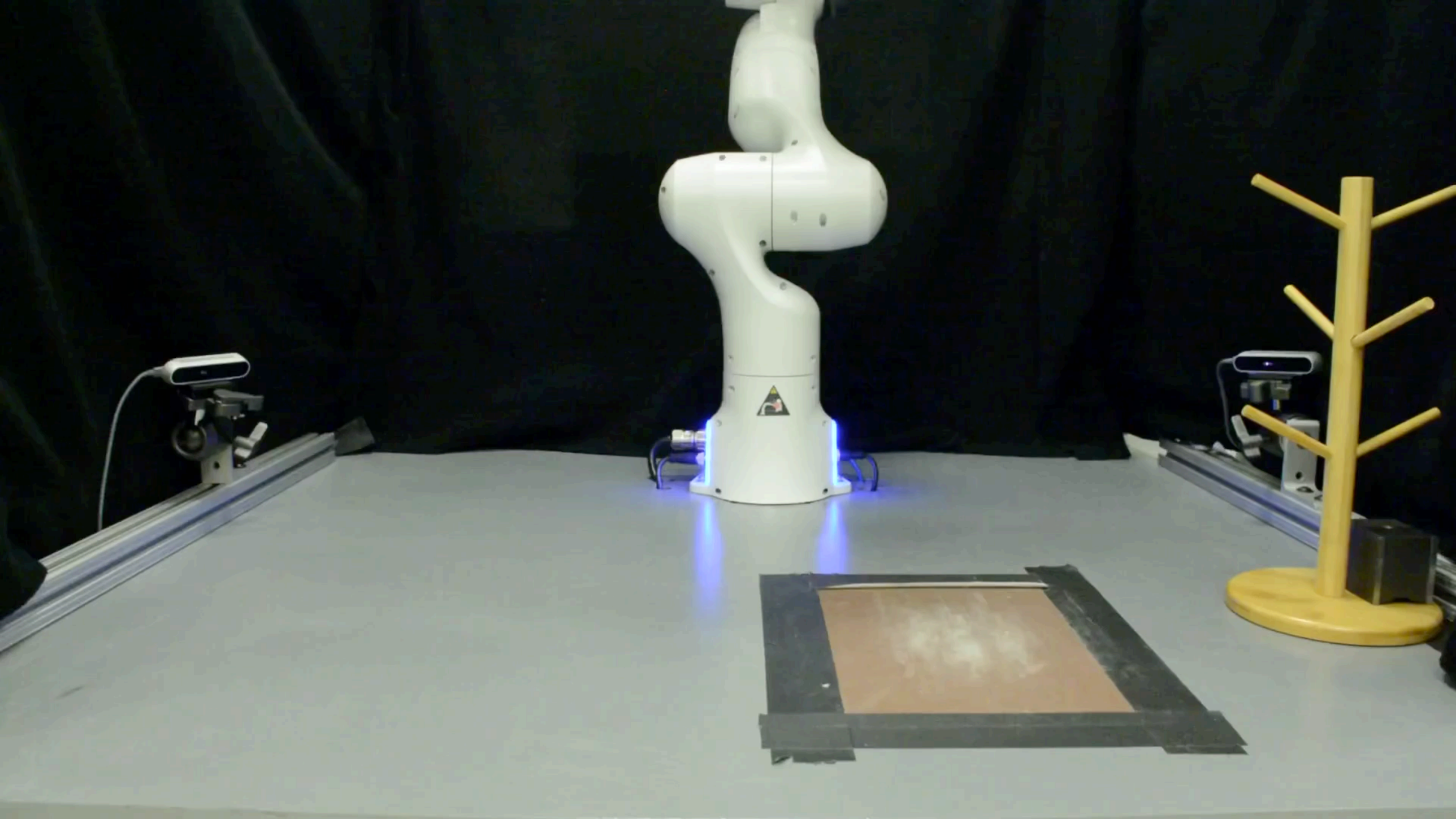
This is the first offering of the course - thanks for taking it!!

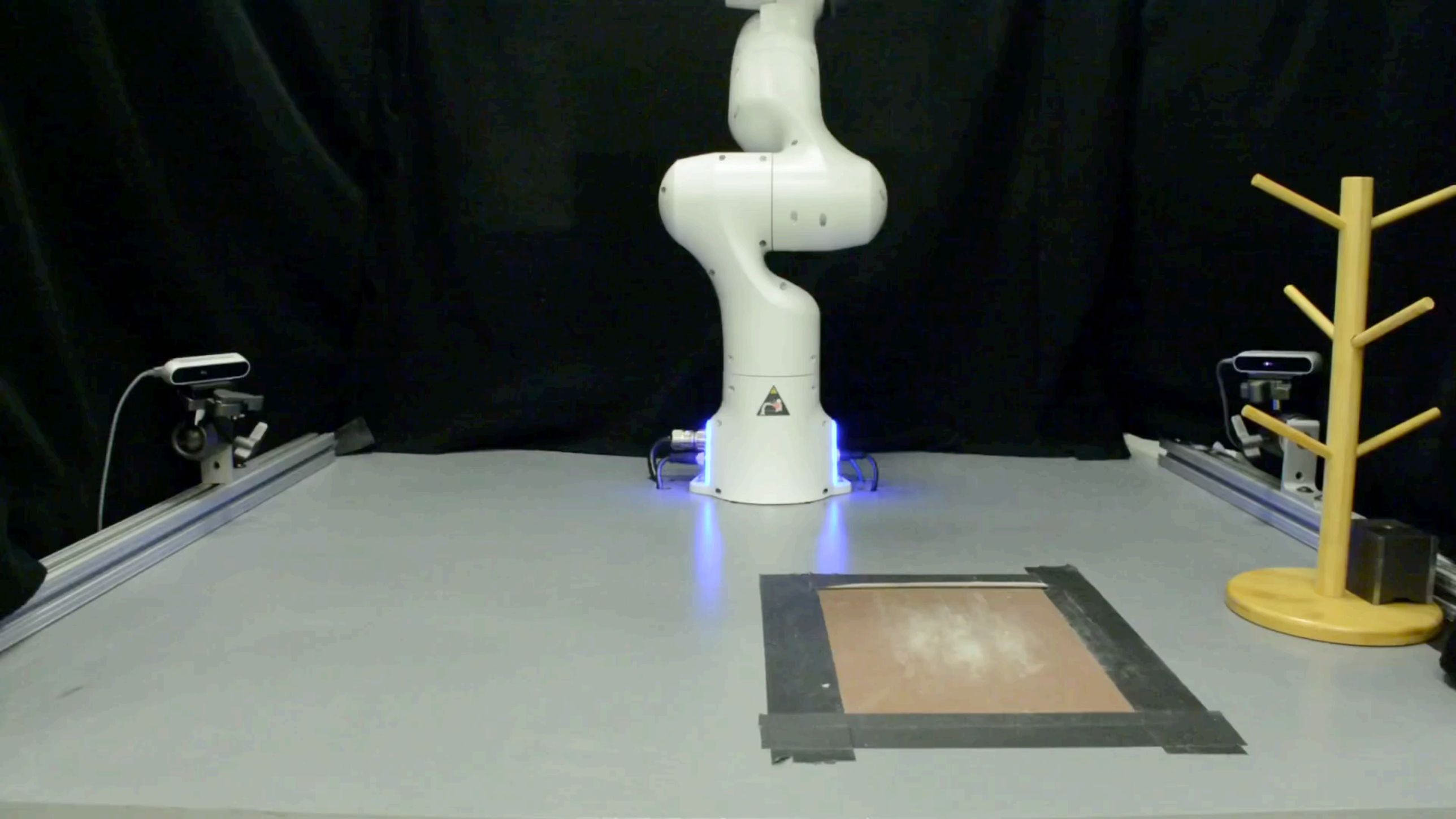
I kept track of lots of things to improve, but I want to know if (1) you liked it (2) what else I could improve!

Evaluate the course here:
https://registrar.mit.edu/subjectevaluation









Goal: Manipulation from a few demonstrations

Supervision

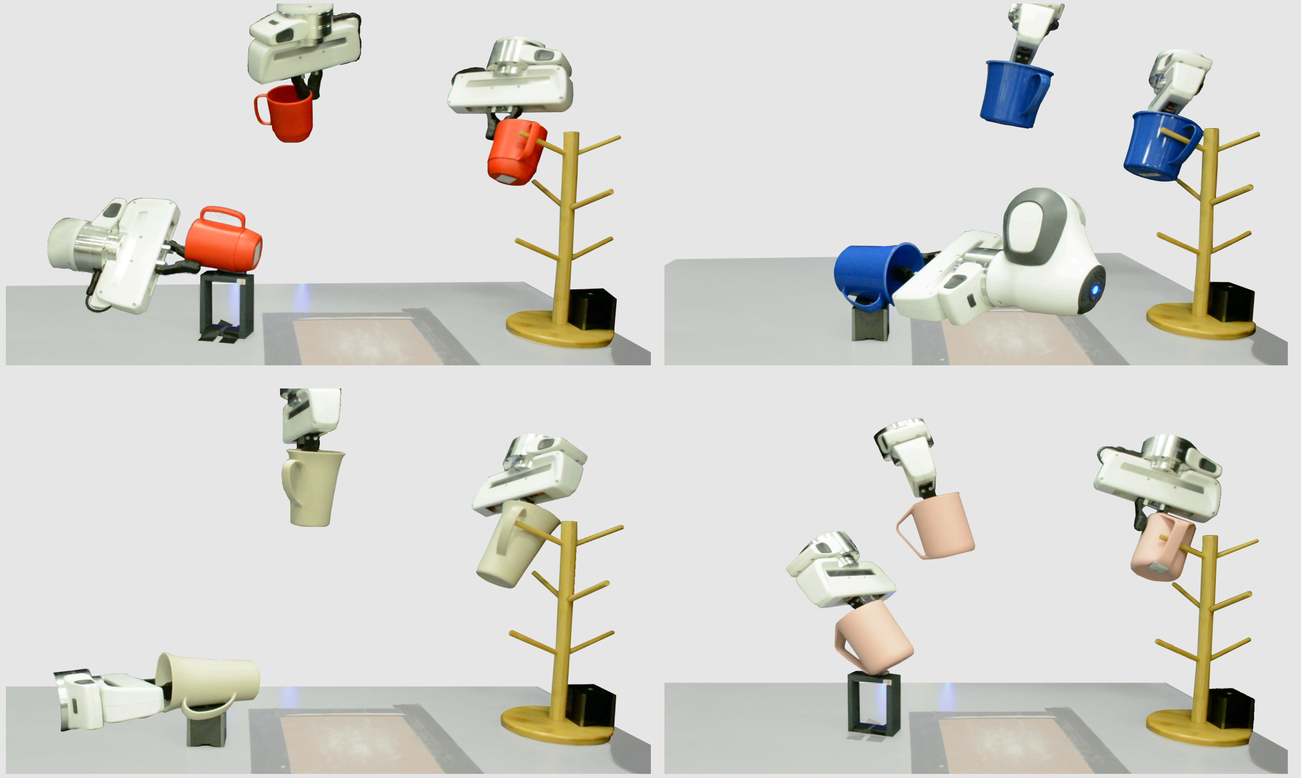
5-10 demonstrations



...

Test cases

Pick-*and*-place unseen objects in unseen poses



Option 1: Hand-crafted imitation learners

Goal: Manipulation from a few demonstrations

Challenges to overcome for generalization

Small number of demonstrations

Unseen object shape instances

Specifying target configurations for placing

Objects can be in arbitrary 6-DoF poses

Test cases

d-place unseen objects in unseen poses

Goal: Manipulation from a few demonstrations

Challenges to overcome for generalization

Small number of demonstrations

Unseen object shape instances

Specifying target configurations for placing

Objects can be in arbitrary 6-DoF poses

Test cases

d-place unseen objects in unseen poses

Goal: Manipulation from a few demonstrations

Challenges to overcome for generalization

Small number of demonstrations ✓

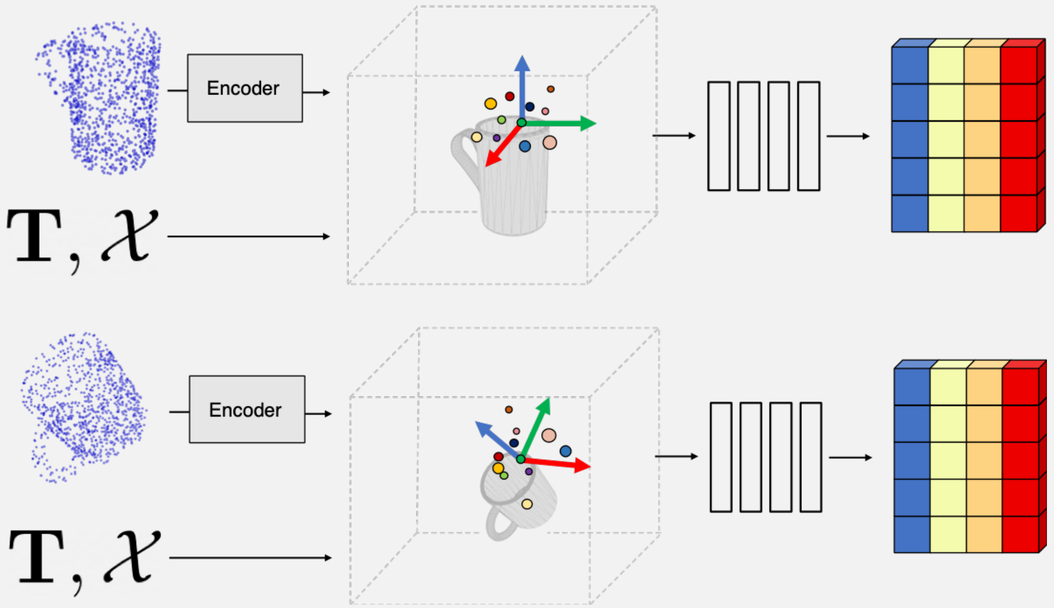
Unseen object shape instances ✓

Specifying target configurations for placing ✓

Objects can be in arbitrary 6-DoF poses ✓

Ours: Neural Descriptor Fields

Represent objects as continuous 3D fields of descriptors



Goal: Manipulation from a few demonstrations

Challenges to overcome for generalization

Small number of demonstrations ✓

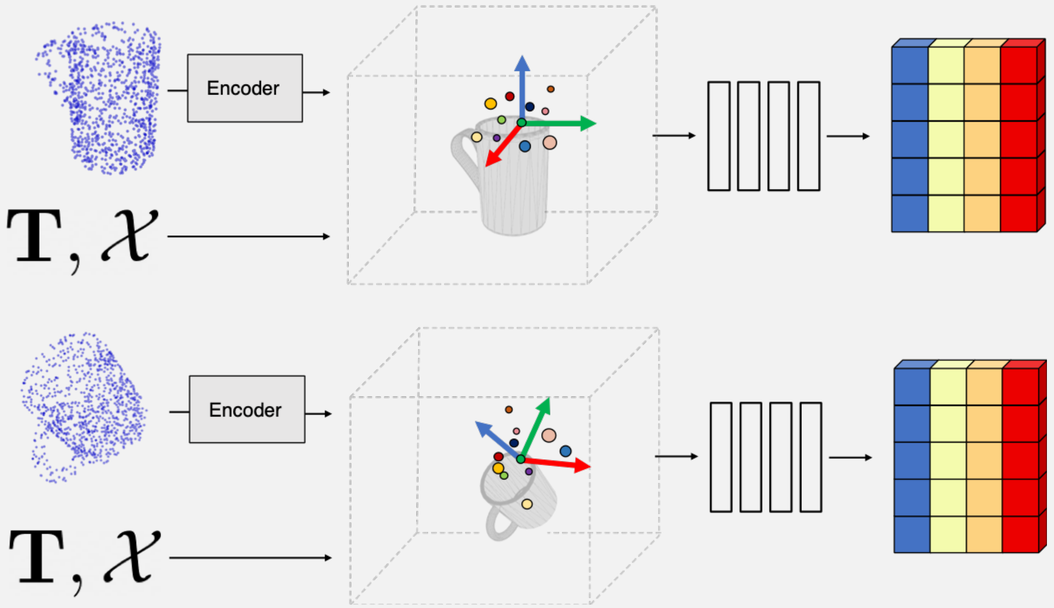
Unseen object shape instances ✓

Specifying target configurations for placing ✓

Objects can be in arbitrary 6-DoF poses ✓

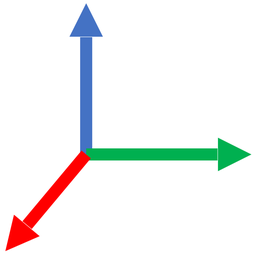
Ours: Neural Descriptor Fields

Represent objects as continuous 3D fields of descriptors



Simpler task: Identical shapes





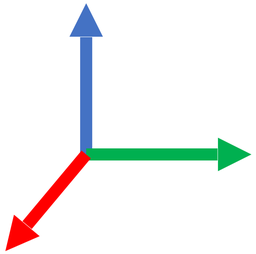
Demonstration



Test

Simpler task: Identical shapes





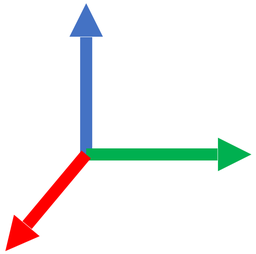
Demonstration



Test

Simpler task: Identical shapes

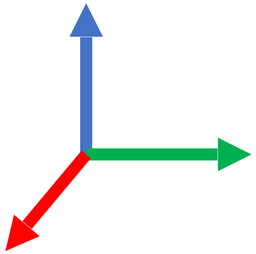






Simpler task: Identical shapes



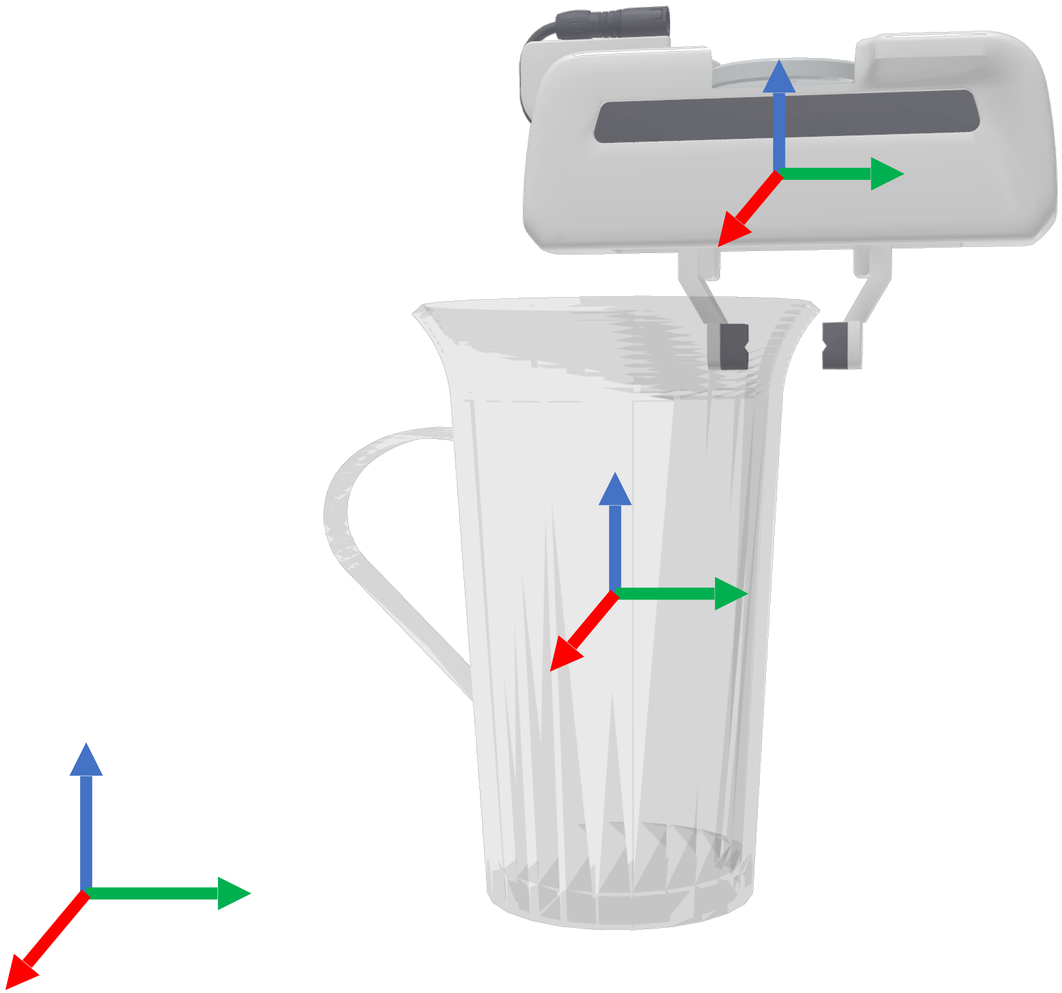




Simpler task: Identical shapes

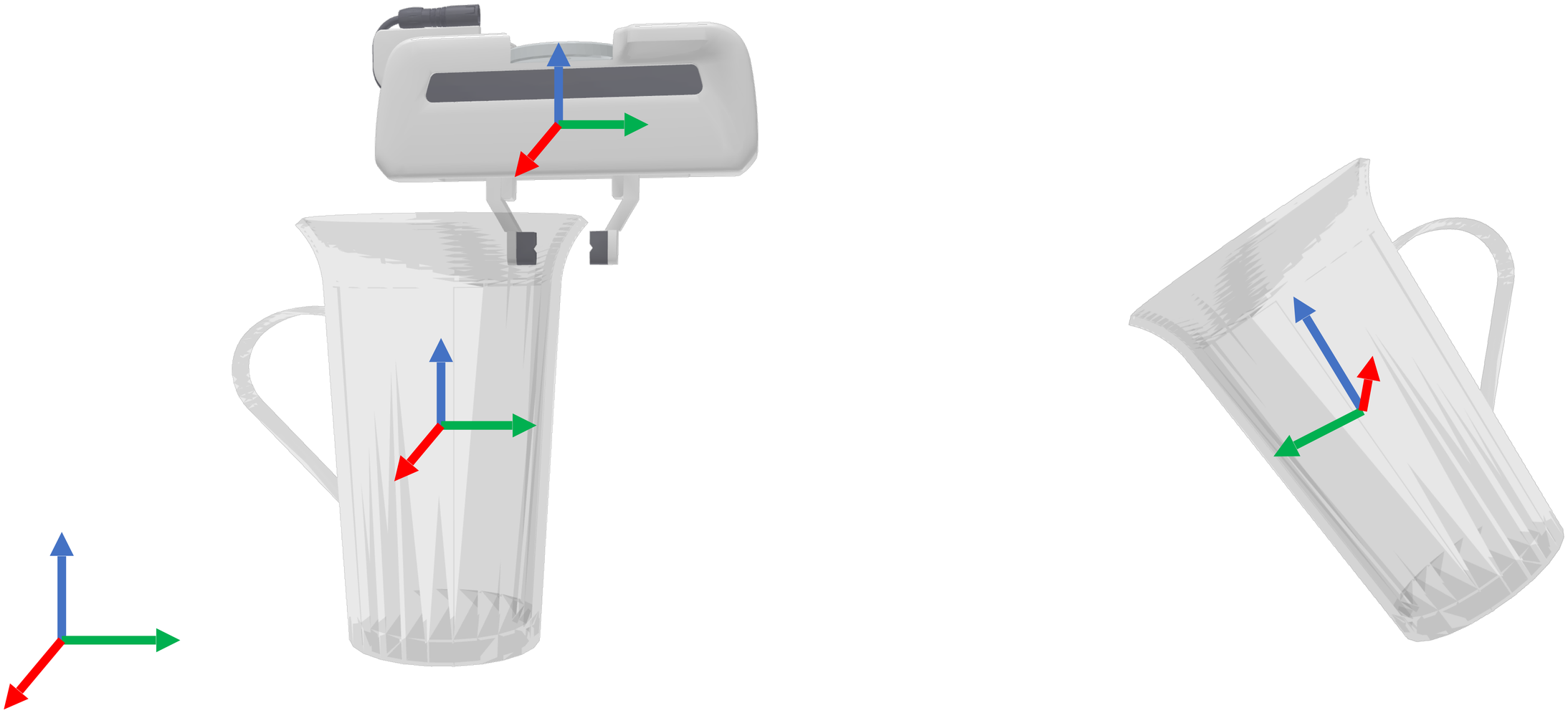


Simpler task: Identical shapes

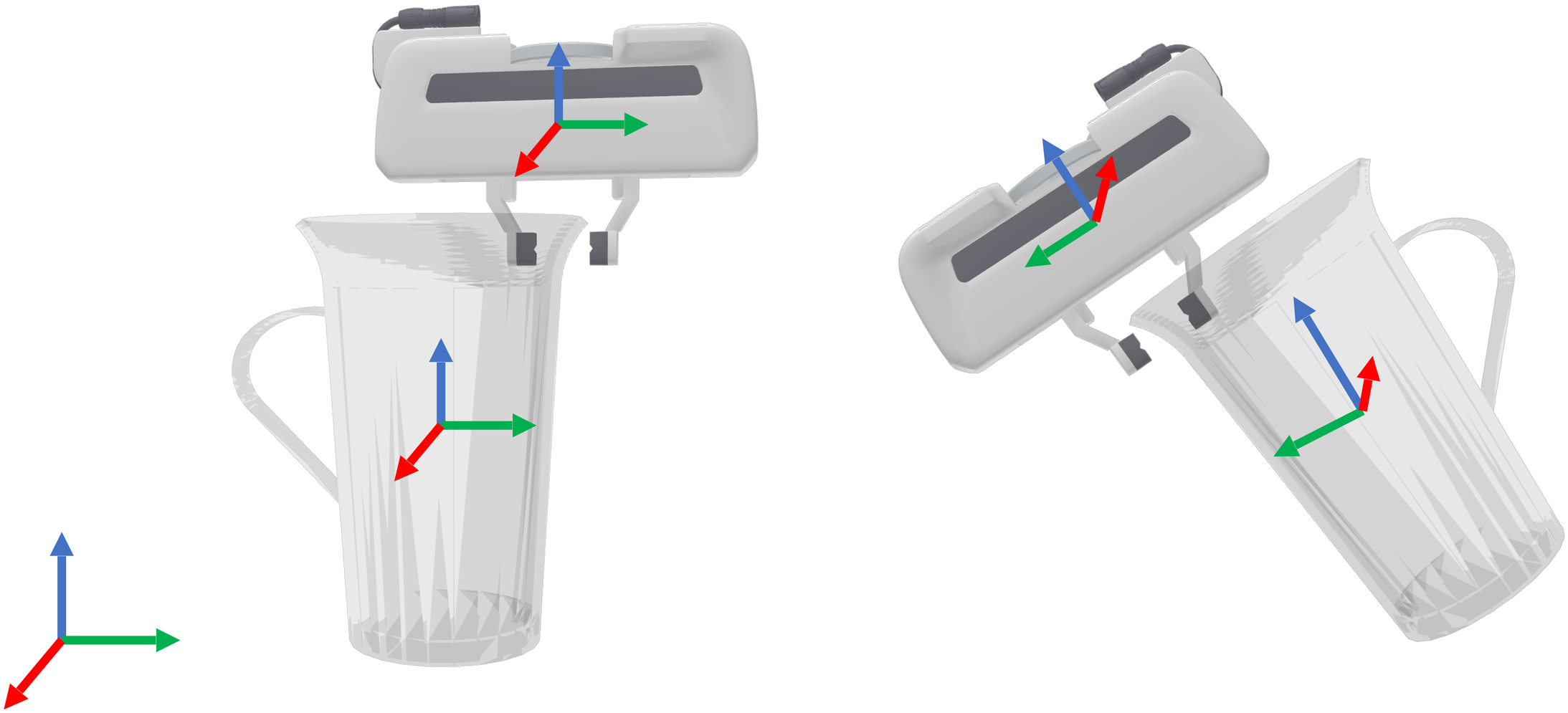




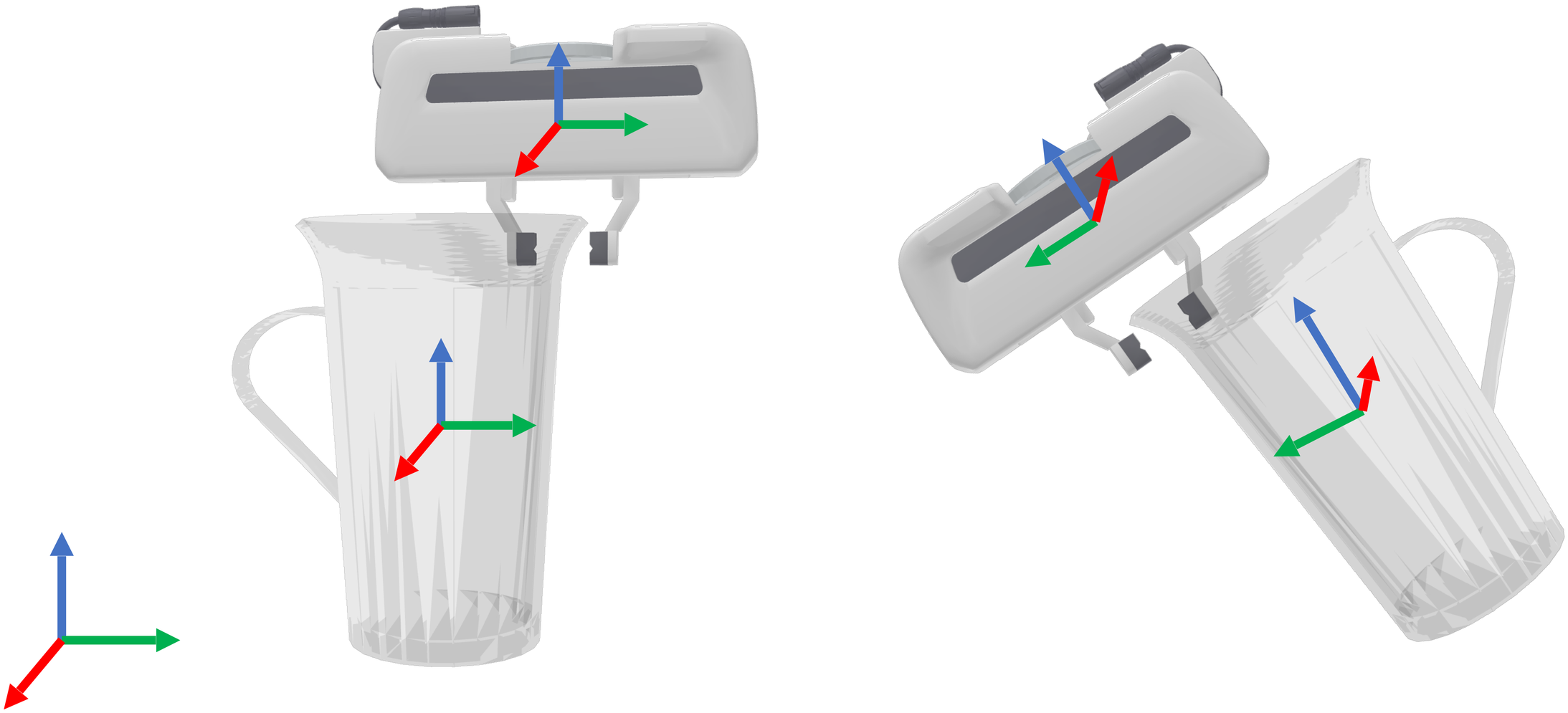
Simpler task: Identical shapes



Simpler task: Identical shapes



Simpler task: Identical shapes



Simpler task: Identical shapes



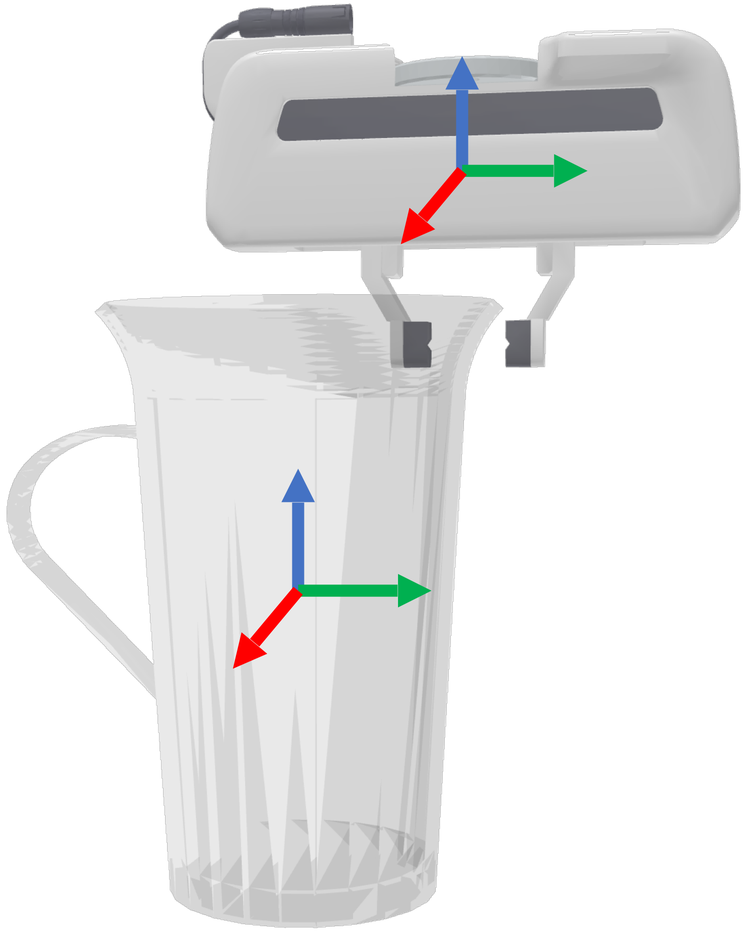


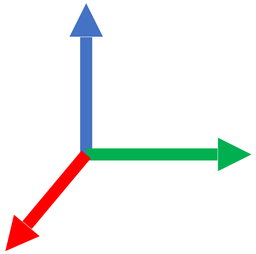
Simpler task: Identical shapes





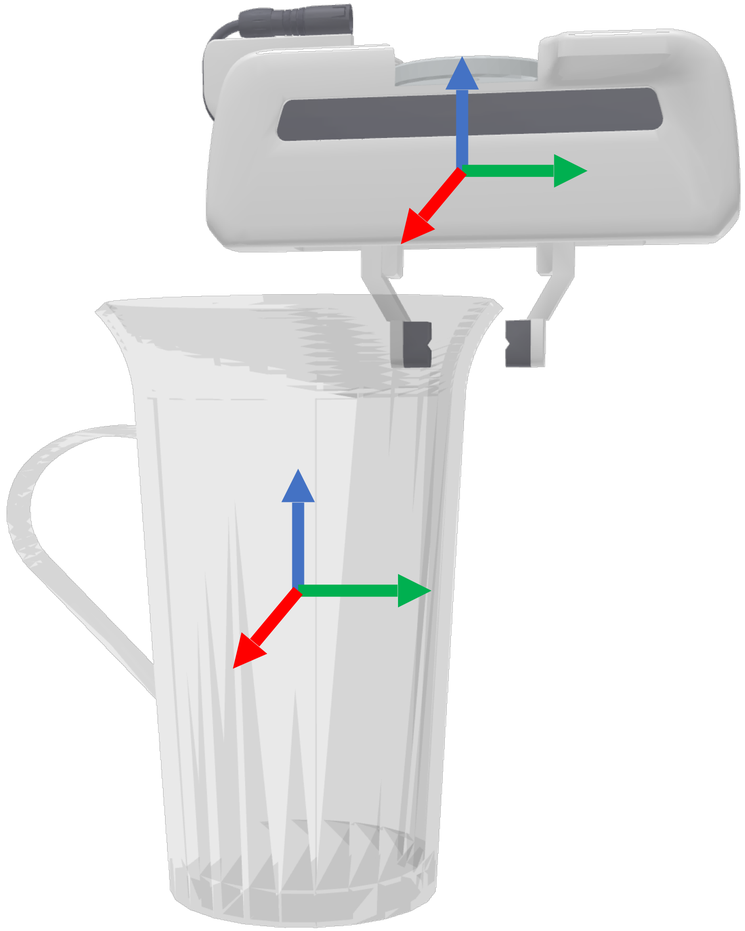
Different shapes: same procedure **fails**

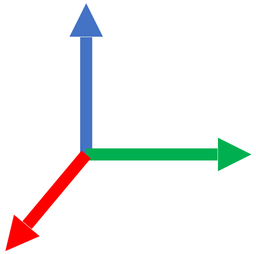






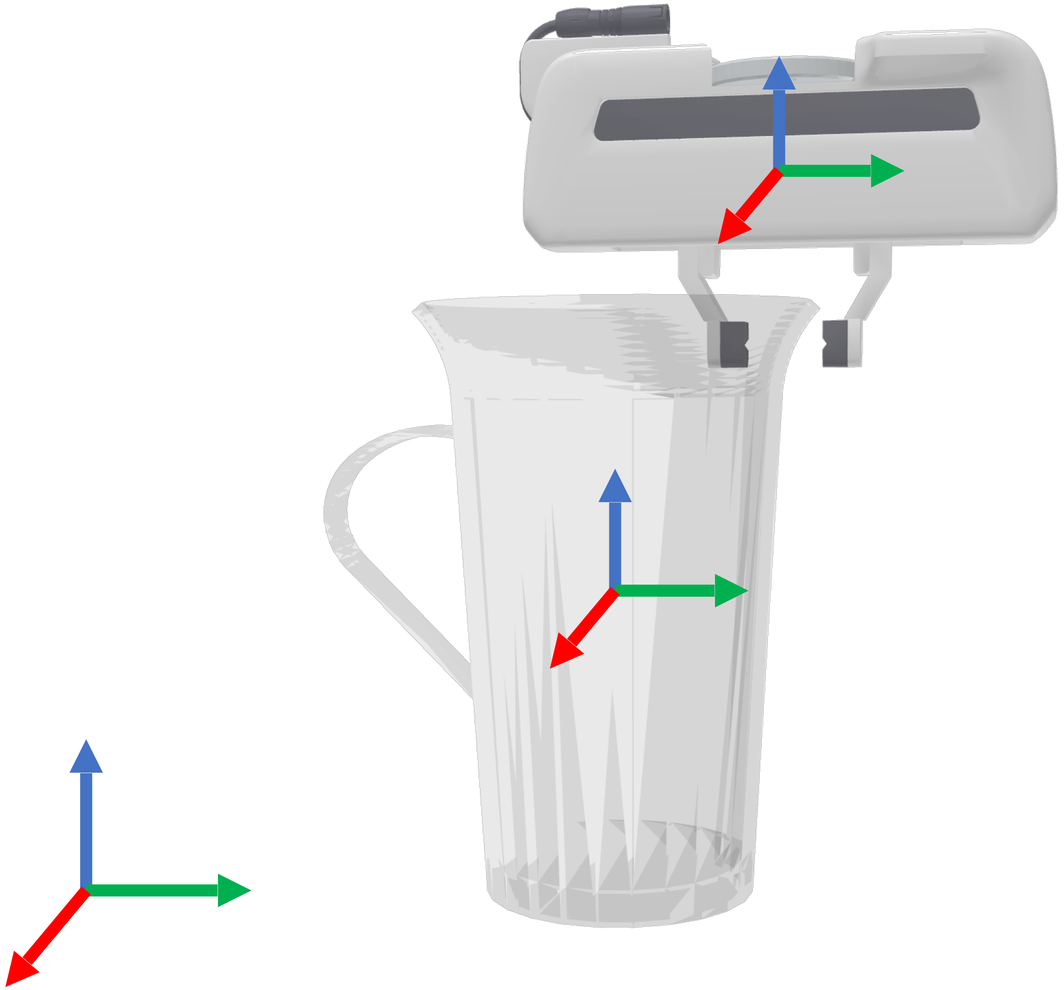
Different shapes: same procedure **fails**

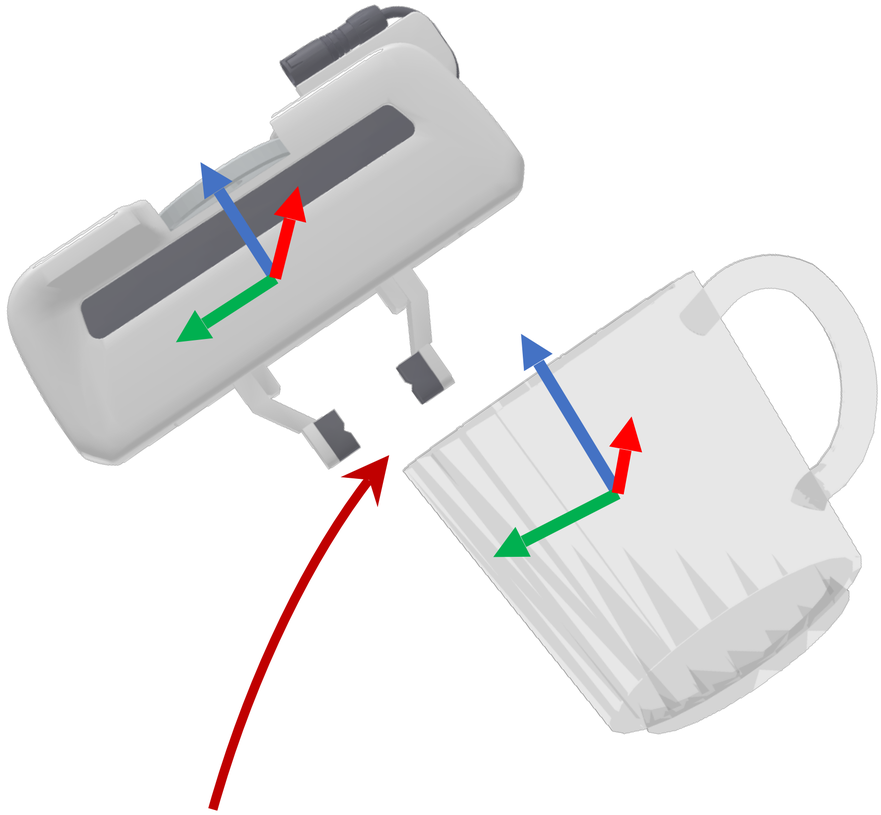






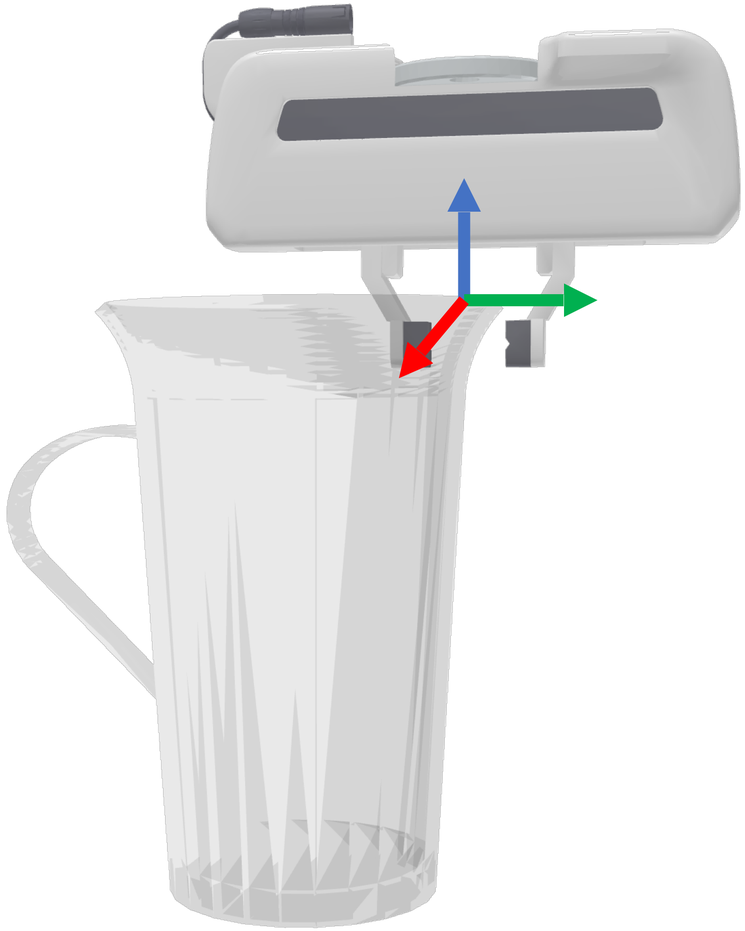
Different shapes: same procedure **fails**

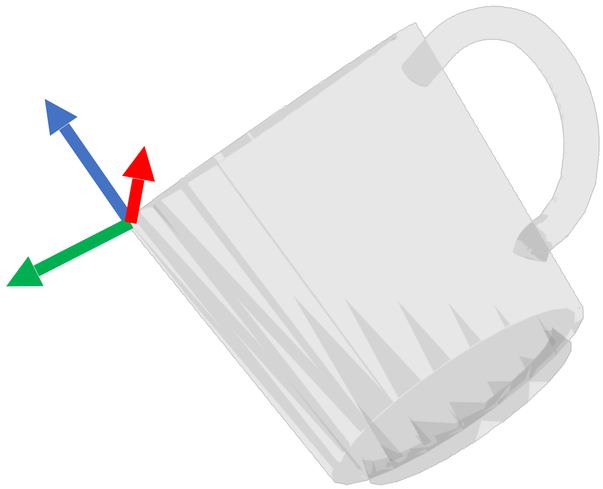




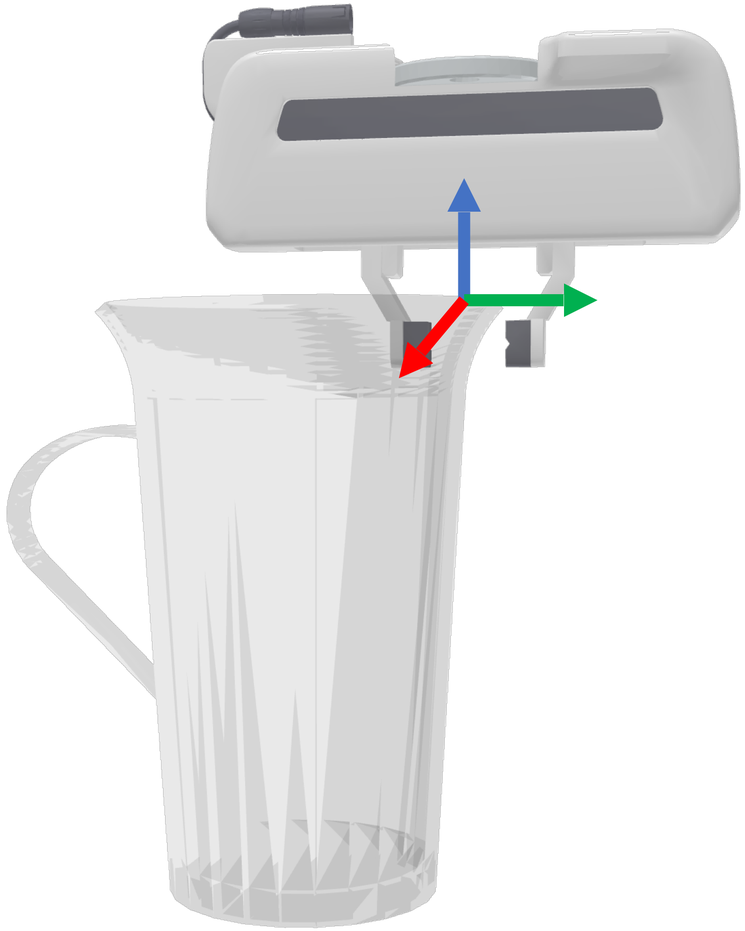
Failed grasp!

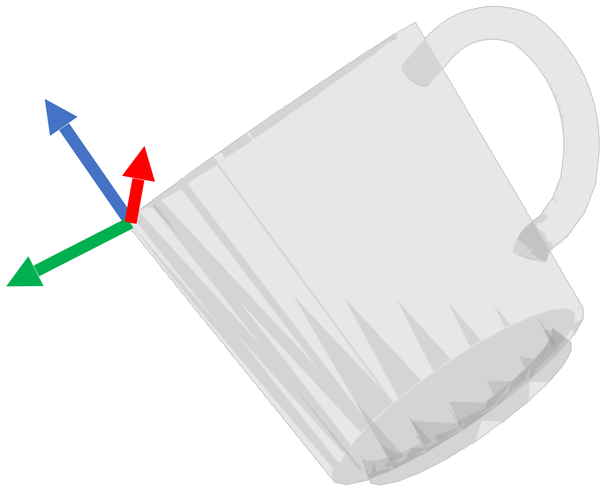
Need **local** coordinate frame at **task-relevant** geometric feature



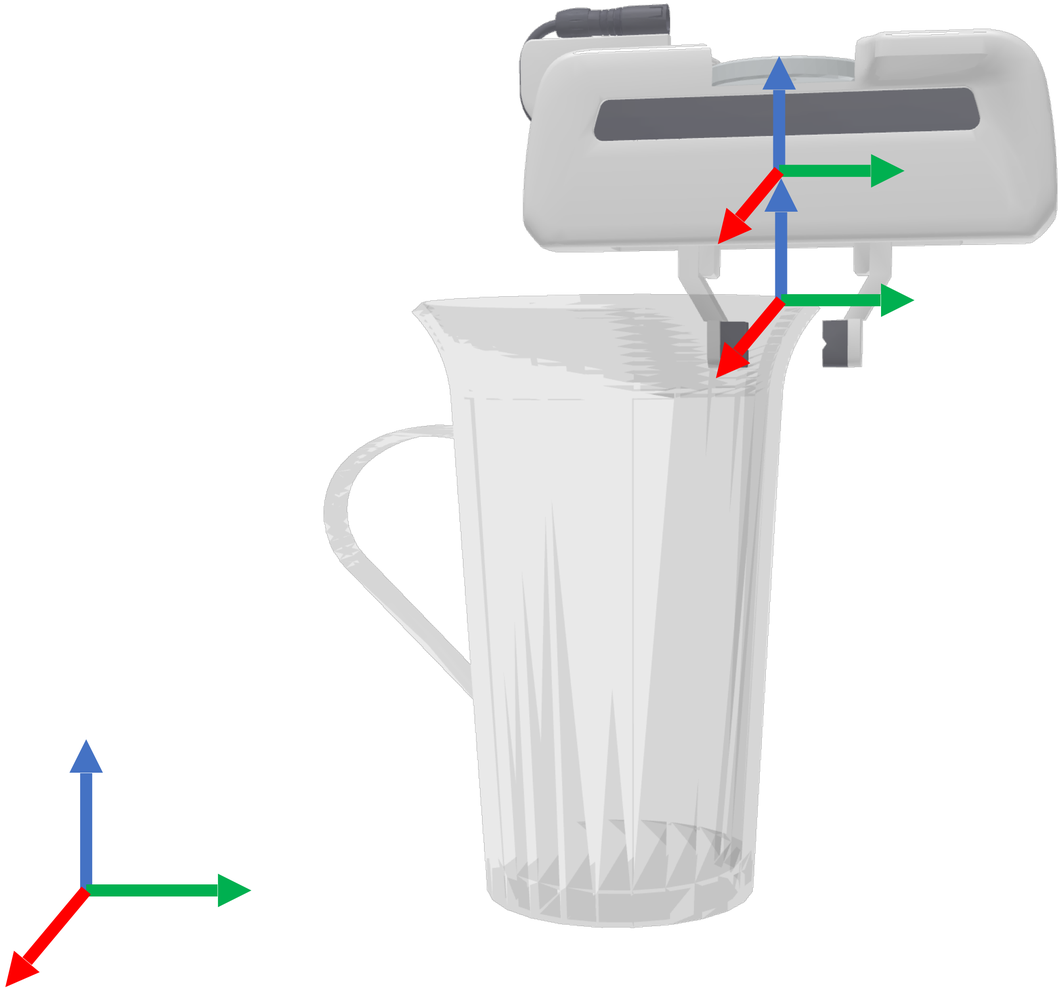


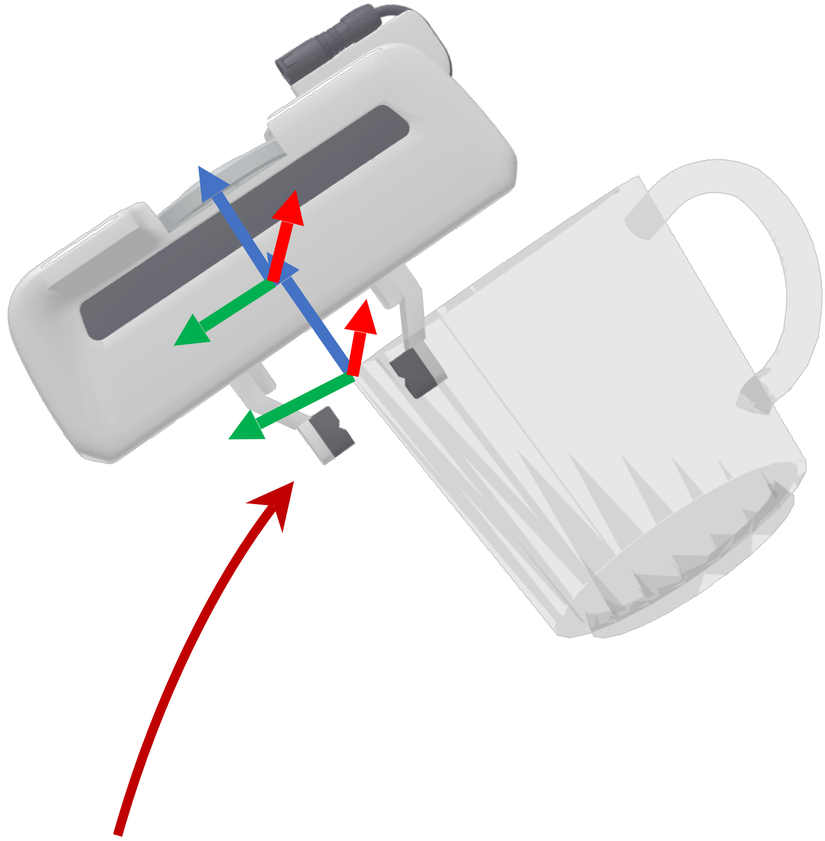
Need **local** coordinate frame at **task-relevant** geometric feature





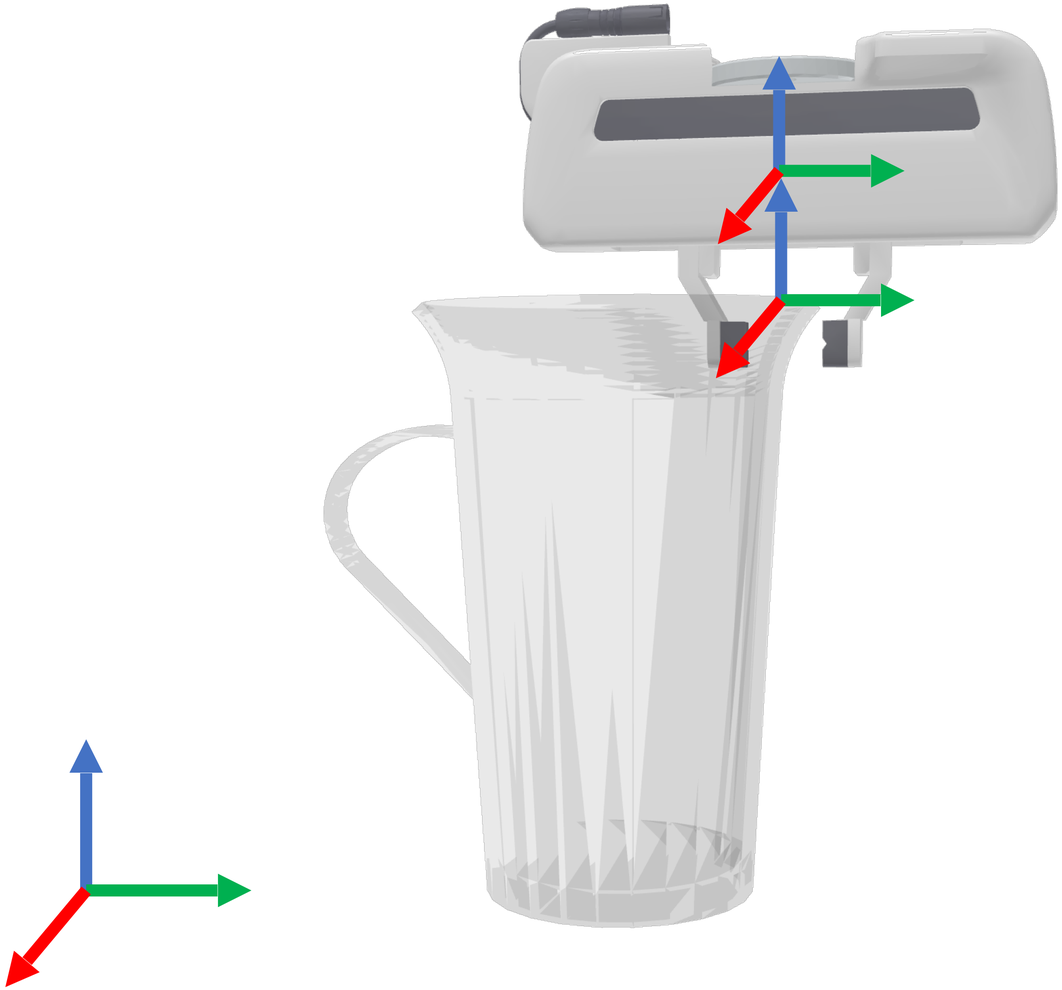
Need **local** coordinate frame at **task-relevant** geometric feature

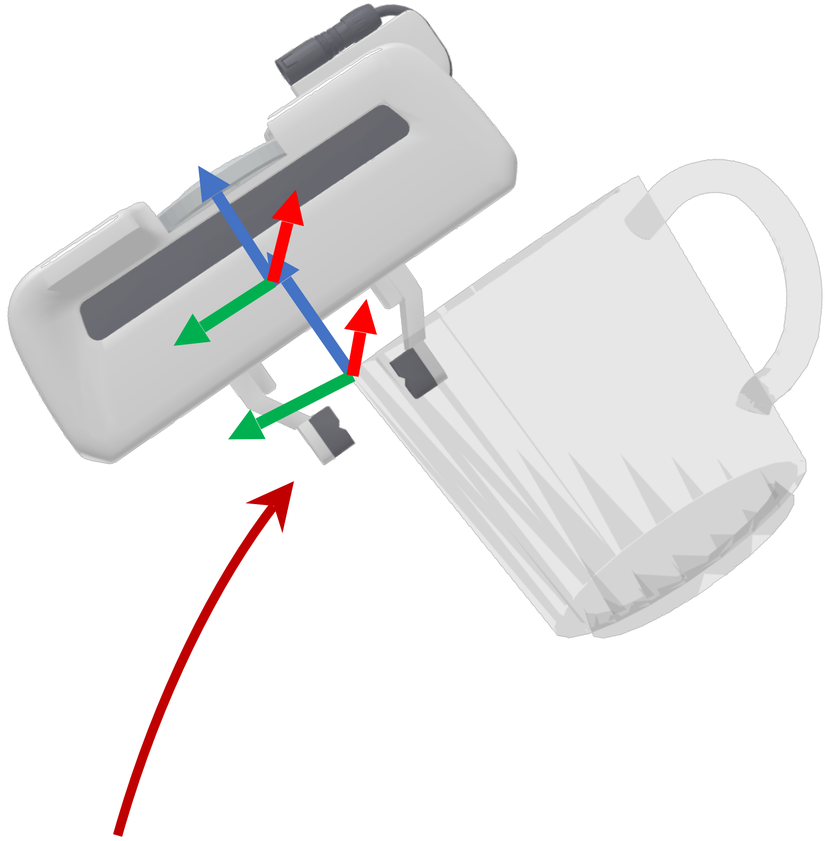




Grasp succeeds

Need **local** coordinate frame at **task-relevant** geometric feature





Grasp succeeds

Need **local** coordinate frame at **task-relevant** geometric feature





Need **local** coordinate frame at **task-relevant** geometric feature





Need **local** coordinate frame at **task-relevant** geometric feature



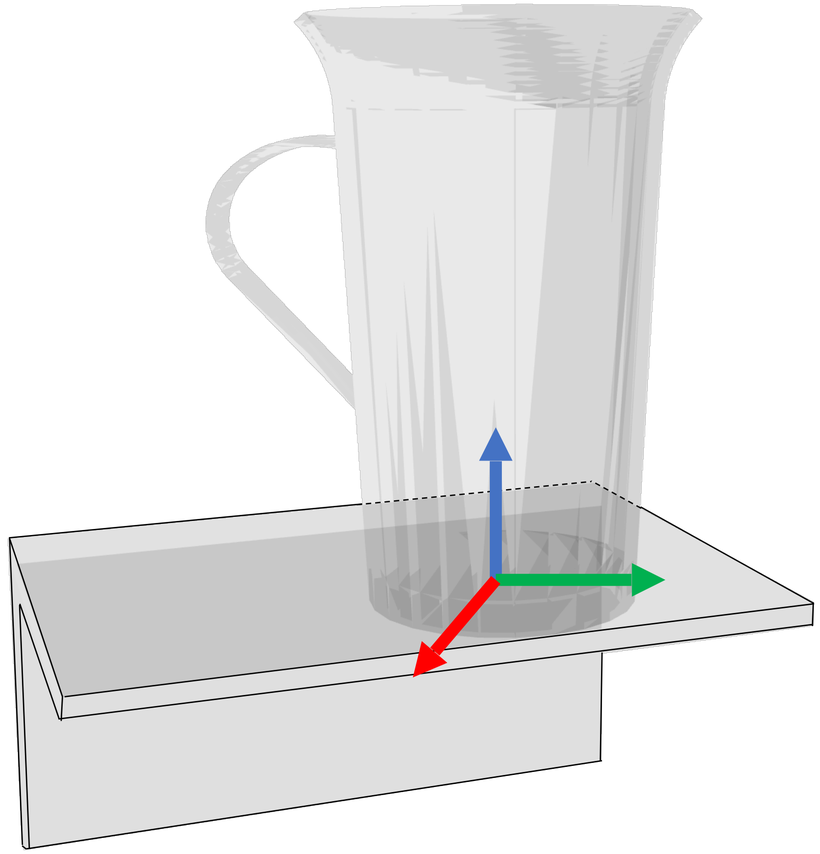


Need **local** coordinate frame at **task-relevant** geometric feature



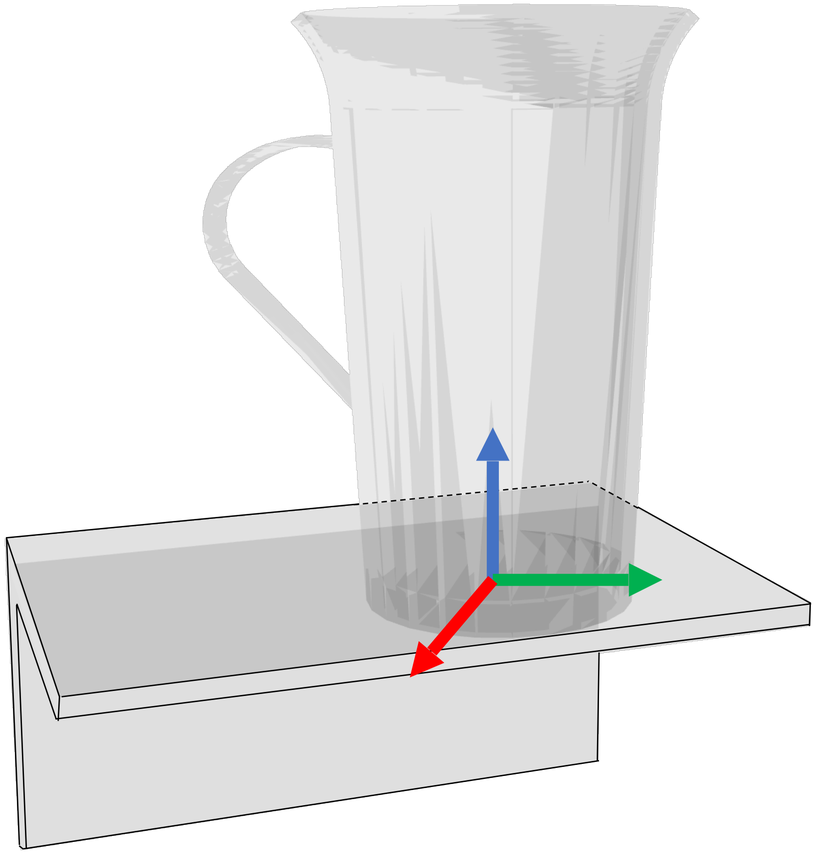


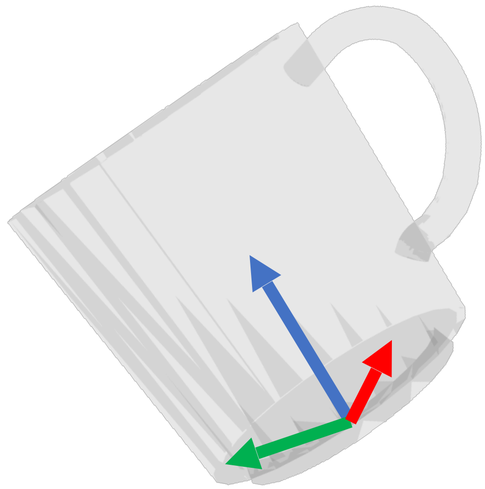
Need **local** coordinate frame at **task-relevant** geometric feature



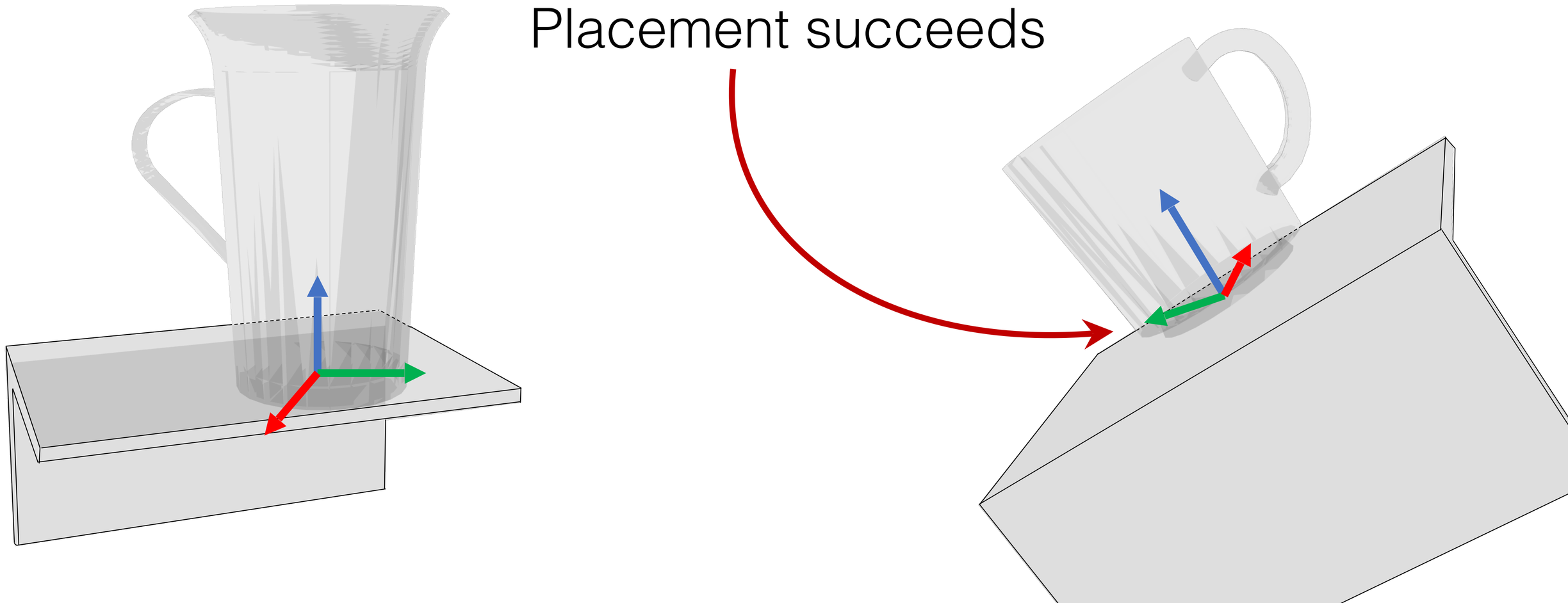


Need **local** coordinate frame at **task-relevant** geometric feature





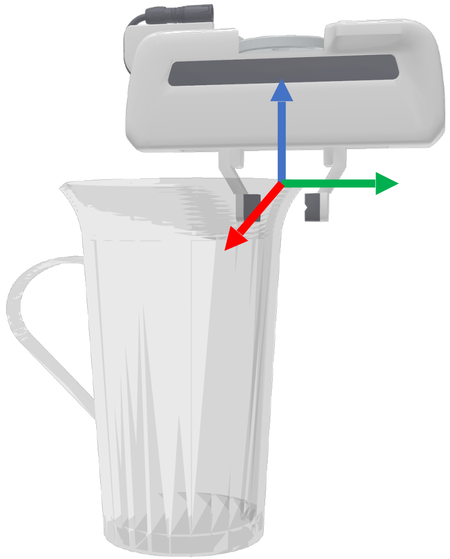
Need **local** coordinate frame at **task-relevant** geometric feature

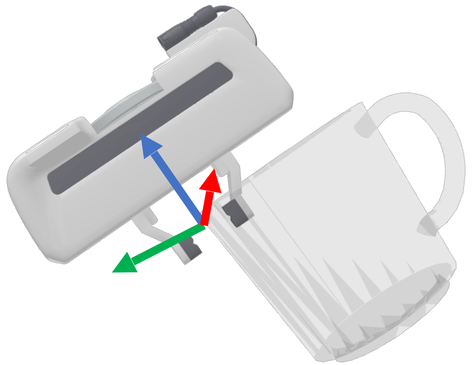


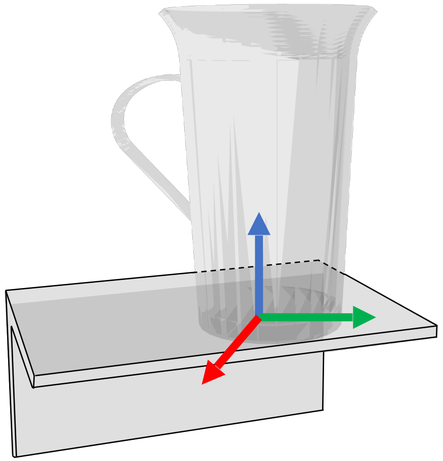
Desiderata:

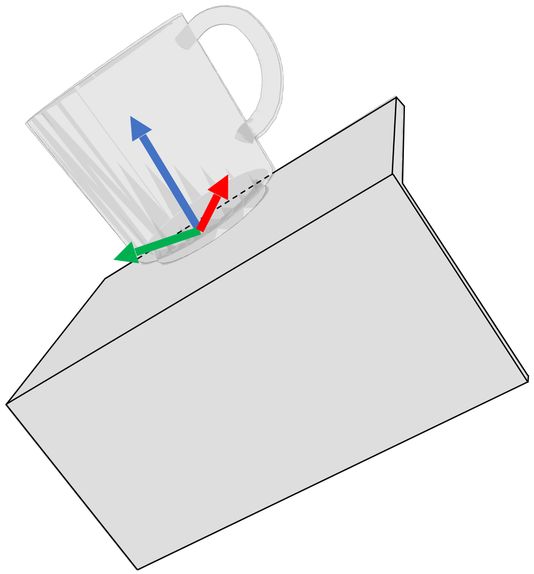
Correspond **any** local coordinate frame across instances

Use demonstrations to obtain **specific, task-relevant** coordinate frame





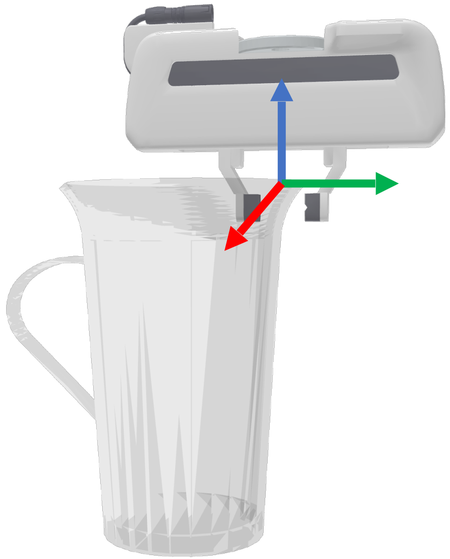


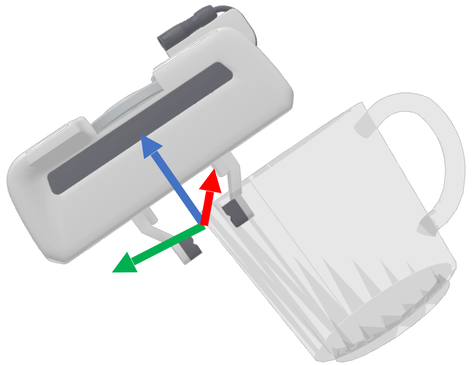


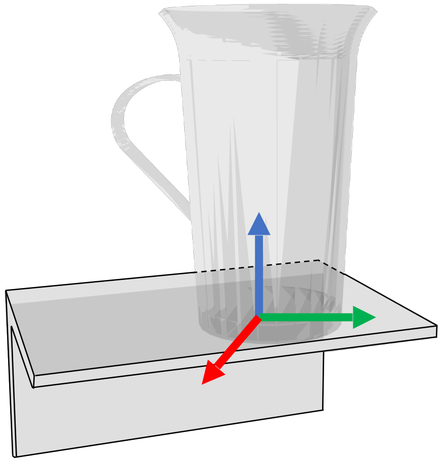
Desiderata:

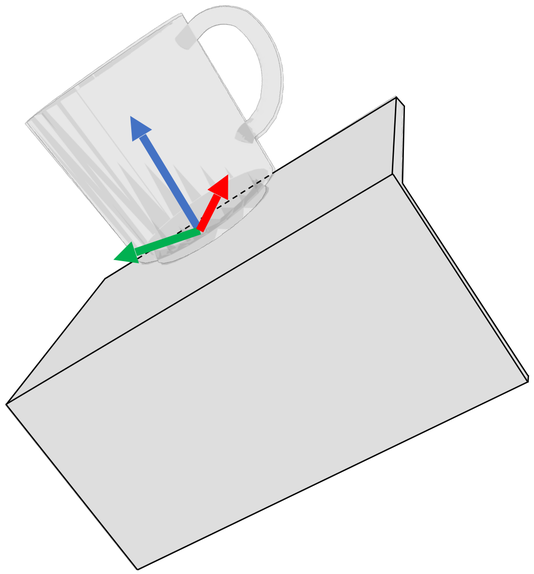
Correspond **any** local coordinate frame across instances

Use demonstrations to obtain **specific, task-relevant** coordinate frame







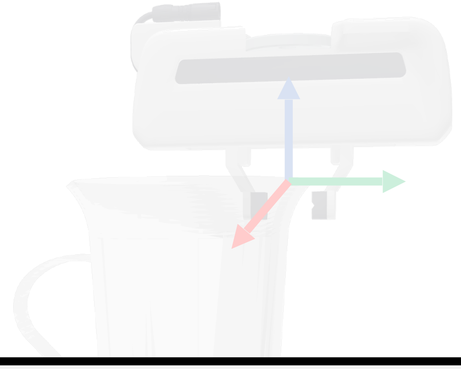


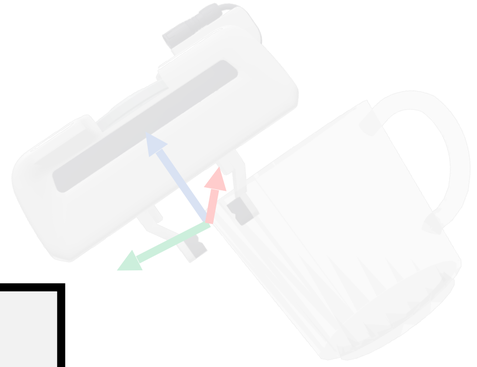
Desiderata:

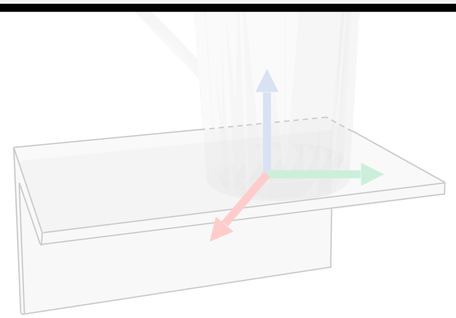
Correspond **any** local co
across instances

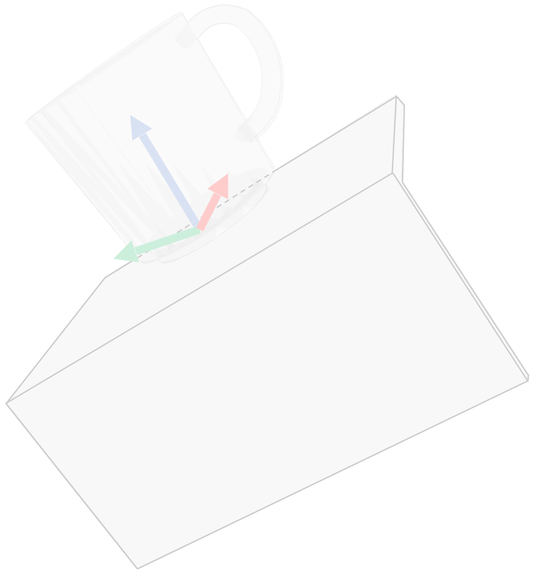
Use demonstrations to ob
task-relevant coordinate

Neural Descriptor Fields







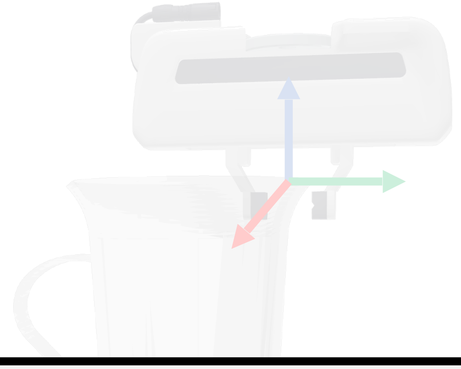


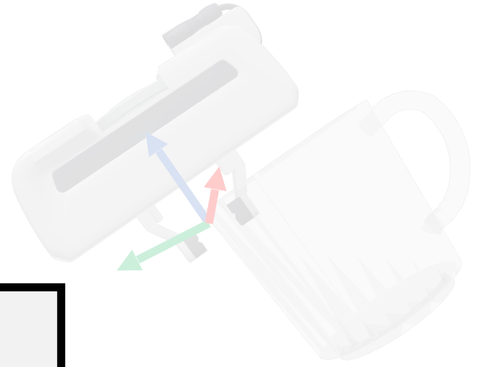
Desiderata:

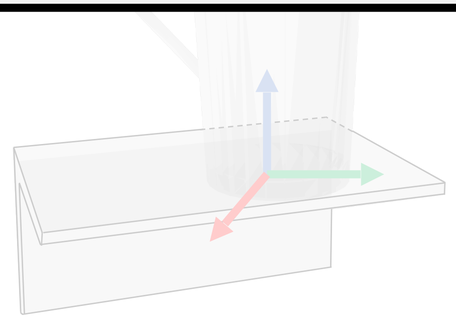
Correspond **any** local co
across instances

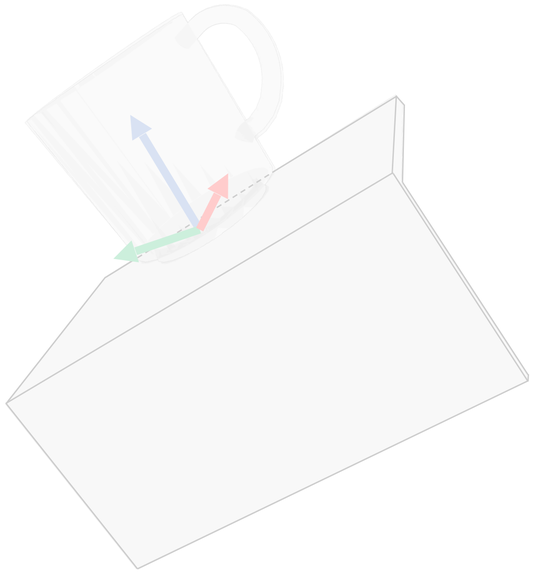
Use demonstrations to ob
task-relevant coordinate

Neural Descriptor Fields

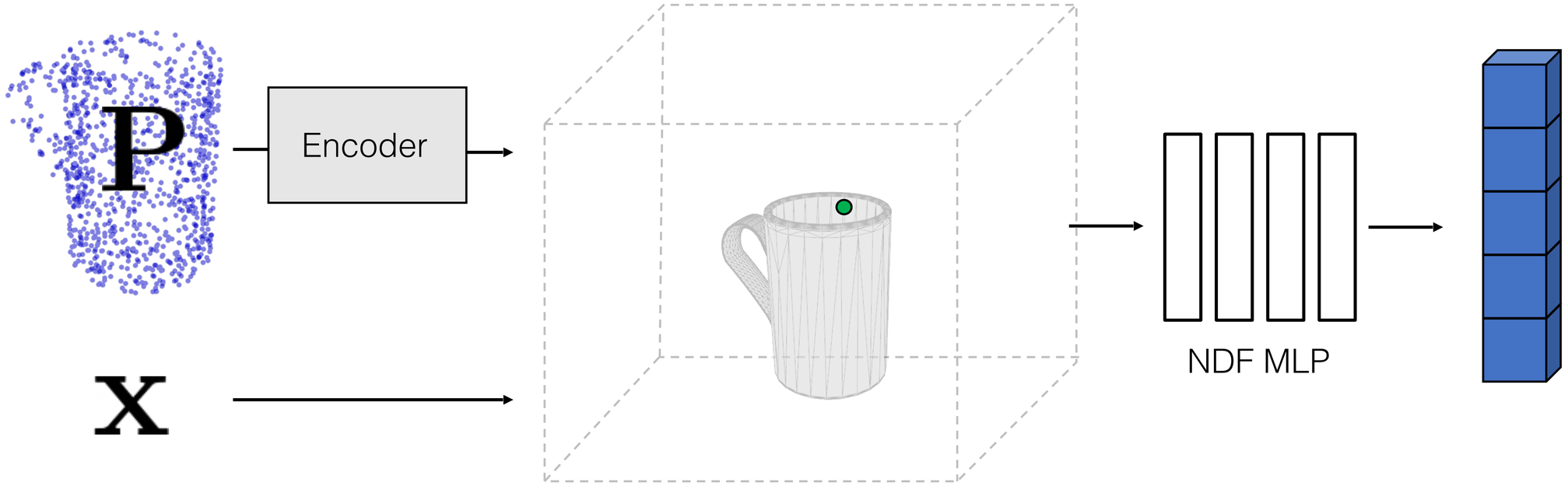






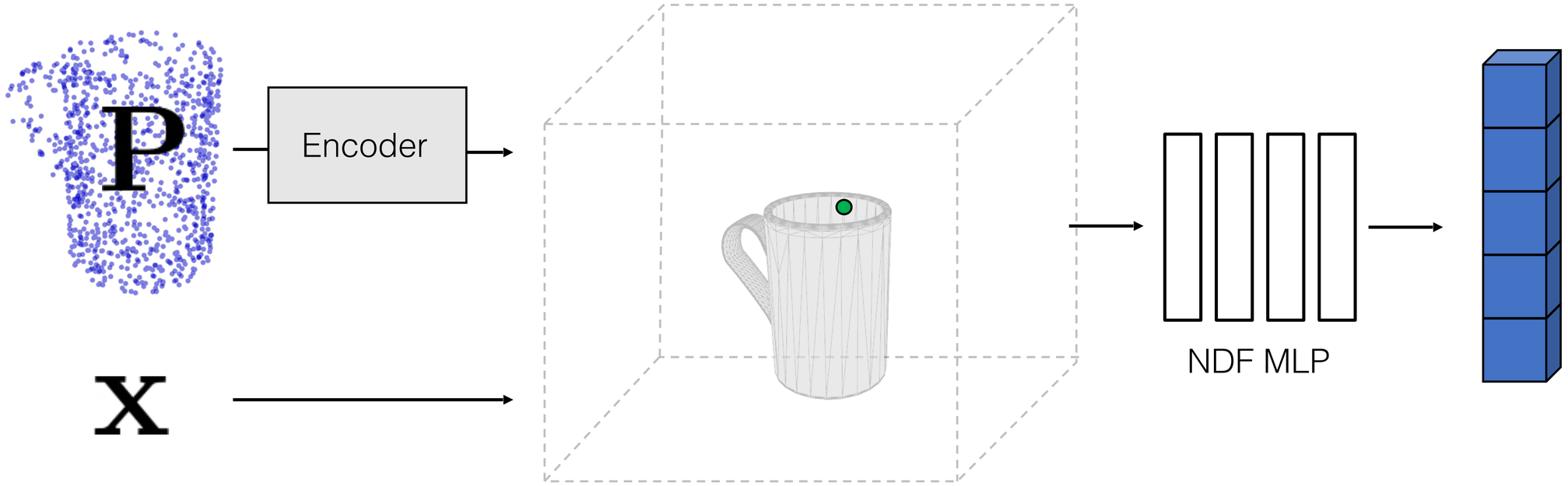


Neural Point Descriptor Fields



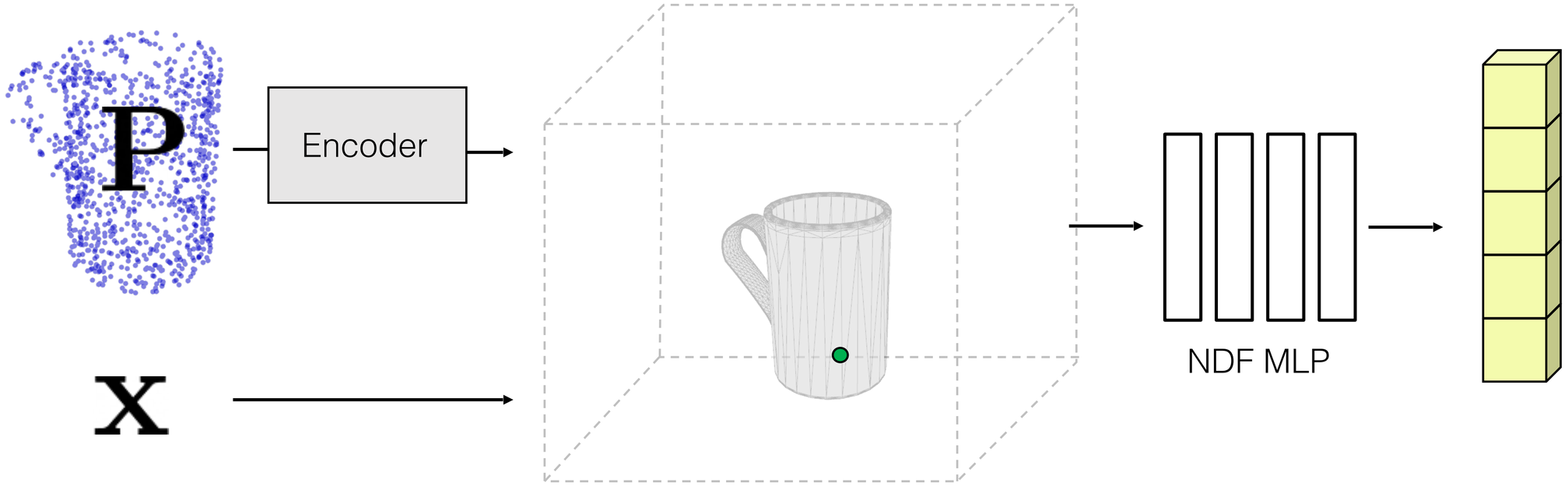
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



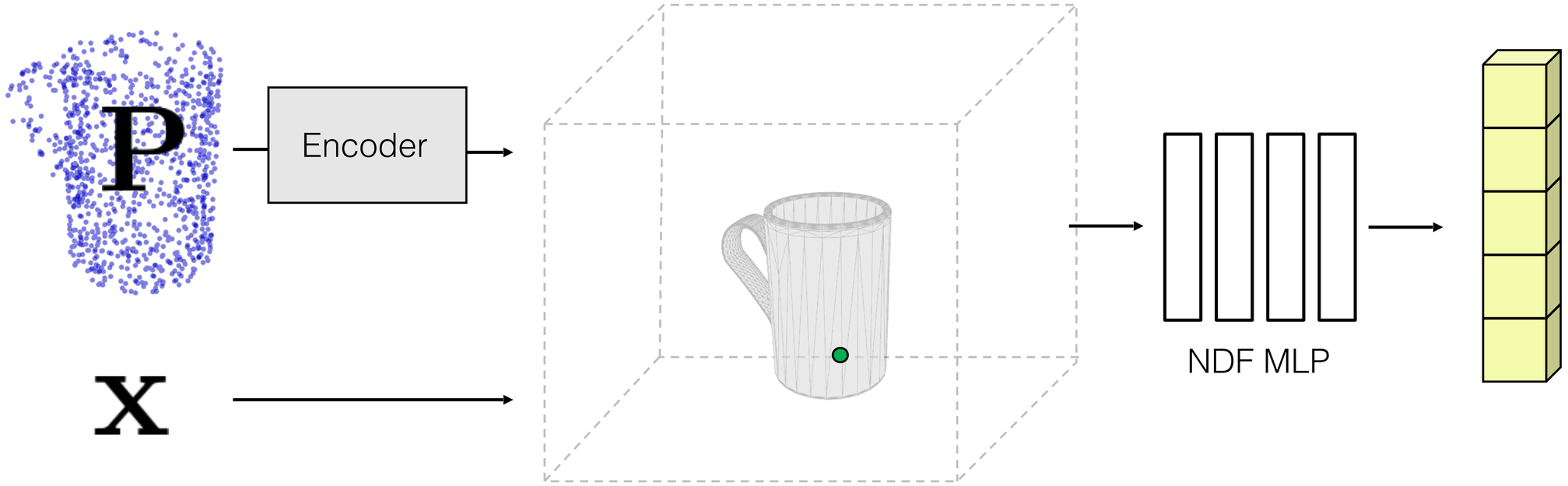
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



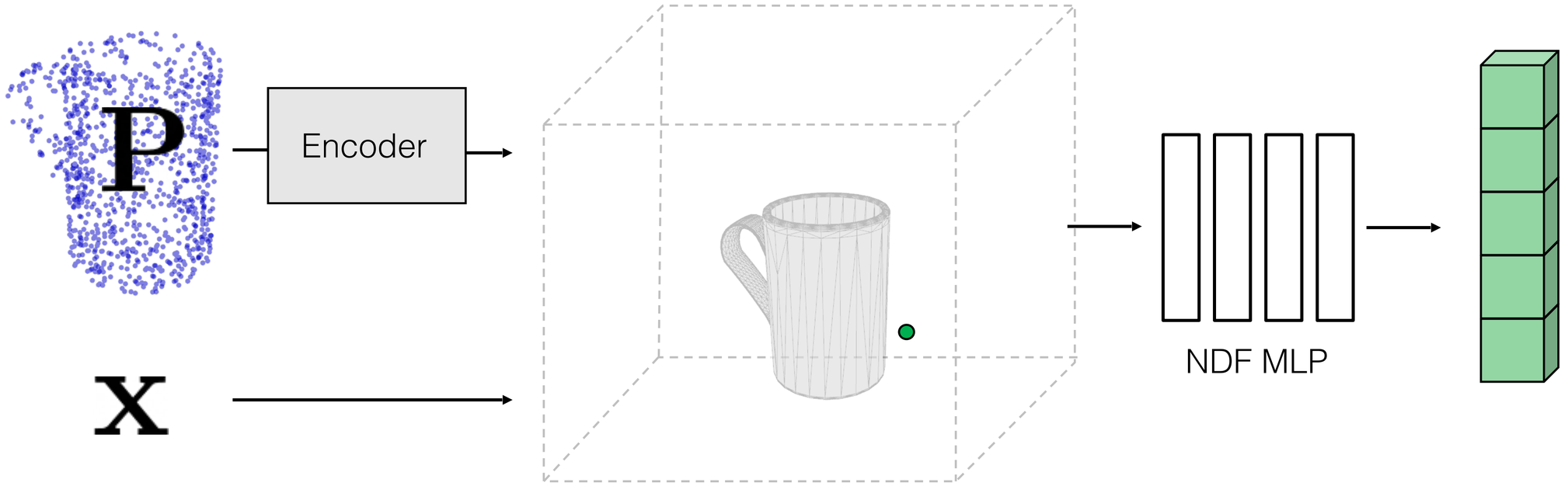
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



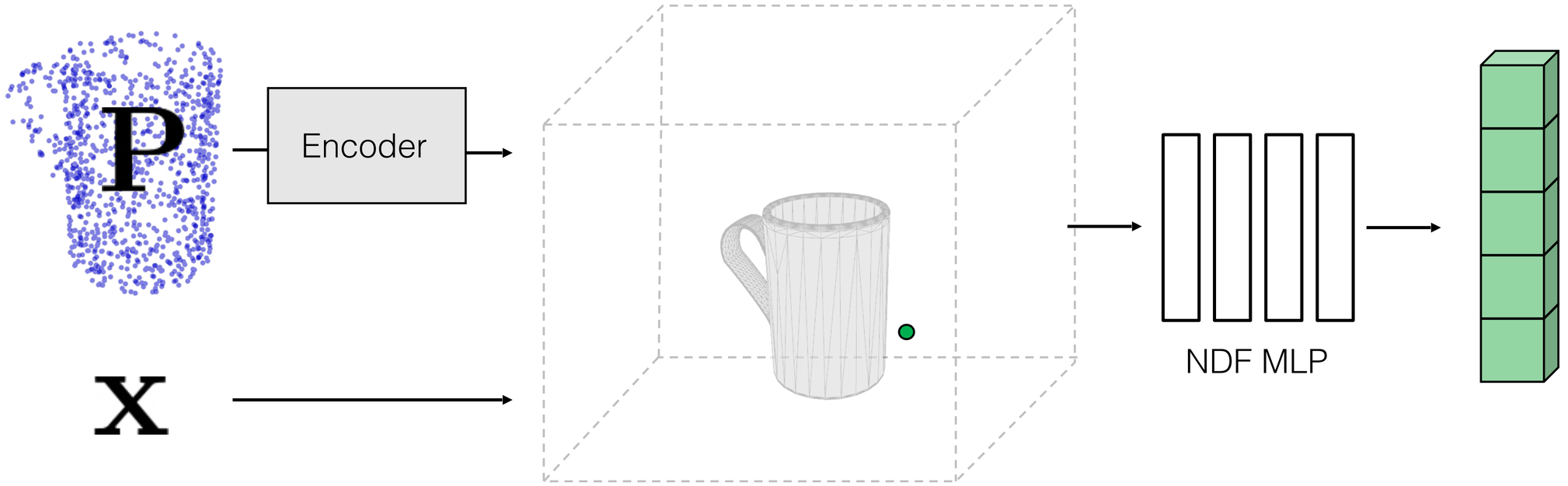
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



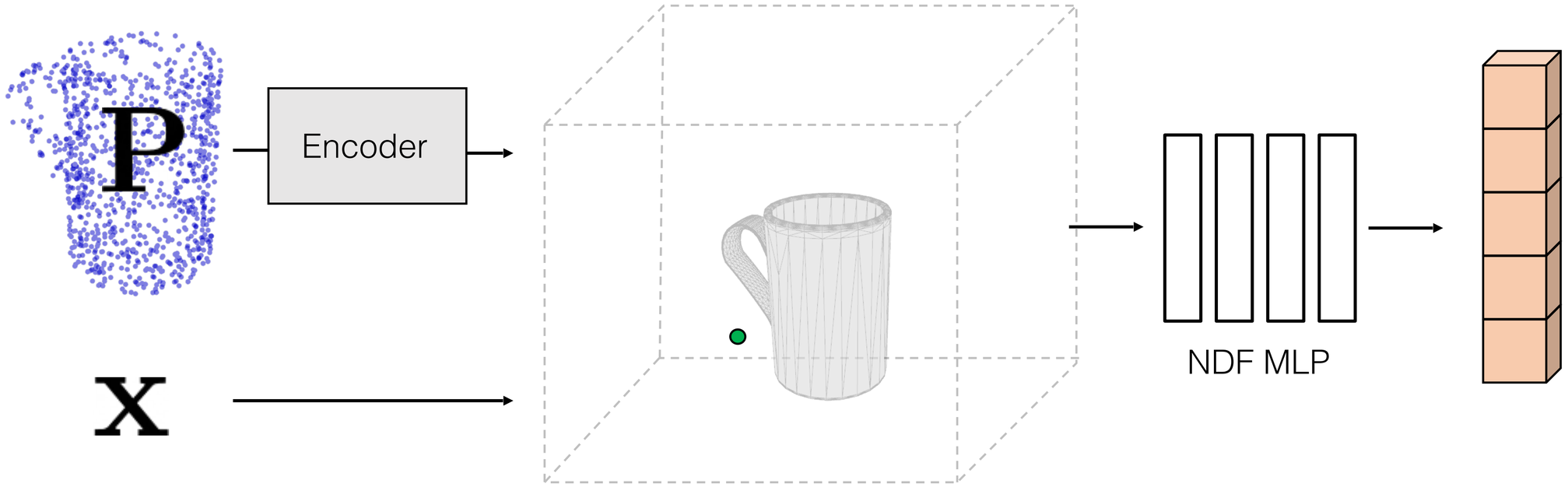
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



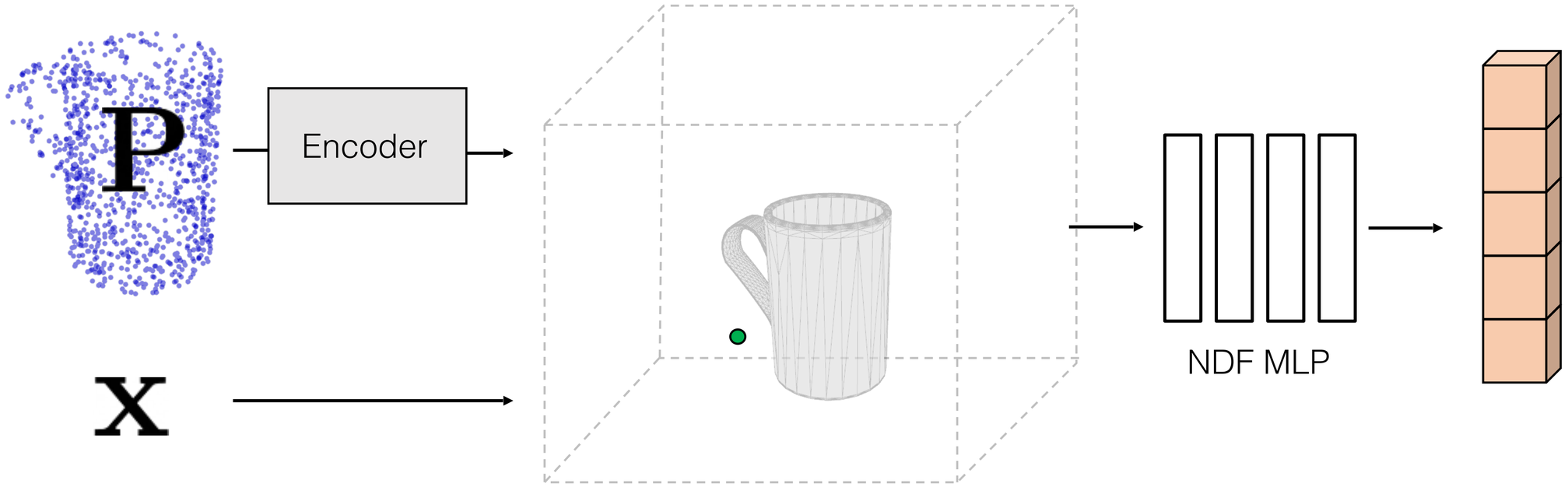
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



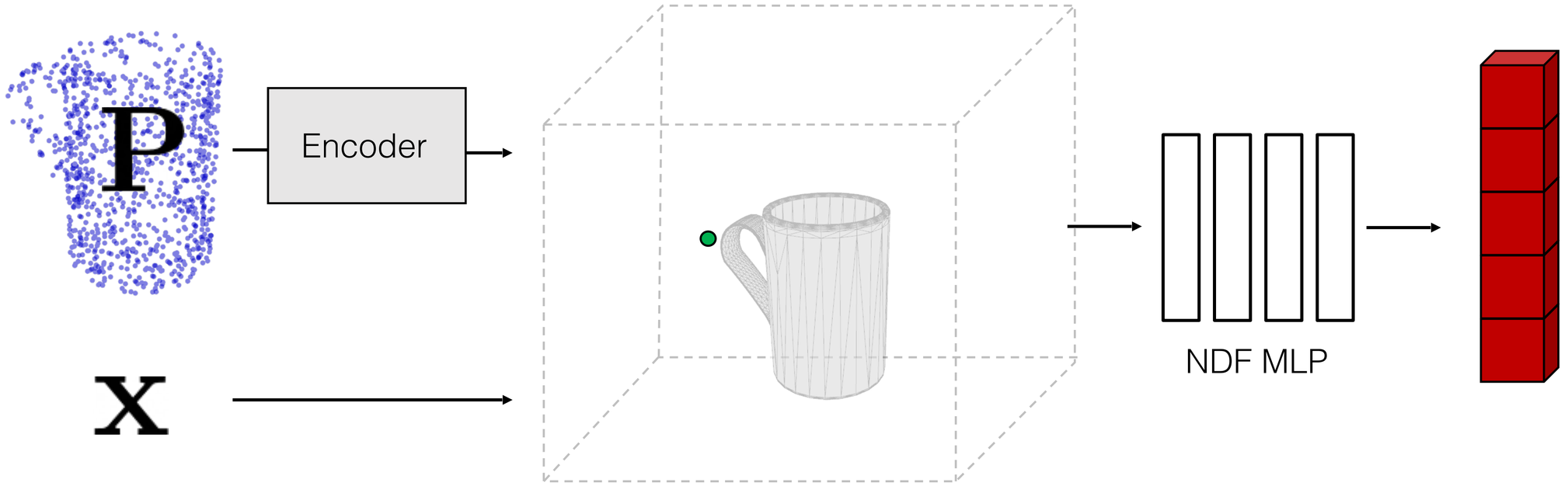
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields



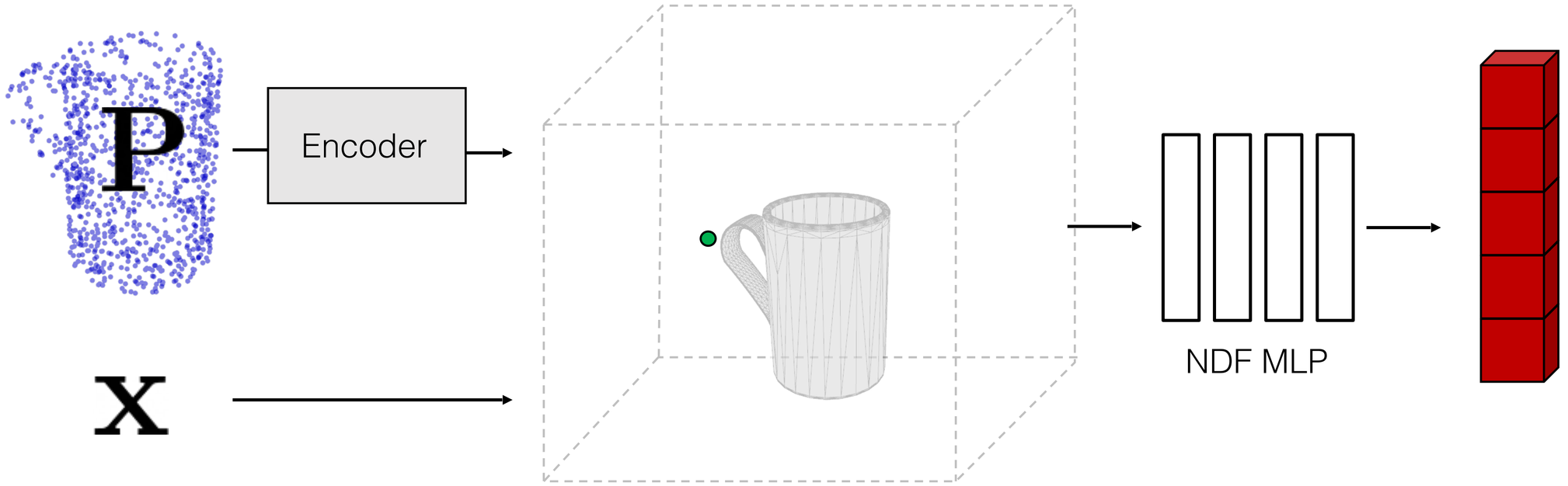
$$f(\mathbf{x}|\mathbf{P})$$

Neural Point Descriptor Fields

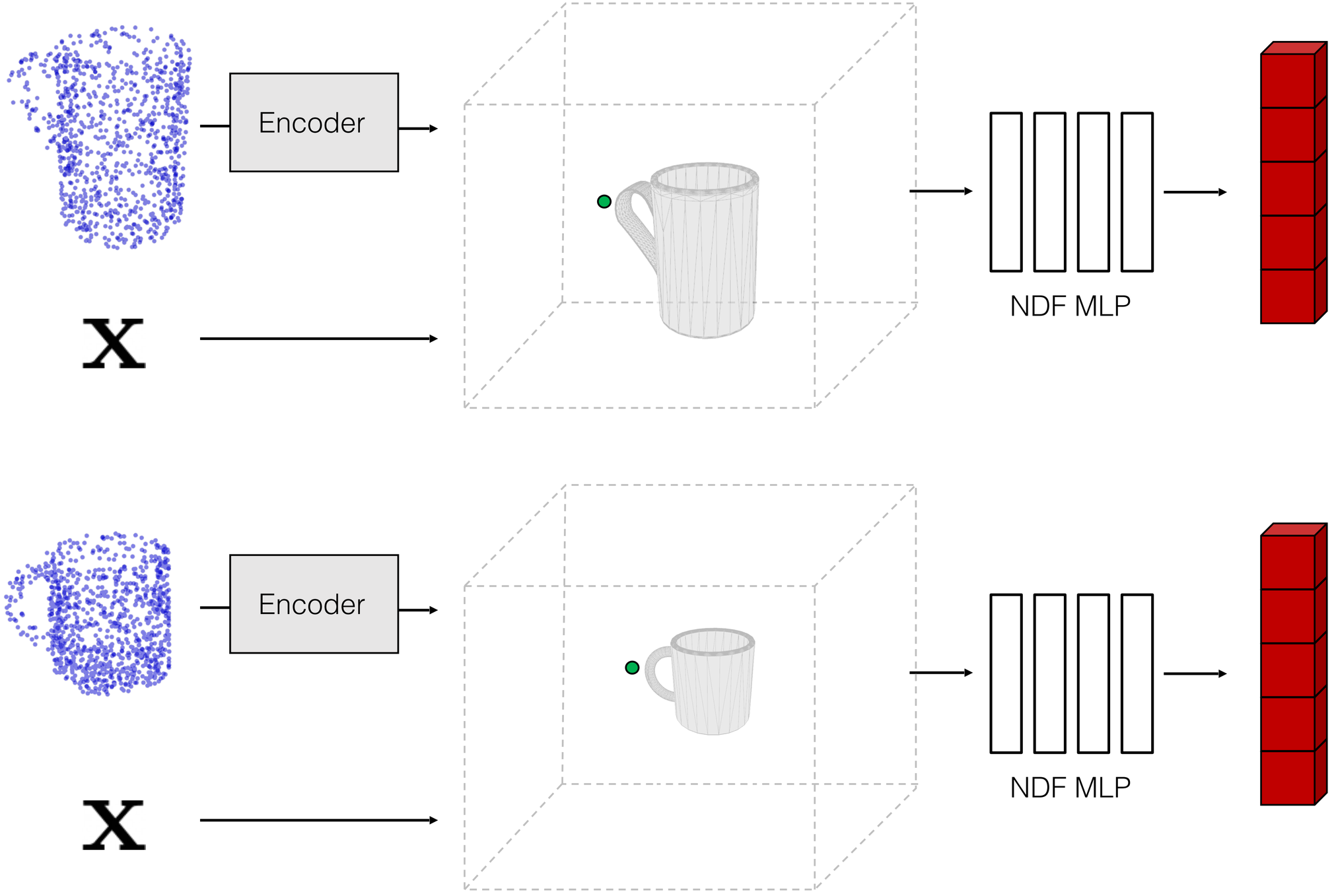


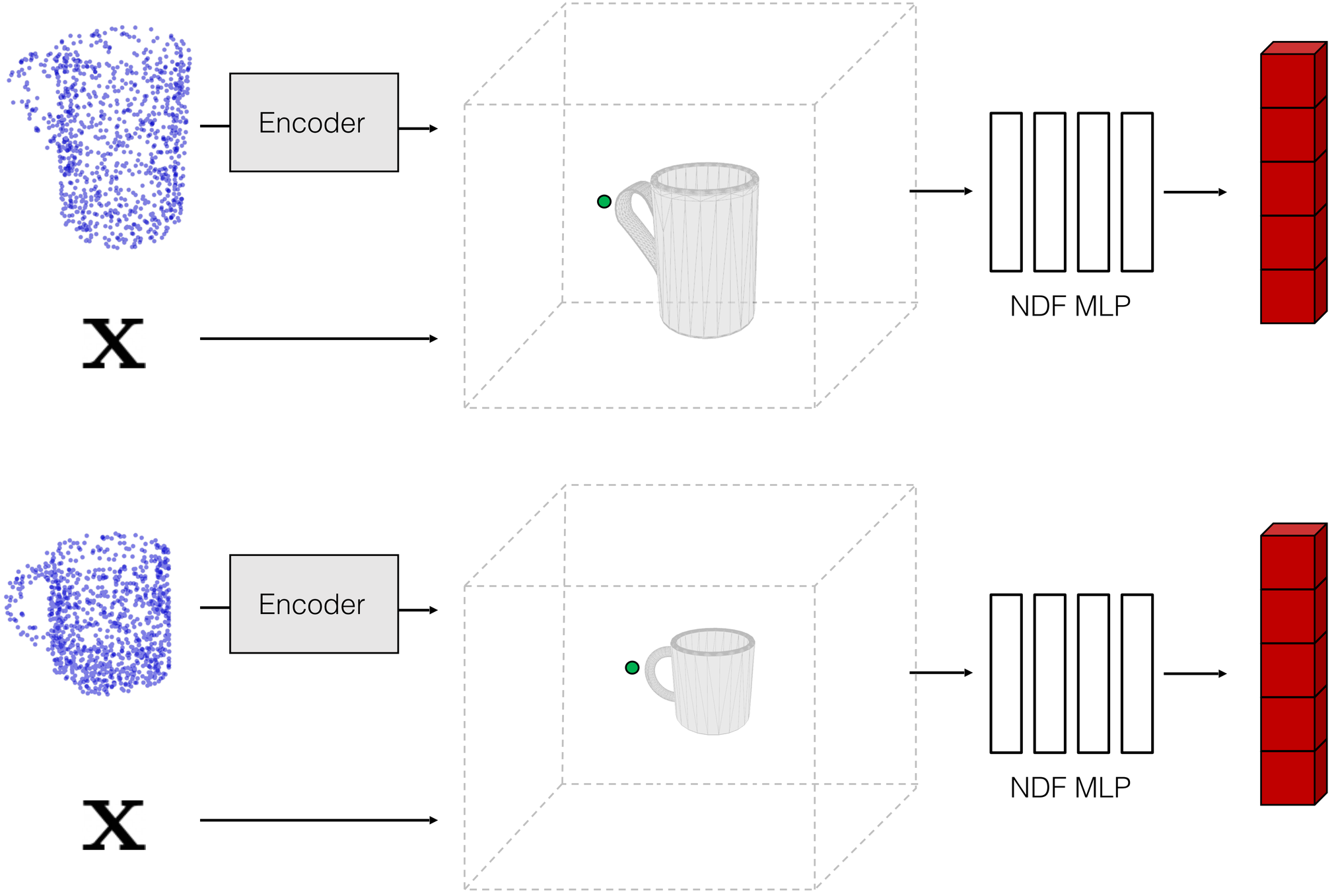
$$f(\mathbf{x}|\mathbf{P})$$

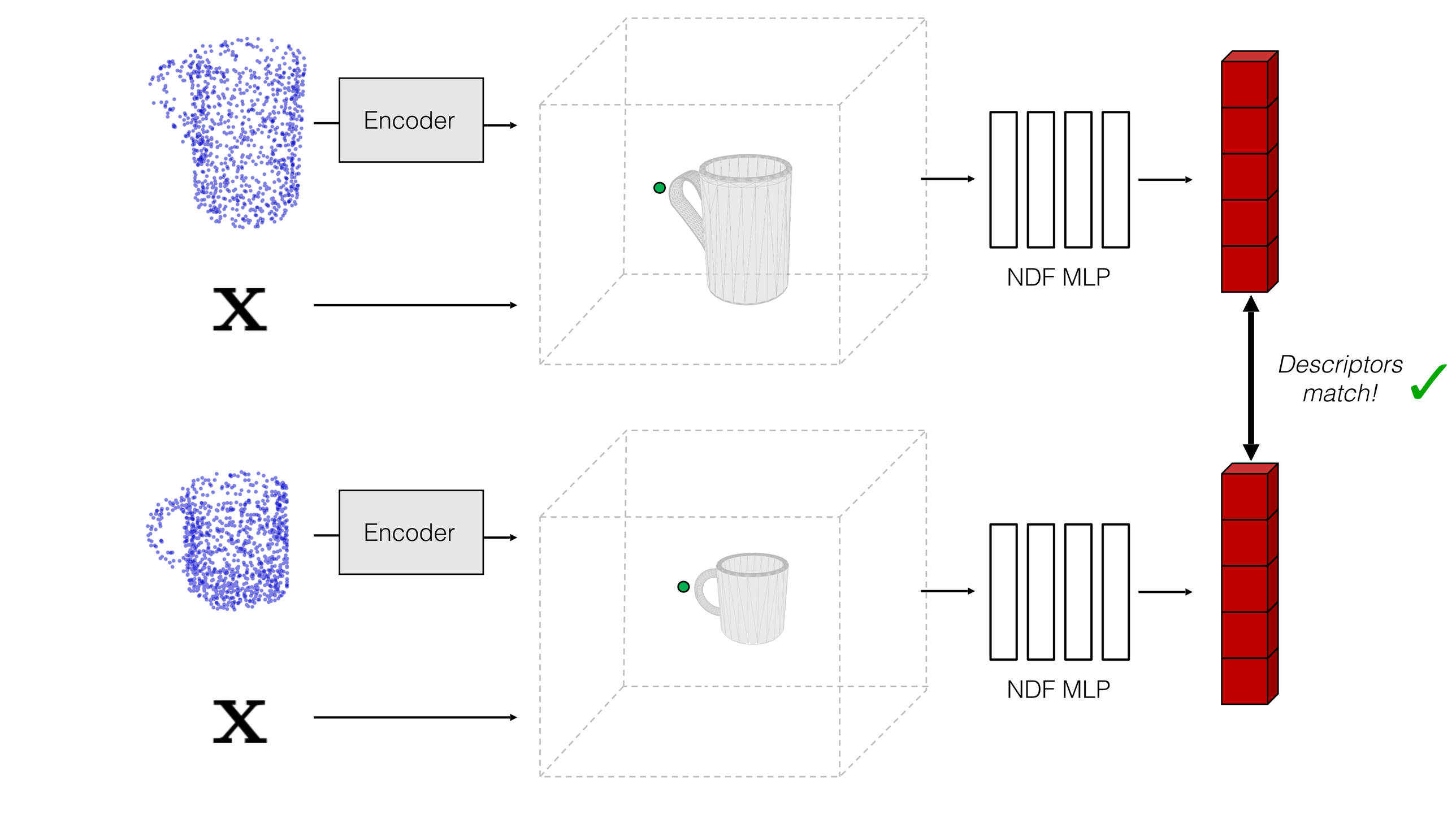
Neural Point Descriptor Fields



$$f(\mathbf{x}|\mathbf{P})$$


Encoder
X
NDF MLP
Encoder
X
NDF MLP


Encoder
X
NDF MLP
Encoder
X
NDF MLP


Encoder
X
NDF MLP
Descriptors match!
Encoder
X
NDF MLP

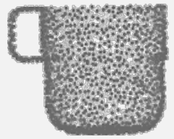
How to obtain descriptors that encode correspondence across shapes?

How to obtain descriptors that encode correspondence across shapes?

Training Neural Descriptor Fields

Training Neural Descriptor Fields

Training Neural Descriptor Fields



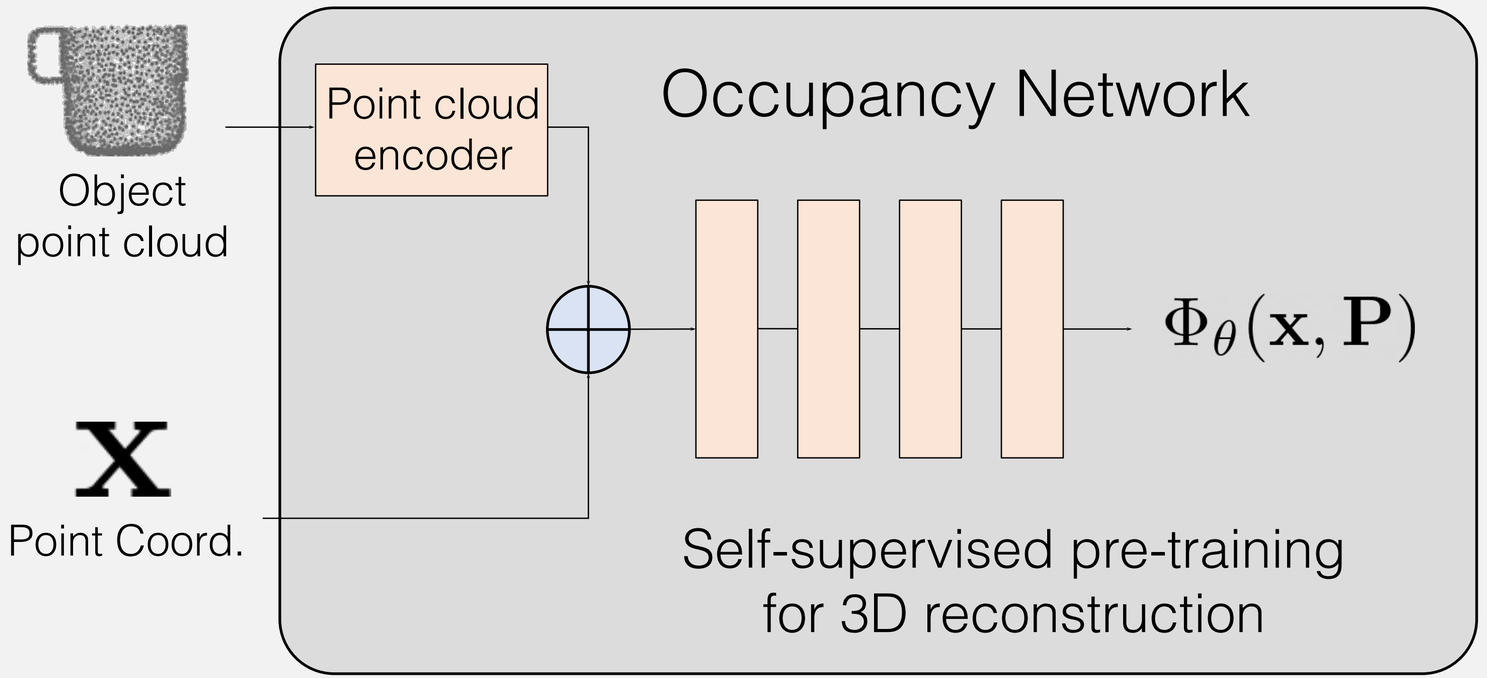
Object
point cloud

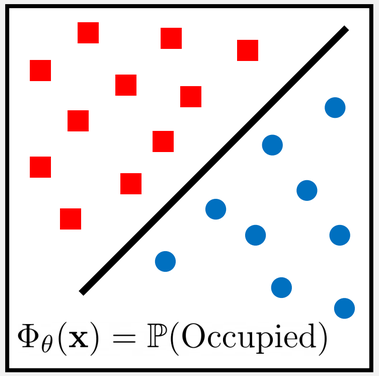
$\mathbf{x}$

Point Coord.

L. Mescheder, et. al, “Occupancy Networks: Learning 3D Reconstruction in Function Space”, 2019

Training Neural Descriptor Fields

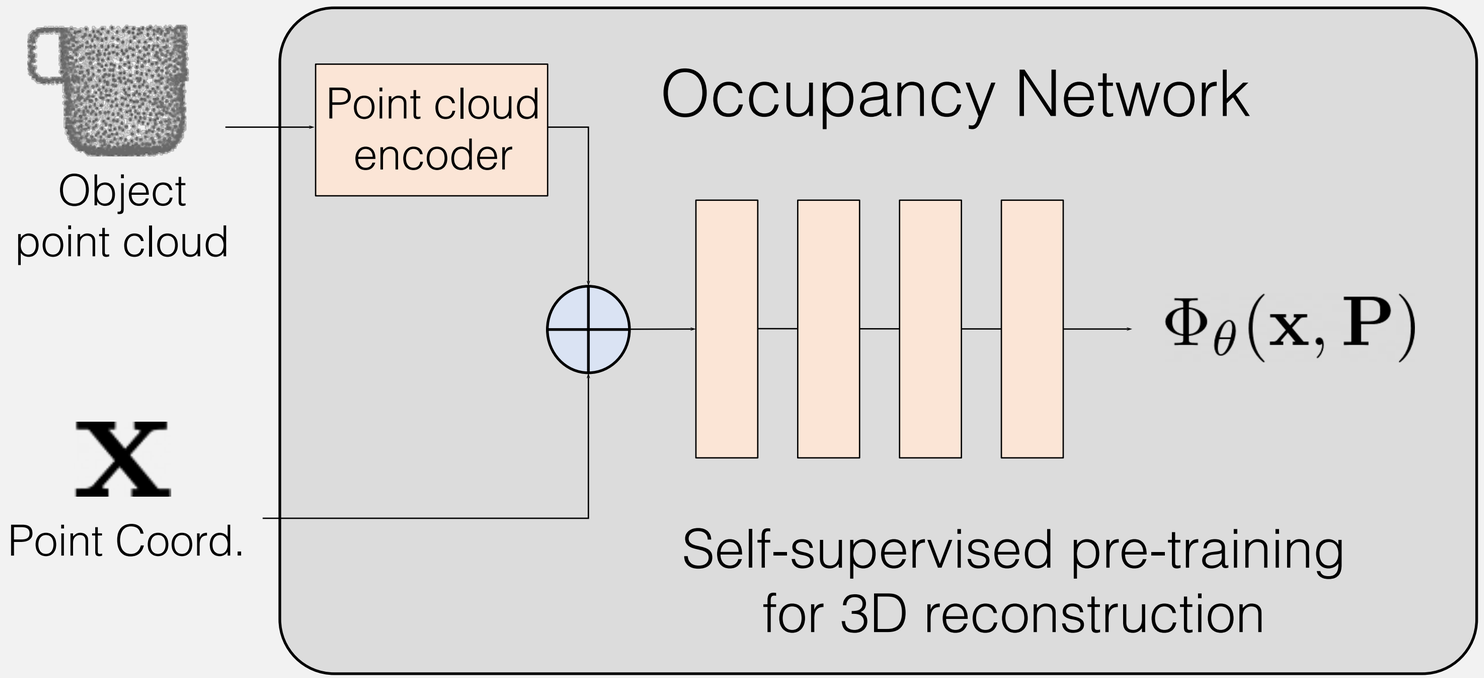




(3D shape data generated fully offline and in simulation)

L. Mescheder, et. al, “Occupancy Networks: Learning 3D Reconstruction in Function Space”, 2019

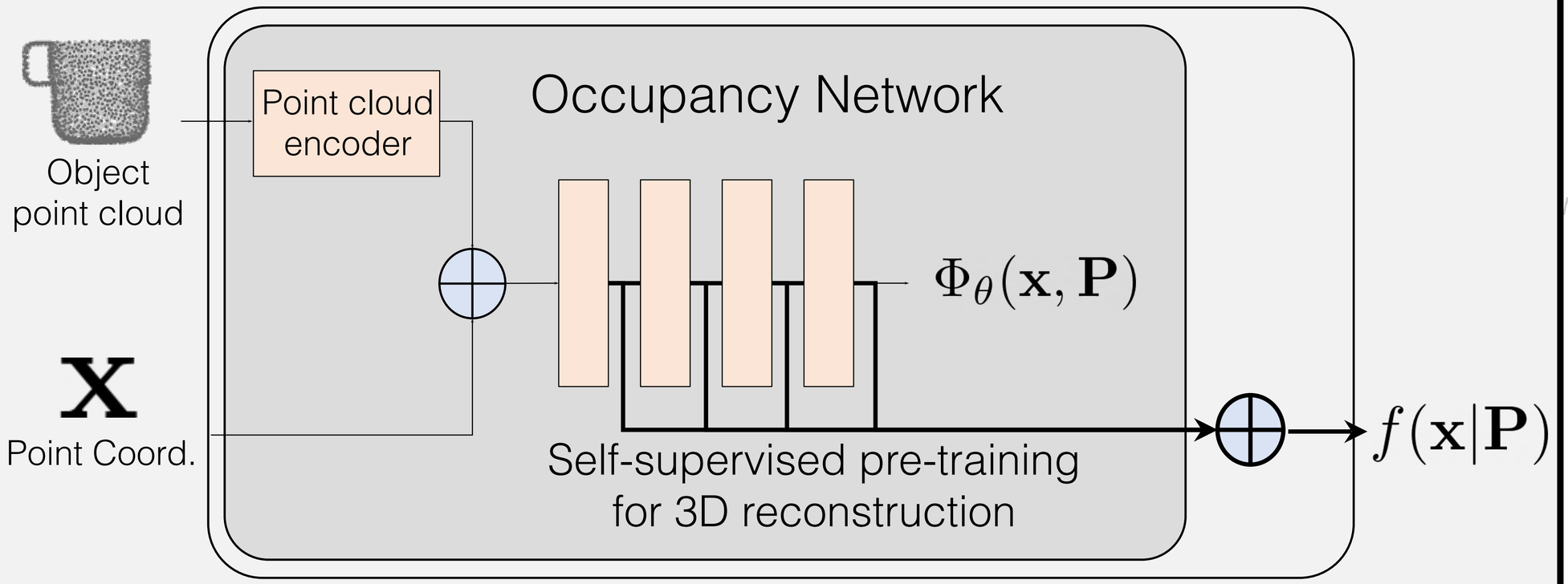
Training Neural Descriptor Fields



(3D shape data generated fully offline and in simulation)

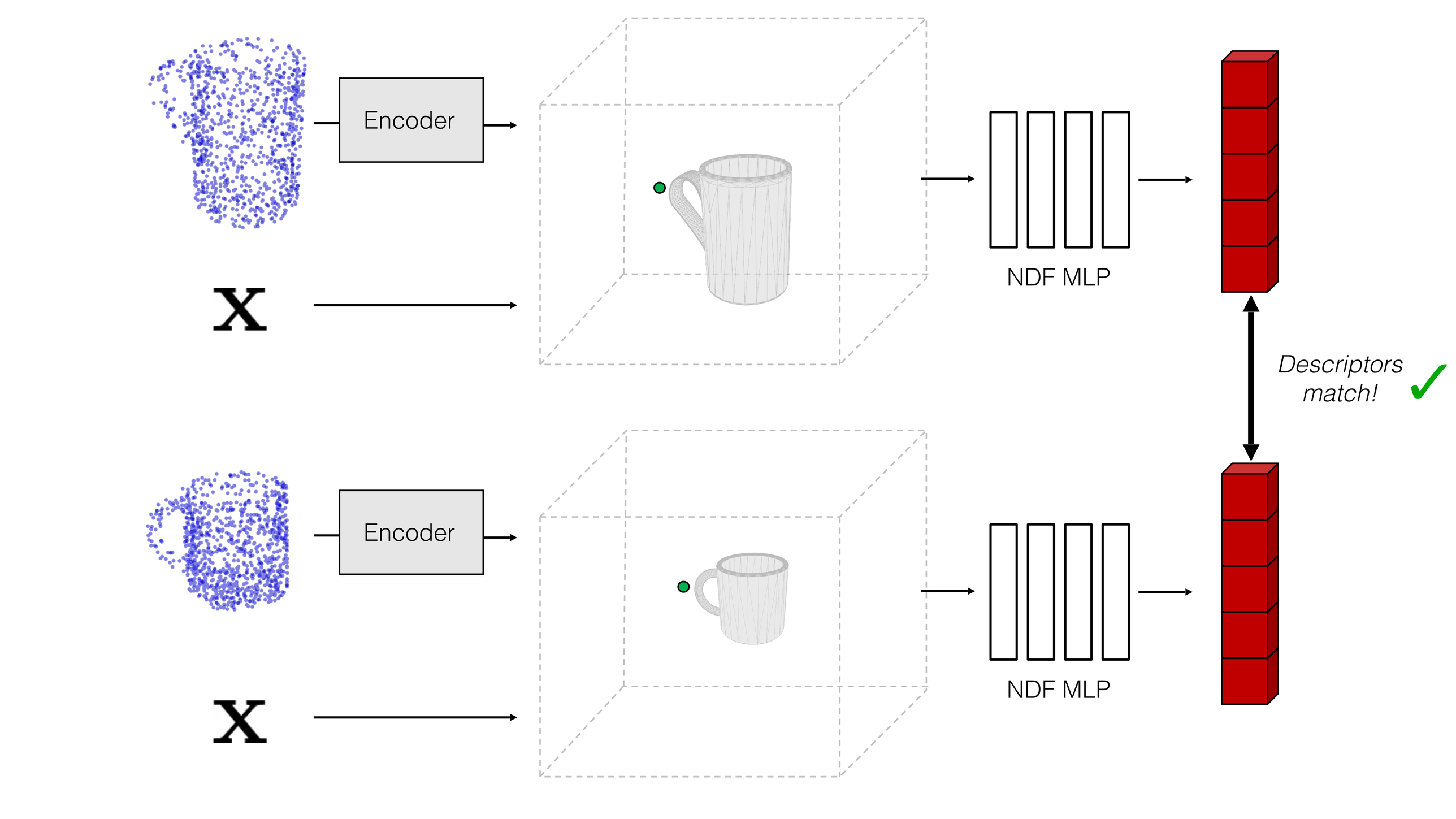
L. Mescheder, et. al, “Occupancy Networks: Learning 3D Reconstruction in Function Space”, 2019

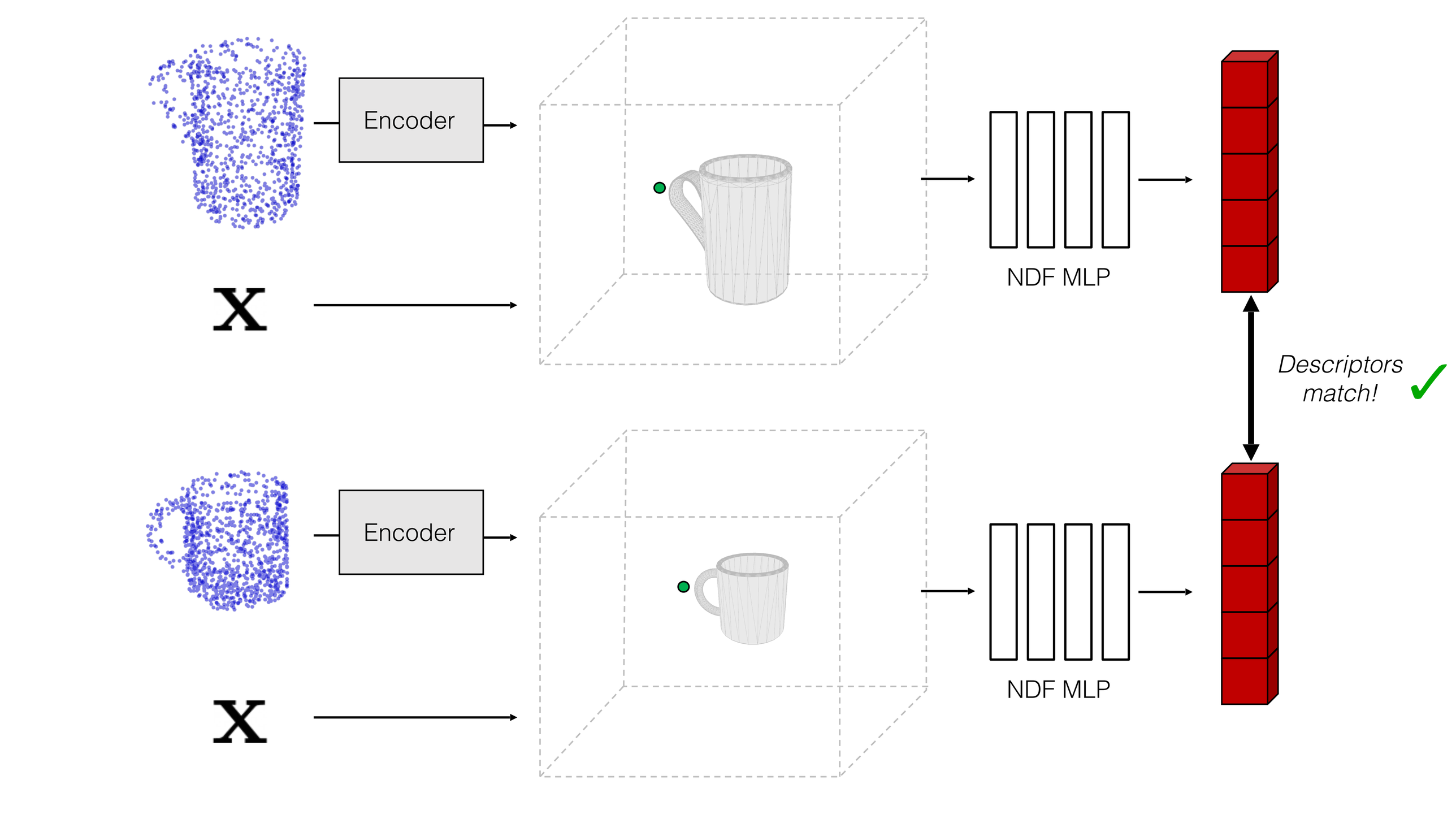
Training Neural Descriptor Fields

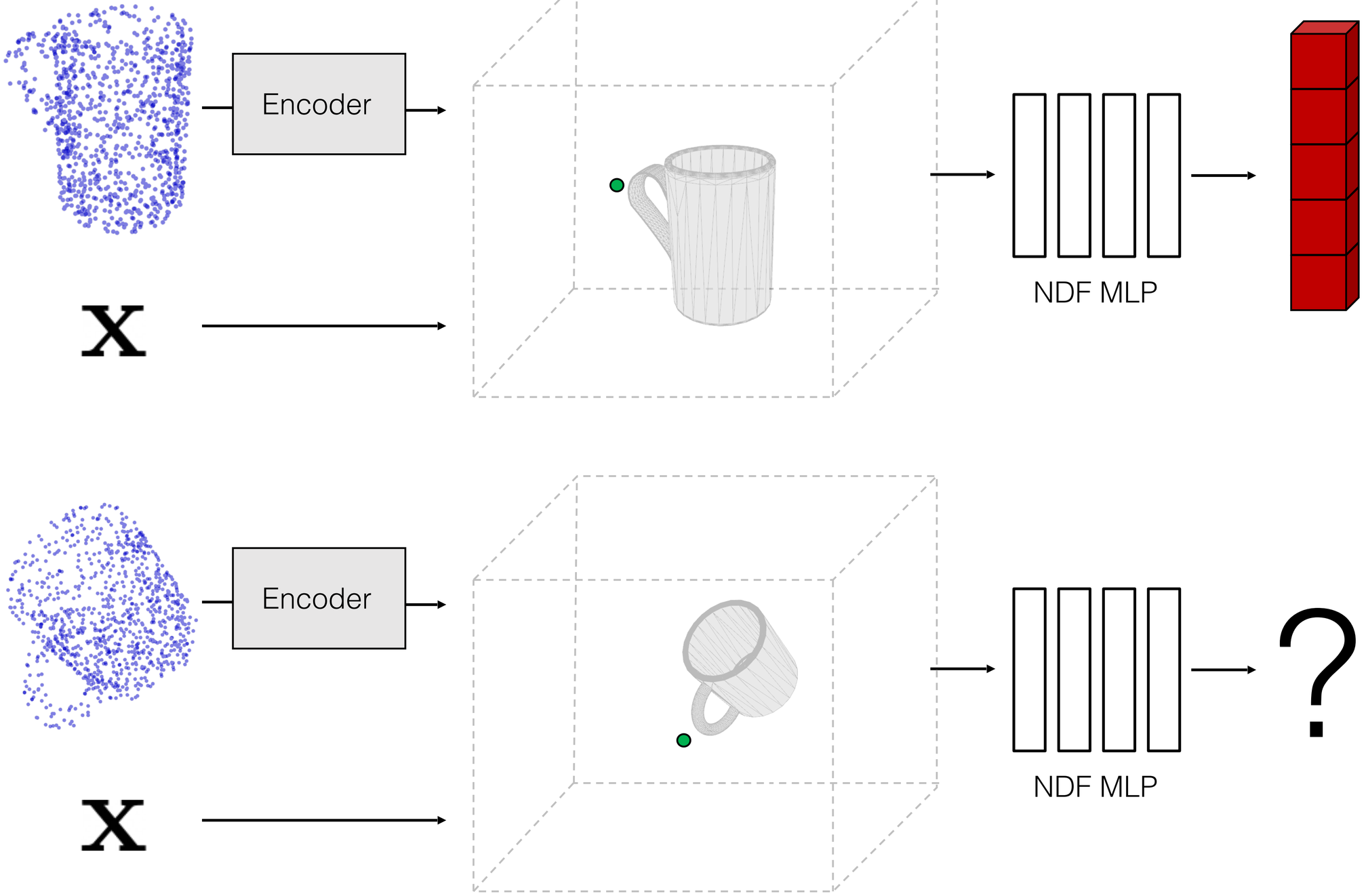


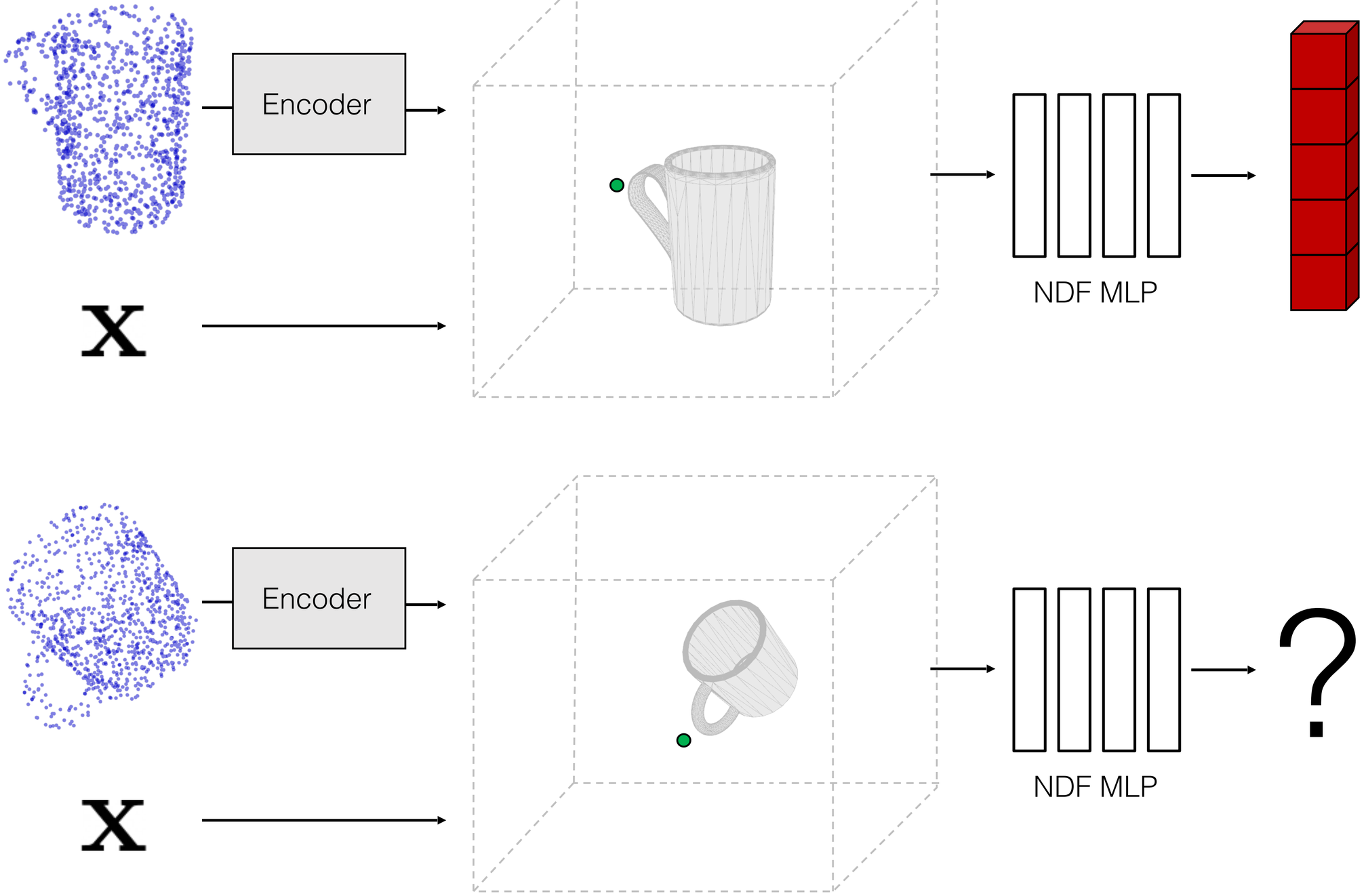
(3D shape data generated fully offline and in simulation)

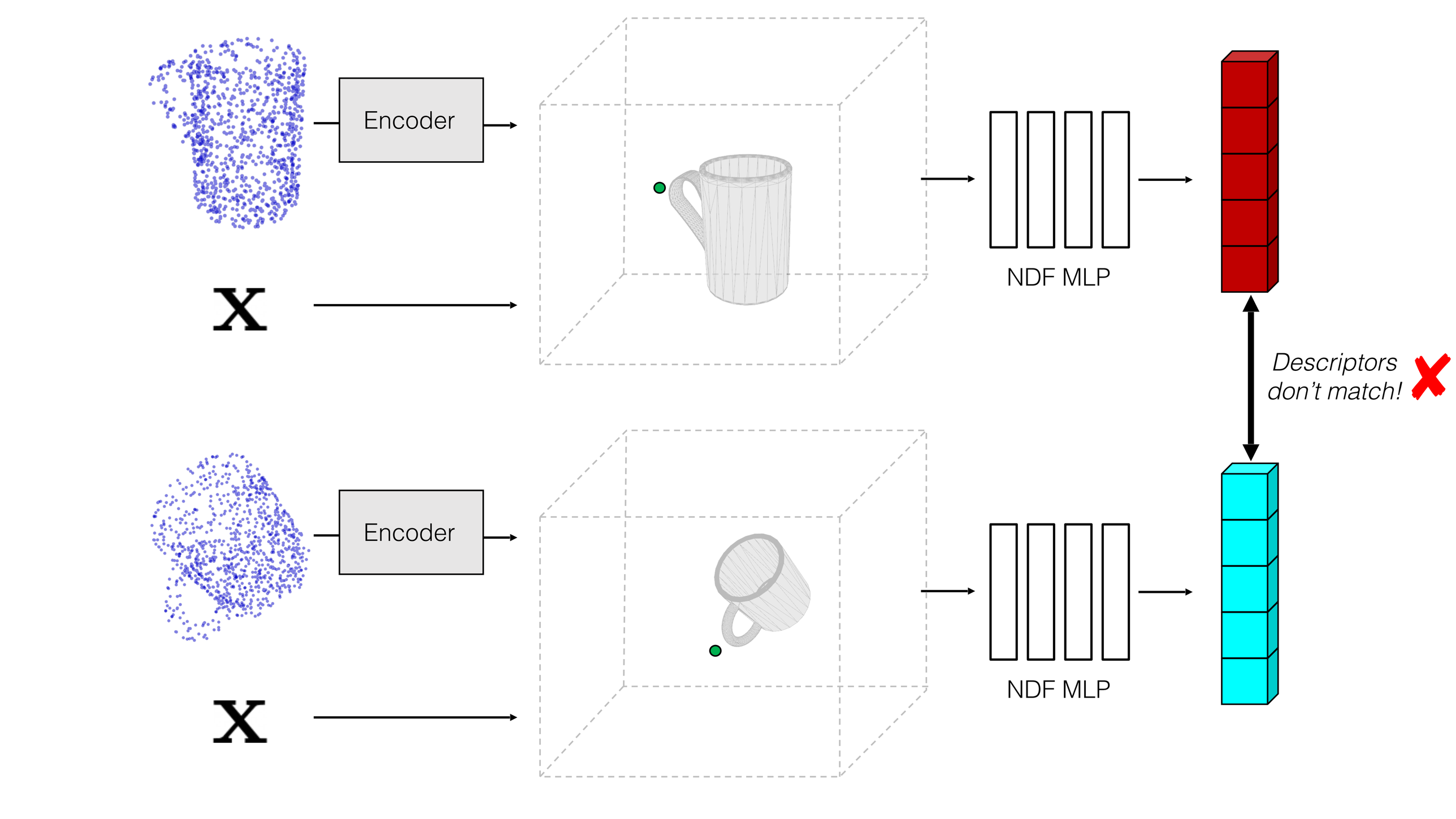
L. Mescheder, et. al, “Occupancy Networks: Learning 3D Reconstruction in Function Space”, 2019

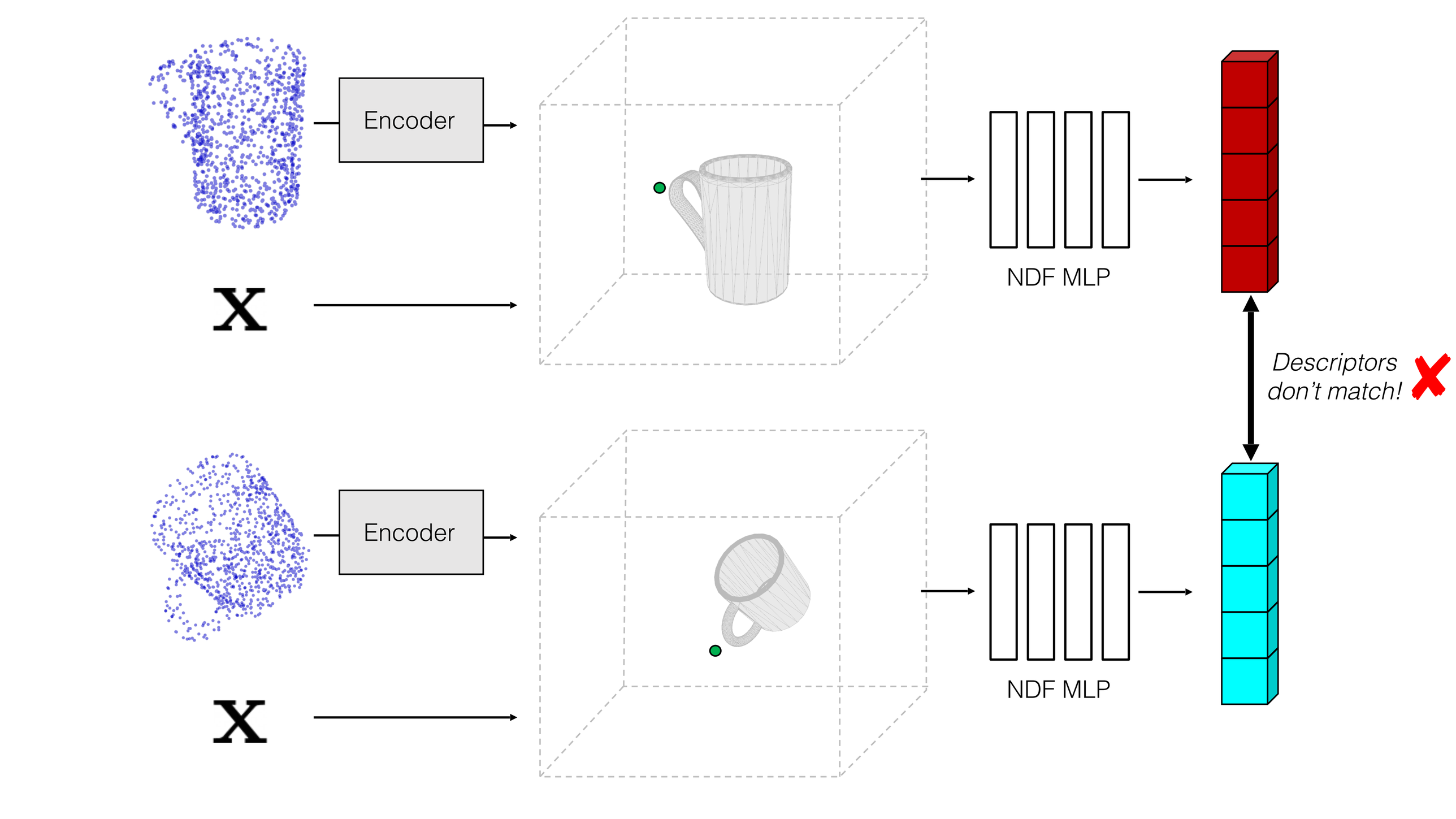

Encoder
X
NDF MLP
Descriptors match!
Encoder
X
NDF MLP


Encoder
X
NDF MLP
Descriptors match!
Encoder
X
NDF MLP


Encoder
x
NDF MLP
Encoder
x
NDF MLP
?

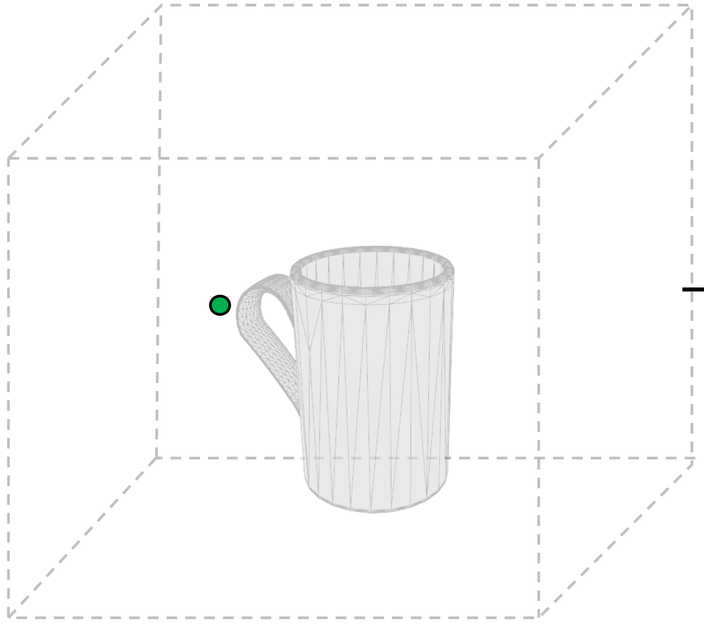

Encoder
x
NDF MLP
Encoder
x
NDF MLP
?


Encoder
X
NDF MLP
Descriptors
don't match!
Encoder
X
NDF MLP

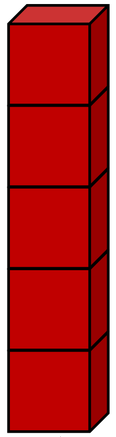

Encoder
X
NDF MLP
Descriptors
don't match!
Encoder
X
NDF MLP



Encoder



NDF MLP

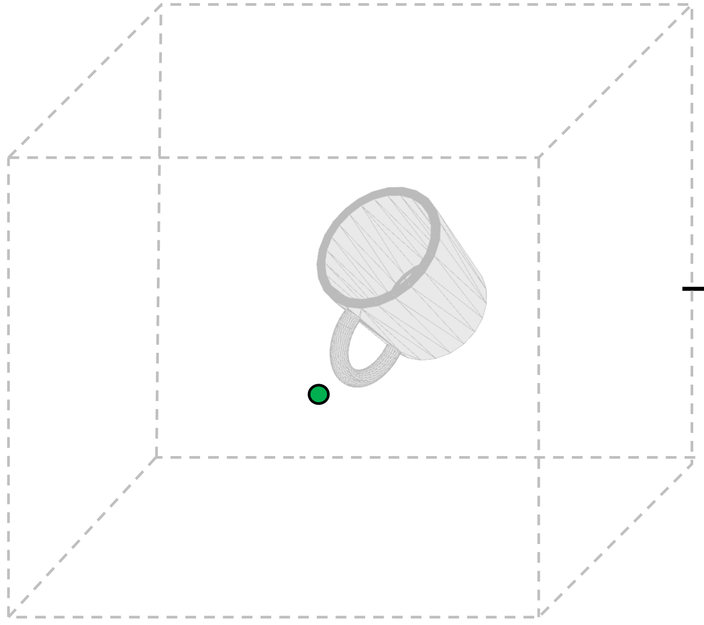


Descriptors match! ✓

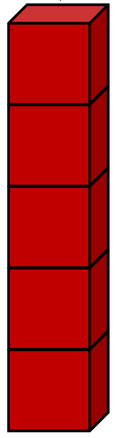
Ideally, descriptor field transforms with the shape



Encoder

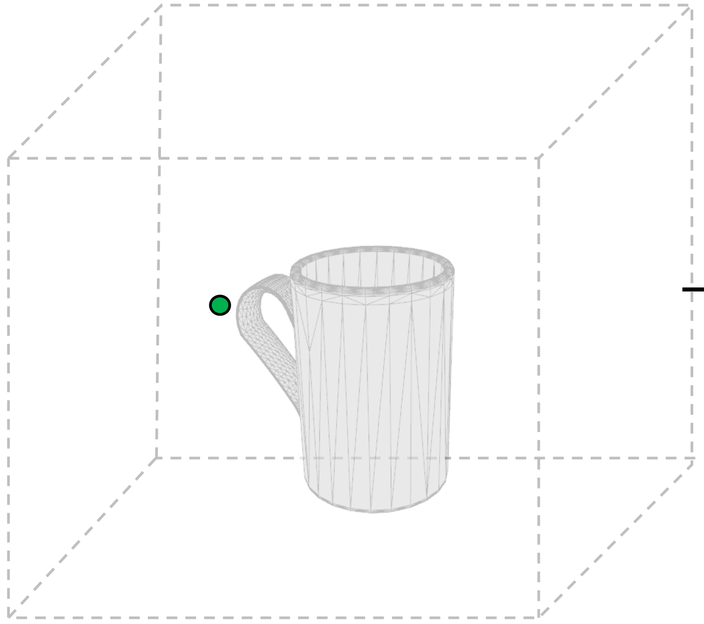


NDF MLP

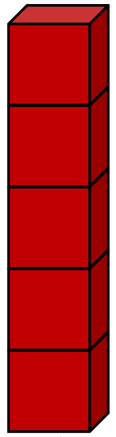




Encoder



NDF MLP

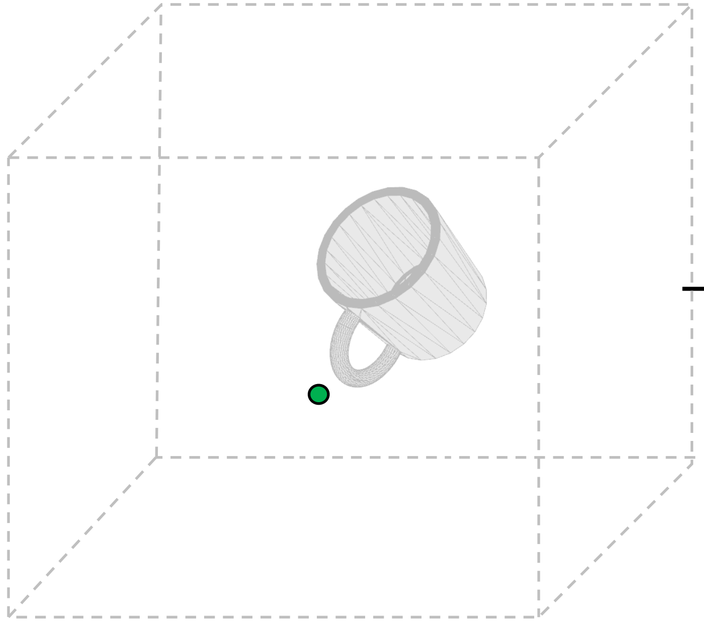


Descriptors match! ✓

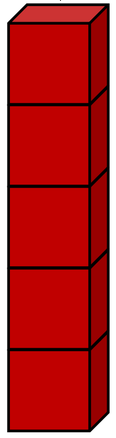
Ideally, descriptor field transforms with the shape

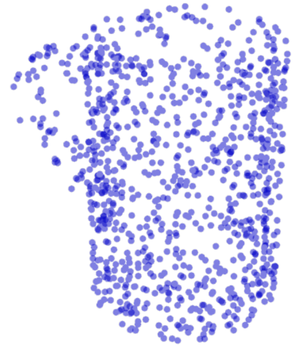


Encoder



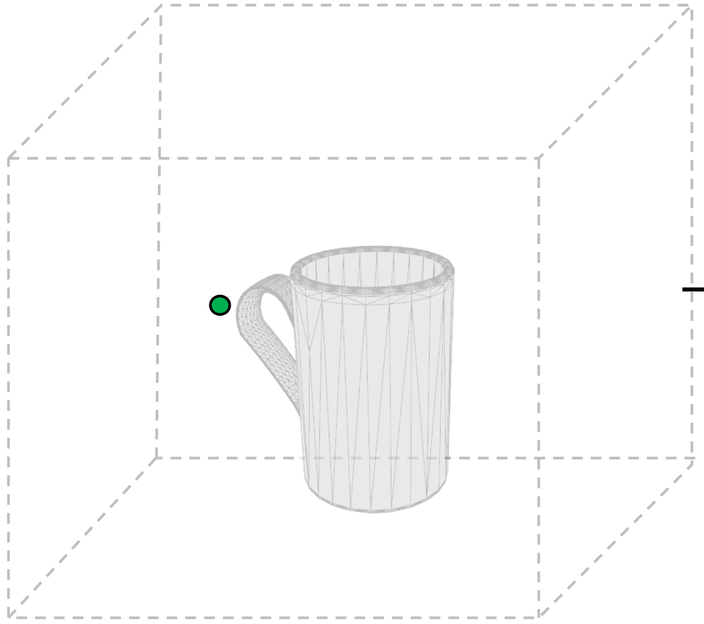
NDF MLP



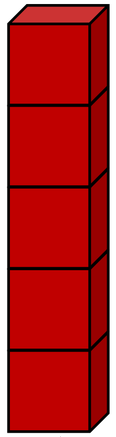


Encoder

$\mathbf{X}$



NDF MLP



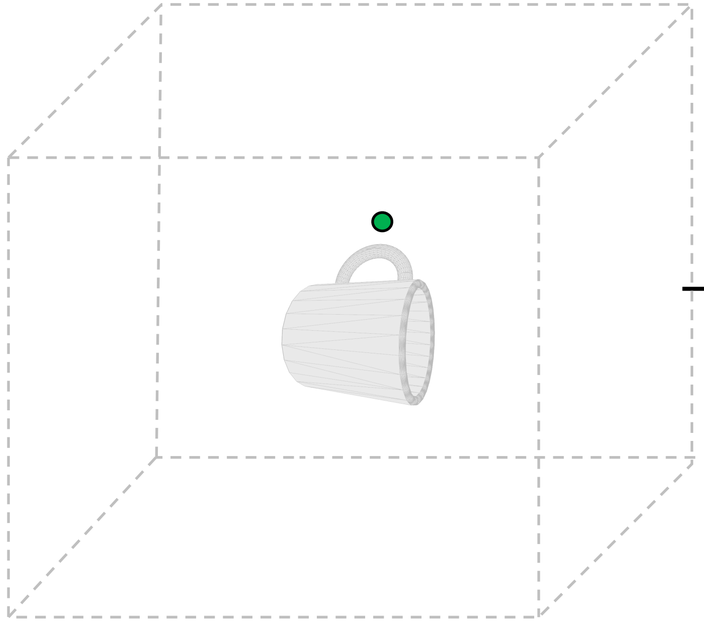
$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{Tx}|\mathbf{TP}), \mathbf{T} \in \mathbf{SE}(3)$$

Descriptors match! ✓

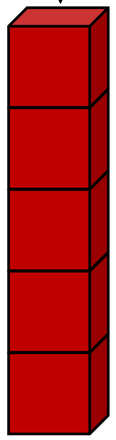


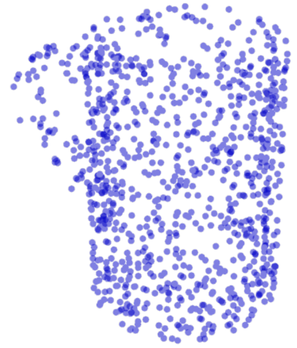
Encoder

$\mathbf{X}$



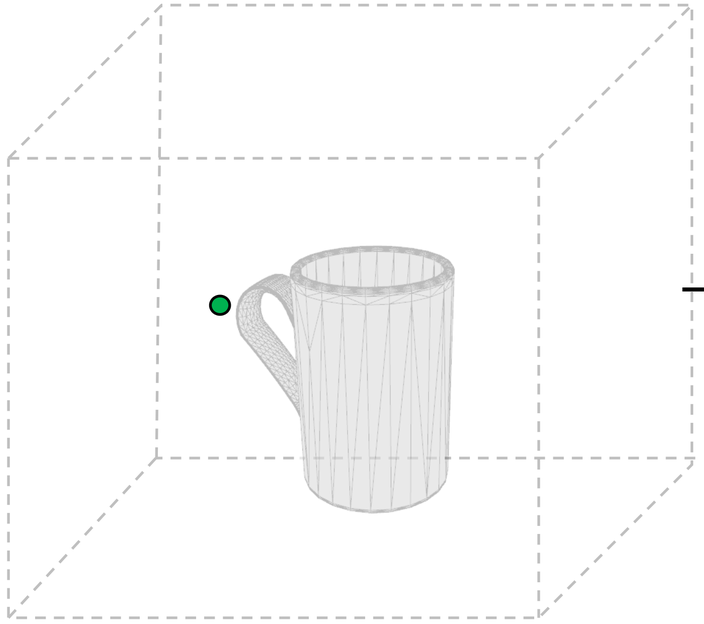
NDF MLP



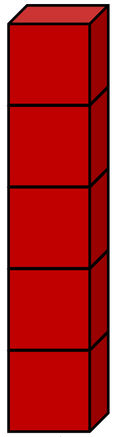


Encoder

$\mathbf{X}$



NDF MLP



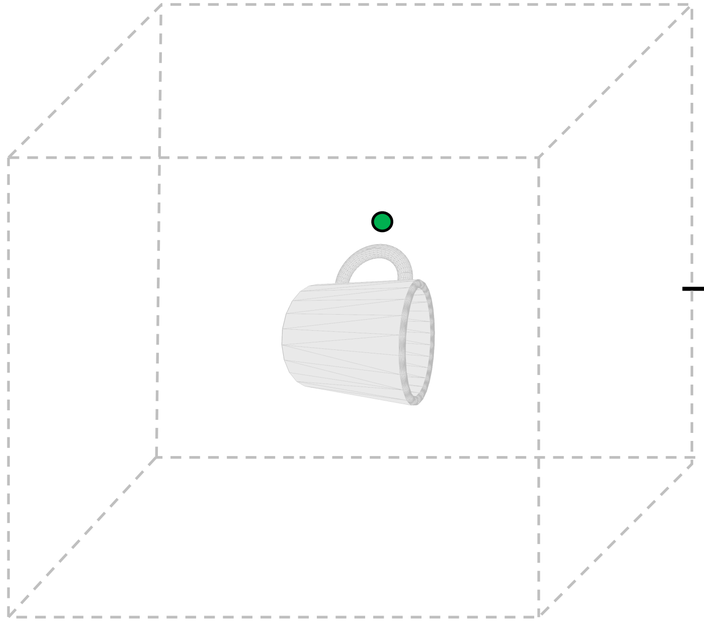
$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{Tx}|\mathbf{TP}), \mathbf{T} \in \mathbf{SE}(3)$$

Descriptors match! ✓

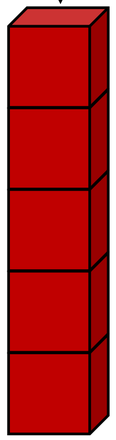


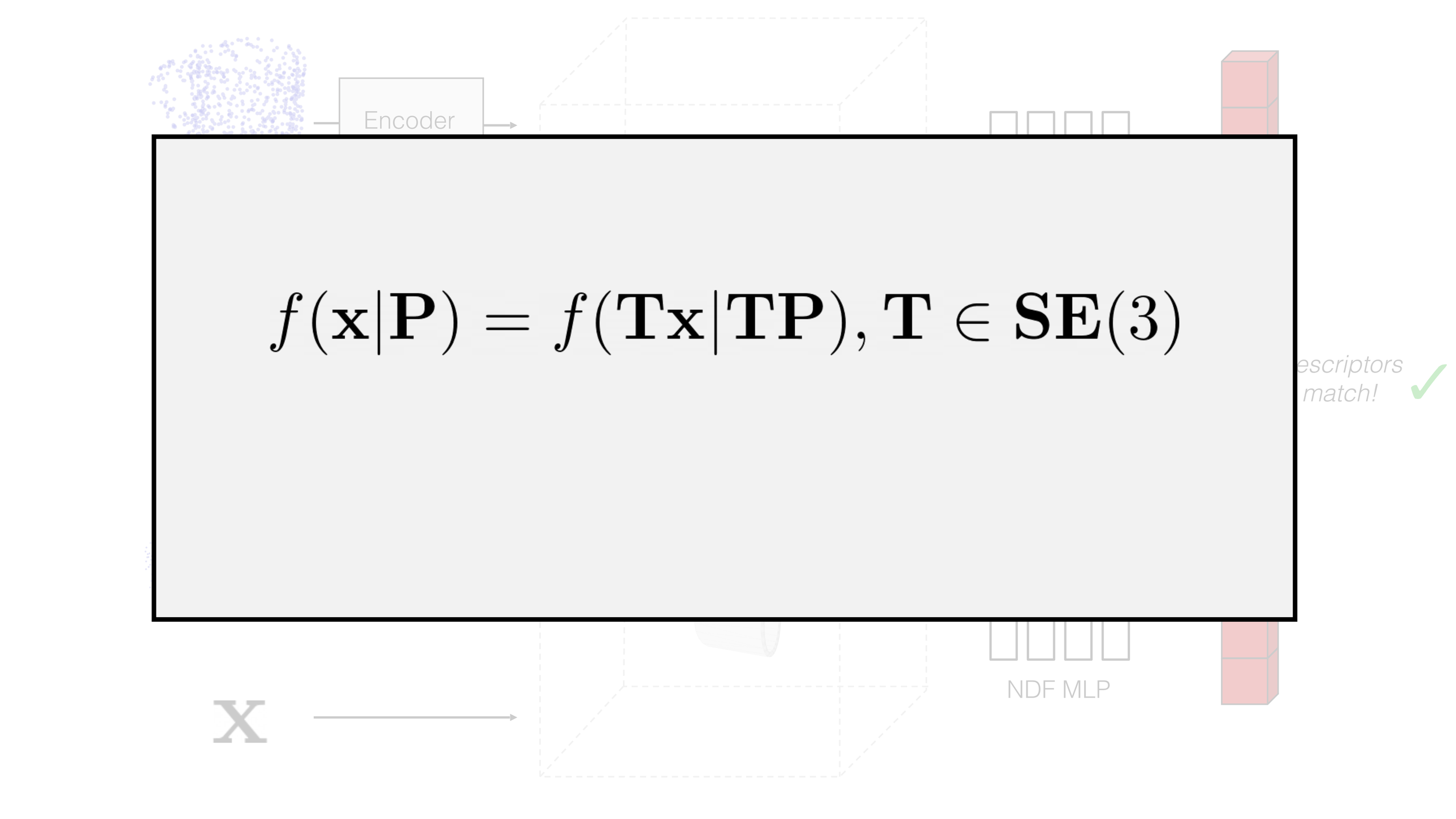
Encoder

$\mathbf{X}$

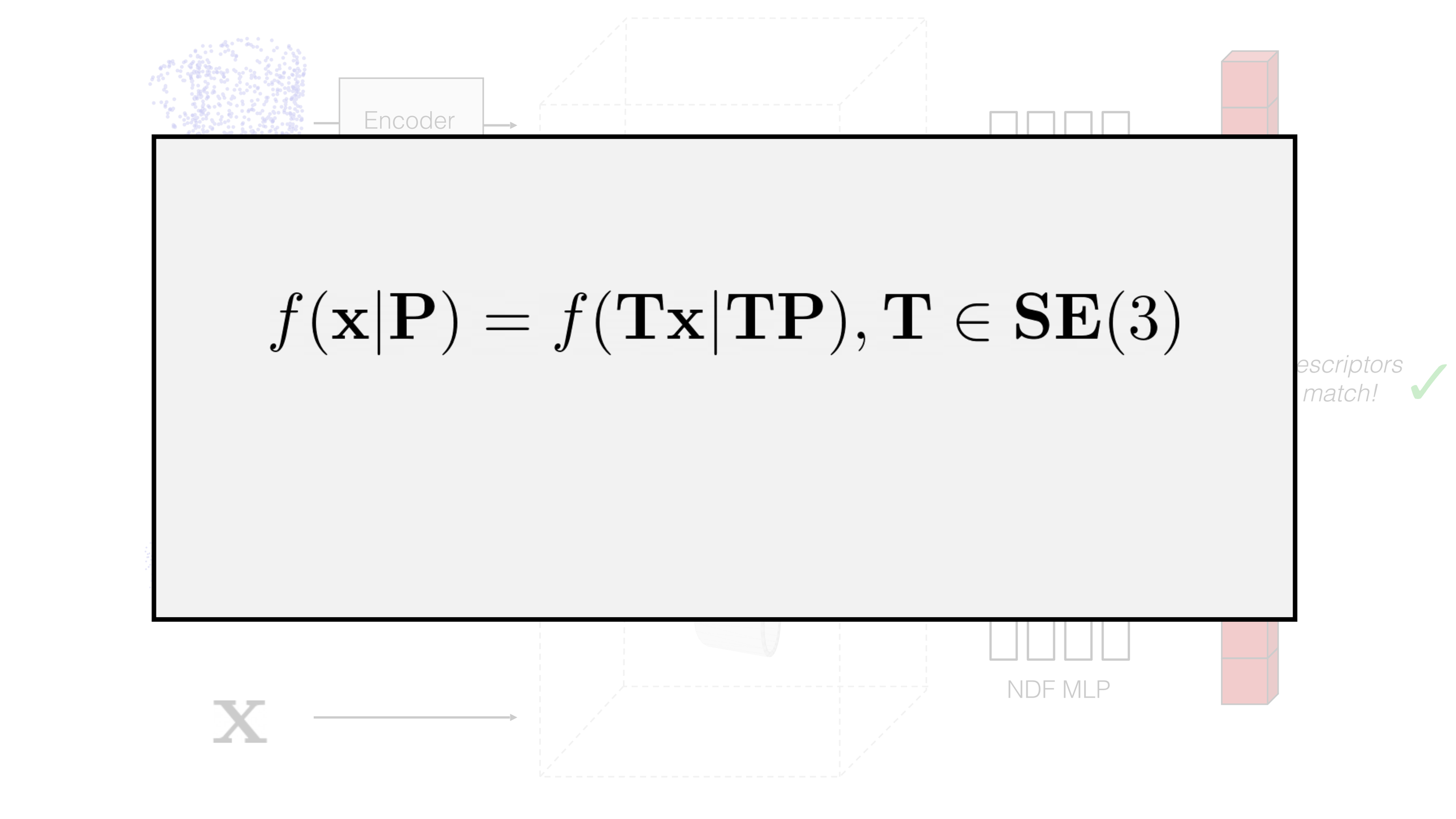


NDF MLP

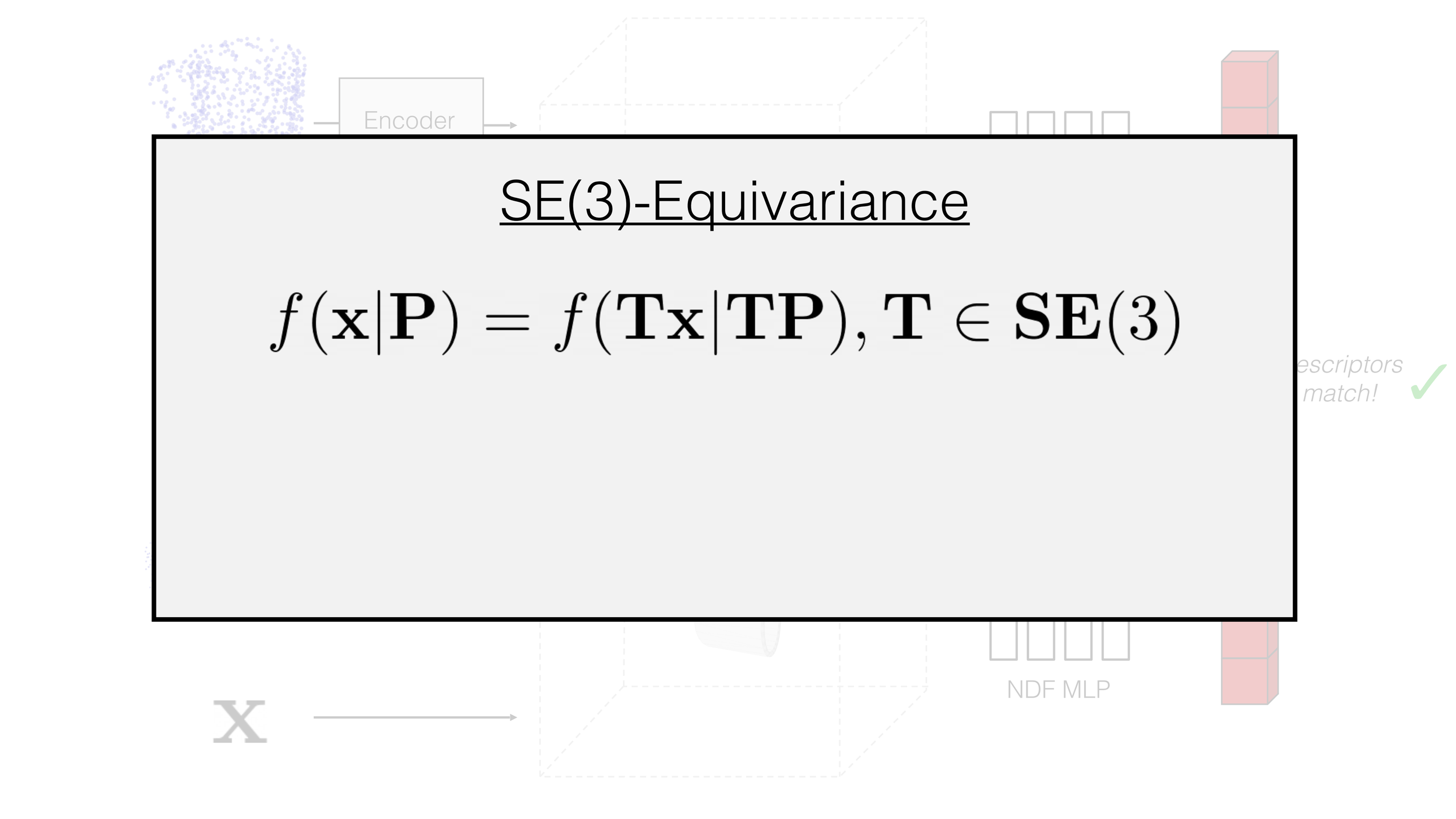




$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}), \mathbf{T} \in \mathbf{SE}(3)$$

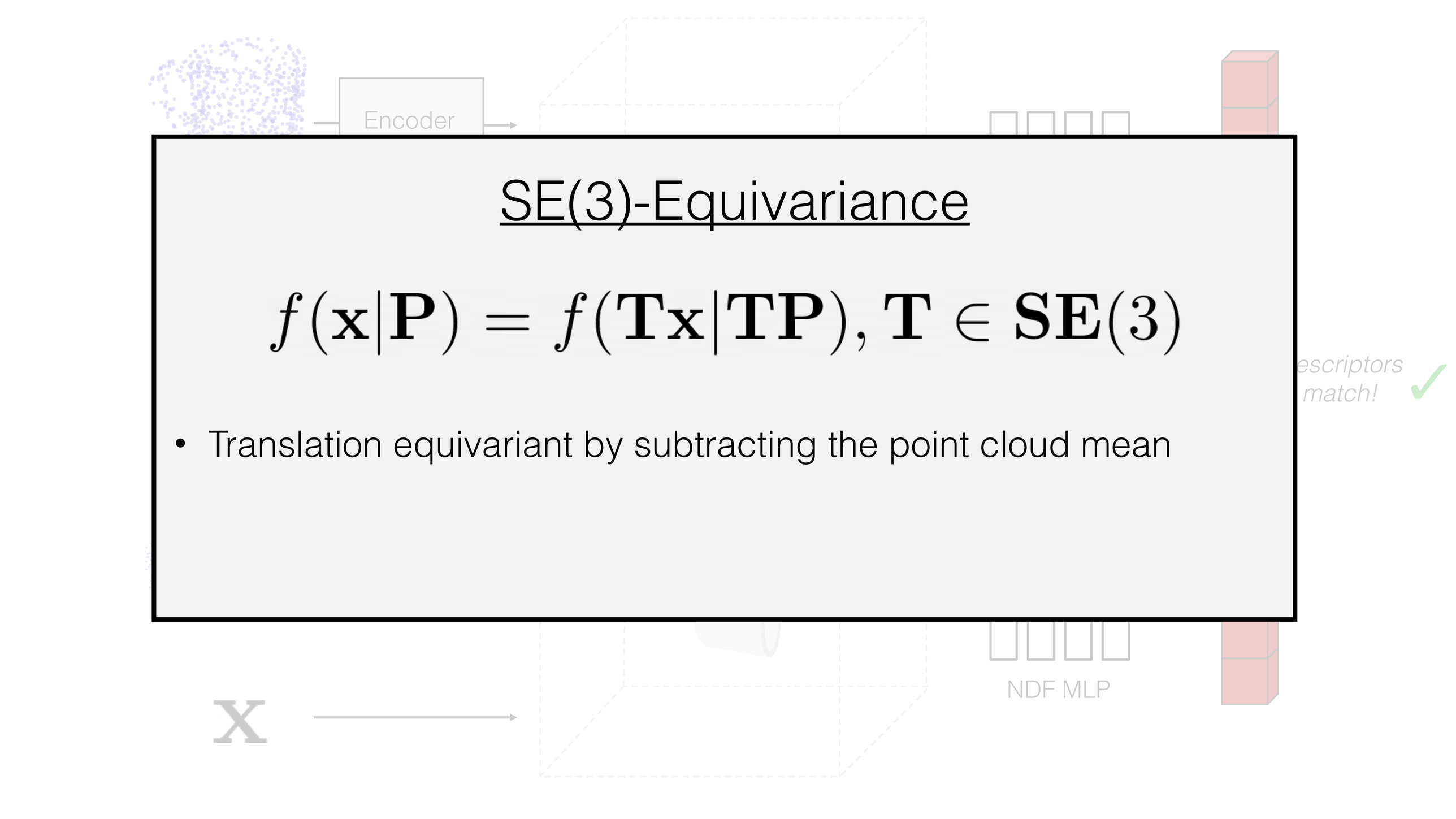


$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}), \mathbf{T} \in \mathbf{SE}(3)$$



SE(3)-Equivariance

$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}), \mathbf{T} \in \mathbf{SE}(3)$$



SE(3)-Equivariance

$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}), \mathbf{T} \in \mathbf{SE}(3)$$

- Translation equivariant by subtracting the point cloud mean

Encoder

SE(3)-Equivariance

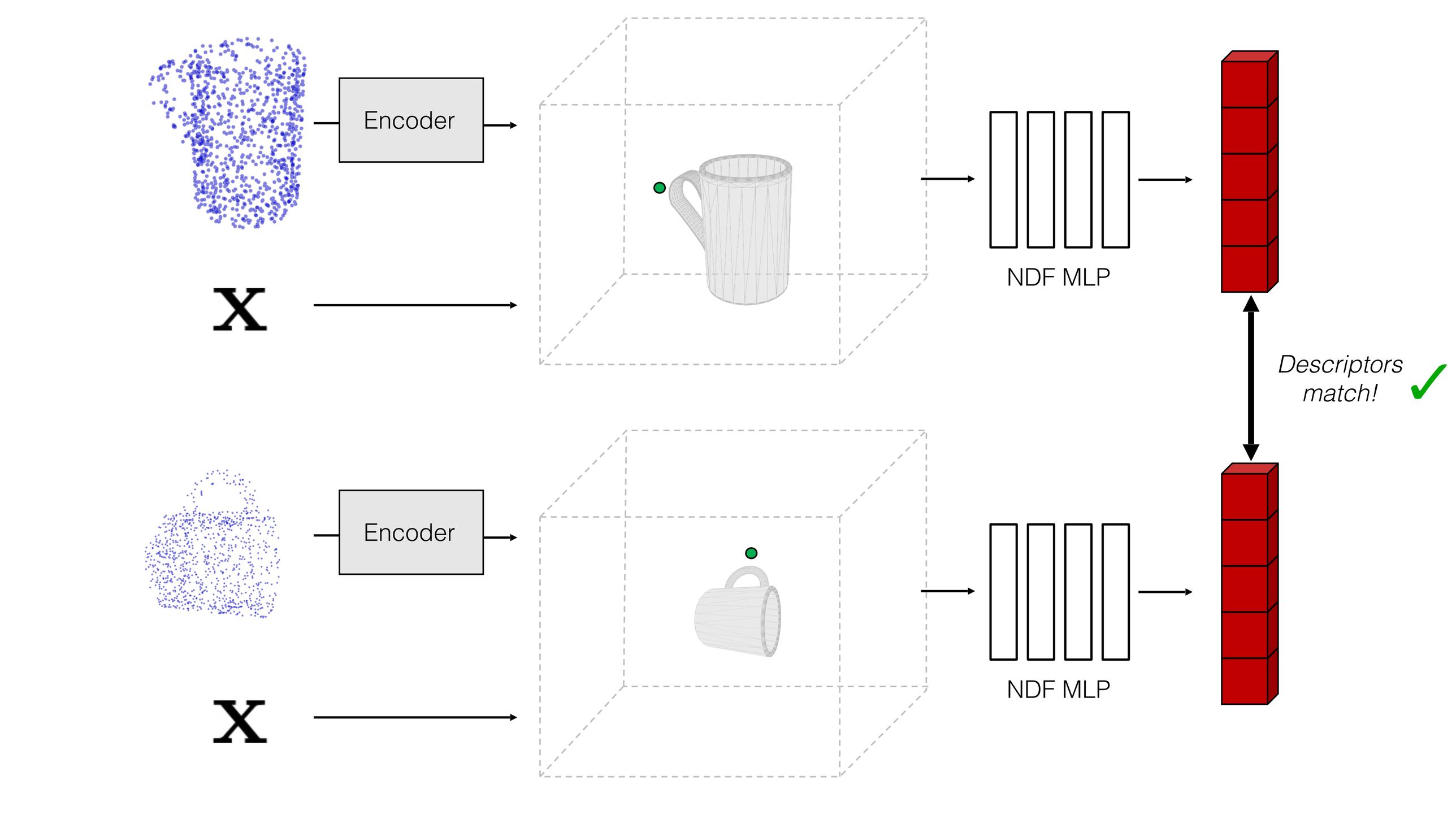
$$f(\mathbf{x}|\mathbf{P}) = f(\mathbf{T}\mathbf{x}|\mathbf{T}\mathbf{P}), \mathbf{T} \in \mathbf{SE}(3)$$

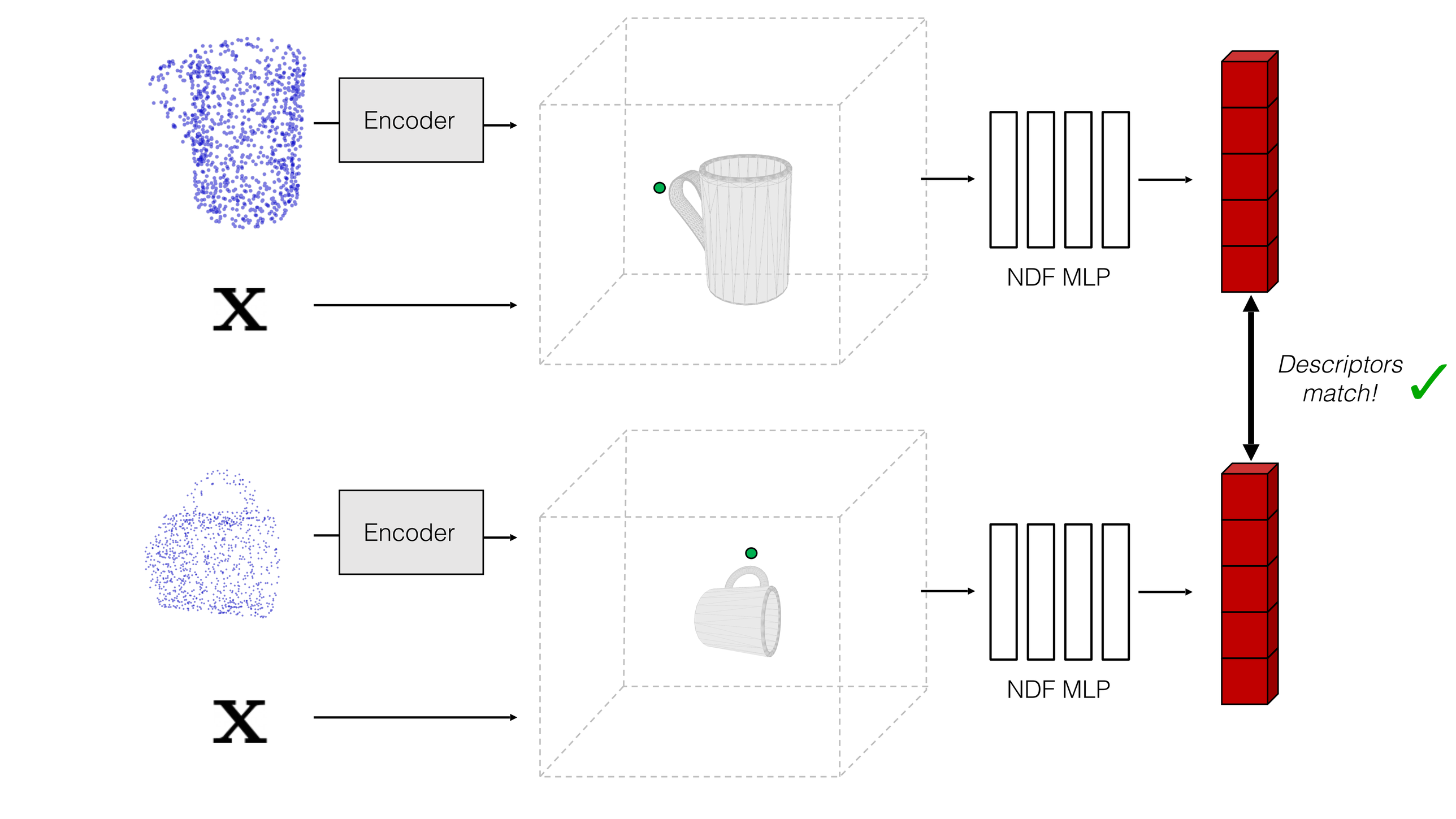
- Translation equivariant by subtracting the point cloud mean
- Rotation equivariant with *Vector Neuron* [1] layers in the encoder

[1] C. Deng, et. Al, “Vector Neurons: A General Framework for SO(3)-Equivariant Neural Networks”, ICCV 2021

escriptors match!

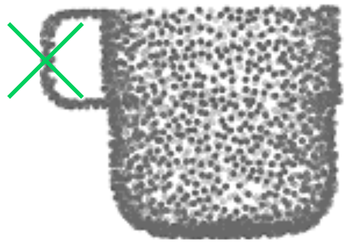
NDF MLP


Encoder
X
NDF MLP
Descriptors match!
Encoder
X
NDF MLP


Encoder
X
NDF MLP
Descriptors match!
Encoder
X
NDF MLP

Correspondence matching via NDF energy minimization

Demonstration

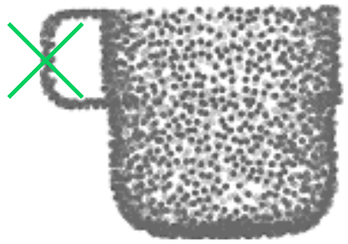


Point cloud $\hat{\mathbf{P}}$

Target point $\hat{\mathbf{x}}$

Correspondence matching via NDF energy minimization

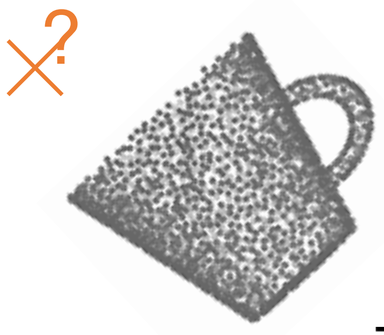
Demonstration



Point cloud $\hat{\mathbf{P}}$

Target point $\hat{\mathbf{x}}$

Test-Time

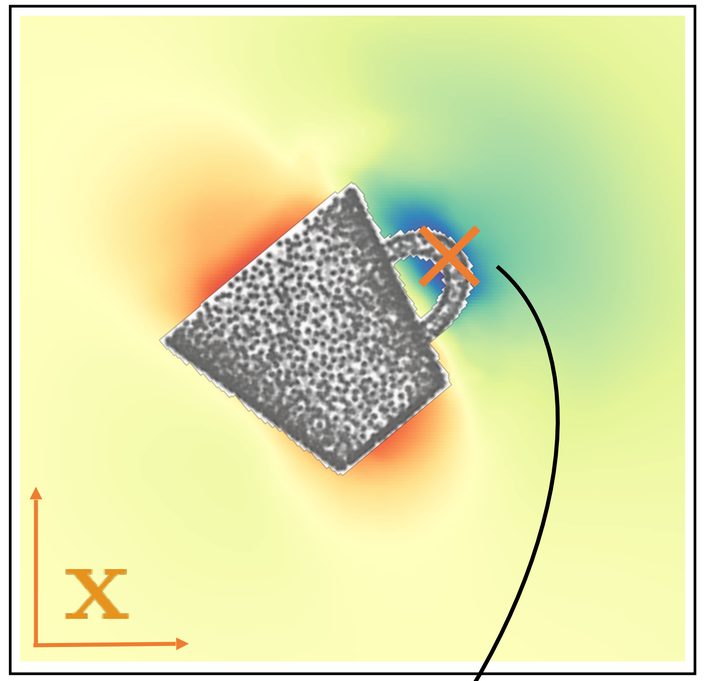


Point cloud $\mathbf{P}$

Unknown point $\mathbf{x}$

$$\left| f(\hat{\mathbf{x}}|\bar{\mathbf{P}}) - f(\mathbf{x}|\mathbf{P}) \right| =$$

Descriptor Difference: $E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathbf{x}})$



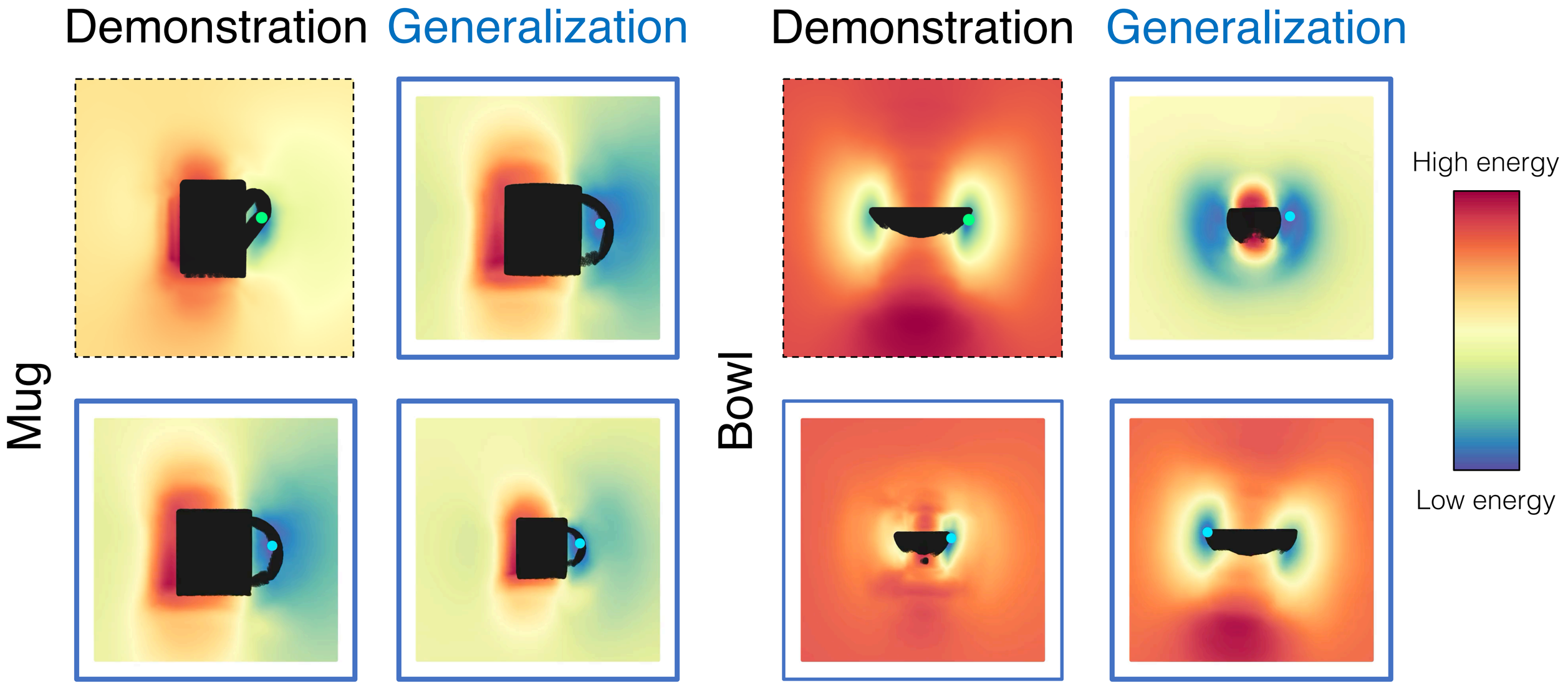
High energy



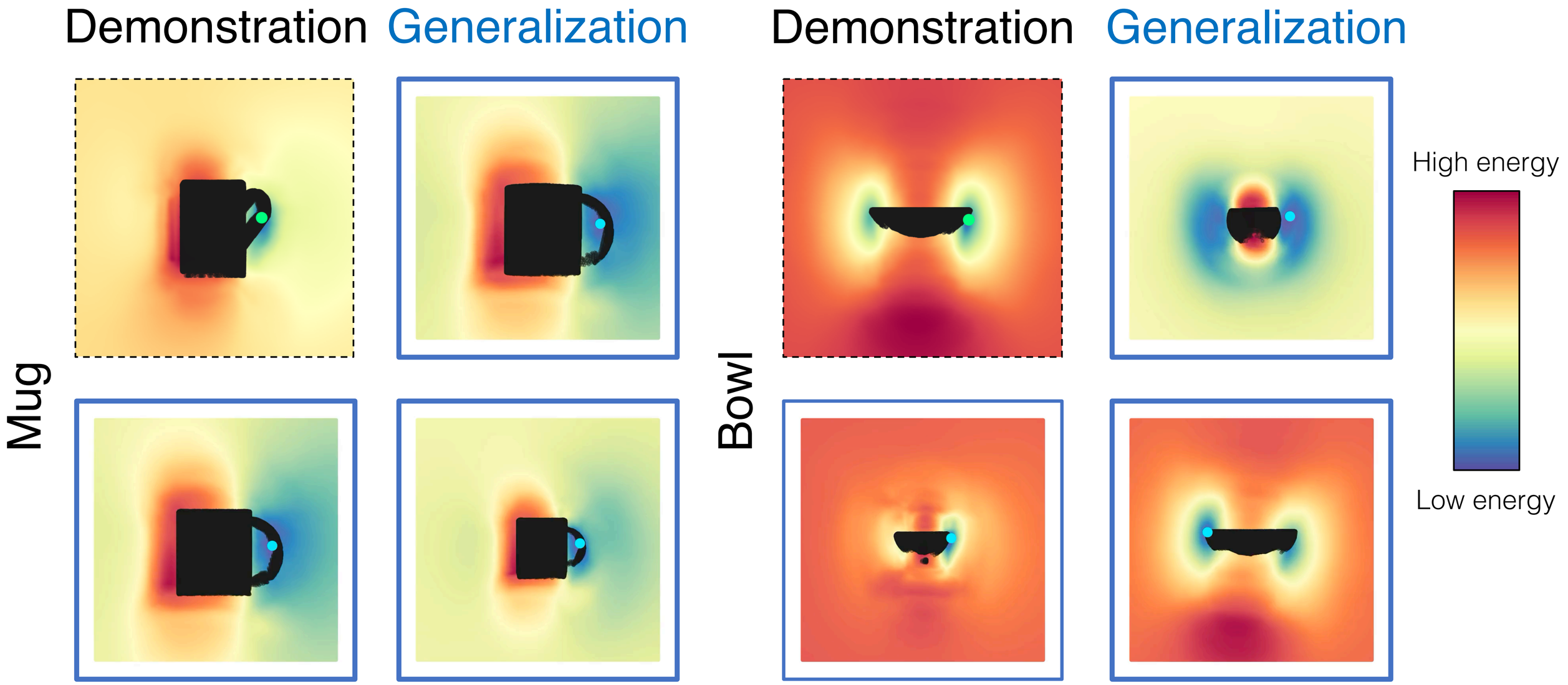
Low energy

$$\underset{\mathbf{x}}{\operatorname{argmin}}\ E(\mathbf{x}|\hat{\mathbf{P}}, \mathbf{P}, \hat{\mathbf{x}})$$

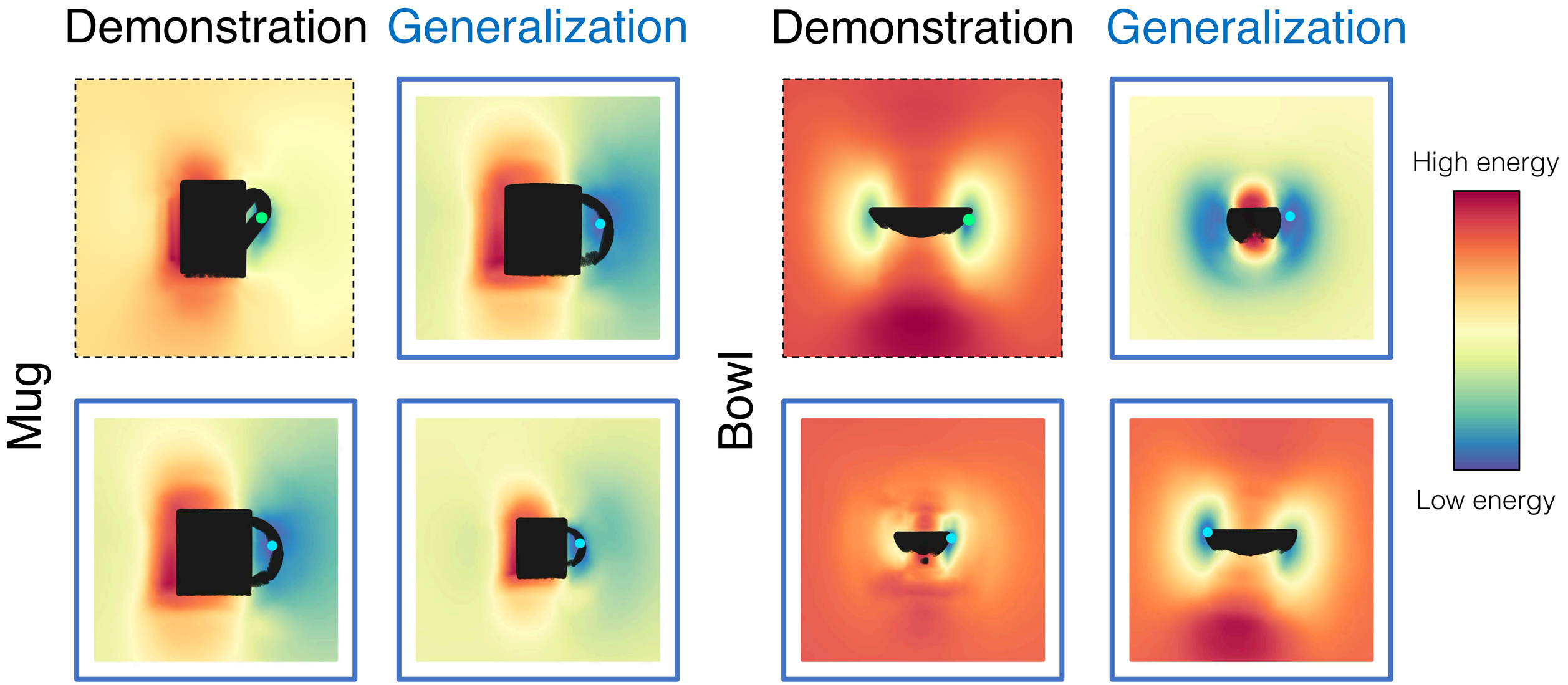
Correspondence matching via NDF energy minimization



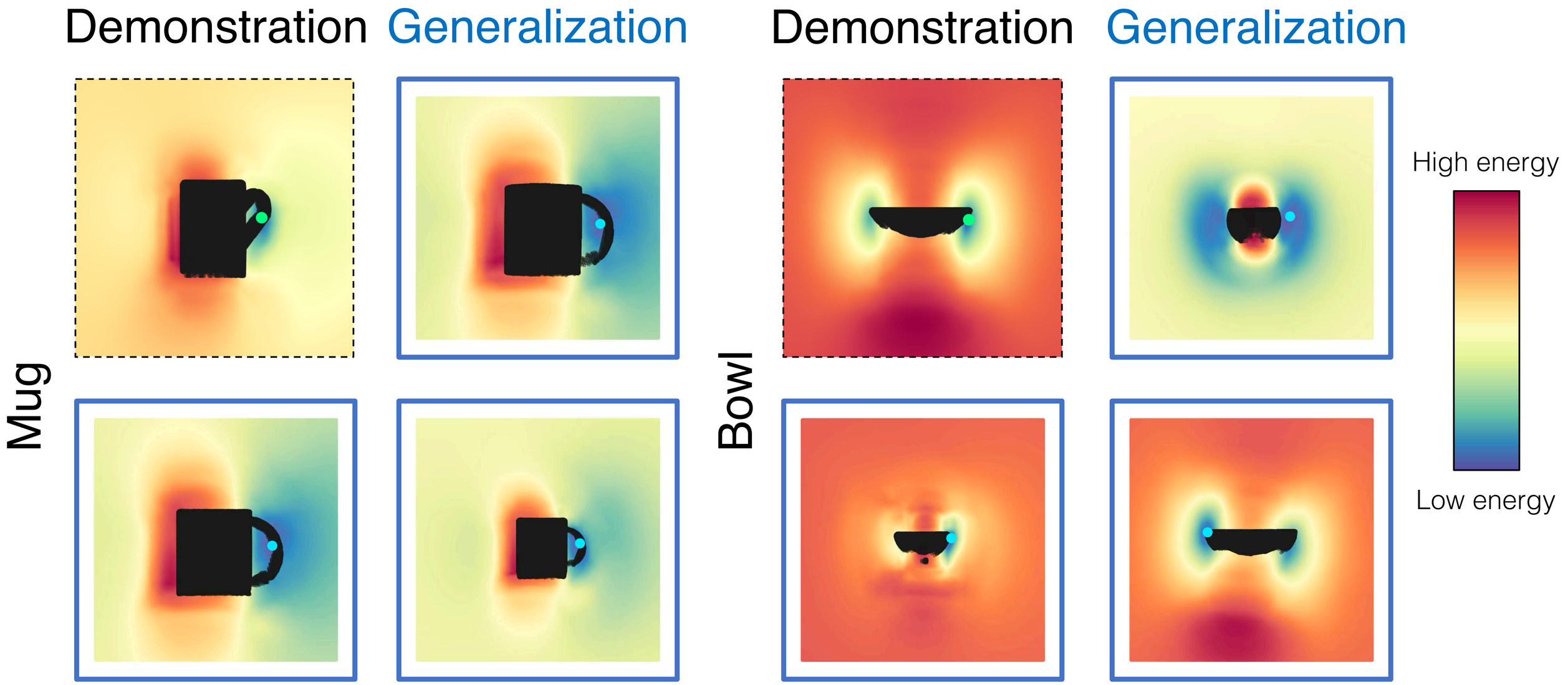
Correspondence matching via NDF energy minimization



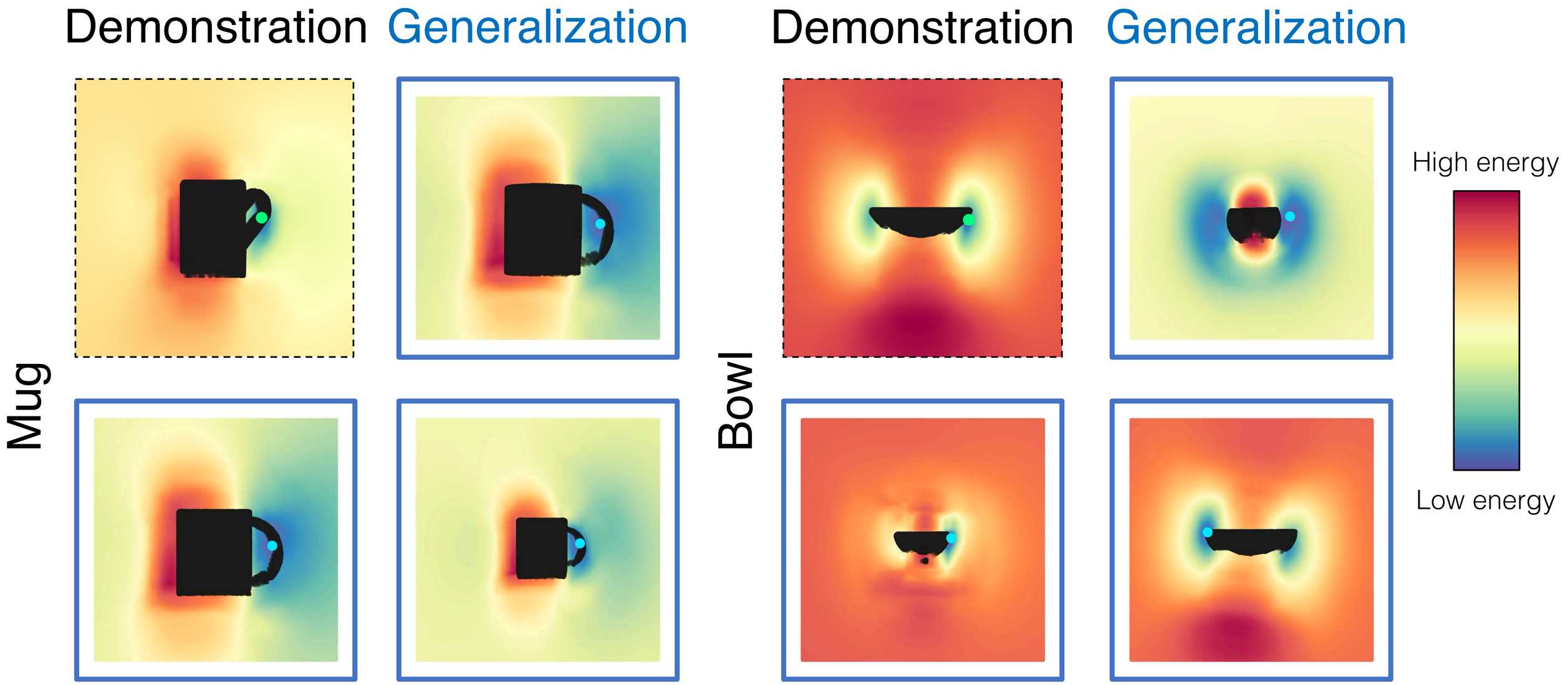
Correspondence matching via NDF energy minimization



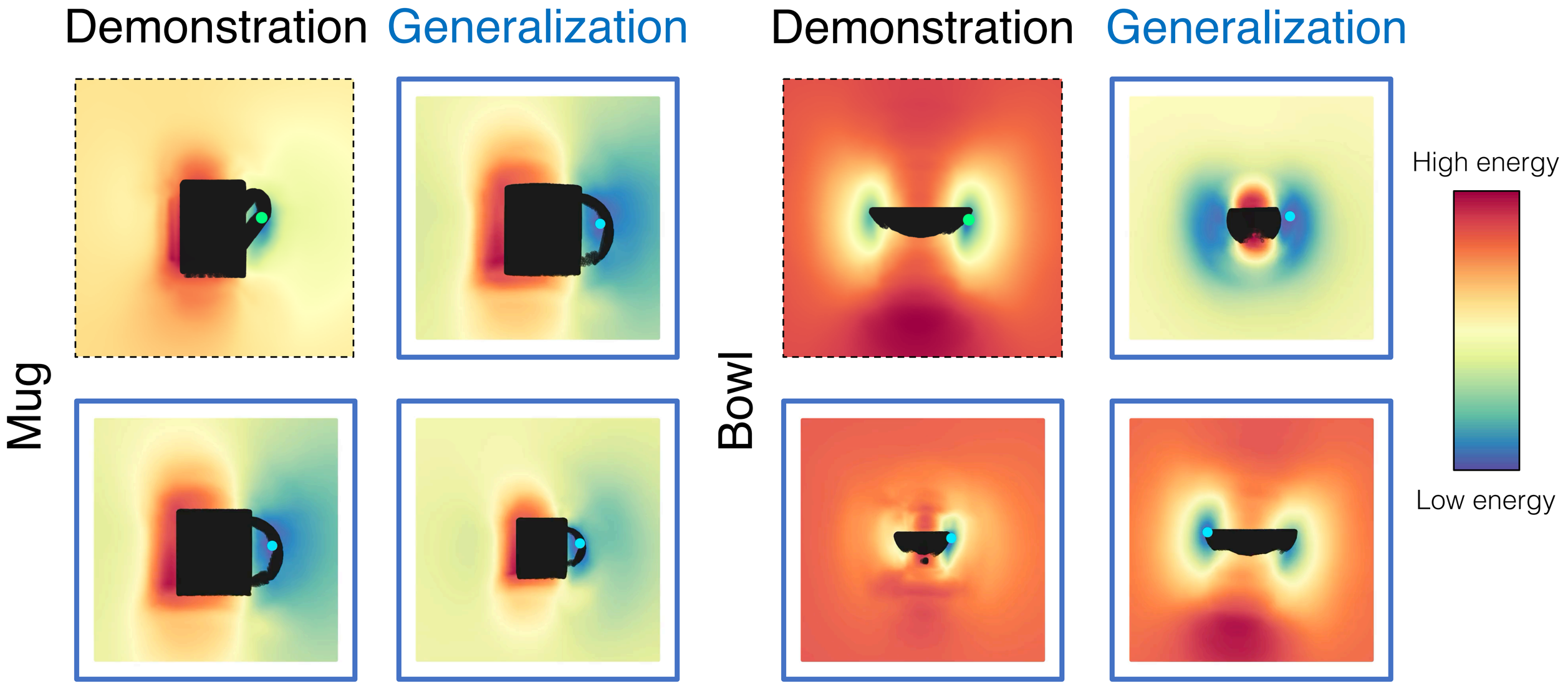
Correspondence matching via NDF energy minimization



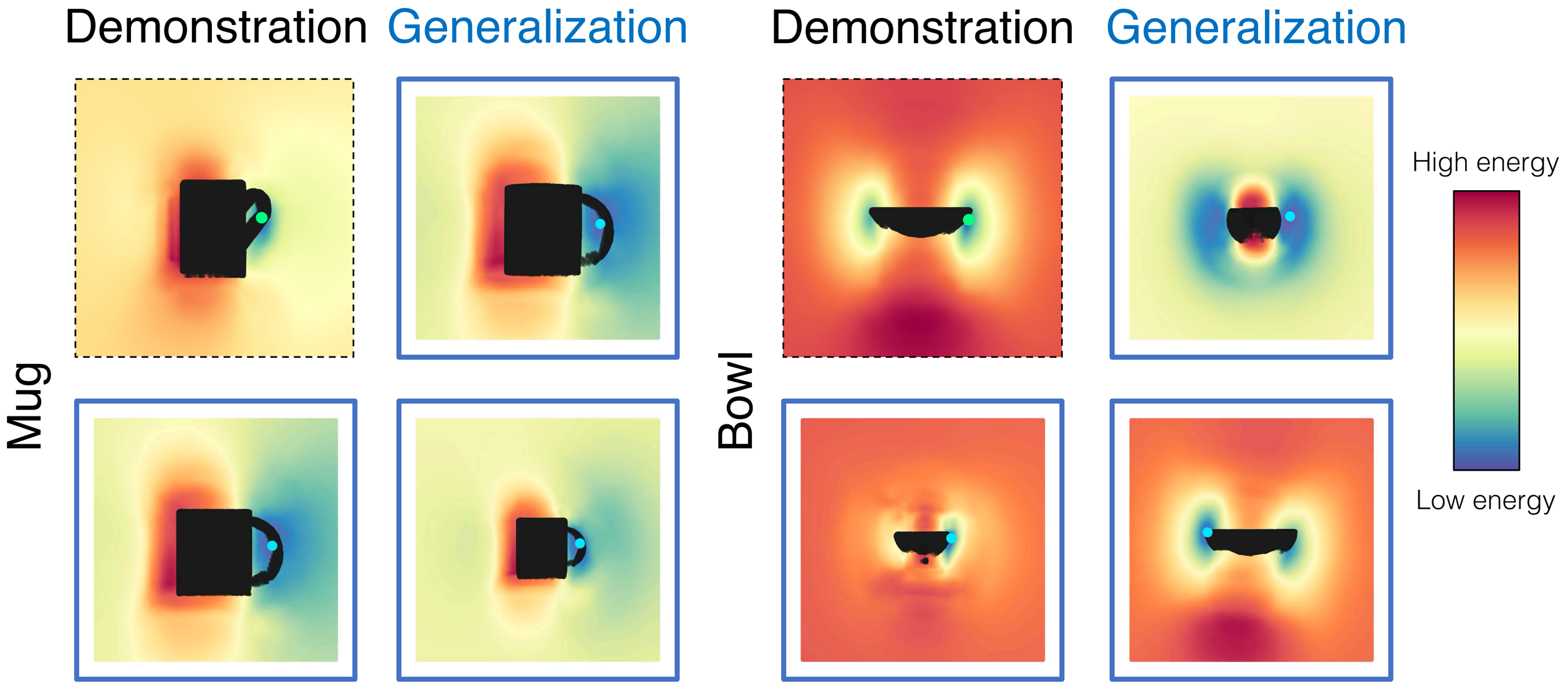
Correspondence matching via NDF energy minimization



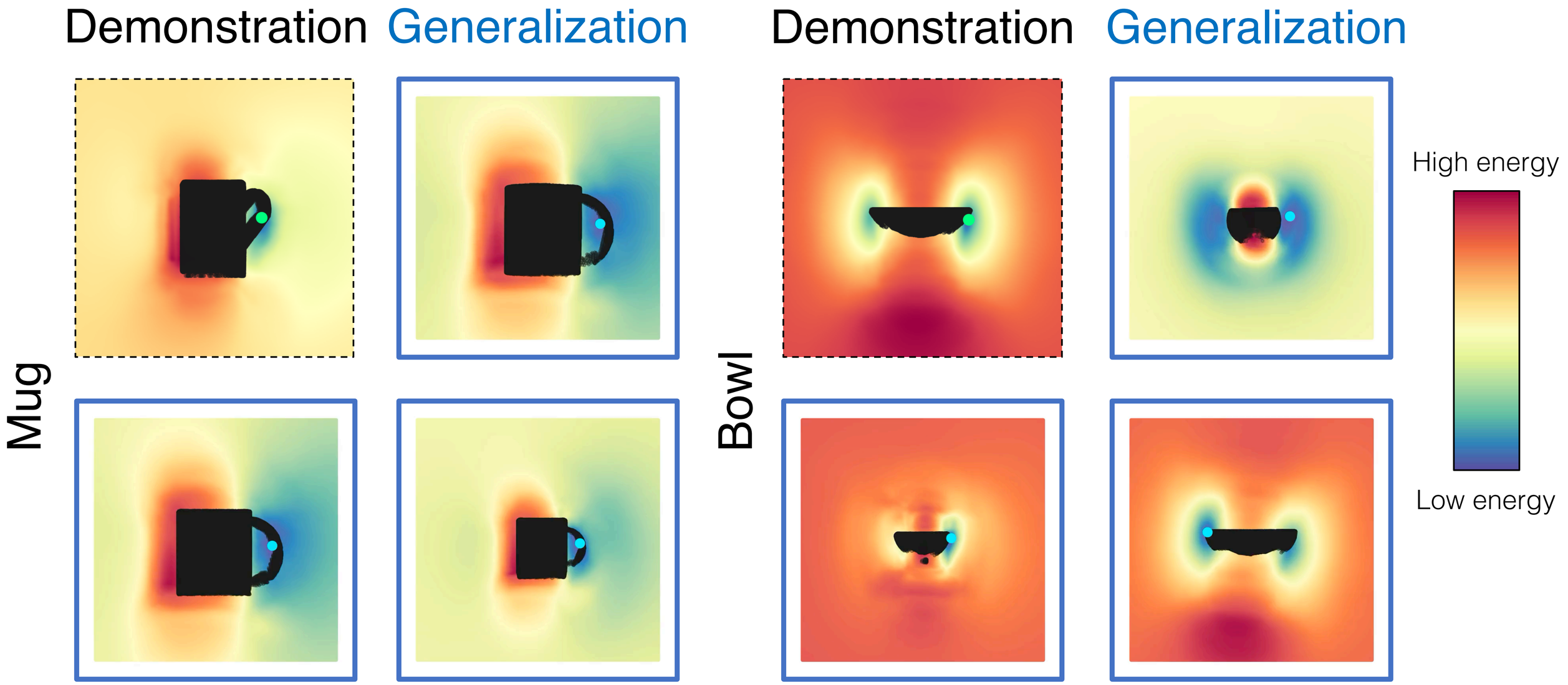
Correspondence matching via NDF energy minimization



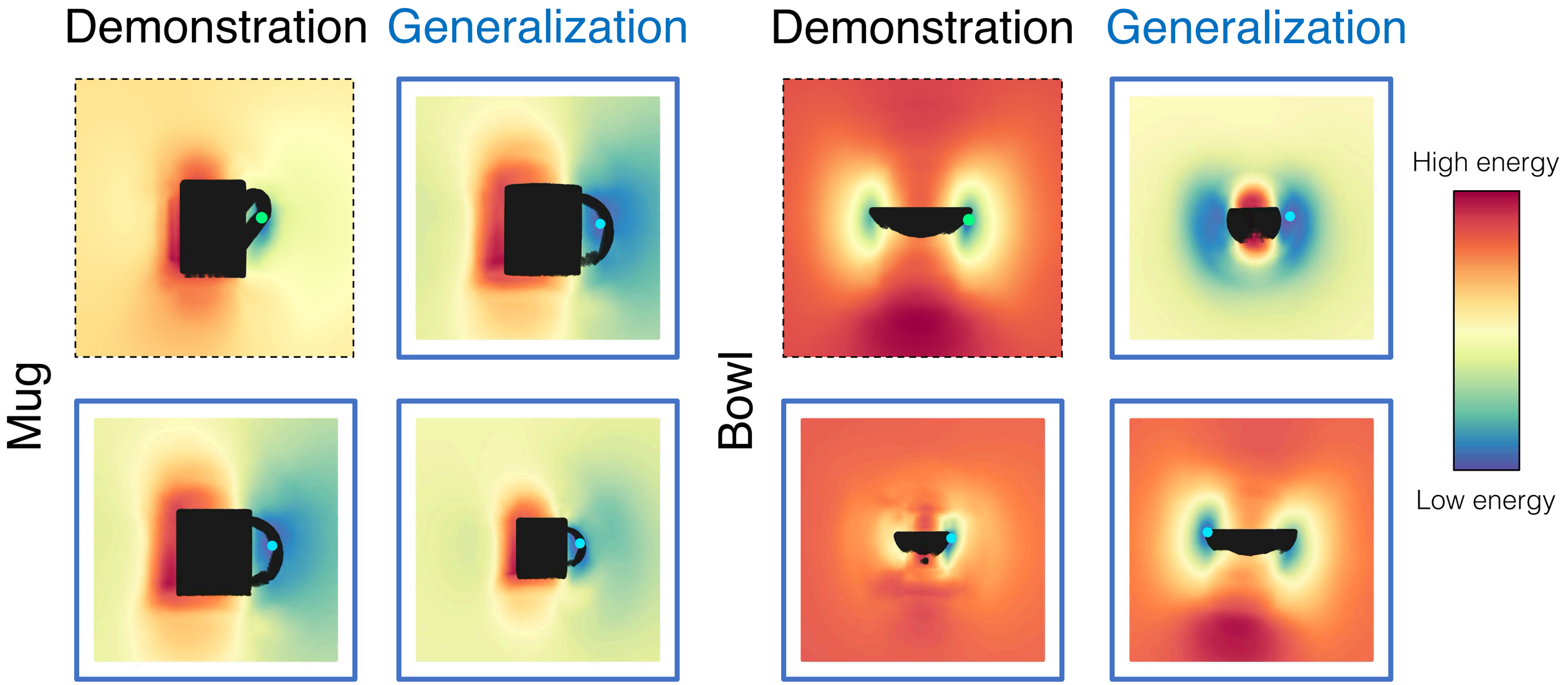
Correspondence matching via NDF energy minimization



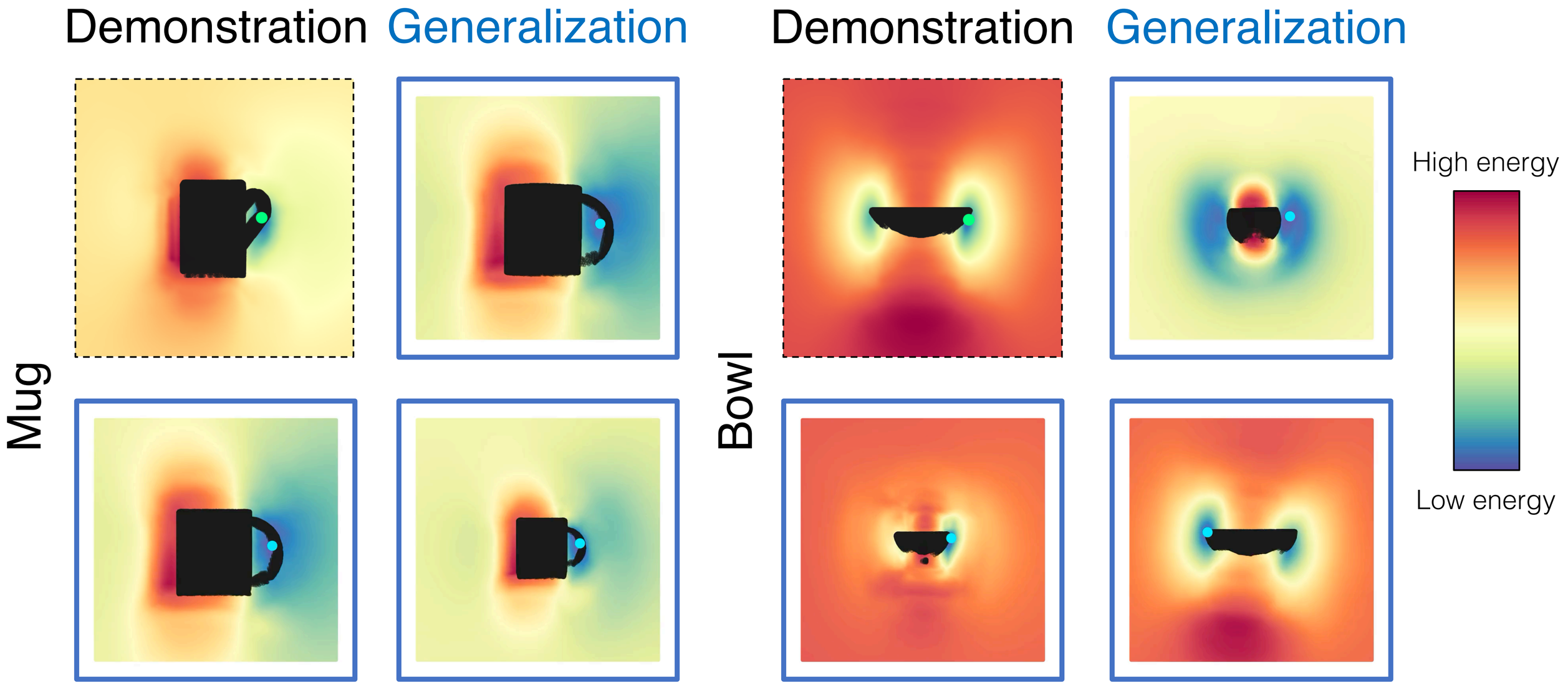
Correspondence matching via NDF energy minimization

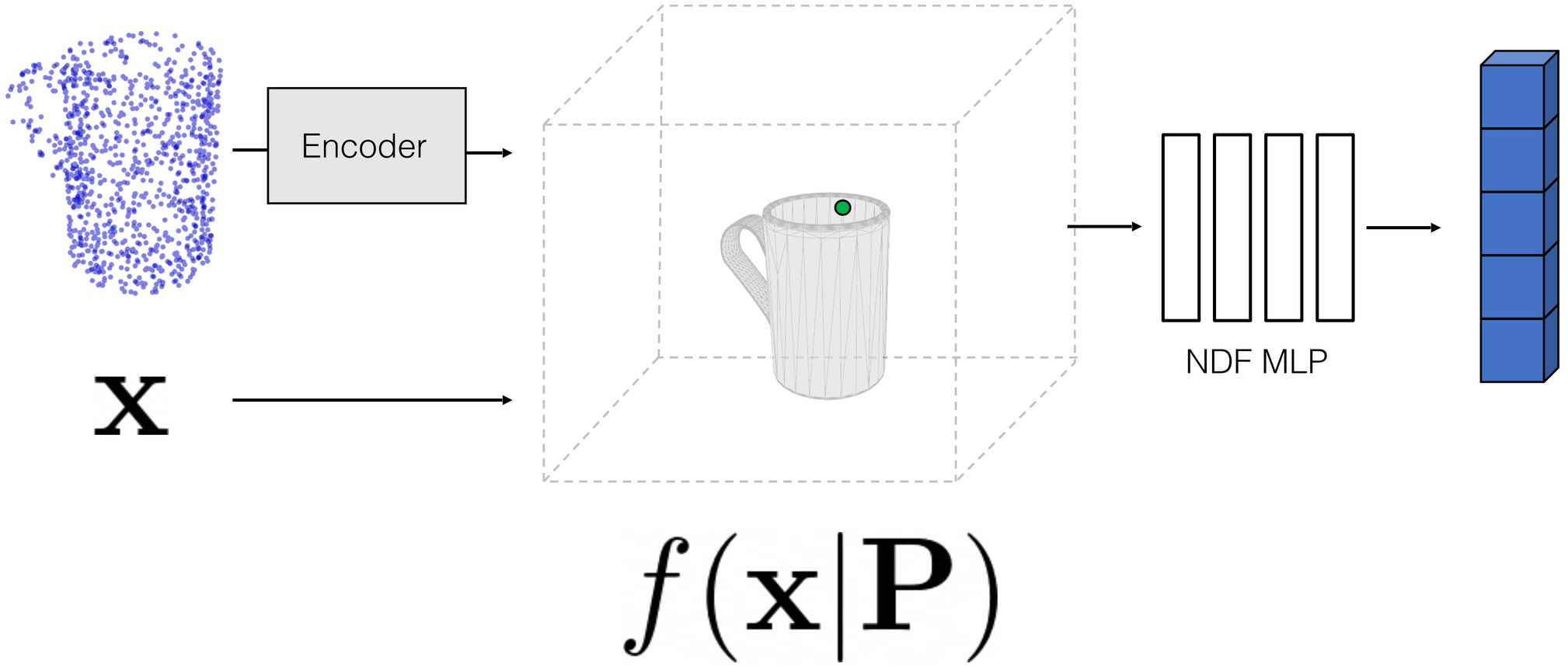


Correspondence matching via NDF energy minimization



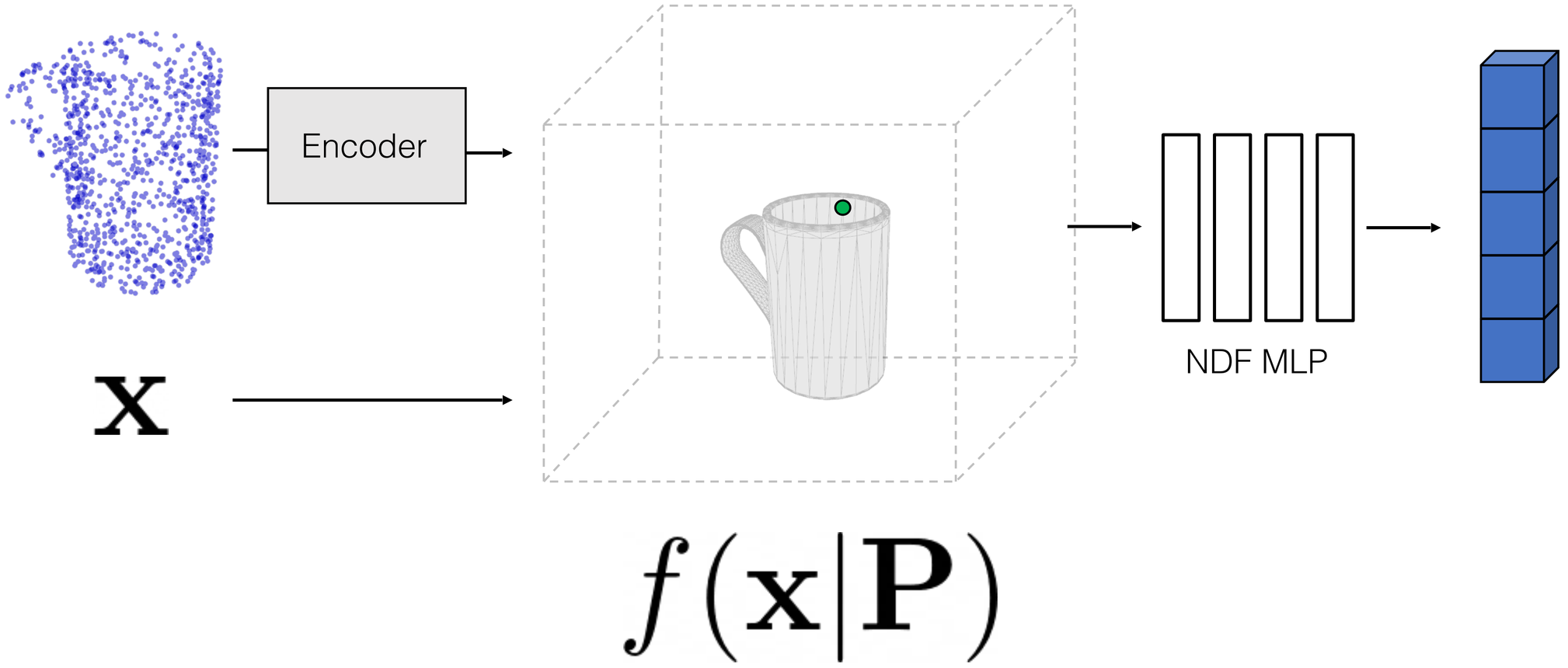
Correspondence matching via NDF energy minimization





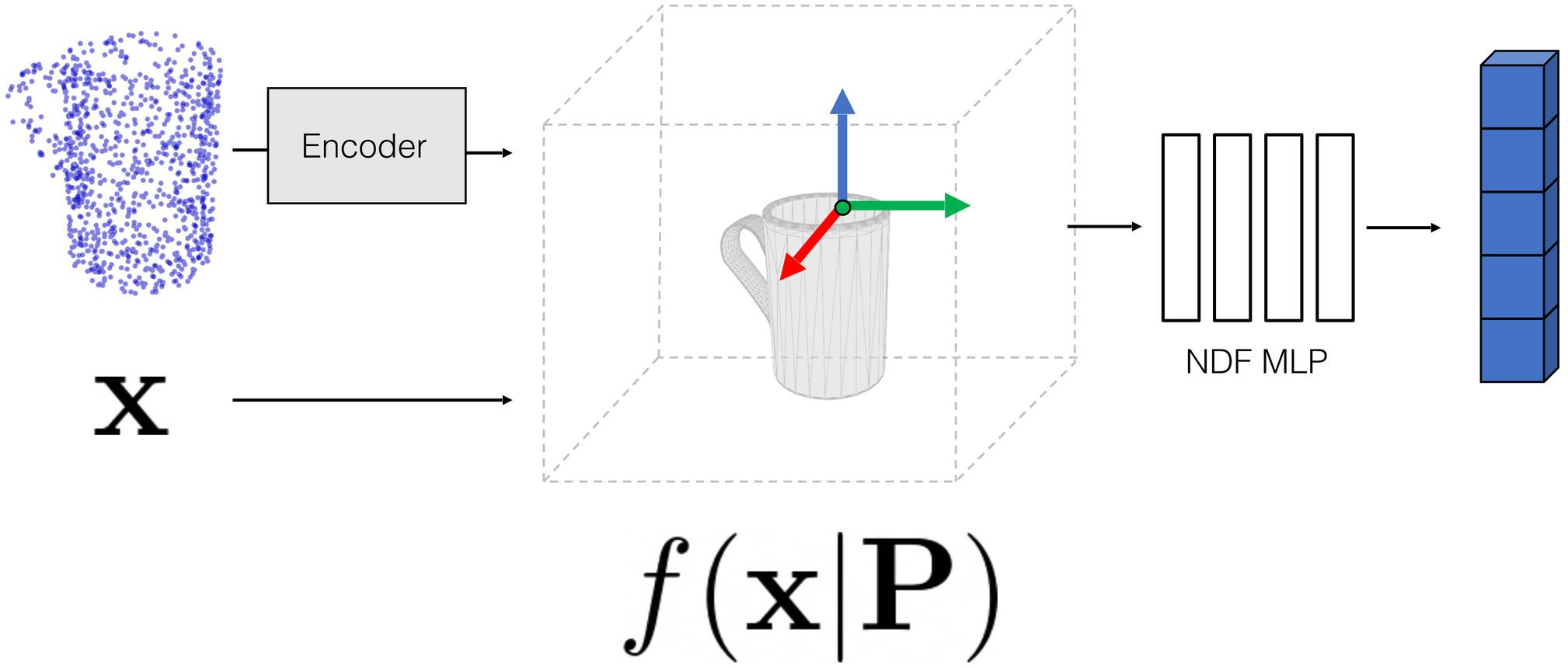
$\mathbf{x}$

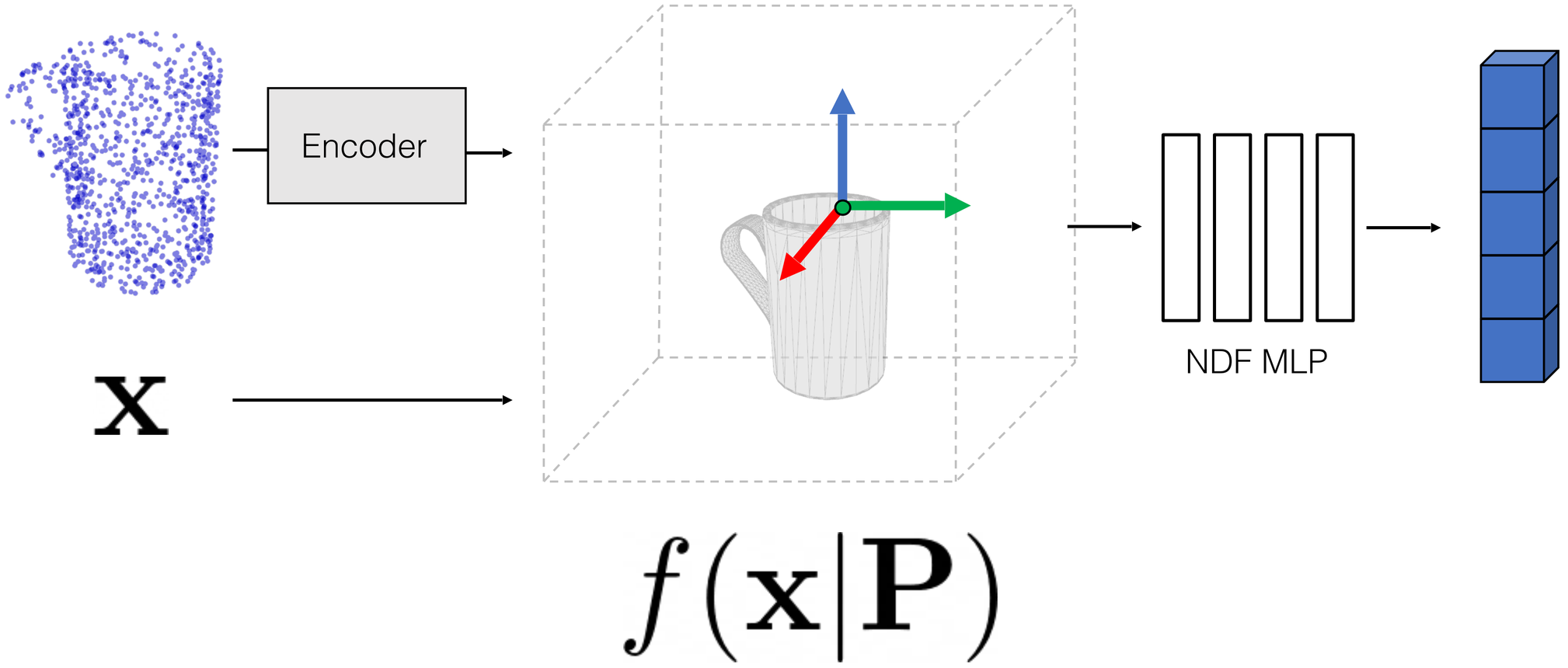
$f(\mathbf{x}|\mathbf{P})$

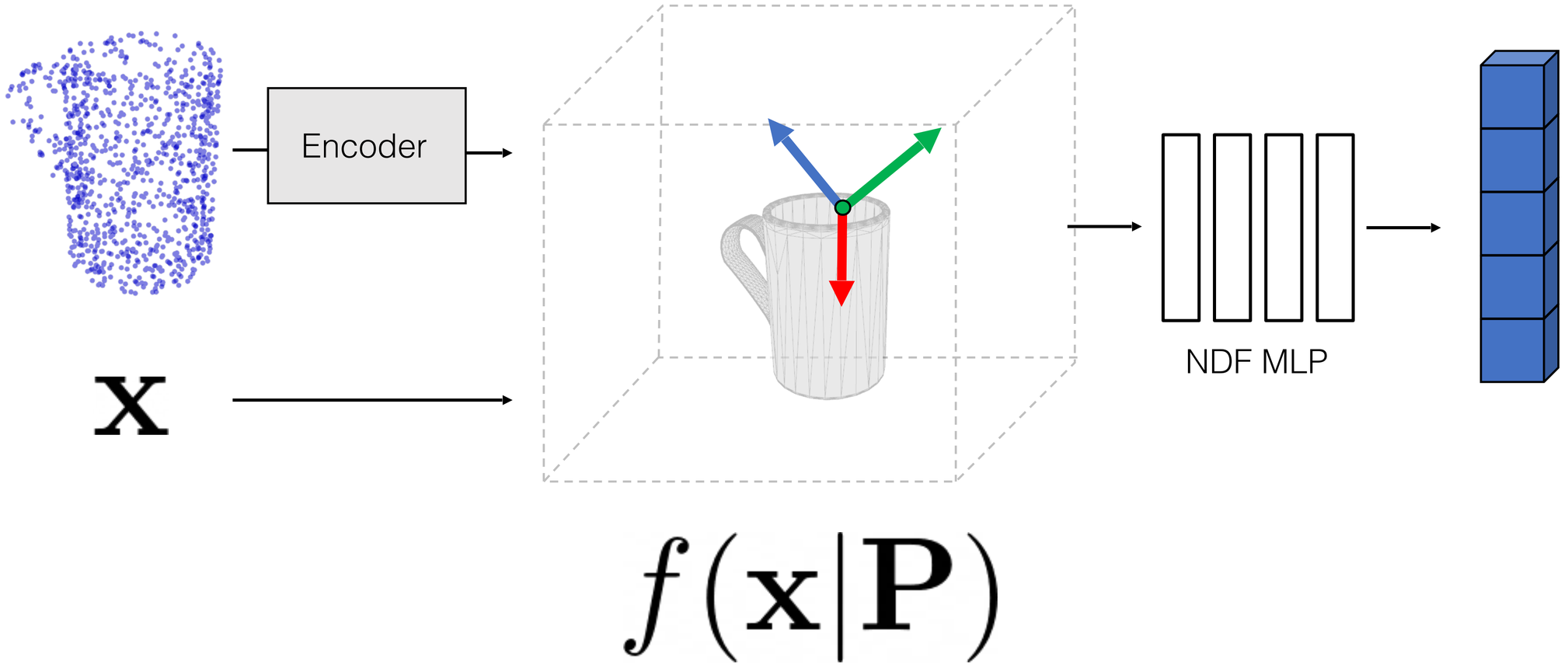


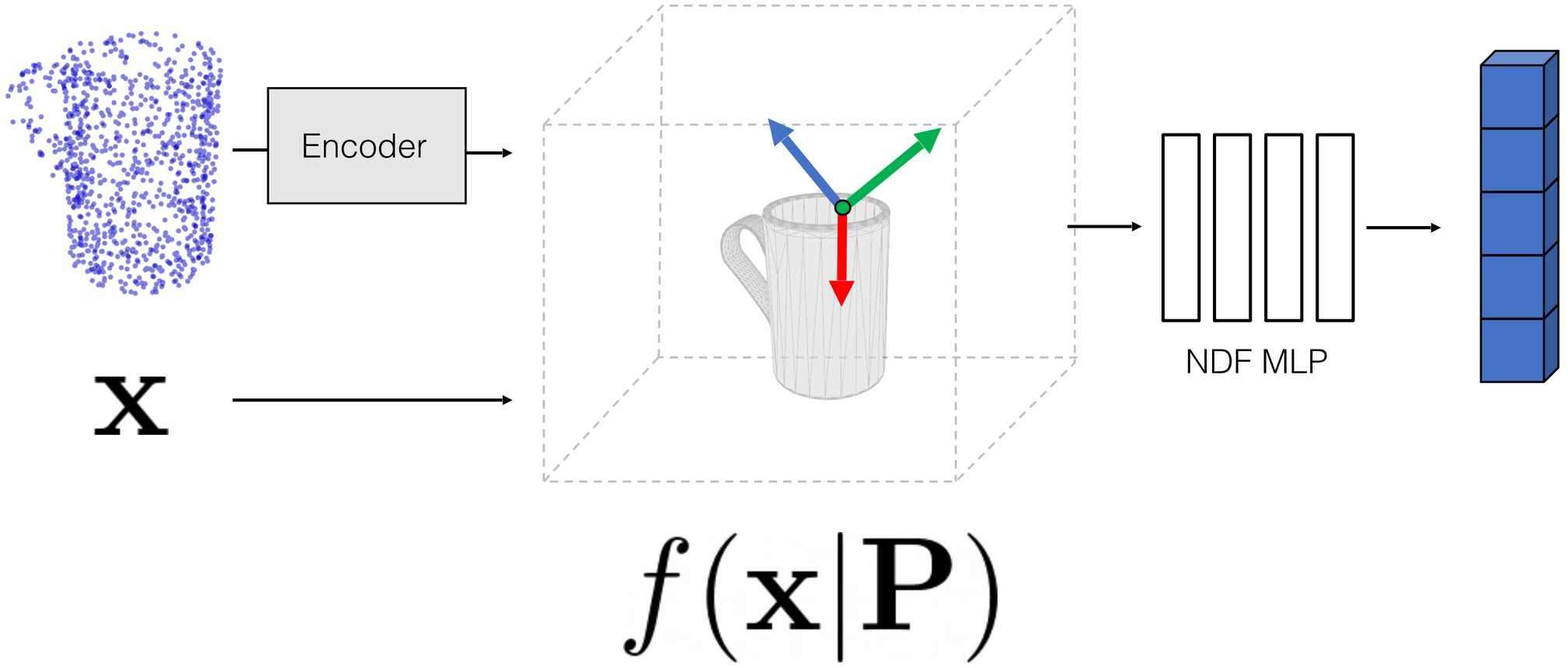
$\mathbf{x}$

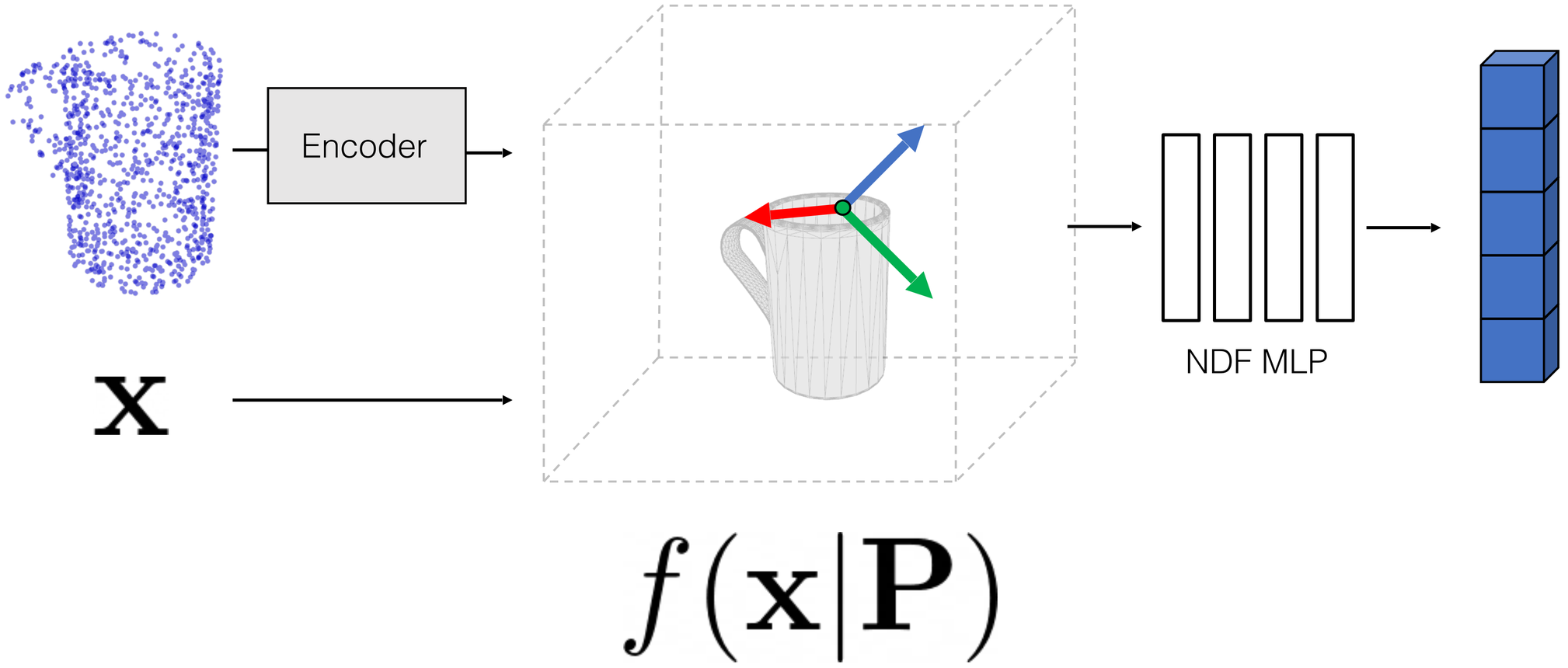
$f(\mathbf{x}|\mathbf{P})$

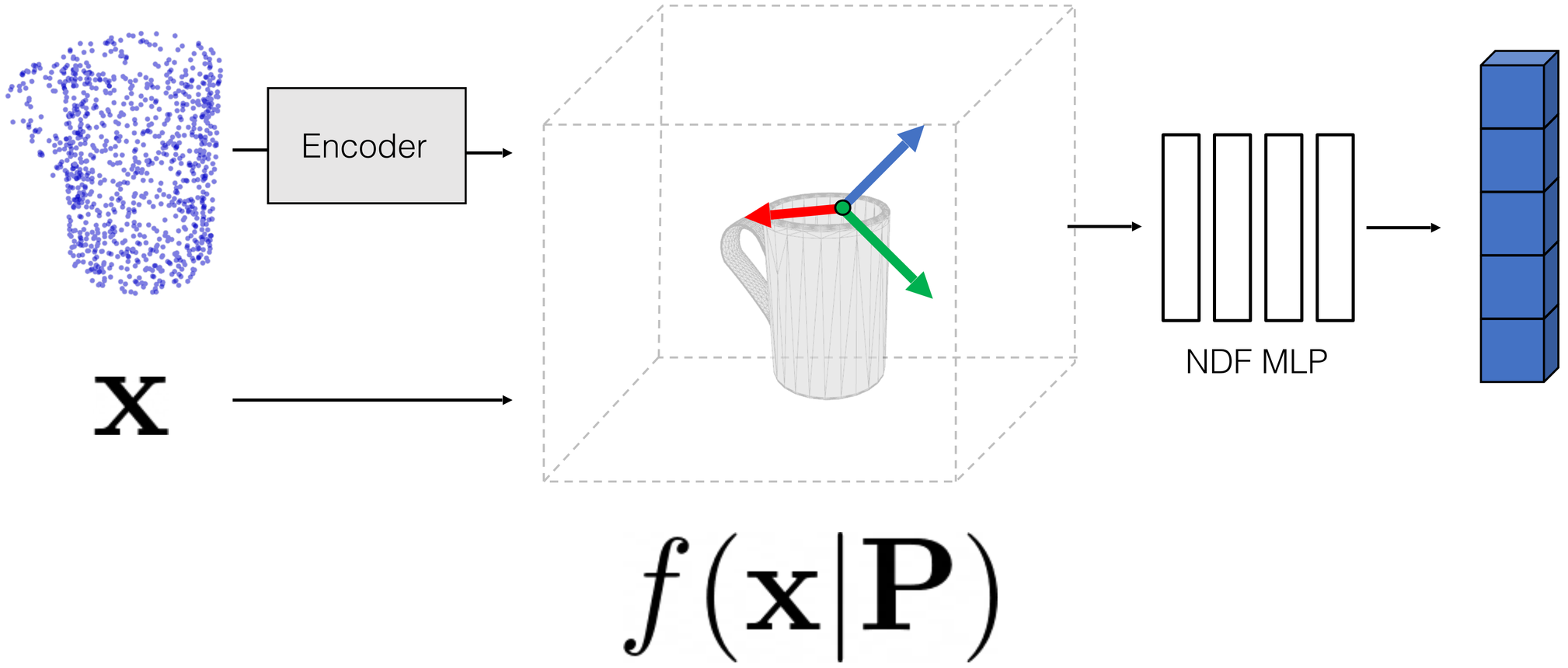

Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$

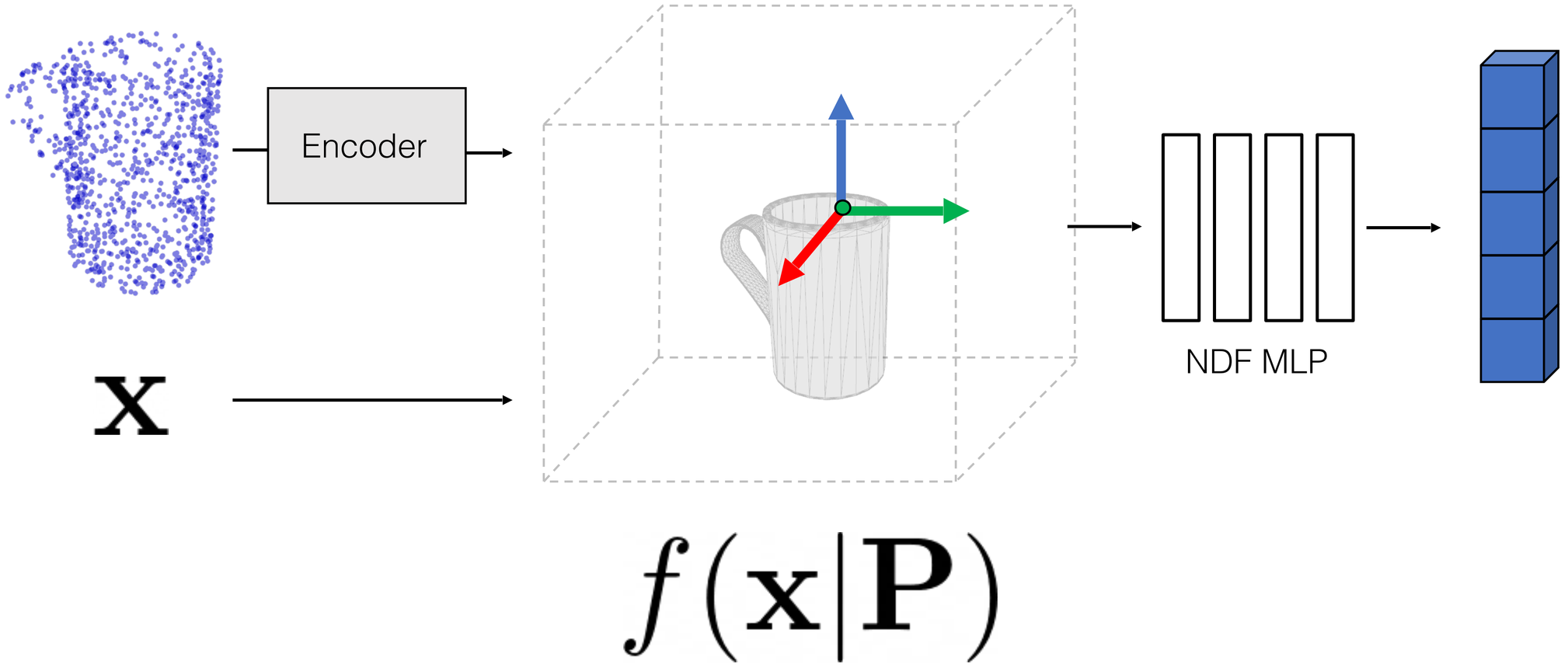

Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$

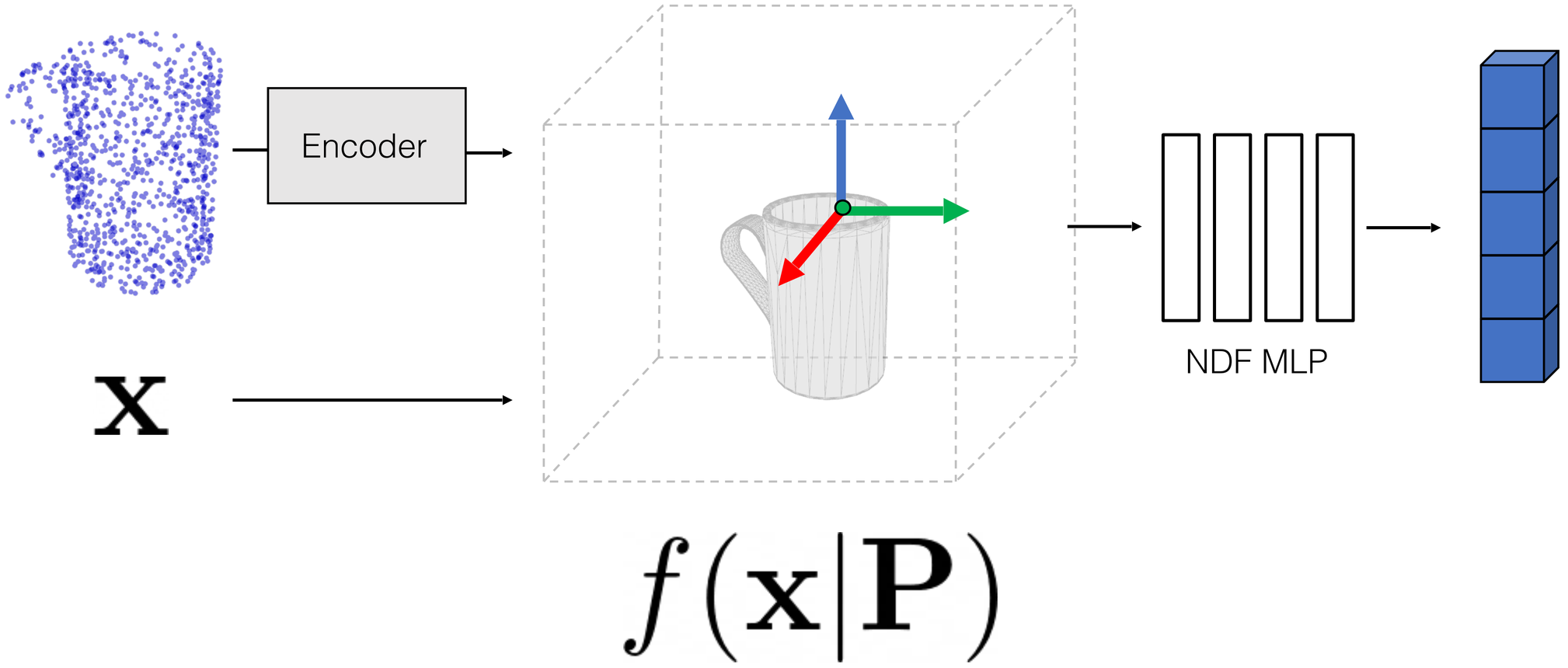

Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$

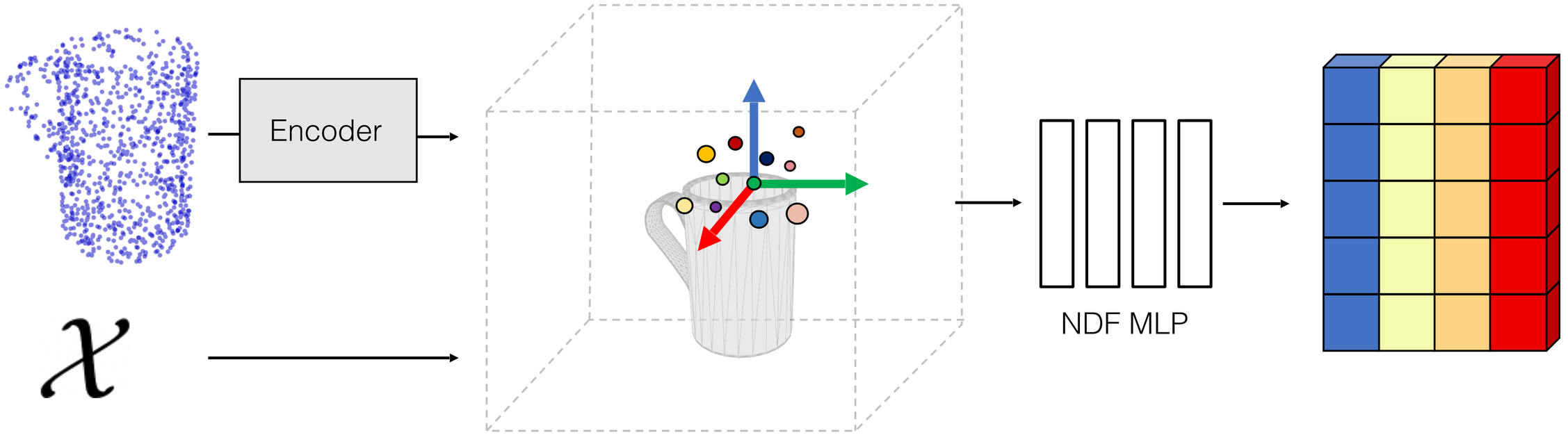

Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$


Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$

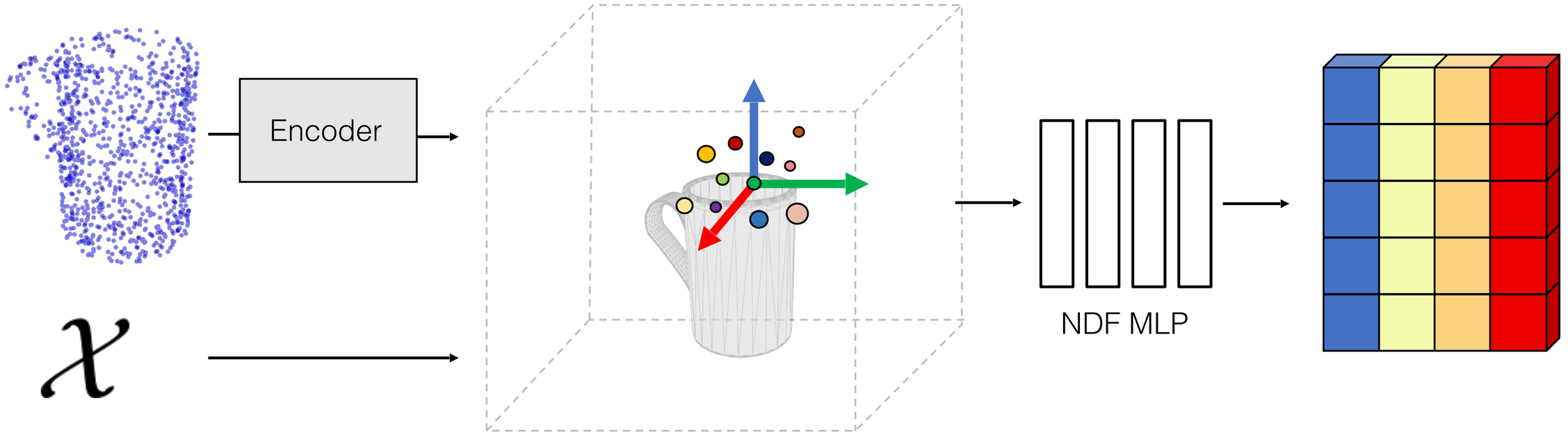

Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$


Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$


Encoder
$\mathbf{x}$
NDF MLP
$f(\mathbf{x}|\mathbf{P})$



$$\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$$



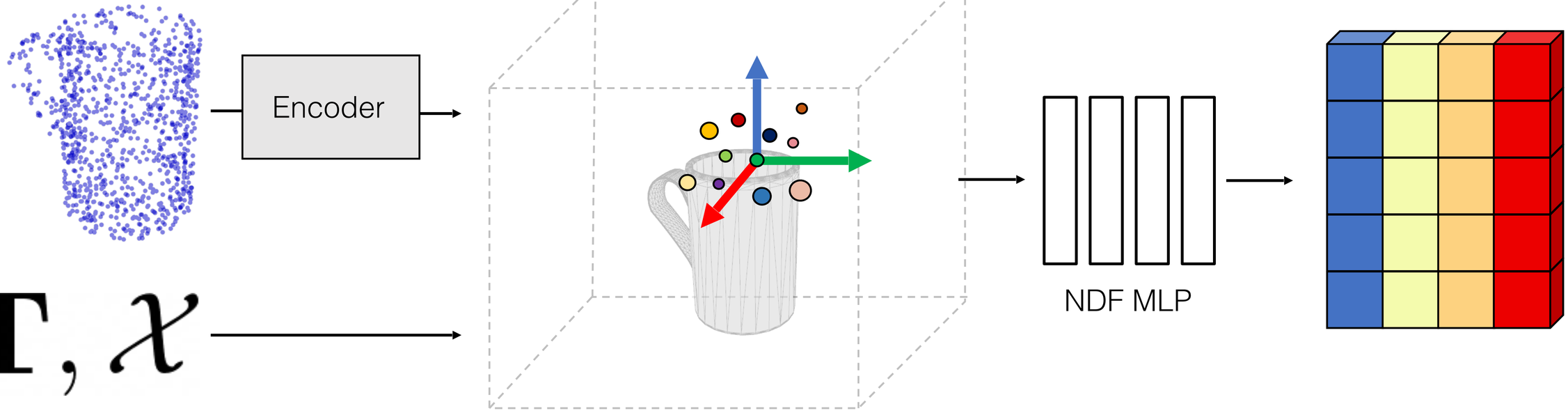
$$\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]$$

Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

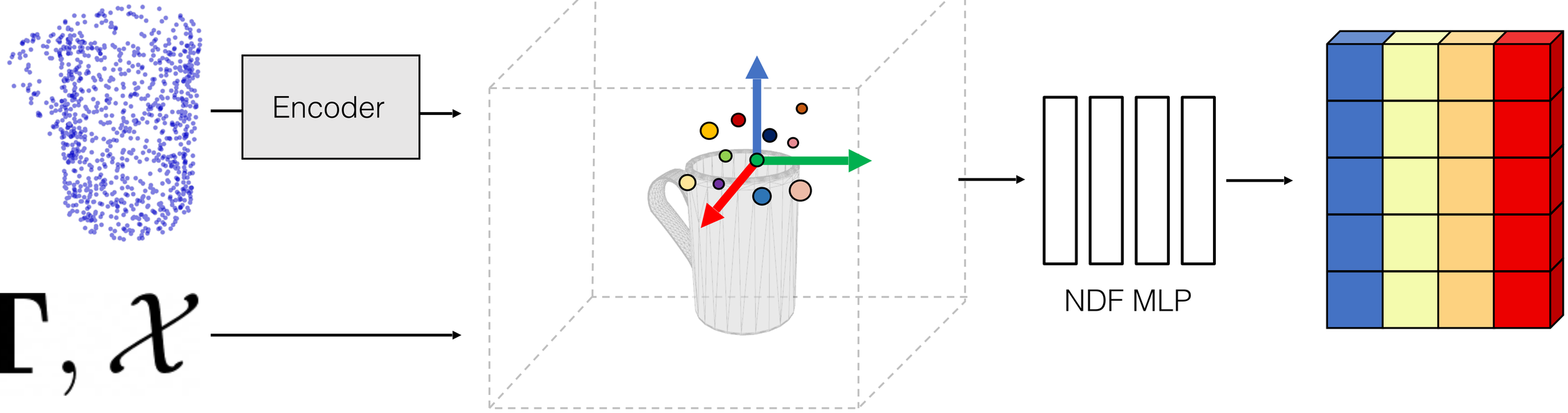


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

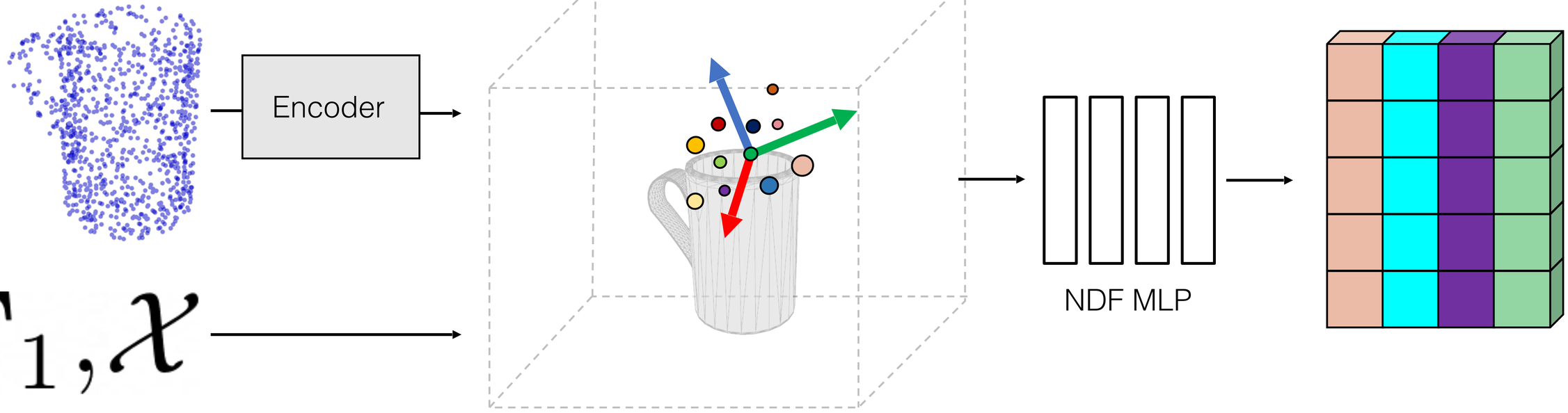


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

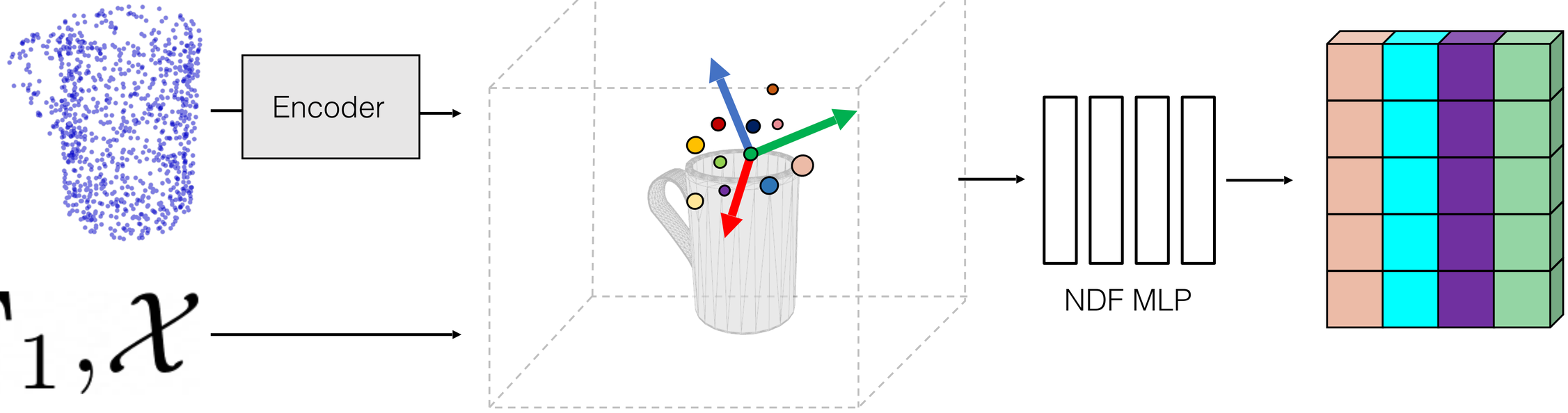


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

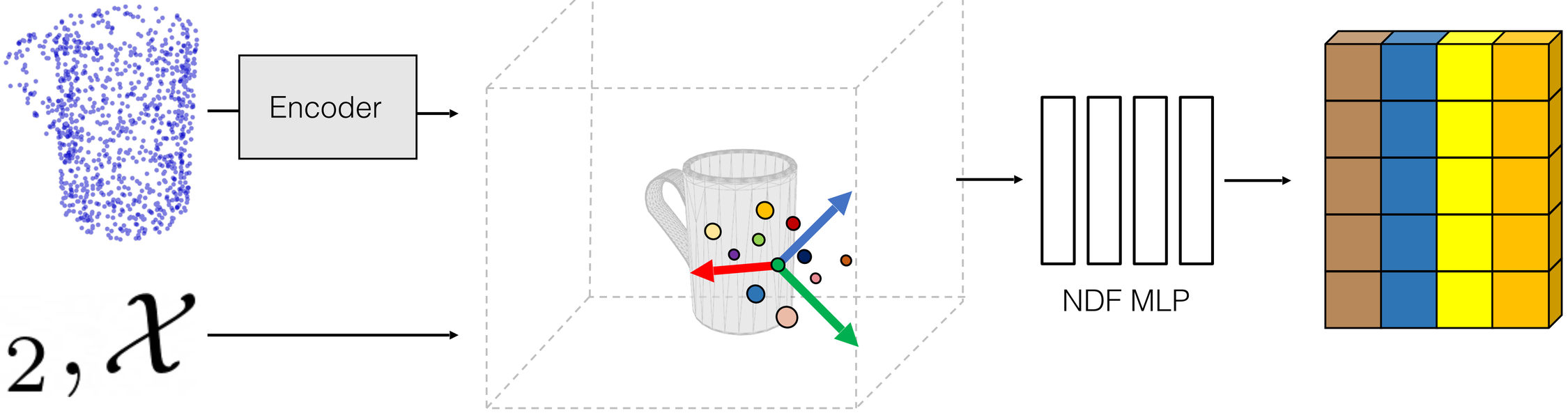


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

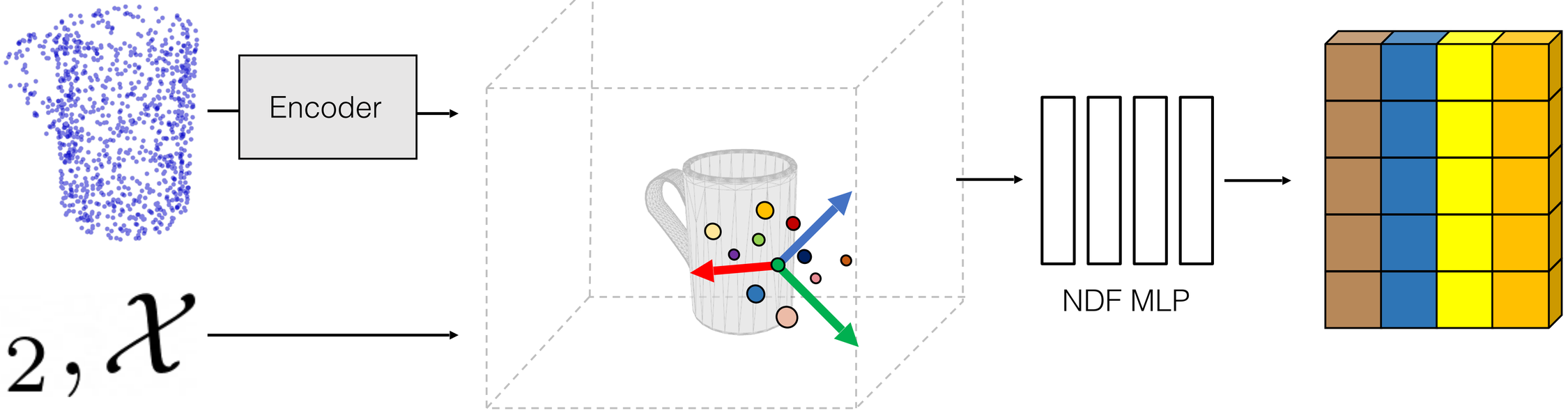


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

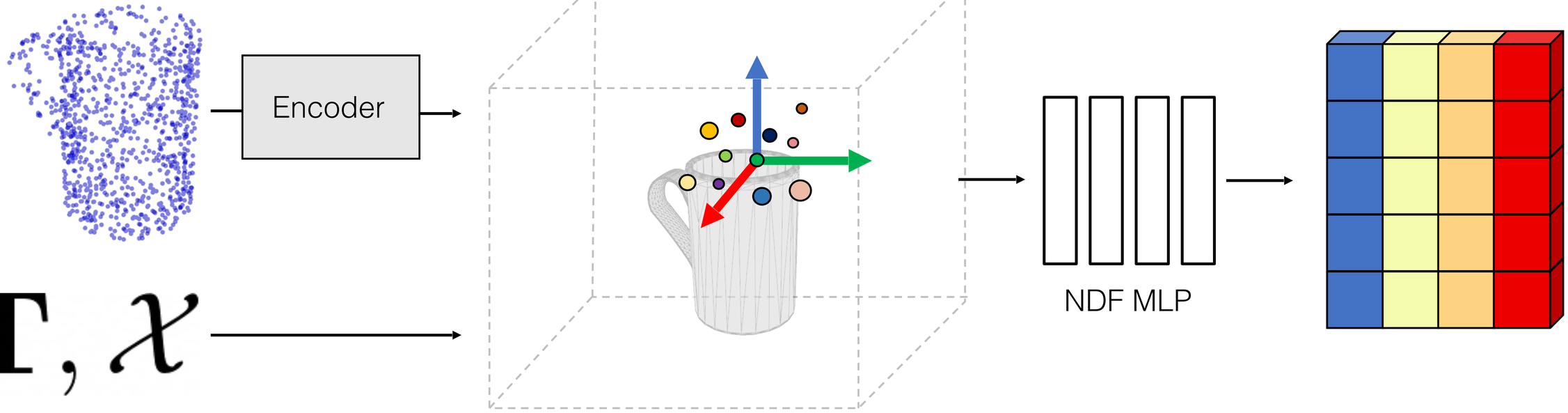


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$

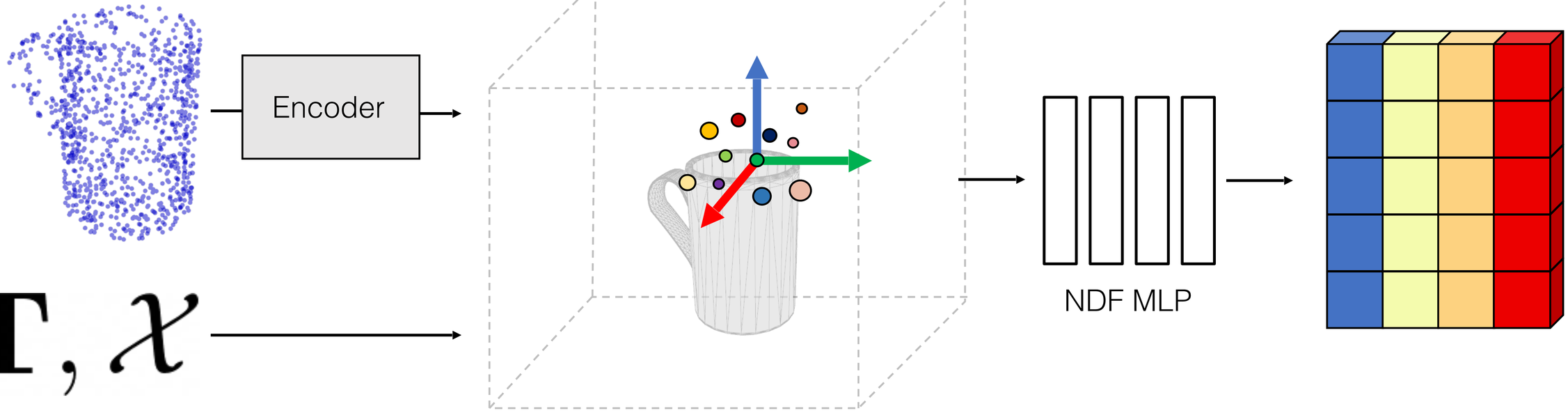


Neural Pose Descriptor Fields

$$\mathbf{T} \in \mathbf{SE}(3)$$

$$\mathbf{T} = (\mathbf{R}, \mathbf{t})$$

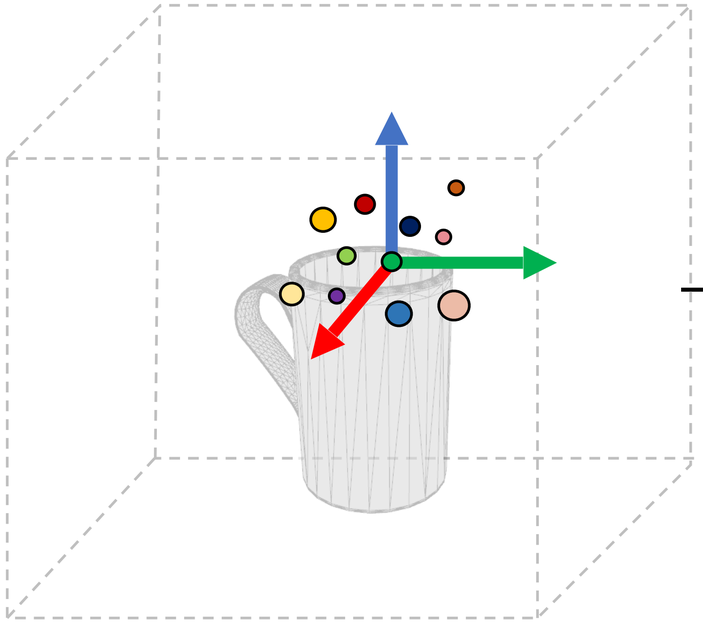
$$F(\mathbf{T}|\mathbf{P}) = \bigoplus_{\mathbf{x}_i \in \mathcal{X}} f(\mathbf{R}\mathbf{x}_i + \mathbf{t})$$



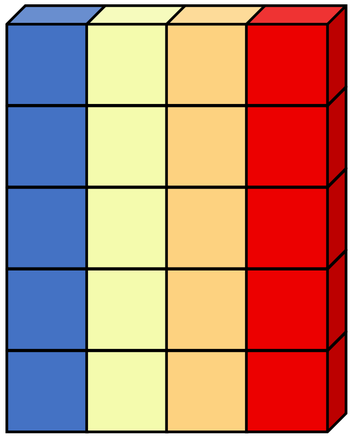


Encoder

$\mathbf{T}, \mathcal{X}$



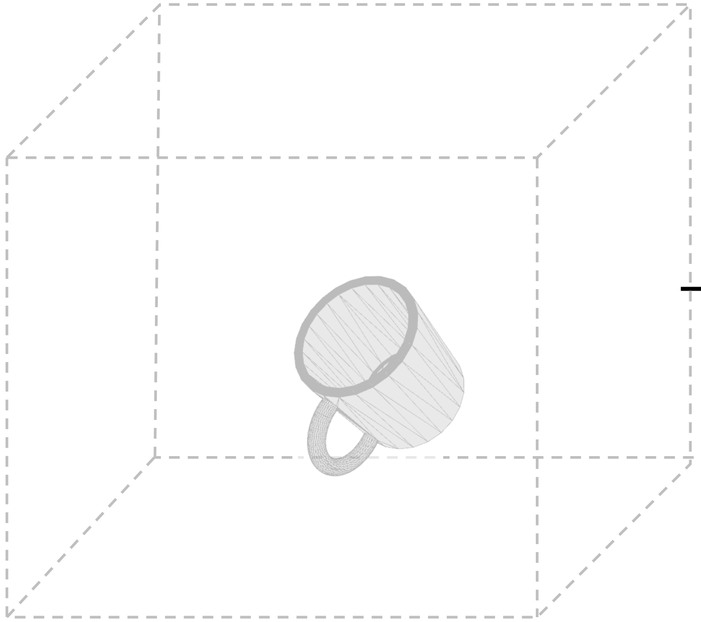
NDF MLP





Encoder

$\mathbf{T}, \mathcal{X}$

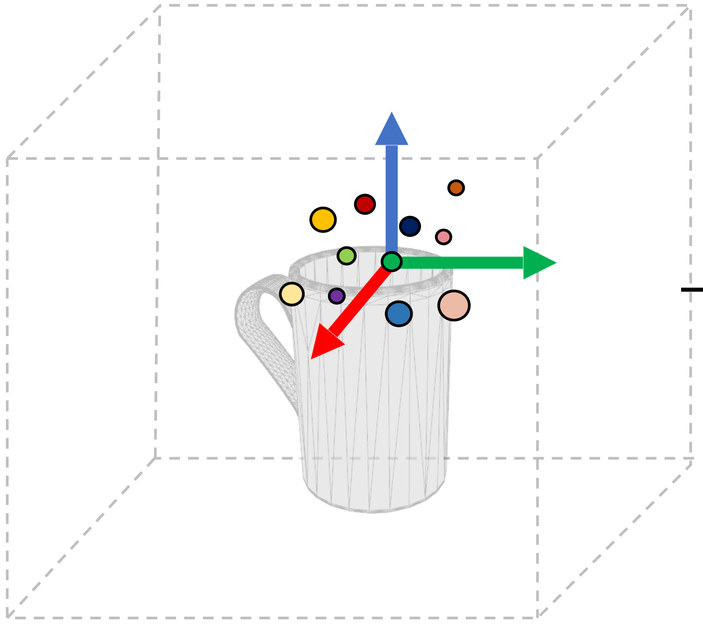


NDF MLP

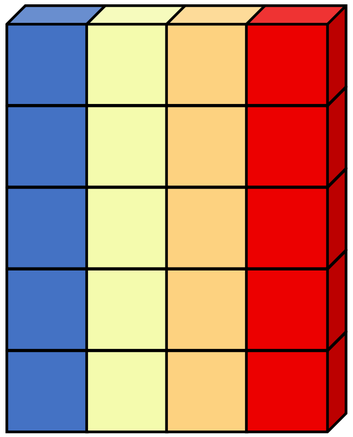


Encoder

$\mathbf{T}, \mathcal{X}$



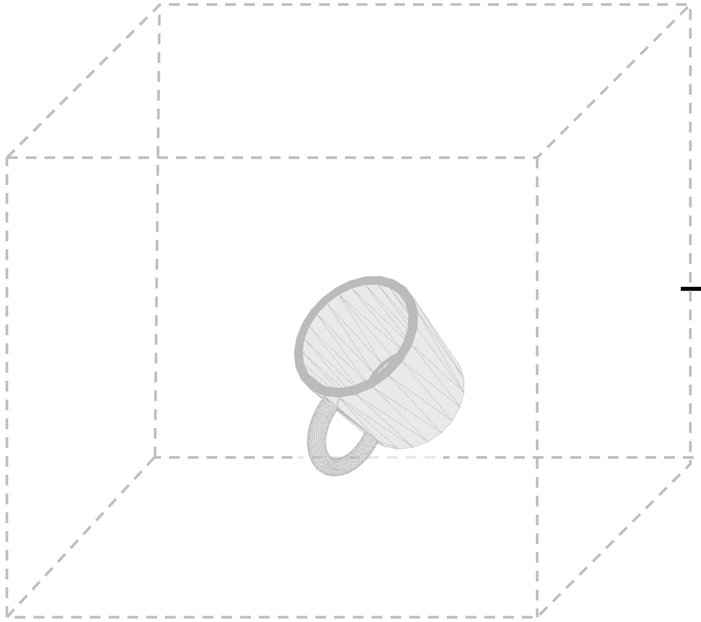
NDF MLP



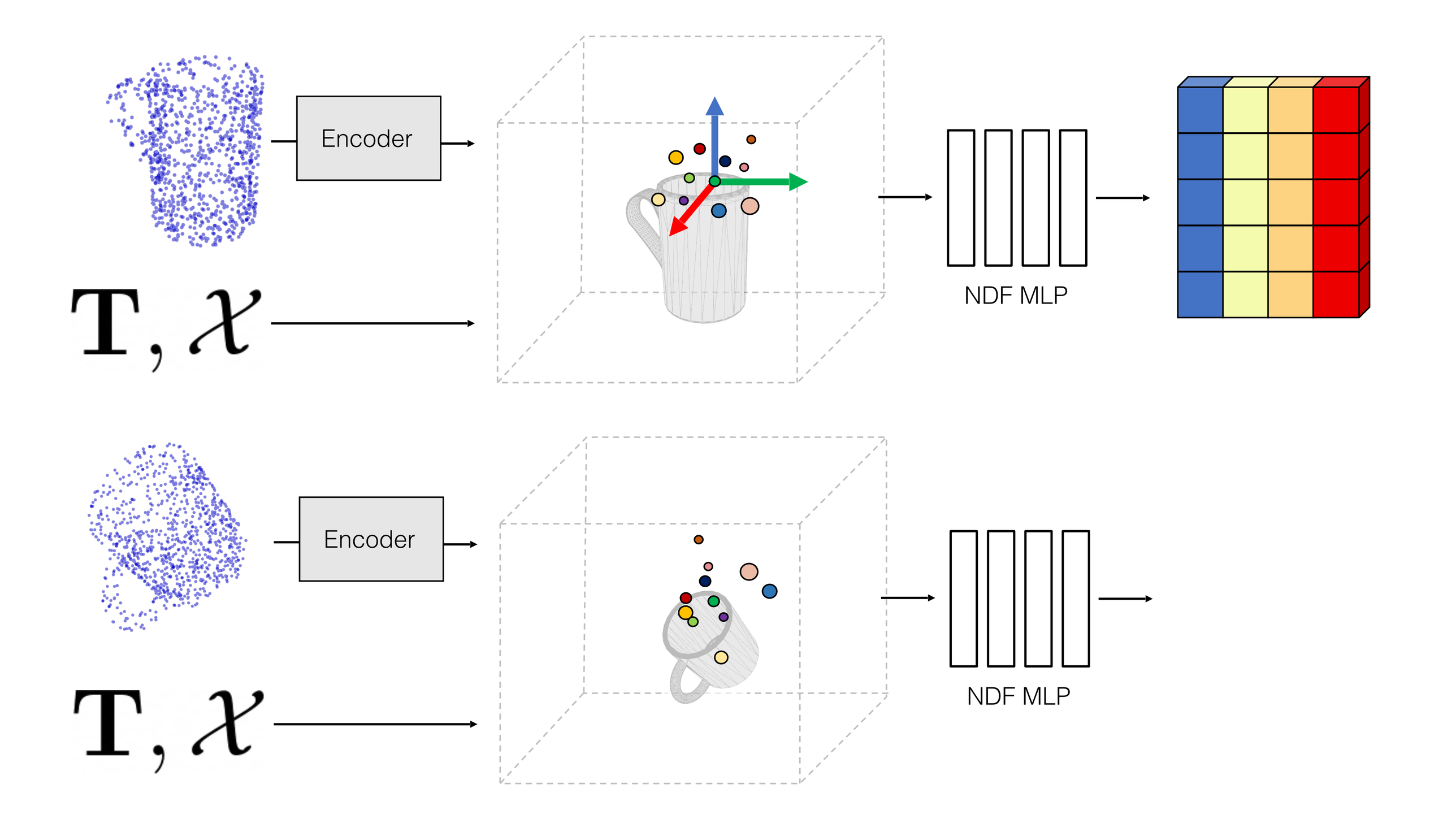


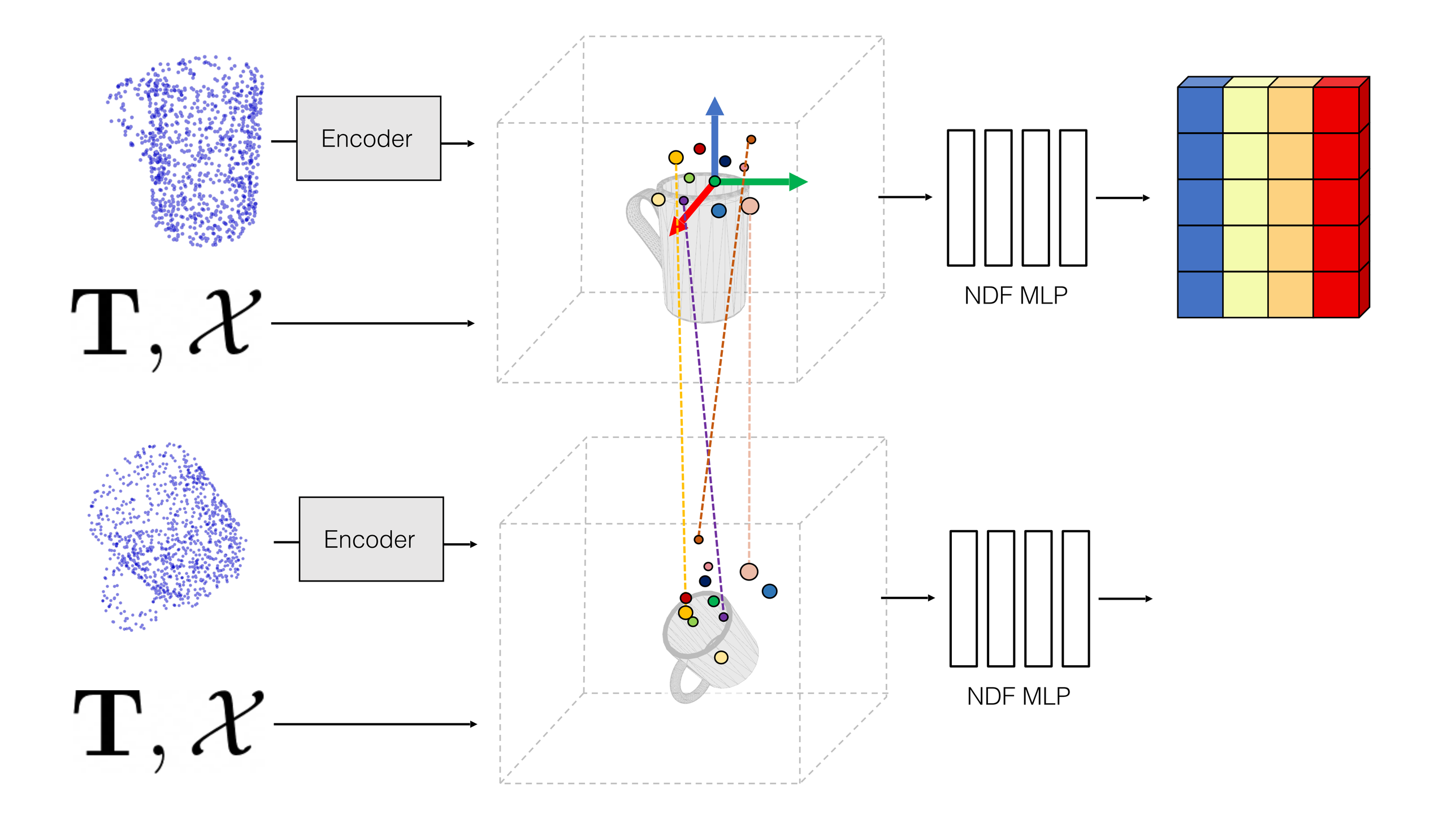
Encoder

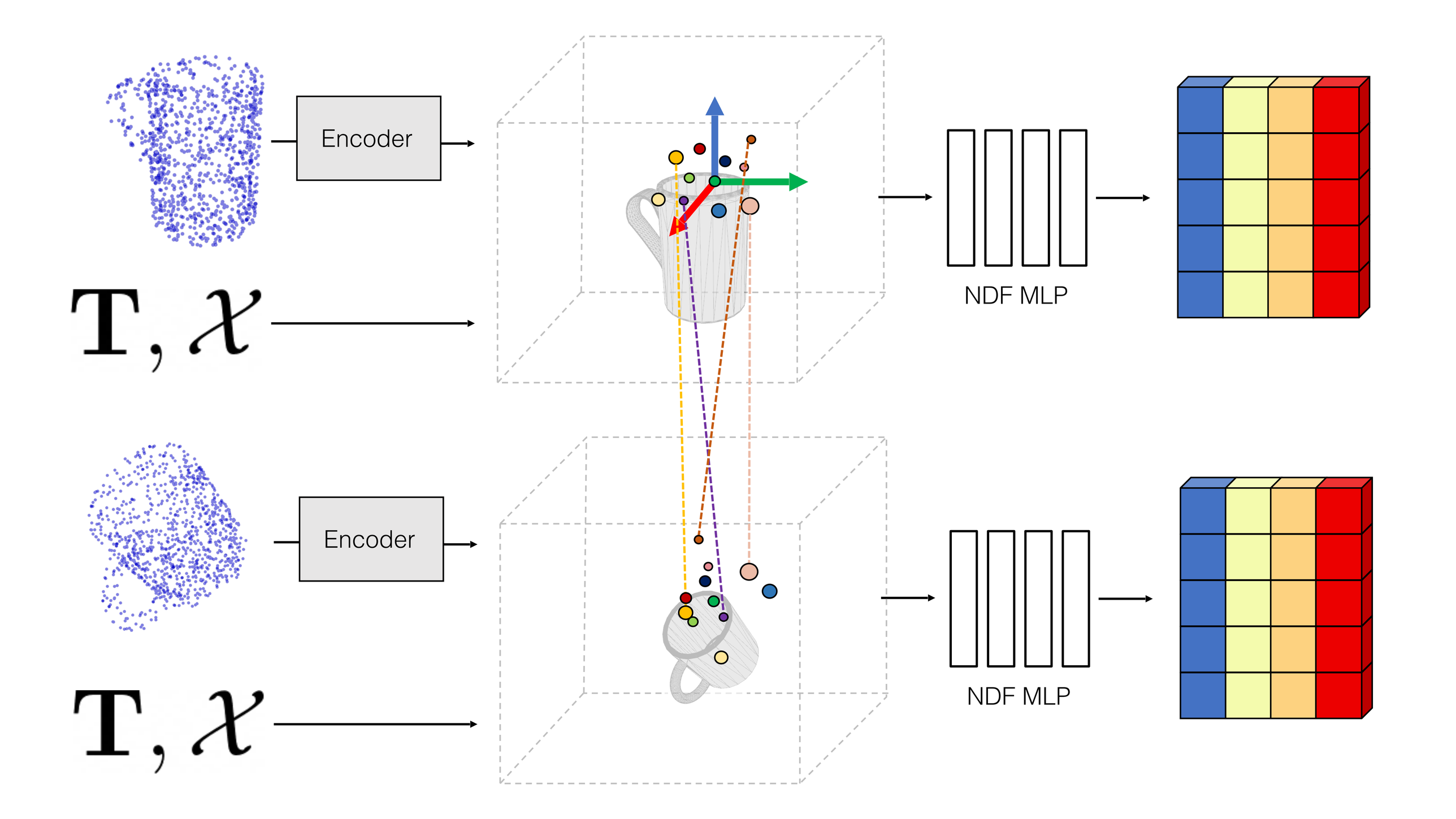
$\mathbf{T}, \mathcal{X}$

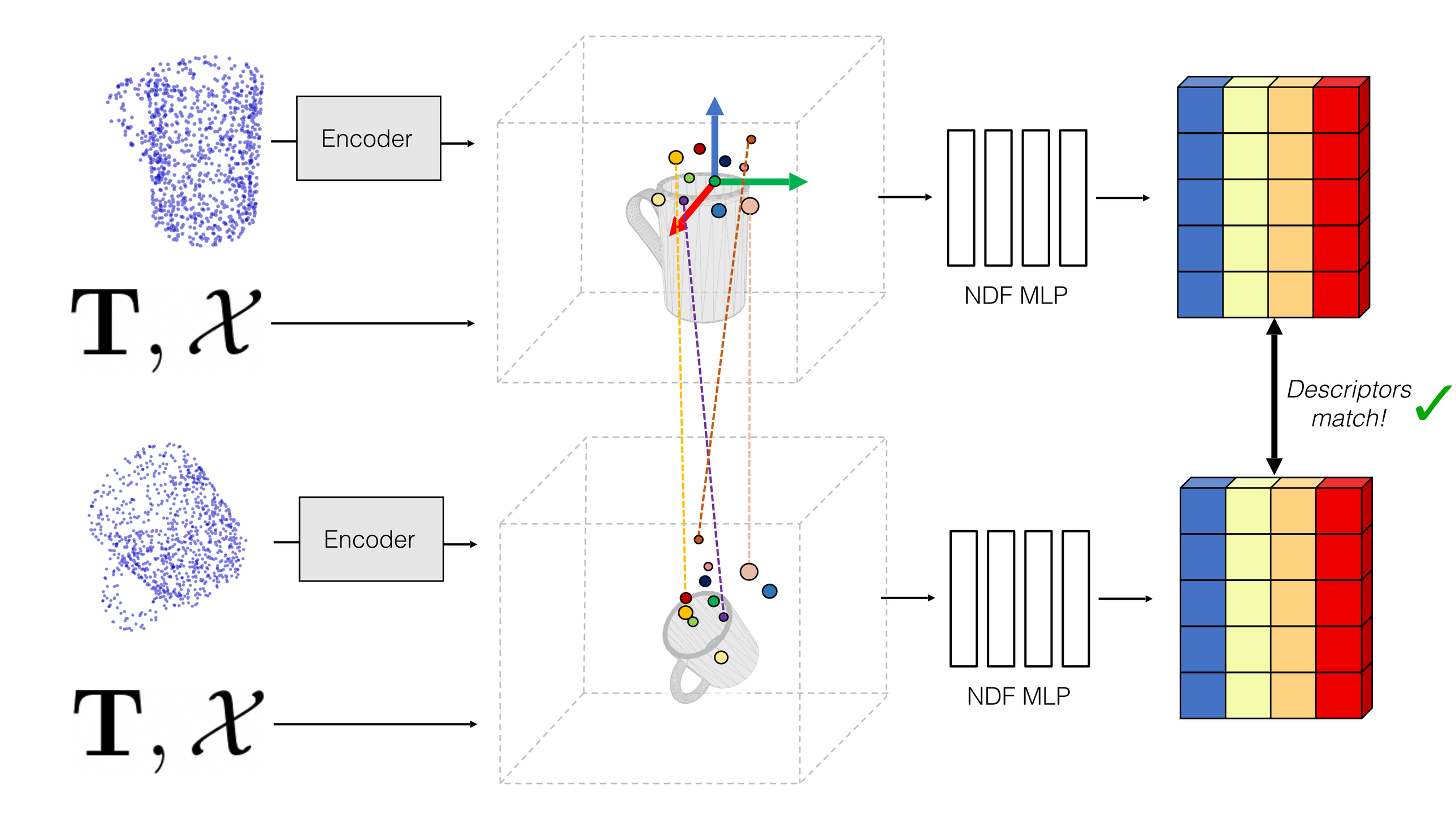


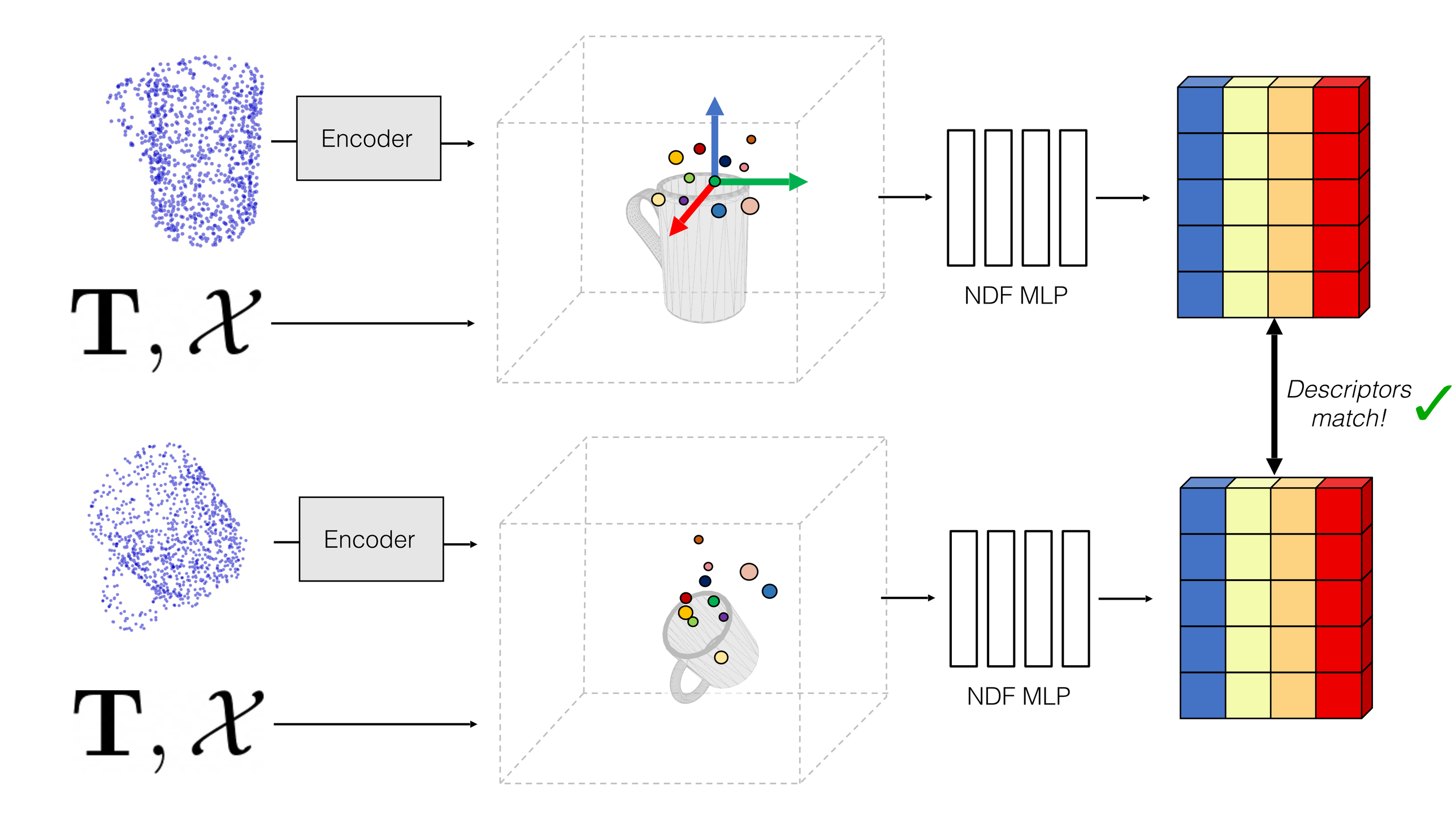
NDF MLP

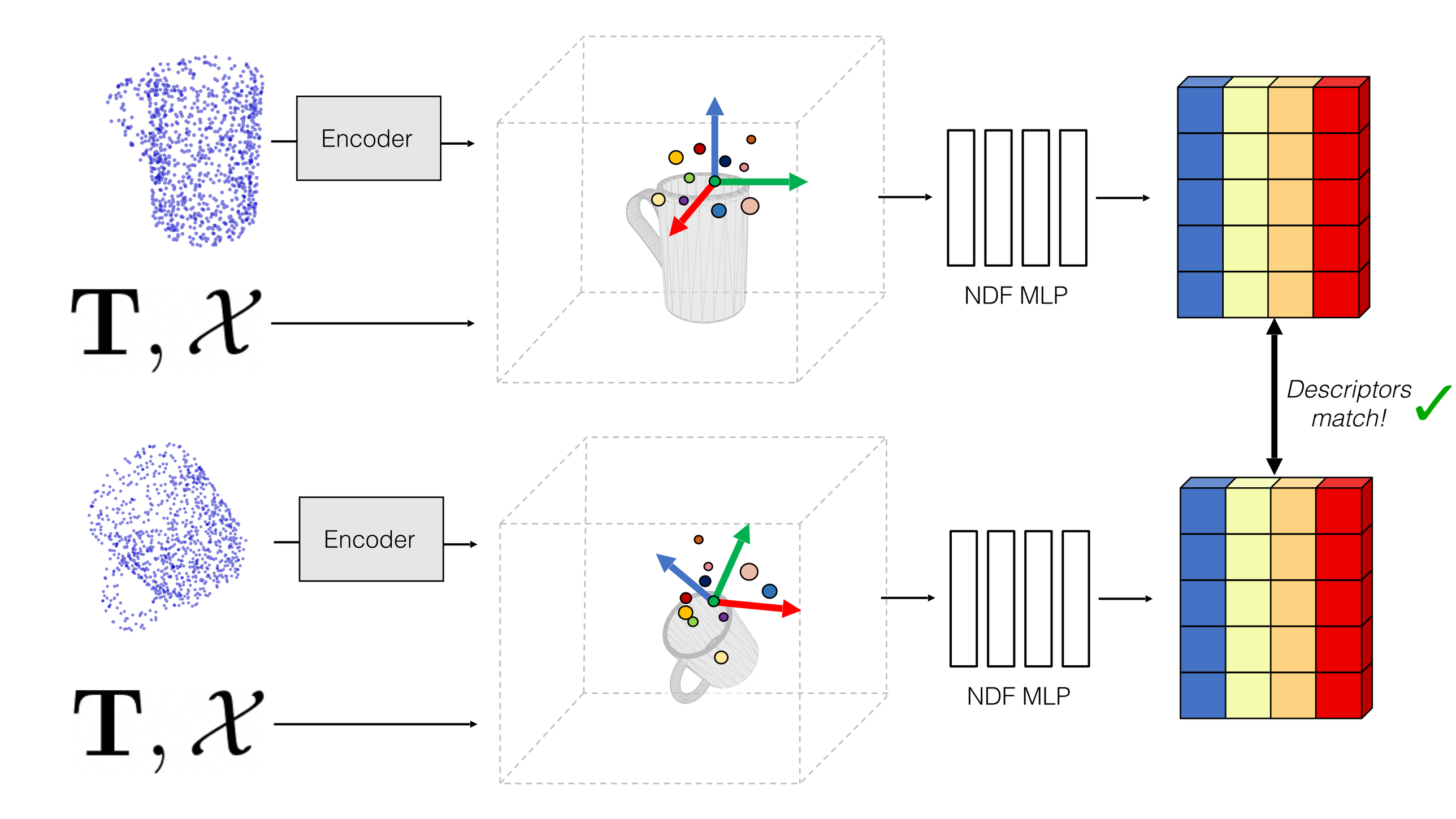

Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP


Encoder
$\mathbf{T}, \mathcal{X}$
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
NDF MLP

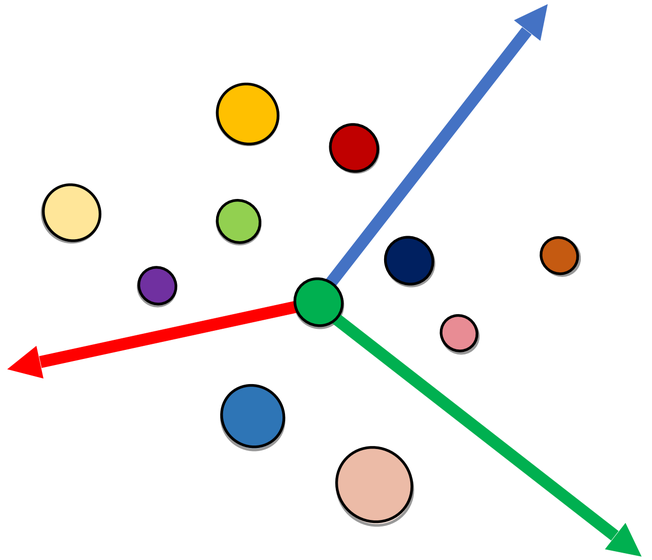

Encoder
$\mathbf{T}, \mathcal{X}$
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
NDF MLP

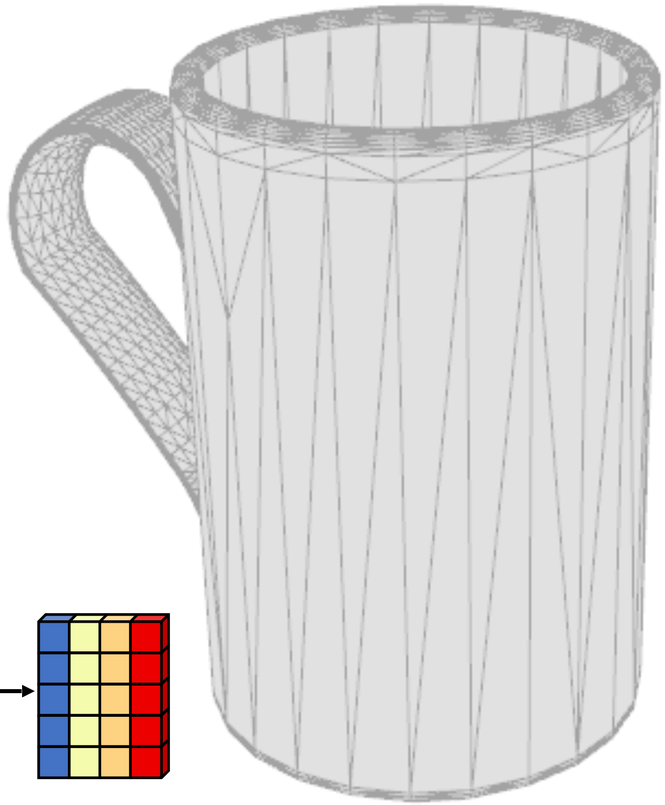

Encoder
$\mathbf{T}, \mathcal{X}$
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
NDF MLP
Descriptors match!

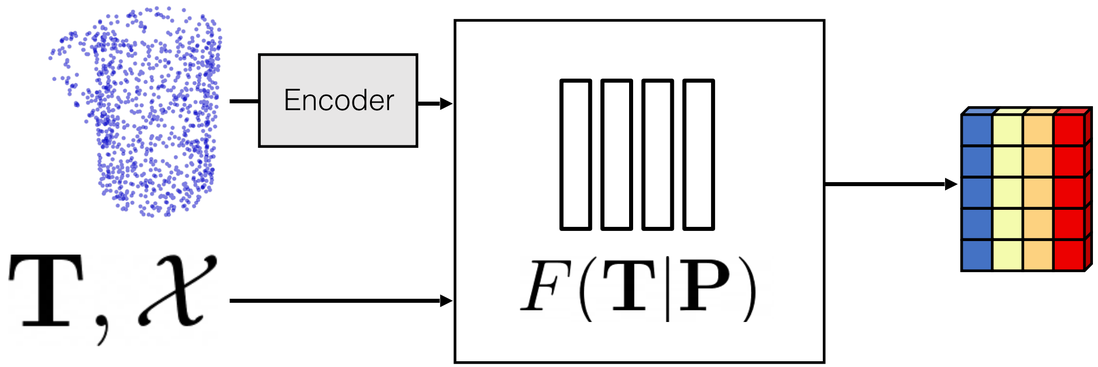

Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
Descriptors match!
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP


Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP
Descriptors match!
Encoder
$\mathbf{T}, \mathcal{X}$
NDF MLP

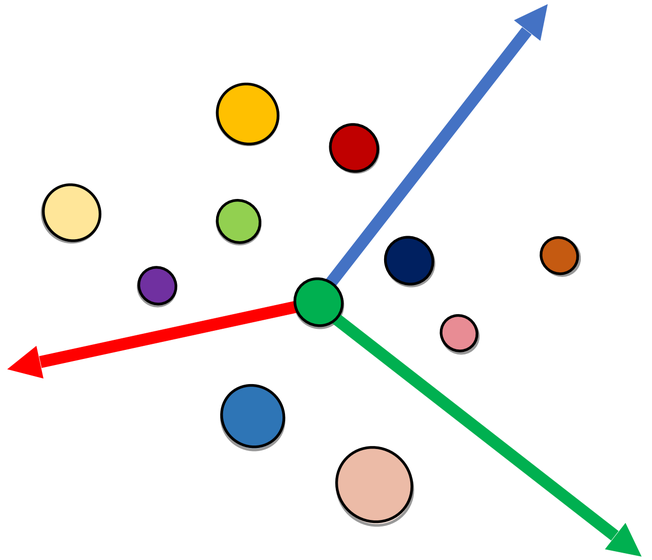
External object pose can parameterize a local coordinate frame

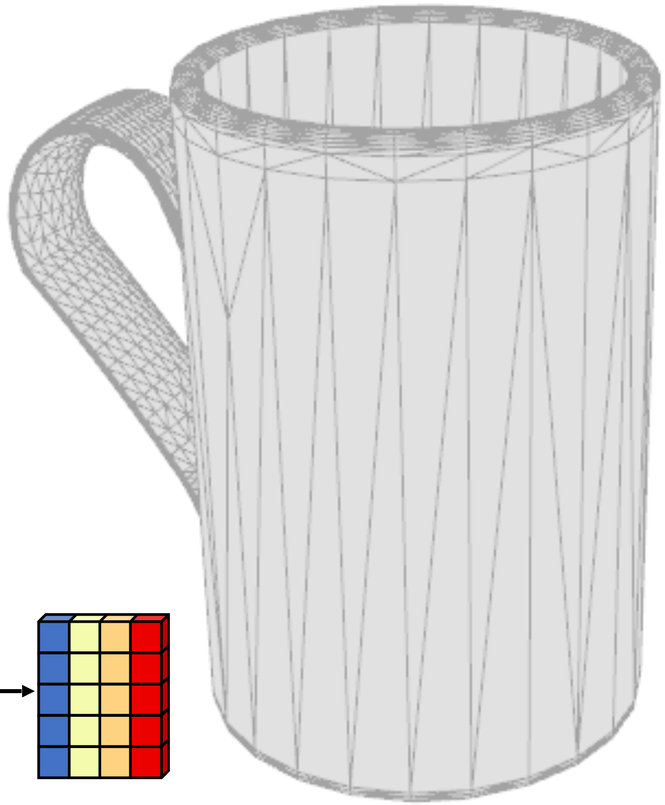


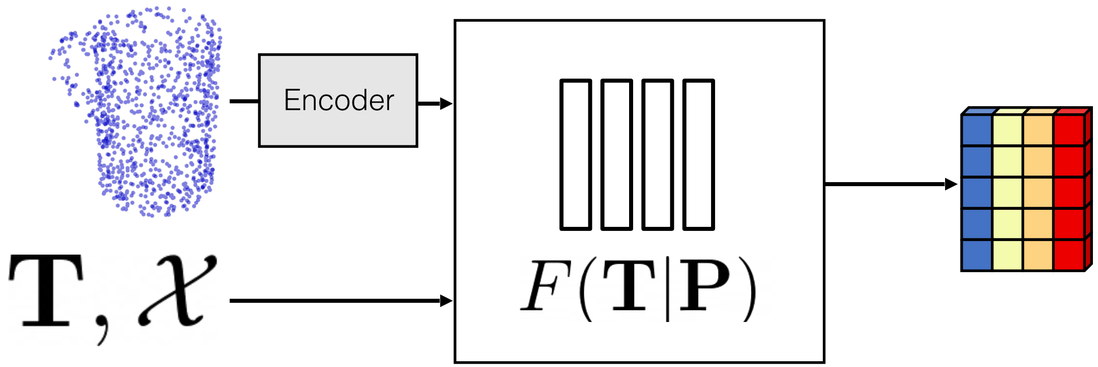




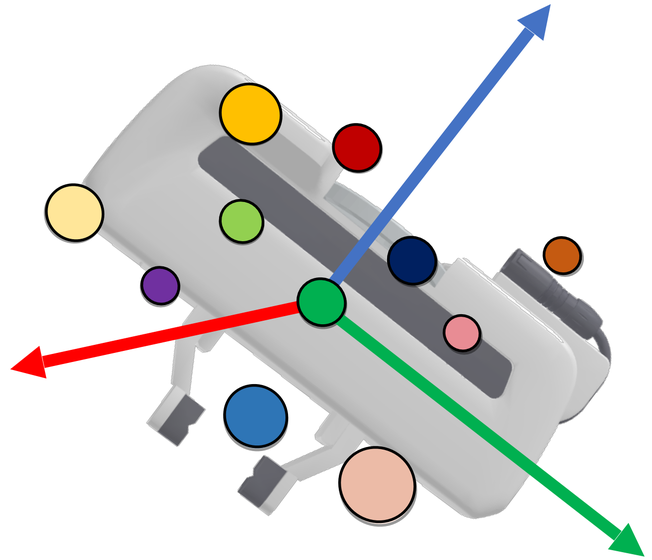
External object pose can parameterize a local coordinate frame

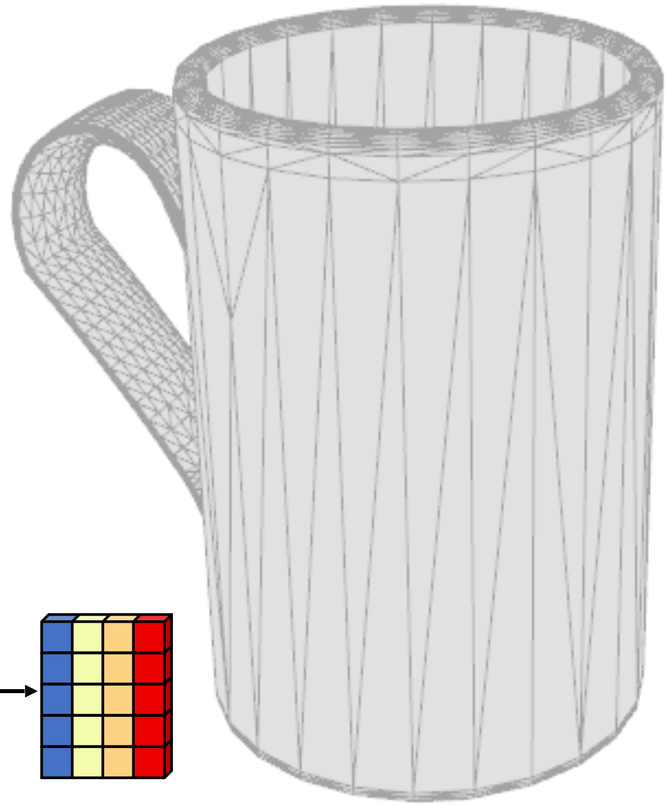


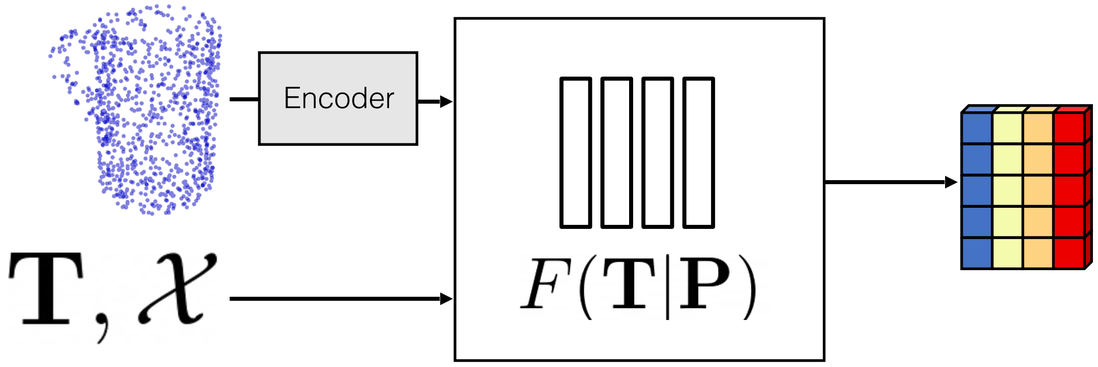




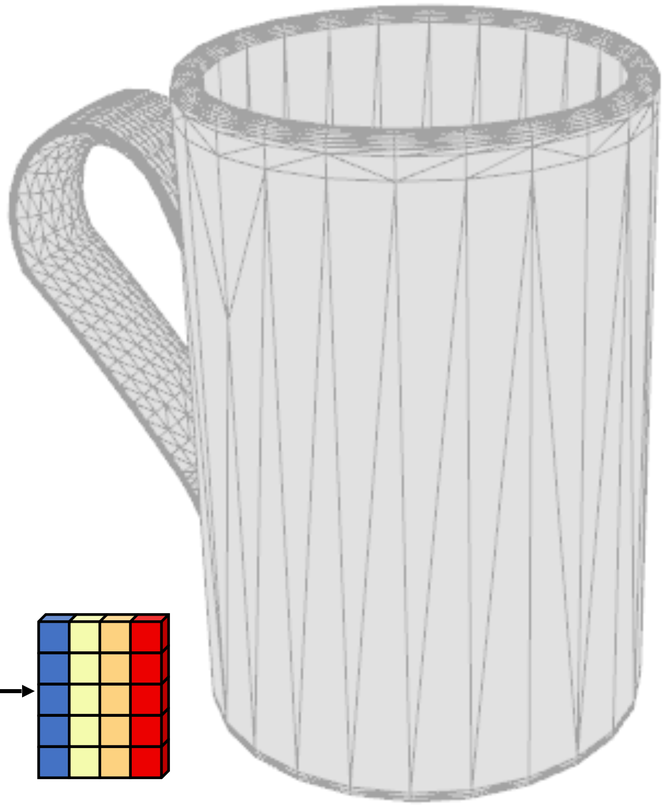
External object pose can parameterize a local coordinate frame

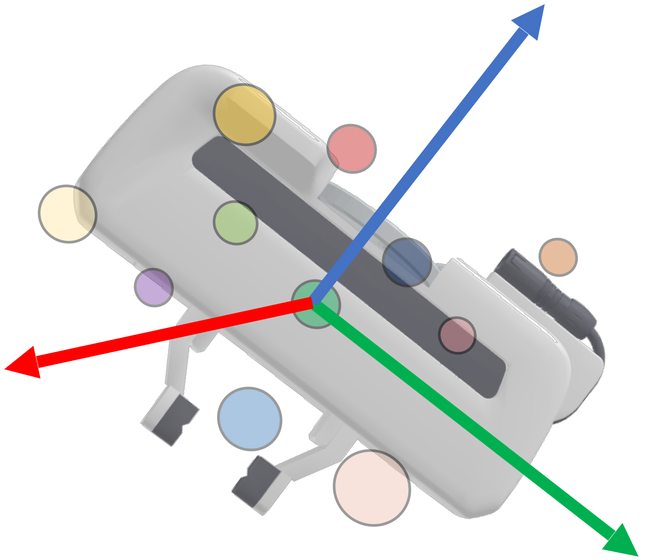


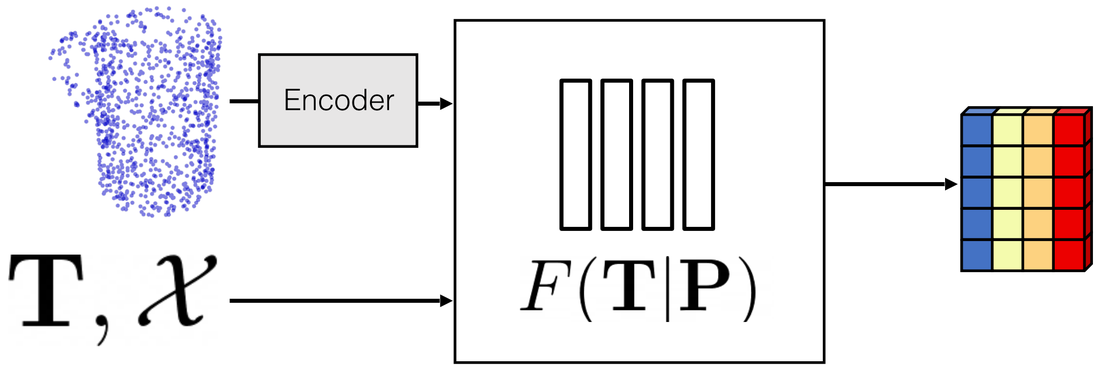




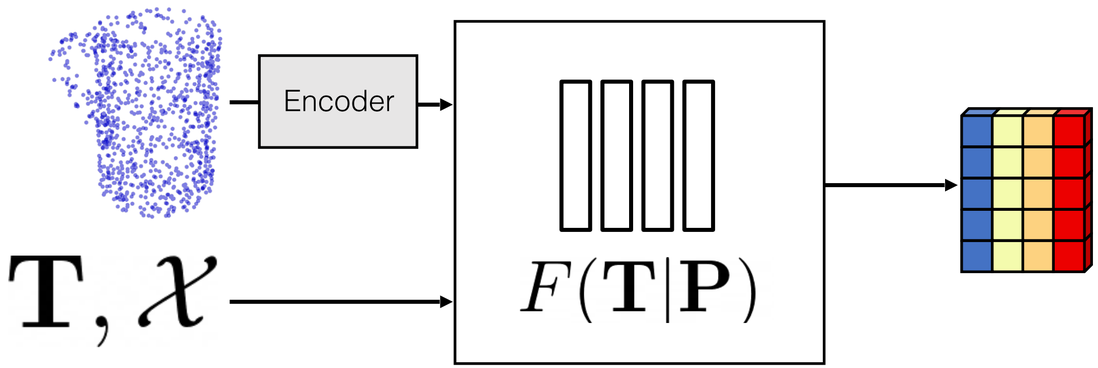
External object pose can parameterize a local coordinate frame

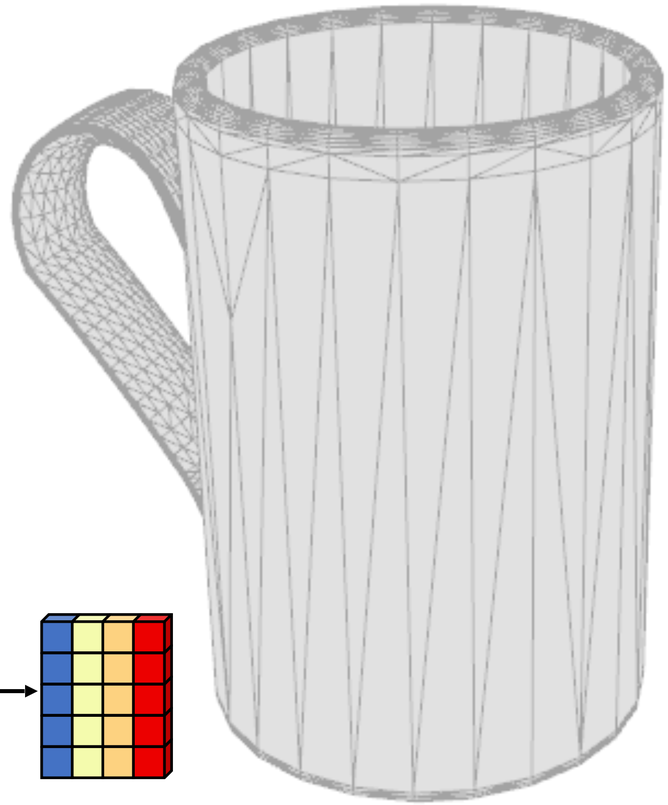


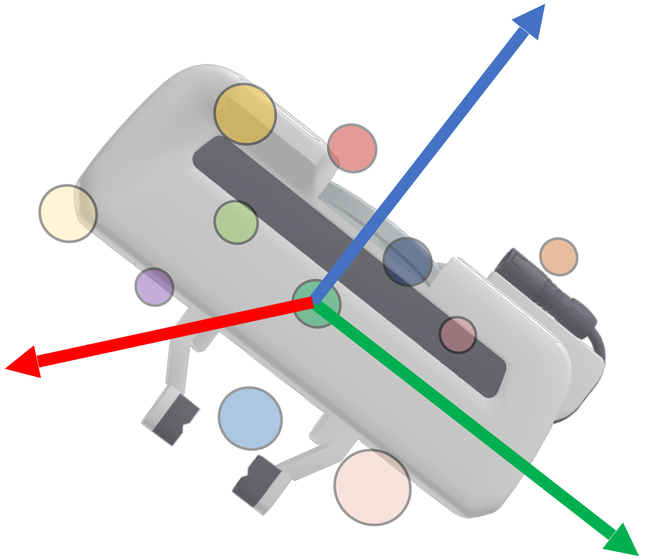




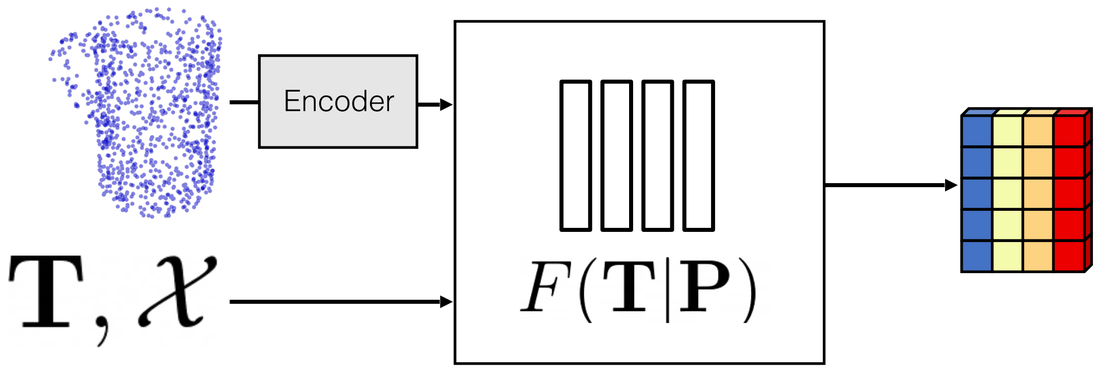
External object pose can parameterize a local coordinate frame

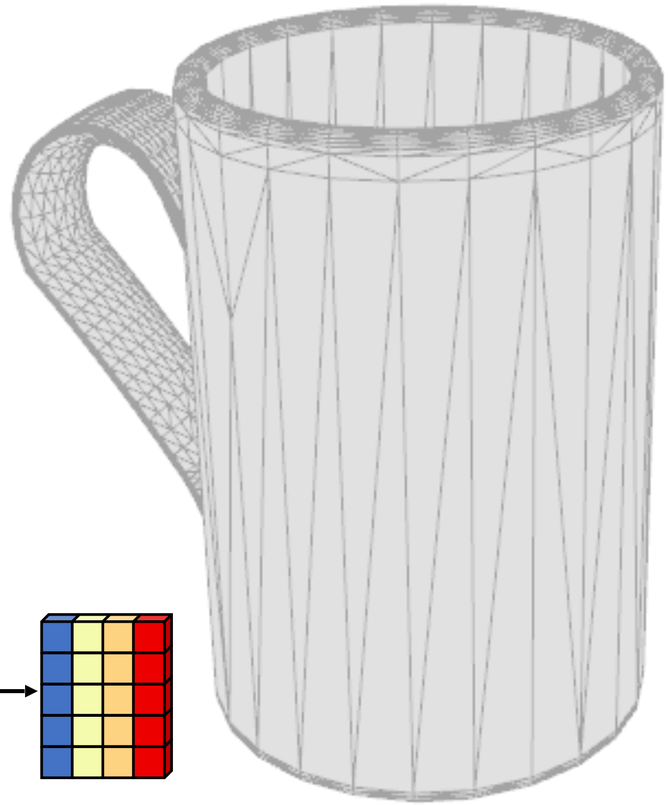


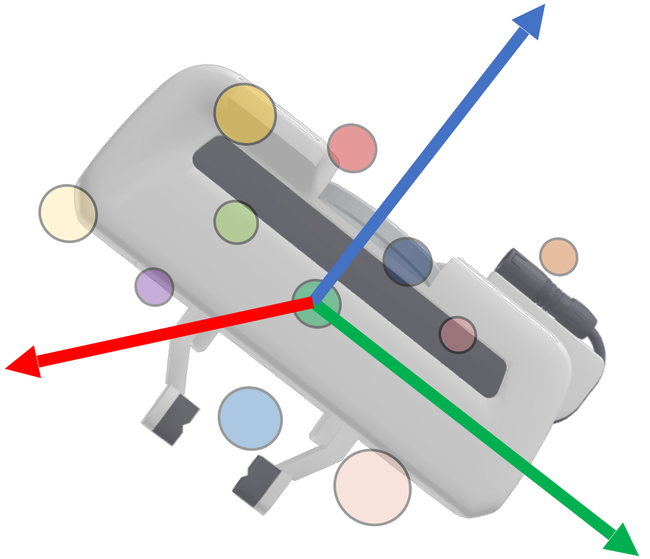




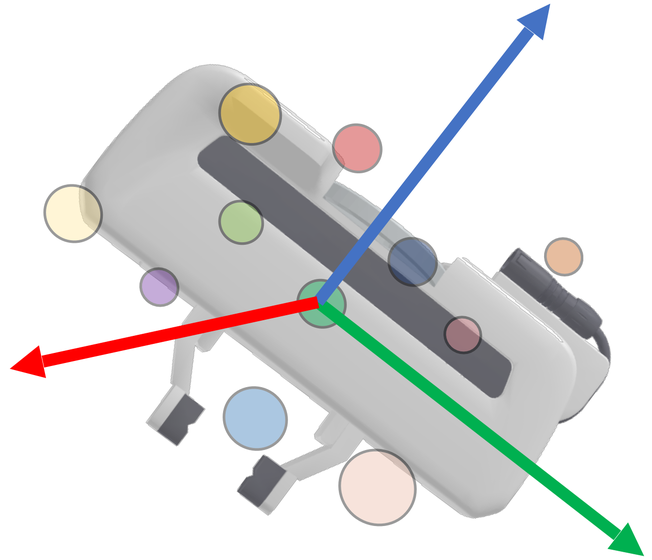
External object pose can parameterize a local coordinate frame

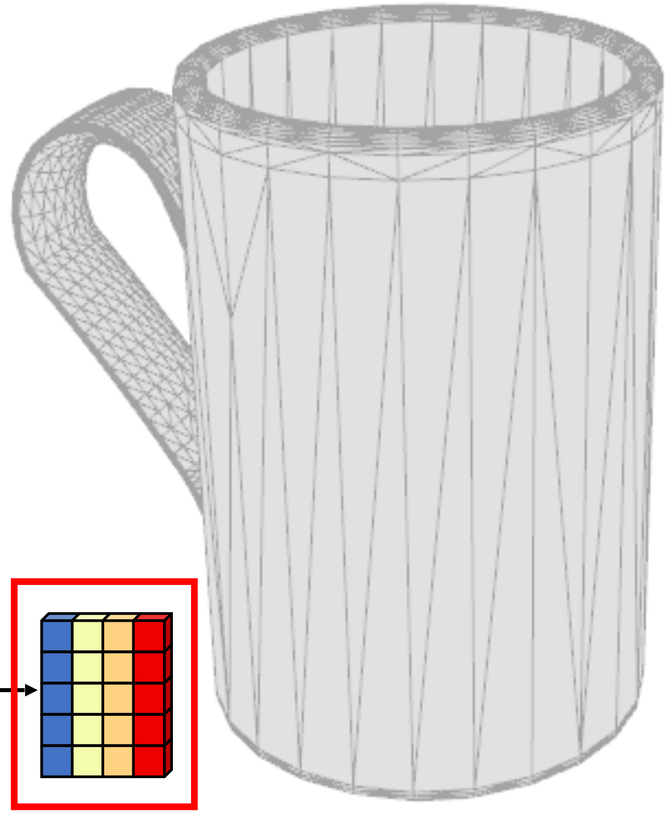


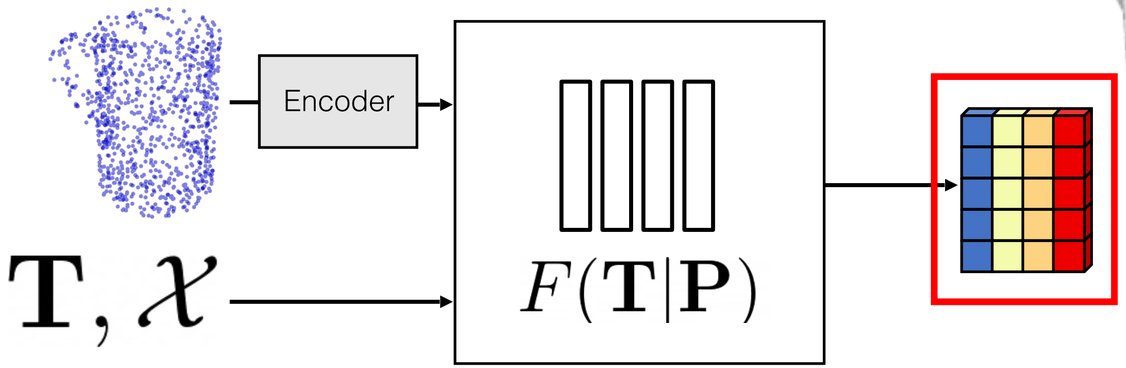




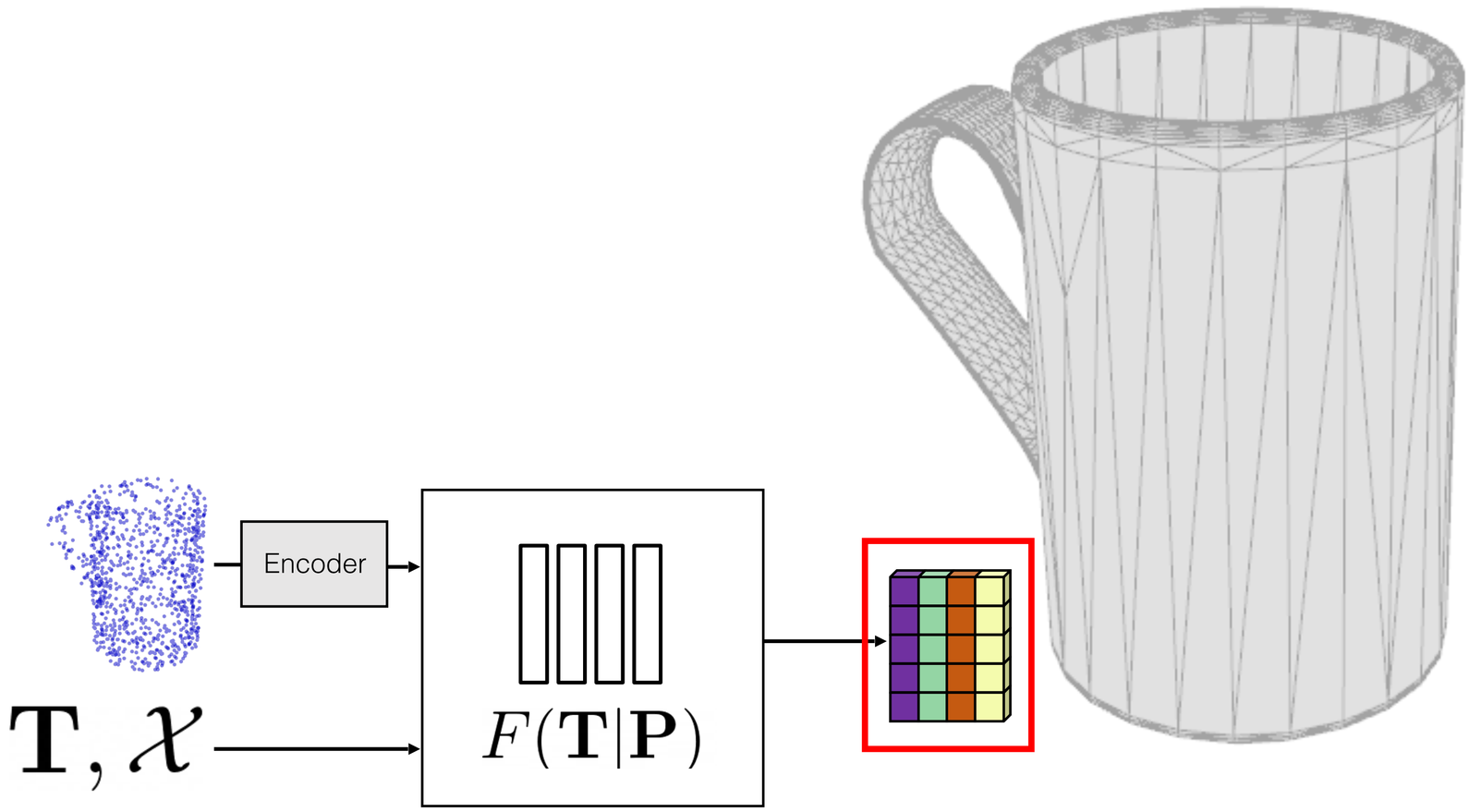
External object pose can parameterize a local coordinate frame

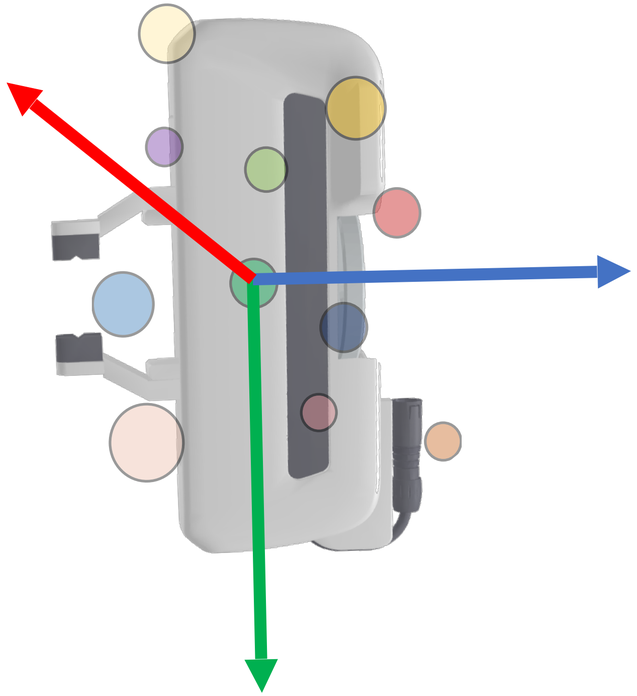




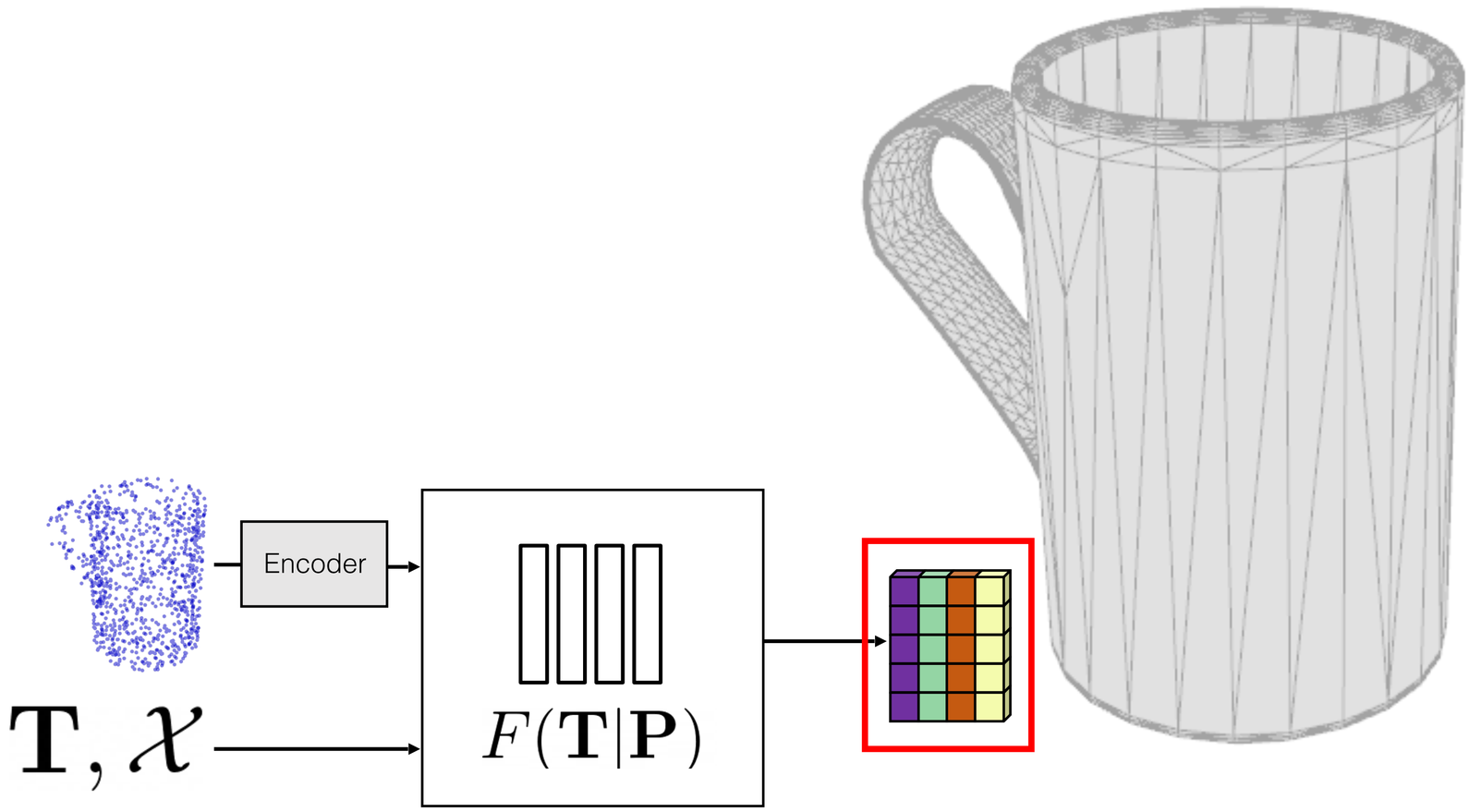


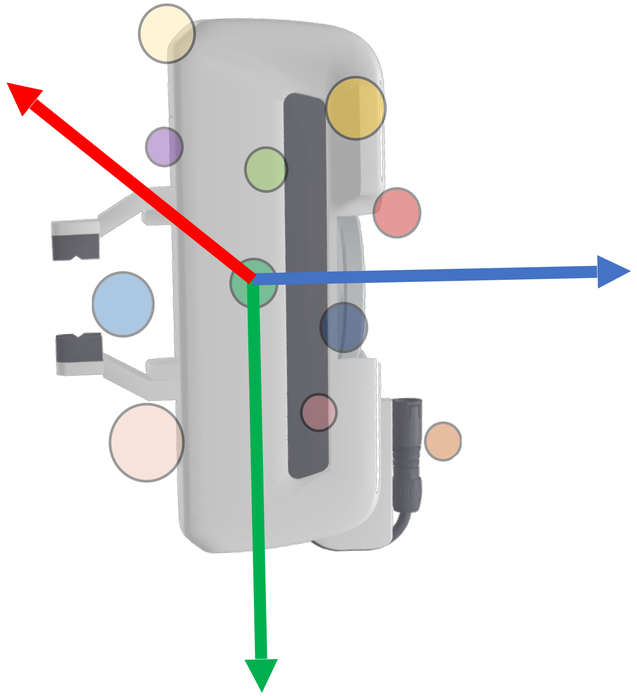
External object pose can parameterize a local coordinate frame



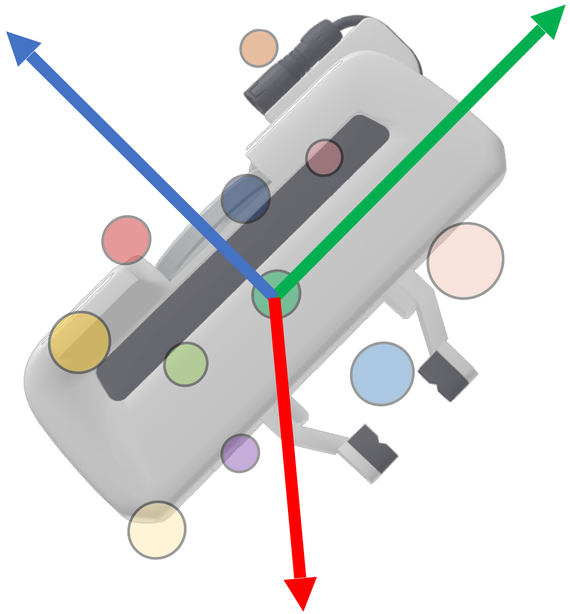


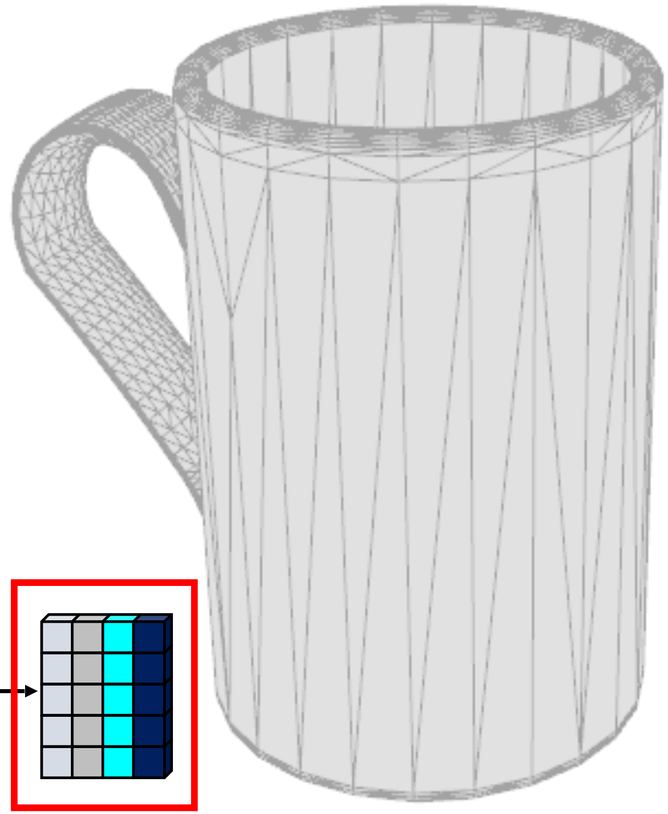
External object pose can parameterize a local coordinate frame





External object pose can parameterize a local coordinate frame





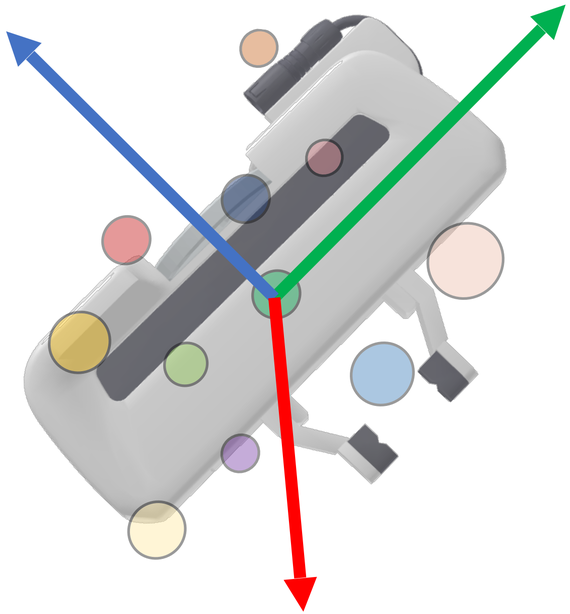


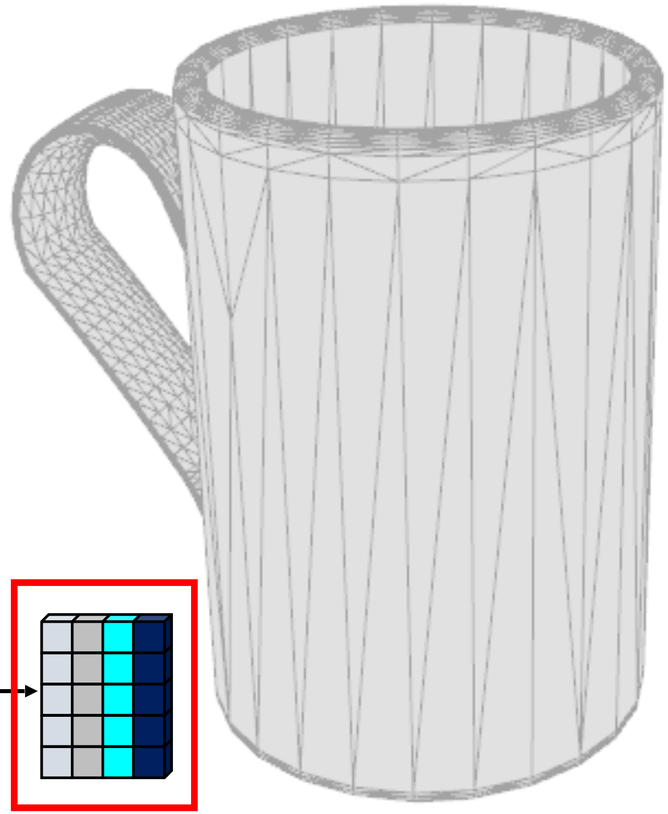
Encoder

$\mathbf{T}, \mathcal{X}$

$F(\mathbf{T}|\mathbf{P})$

External object pose can parameterize a local coordinate frame





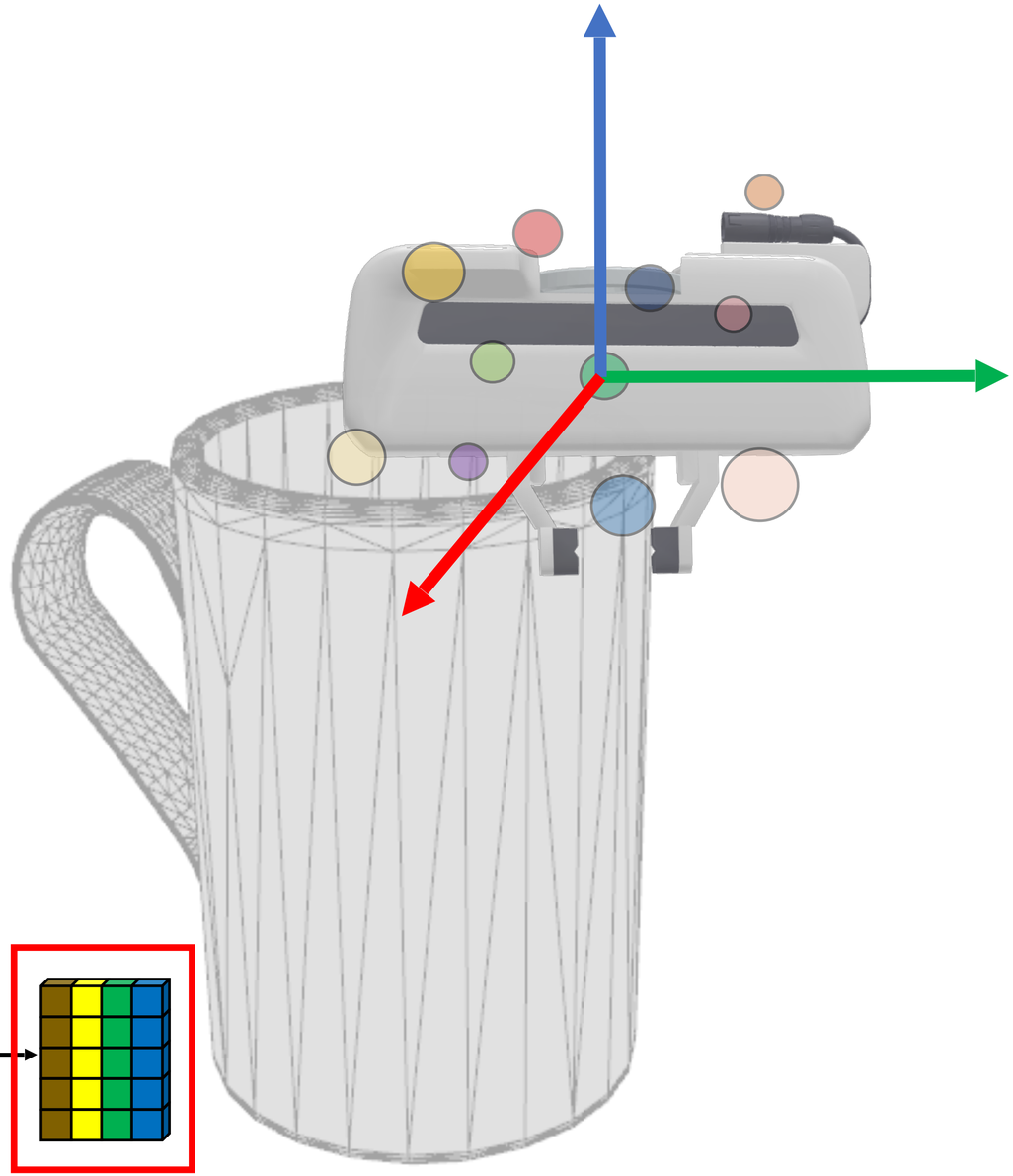


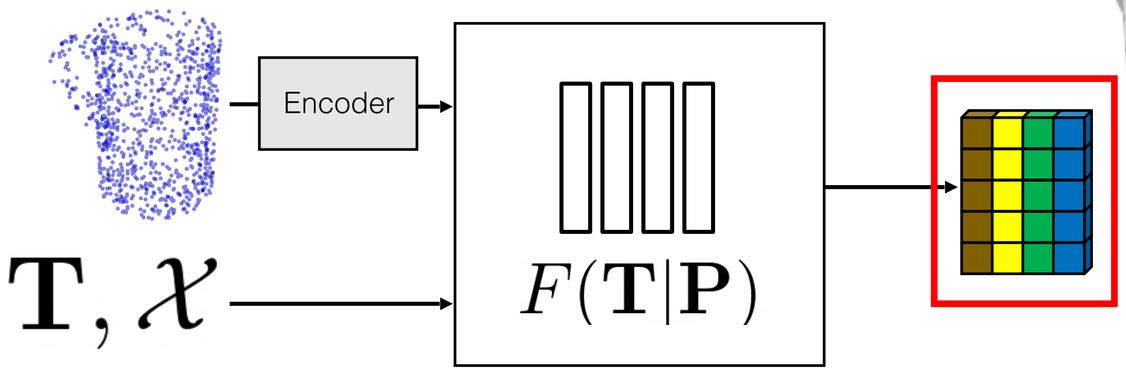
Encoder

$\mathbf{T}, \mathcal{X}$

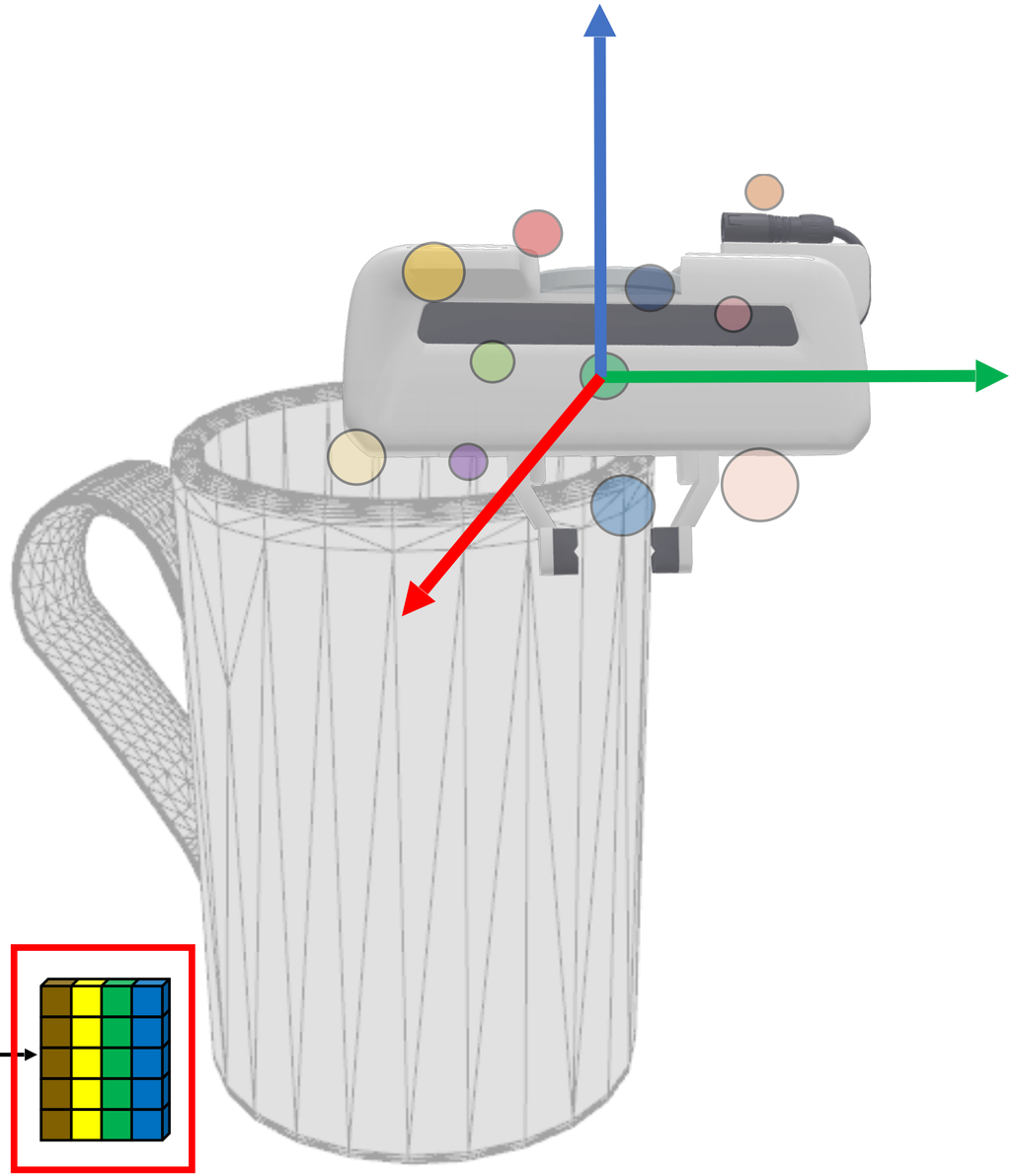
$F(\mathbf{T}|\mathbf{P})$

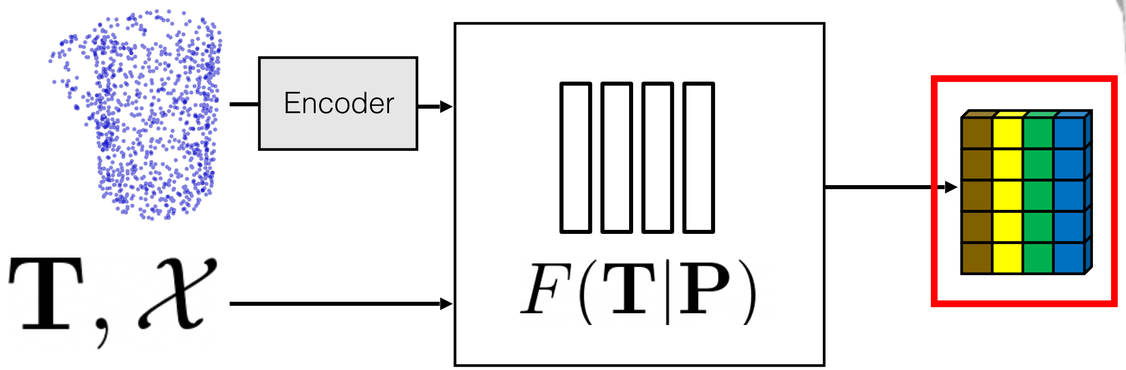
External object pose can parameterize a local coordinate frame





External object pose can parameterize a local coordinate frame



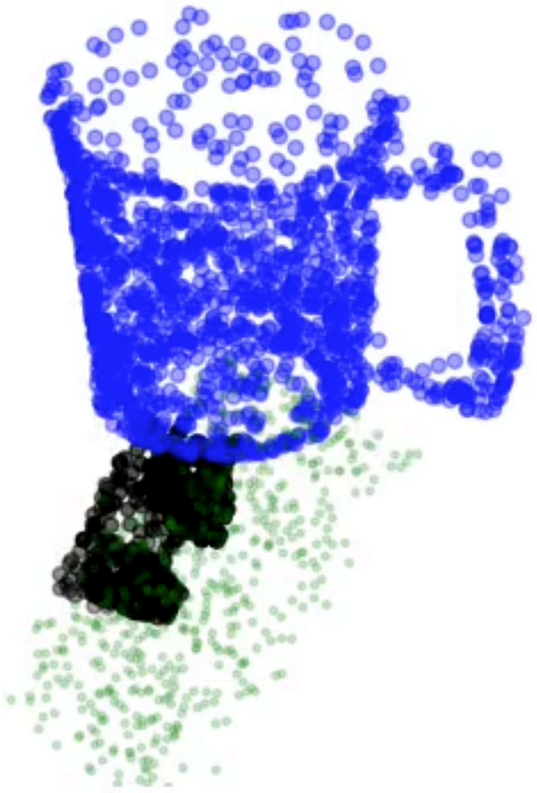


Local frame correspondence via SE(3) transform optimization

Demo object point cloud



$\hat{\mathbf{P}}$

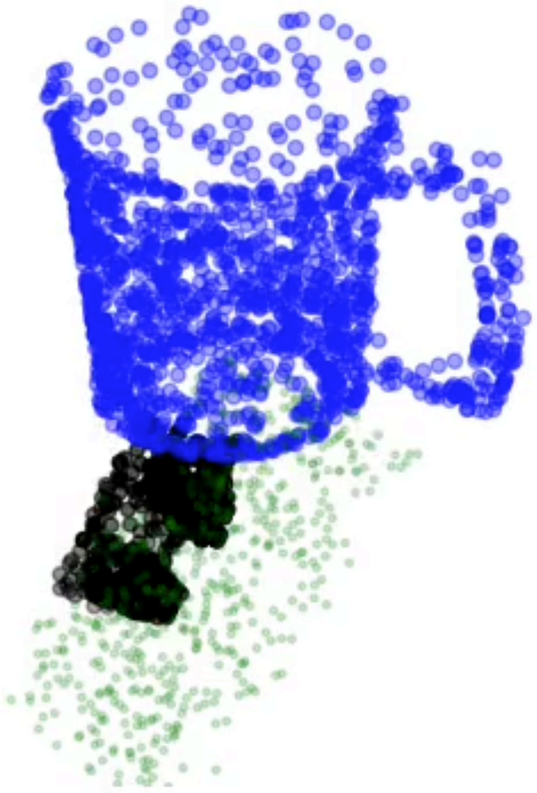


Local frame correspondence via SE(3) transform optimization

Demo object point cloud

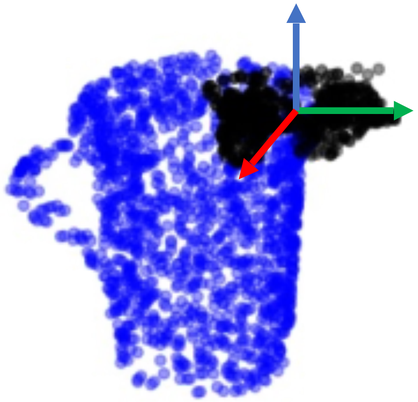


$\hat{\mathbf{P}}$



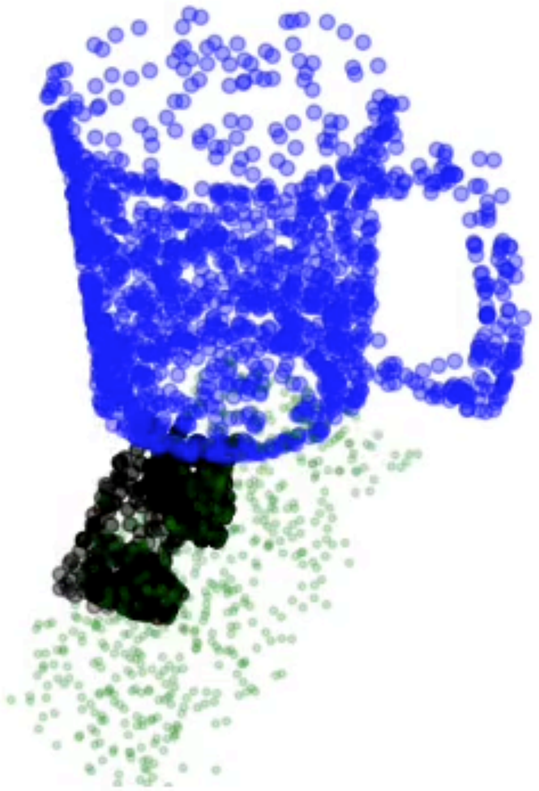
Local frame correspondence via SE(3) transform optimization

Demo object point cloud



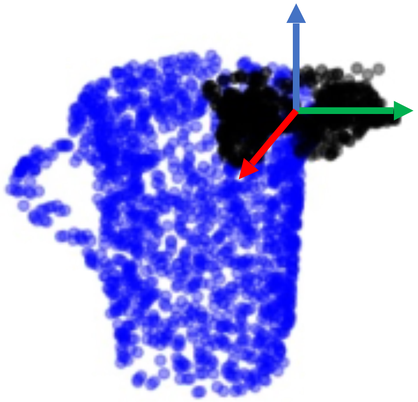
$\hat{\mathbf{P}}$

$\hat{\mathbf{T}}_{\text{grasp}}$



Local frame correspondence via SE(3) transform optimization

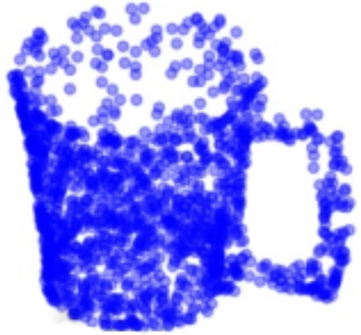
Demo object point cloud



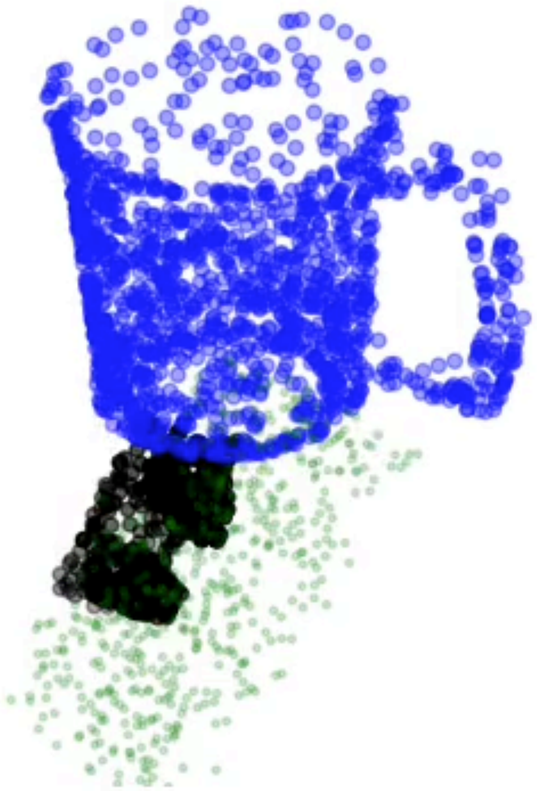
$\hat{\mathbf{P}}$

$\hat{\mathbf{T}}_{\text{grasp}}$

New point cloud

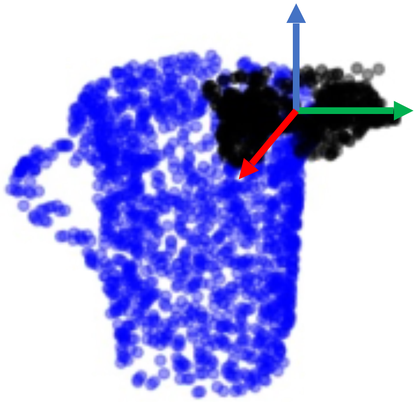


$\mathbf{P}$



Local frame correspondence via SE(3) transform optimization

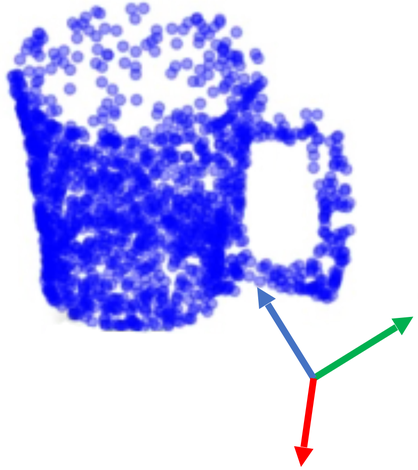
Demo object point cloud



$\hat{\mathbf{P}}$

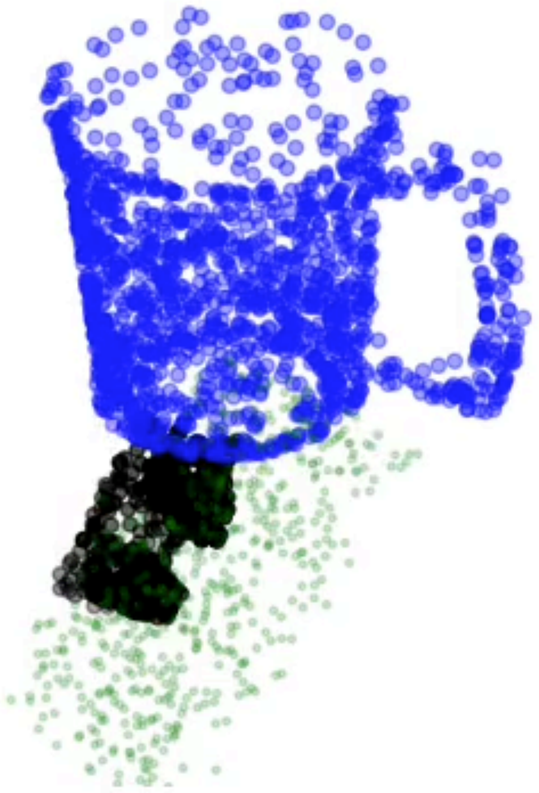
$\hat{\mathbf{T}}_{\text{grasp}}$

New point cloud



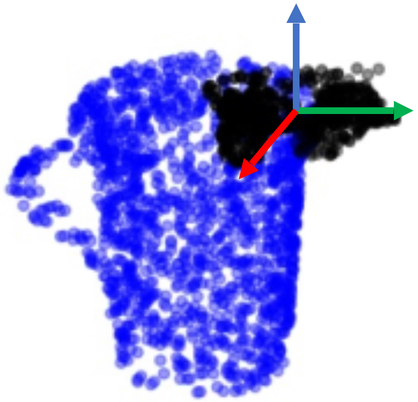
$\mathbf{P}$

$\mathbf{T}_{?}$



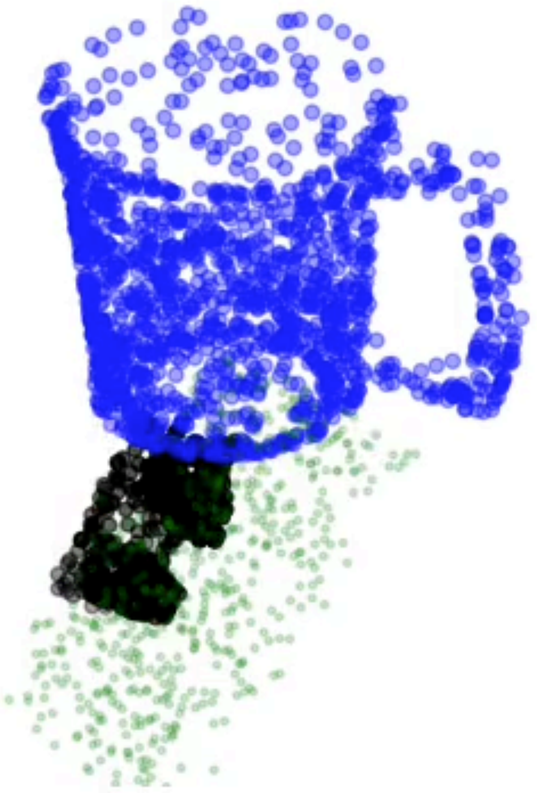
Local frame correspondence via SE(3) transform optimization

Demo object point cloud

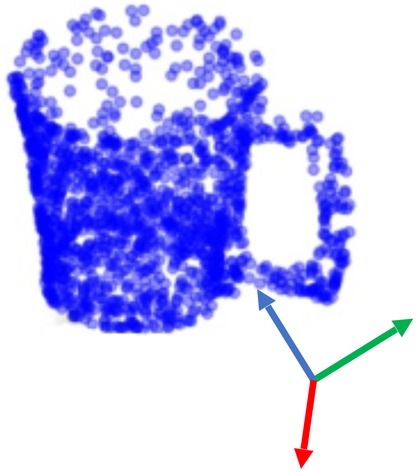


$\hat{\mathbf{P}}$

$\hat{\mathbf{T}}_{\text{grasp}}$



New point cloud



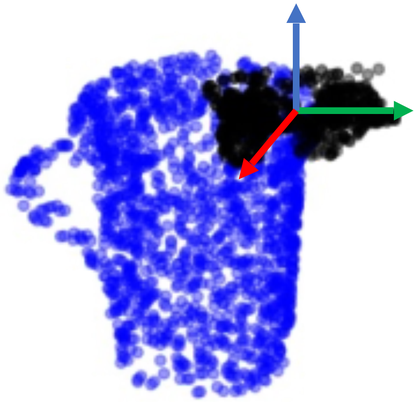
$\mathbf{P}$

$\mathbf{T}_?$

$$\bar{\mathbf{T}} = \underset{\mathbf{T}}{\text{argmin}} || F(\mathbf{T}|\mathbf{P}) - F(\hat{\mathbf{T}}_*|\hat{\mathbf{P}}_*) ||$$

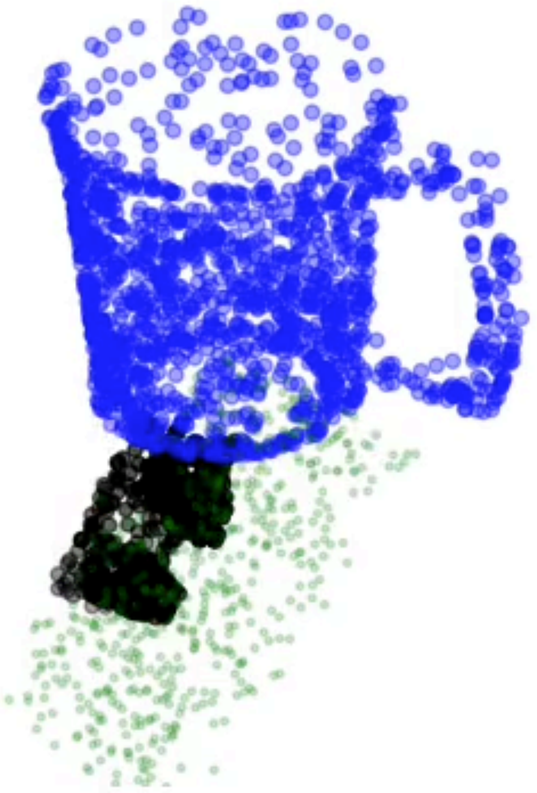
Local frame correspondence via SE(3) transform optimization

Demo object point cloud

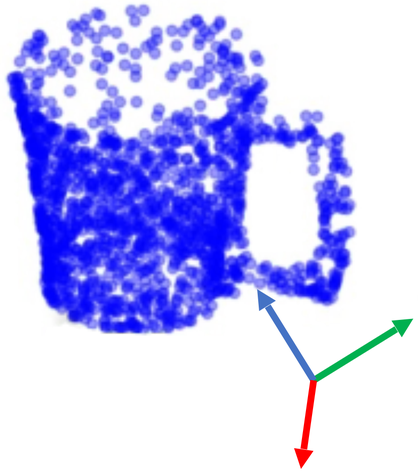


$\hat{\mathbf{P}}$

$\hat{\mathbf{T}}_{\text{grasp}}$



New point cloud



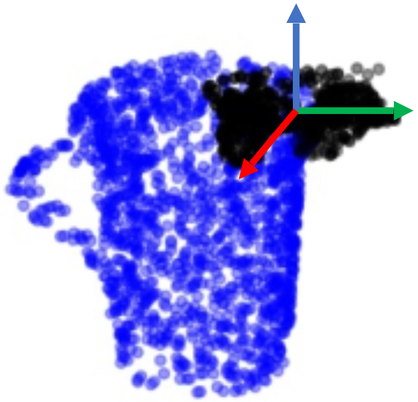
$\mathbf{P}$

$\mathbf{T}_?$

$$\bar{\mathbf{T}} = \underset{\mathbf{T}}{\text{argmin}} || F(\mathbf{T}|\mathbf{P}) - F(\hat{\mathbf{T}}_*|\hat{\mathbf{P}}_*) ||$$

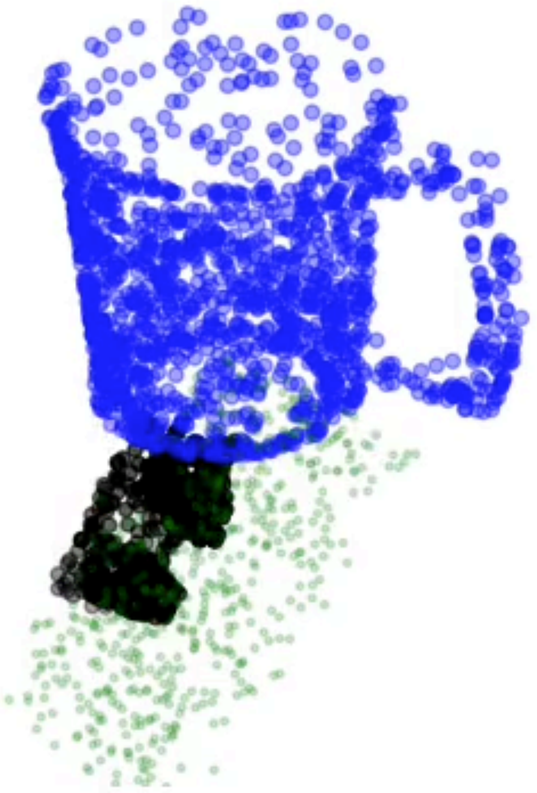
Local frame correspondence via SE(3) transform optimization

Demo object point cloud

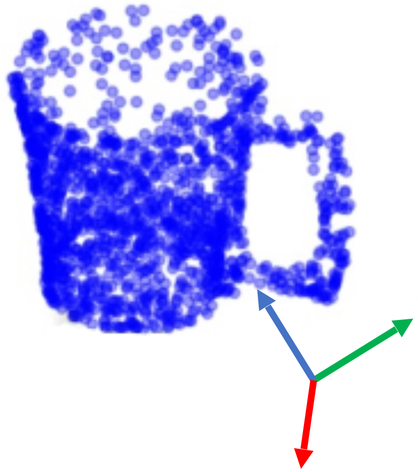


$\hat{\mathbf{P}}$

$\hat{\mathbf{T}}_{\text{grasp}}$



New point cloud

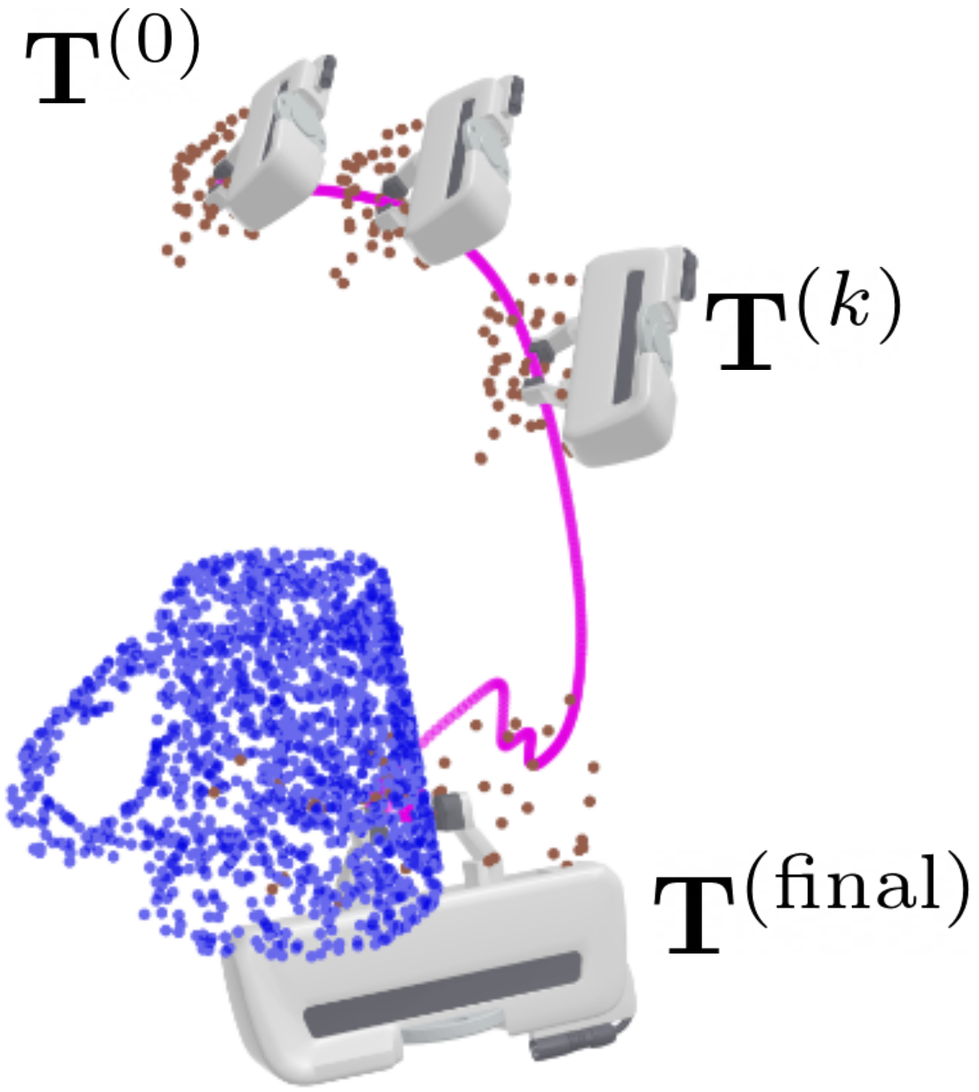


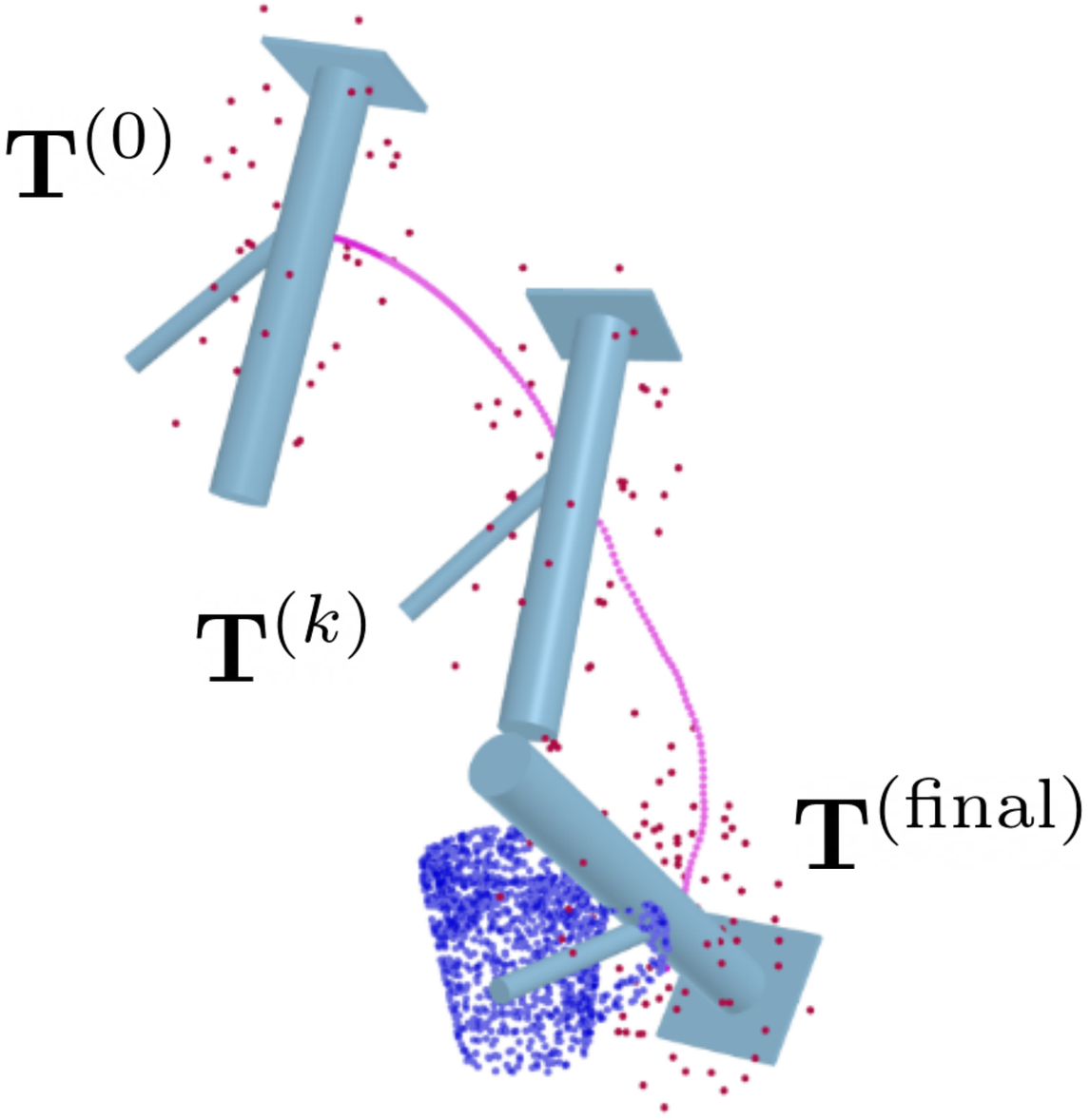
$\mathbf{P}$

$\mathbf{T}_?$

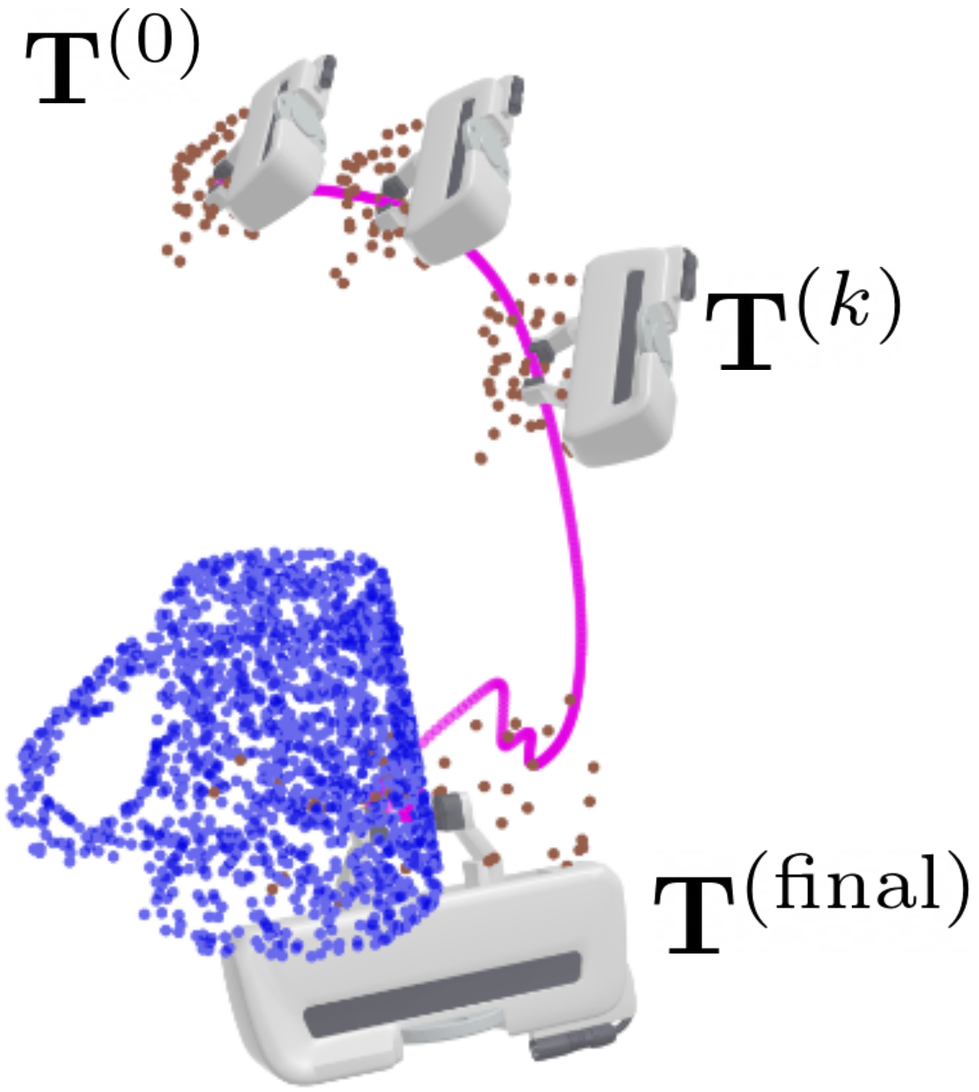
$$\bar{\mathbf{T}} = \underset{\mathbf{T}}{\text{argmin}} || F(\mathbf{T}|\mathbf{P}) - F(\hat{\mathbf{T}}_*|\hat{\mathbf{P}}_*) ||$$

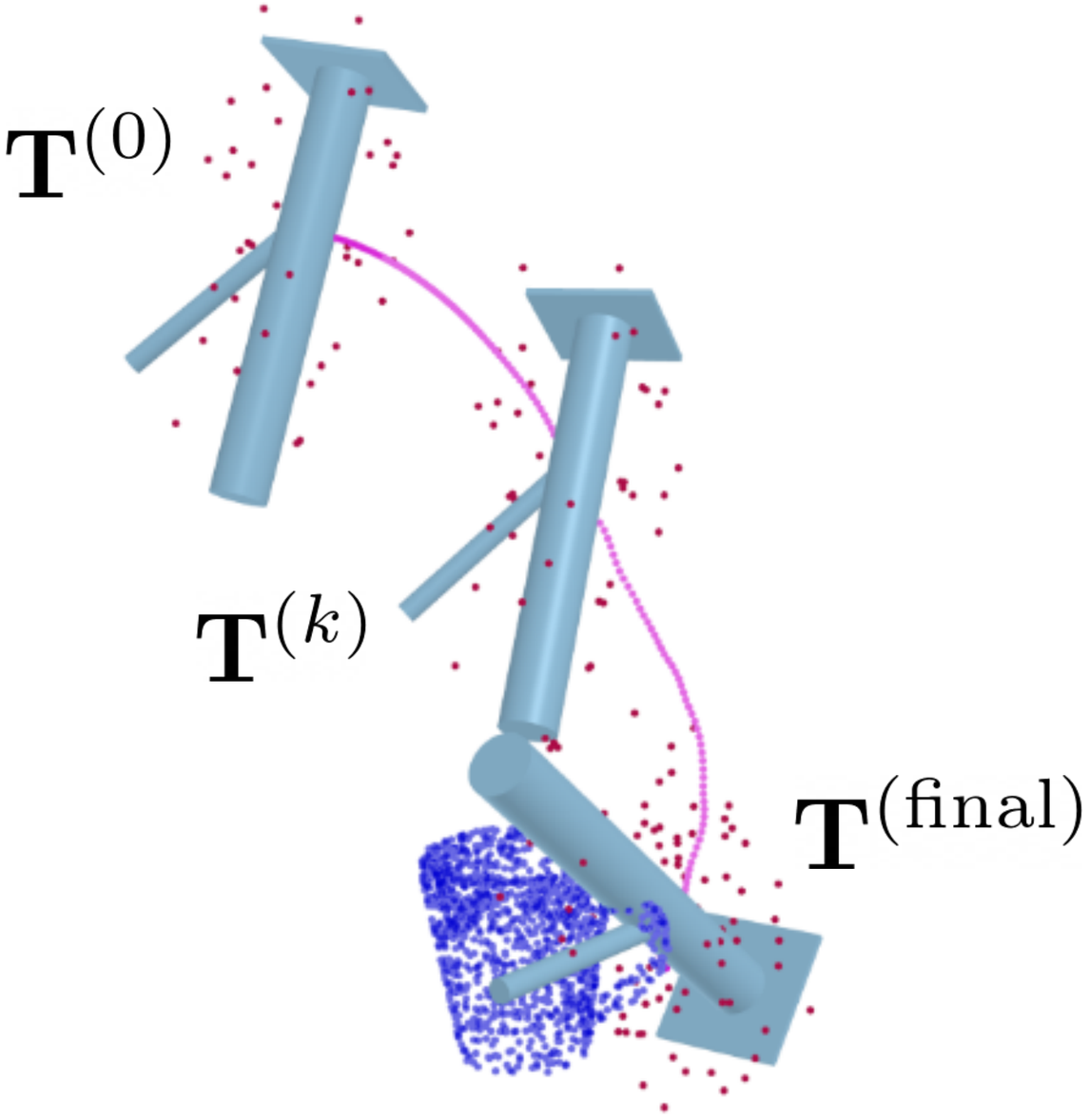
Repeat identical transform optimization for *grasping* and *placing*

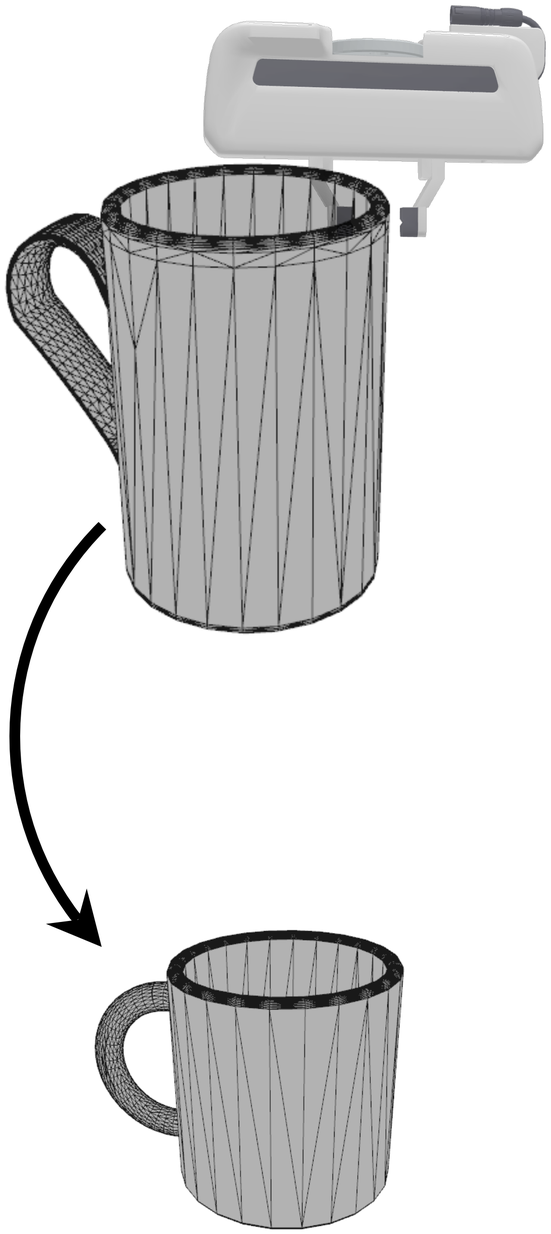


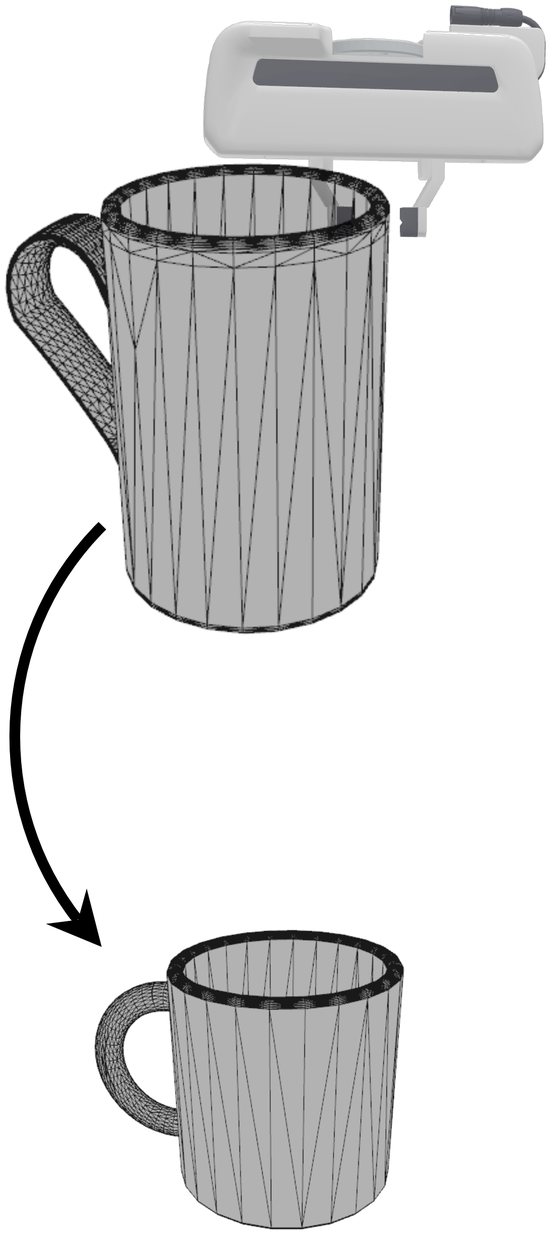


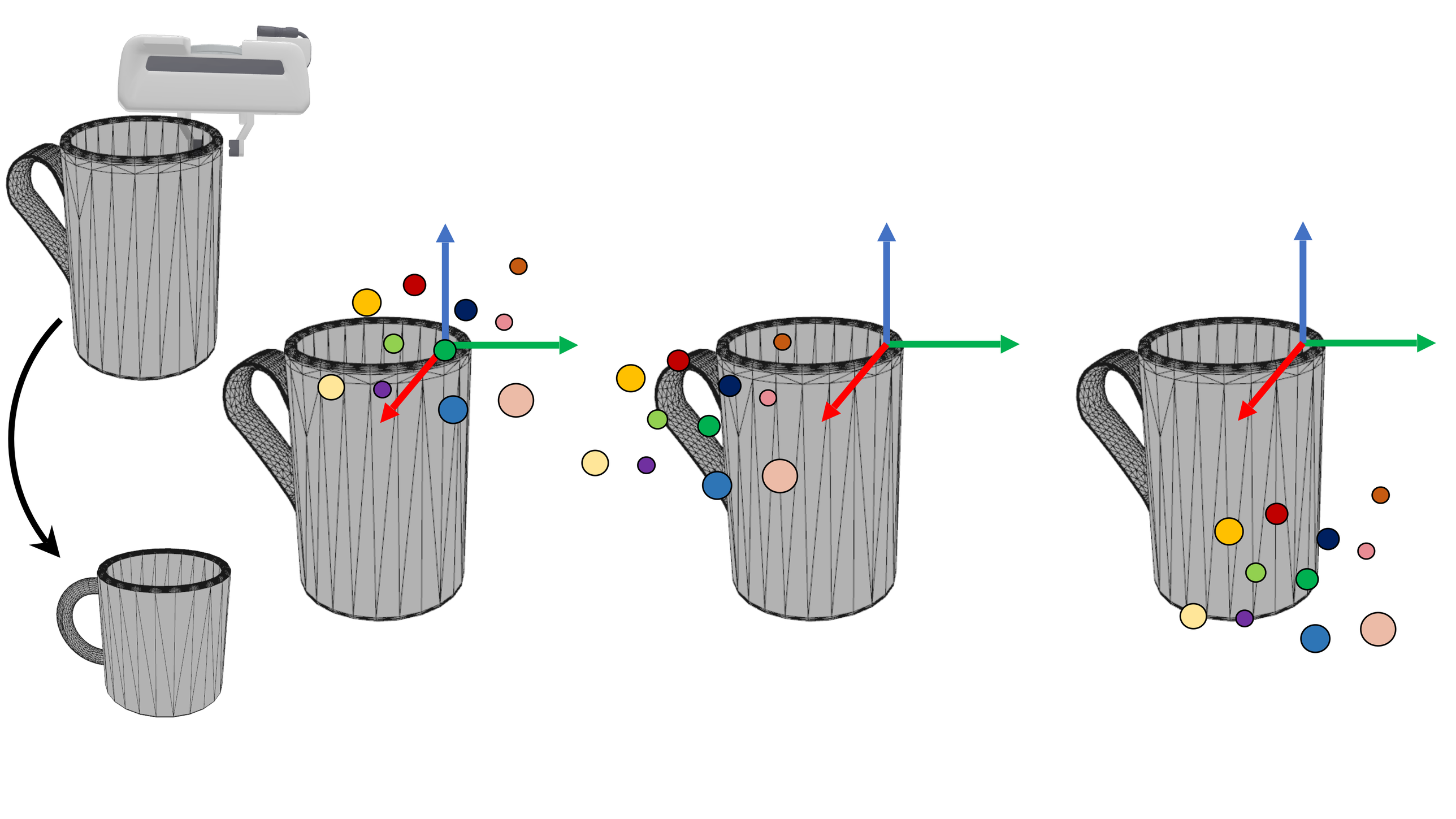
Repeat identical transform optimization for *grasping* and *placing*

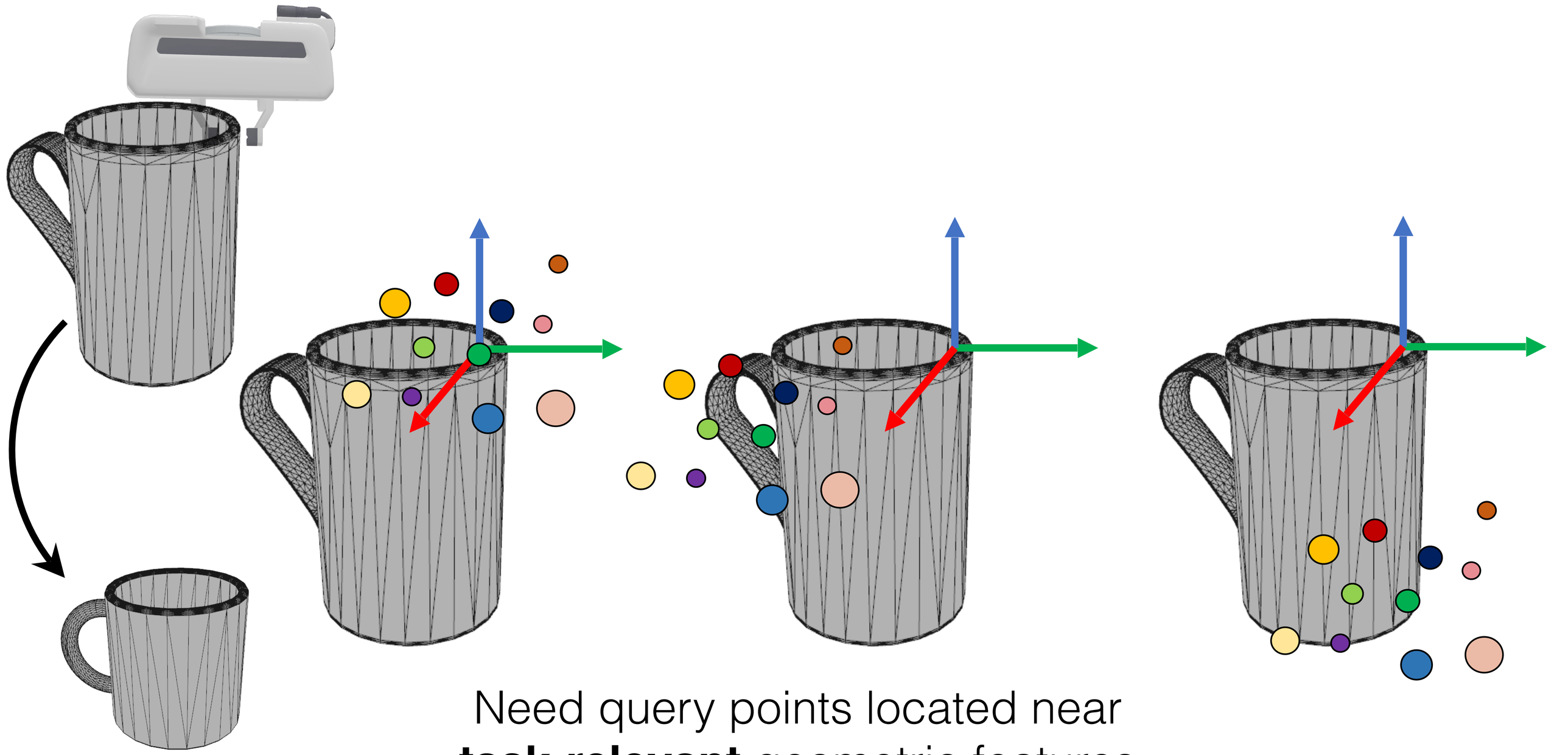




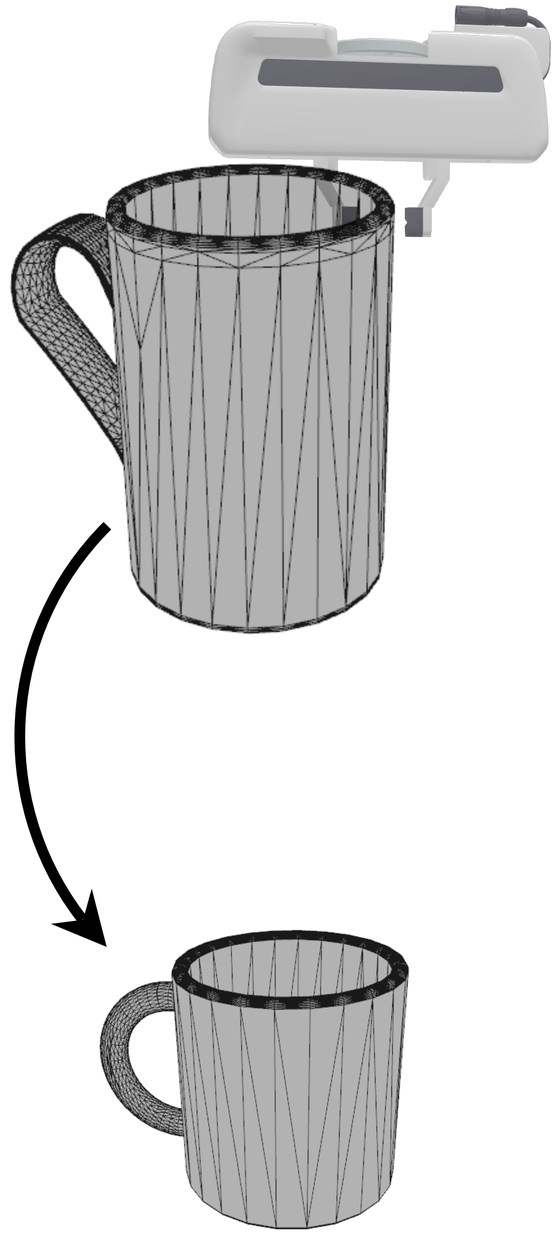


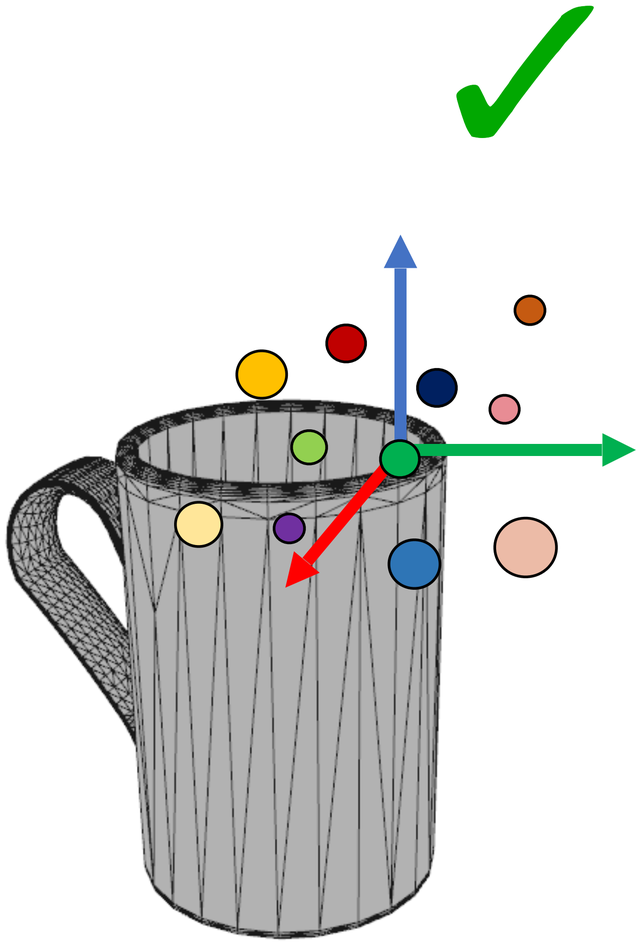


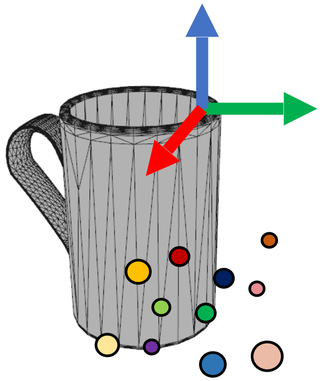


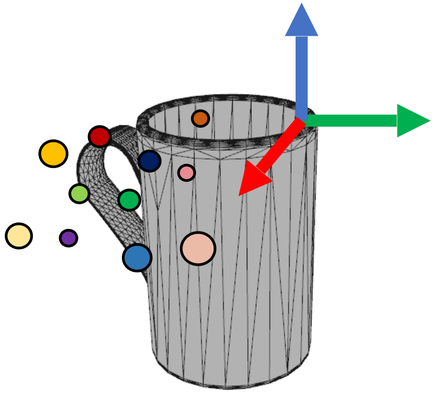


Need query points located near
task-relevant geometric features

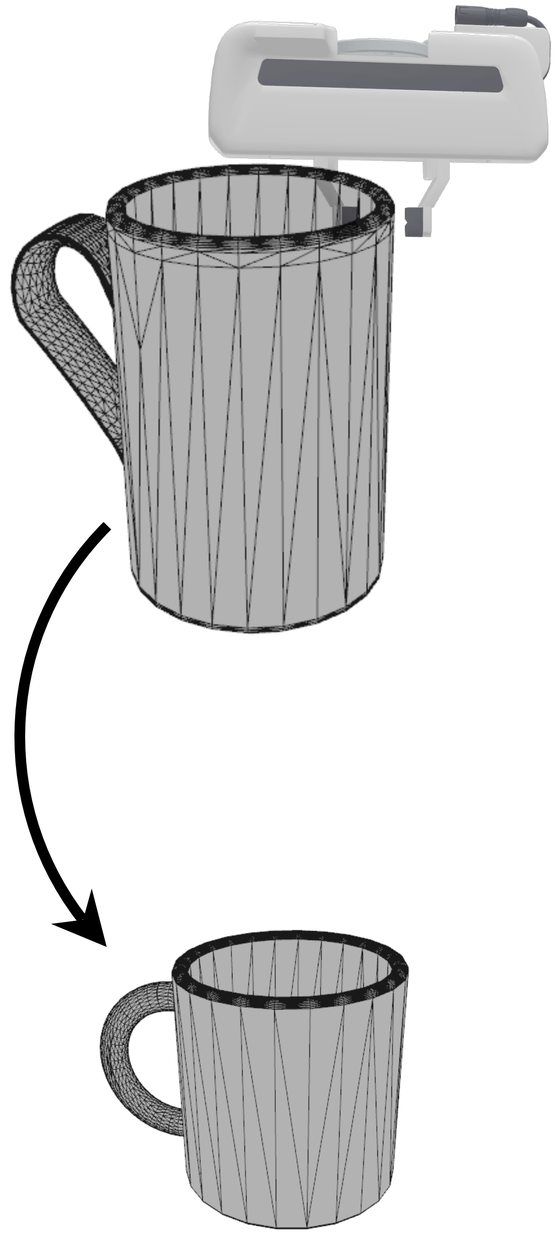


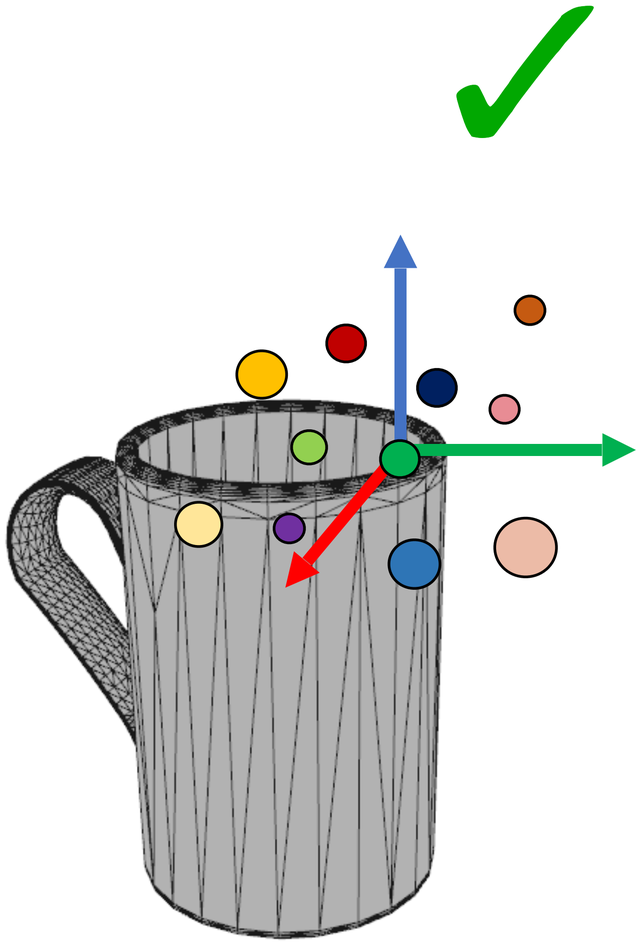


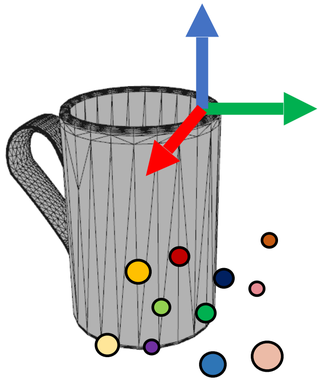


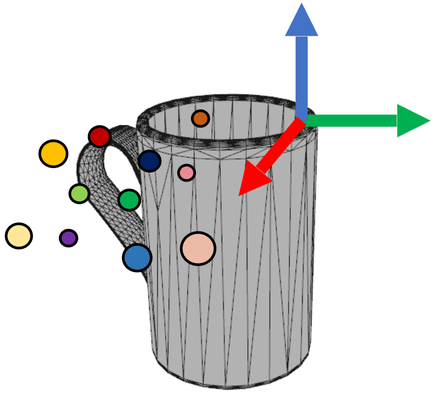


Sample query points near **contact locations**

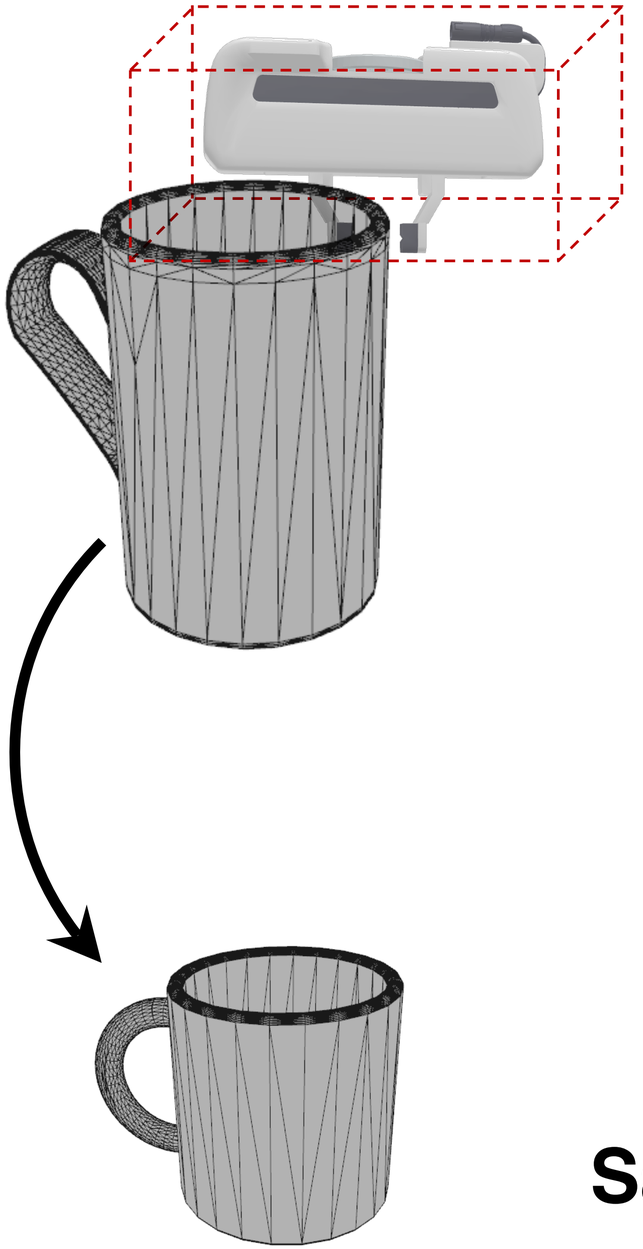


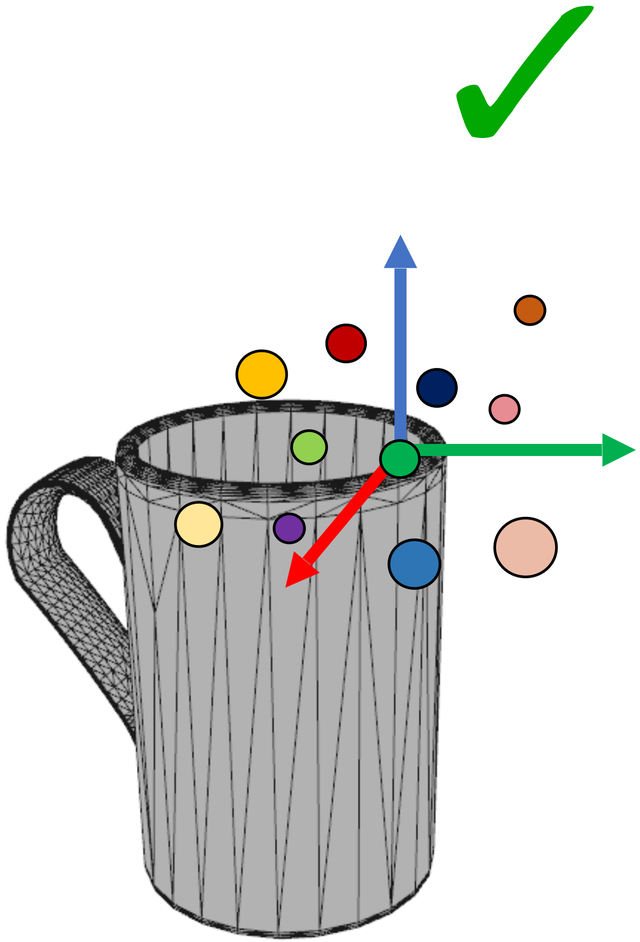


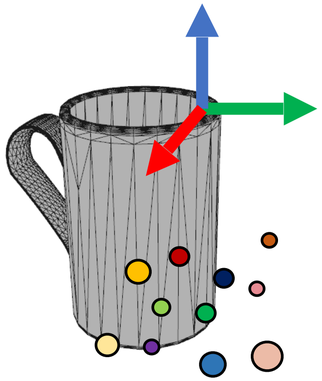


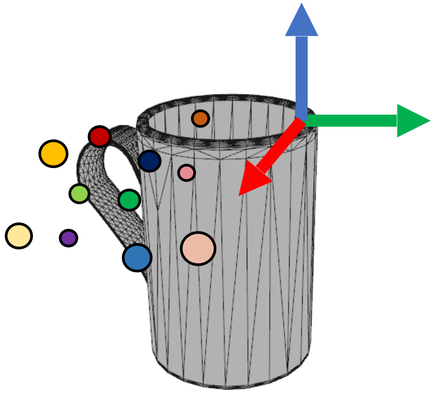


Sample query points near **contact locations**



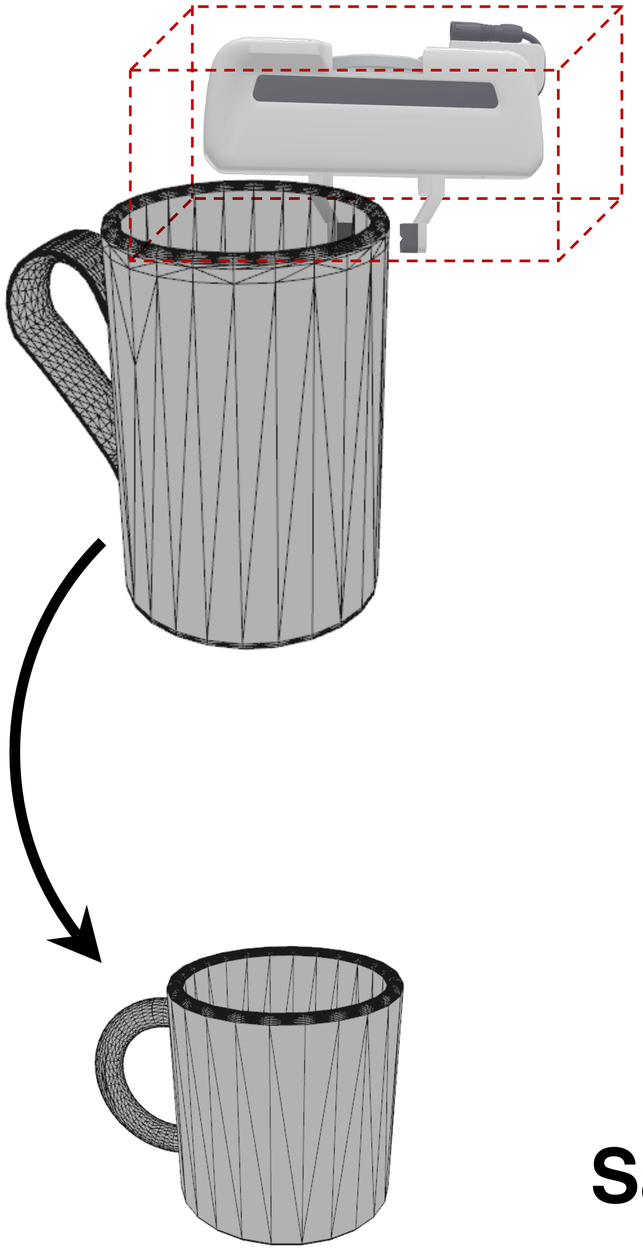


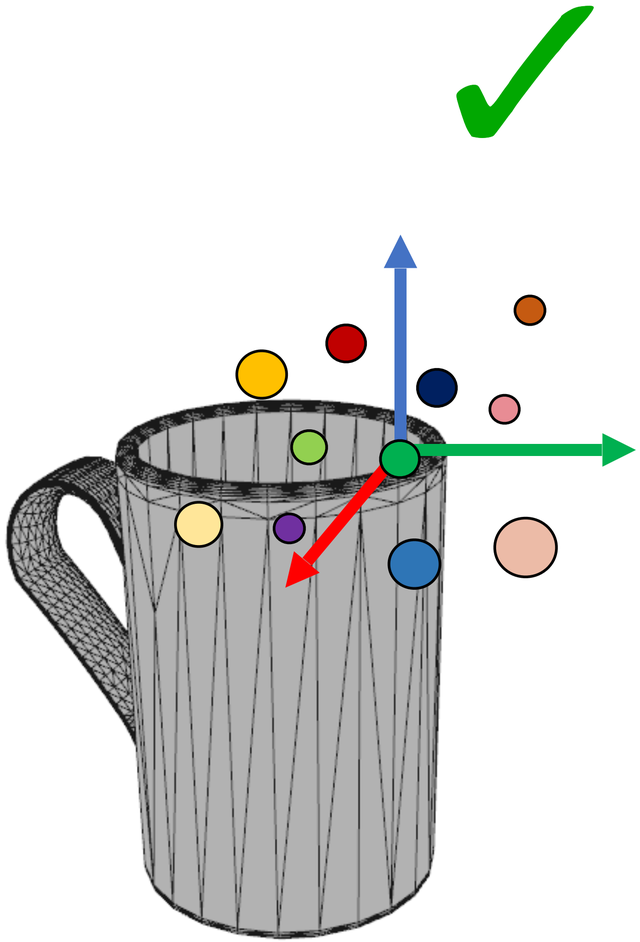


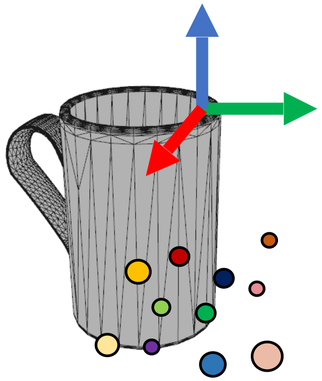


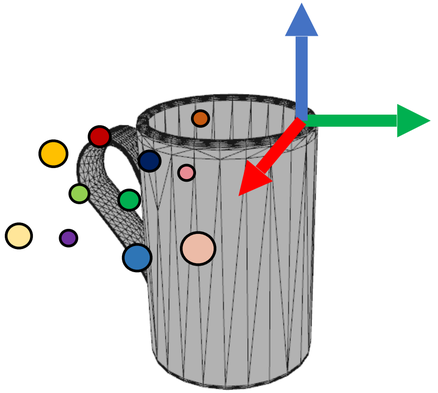
Sample query points near **contact locations**

- Achieved by sampling in the bounding box of the external object (gripper, rack, shelf)



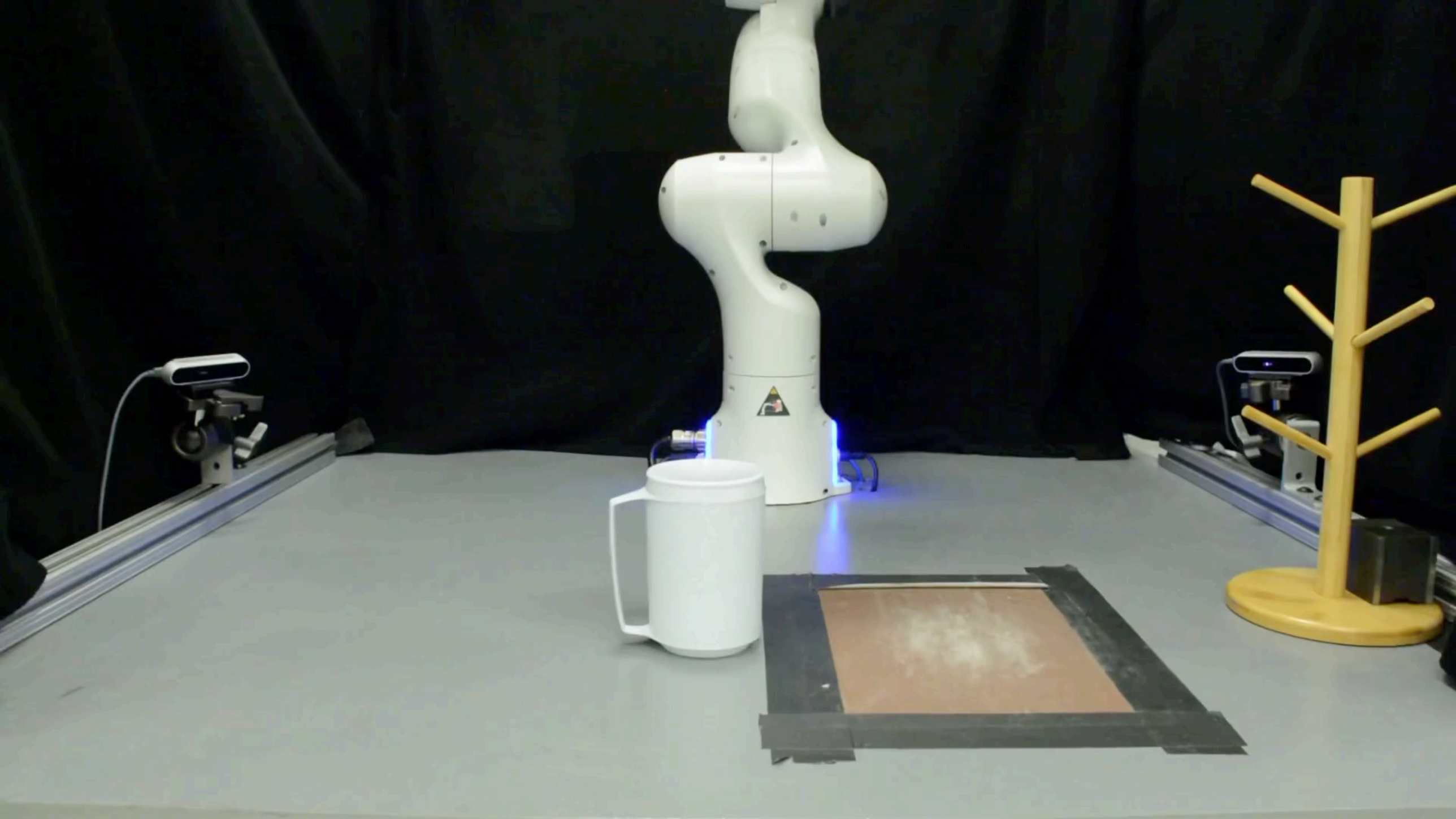


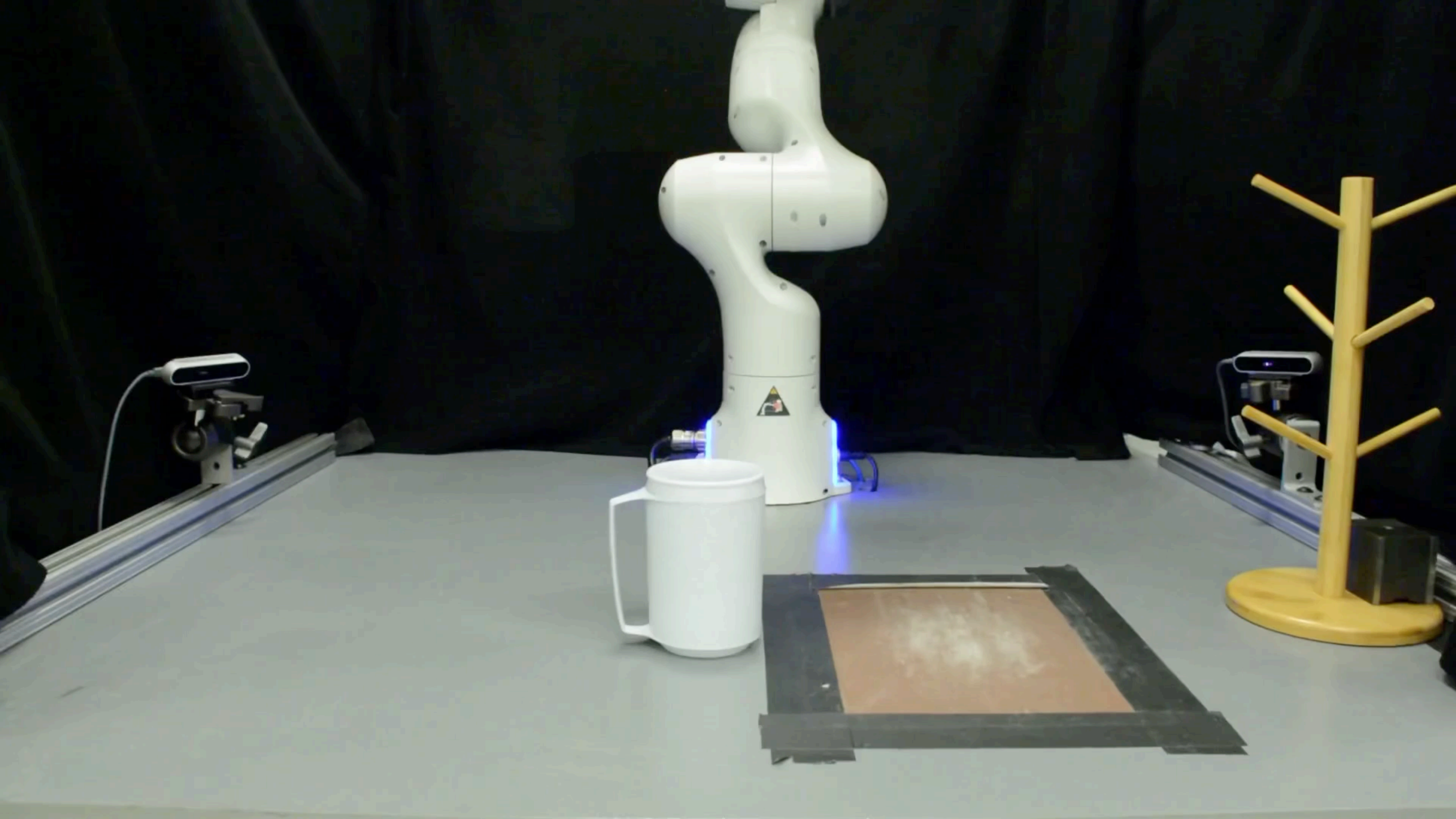


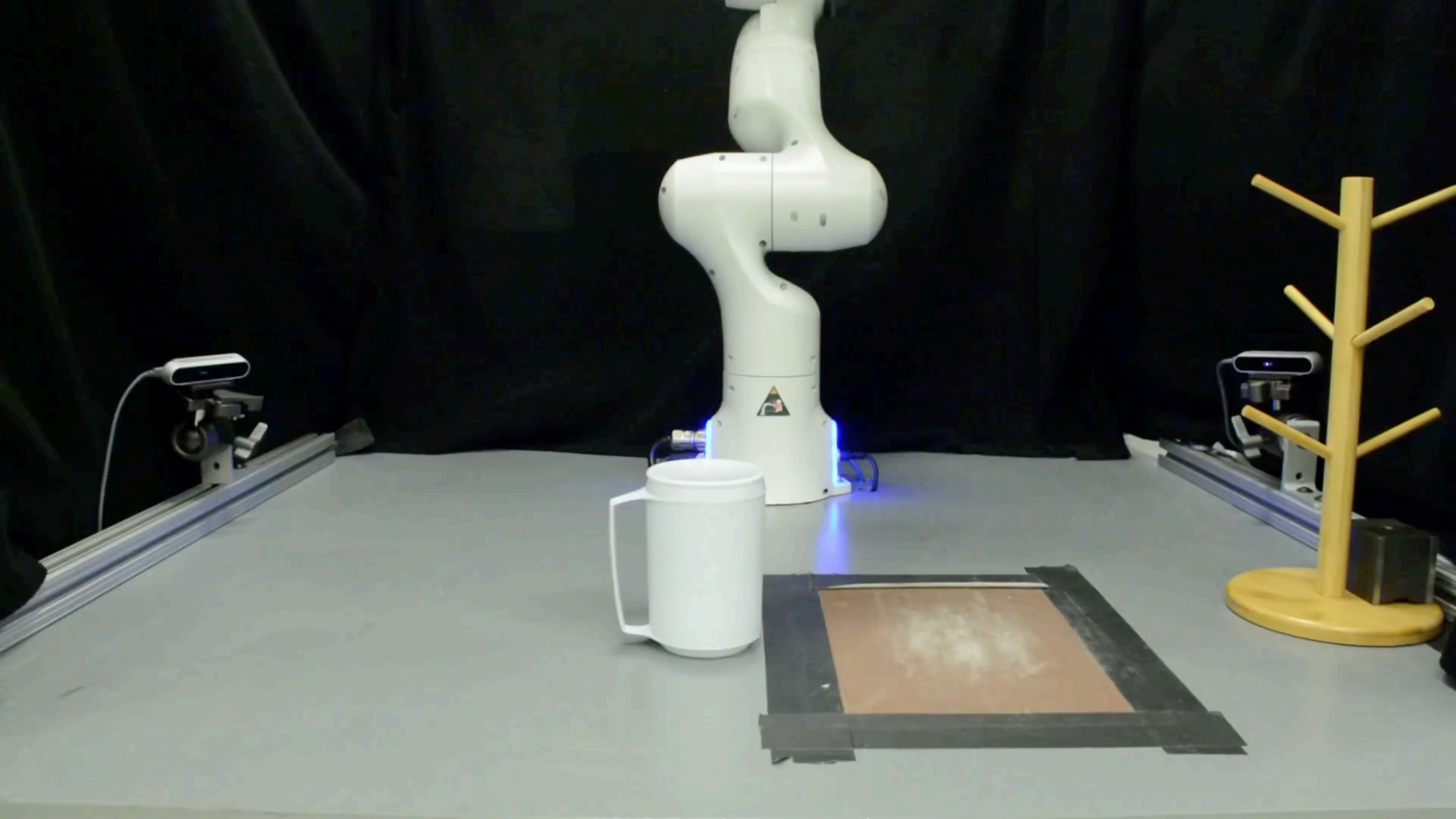


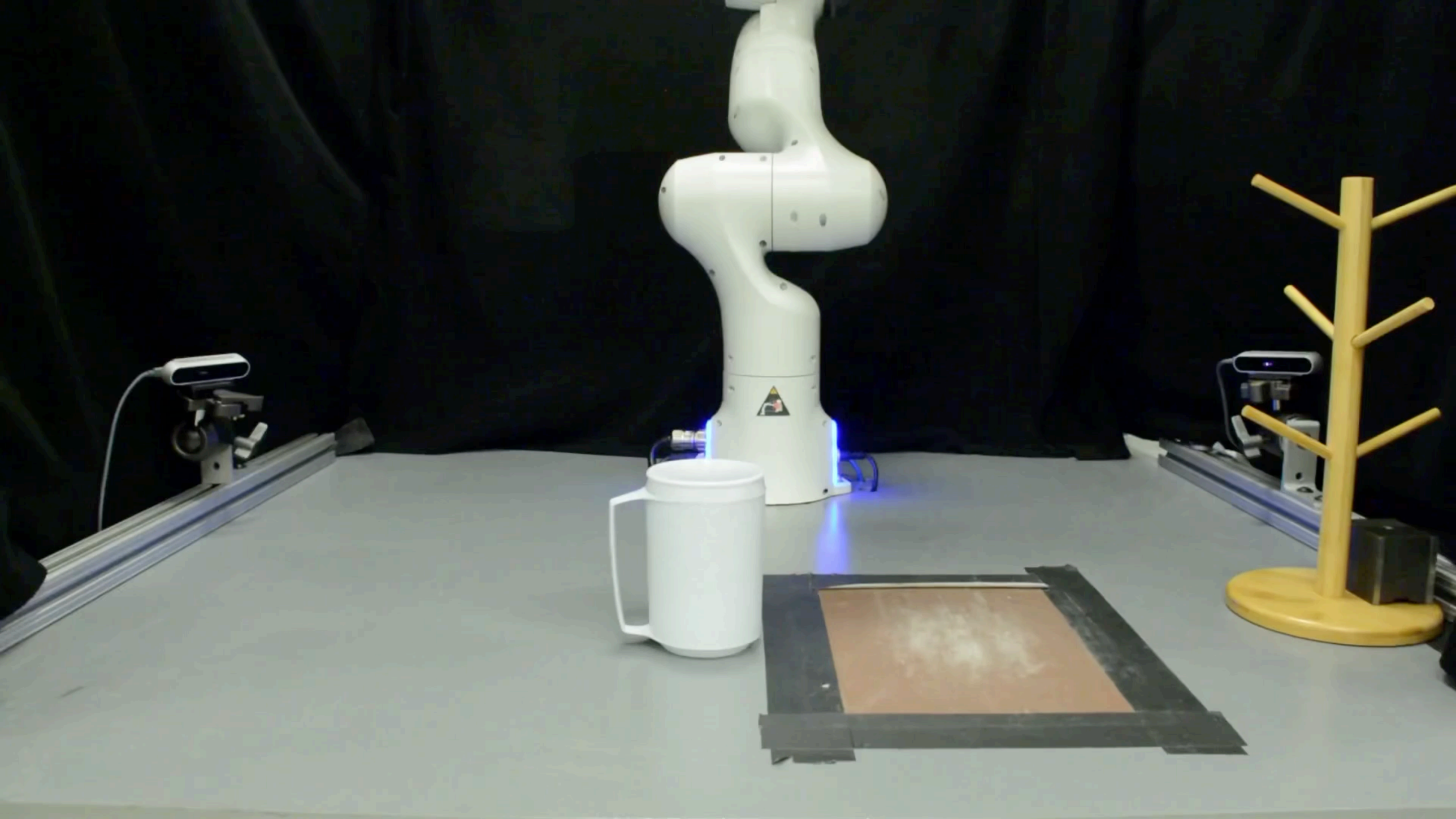
Sample query points near **contact locations**

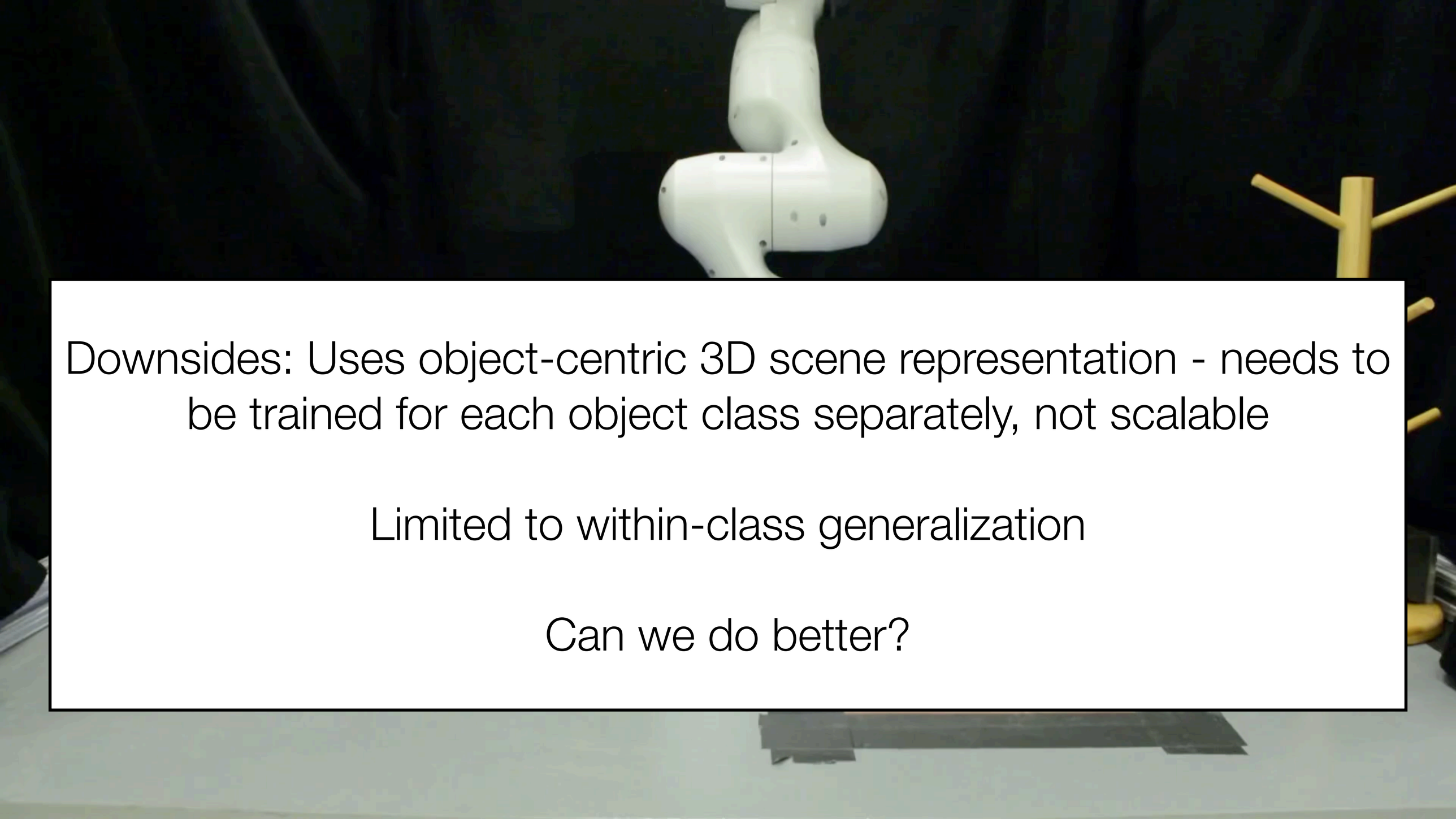
- Achieved by sampling in the bounding box of the external object (gripper, rack, shelf)









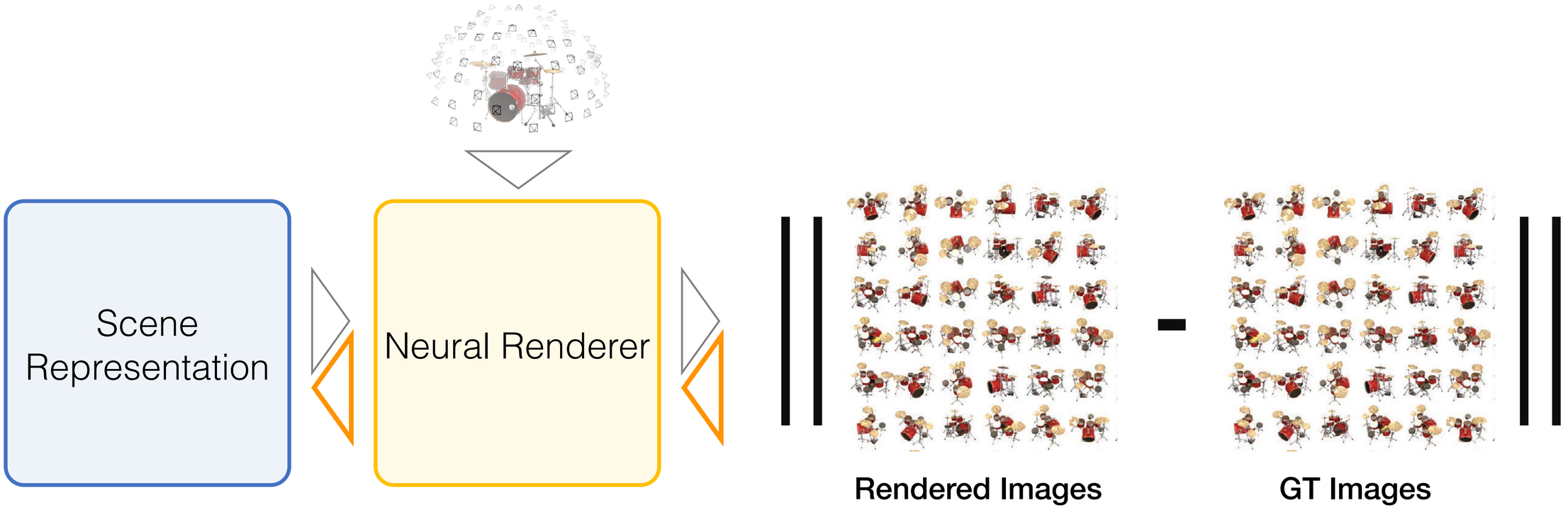


Downsides: Uses object-centric 3D scene representation - needs to be trained for each object class separately, not scalable

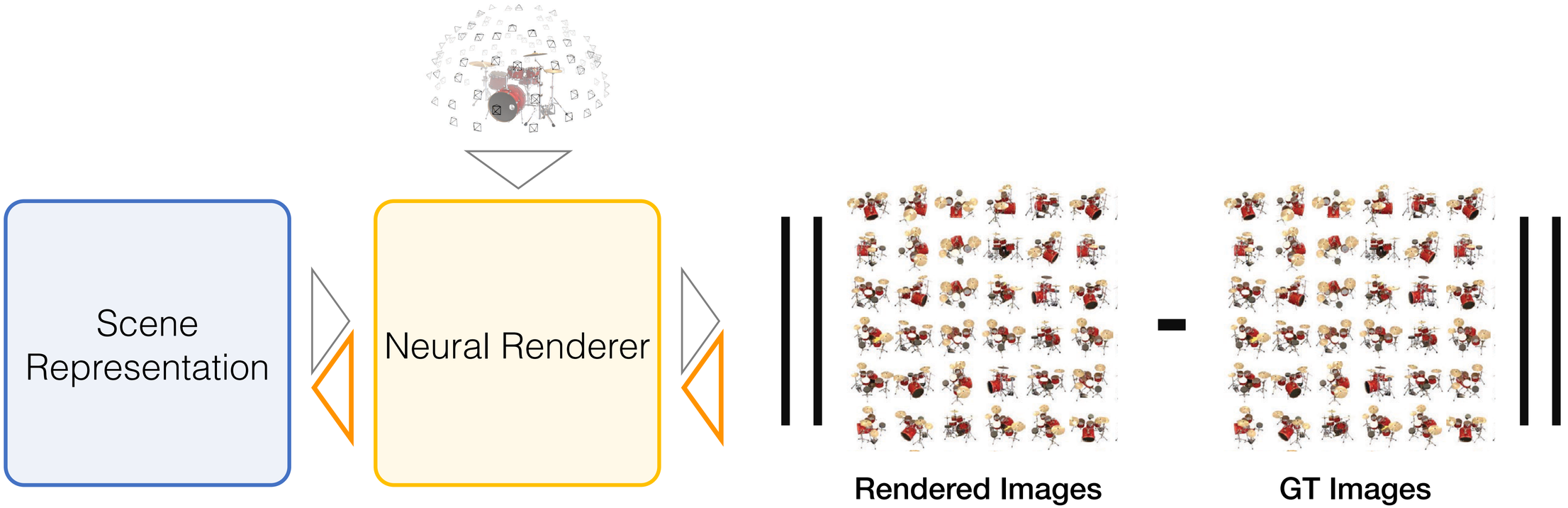
Limited to within-class generalization

Can we do better?

3D Reconstruction from *hundreds* of Images



3D Reconstruction from *hundreds* of Images



3D Reconstruction from *hundreds* of Images



NeRF, ECCV 2021



Plenoxels, CVPR 2022

3D Reconstruction from *hundreds* of Images



NeRF, ECCV 2021



Plenoxels, CVPR 2022

No *Understanding* of Content!



NeRF, ECCV 2021



Plenoxels, CVPR 2022

No *Understanding* of Content!



NeRF, ECCV 2021



Plenoxels, CVPR 2022

Usual formulation: Fit to RGB images.





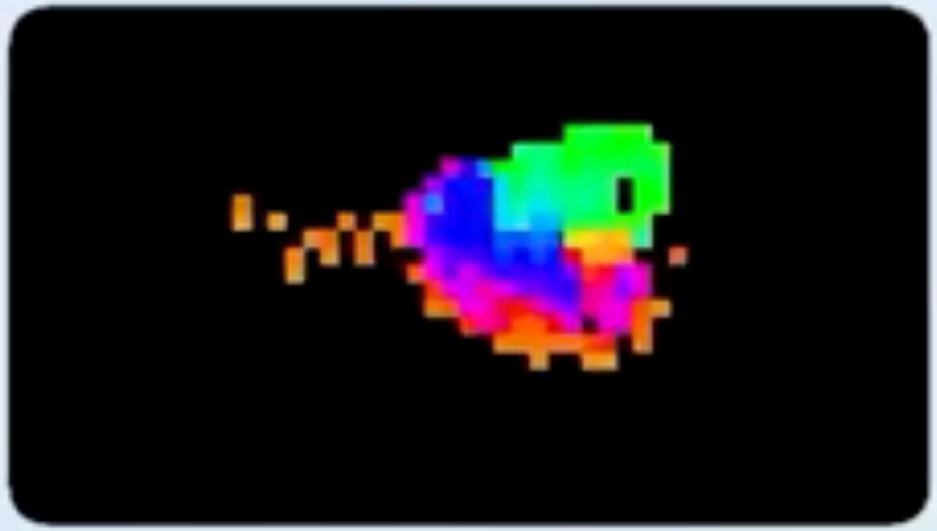




Can we leverage **2D** vision models?



DINO



DINOv2

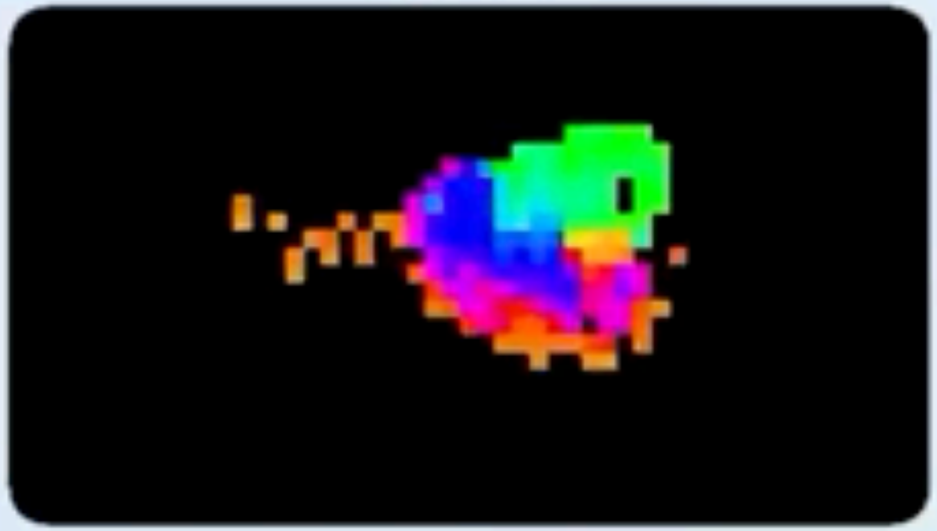


DINOv2, Oquab et al., arXiv 2023

Can we leverage **2D** vision models?



DINO



DINOv2



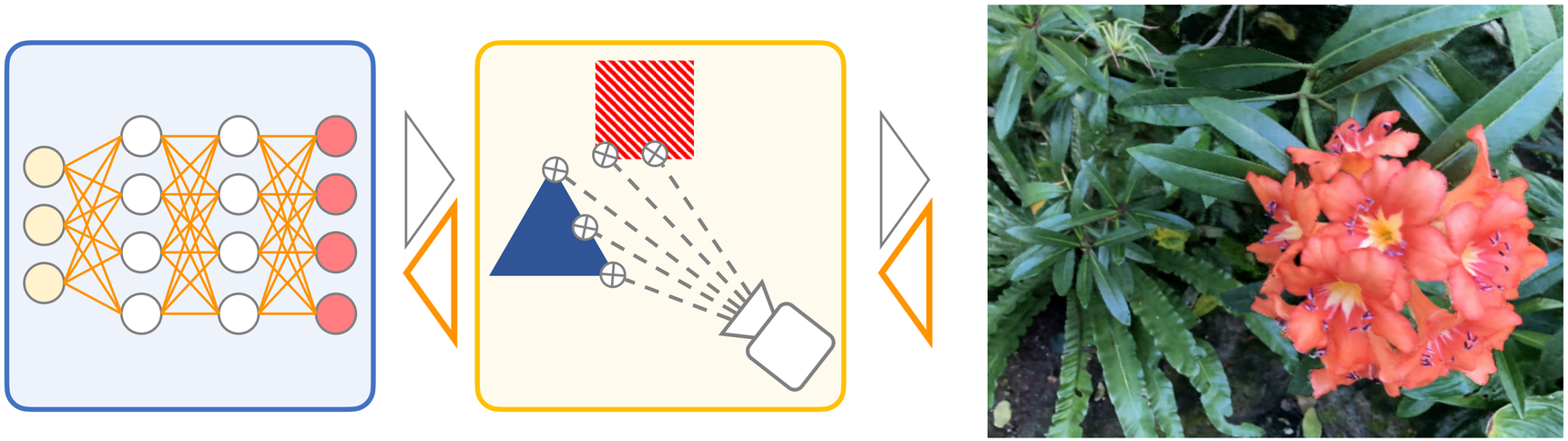
DINOv2, Oquab et al., arXiv 2023

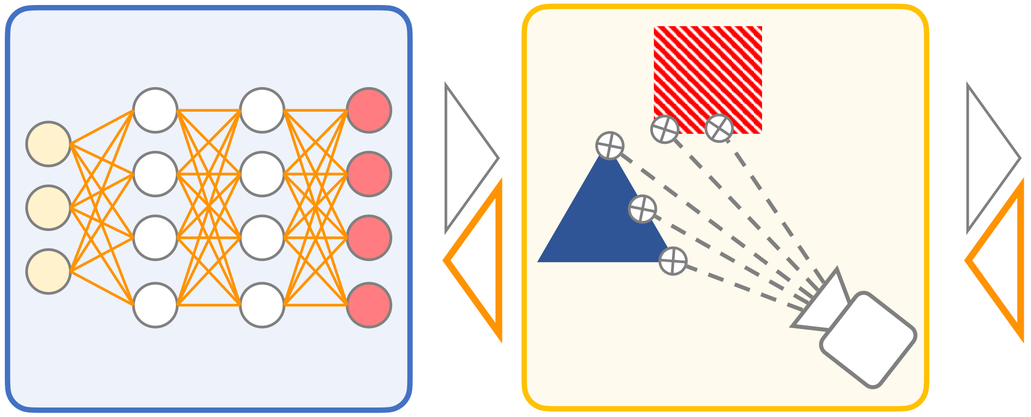
What happens if we fit on features?



DINO: Emerging properties in self-supervised vision transformers, Caron et al.



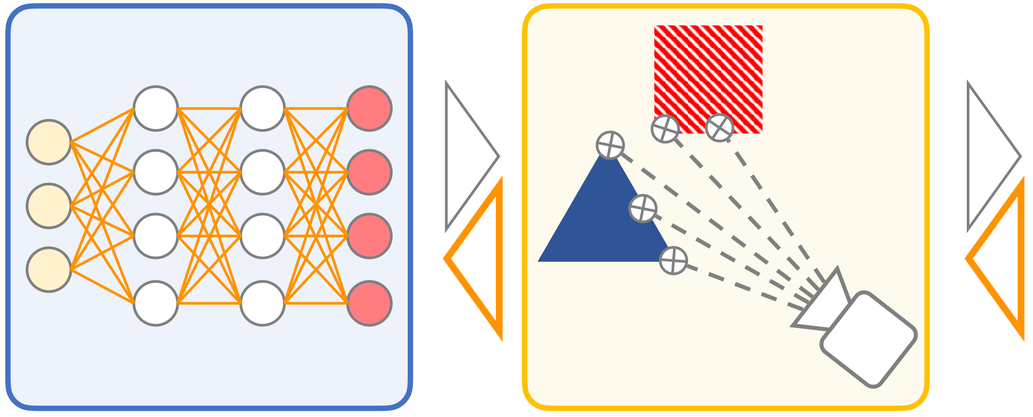


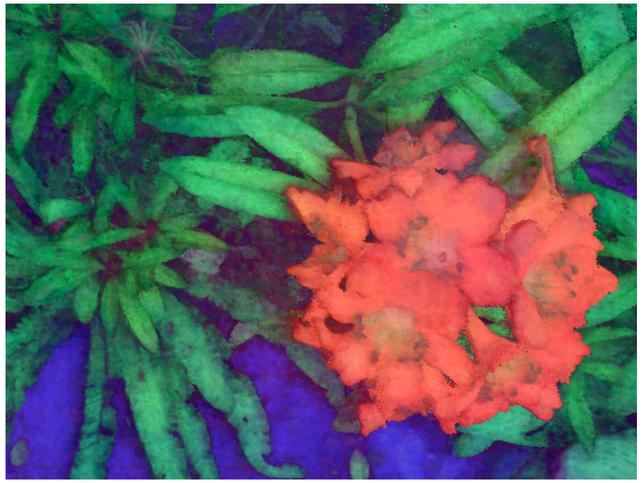




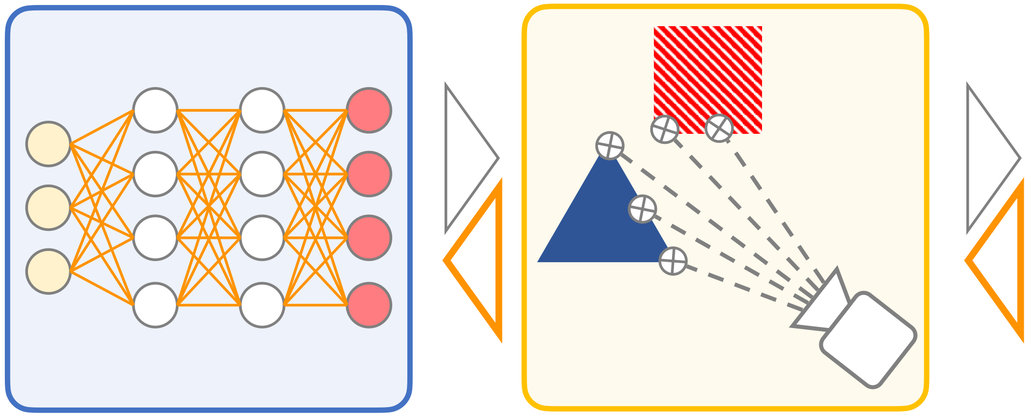
RGB

Image Feature Extractor





Features

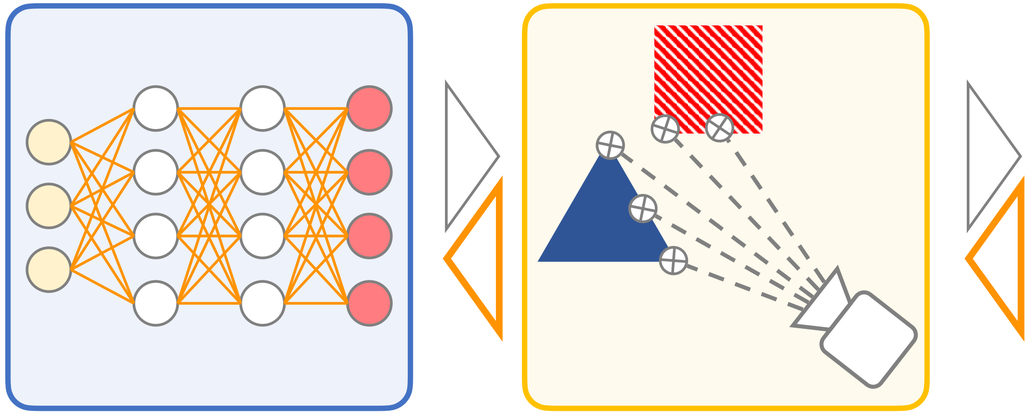


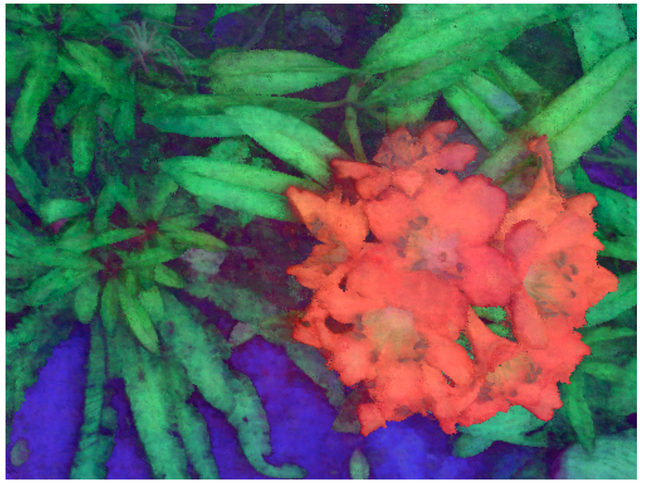


RGB

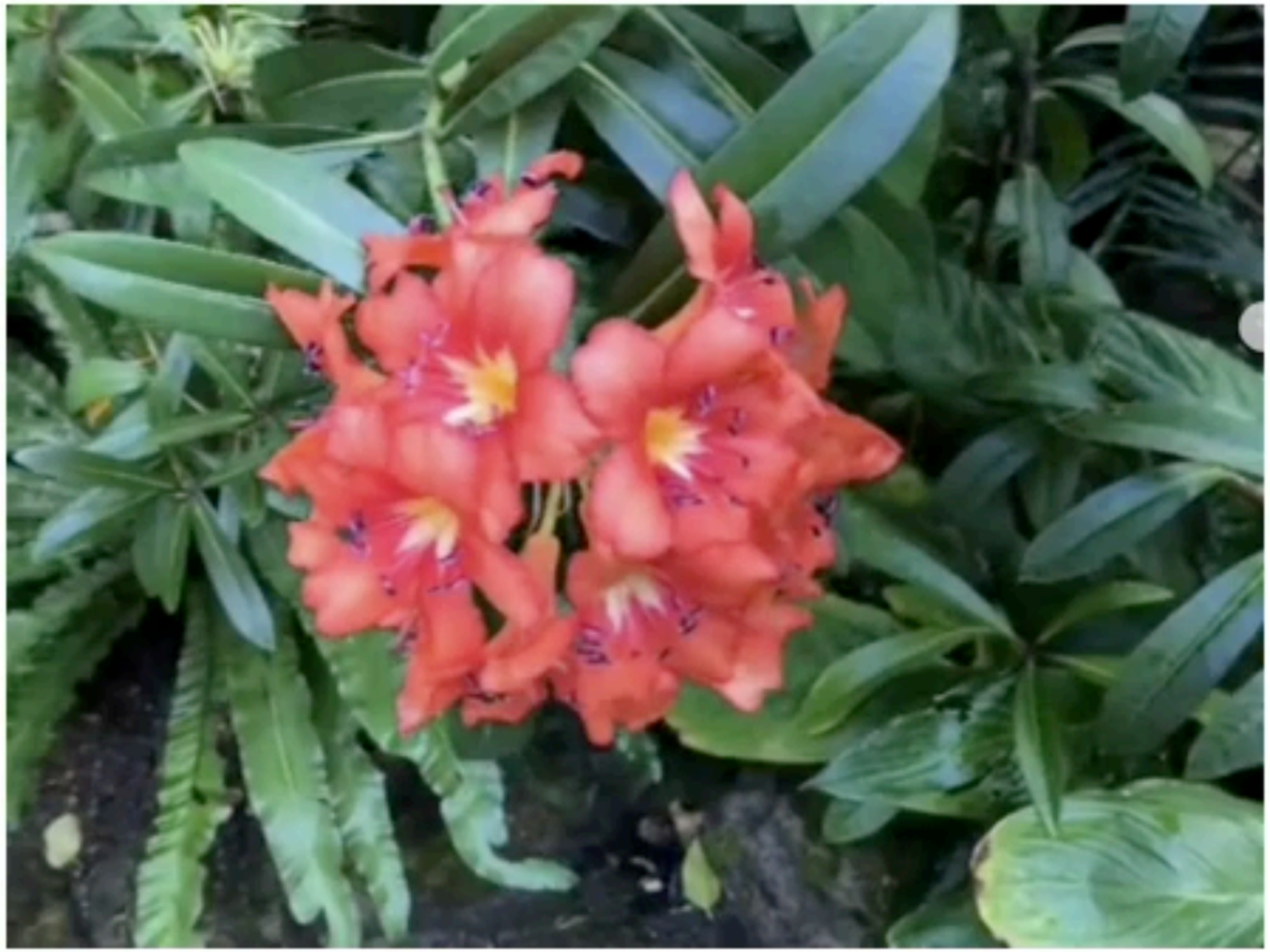
Image Feature Extractor

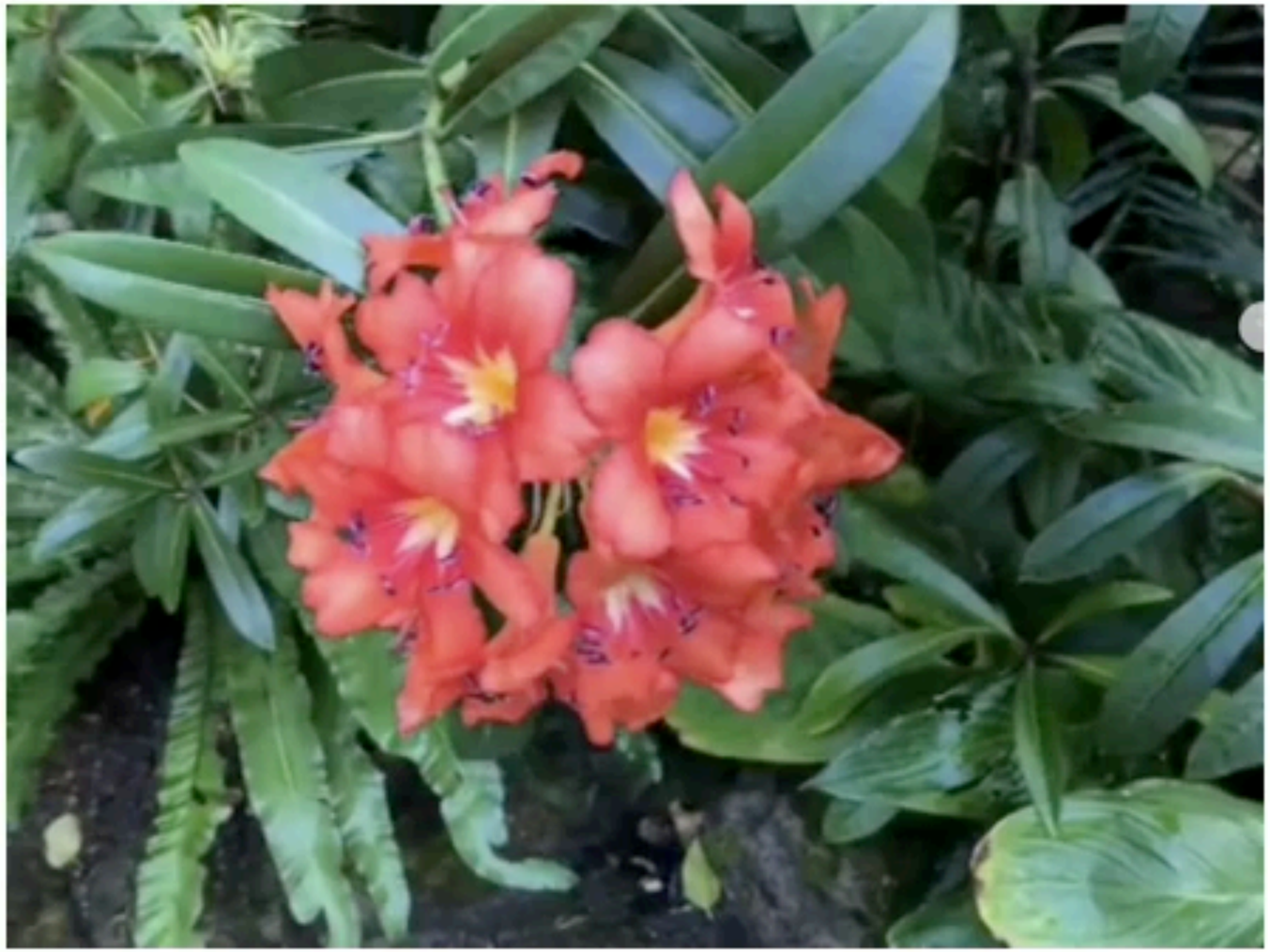
Distilled Feature Fields





Features





Follow-up Work: Distilled Feature Fields Enable Few-Shot Manipulation



1. Scan Scene

Distilled Feature Fields Enable Few-Shot Manipulation, Shen et al., 2023

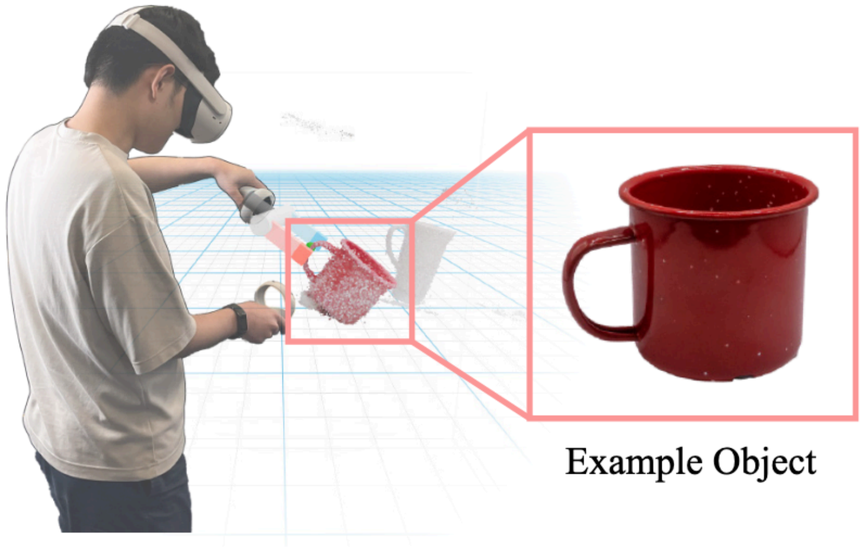
Follow-up Work: Distilled Feature Fields Enable Few-Shot Manipulation



1. Scan Scene

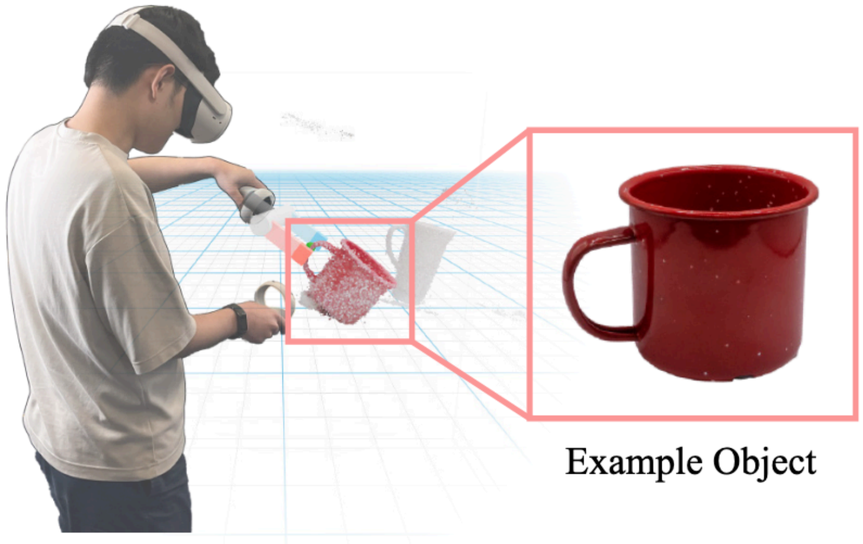
Distilled Feature Fields Enable Few-Shot Manipulation, Shen et al., 2023

Collecting Demonstrations

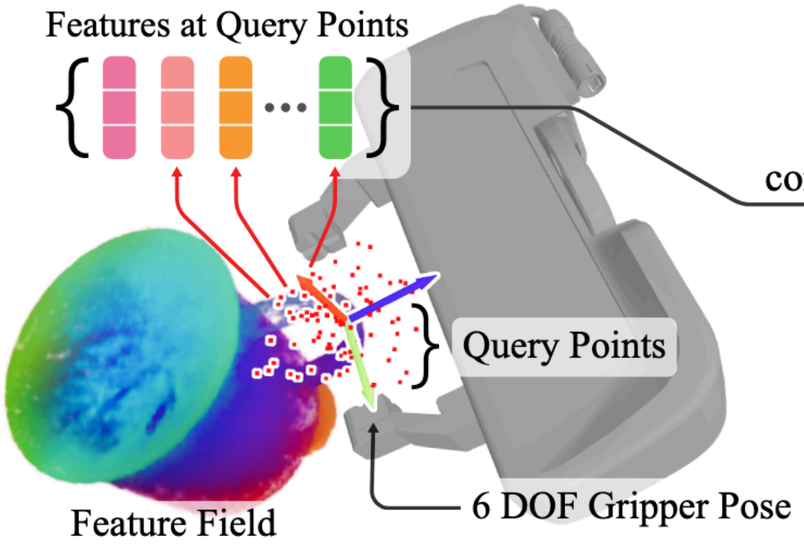


(a) Collect Demonstrations in VR

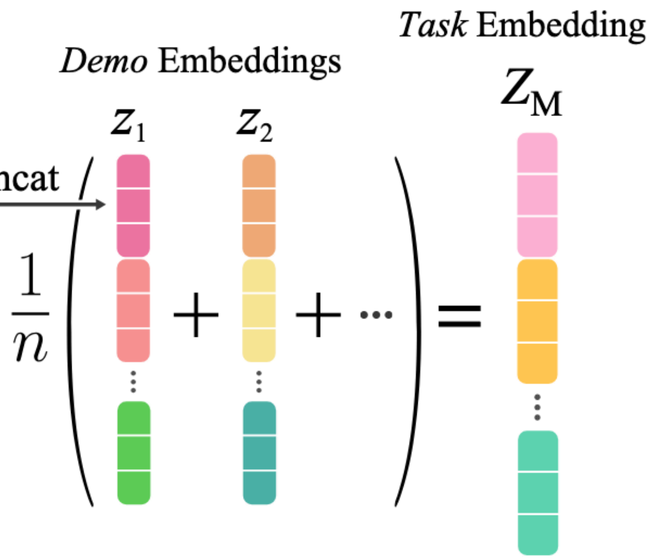
Collecting Demonstrations



(a) Collect Demonstrations in VR

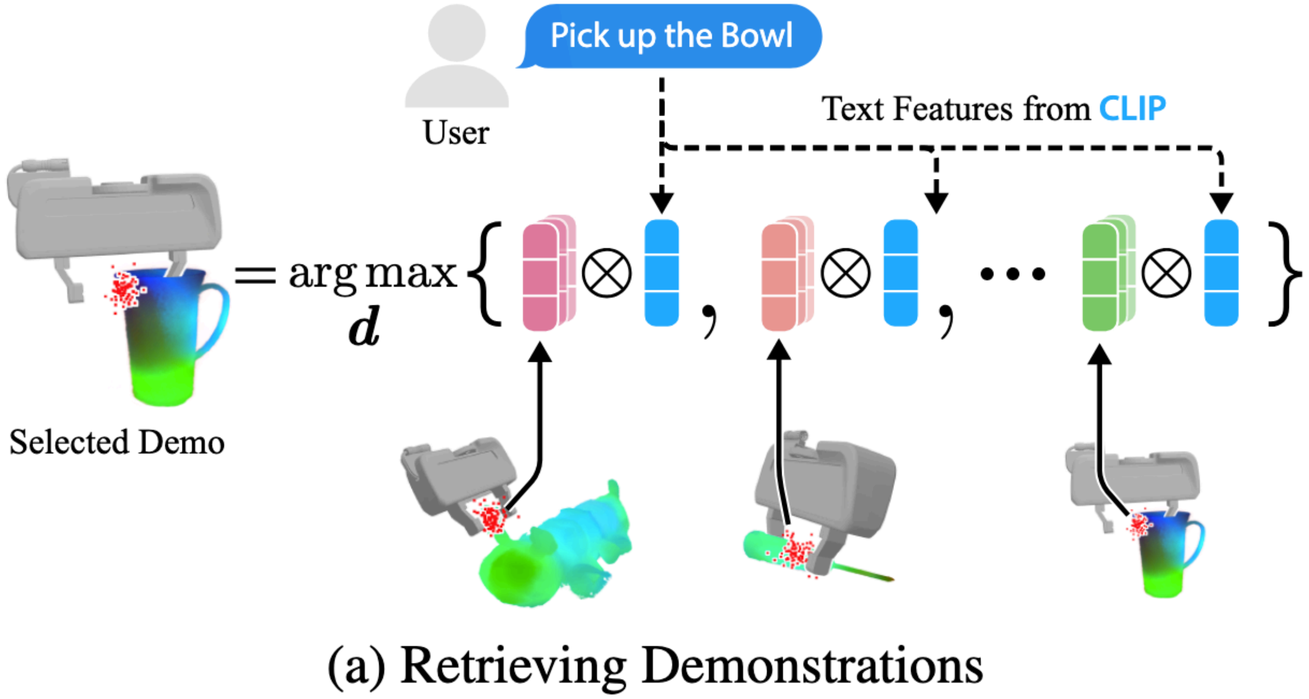


(b) Sample Feature Vectors



(c) Average Over *n* Demos

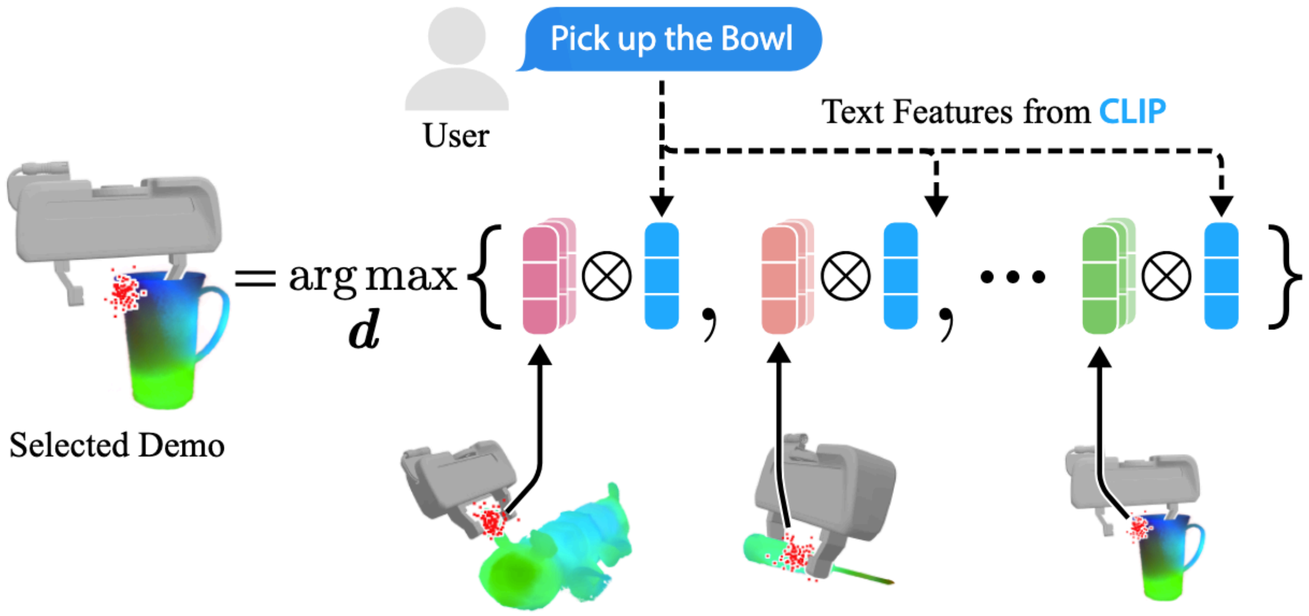
Retrieval & Execution



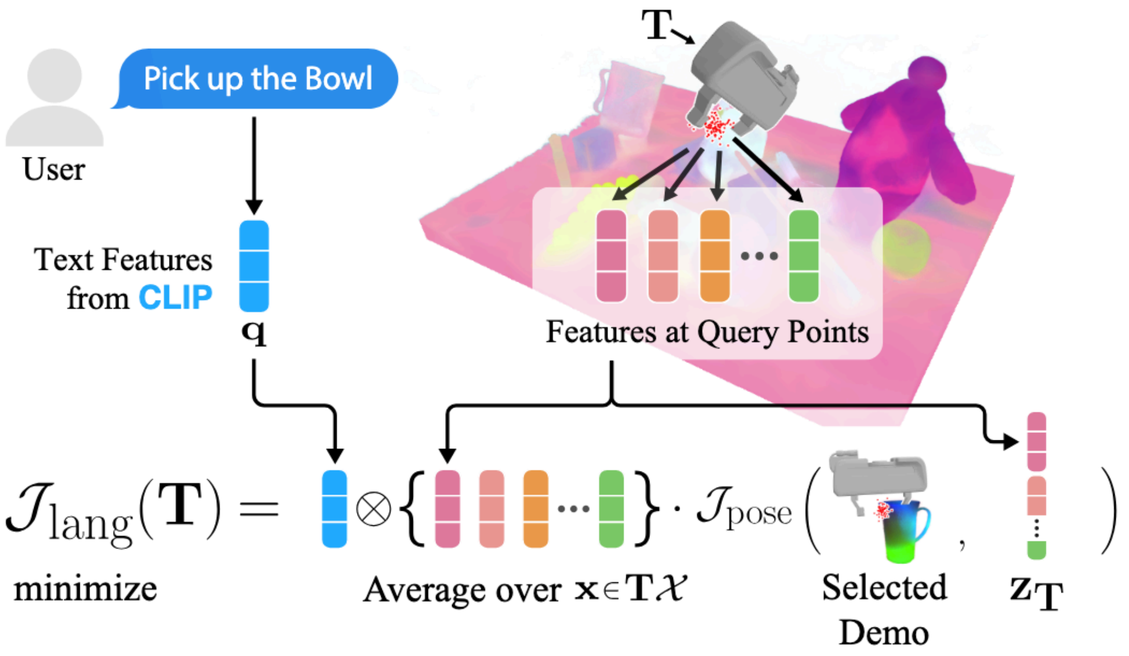
(a) Retrieving Demonstrations

Find demo where the average CLIP feature at the demo pose is highest

Retrieval & Execution



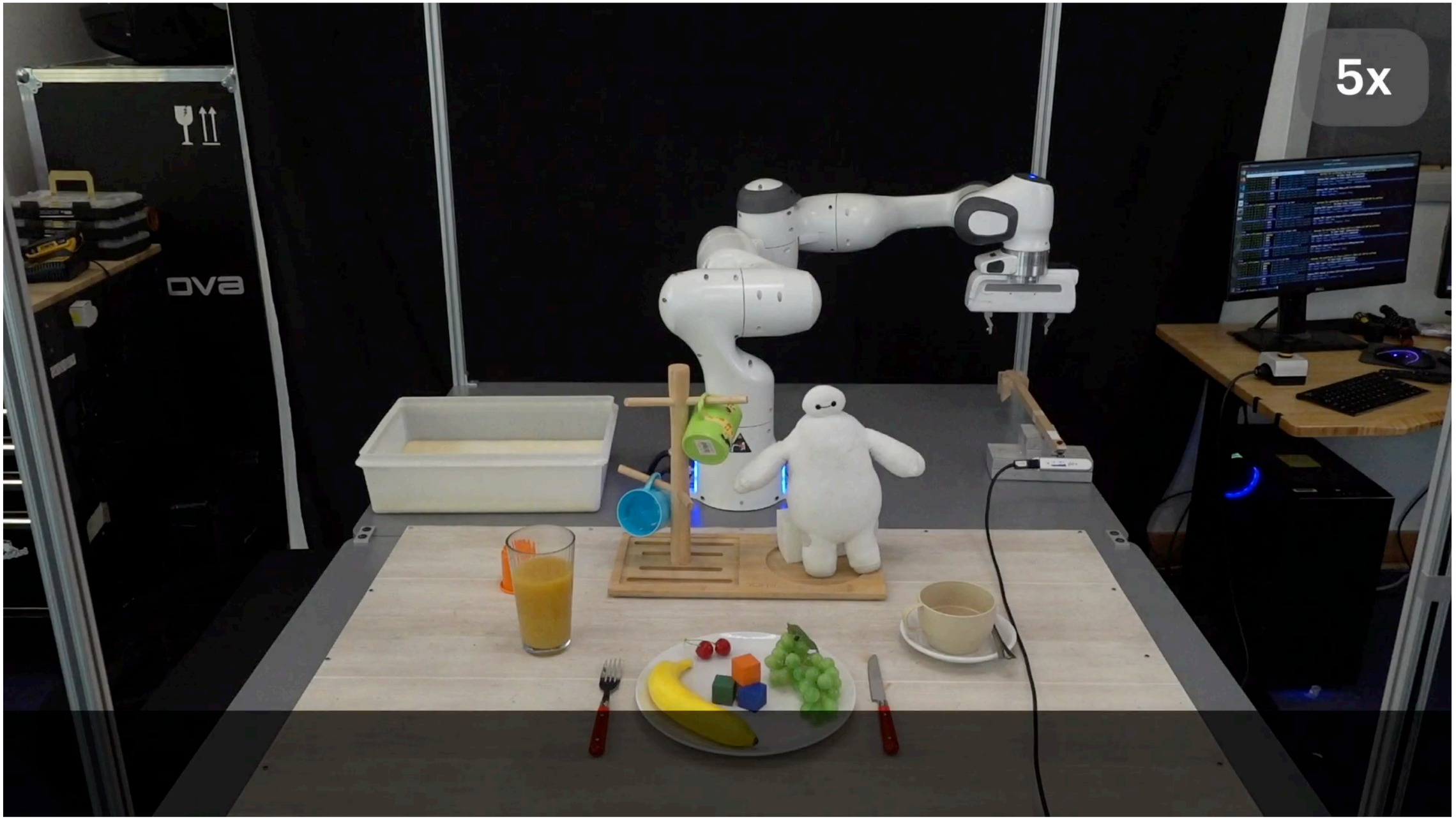
(a) Retrieving Demonstrations



(b) Language-Guided Pose Optimization

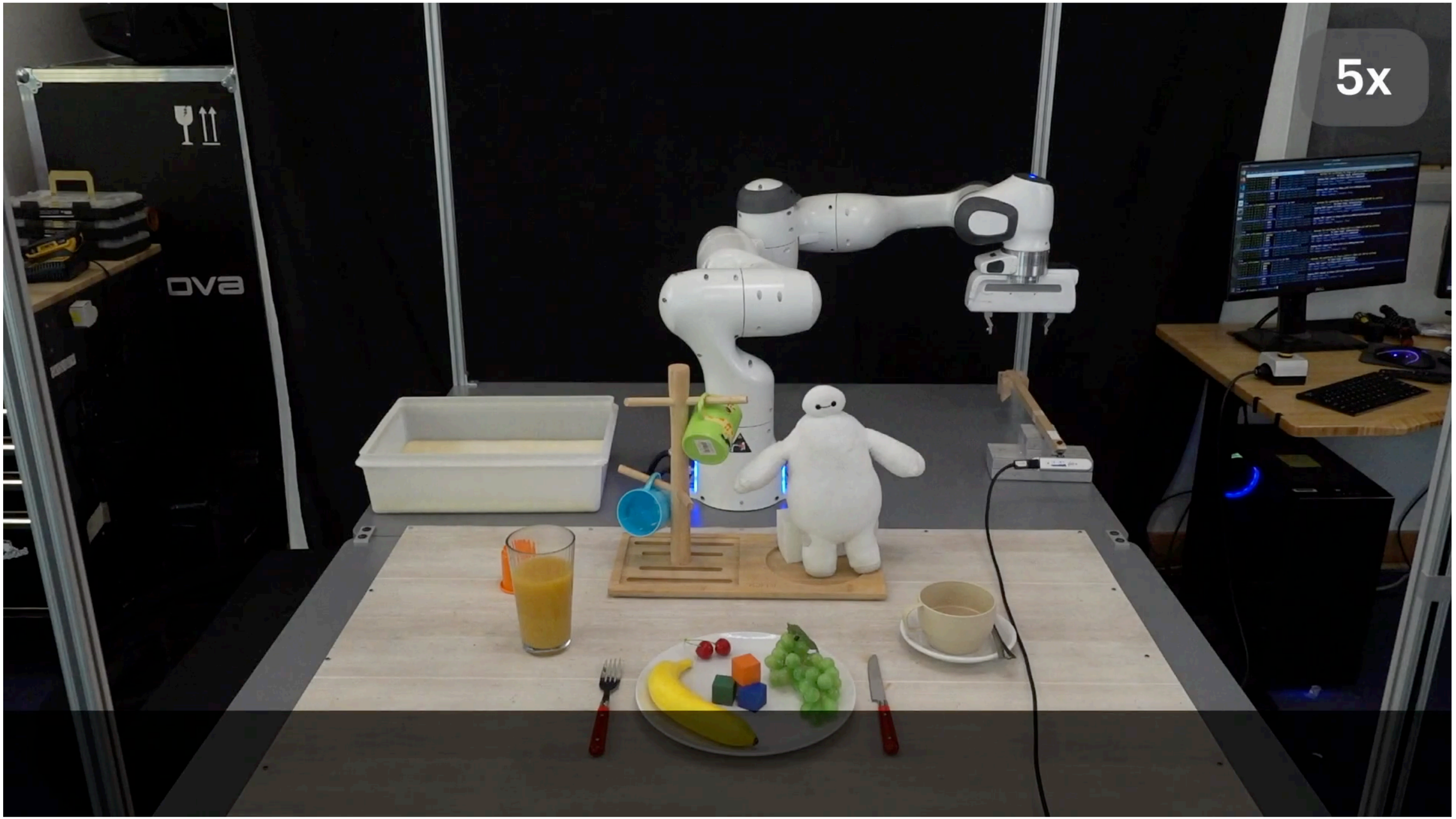
Optimize for pose such that features at query points match those in the demo (like in Neural Descriptor Fields)

Distilled Feature Fields Enable Few-Shot Manipulation



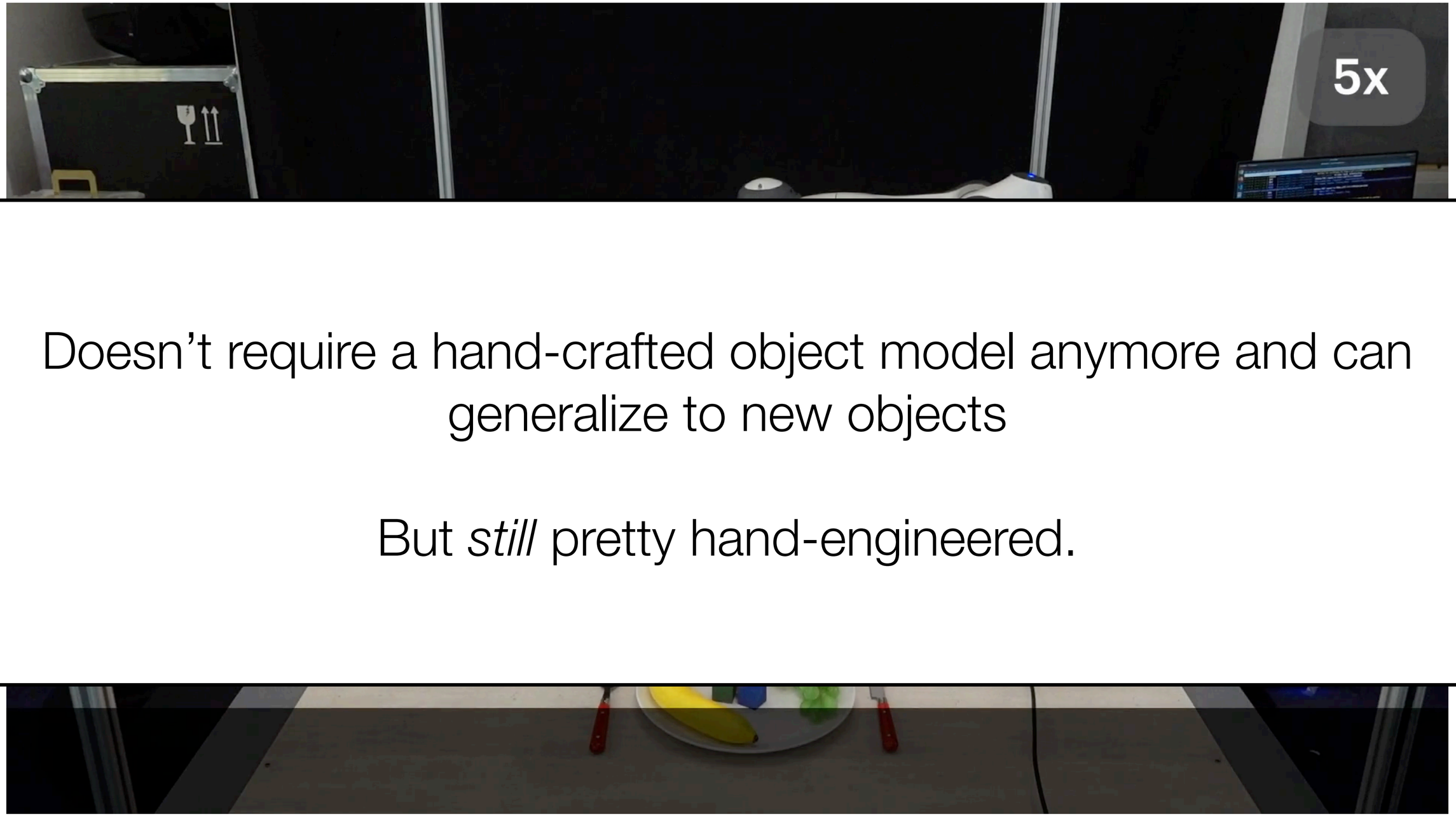
Distilled Feature Fields Enable Few-Shot Manipulation, Shen et al., 2023

Distilled Feature Fields Enable Few-Shot Manipulation



Distilled Feature Fields Enable Few-Shot Manipulation, Shen et al., 2023

Distilled Feature Fields Enable Few-Shot Manipulation



Doesn’t require a hand-crafted object model anymore and can generalize to new objects

But *still* pretty hand-engineered.

Distilled Feature Fields Enable Few-Shot Manipulation, Shen et al., 2023

Collecting Demonstrations



Collecting Demonstrations

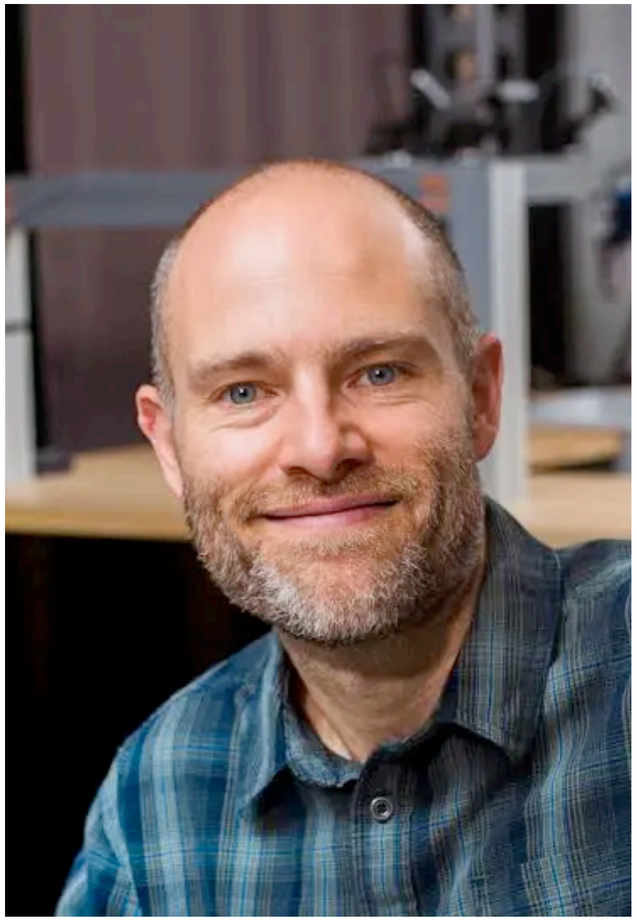


Behavior Cloning:
Direct Supervised Learning of Policies

Many slides from:
Sergey Levine, Berkeley CS 285
Russ Tedrake, RSS tutorials on Diffusion Policy

Check out Sergey's and Russ's talks & lectures if you want to learn more!

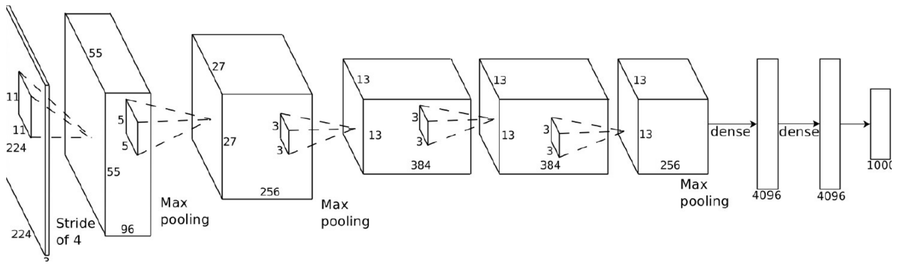




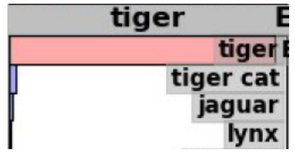
Terminology & notation



$\mathbf{o}$



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



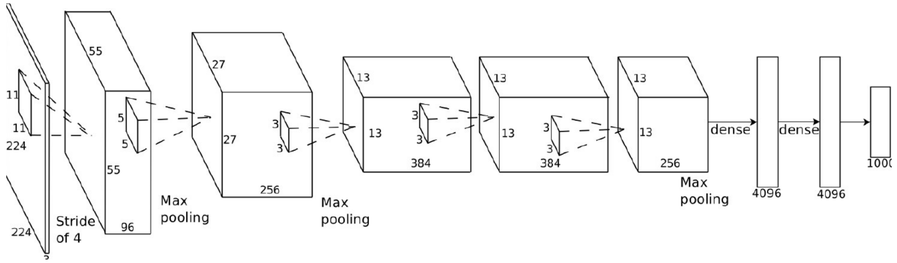
$\mathbf{a}$

Slide credit: Sergey Levine

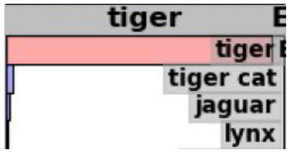
Terminology & notation



$\mathbf{o}_t$



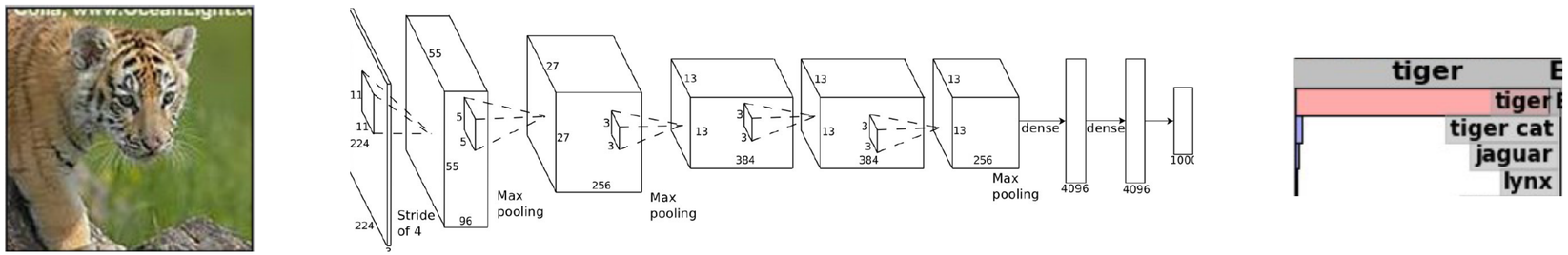
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$



$\mathbf{a}_t$

Slide credit: Sergey Levine

Terminology & notation

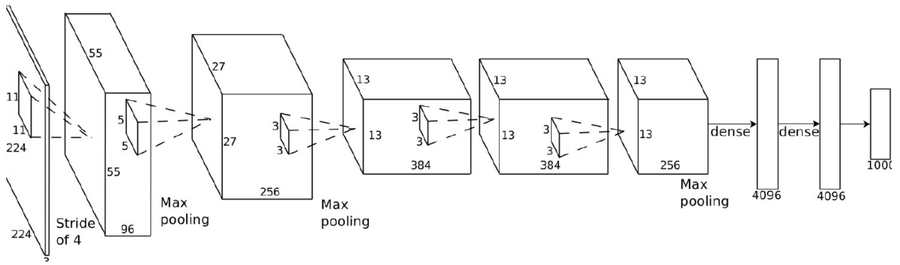


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

Slide credit: Sergey Levine

Terminology & notation



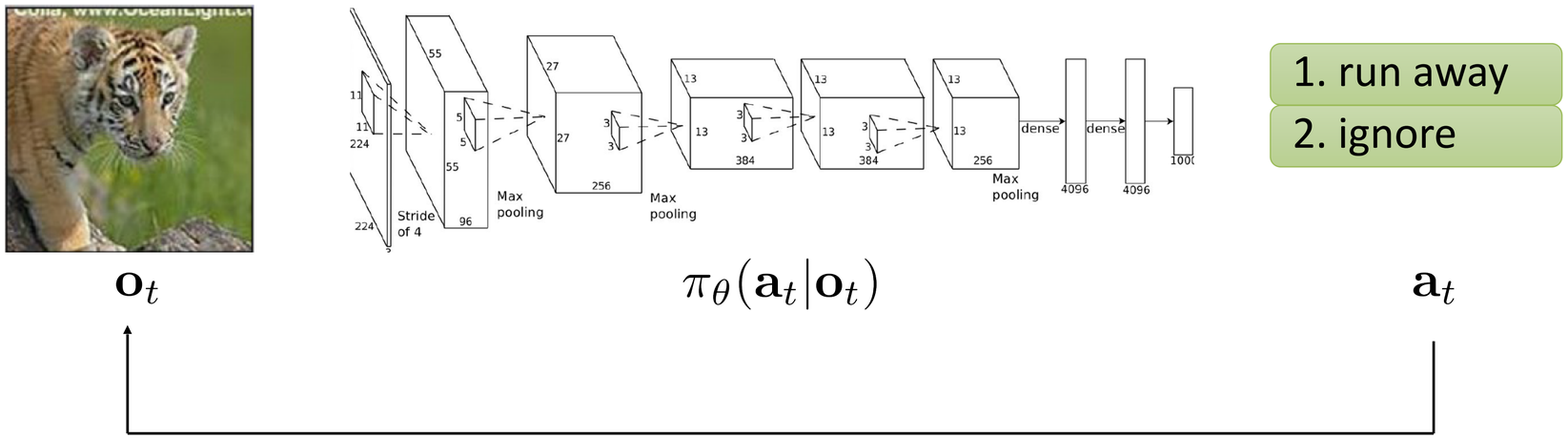


1. run away

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

Slide credit: Sergey Levine

Terminology & notation



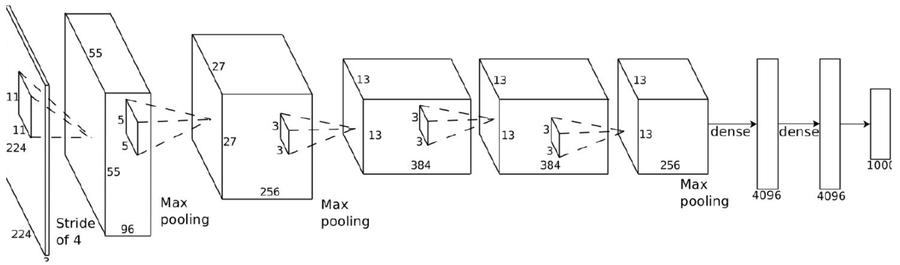
1. run away
2. ignore

Slide credit: Sergey Levine

Terminology & notation



$\mathbf{o}_t$



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

1. run away
2. ignore
3. pet

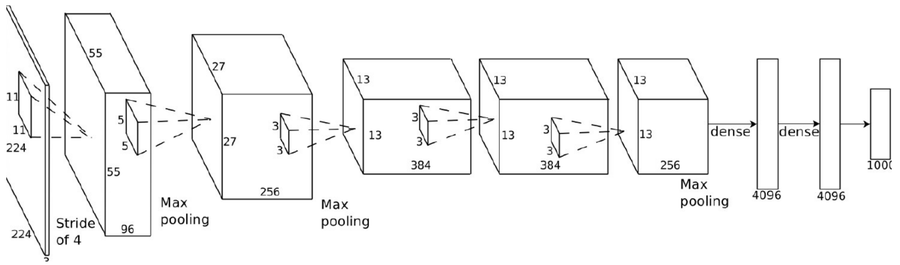
$\mathbf{a}_t$

Slide credit: Sergey Levine

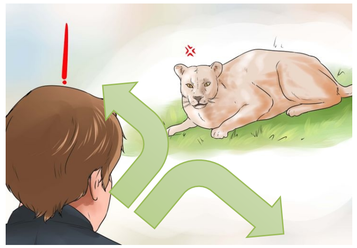
Terminology & notation



$\mathbf{o}_t$



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

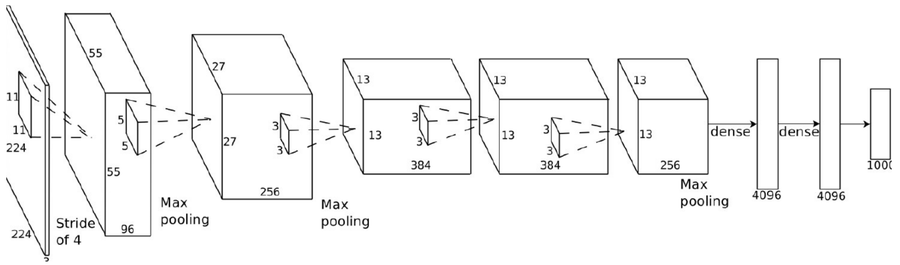


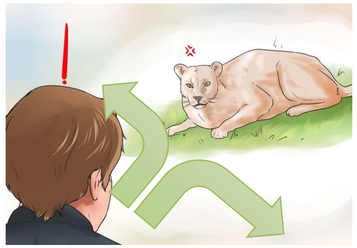
$\mathbf{a}_t$

Slide credit: Sergey Levine

Terminology & notation





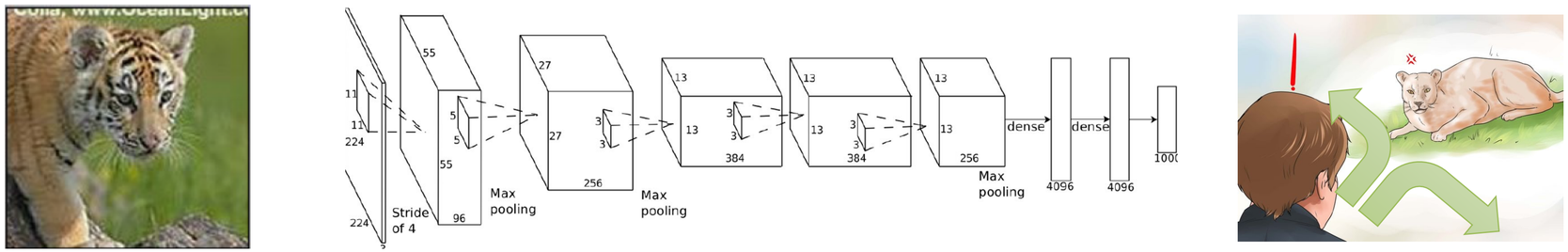


$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{o}_t$ – observation

Slide credit: Sergey Levine

Terminology & notation



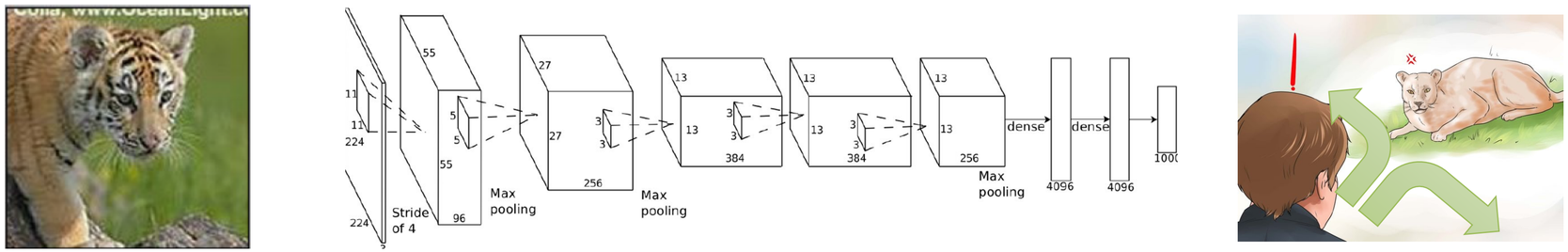
$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{o}_t$ – observation

$\mathbf{a}_t$ – action

Slide credit: Sergey Levine

Terminology & notation



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

$\mathbf{o}_t$ – observation

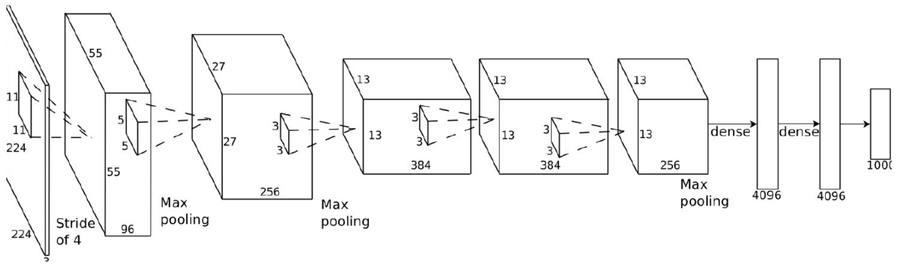
$\mathbf{a}_t$ – action

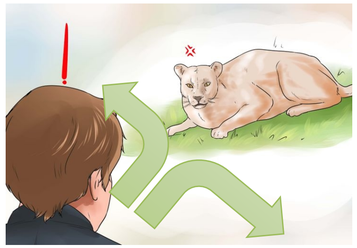
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy

Slide credit: Sergey Levine

Terminology & notation







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

$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation

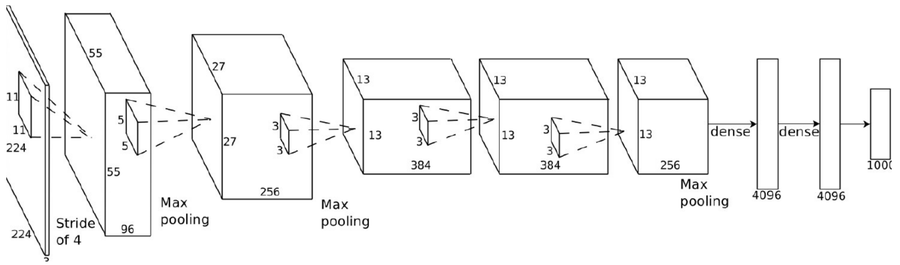
$\mathbf{a}_t$ – action

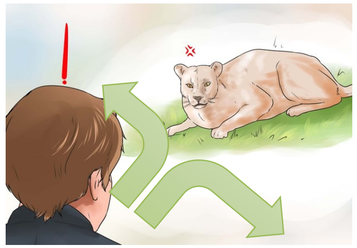
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy

Slide credit: Sergey Levine

Terminology & notation







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

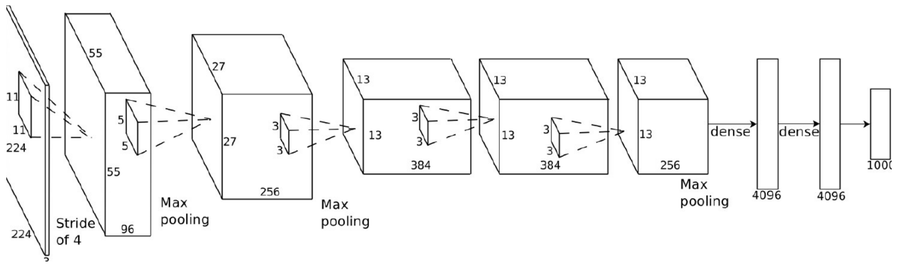
$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

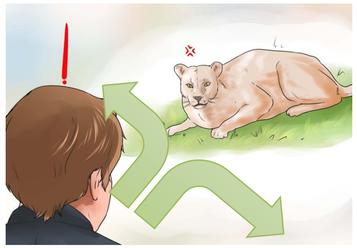
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Slide credit: Sergey Levine

Terminology & notation







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

$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation

$\mathbf{a}_t$ – action

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

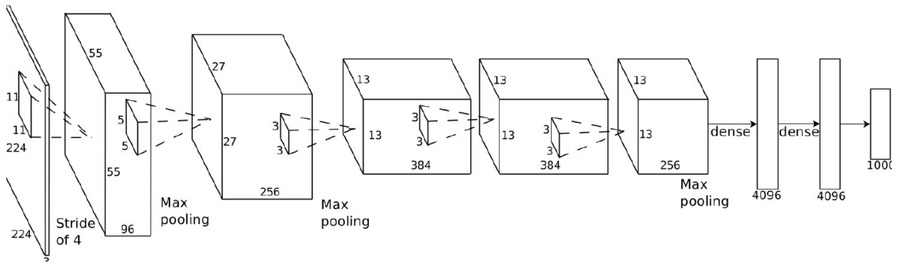
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

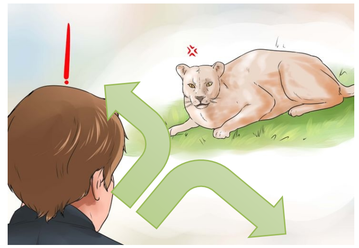


Slide credit: Sergey Levine

Terminology & notation







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

$\mathbf{s}_t$ – 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)



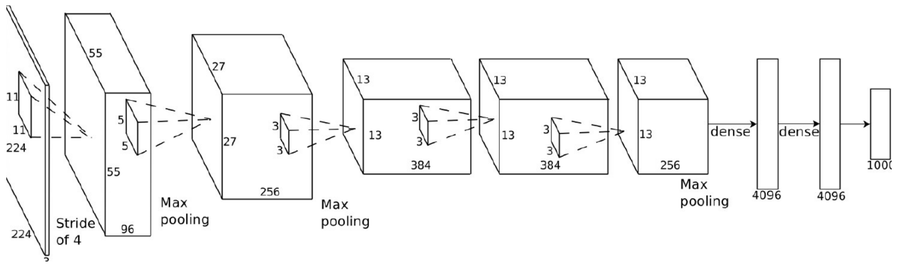
$\mathbf{o}_t$ – observation

Slide credit: Sergey Levine

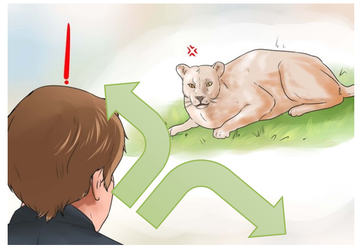
Terminology & notation



$\mathbf{o}_t$



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



$\mathbf{a}_t$

$\mathbf{s}_t$ – 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)



$\mathbf{o}_t$ – observation



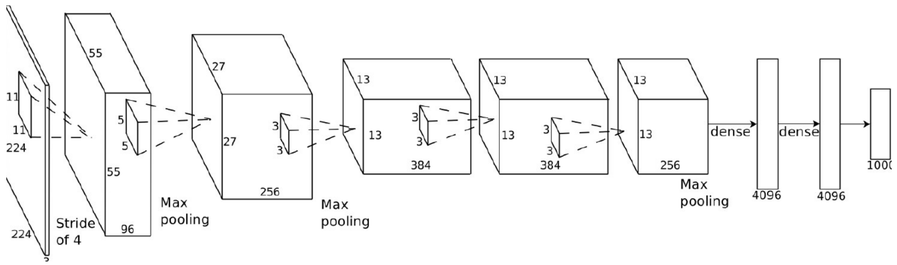
$\mathbf{s}_t$ – state

Slide credit: Sergey Levine

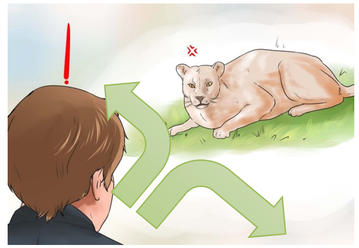
Terminology & notation



$\mathbf{o}_t$



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



$\mathbf{a}_t$

$\mathbf{s}_t$ – 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)



$\mathbf{o}_t$ – observation



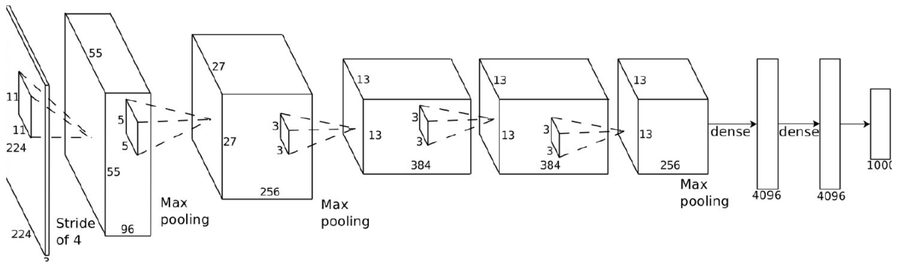
$\mathbf{s}_t$ – state

Slide credit: Sergey Levine

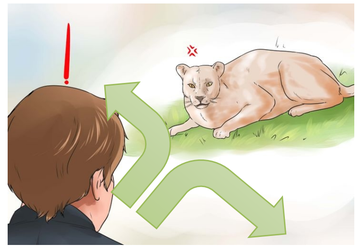
Terminology & notation



$\mathbf{o}_t$



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



$\mathbf{a}_t$

$\mathbf{s}_t$ – 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)



$\mathbf{o}_t$ – observation

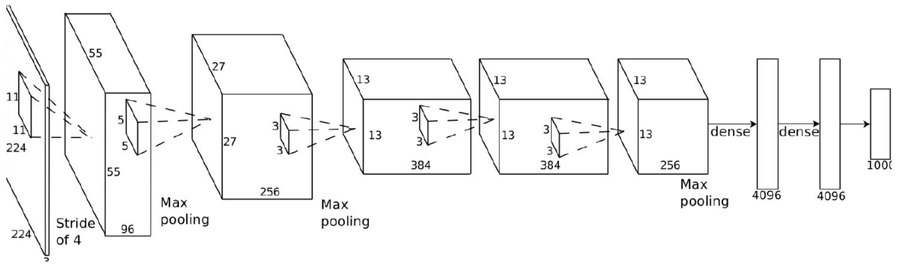


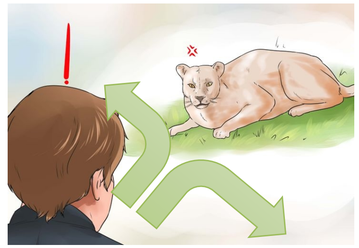
$\mathbf{s}_t$ – state

Slide credit: Sergey Levine

Terminology & notation







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

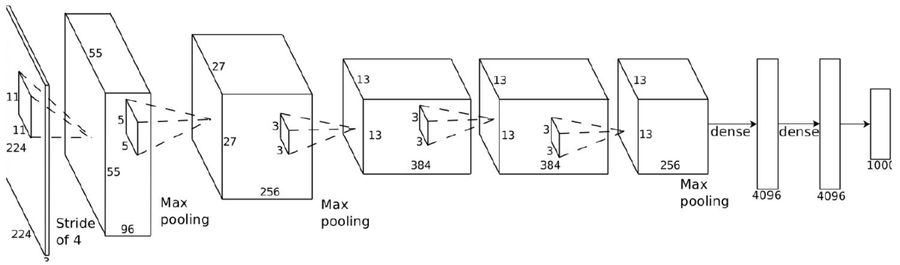
$\mathbf{s}_t$ – state
$\mathbf{o}_t$ – observation
$\mathbf{a}_t$ – action

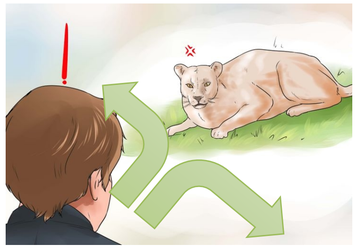
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

Slide credit: Sergey Levine

Terminology & notation



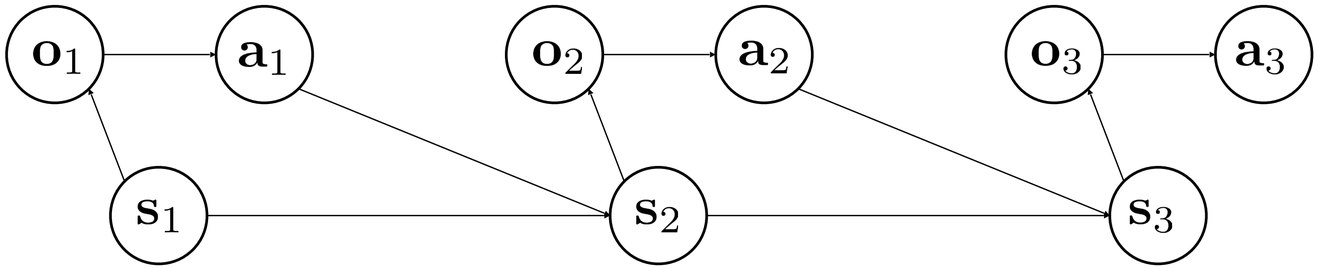




$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

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

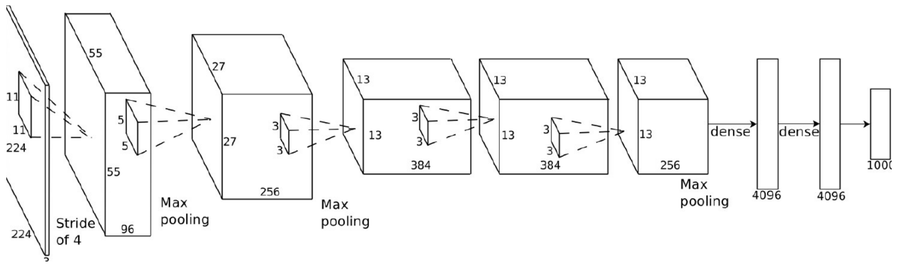
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

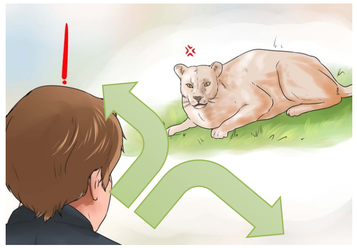


Slide credit: Sergey Levine

Terminology & notation



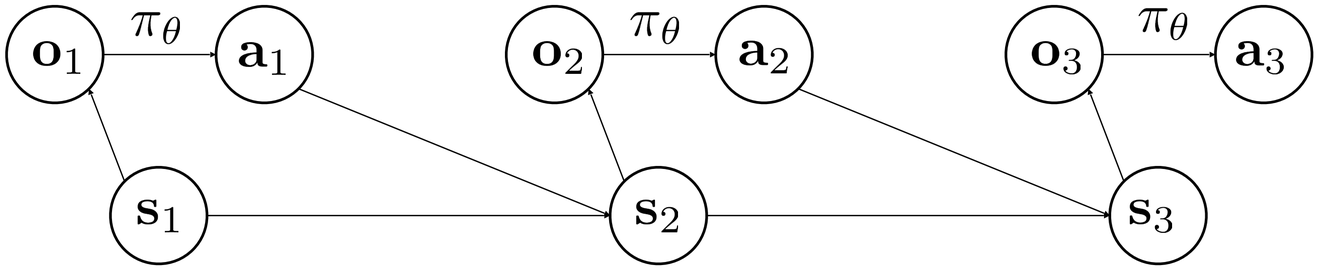




$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

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

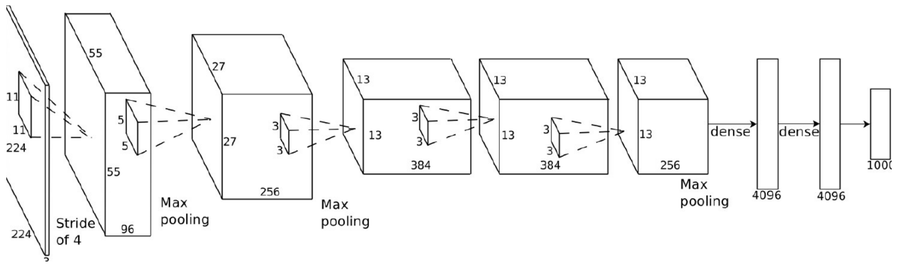
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

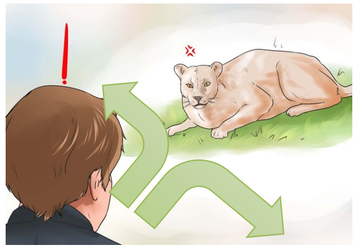


Slide credit: Sergey Levine

Terminology & notation







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

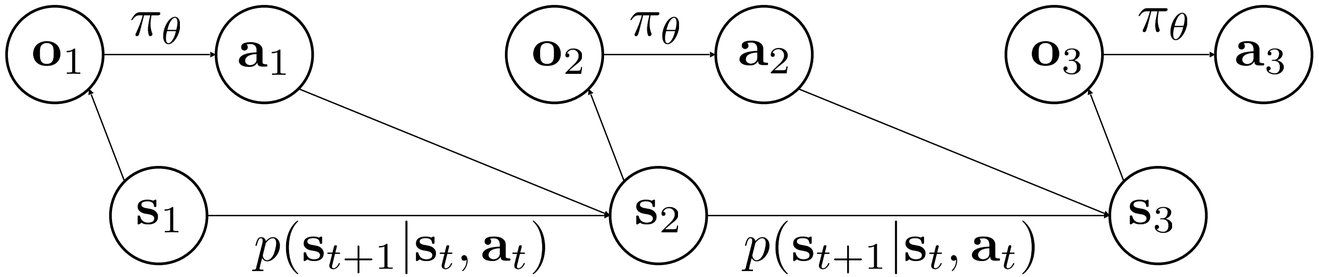
$\mathbf{s}_t$ – state

$\mathbf{o}_t$ – observation

$\mathbf{a}_t$ – action

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

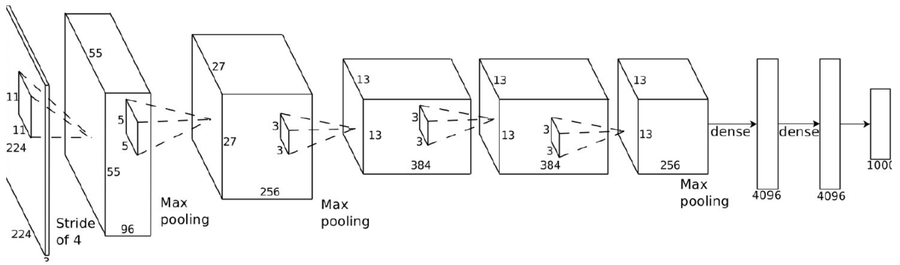
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

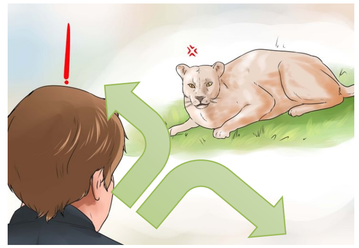


Slide credit: Sergey Levine

Terminology & notation



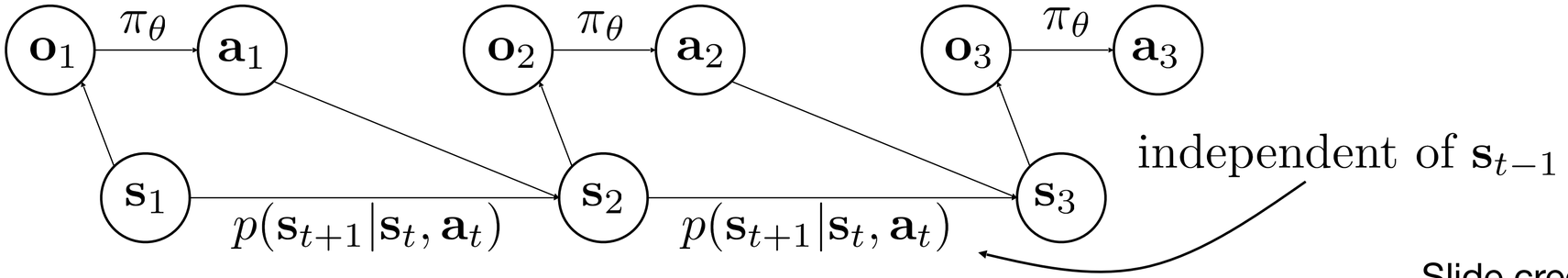




$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

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

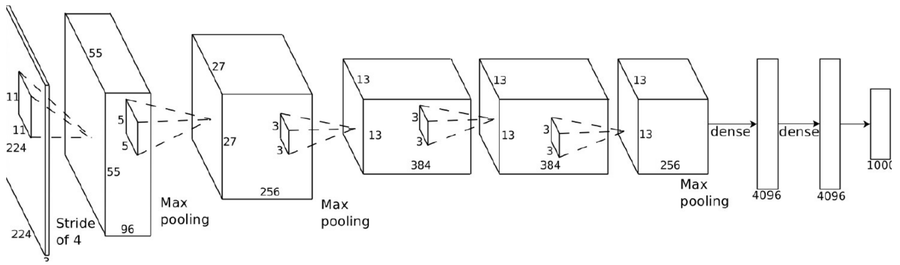
$\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ – policy
$\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ – policy (fully observed)

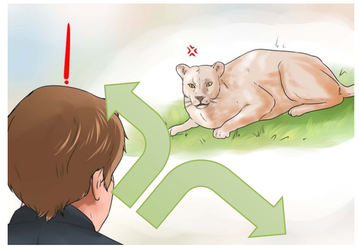


Slide credit: Sergey Levine

Terminology & notation



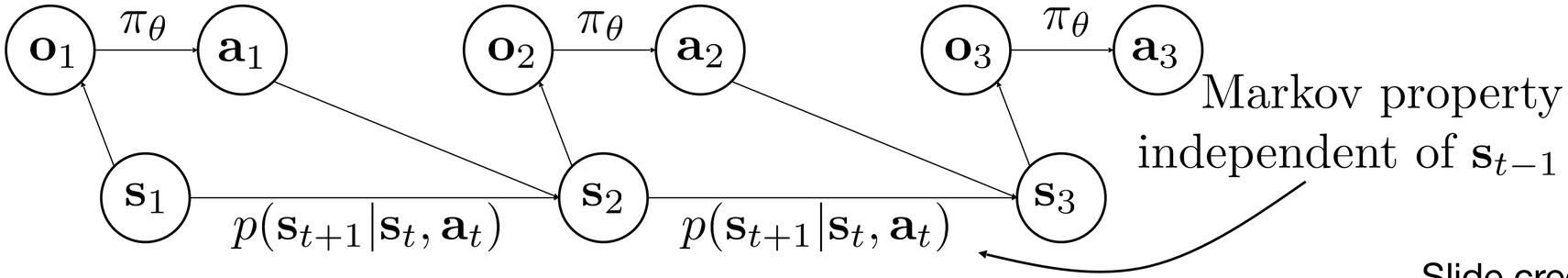




$\mathbf{o}_t$ $\pi_\theta(\mathbf{a}_t|\mathbf{o}_t)$ $\mathbf{a}_t$

$\mathbf{s}_t$ – 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)

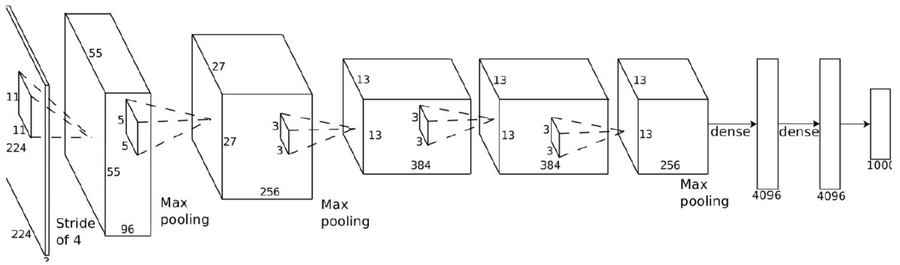


Slide credit: Sergey Levine

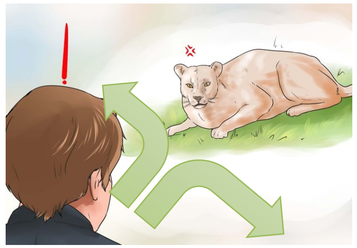
Imitation Learning



$\mathbf{o}_t$



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



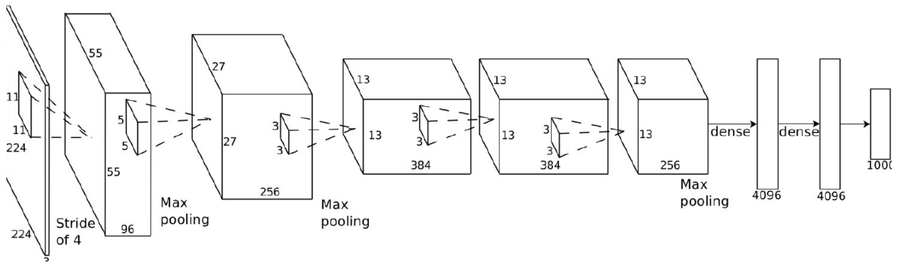
$\mathbf{a}_t$

Slide credit: Sergey Levine

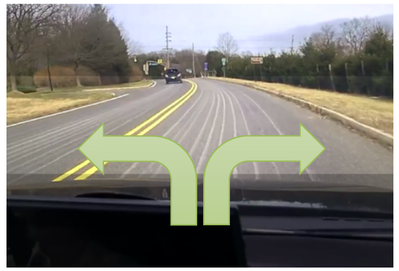
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{a}_t$

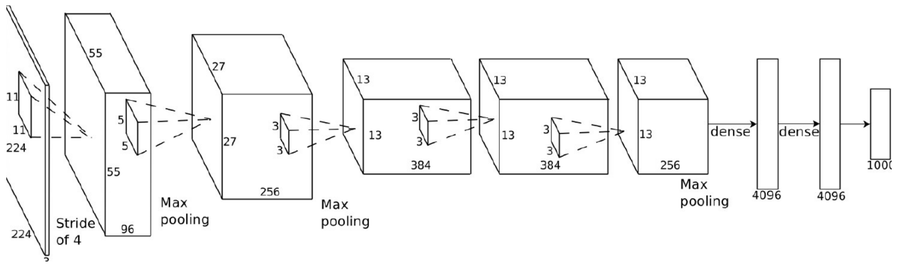
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

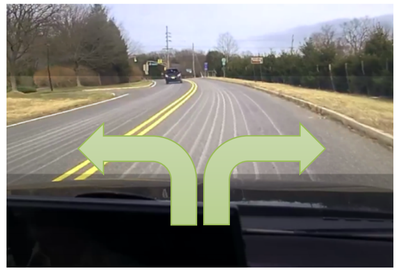
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{a}_t$



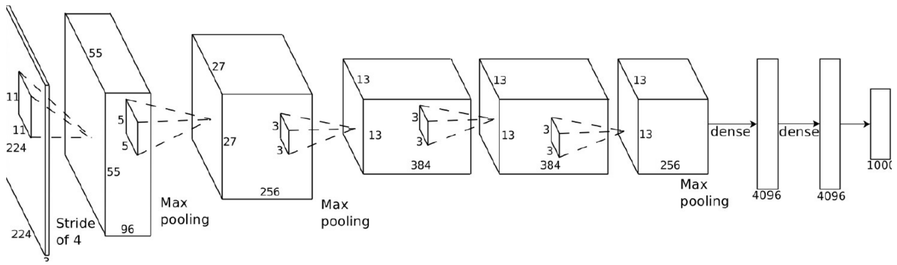
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

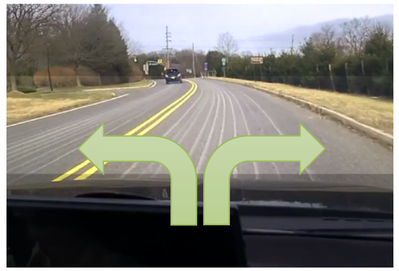
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{a}_t$



$\mathbf{o}_t$

$\mathbf{a}_t$

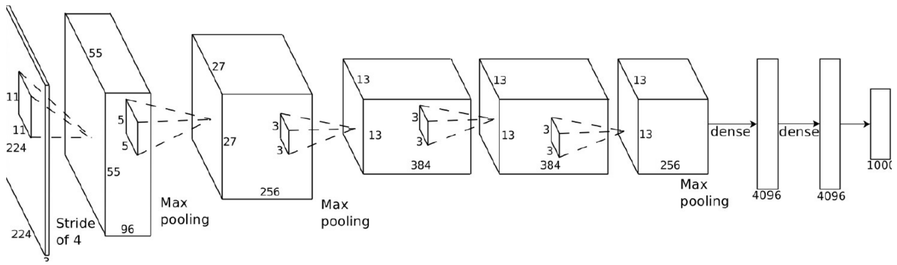
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

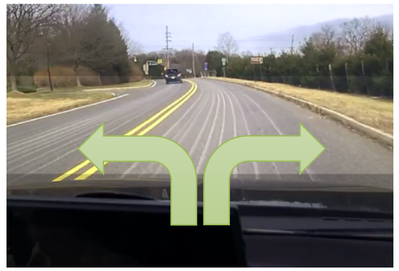
Imitation Learning



$\mathbf{o}_t$



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



$\mathbf{a}_t$



$\mathbf{o}_t$

$\mathbf{a}_t$



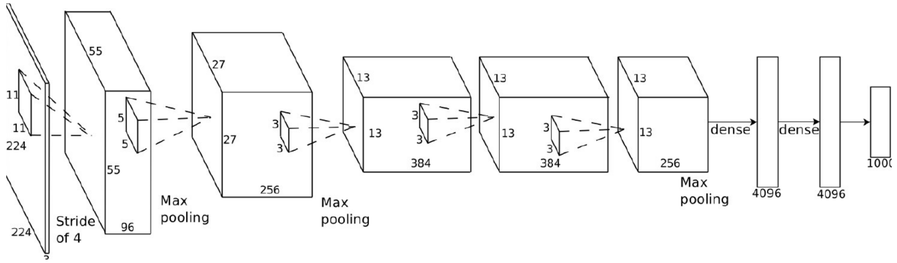
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

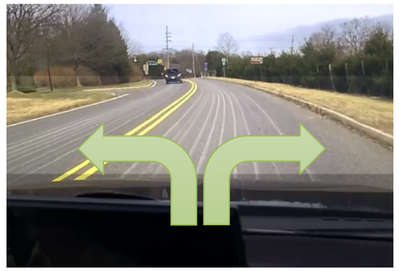
Imitation Learning



$\mathbf{o}_t$



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

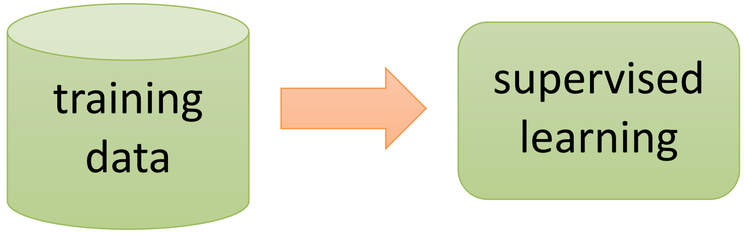


$\mathbf{a}_t$



$\mathbf{o}_t$

$\mathbf{a}_t$



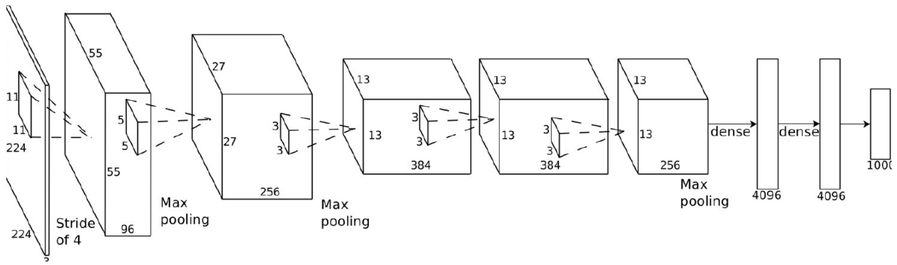
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

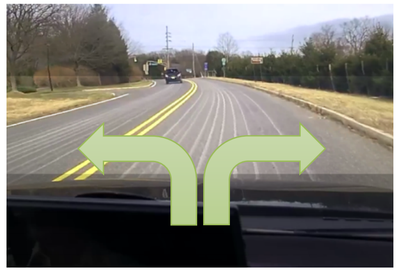
Imitation Learning



$\mathbf{o}_t$



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

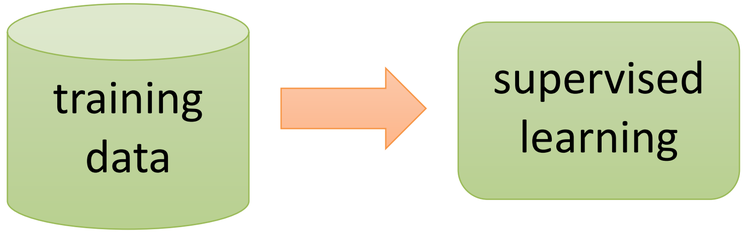


$\mathbf{a}_t$



$\mathbf{o}_t$

$\mathbf{a}_t$



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

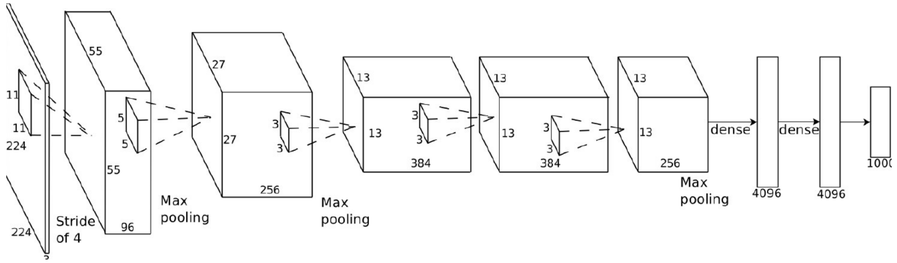
Images: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

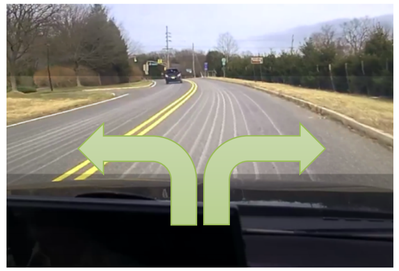
Imitation Learning



$\mathbf{o}_t$



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

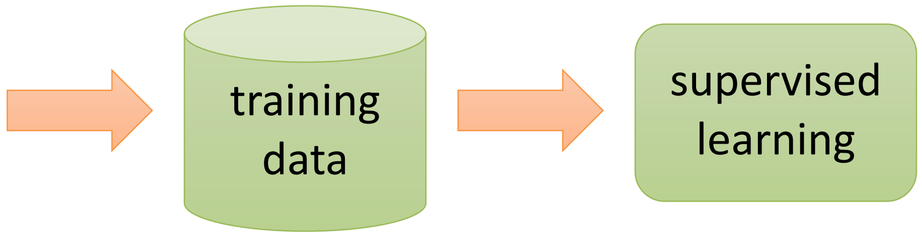


$\mathbf{a}_t$



$\mathbf{o}_t$

$\mathbf{a}_t$



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

behavioral cloning

Images: Bojarski et al. ‘16, NVIDIA

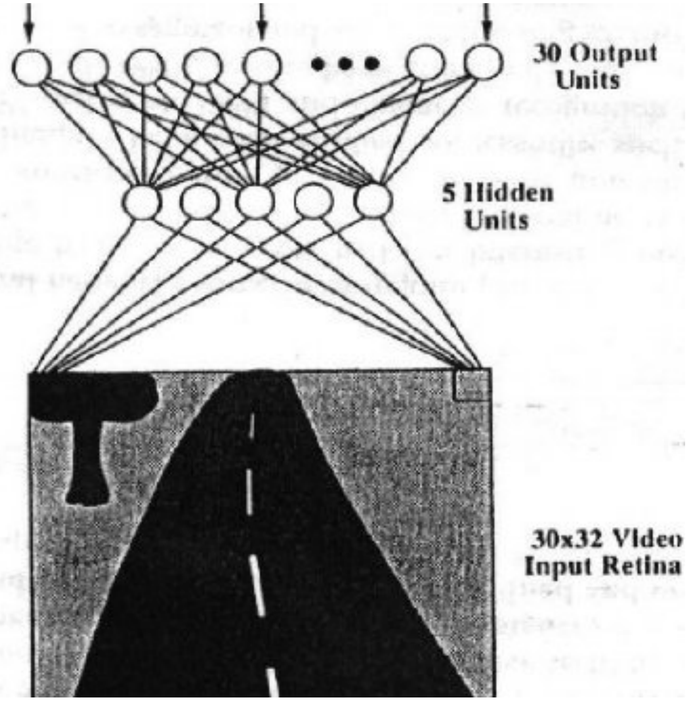
Slide credit: Sergey Levine

The original deep imitation learning system

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork

1989









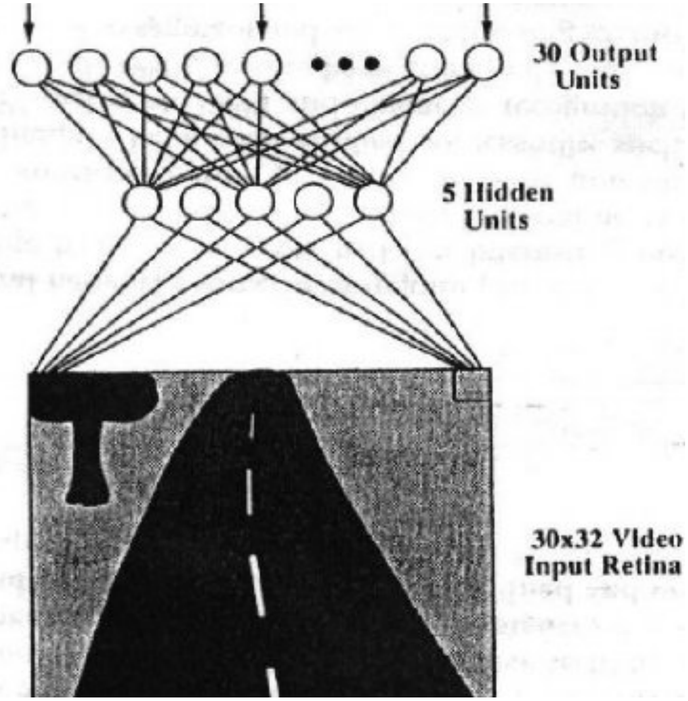
Slide credit: Sergey Levine

The original deep imitation learning system

ALVINN: **A**utonomous **L**and **V**ehicle **I**n a **N**eural **N**etwork

1989









Slide credit: Sergey Levine

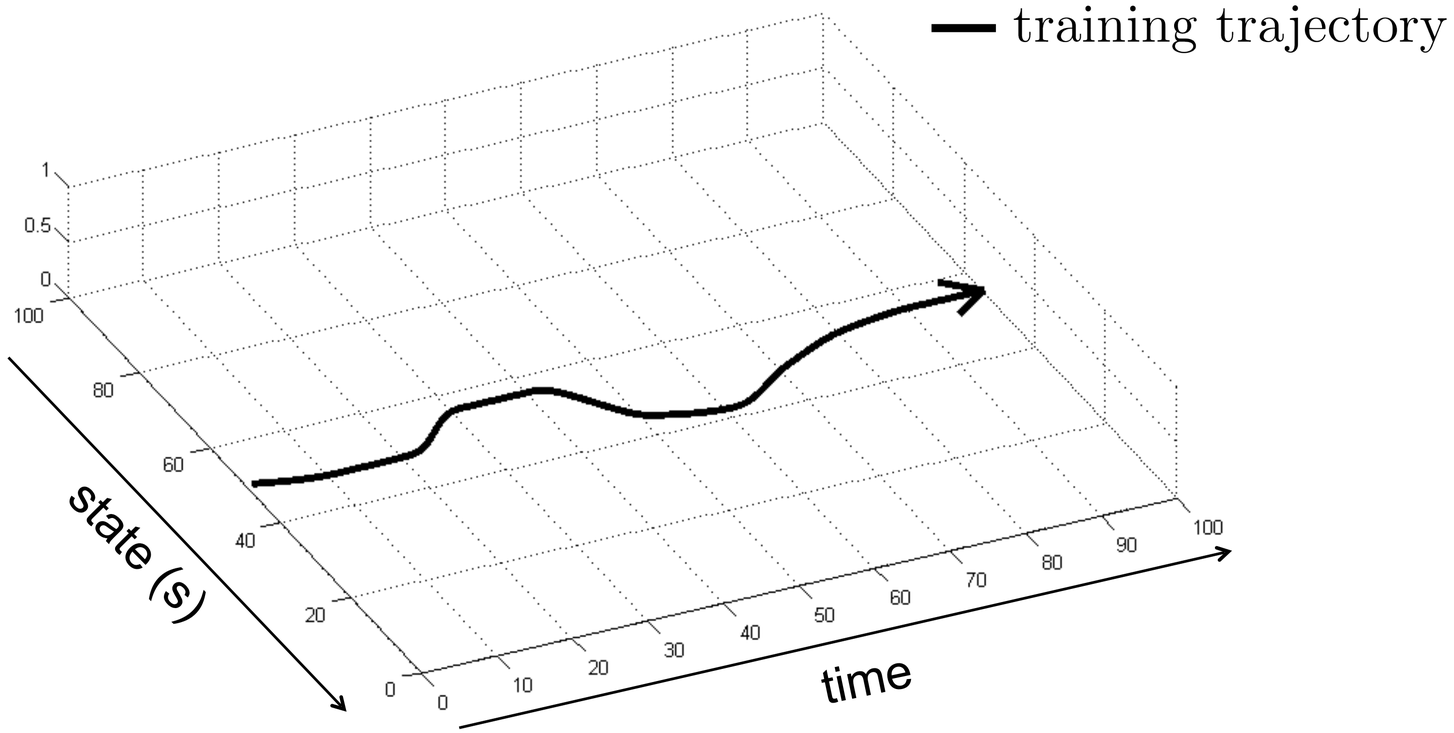
Does it work?

No!

Slide credit: Sergey Levine

Does it work?

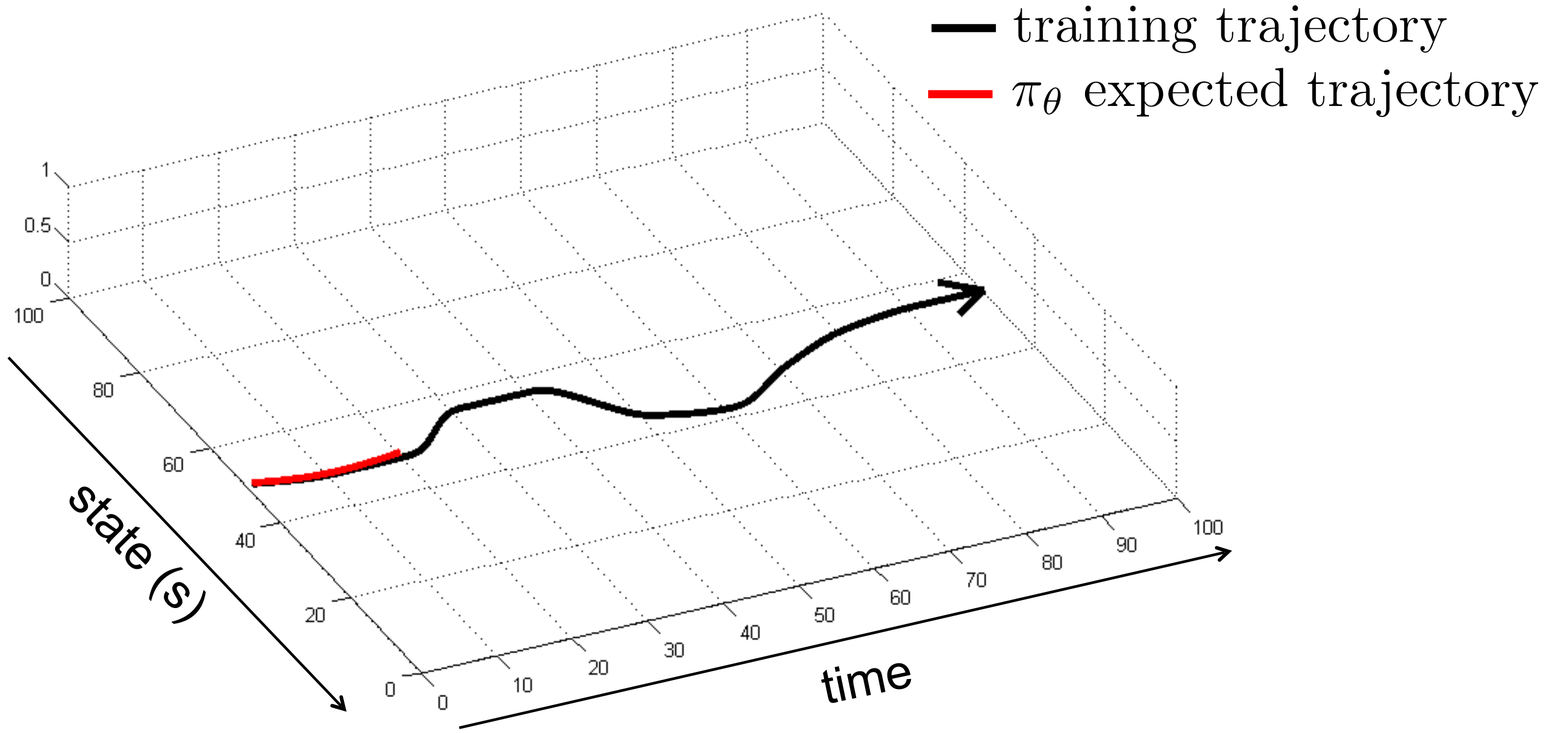
No!



Slide credit: Sergey Levine

Does it work?

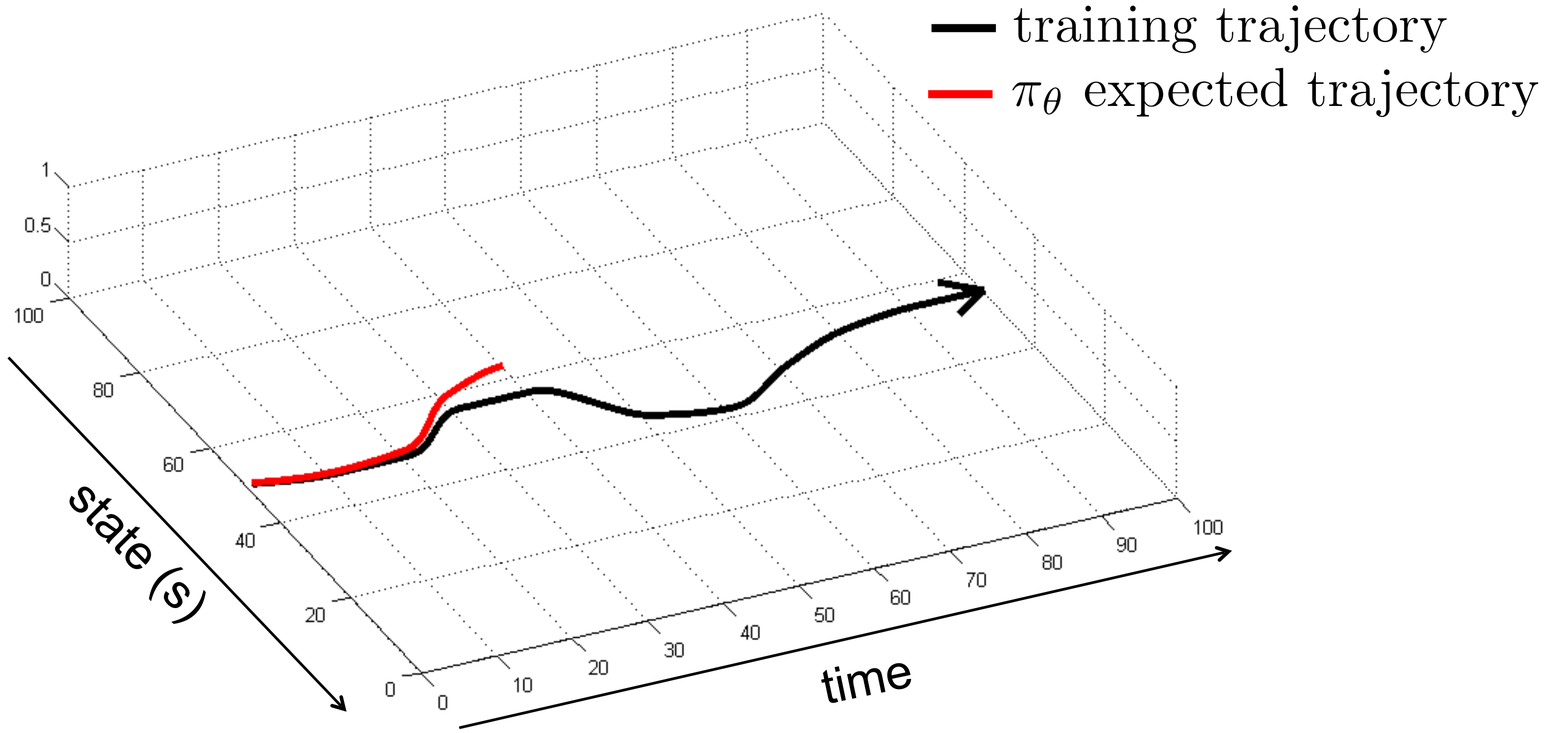
No!



Slide credit: Sergey Levine

Does it work?

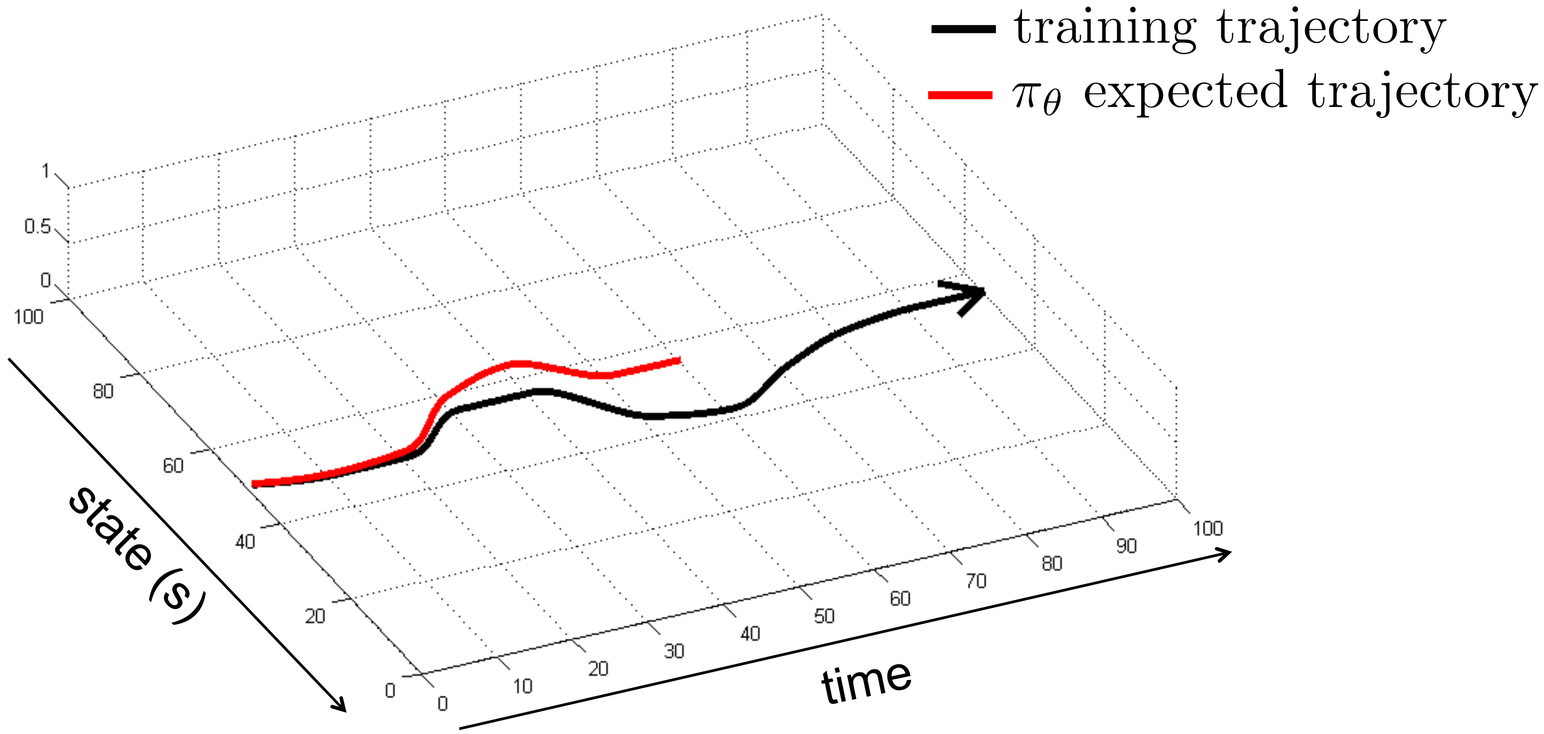
No!



Slide credit: Sergey Levine

Does it work?

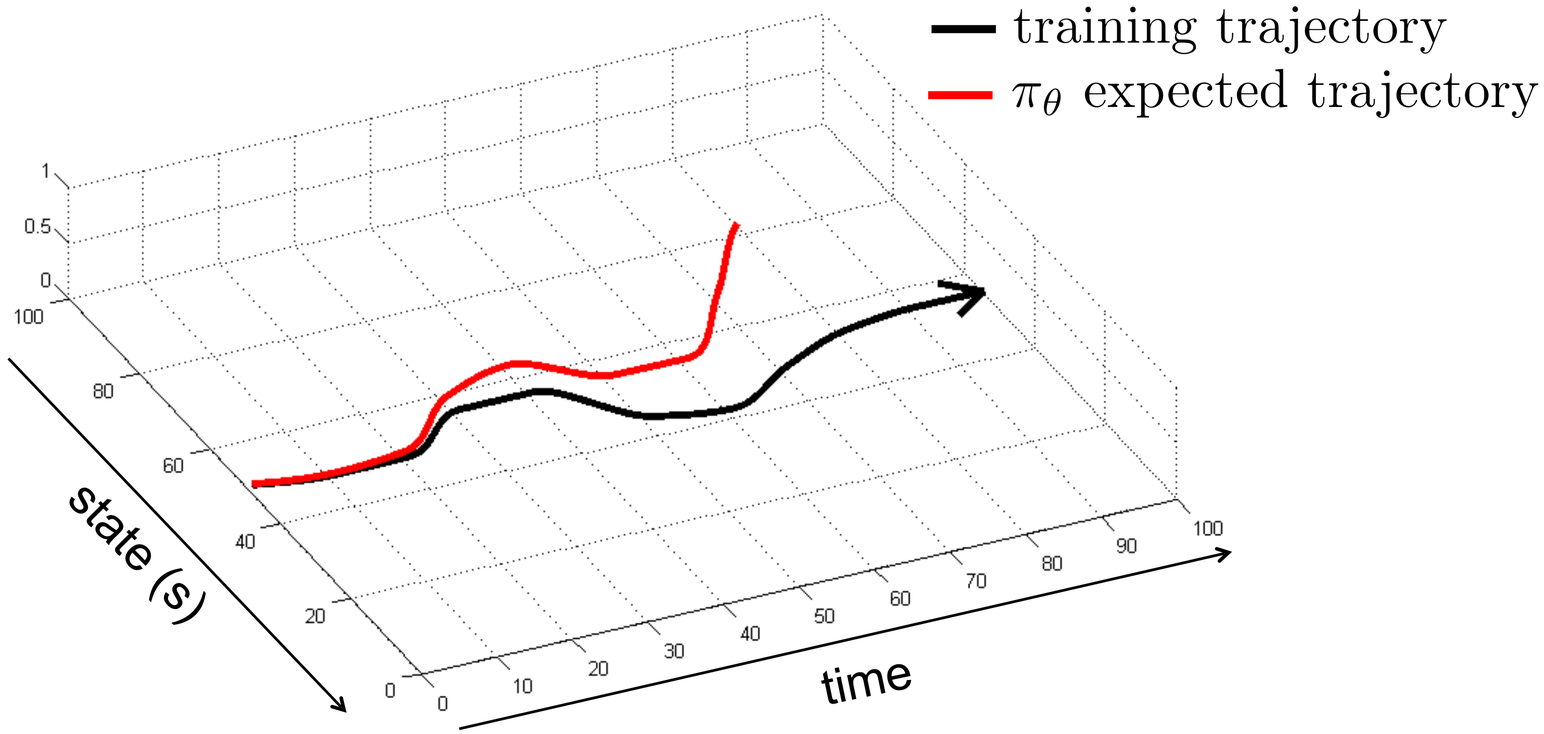
No!



Slide credit: Sergey Levine

Does it work?

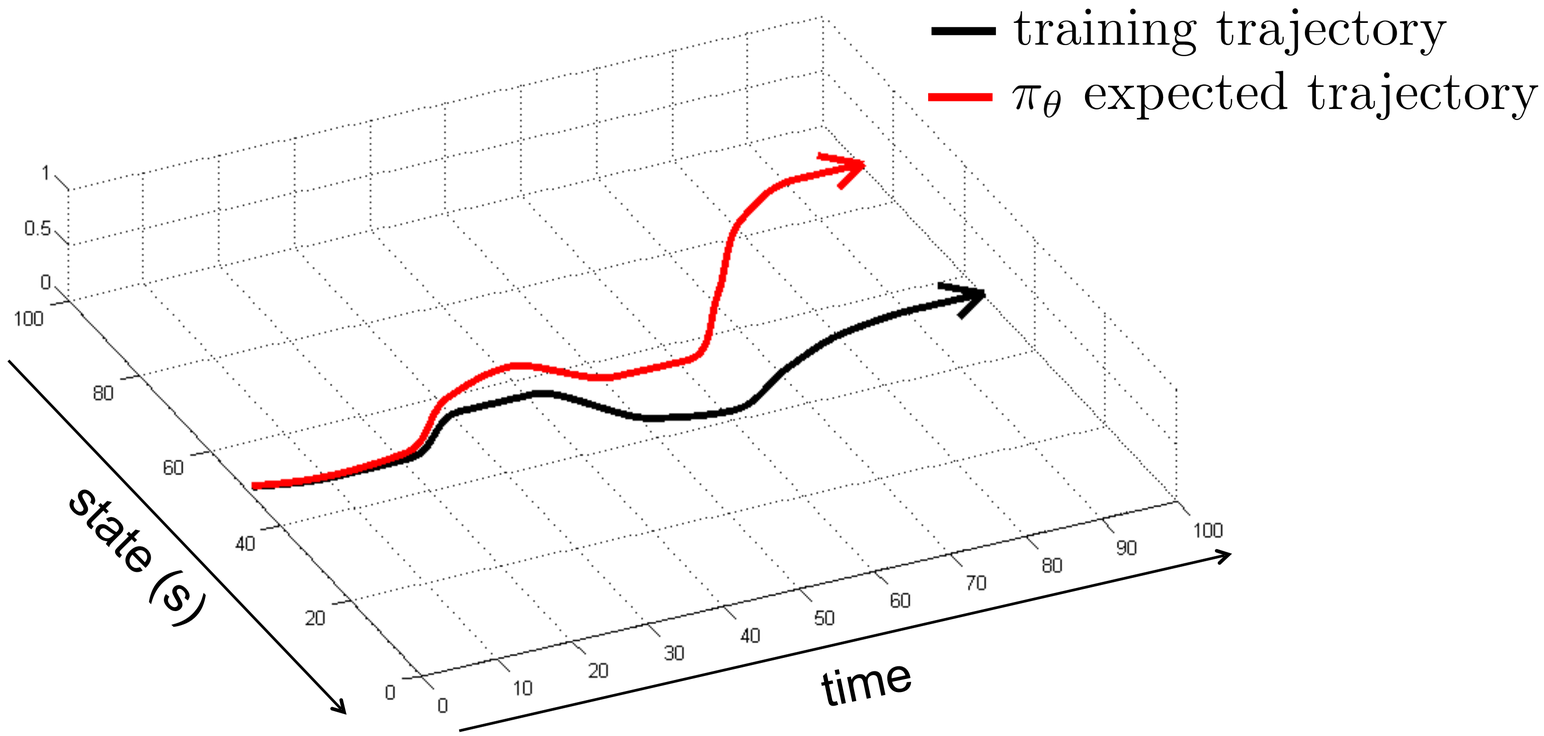
No!



Slide credit: Sergey Levine

Does it work?

No!



Slide credit: Sergey Levine

Does it work? Yes!



Video: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

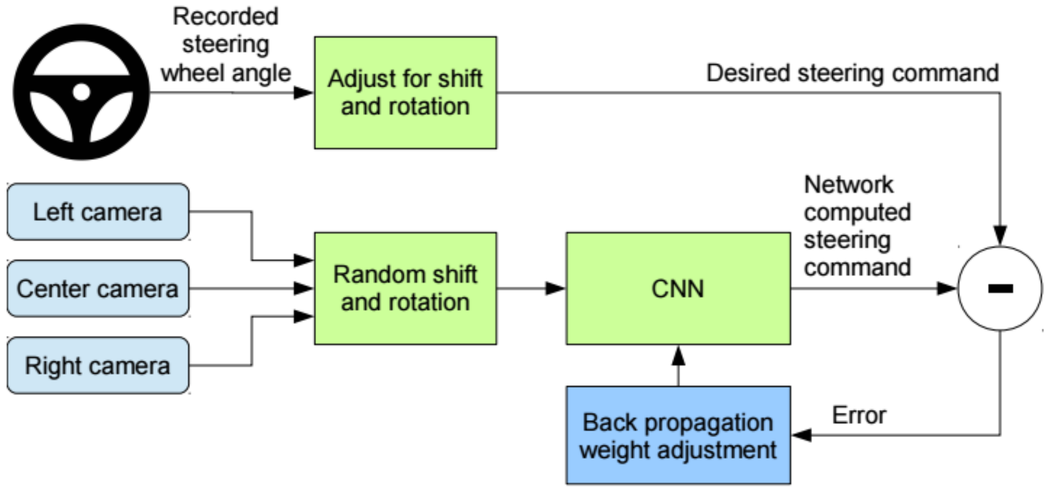
Does it work? Yes!



Video: Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

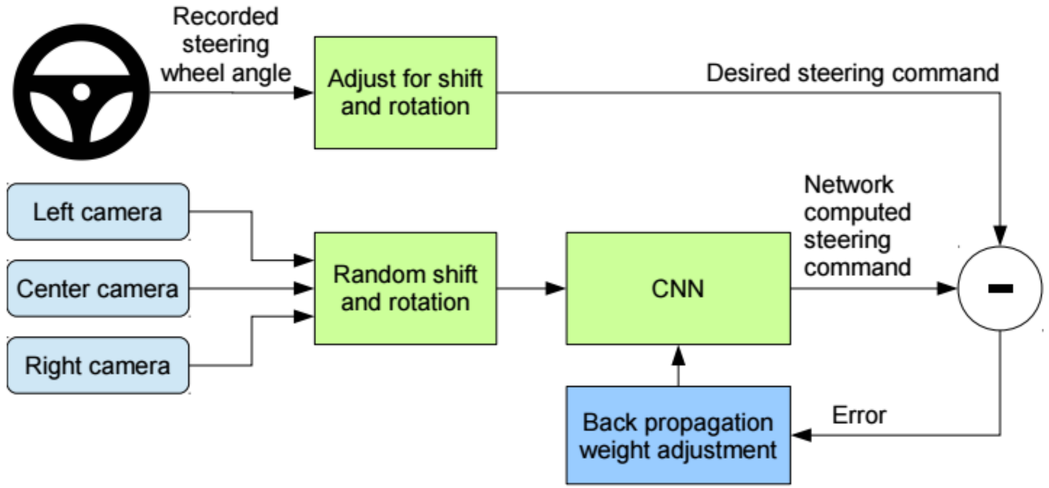
Why did that work?



Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

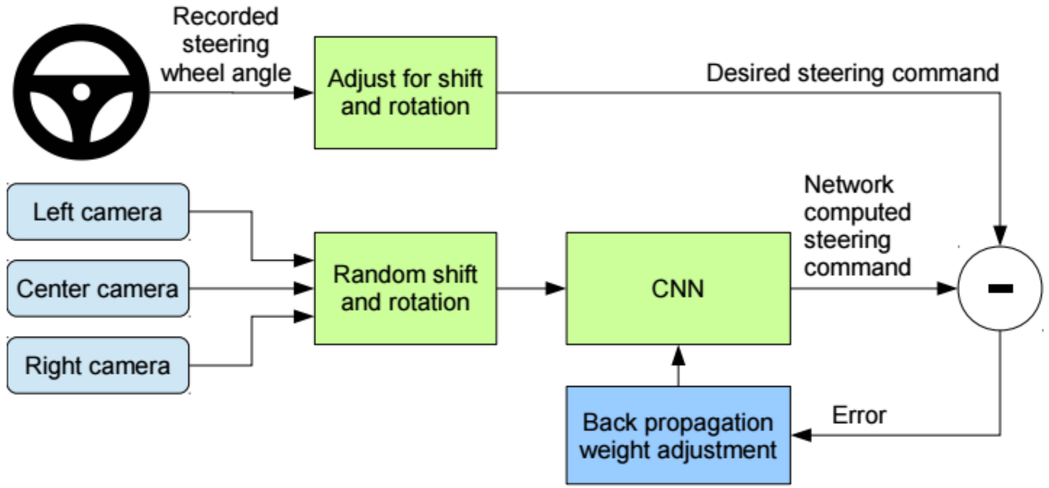




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

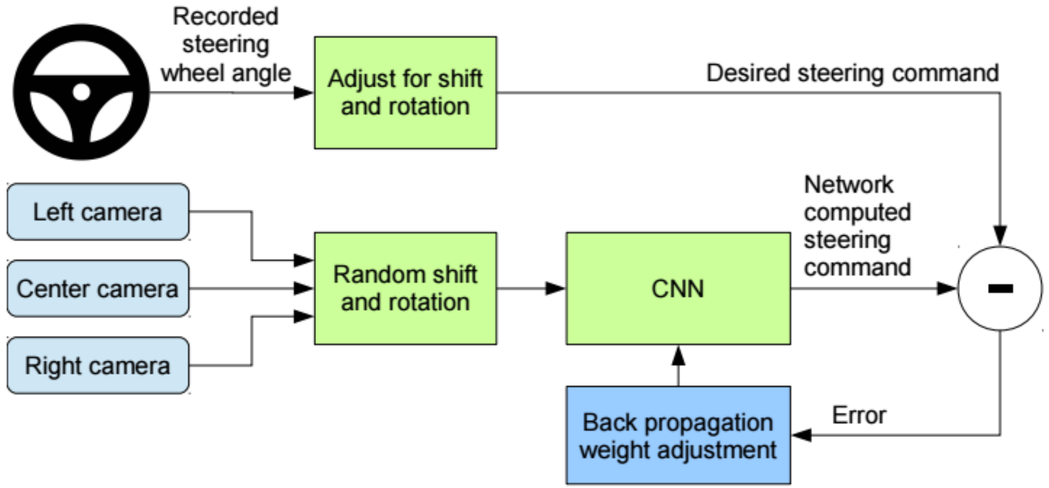




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

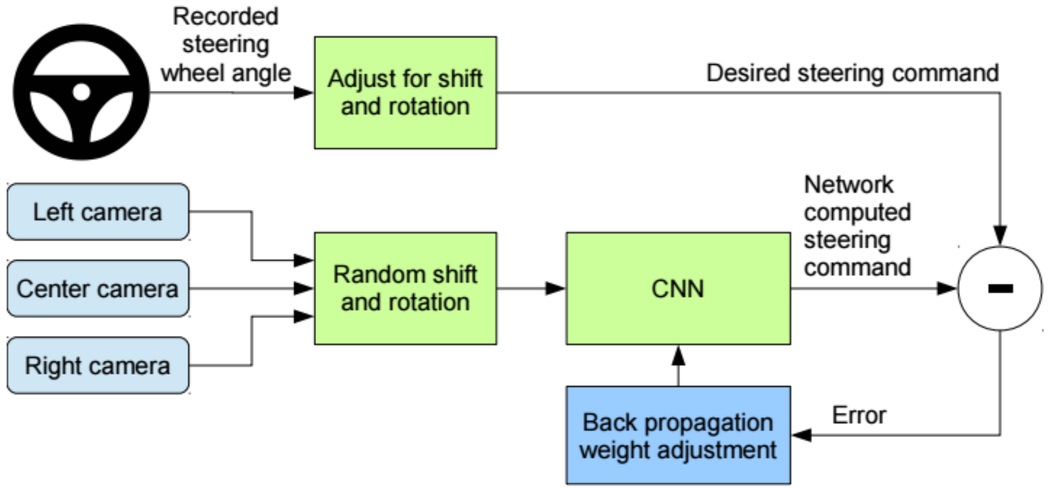




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

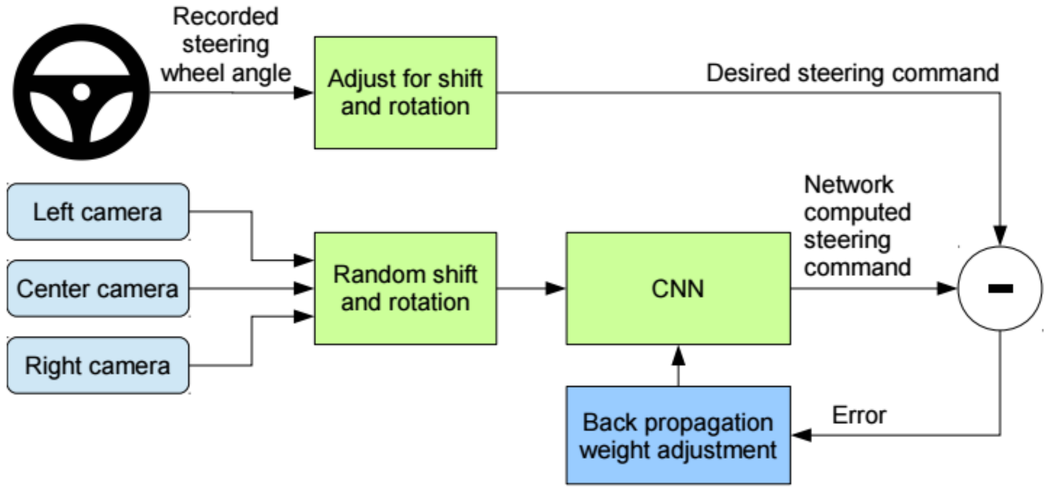




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

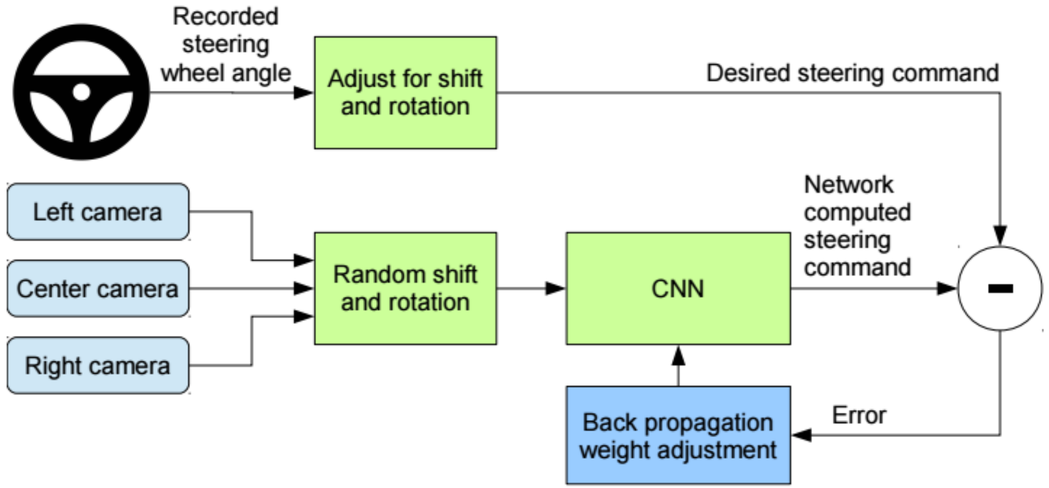




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?

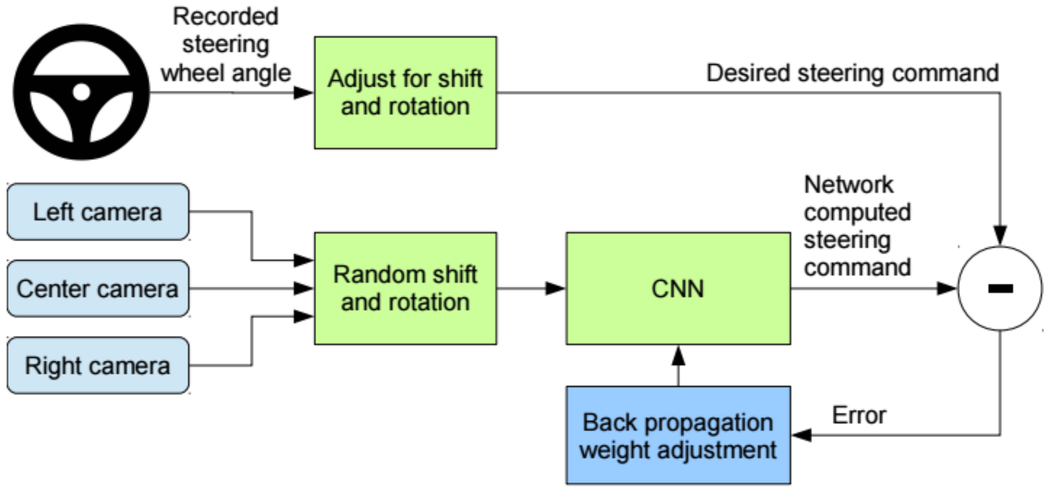




Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine

Why did that work?





Bojarski et al. ‘16, NVIDIA

Slide credit: Sergey Levine



Vincent Vanhoucke

Oct 27 · 4 min read · Member-only · Listen

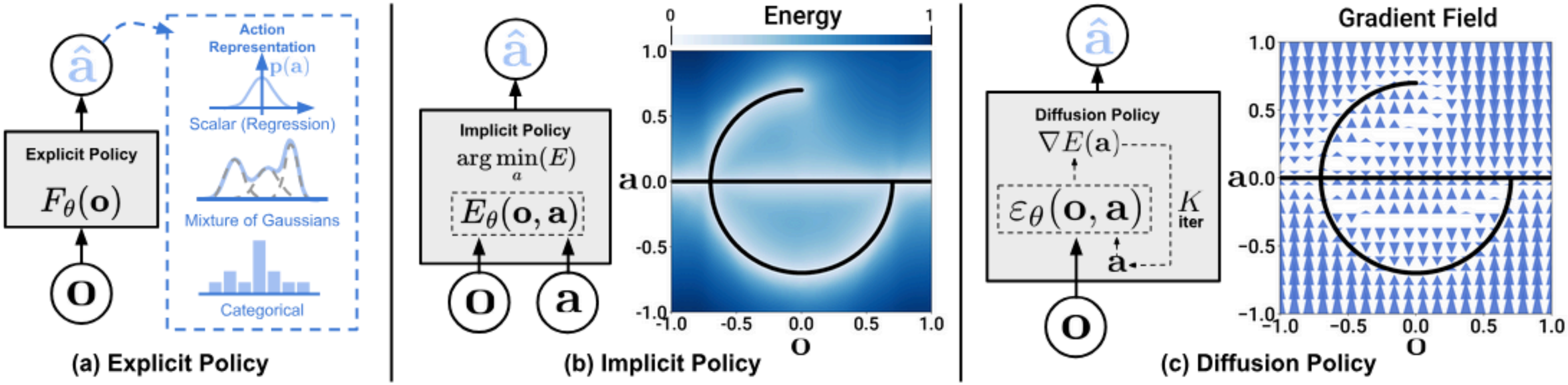
Shifting Winds in Robot Learning Research

Reinforcement Learning is dead, long live Reinforcement Learning!

"And then ... BC methods started to get good. Really good. So good that our best manipulation system today mostly uses BC, with a sprinkle of Q learning on top to perform high-level action selection. Today, less than 20% of our research investments is on RL, and the research runway for BC-based methods feels more robust."

Slide credit: Russ Tedrake

Case Study: Diffusion Policy



Diffusion Policy, Visuomotor Policy Learning via Action Diffusion, Chi et al. 2023

Reminder: Diffusion

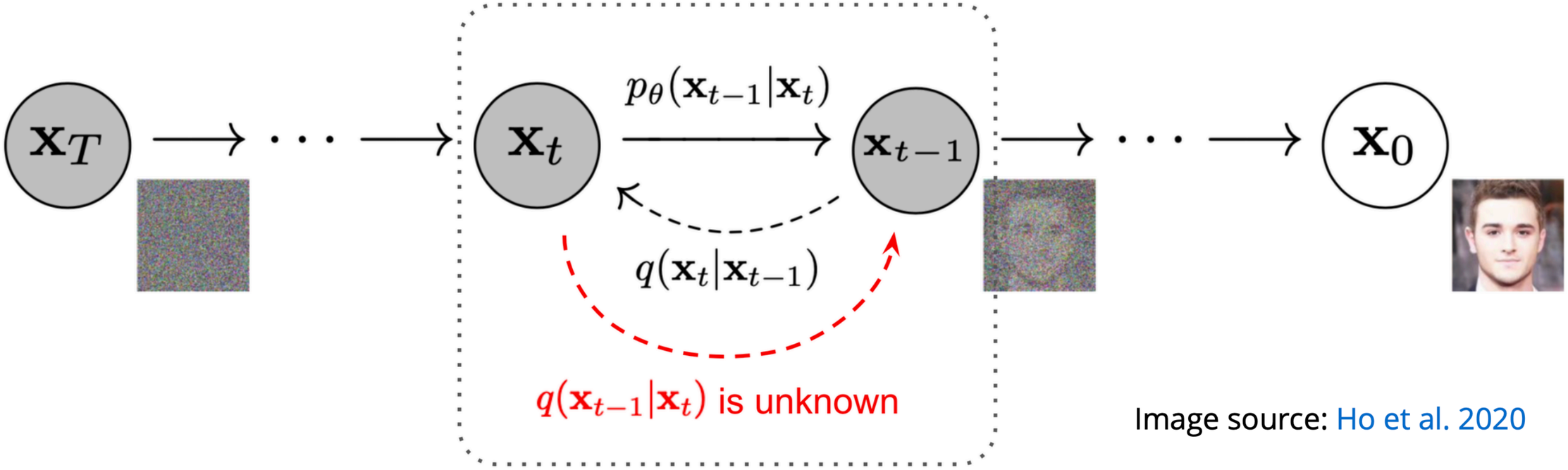
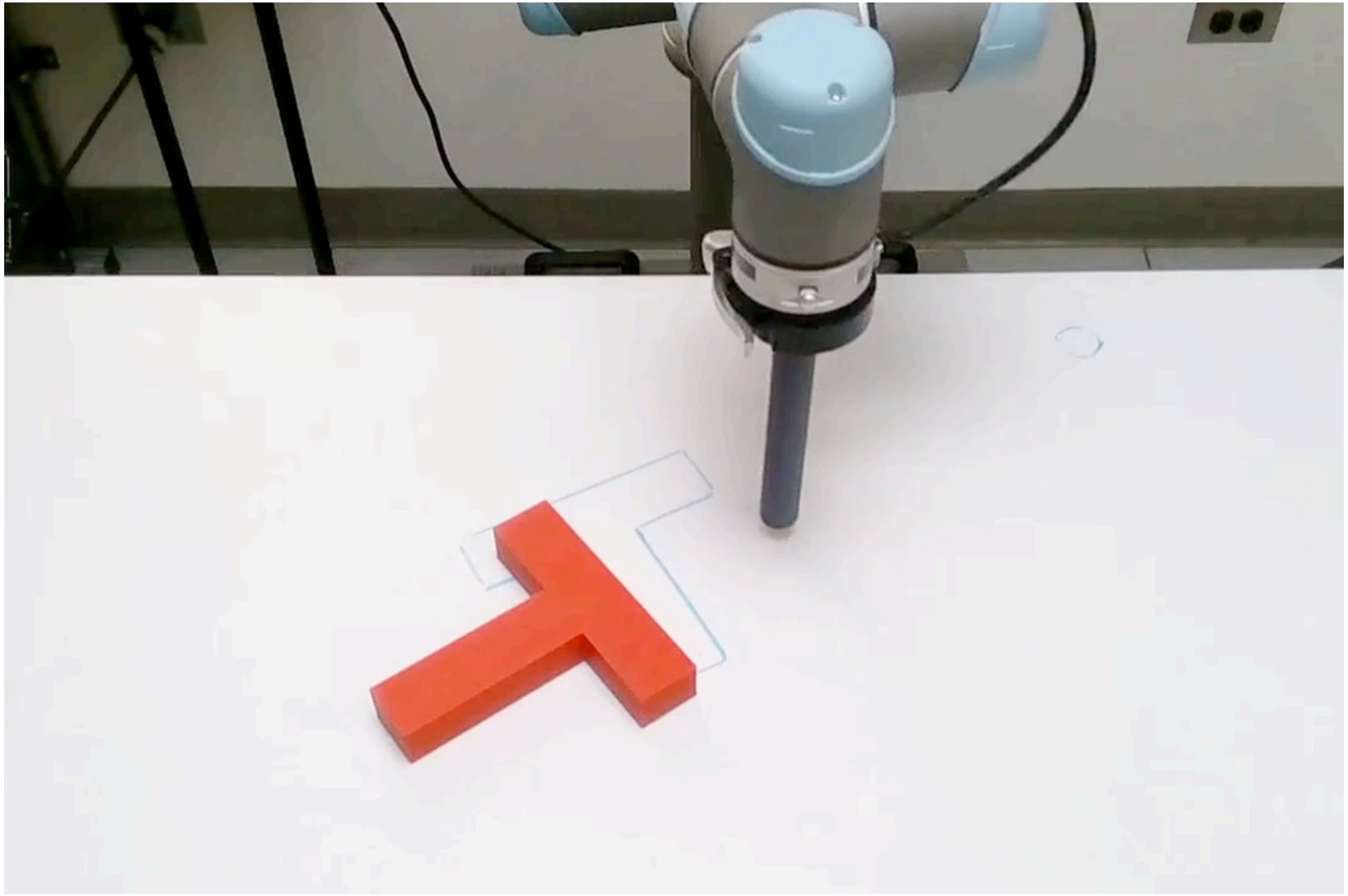
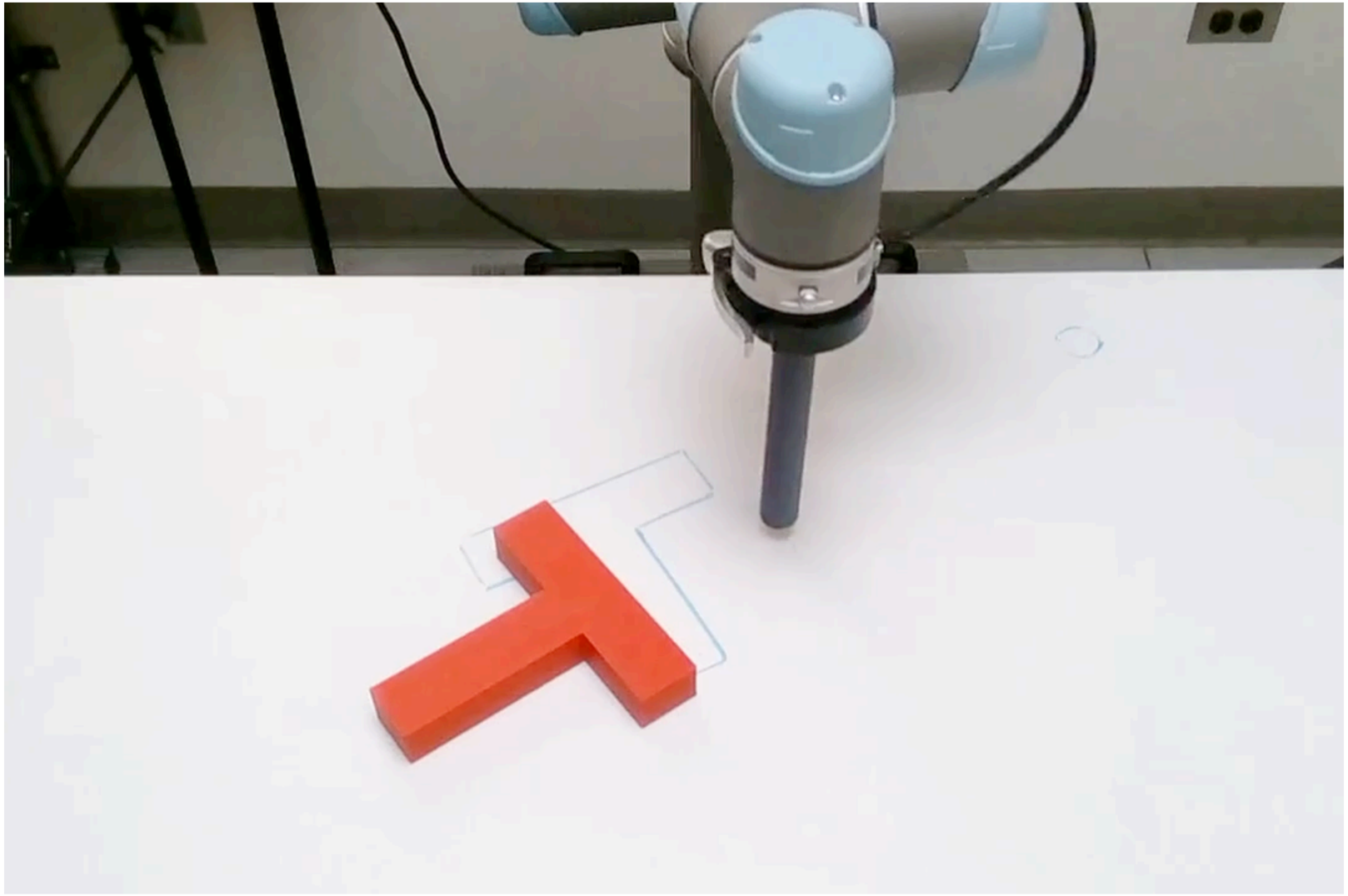


Image source: Ho et al. 2020

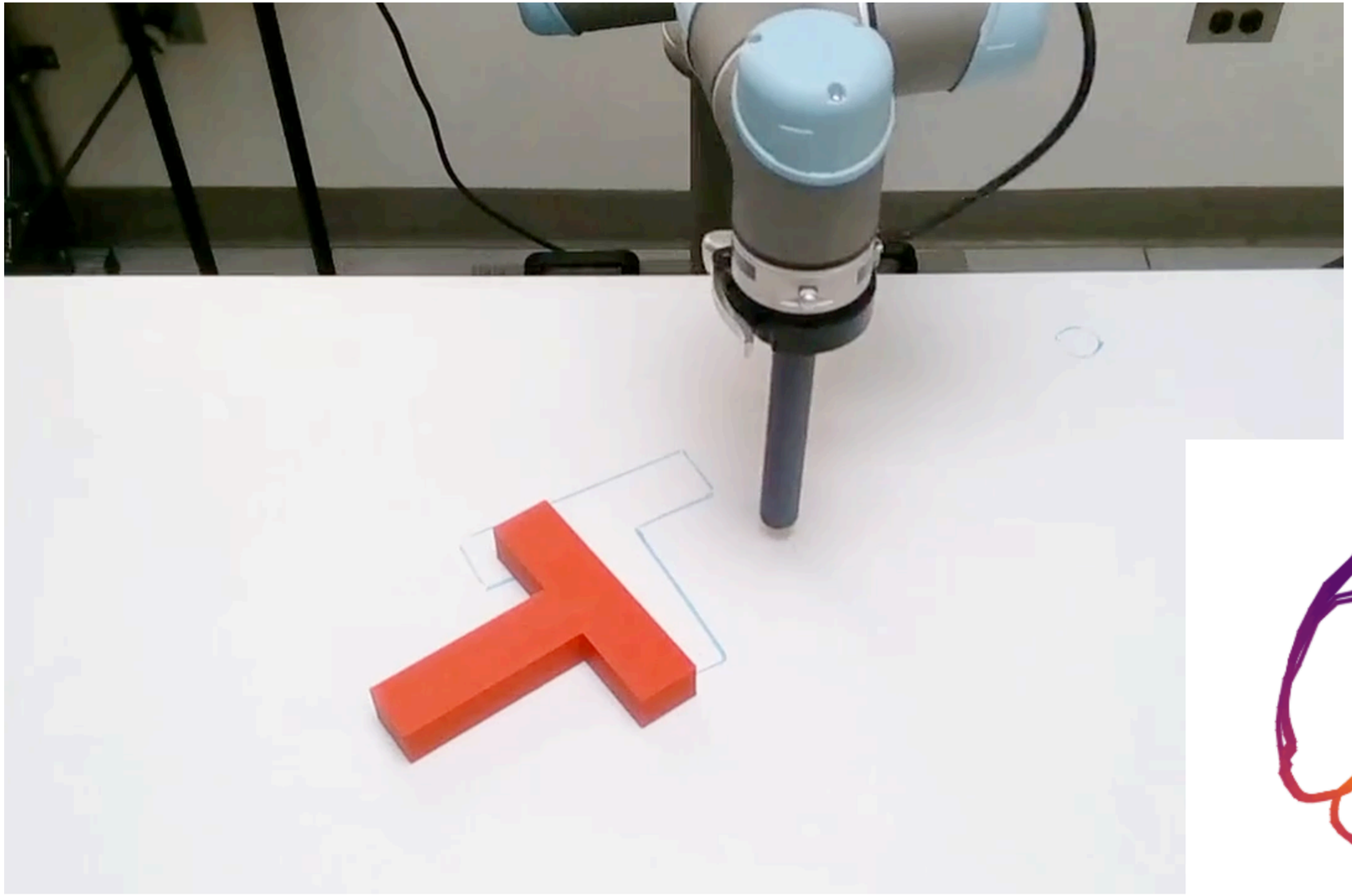
Denoiser can be conditioned on additional inputs, u: $p_\theta(x_{t-1}|x_t, u)$

Slide credit: Russ Tedrake

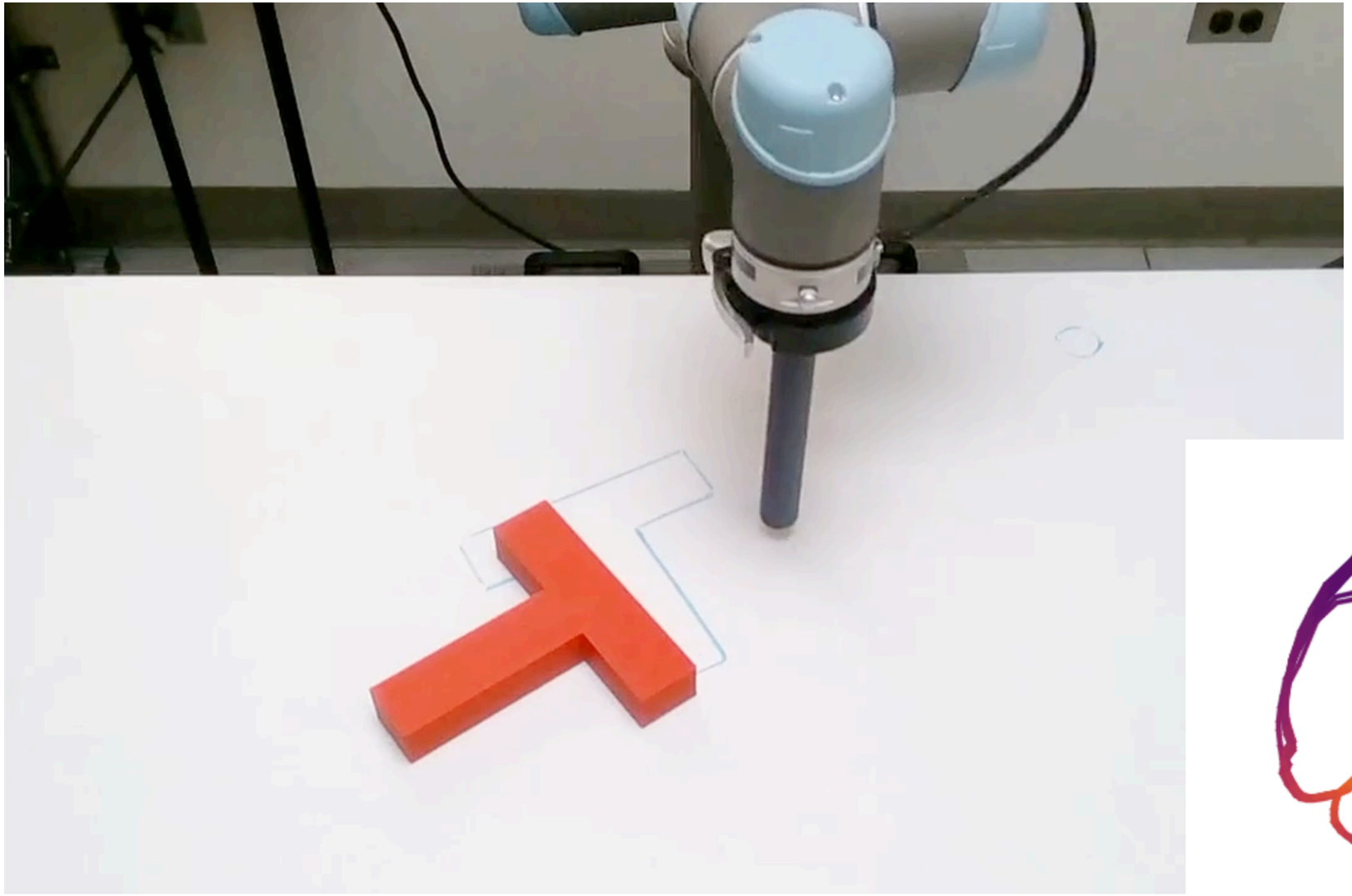




Naturally deals with multi-modal demonstrations



Naturally deals with multi-modal demonstrations



Modeling Sequences



"Diffusion Policy" is an auto-regressive (ARX) model with *forecasting*

$$[y_{n+1}, \ldots, y_{n+P}] = f_\theta(u_n, \ldots, u_{n-H}, y_n, \ldots, y_{n-H})$$

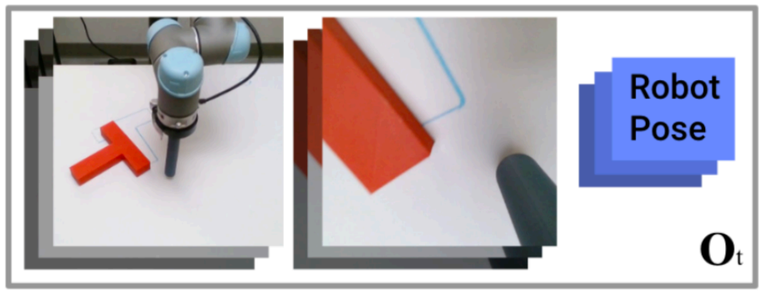
H is the length of the history,

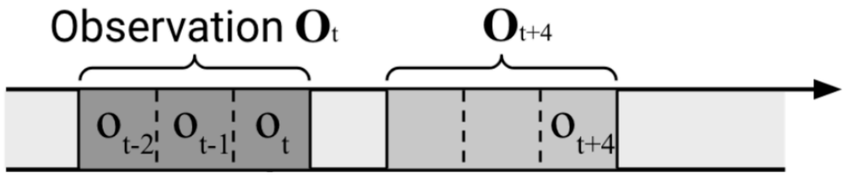
P is the length of the prediction

Conditional denoiser produces the forecast, conditional on the history

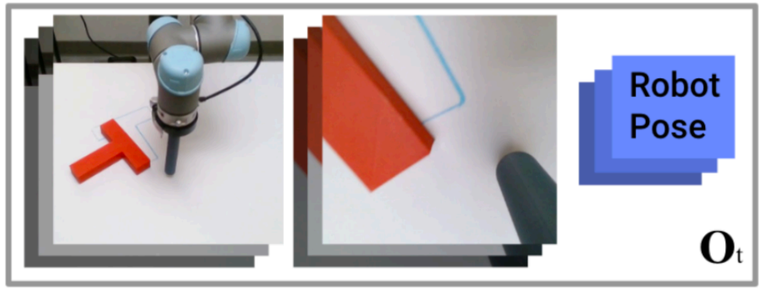
Slide credit: Russ Tedrake

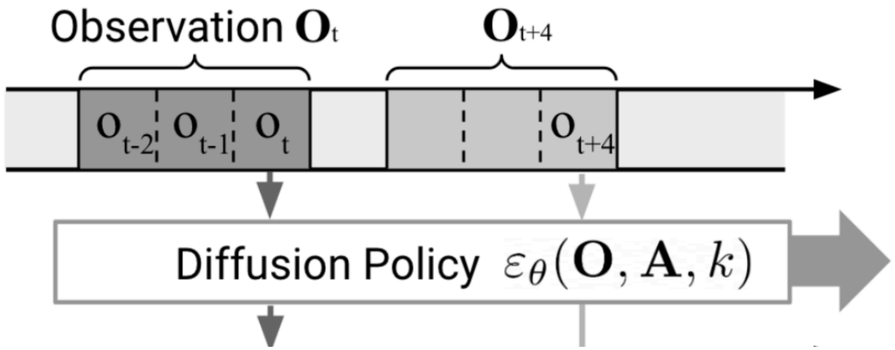
Input: Image Observation Sequence



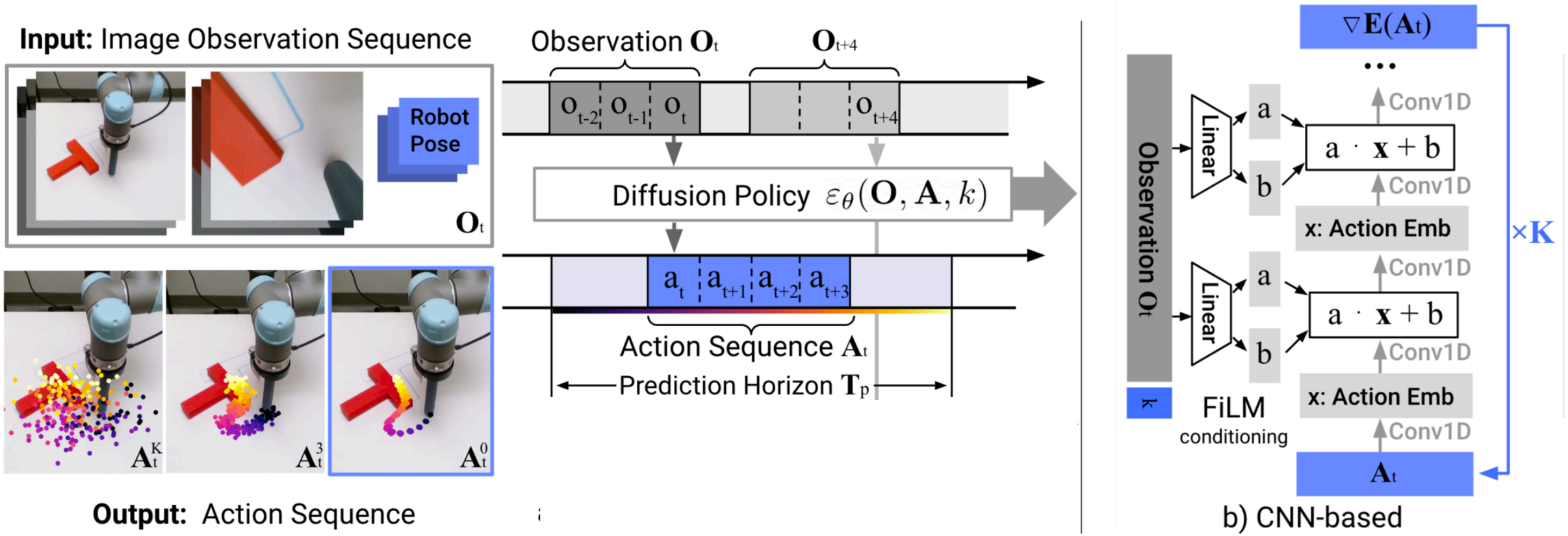


Input: Image Observation Sequence





)


Input: Image Observation Sequence
Robot Pose
$\mathbf{O}_t$
$\mathbf{A}_t^K$
$\mathbf{A}_t^3$
$\mathbf{A}_t^0$
Output: Action Sequence
Observation $\mathbf{O}_t$
$\mathbf{O}_{t+4}$
o_{t-2} o_{t-1} o_t o_{t+4}
Diffusion Policy $\varepsilon_\theta(\mathbf{O}, \mathbf{A}, k)$
a_t a_{t+1} a_{t+2} a_{t+3}
Action Sequence $\mathbf{A}_t$
Prediction Horizon $\mathbf{T}_p$
$\nabla \mathbf{E}(\mathbf{A}_t)$
Observation $\mathbf{O}_t$
Linear
a
b
Conv1D
$a \cdot x + b$
x: Action Emb
k
FiLM
conditioning
$\mathbf{A}_t$
×K
b) CNN-based

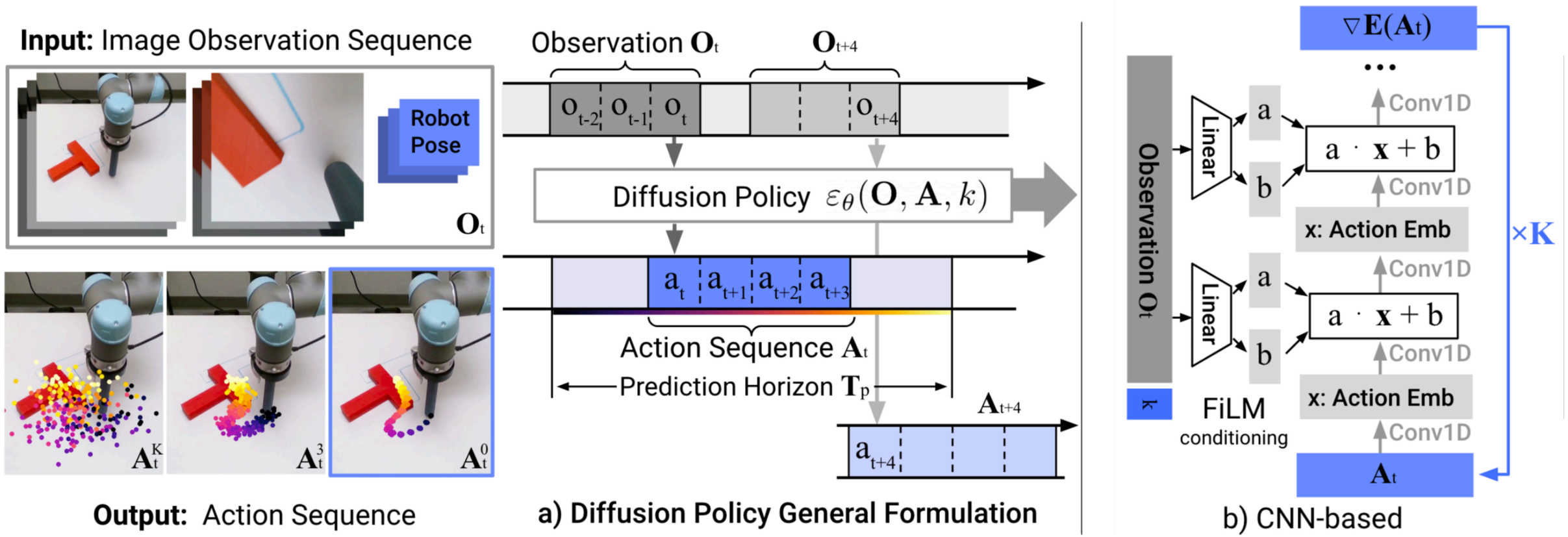


Image backbone: ResNet-18 (pretrained on ImageNet)
Total: 110M-150M Parameters
Training Time: 3-6 GPU Days ($150-$300)

Slide credit: Russ Tedrake

Collecting Demonstrations



Collecting Demonstrations



Diffusion Policy



Diffusion Policy







Aloha Unleashed

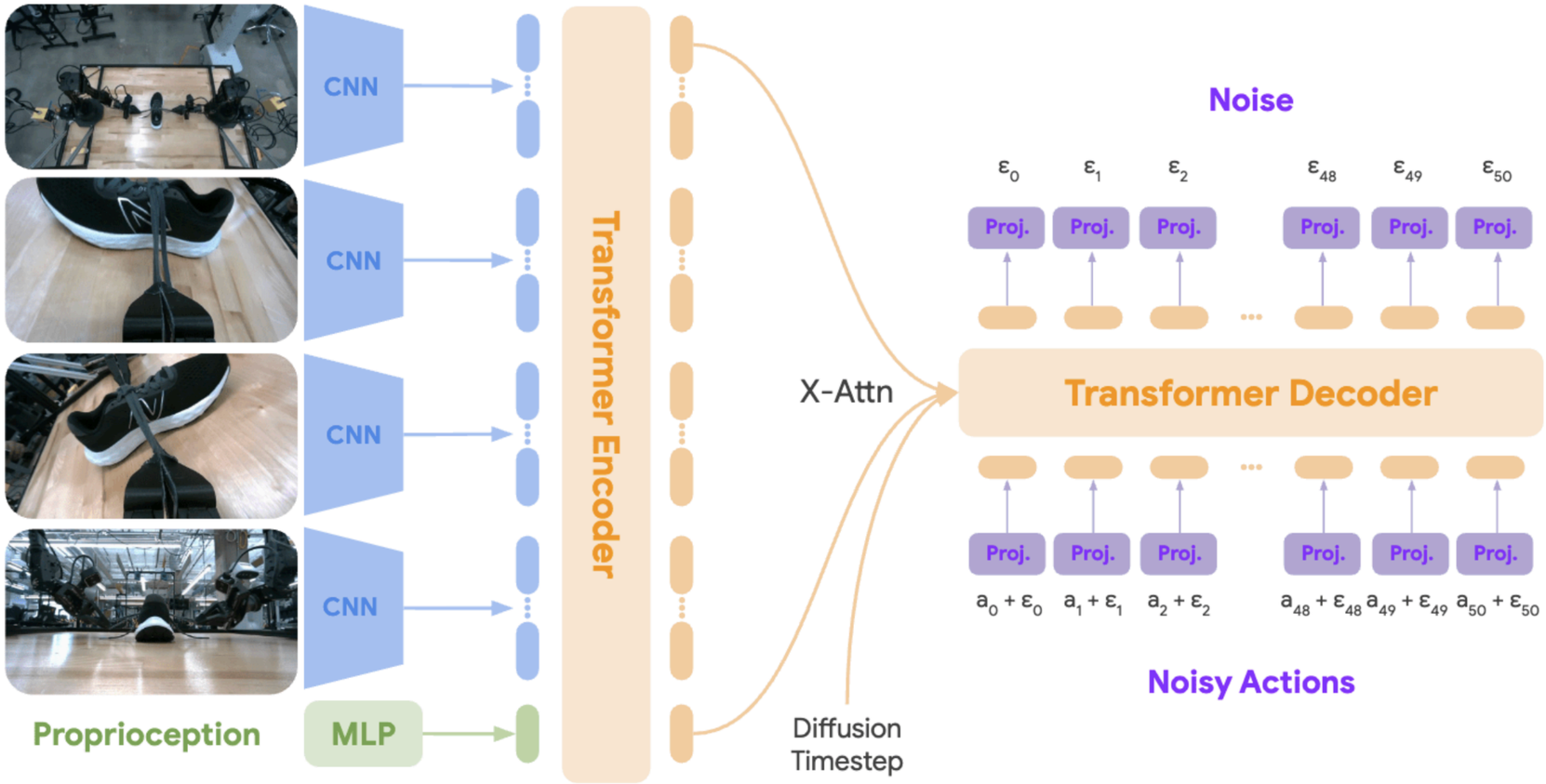
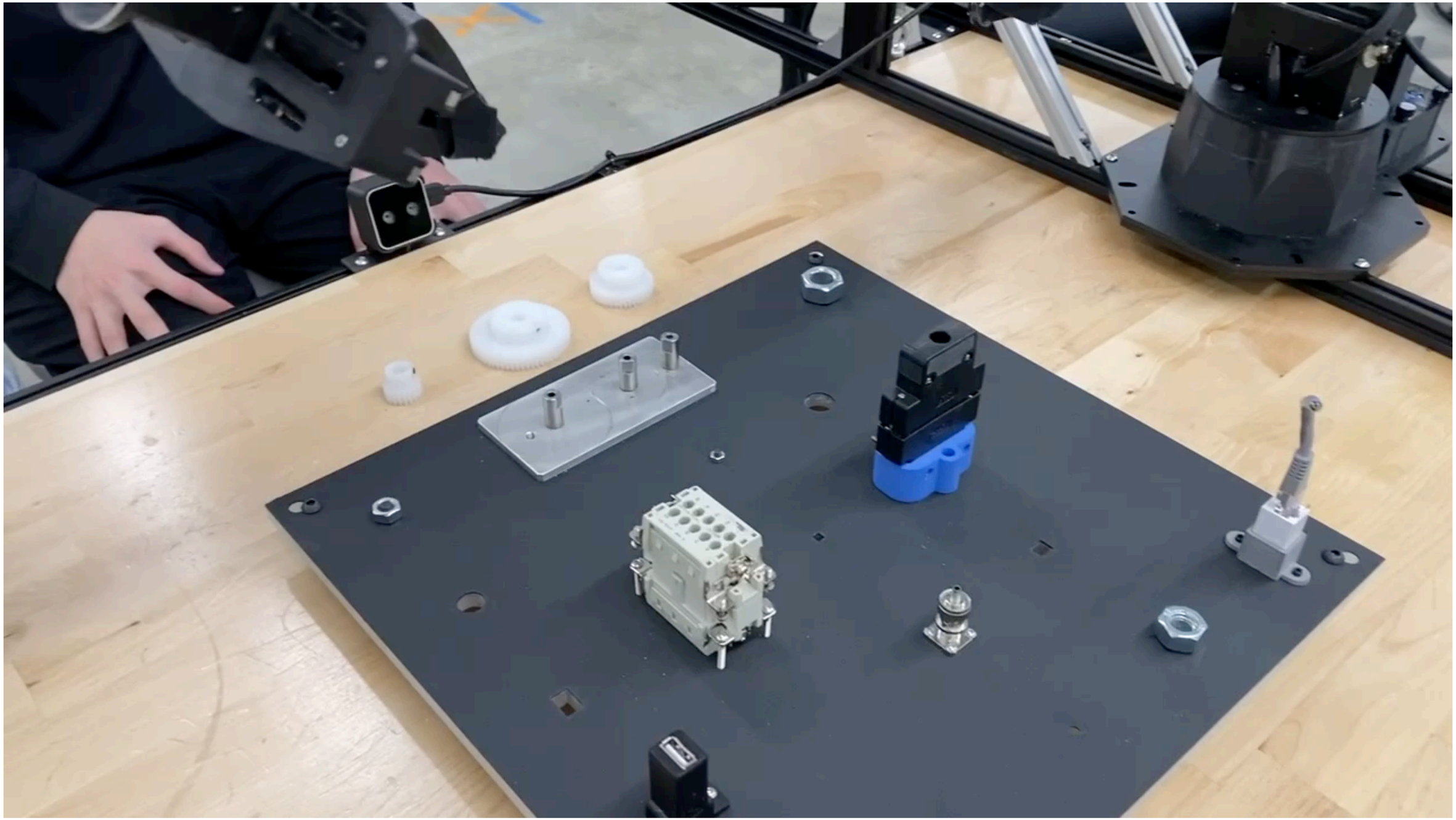
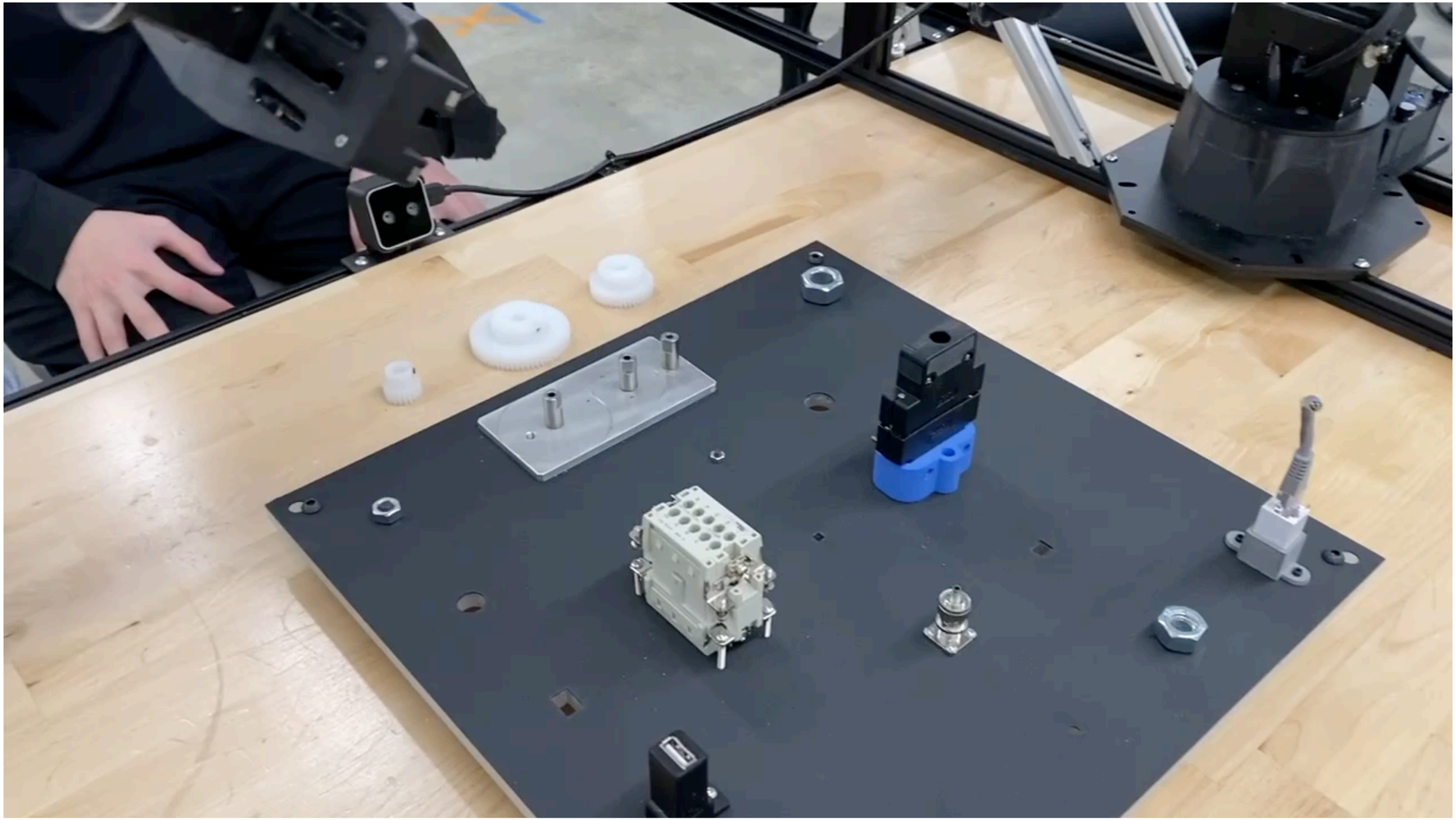
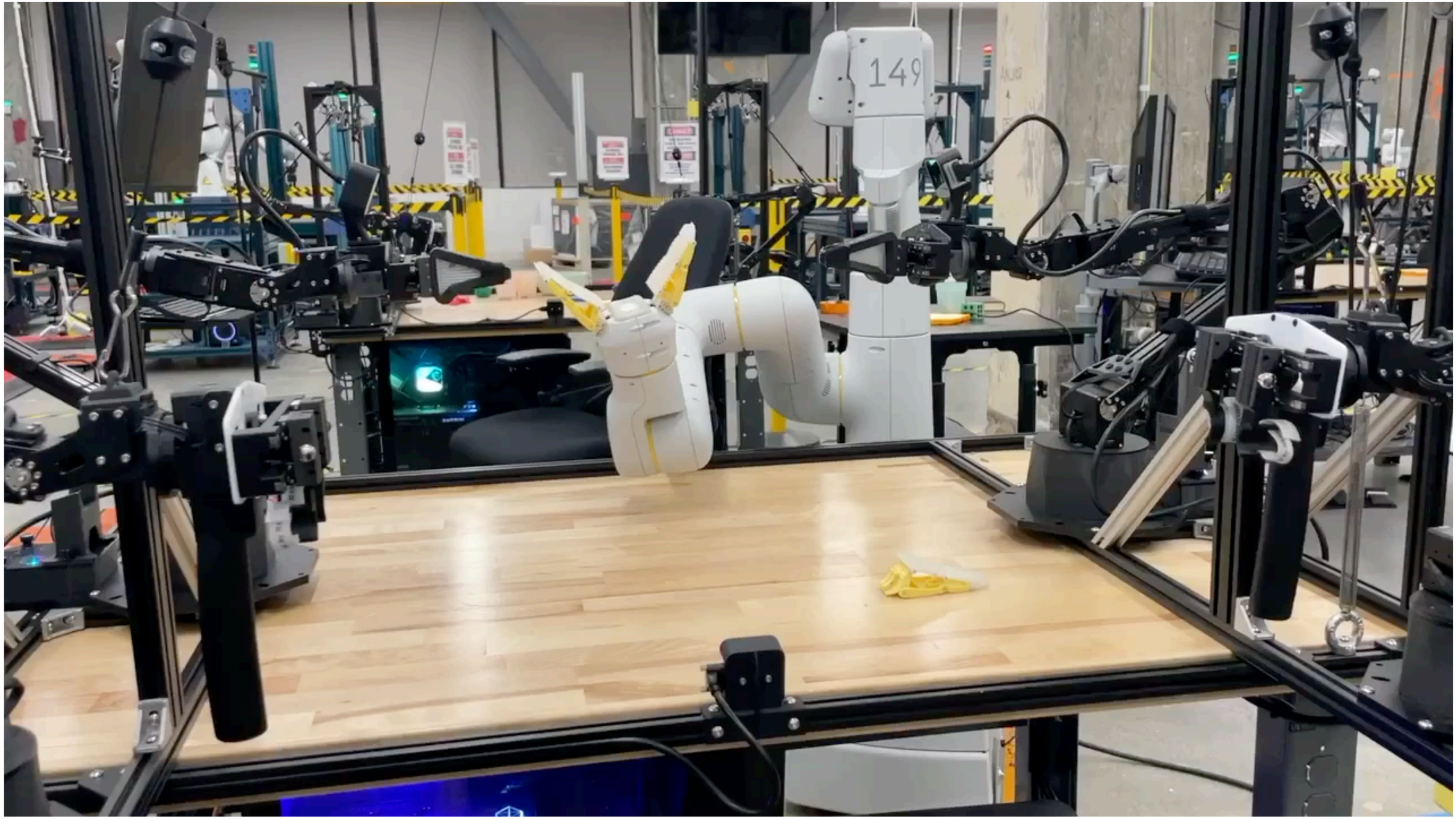
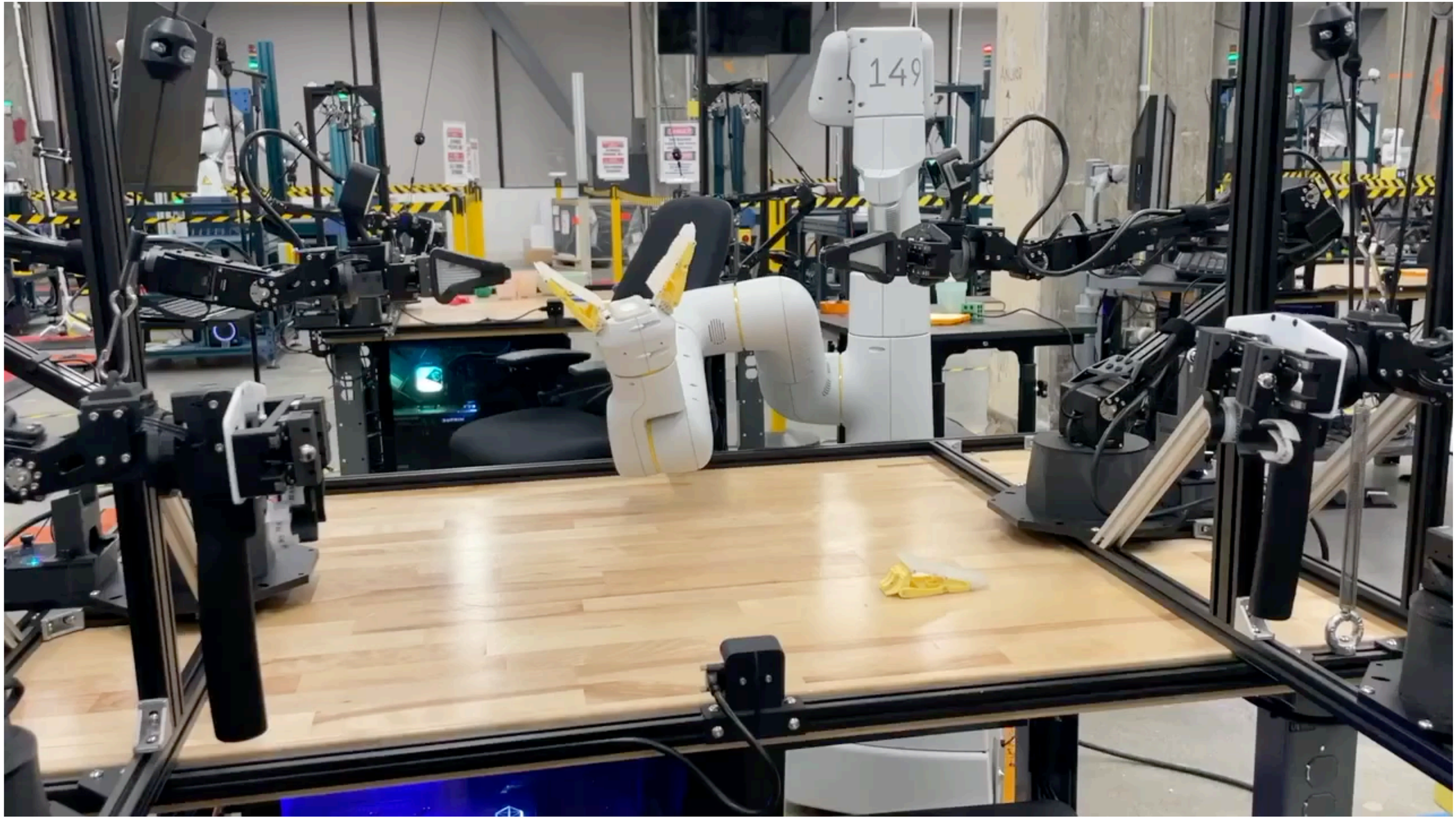


Figure 2: ***Policy Architecture.*** Each camera view is embedded with a ResNet50 [46]. The Transformer Encoder embeds the observations and produces latent embeddings. The Transformer Decoder takes in noisy actions and cross-attends to the latent embeddings produced by the encoder, outputting the predicted noise. The decoder portion runs 50 times during inference to iteratively denoise an action chunk.





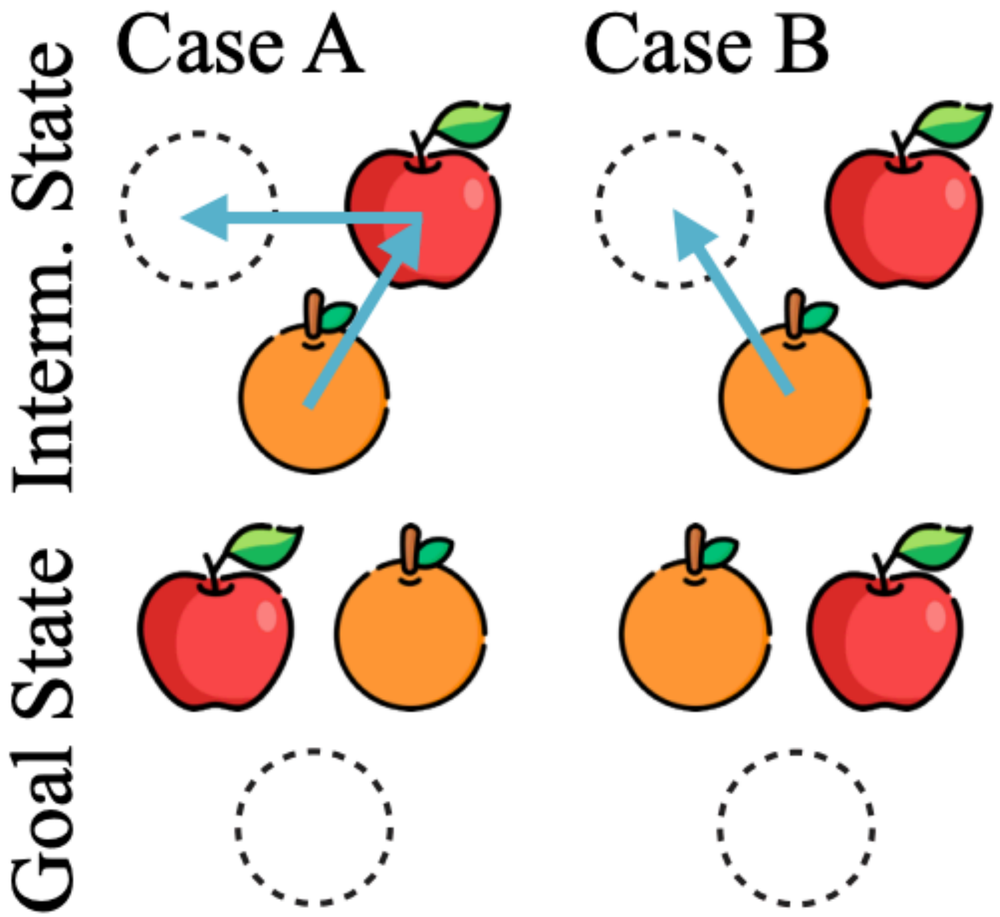

149


149

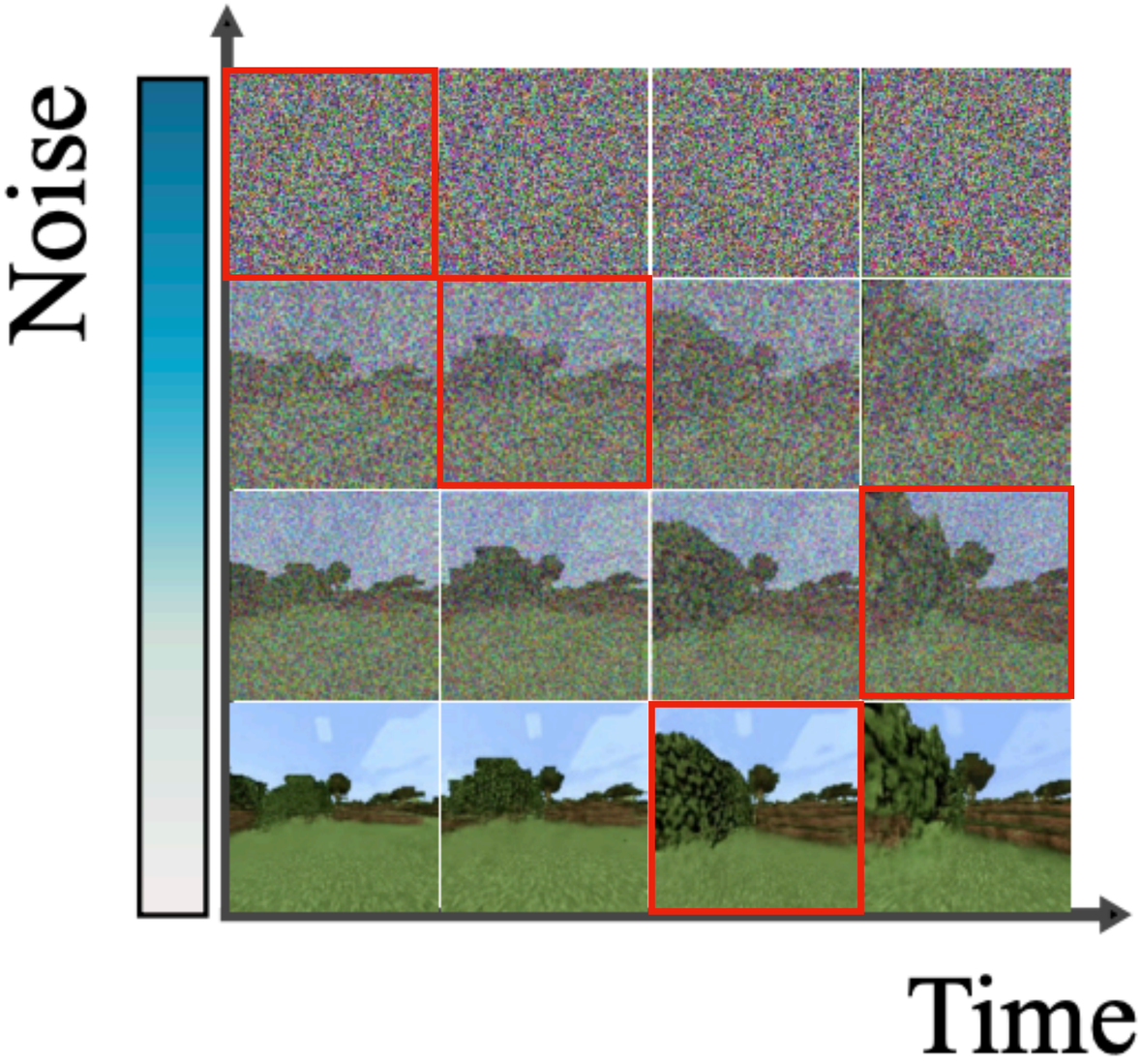




Long-Horizon, Stateful Behavior Cloning



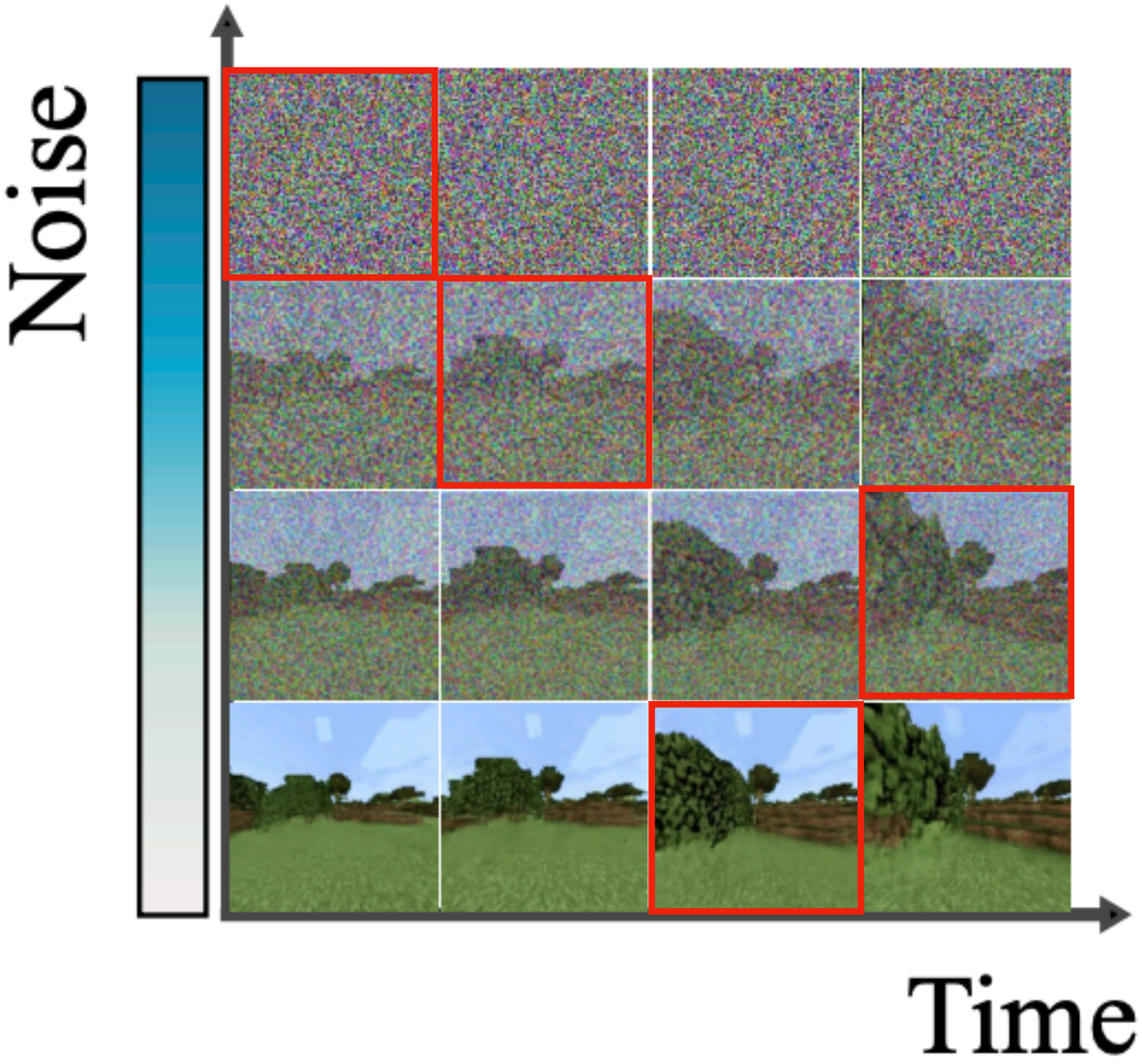
Diffusion Forcing



Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

[Chen et al. 2024, Song and Chen et al. 2024]

Diffusion Forcing

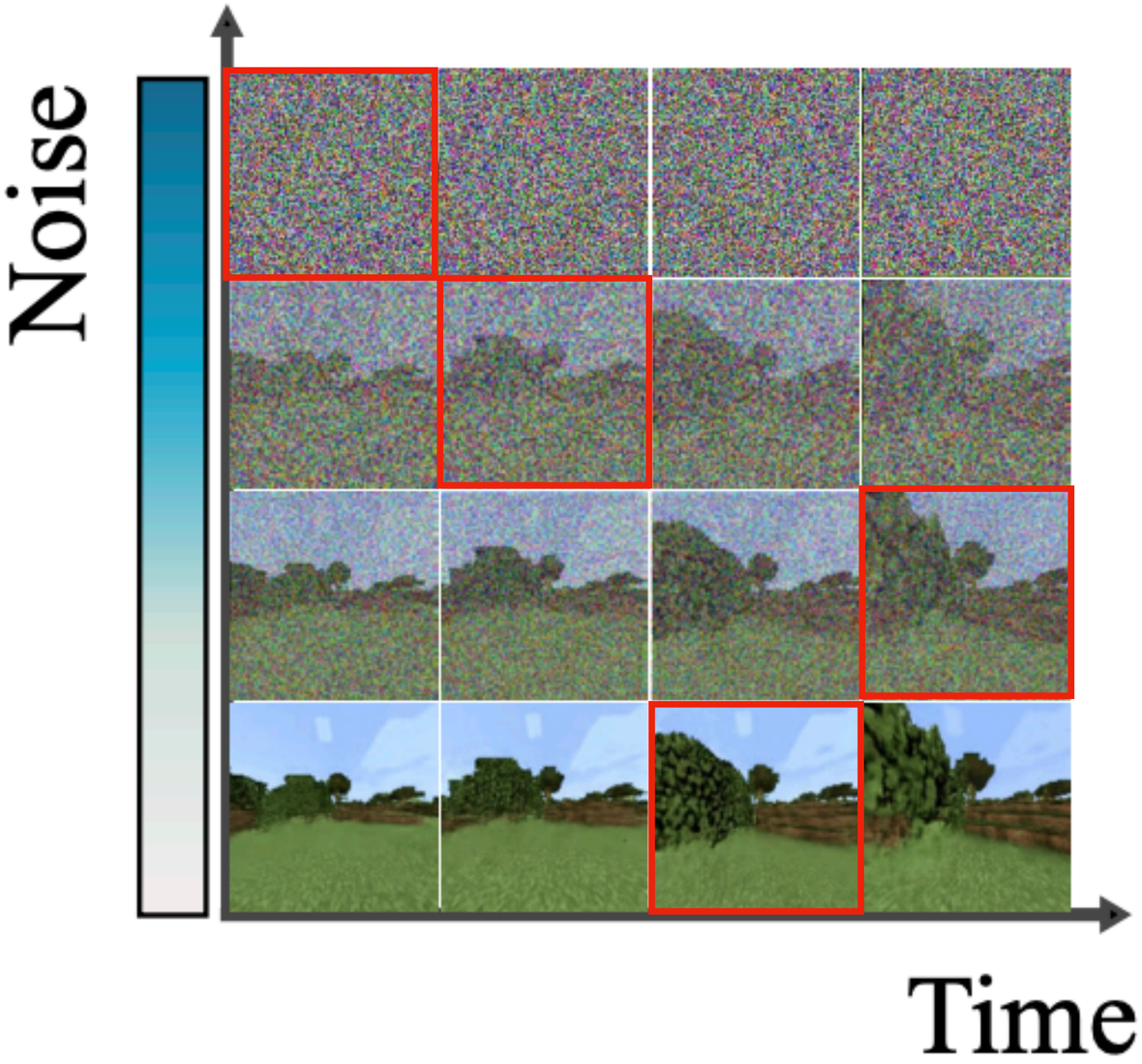


Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

Over time, the model sees *every possible combination of noise levels*

[Chen et al. 2024, Song and Chen et al. 2024]

Diffusion Forcing



Key Idea: Train the model to denoise frames with *random combinations of noise levels!*

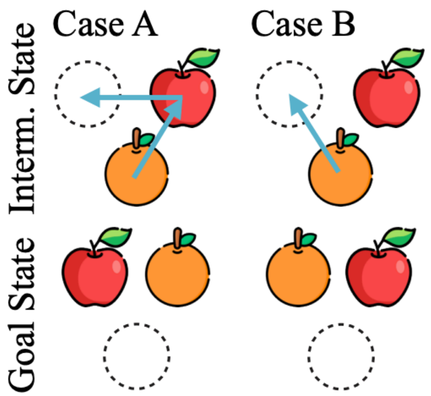
Over time, the model sees *every possible combination of noise levels*

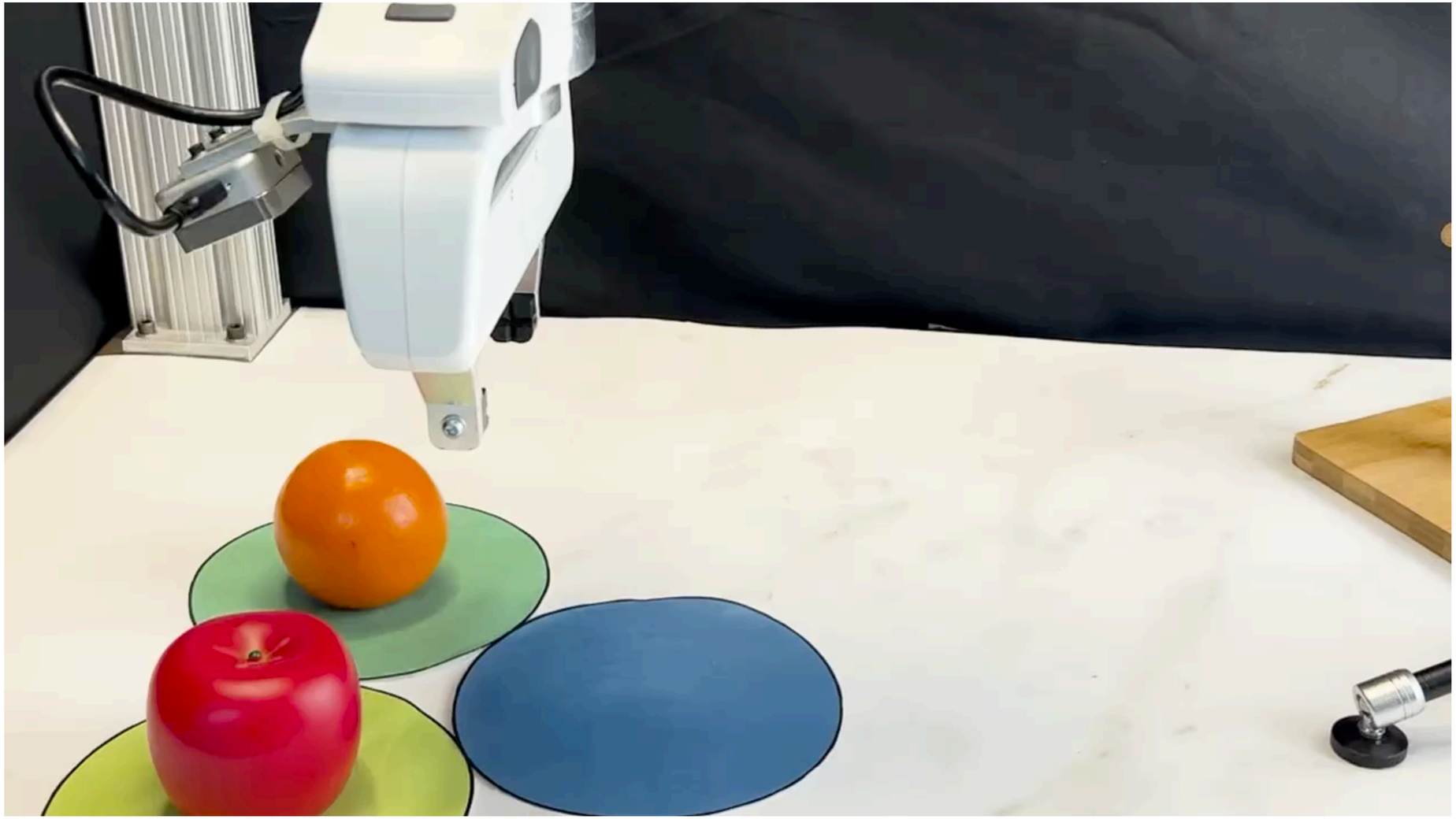
This generalizes both auto-regressive sampling and full-sequence diffusion

[Chen et al. 2024, Song and Chen et al. 2024]

Long-Horizon, Stateful Behavior Cloning

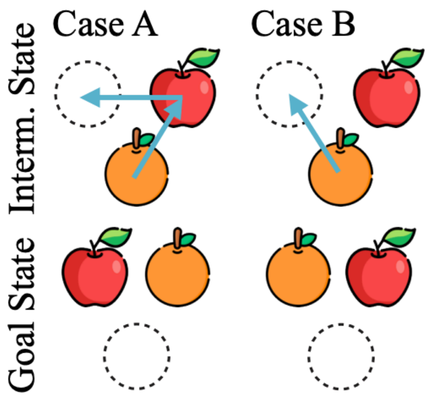
Diffusion Forcing, Chen et al. 2024

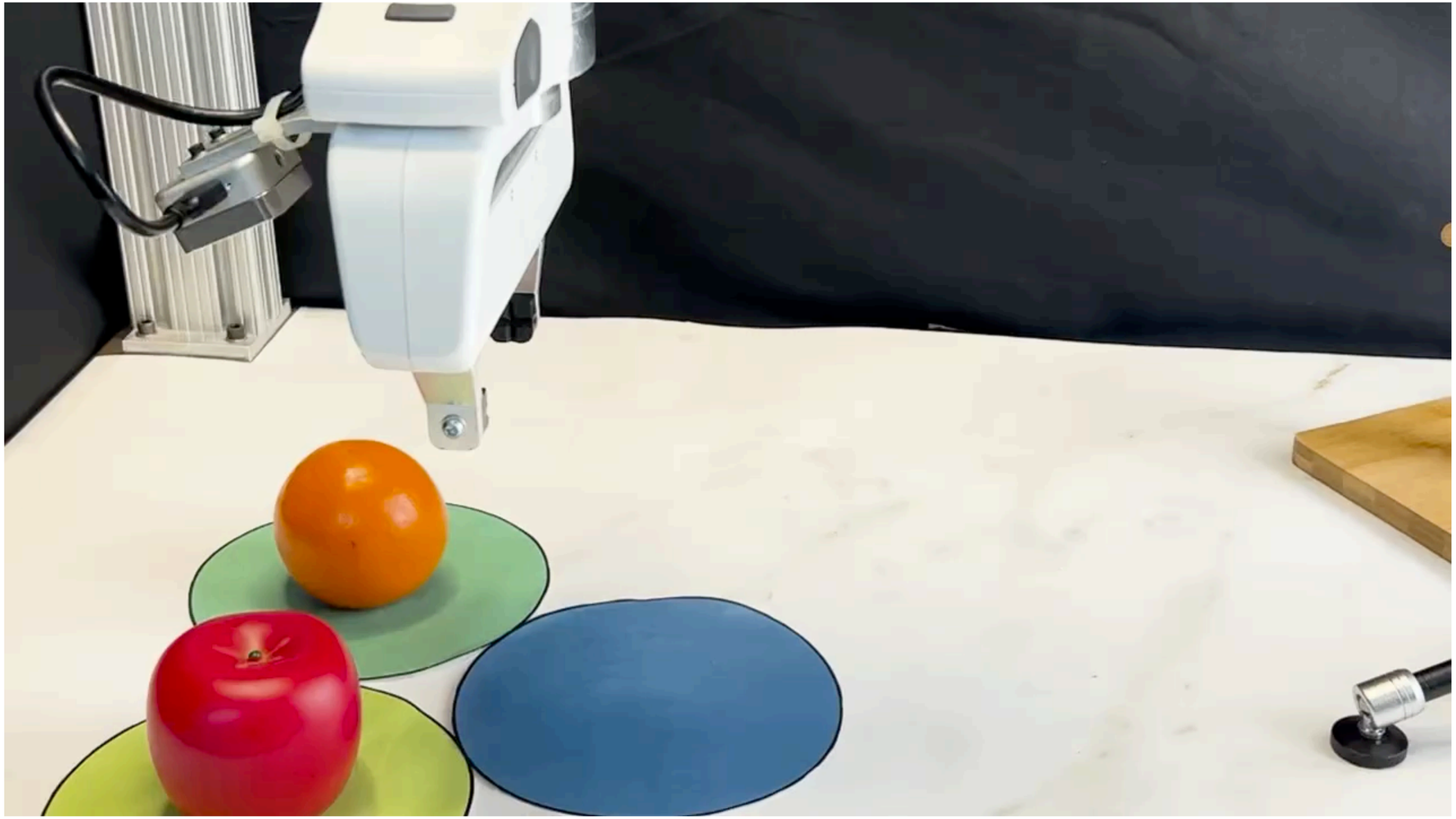




Long-Horizon, Stateful Behavior Cloning

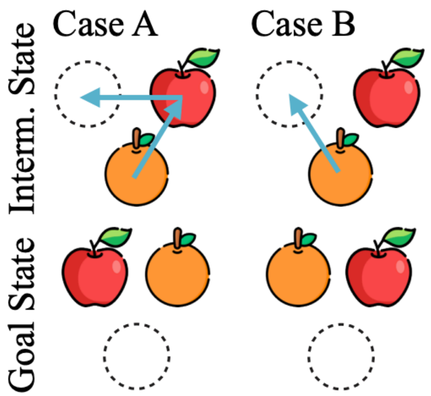
Diffusion Forcing, Chen et al. 2024

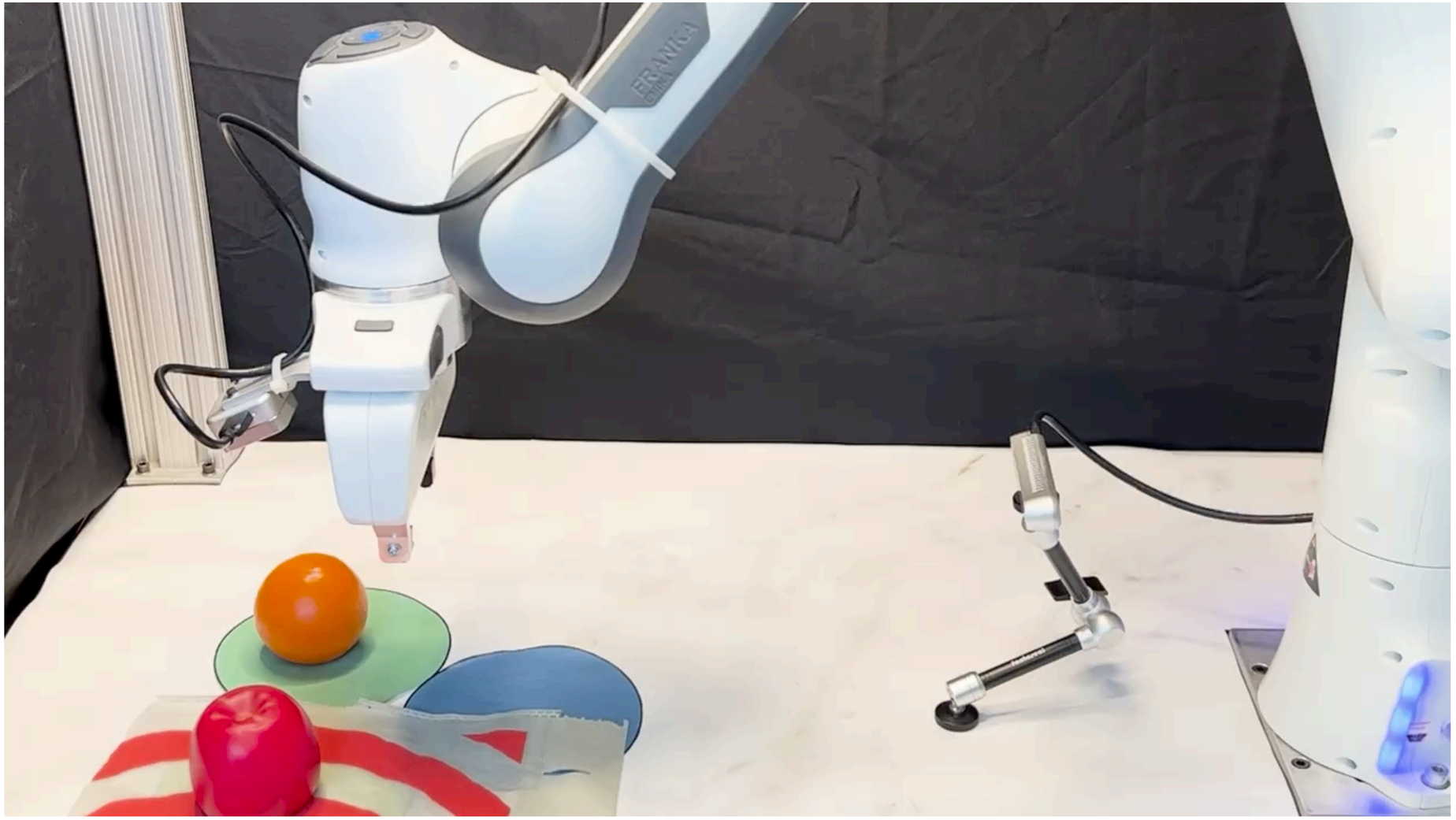




Noise makes model robust to disturbances

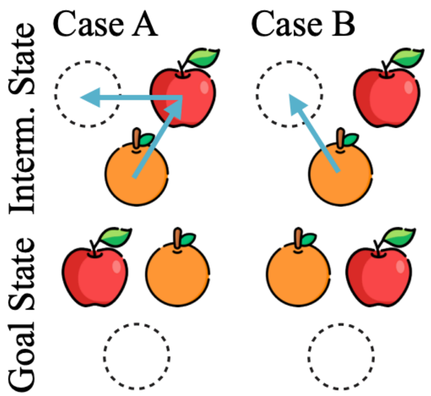
Diffusion Forcing, Chen et al. 2024

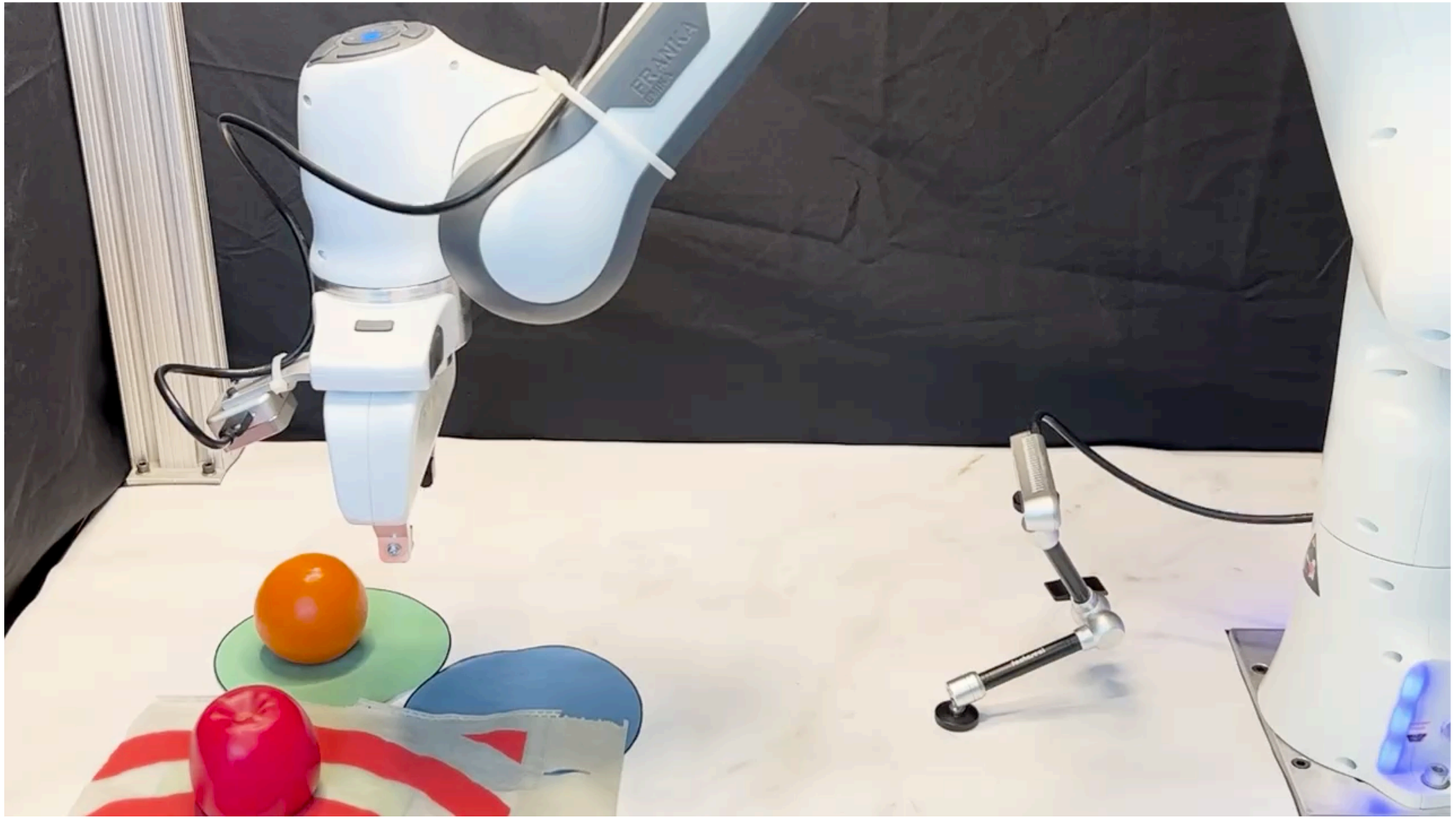




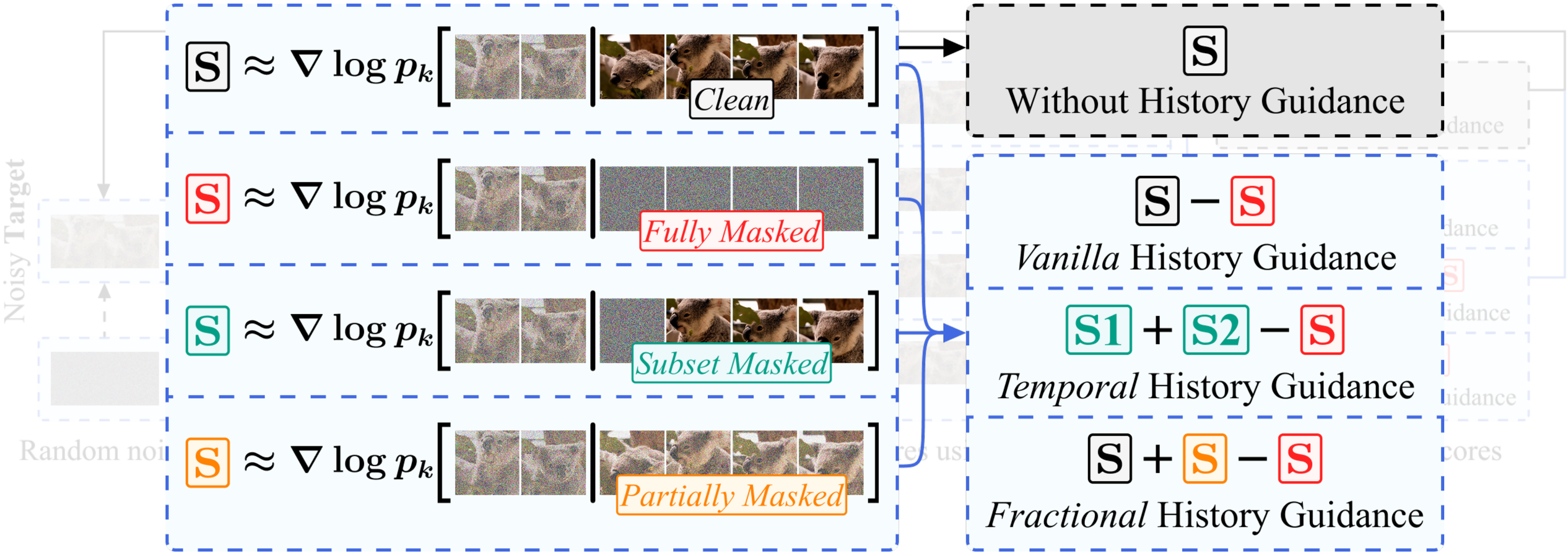
Noise makes model robust to disturbances

Diffusion Forcing, Chen et al. 2024





History-Guided Video Diffusion

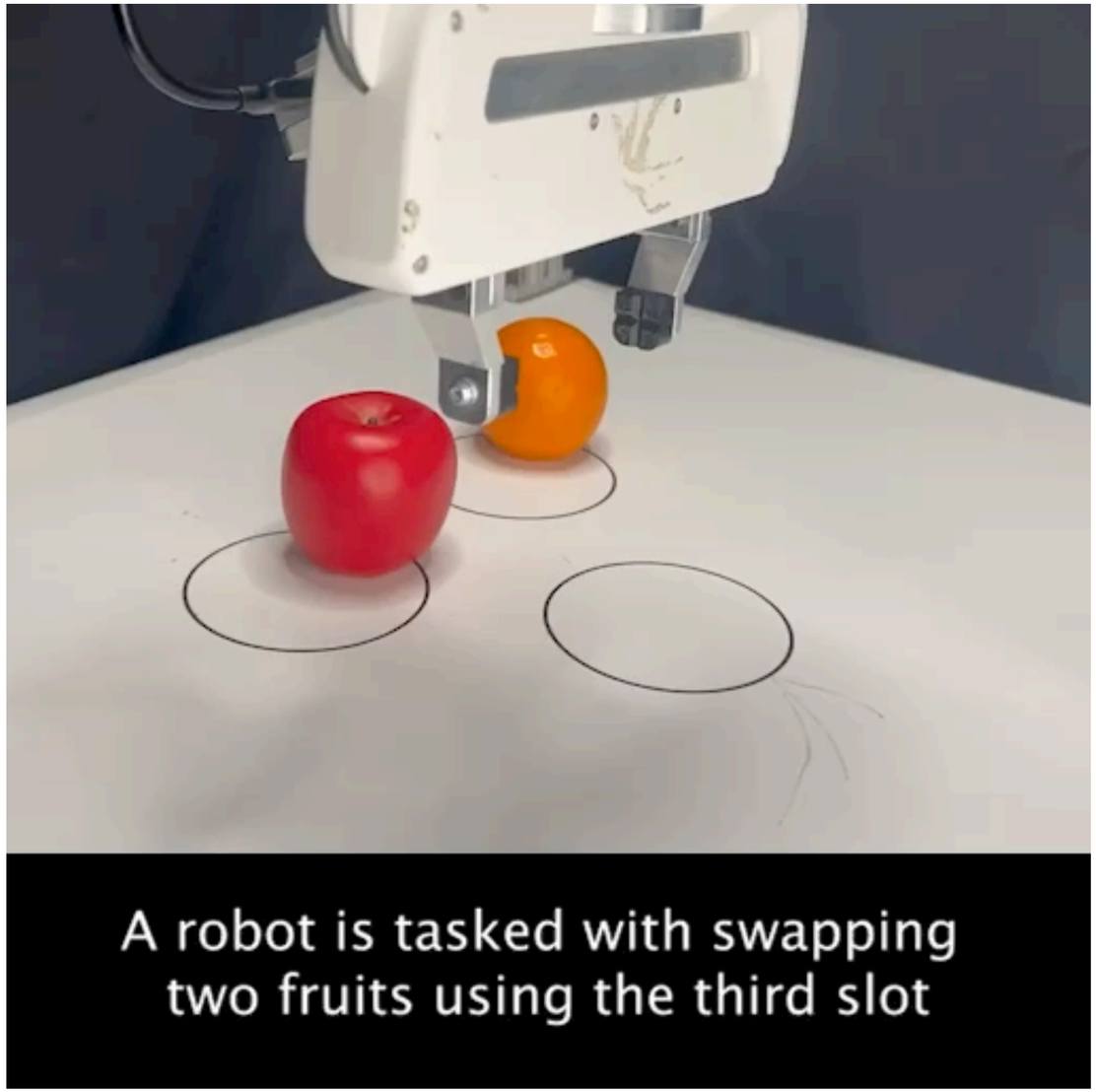


History-Guided Video Diffusion:
Composing Robot Actions

S − S
Vanilla History Guidance

S1 + S2 − S
Temporal History Guidance

S + S − S
Fractional History Guidance



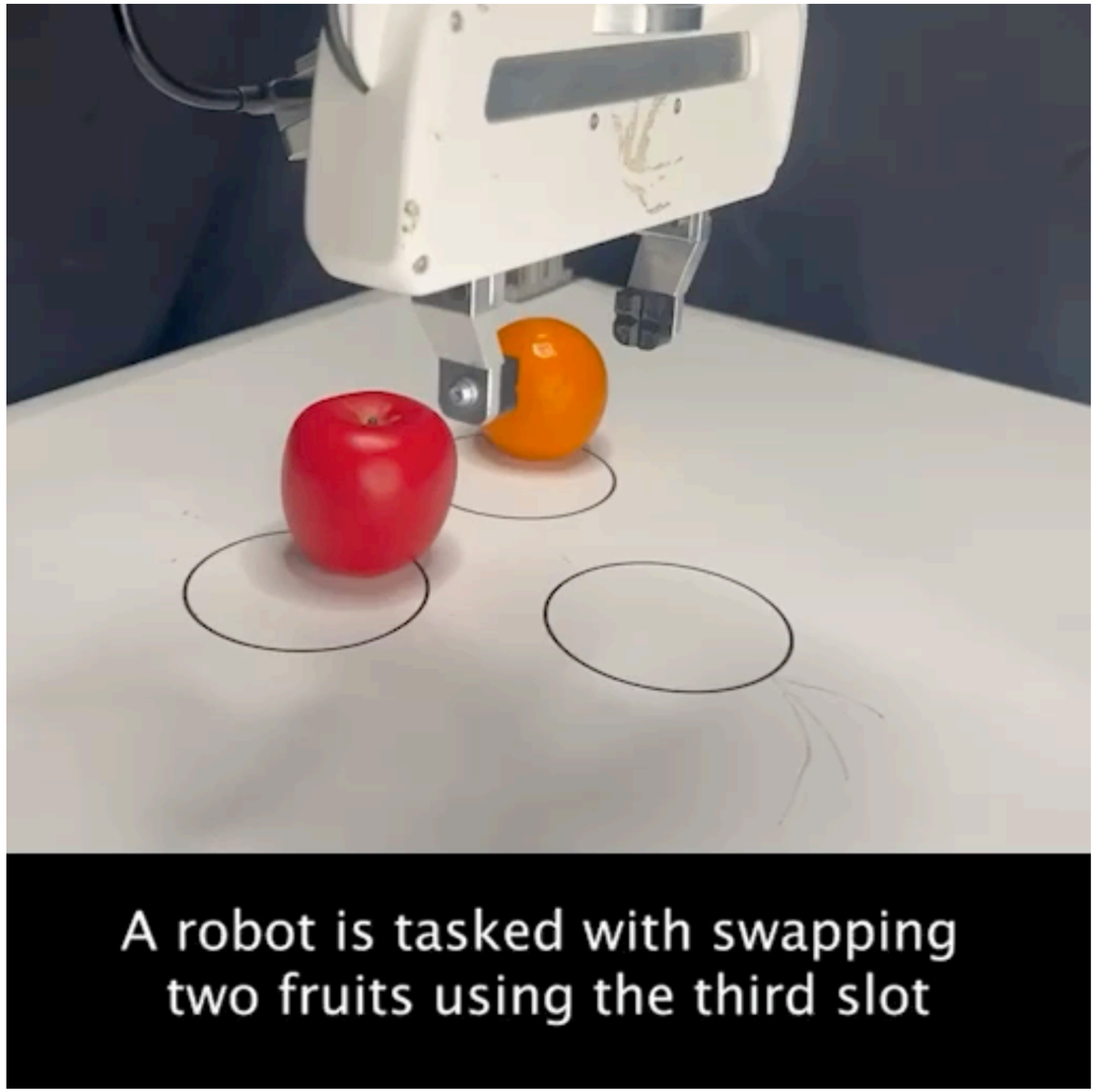
In imitation learning, temporal history guidance enables *composition of reactive and long-horizon behavior.*

History-Guided Video Diffusion:
Composing Robot Actions

S − S
Vanilla History Guidance

S1 + S2 − S
Temporal History Guidance

S + S − S
Fractional History Guidance



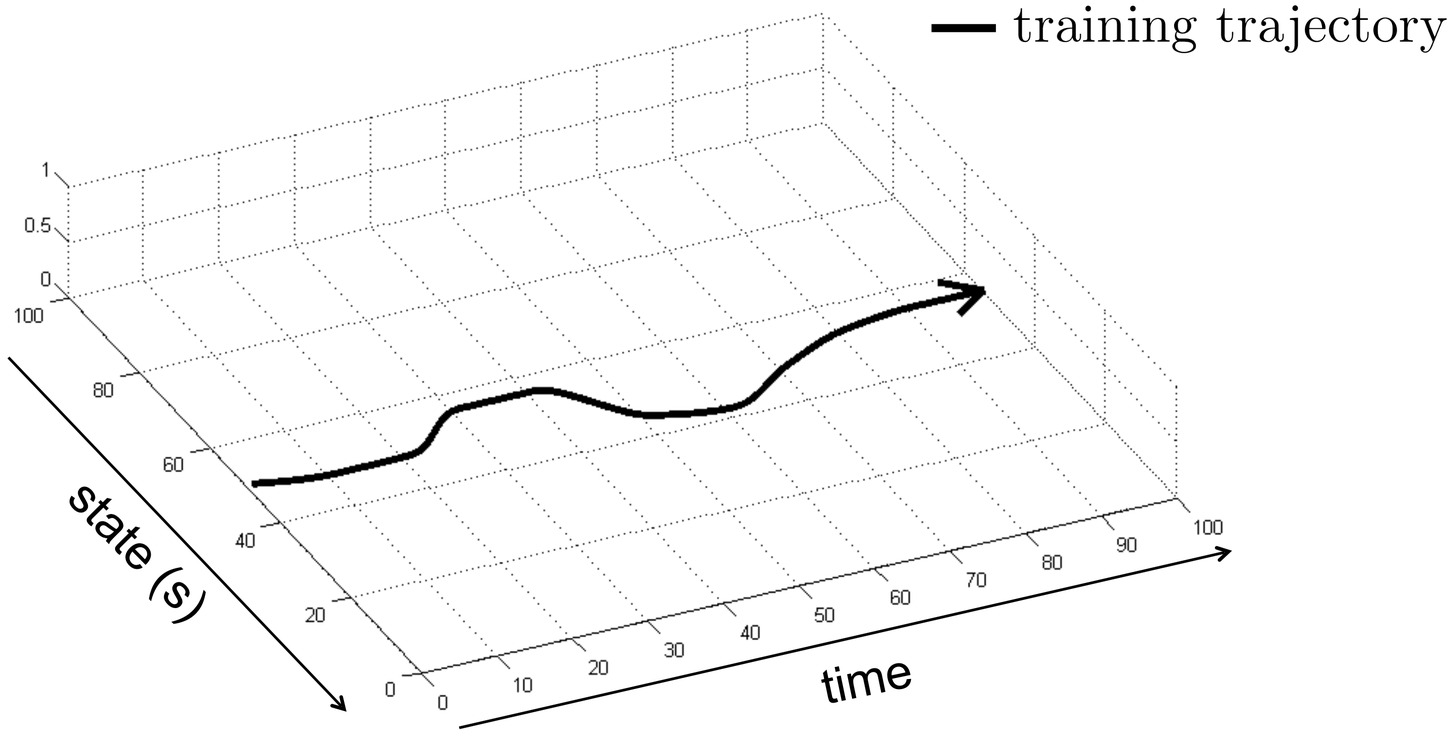
In imitation learning, temporal history guidance enables *composition of reactive and long-horizon behavior.*

Does it work?

No!

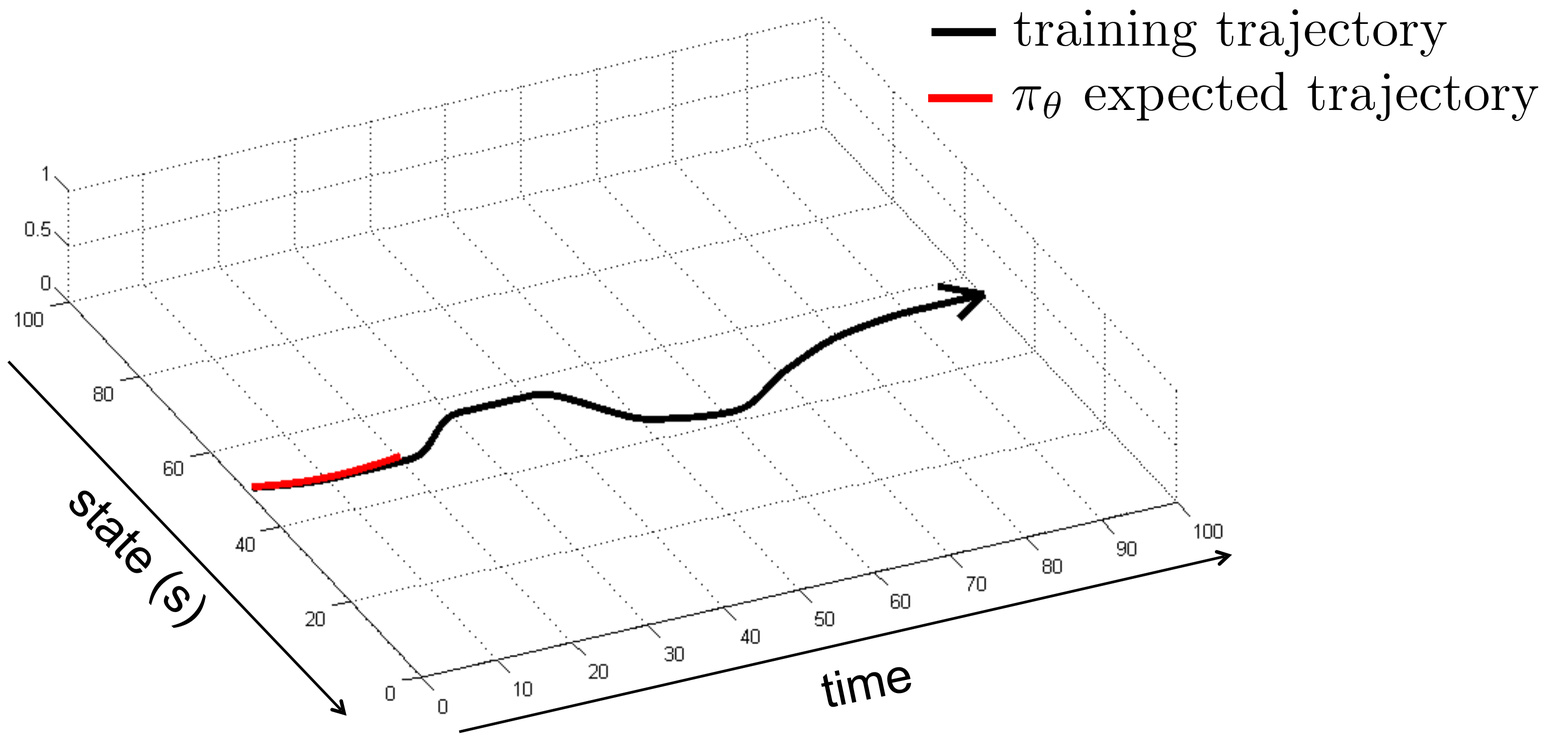
Does it work?

No!



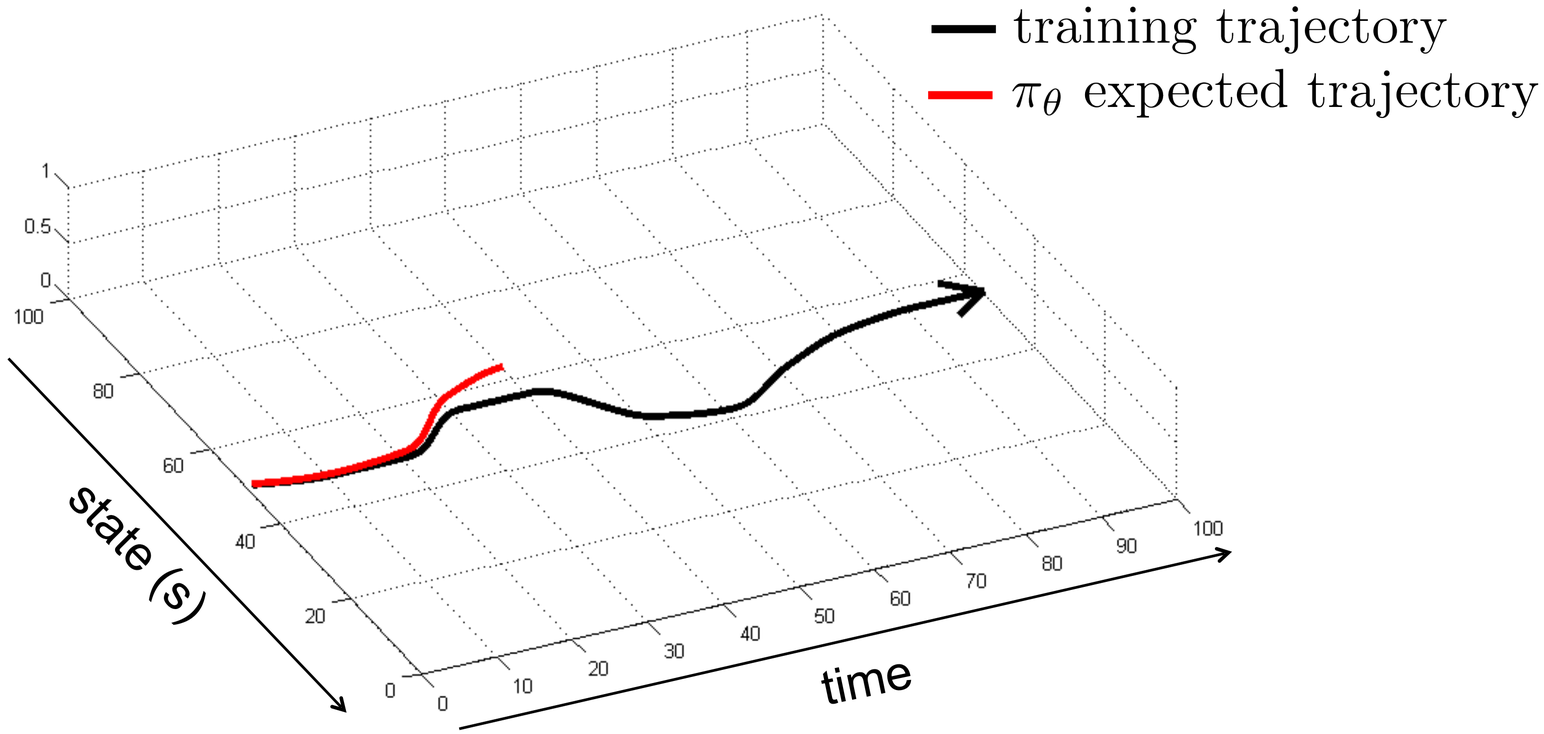
Does it work?

No!



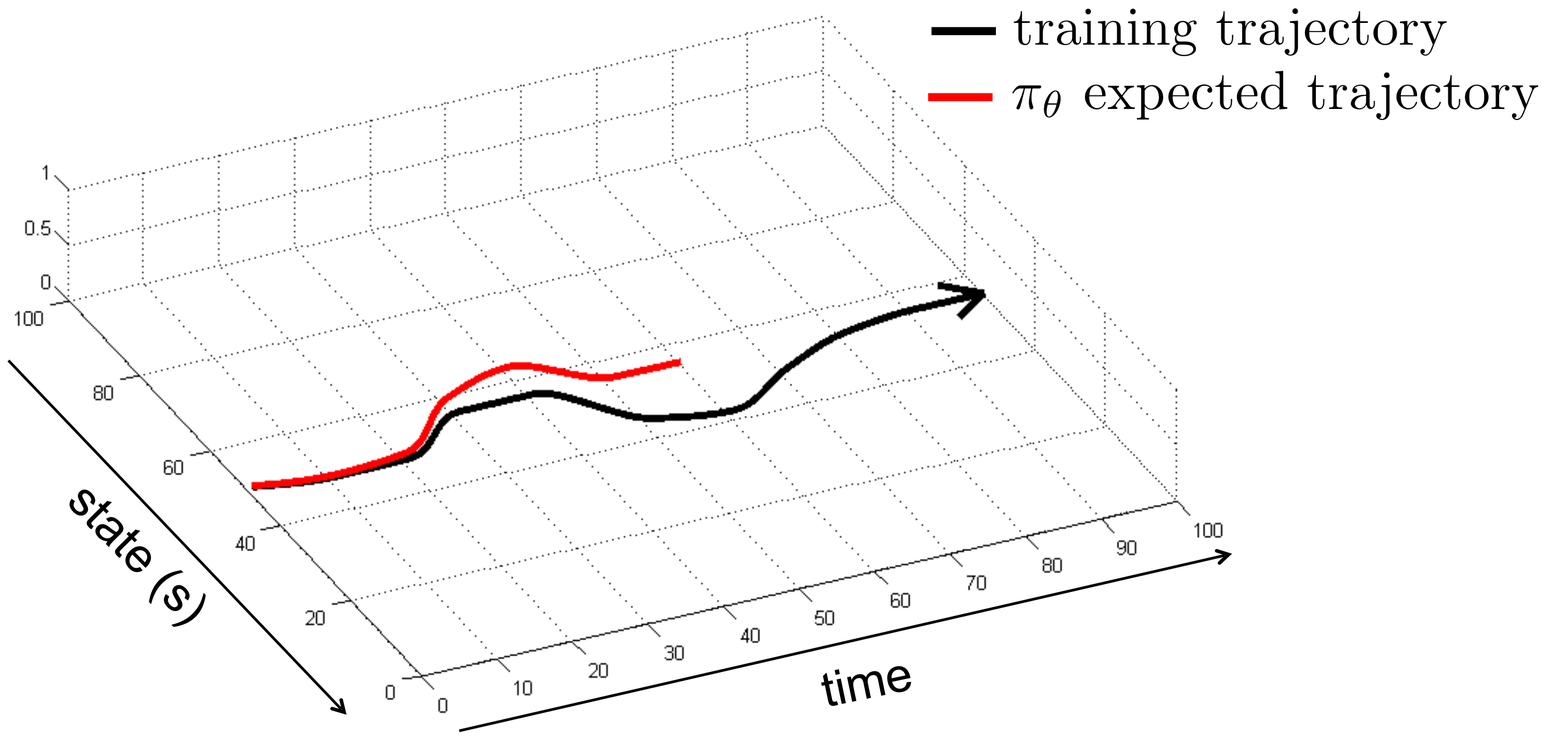
Does it work?

No!



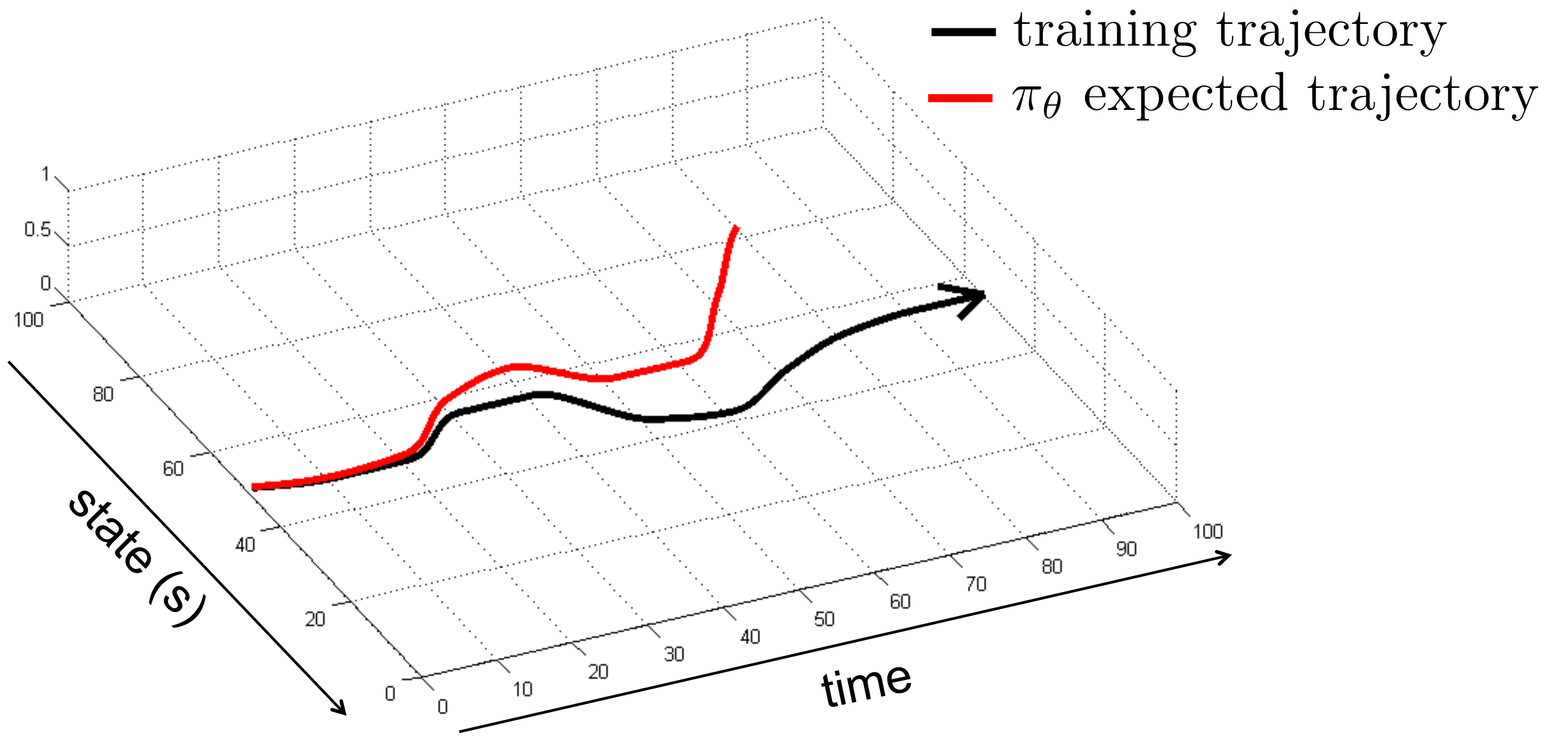
Does it work?

No!



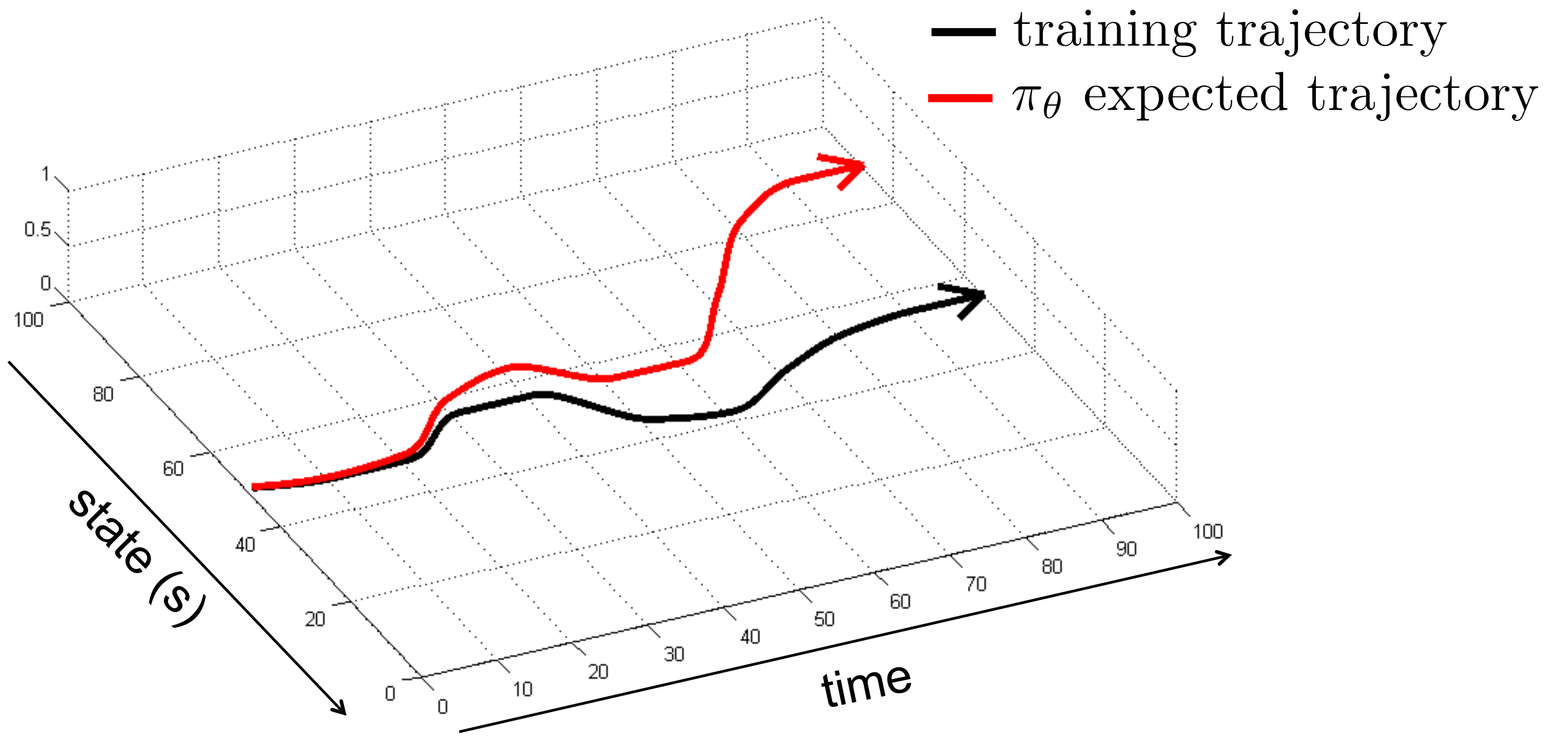
Does it work?

No!



Does it work?

No!



Imitation learning: what’s the problem?

Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful

Imitation learning: what's the problem?

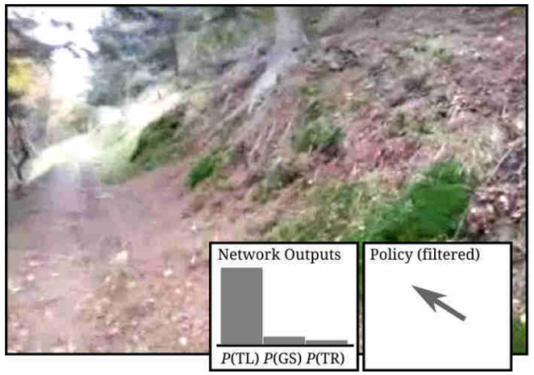
- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions

Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions

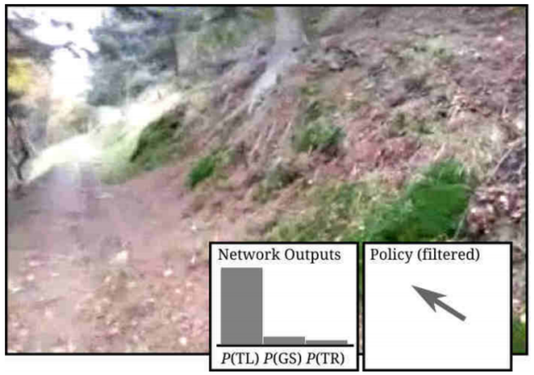
Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions



Imitation learning: what's the problem?

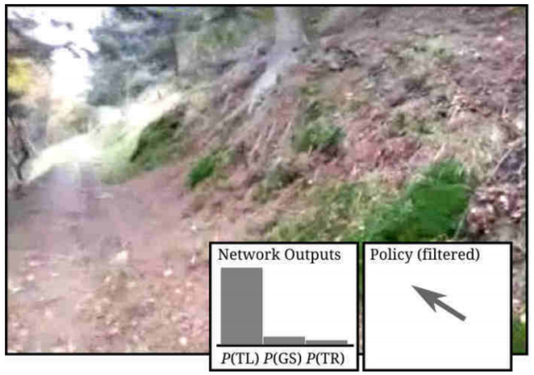
- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions



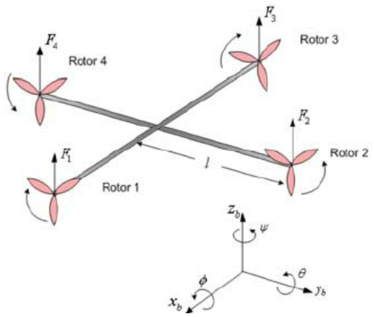


Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions

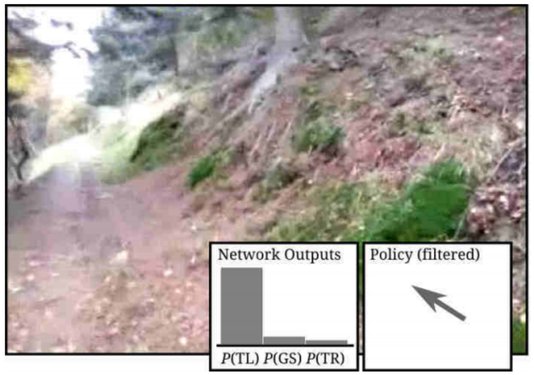




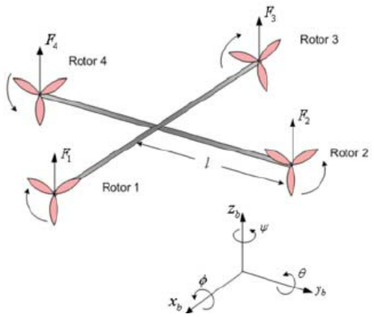


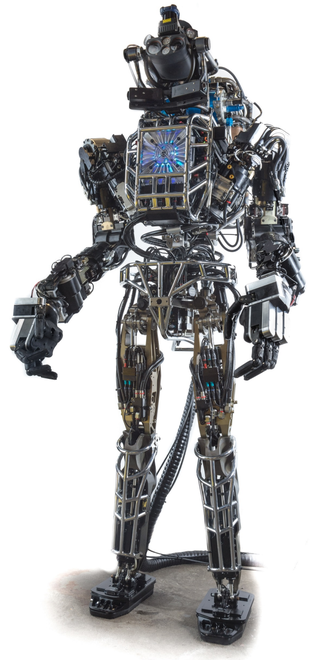
Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions



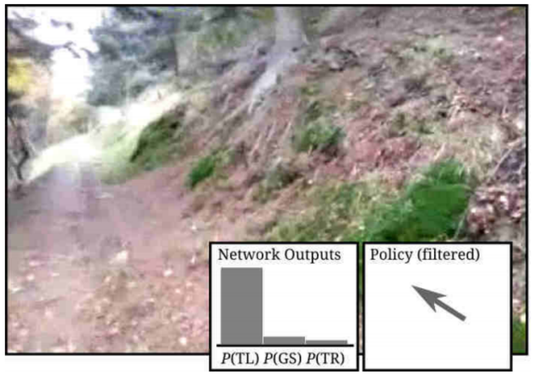




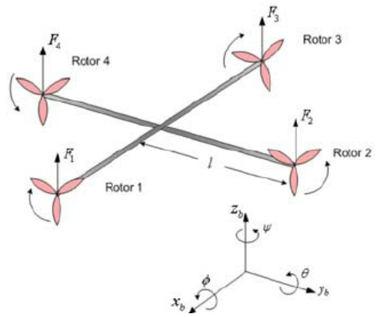


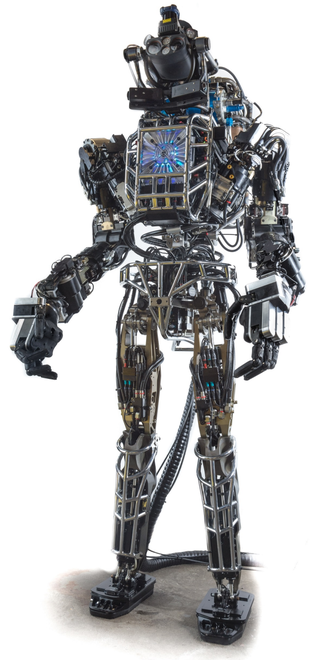
Imitation learning: what's the problem?

- Humans need to provide data, which is typically finite
  - Deep learning works best when data is plentiful
- Humans are not good at providing some kinds of actions









- Humans can learn autonomously; can our machines do the same?
  - Unlimited data from own experience
  - Continuous self-improvement