Efficient Learning for Robot Manipulation: Insights Applicable to Deformable Objects

Renaud Detry

KU LEUVEN

ROMANDIC – Feb 9, 2025

- Diffusion policies and equivariant models have had a significant impact on robot manipulation, including the manipulation of deformable objects.
- Unfortunately, both typically come at the cost of increased compute.
- Today, we discuss means of reducing the compute cost of both DDPM and equivariance in robot manipulation.

Published as a conference paper at ICLR 2025

ET-SEED: EFFICIENT TRAJECTORY-LEVEL SE(3) EQUIVARIANT DIFFUSION POLICY

Chenrui Tie[1,2*] **Yue Chen**[1*] **Ruihai Wu**[1*]
Boxuan Dong[1] **Zeyi Li**[1] **Chongkai Gao**[2†] **Hao Dong**[1†]
[1]Peking University [2]National University of Singapore
chenrui.tie@u.nus.edu yuechen@stu.pku.edu.cn wuruihai@pku.edu.cn

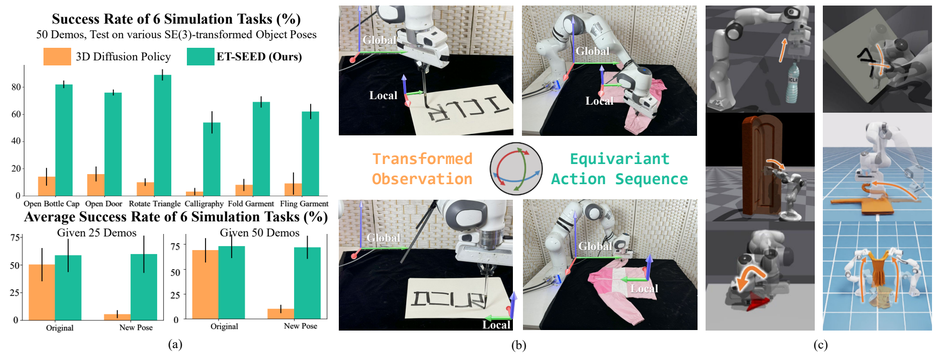


Figure 1: ET-SEED is a visual imitation learning algorithm that marries $SE(3)$ equivariant visual representations with diffusion policies. (a) ET-SEED achieve surprising **efficiency** and **spatial generalization** than baselines. (b) When the input object observation is rotated or translated, the output action sequence change equivariantly. (c) Visualizations of simulation environments.

2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

EquivAct: SIM(3)-Equivariant Visuomotor Policies beyond Rigid Object Manipulation

Jingyun Yang*[1], Congyue Deng*[1], Jimmy Wu[2], Rika Antonova[1], Leonidas Guibas[1], Jeannette Bohg[1]

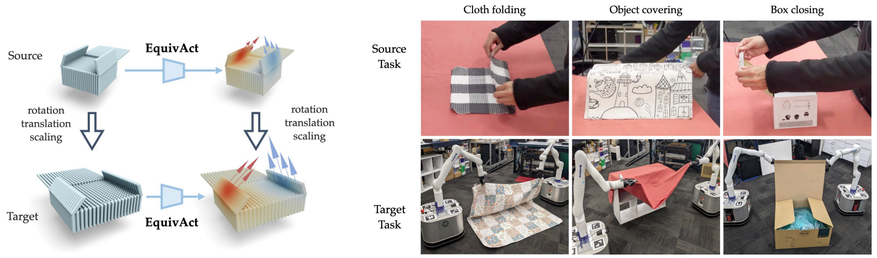


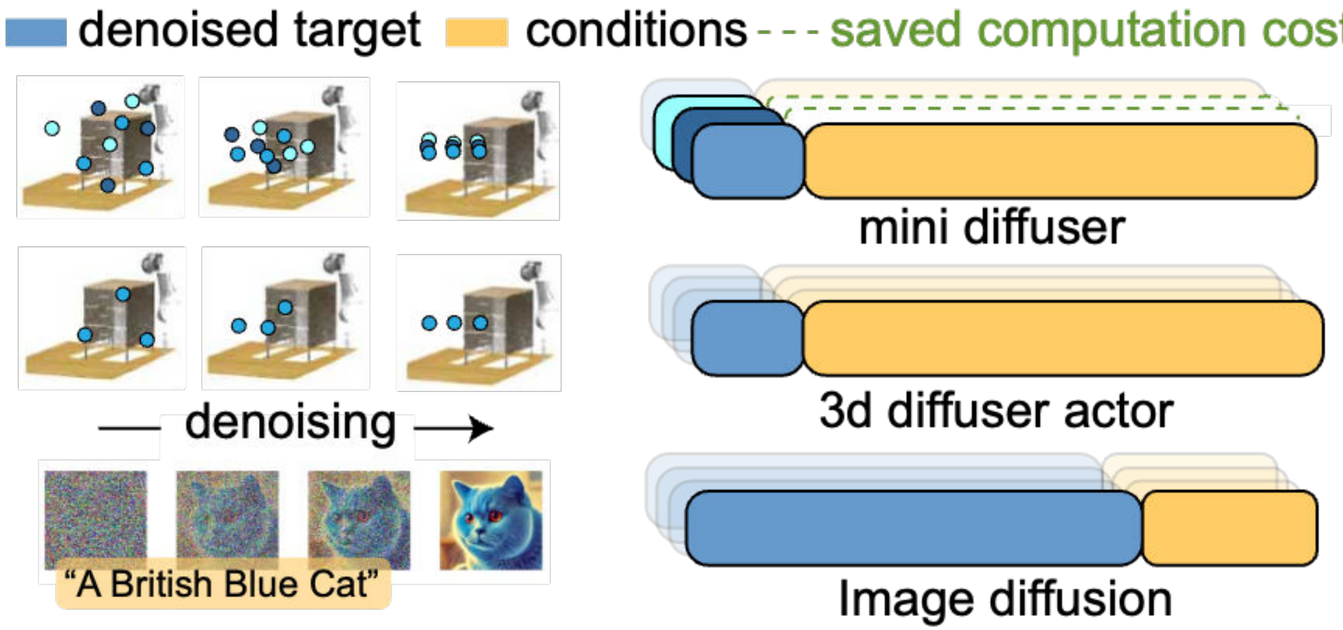
Fig. 1: **Overview.** Constructed with SIM(3)-equivariant point cloud networks, our method takes a few examples of solving a source task, then generalizes zero-shot to changes in object *appearances*, *scales*, and *poses*.

***Abstract*— If a robot masters folding a kitchen towel, we would expect it to master folding a large beach towel. However, existing policy learning methods that rely on data augmentation still don't guarantee such generalization. Our insight is to add equivariance to both the visual object representation and policy architecture. We propose *EquivAct* which utilizes SIM(3)-equivariant network structures that guarantee generalization across all possible object translations, 3D rotations, and scales by construction. EquivAct is trained in two phases. We first pre-train a SIM(3)-equivariant visual representation on simulated scene point clouds. Then, we learn a SIM(3)-equivariant visuomotor policy using a small amount of source task demonstrations. We show that the learned policy directly transfers to objects that substantially differ from demonstrations in scale, position, and orientation. We**

In this work, we focus on the problem of learning visuomotor policies that can take a few example trajectories from a single source manipulation scenario as input, then generalize zero-shot to scenarios with changes in objects' appearances, scales, and poses. We go beyond pick-and-place of rigid objects and also handle deformable and articulated objects, such as clothes and boxes. Our insight is to *add equivariance to both the visual object representation and policy architecture*, enabling policies to generalize to novel object positions, orientations, and scales *by construction*.

We propose *EquivAct*, a novel visuomotor policy learning method that can learn closed-loop policies for 3D robot

Mini Diffuser: Fast Multi-task Diffusion Policy Training Using Two-level Mini-batches



Yutong Hu, Pinhao Song,
Kehan Wen, Renaud Detry

"*Reduces by an order of magnitude*
the time and memory needed to train
multi-task vision-language robotic diffusion policies."

Image vs robot diffusion: same model, different target

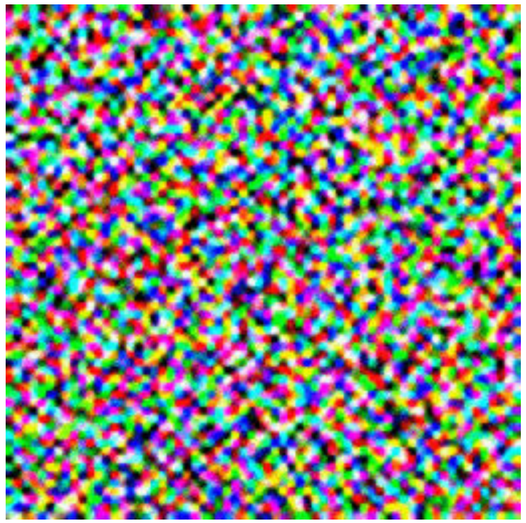
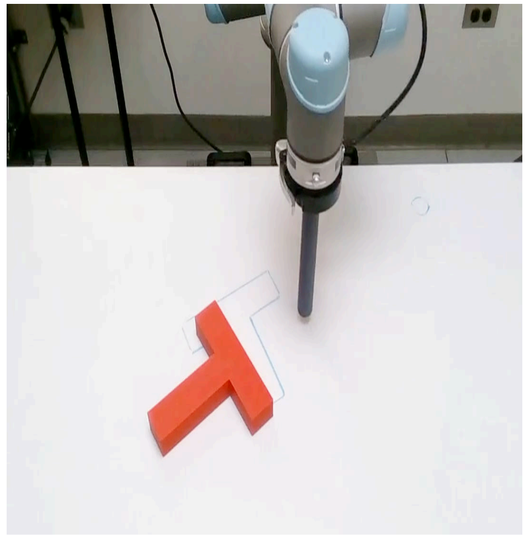
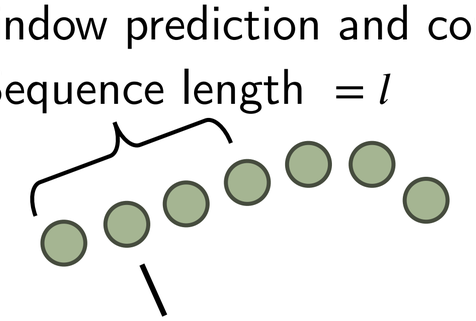


Image Diffusion [1]



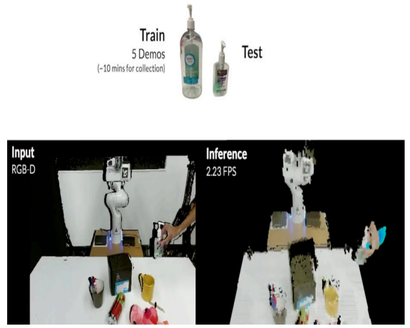
Policy Diffusion [2]

Rolling window prediction and control



$l = 1$, only predict next pose

Denoising process
for action $\boldsymbol{a}(\boldsymbol{s})$



$\boldsymbol{a}$ given
different $\boldsymbol{s}$ [3]

[1] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models."

[2] Chi, Cheng, et al. "Diffusion policy: Visuomotor policy learning via action diffusion."

[3] Shridhar, Mohit, Lucas Manuelli, and Dieter Fox. "Perceiver-actor: A multi-task transformer for robotic manipulation."

Image vs robot diffusion: same model, different target

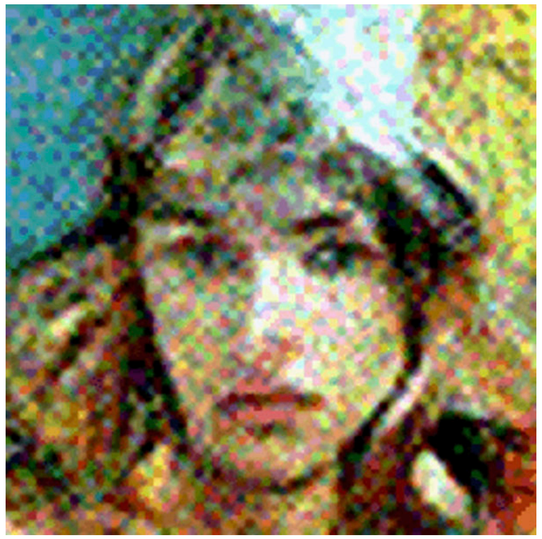
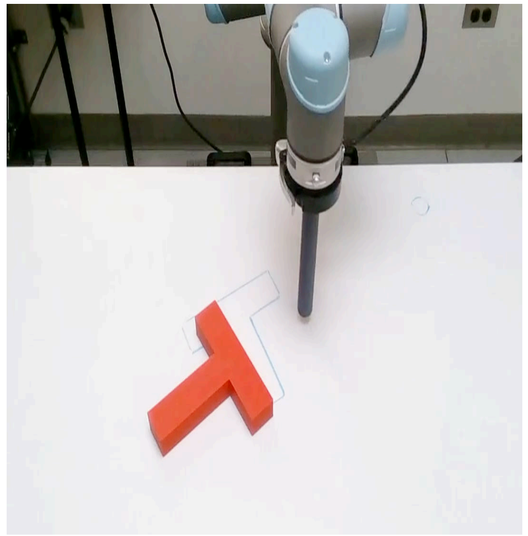


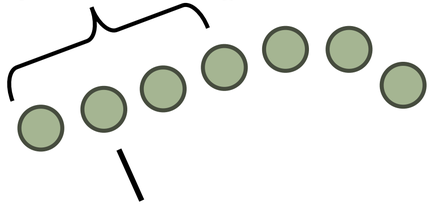
Image Diffusion [1]



Policy Diffusion [2]

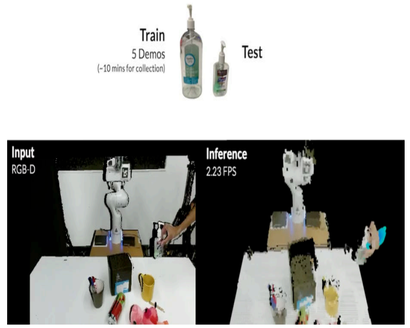
Rolling window prediction and control

Sequence length $= l$



$l = 1$, only predict next pose

Denoising process
for action $\boldsymbol{a}(\boldsymbol{s})$



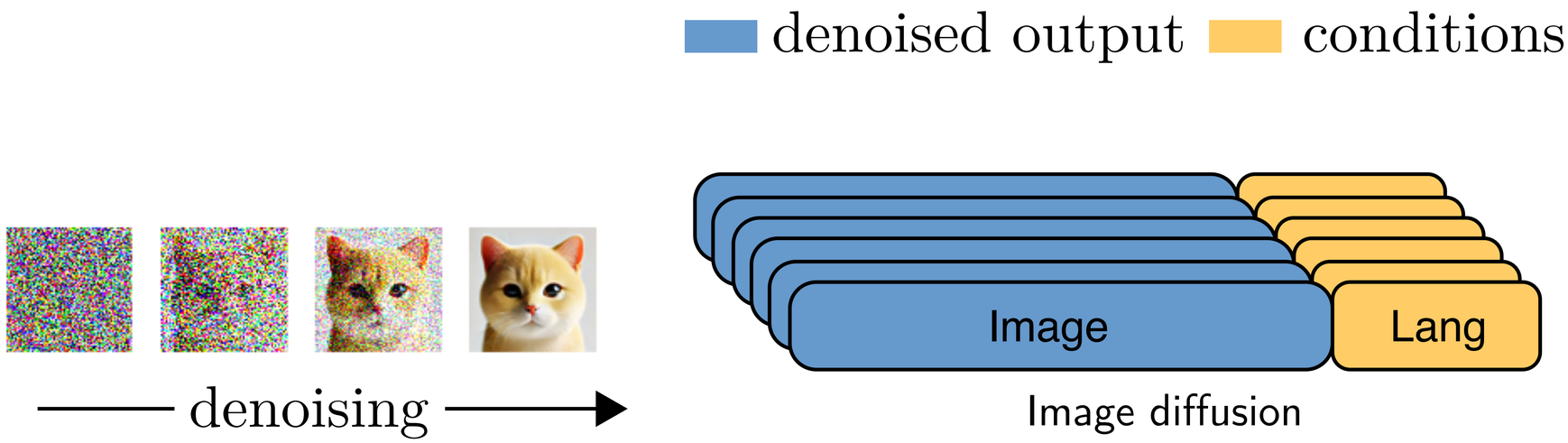
$\boldsymbol{a}$ given
different $\boldsymbol{s}$ [3]

[1] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models."

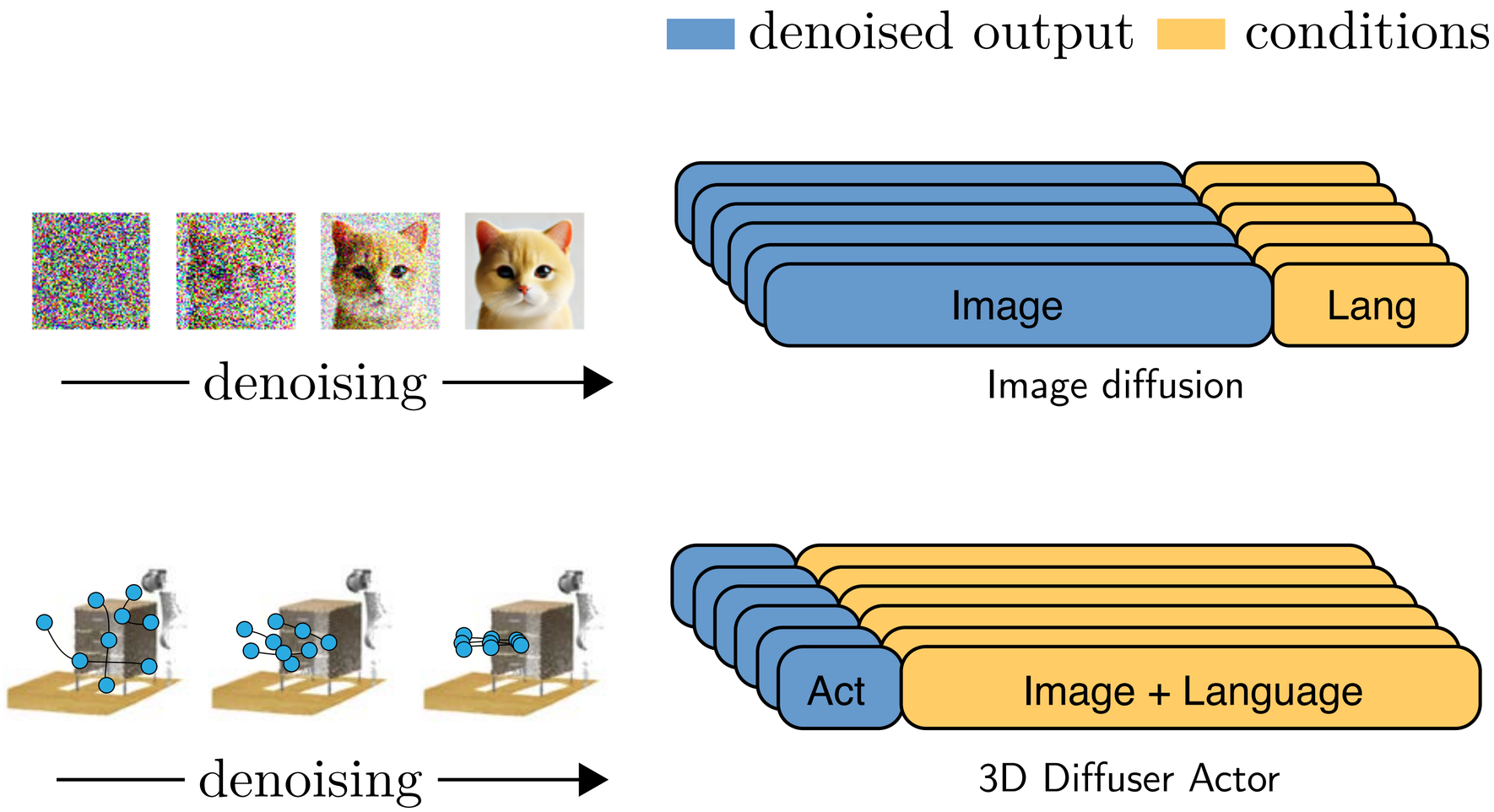
[2] Chi, Cheng, et al. "Diffusion policy: Visuomotor policy learning via action diffusion."

[3] Shridhar, Mohit, Lucas Manuelli, and Dieter Fox. "Perceiver-actor: A multi-task transformer for robotic manipulation."

Batched Training in Diffusion Models

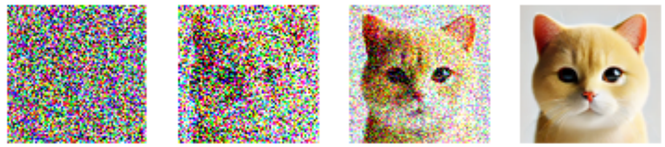


Batched Training in Diffusion Models



Batched Training in Diffusion Models

denoised output conditions



denoising

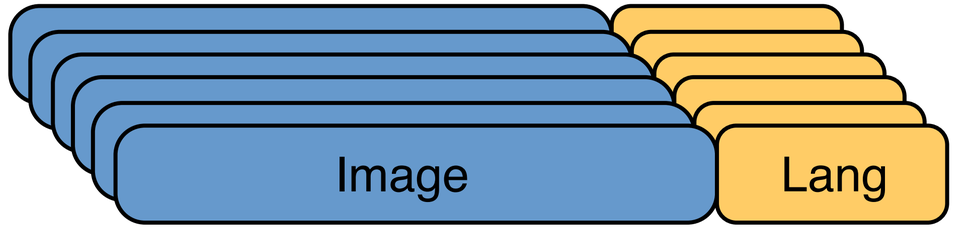
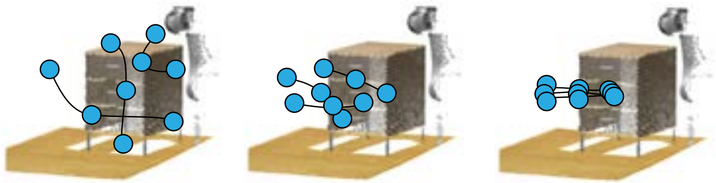
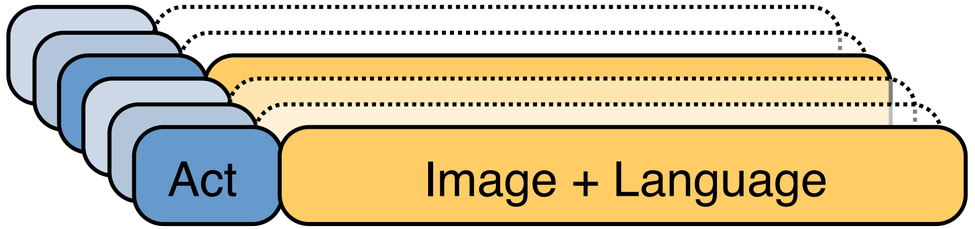


Image diffusion



denoising



3D Diffuser Actor

Two-Level Batch for Action Diffusion

$$a_k = \sqrt{\alpha_k}\, a_0 + \sqrt{1-\alpha_k}\,\epsilon$$

s point clouds, language instructions,
current joint configs ...

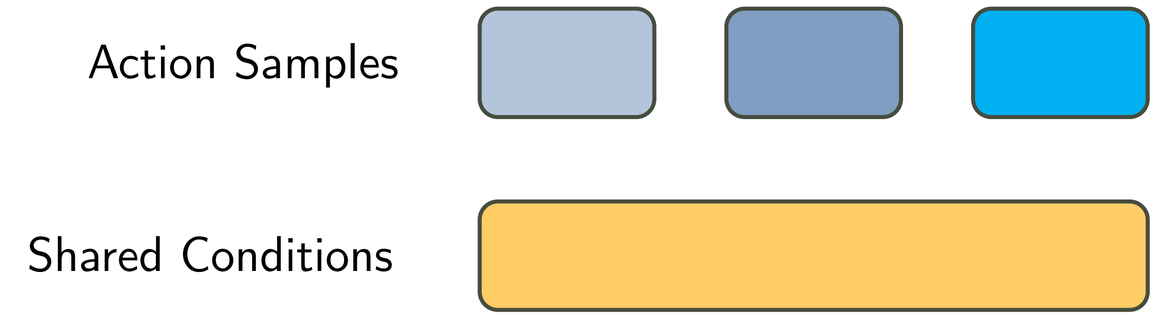
Level-1: We first sample B independent state–action pairs

$$\left\{ \left(s^{(i)}, a_0^{(i)} \right) \right\}_{i=1}^{B}, \left(s^{(i)}, a_0^{(i)} \right) \sim q(a, s)$$

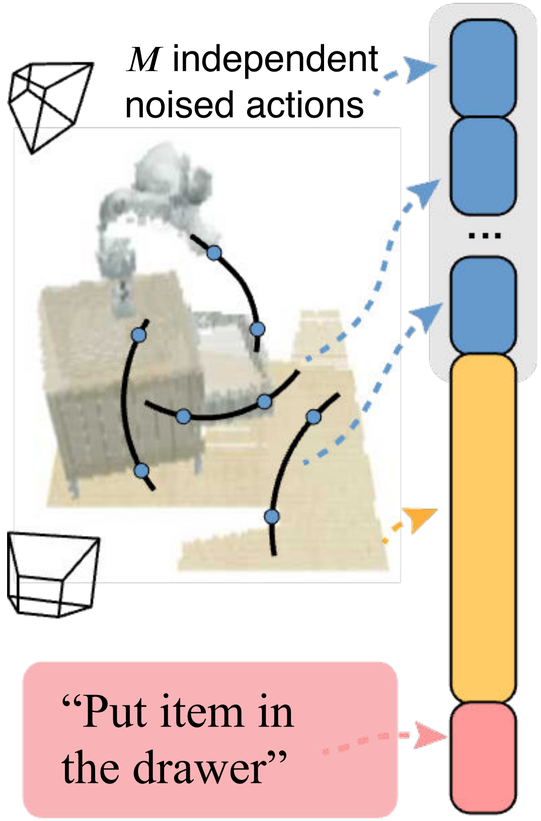
Level-2: For each of those B, we independently draw M step-noise pairs,

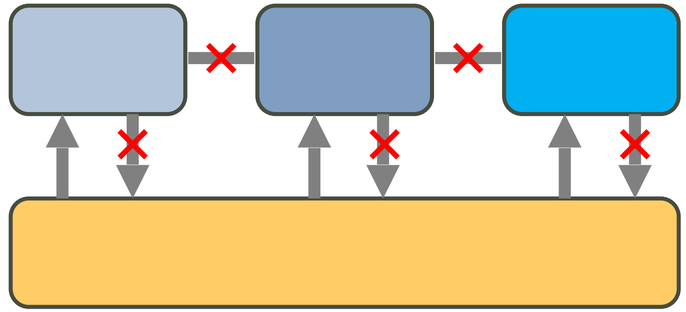
$$\left\{ \left(k^{(i,j)}, \epsilon^{(i,j)} \right) \right\}_{j-1}^{M}, k^{(i,j)} \sim \mathcal{U}(1, K), \epsilon^{(i,j)} \sim \mathcal{N}(0, \mathbf{I}) .$$

Two-Level Batch for Action Diffusion



Masked Attention Protects Action Sample Independence

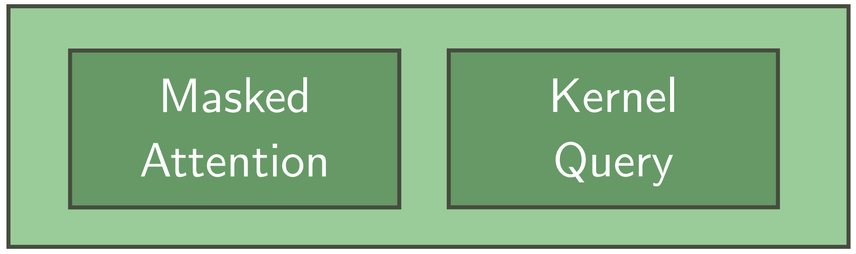


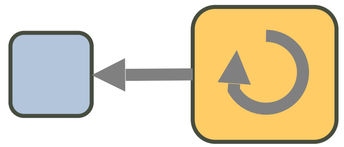


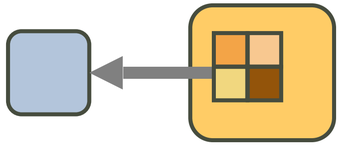
(i) An action sample attends to itself and shared conditions, but not to other action samples

(ii) shared conditions do not attend back to action samples.

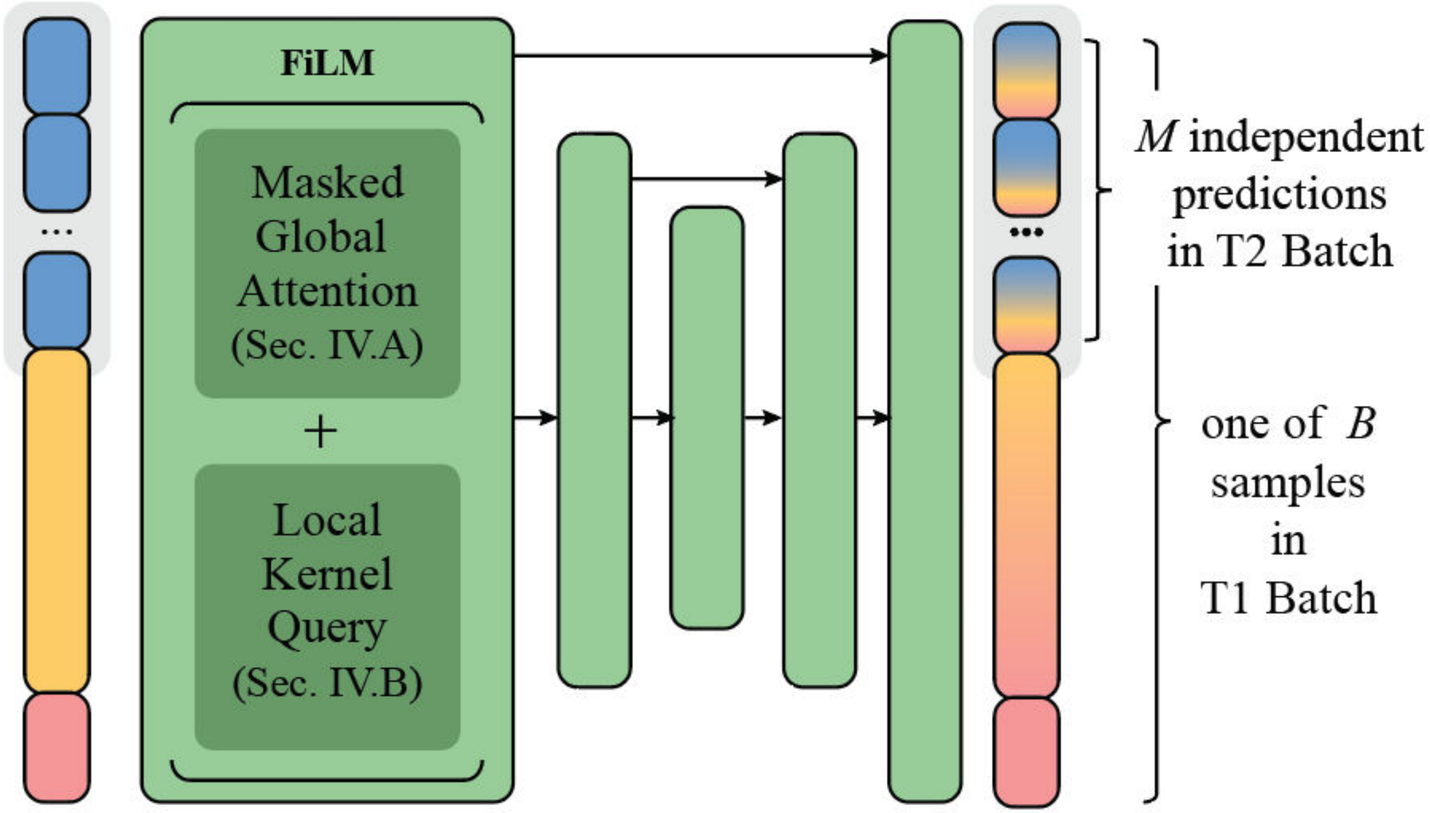
Two modules can perform such a "Non-invasive extraction"

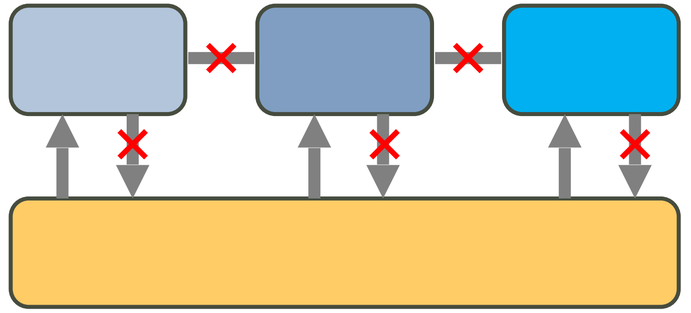



Cross attention


Kernel query

Masked Attention Protects Action Sample Independence

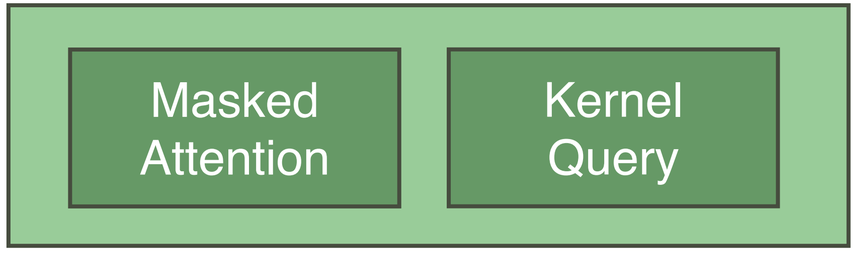


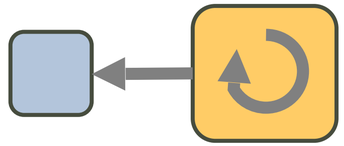


(i) An action sample attends to itself and shared conditions, but not to other action samples

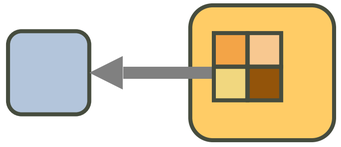
(ii) shared conditions do not attend back to action samples.

Two modules can perform such a "Non-invasive extraction"



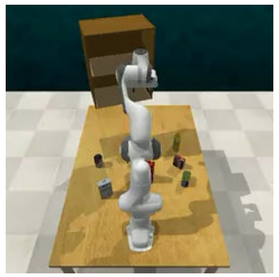
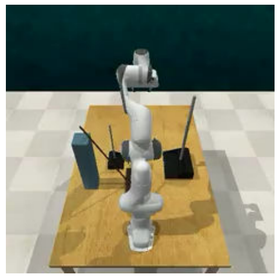
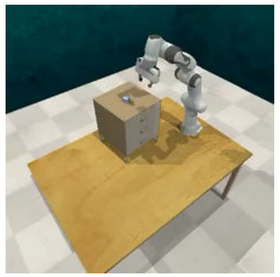


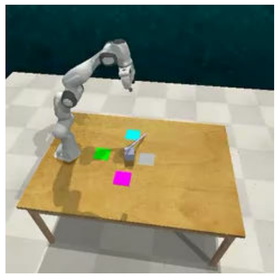
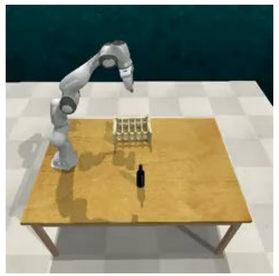
Cross attention



Kernel query

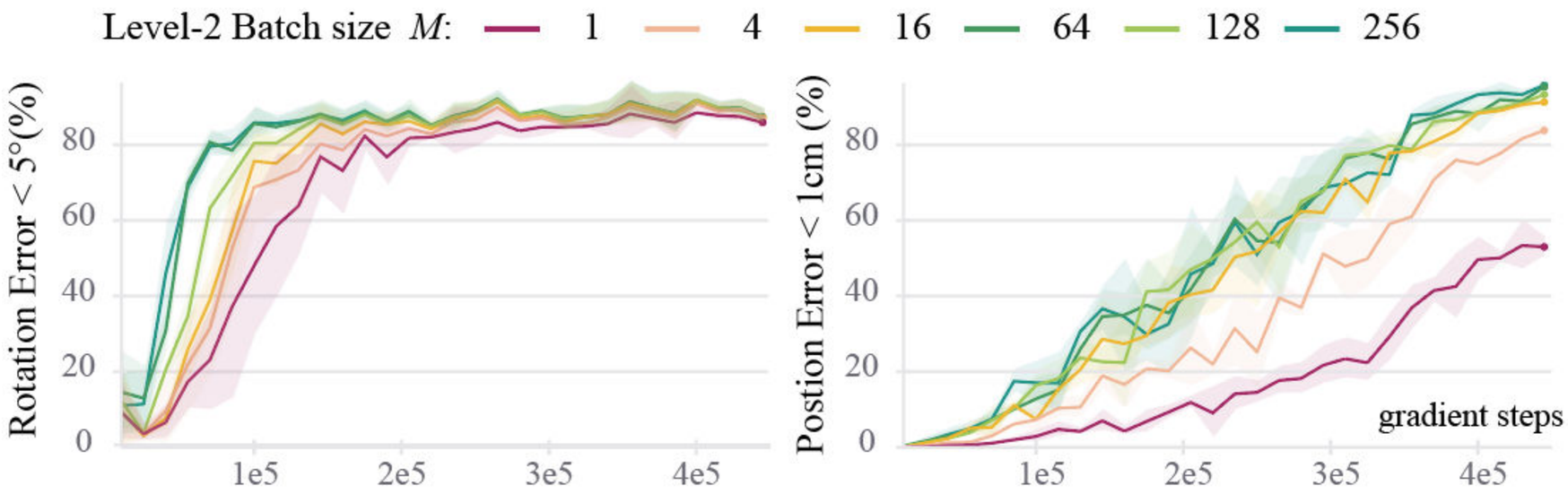
Training an 18-in-1 multi-task model for RL-Bench

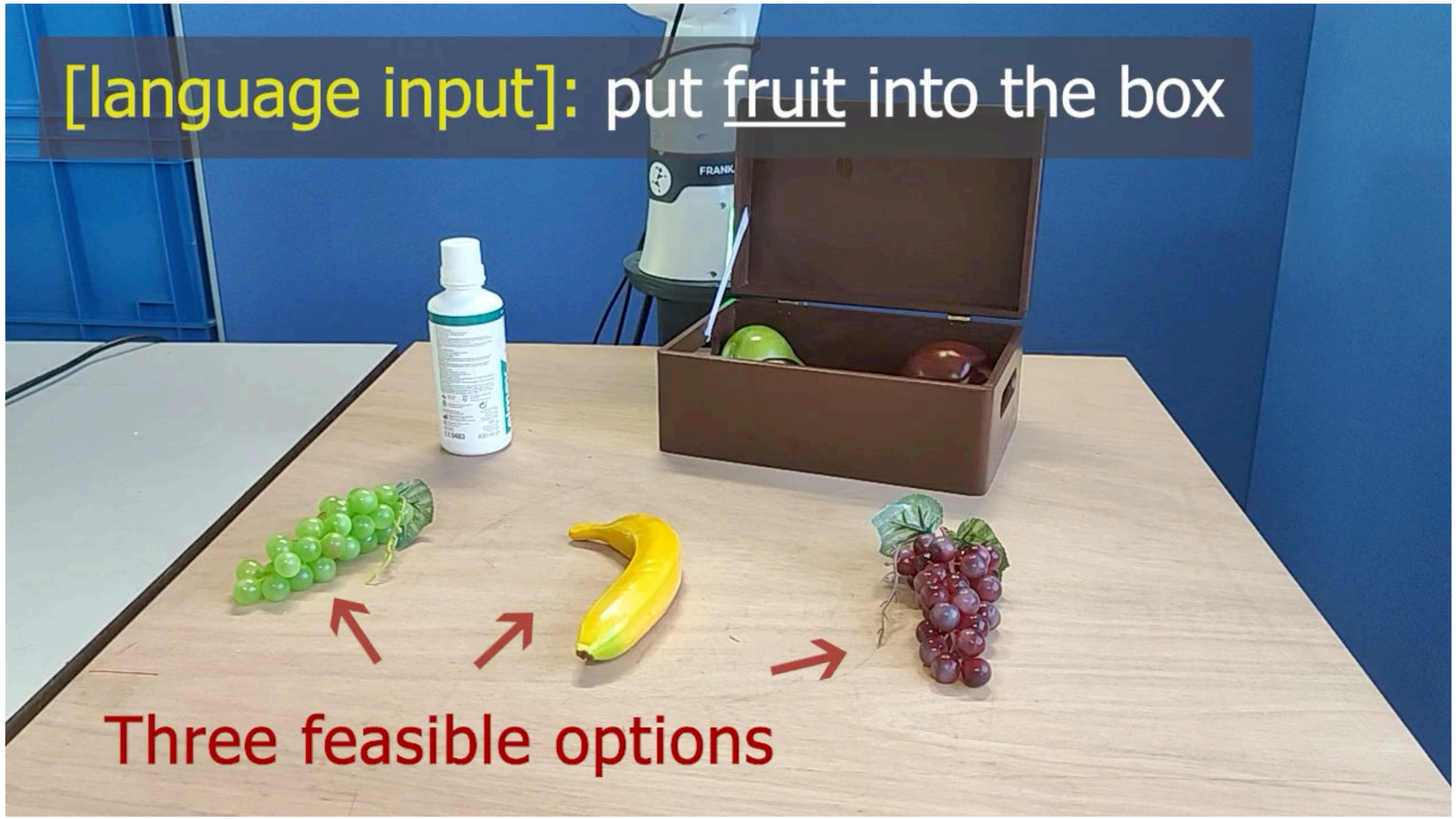
Method	Avg. Suc. (%)	Norm. Time	Memory (GB)	Reported Hardware
PerAct	49.4	128	128	V100×8×16 days
RVT	62.9	8	128	V100×8×1 day
Act3D	63.2	40	128	V100×8×5 days
RVT-2	81.4	6.6	128	V100×8×20 hours
3D-Dif-Actor	81.3 (100%)	39 (100%)	240 (100%)	A100×6×6 days
SAM2Act	86.8	8.3	160	H100×8×12 hours
Mini-diffuser	77.6 (**95.4%**)	**1.9 (4.8%)**	**16 (6.6%)**	4090×13 hours or A100×1 day

Efficiency of Level-2 Batch



Level-1 batches: B Level-2 batches: M	Memory Cost	Time per Gradient Step	Avg. Succ. after 1e5 Steps
B=100 M=64	102.2%	106.3%	78.3
B=100 M=1	100%	100%	44.1
B=200 M=1	188.8%	176.6%	50.8

Train a multi-task Diffuser Actor in the Realworld



https://mini-diffuse-actor.github.io

Mini Diffuser: Fast Multi-task Diffusion Policy Training Using Two-level Mini-batches

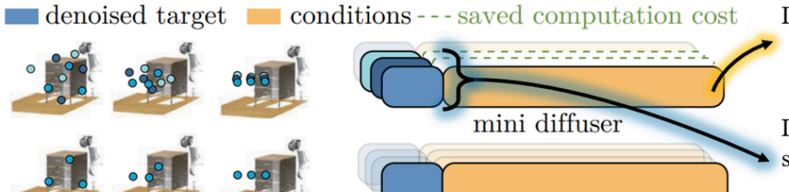
Yutong Hu[1], Pinhao Song[1], Kehan Wen[2], Renaud Detry[1]

[1]KU Leuven [2]ETH Zurich

Paper | Code | Checkpoints | WandB Logs | Real World

Abstract

We introduce **Mini-Diffuser**, a method for training **multi-task robot policies** that can perform a variety of tasks using **vision and language as input**—while training significantly faster and using far less memory than previous approaches. The key insight comes from comparing how **diffusion models** are used in different domains. In image generation, diffusion models refine **high-dimensional pixel data**. In contrast, robot actions are much simpler, typically involving only **3D positions, rotations, and gripper states**. However, the **conditions**—such as images and language instructions—remain high-dimensional. Mini-Diffuser takes advantage of this asymmetry. Instead of generating one action per input, it generates **multiple action samples** for the same vision-language input. This allows the model to train **over 20× more efficiently** with **minimal extra cost**. To support this strategy, we introduce **lightweight architectural changes** that prevent interference between samples during training. Mini-Diffuser offers a **simple, fast, and effective recipe** for training generalist robot policies at scale.

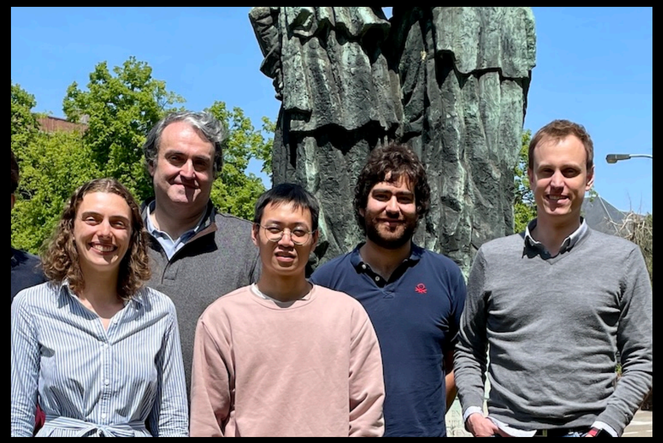


In Level-1, we sample B condition-action pairs independently

$$\{(s_i, a_0^{(i)})\}_{i=1}^{B}, \quad (s_i, a_0^{(i)}) \sim q(a, s).$$

In Level-2, for each Level-1 pair, we independently sample M noise-timestep pairs

Equivariant Volumetric Grasping



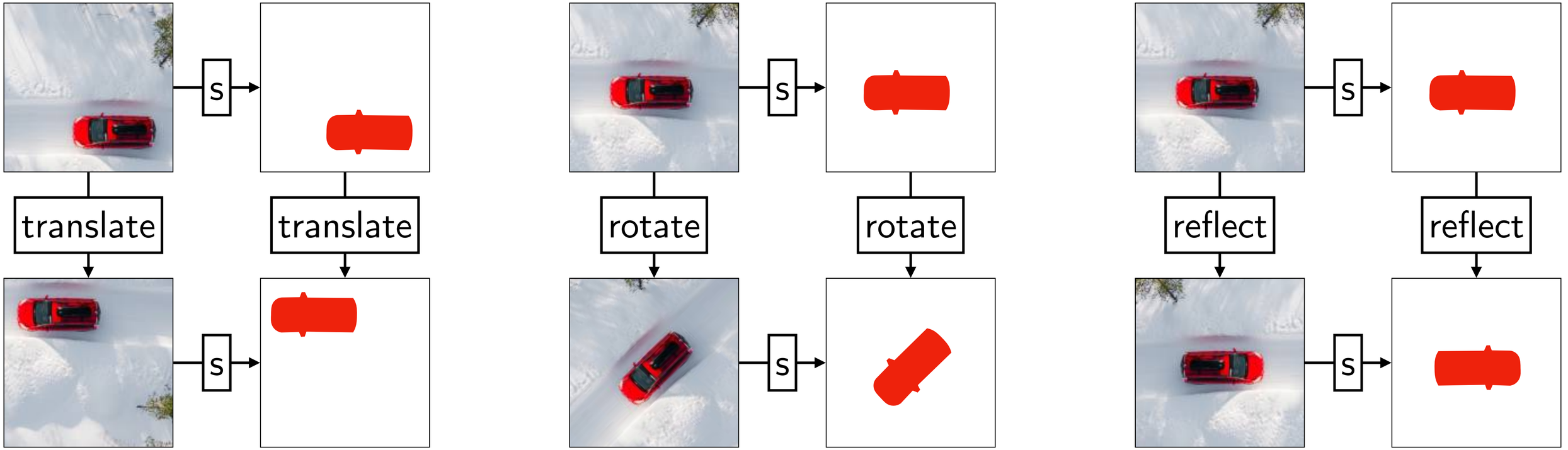
P. Song, Y. Hu, P. Li, and R. Detry

KU LEUVEN

ROMANDIC – Feb 9, 2025

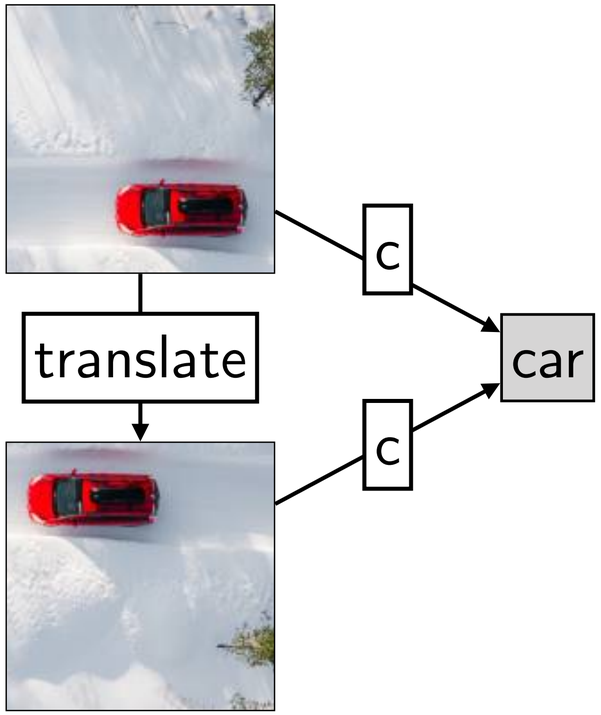
A model is equivariant if its output responds predictably to transformations of its input

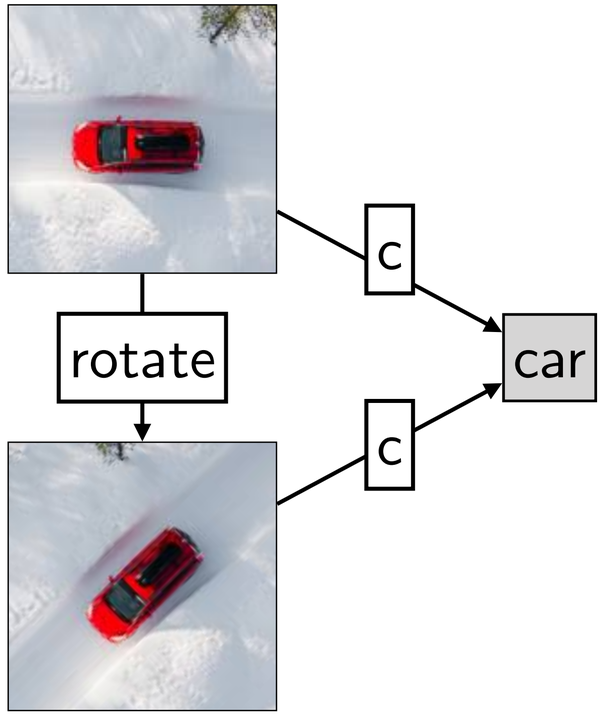
A trivial way of responding predictably: responding *identically*

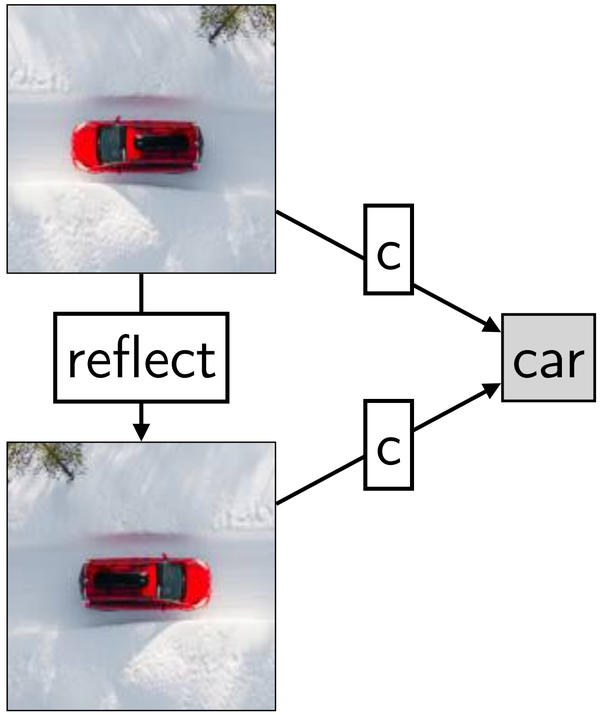


s = image segmentation model

A model is invariant if its output is immutable to transformations of its input

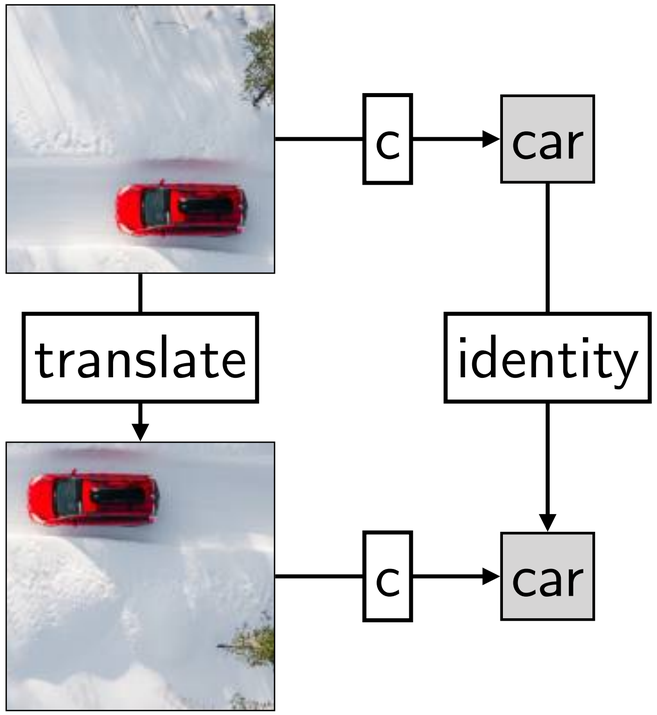


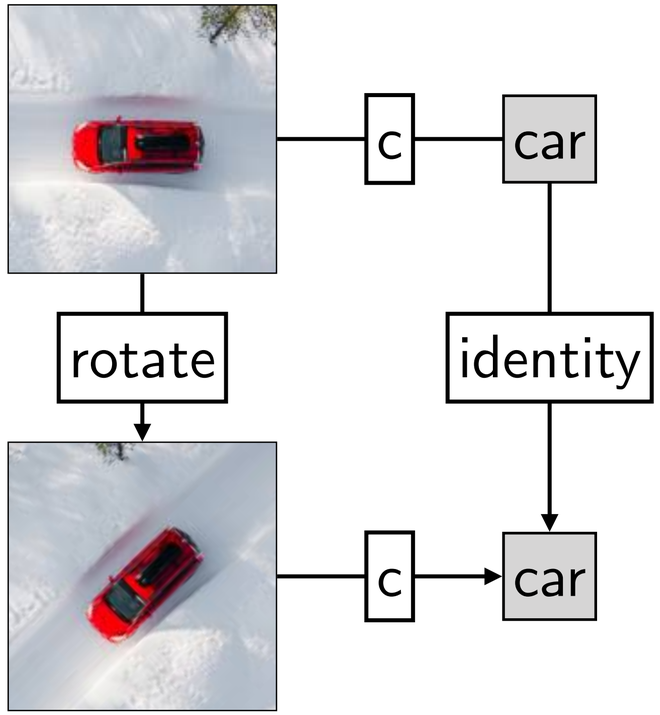


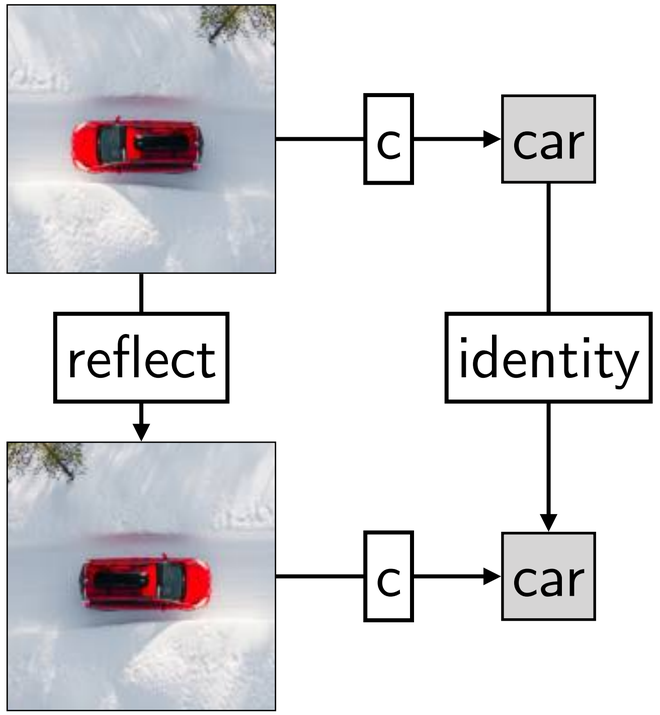


c = image classification model

Invariance is a special case of equivariance,
where the output transformation is the identity transformation







c = image classification model

Data-driven models can achieve equivariance (a) through exposure to tons of data, or (b) through architectural design

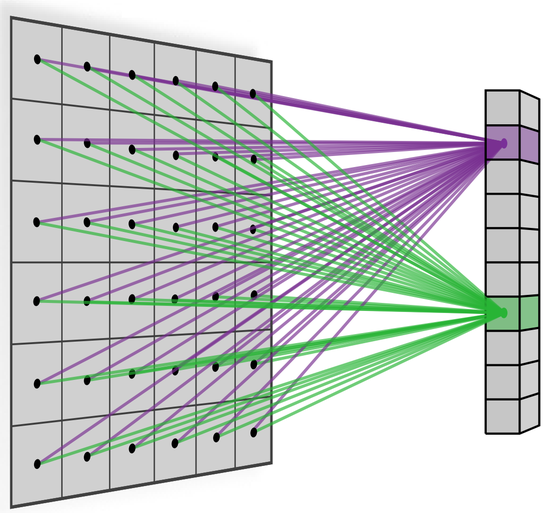
(a)
Vanilla model (e.g., MLP)
trained on:

(b)
Architecturally-equivariant model
trained on:

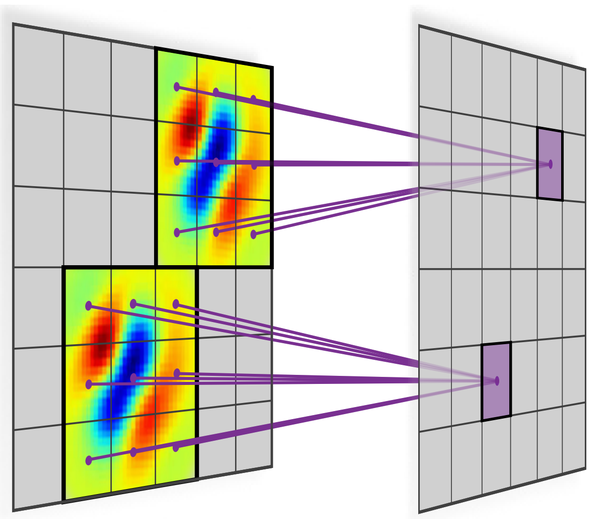
Better *sample efficiency*

A canonical example of a model designed for translation equivariance: the convolutional neural network

(a)
MLP (no architectural equivariance)

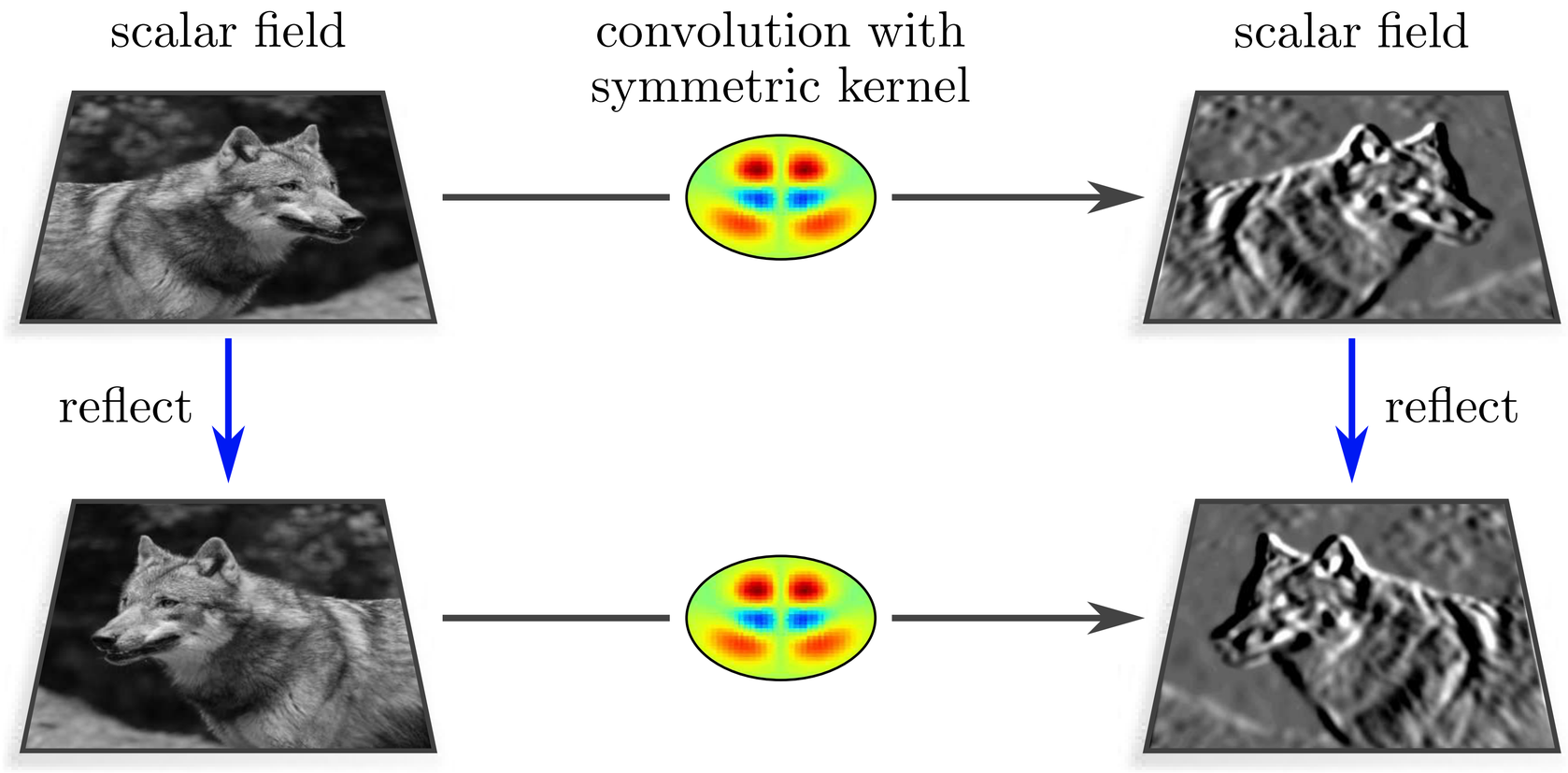


(b)
CNN (architectural equivariance to translations)



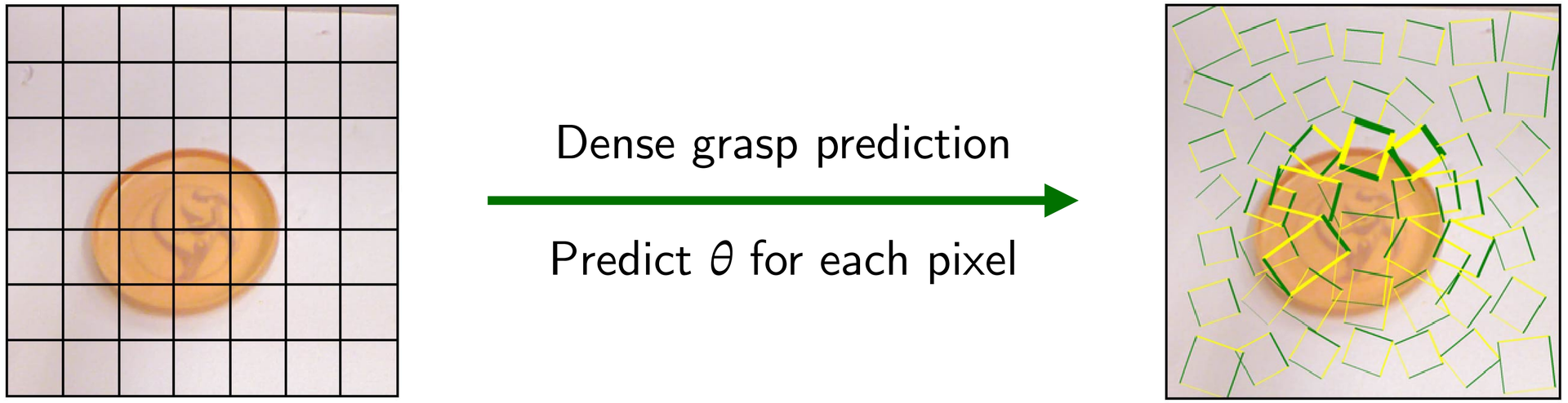
[1] M. Weiler, P. Forre, E. Verlinde, and M. Welling. Equivariant and Coordinate Independent Convolutional Networks. 2023.

DNN building blocks that provide equivariance to rotations or reflections readily exist in software libraries. They are generally referred to as steerable kernels.



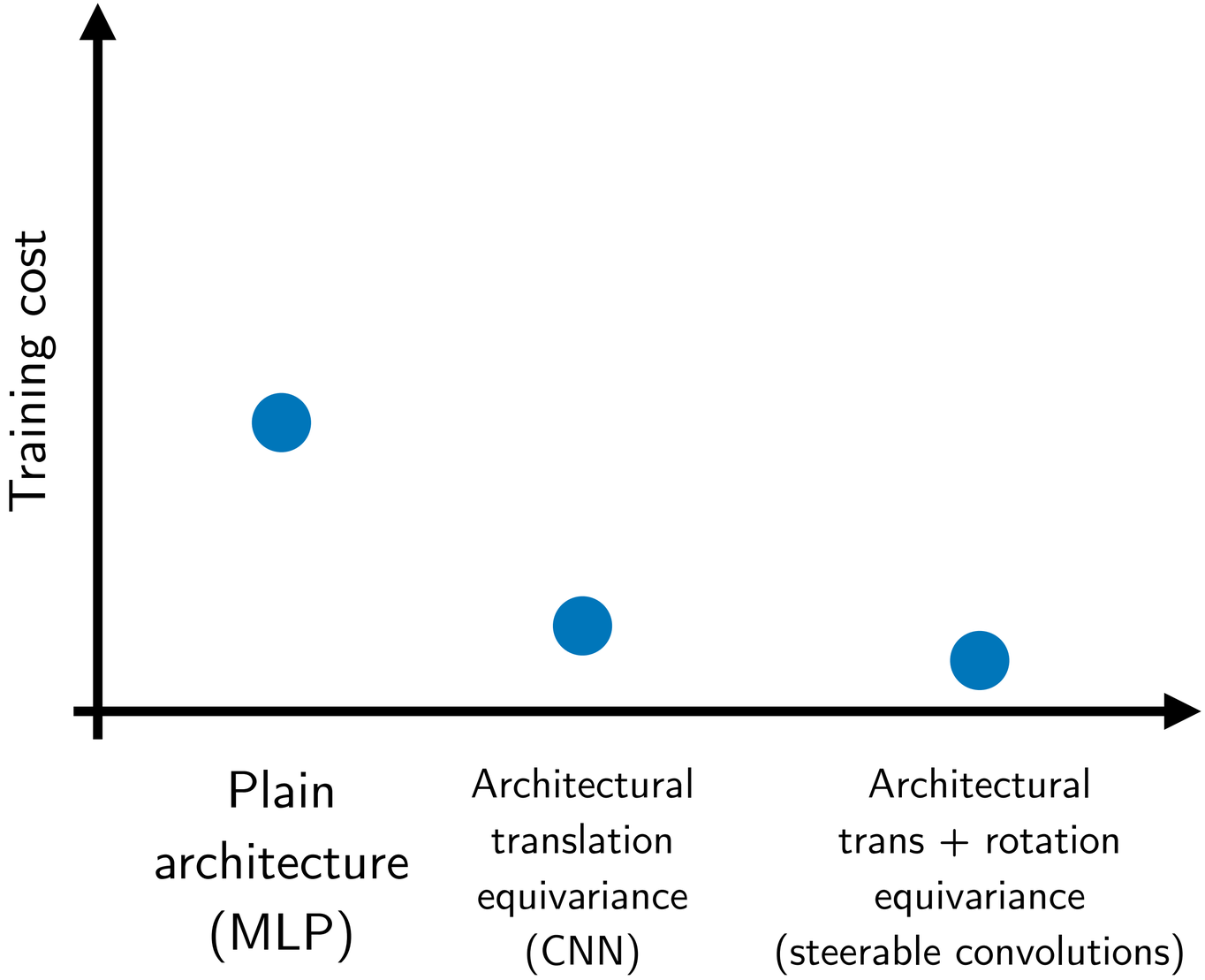
[1] M. Weiler, P. Forre, E. Verlinde, and M. Welling. Equivariant and Coordinate Independent Convolutional Networks. 2023.

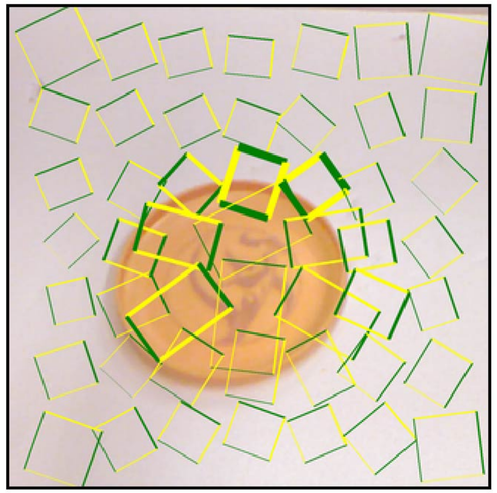
Dense grasp prediction is similar in spirit to dense image processing: it predicts parameters for each pixel of an input image



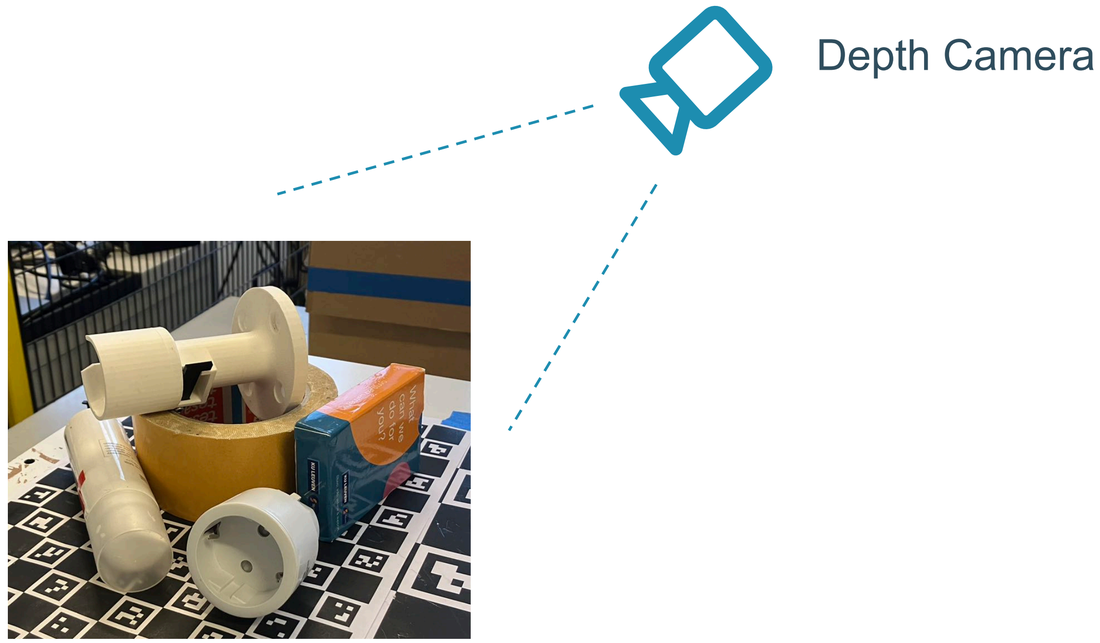
Redmon, J., & Angelova, A. Real-time grasp detection using convolutional neural networks. ICRA 2015.

Architecturally-equivariant models improve sample efficiency and lower training costs



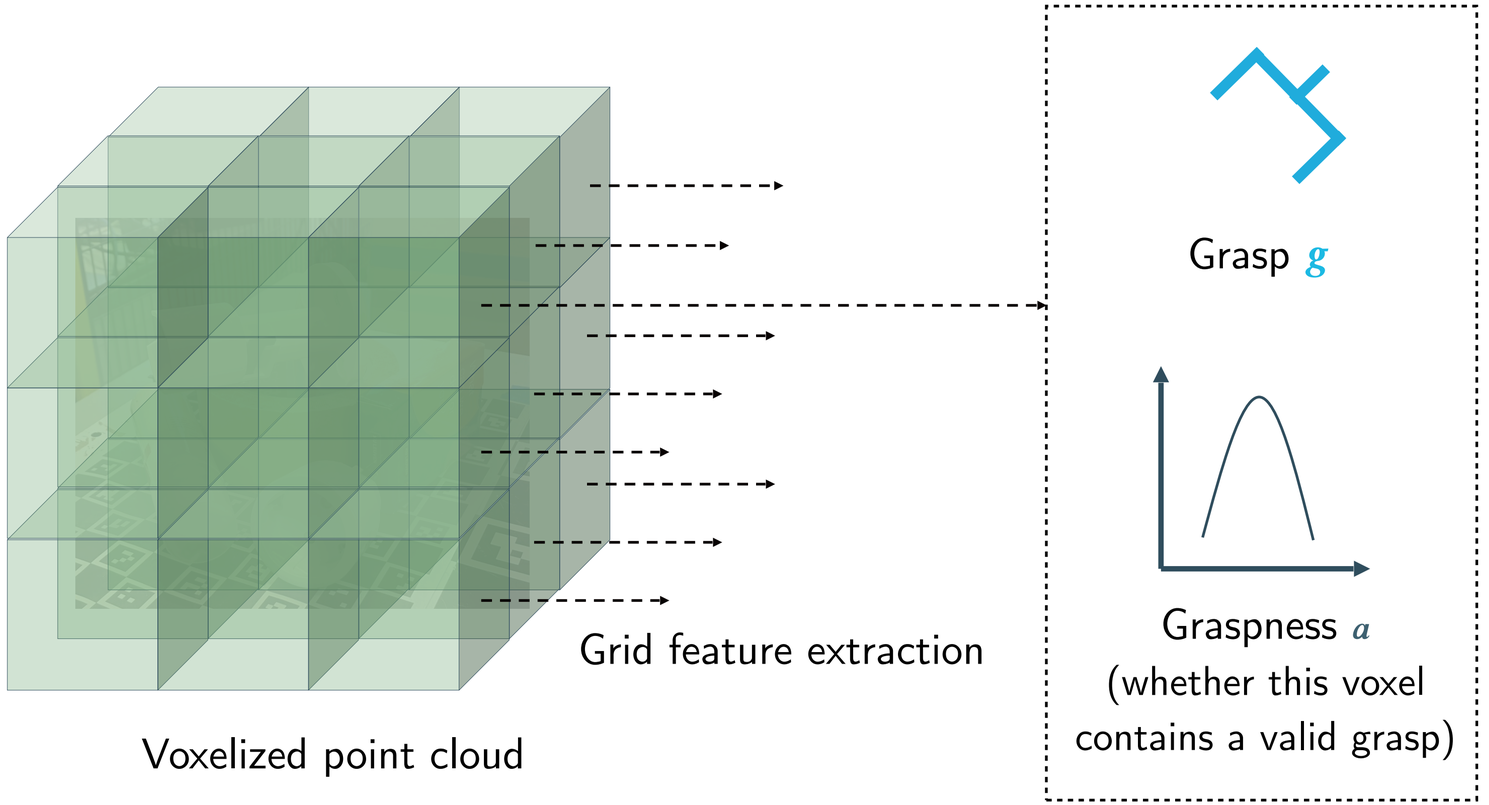


Volumetric Grasping is a popular approach to grasp planning that applies principles of 2D computer vision to 6D grasping



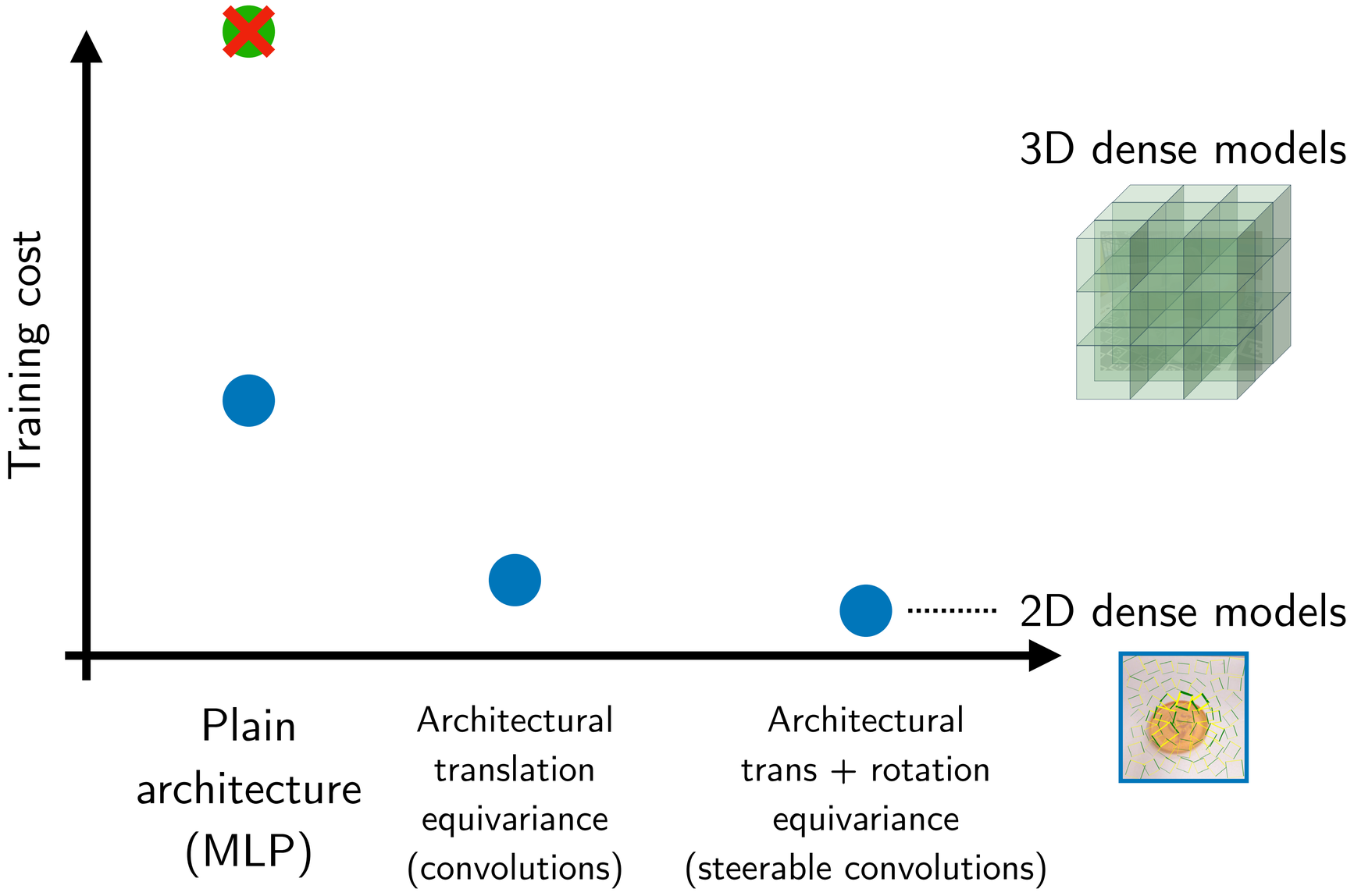
Breyer, Michel, et al. "Volumetric grasping network: Real-time 6 dof grasp detection in clutter." Conference on Robot Learning. PMLR, 2021.

Volumetric Grasping is a popular approach to grasp planning that applies principles of 2D computer vision to 6D grasping



Breyer, Michel, et al. "Volumetric grasping network: Real-time 6 dof grasp detection in clutter." Conference on Robot Learning. PMLR, 2021.

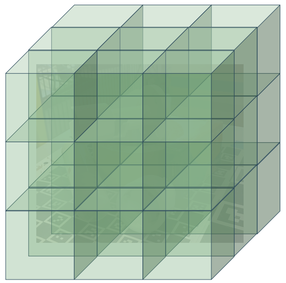
Volumetric grasping with no architectural equivariance has a prohibitively low sample efficiency



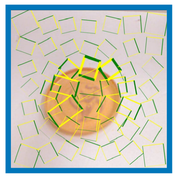
Volumetric grasping with 3D (translation-equivariant) CNNs is a promising concept

Breyer, Michel, et al. "Volumetric grasping network: Real-time 6 dof grasp detection in clutter." Conference on Robot Learning. PMLR, 2021.

3D dense models



2D dense models



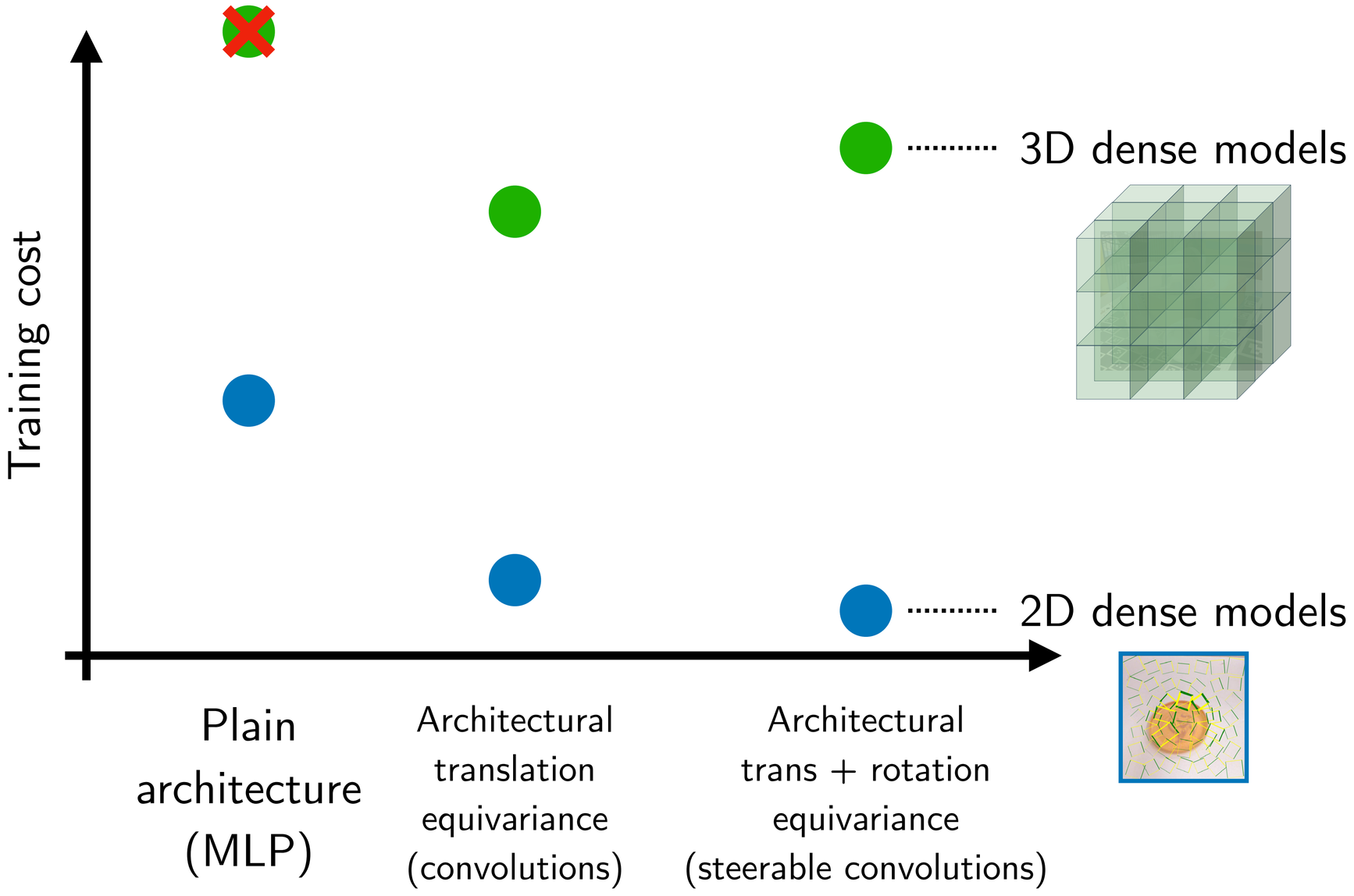
Training cost

Plain architecture (MLP)

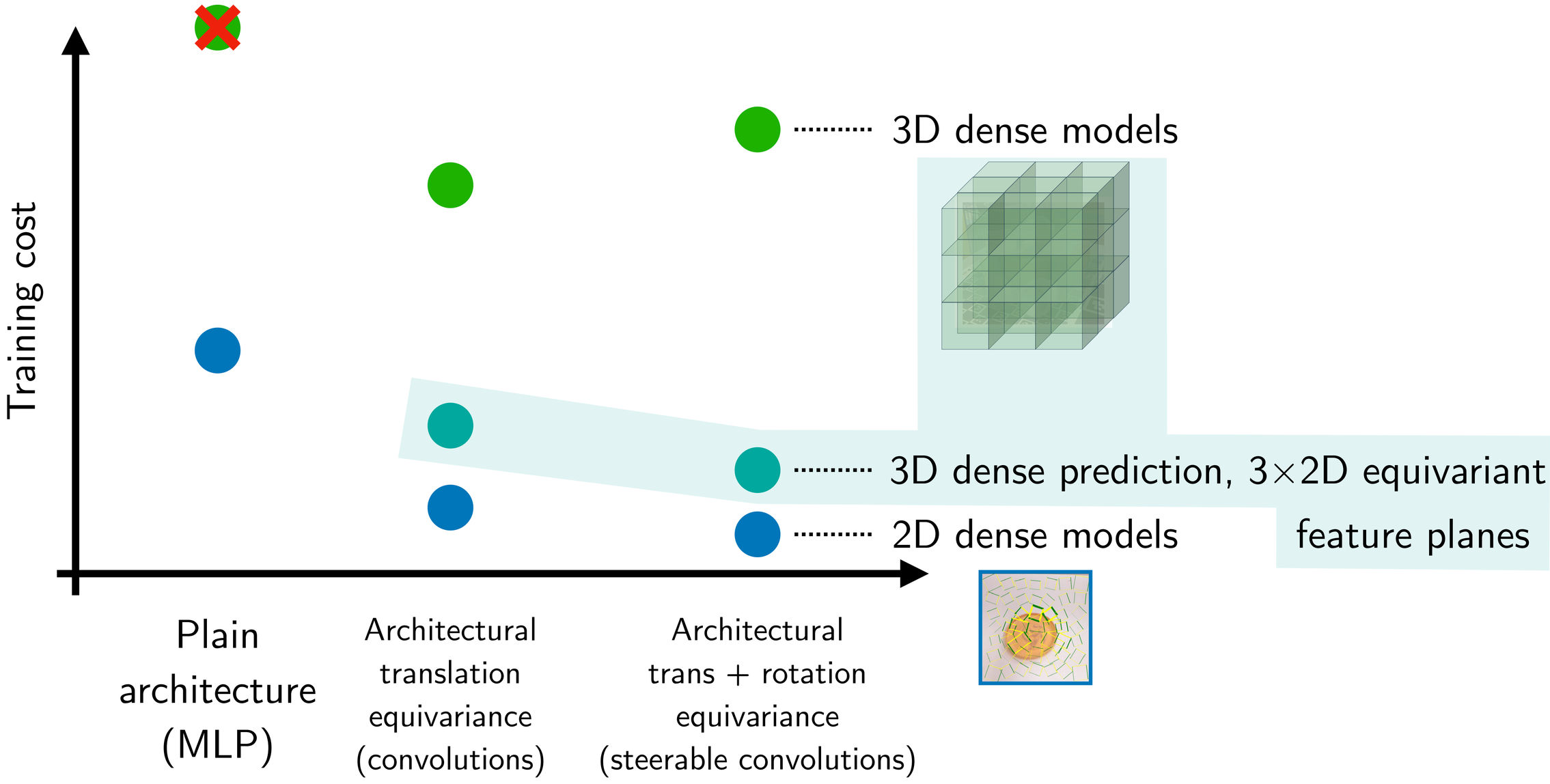
Architectural translation equivariance (convolutions)

Architectural trans + rotation equivariance (steerable convolutions)

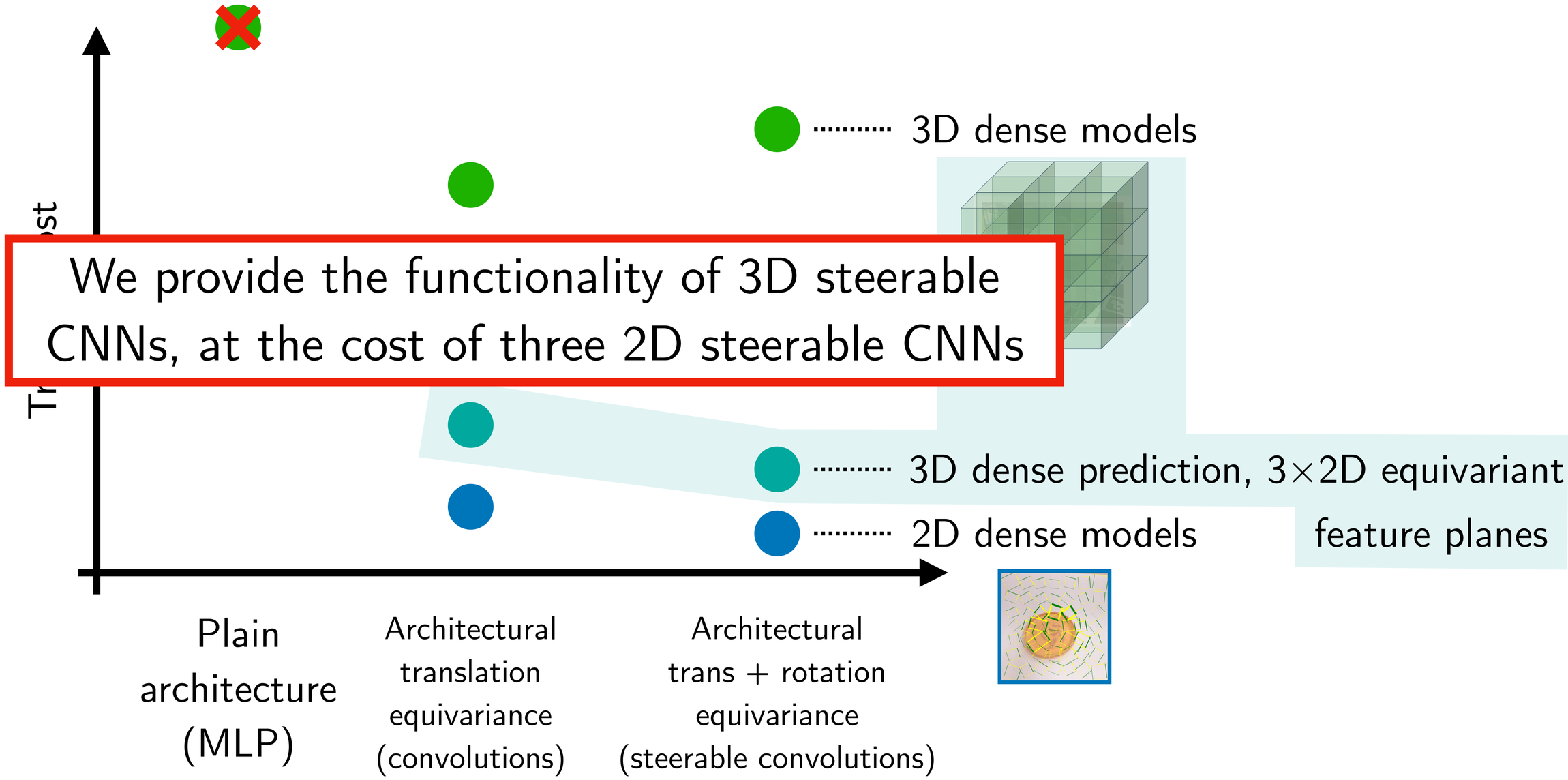
The increased sample efficiency brought by 3D steerable CNNs is insufficient to justify their computational cost



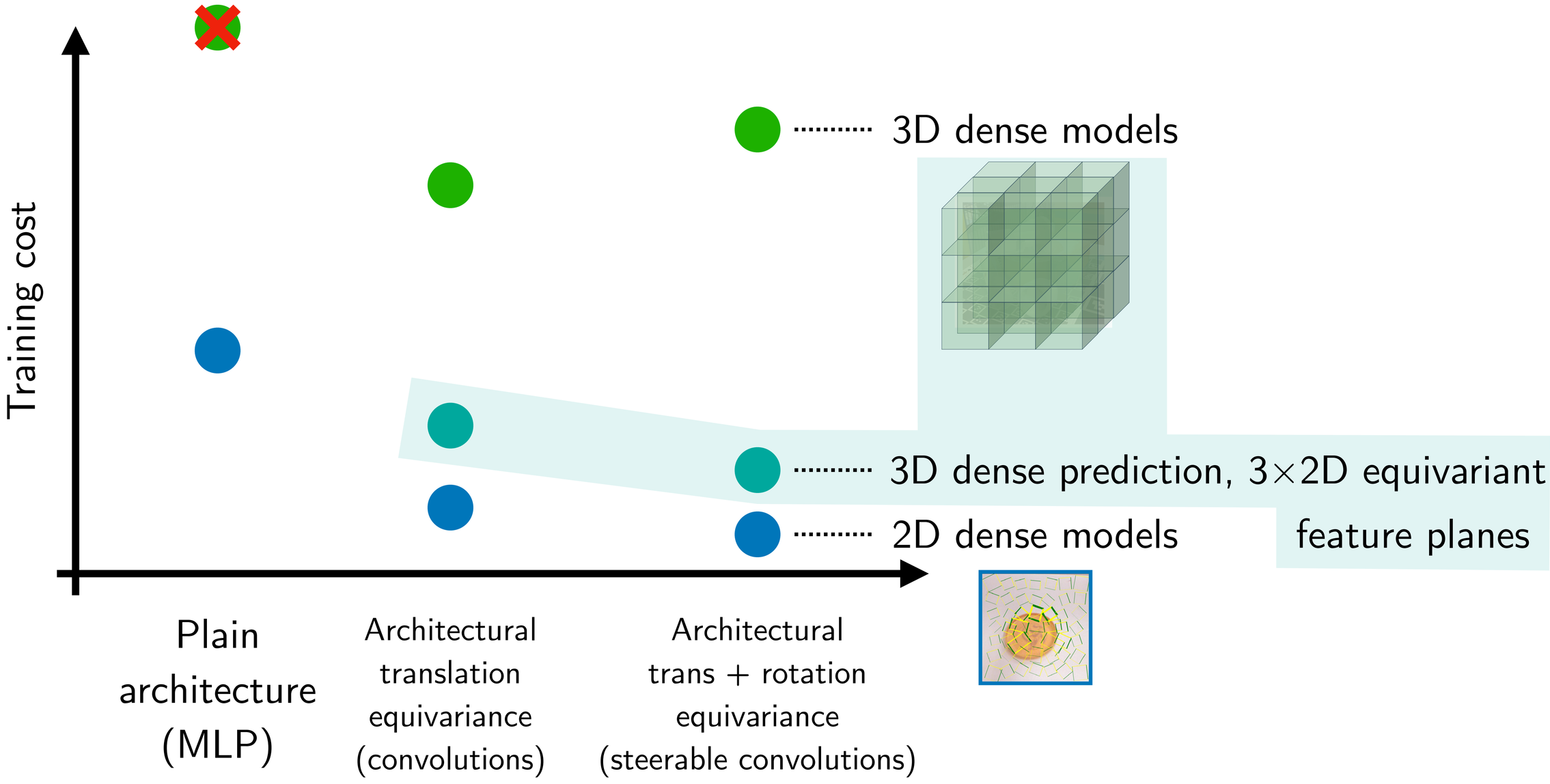
We propose to achieve 3D translation-rotation equivariance
by factorizing the input voxel grid into three orthogonal planar grids, and
designing equivariant features in these three planes.



We propose to achieve 3D translation-rotation equivariance
by factorizing the input voxel grid into three orthogonal planar grids, and
designing equivariant features in these three planes.

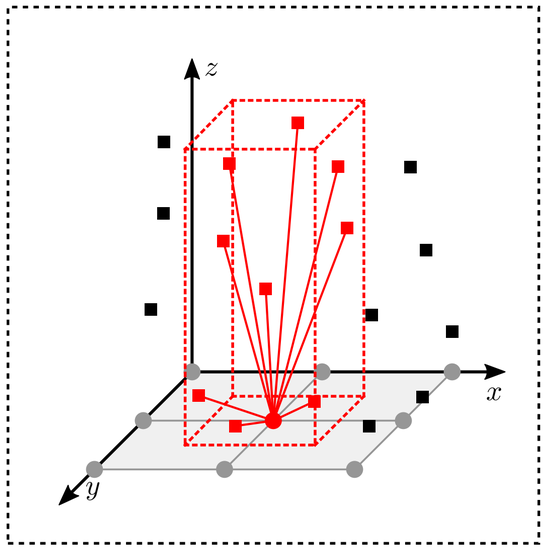


We propose to achieve 3D translation-rotation equivariance
by factorizing the input voxel grid into three orthogonal planar grids, and
designing equivariant features in these three planes.

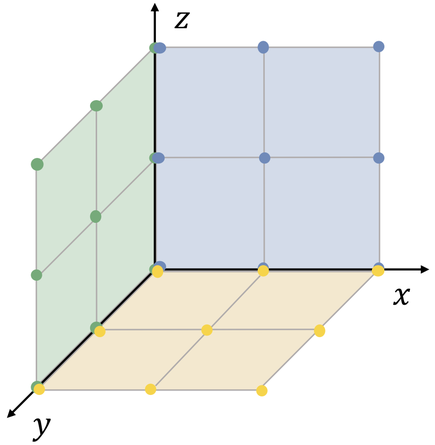


We factorize 3D data into a tri-plane feature grid

projection, aggregation



Projected 2D Feature Grids

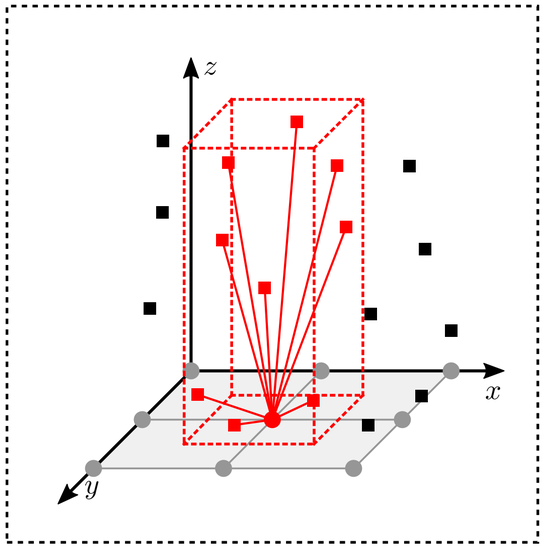


"Synergies Between Affordance and Geometry:
6-DoF Grasp Detection via Implicit Representations (GIGA)", Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, Y. Zhu

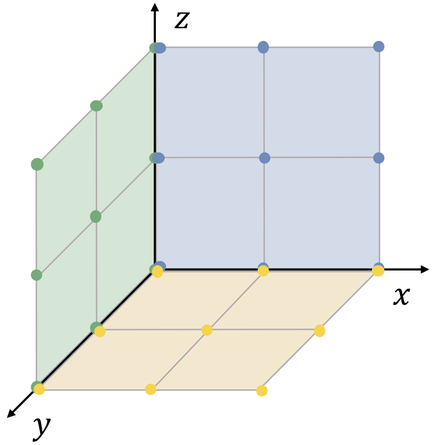
"Convolutional occupancy networks.", Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., & Geiger, A.

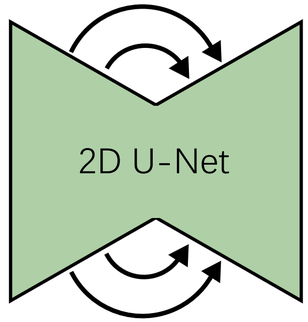
We extract rich features by applying a 2D UNet to each plane

projection, aggregation

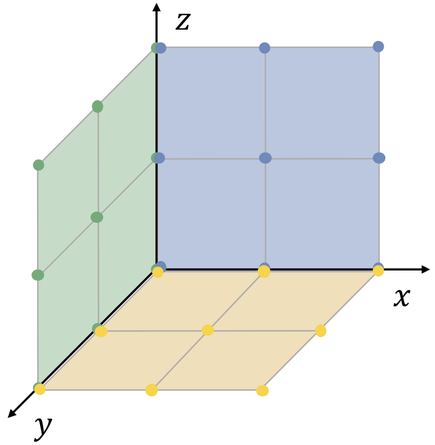


Projected 2D Feature Grids





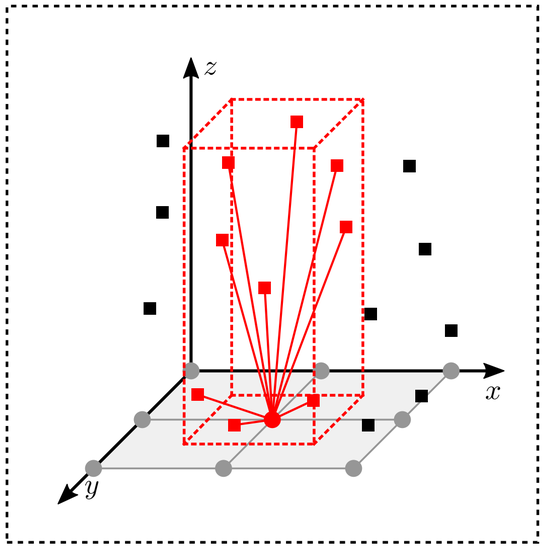
Tri-plane Feature Grids $\boldsymbol{c}$



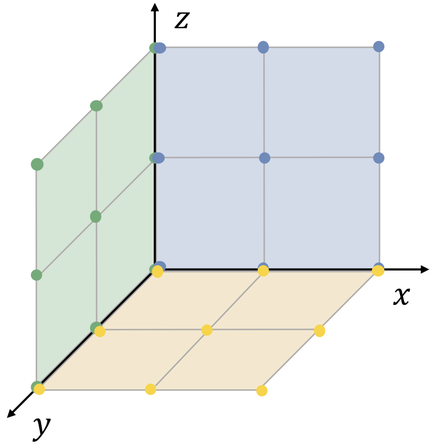
"Synergies Between Affordance and Geometry:
6-DoF Grasp Detection via Implicit Representations (GIGA)", Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, Y. Zhu
"Convolutional occupancy networks.", Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., & Geiger, A.

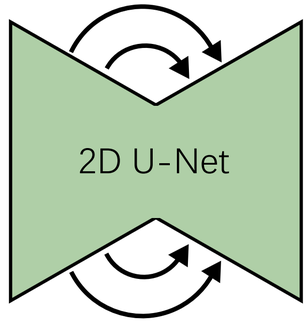
With bilinear interpolation, we can synthesize a feature at any given 3D point, allowing us to query the model in continuous 3D space

projection, aggregation

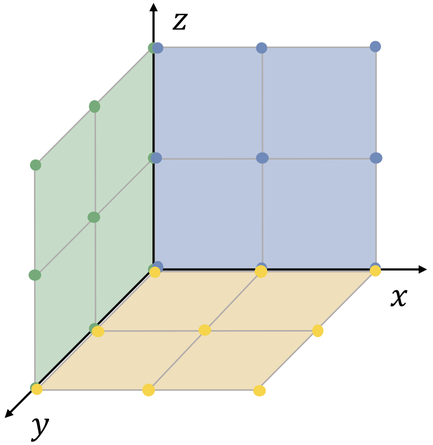


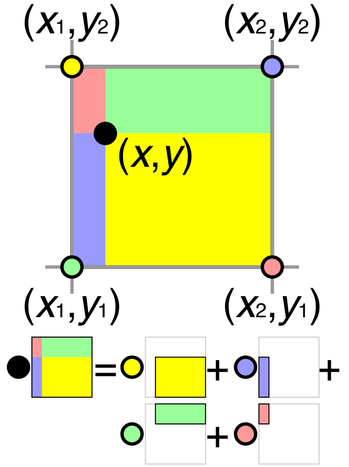
Projected 2D Feature Grids

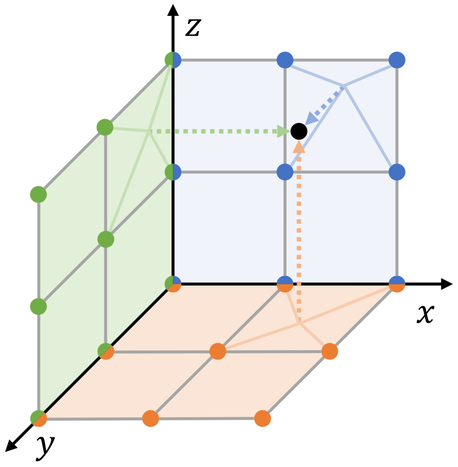




Tri-plane Feature Grids $\boldsymbol{c}$

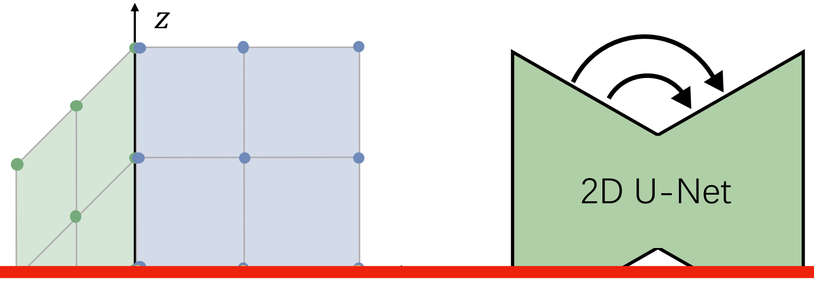




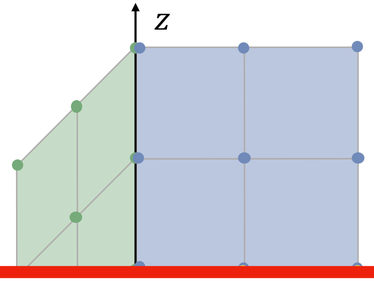


With bilinear interpolation, we can synthesize a feature at any given 3D point, allowing us to query the model in continuous 3D space

Projected 2D Feature Grids

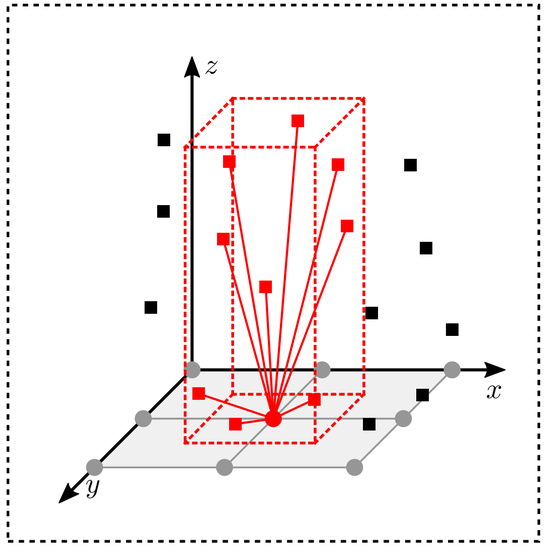


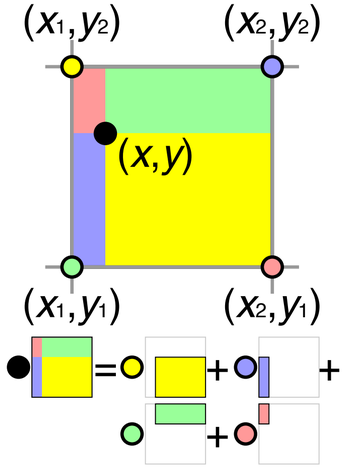
Tri-plane Feature Grids $\boldsymbol{c}$

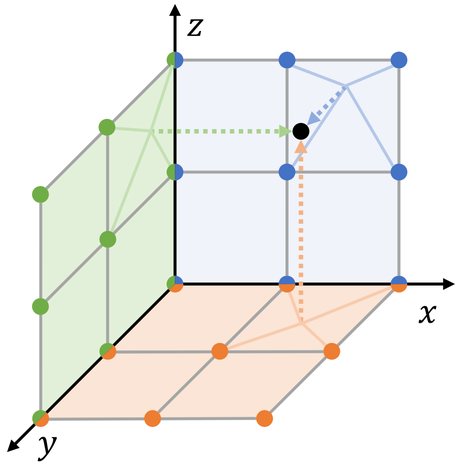


The 3D feature built from interpolated planar features is equivariant to 3D translations

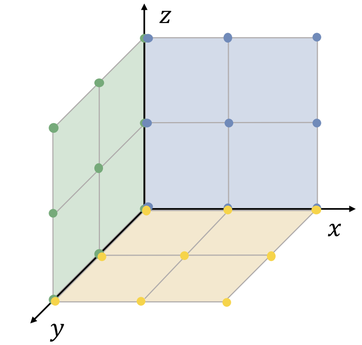
projection, aggregation

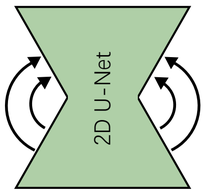


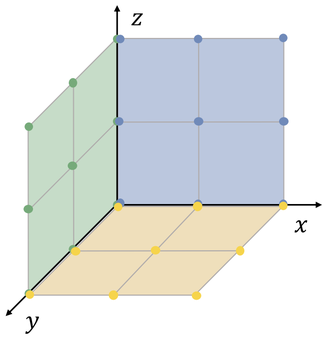


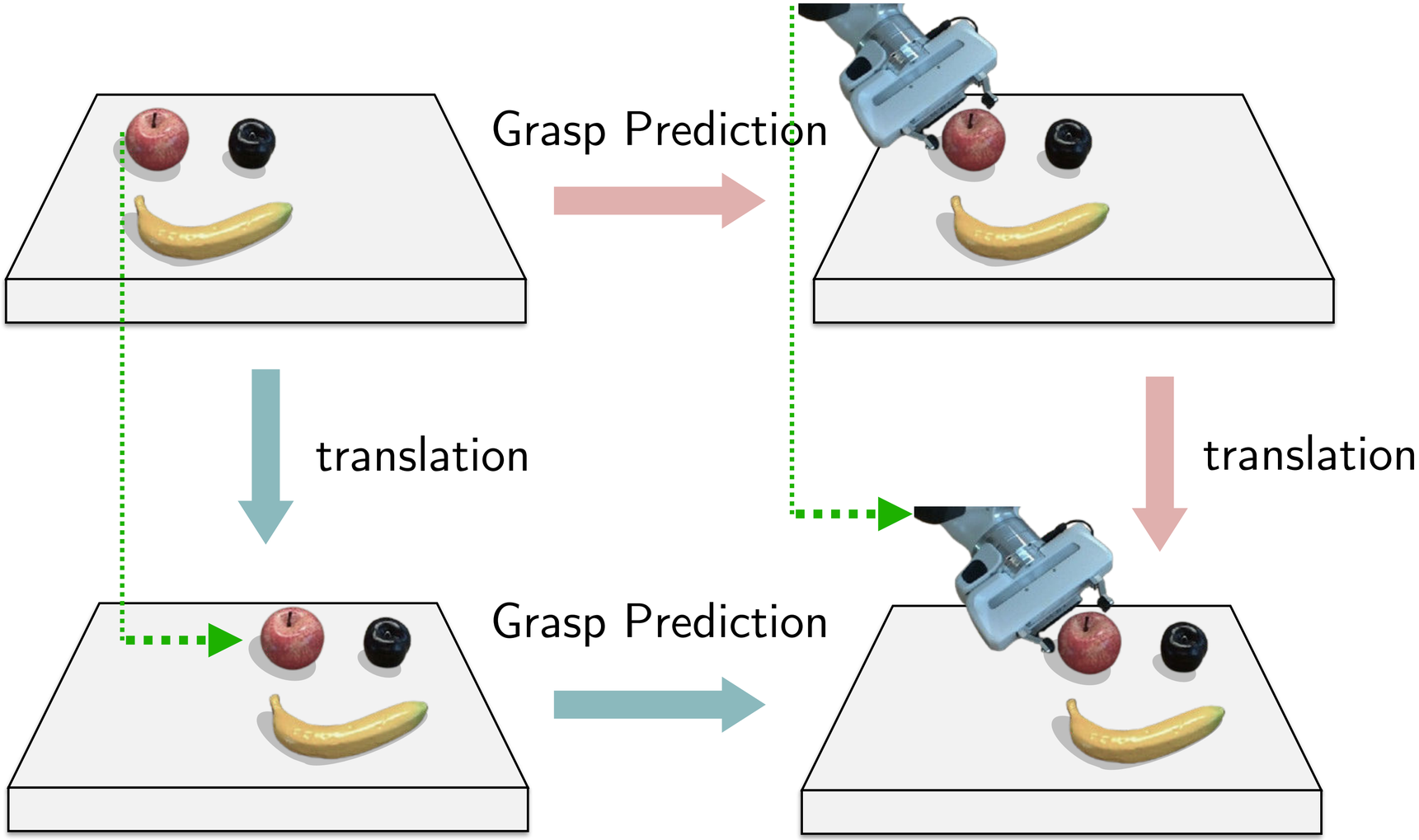


A grasp model trained atop tri-plane features inherits translation equivariance





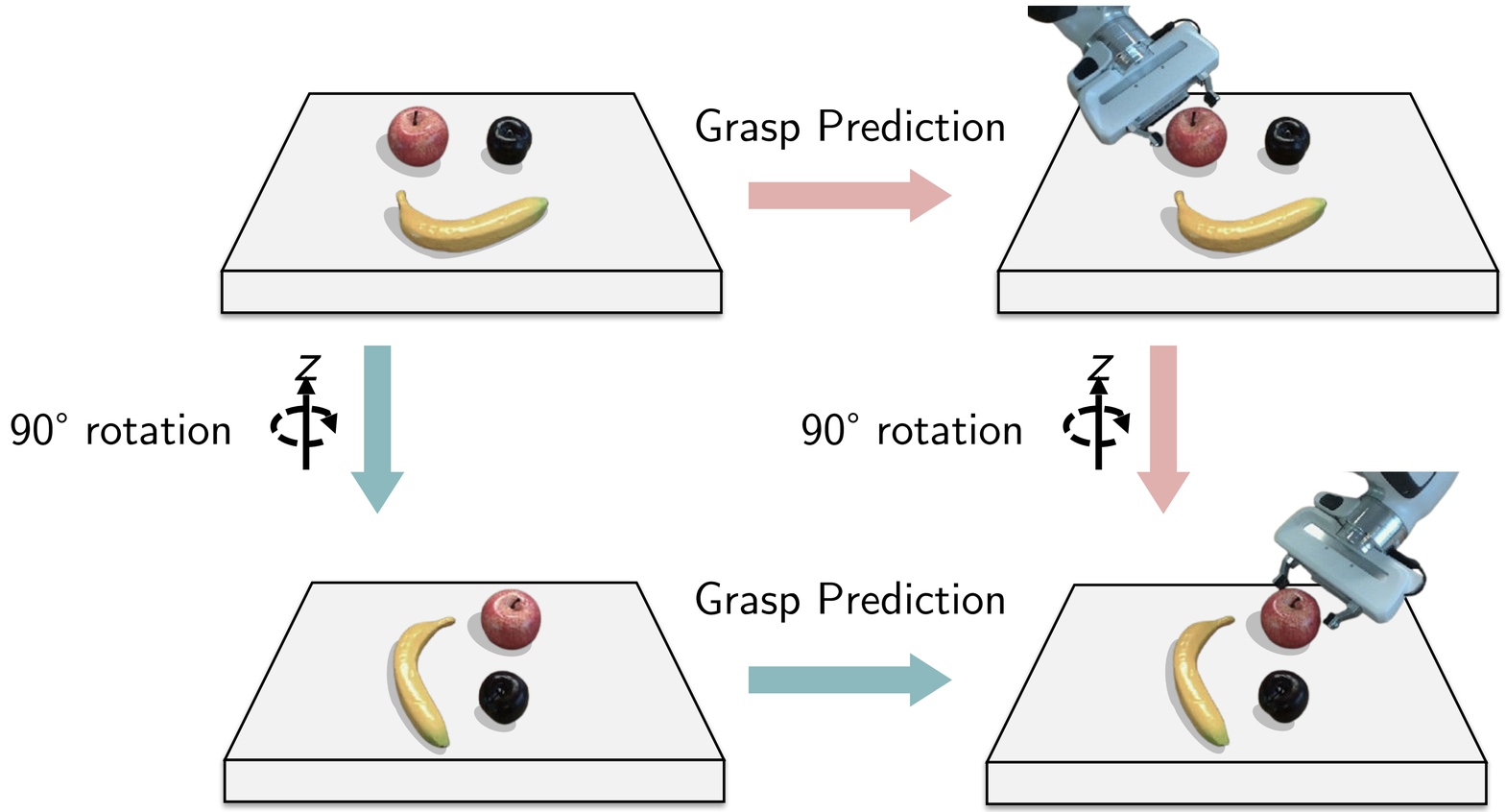




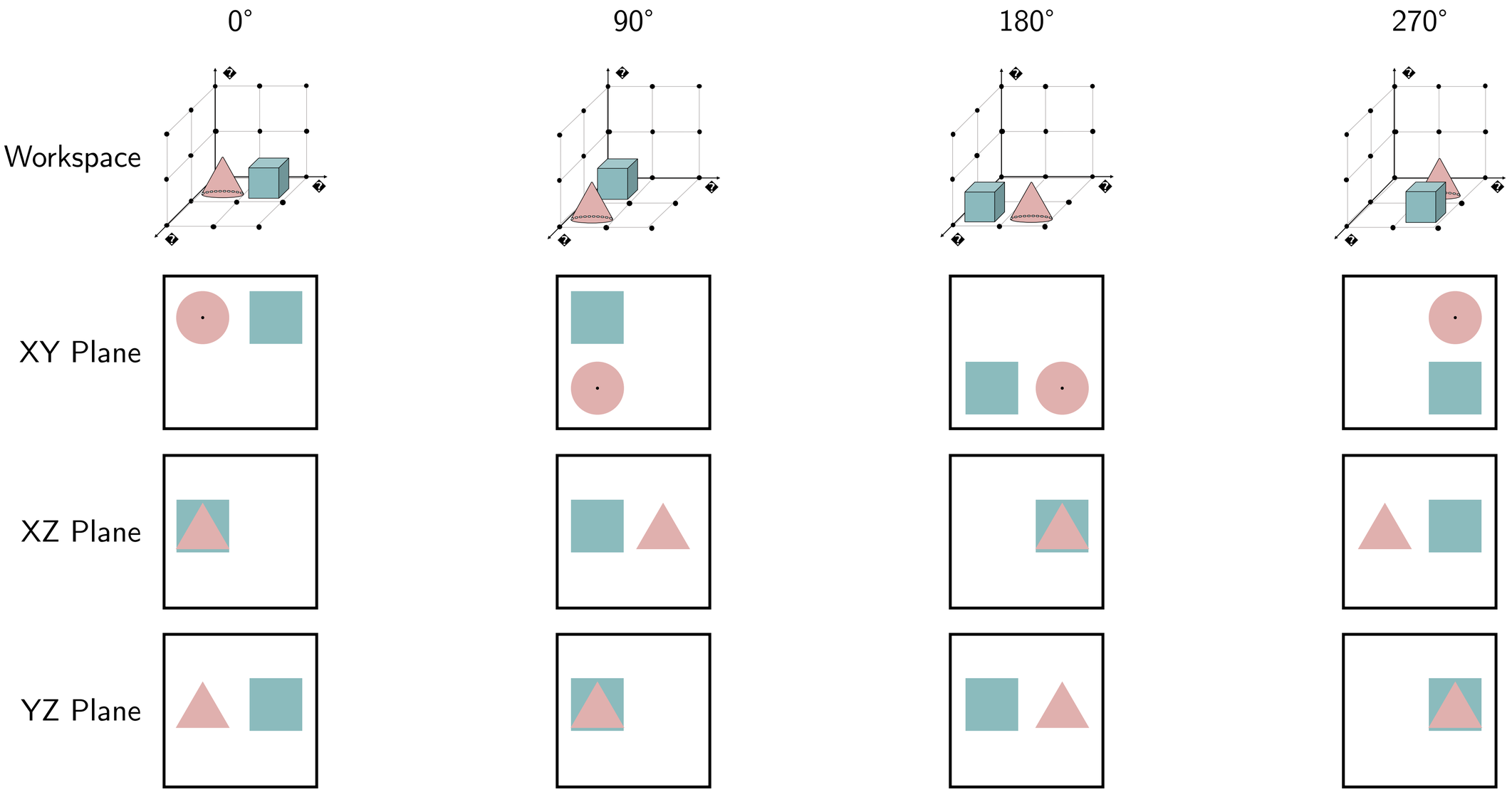
We opt to design for equivariance to 90° rotations around Z

Two relevant observations:

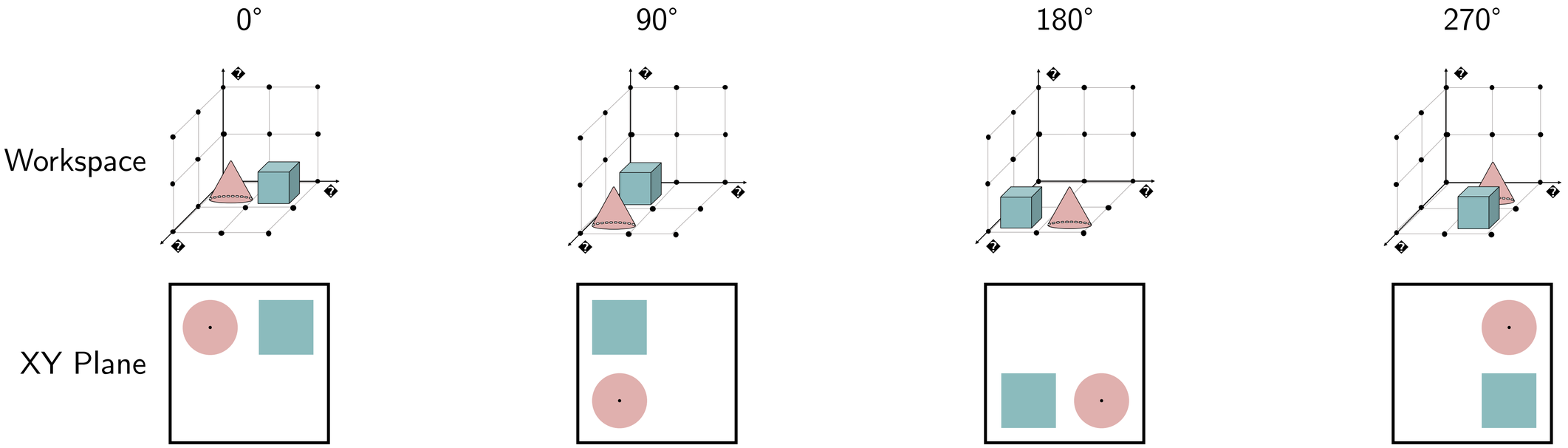
1. The orthogonality of the three feature planes would make it particularly convenient to design equivariance to 90° rotations,
2. Objects set on a table rotate more often around vertical axis.



We must design planar features that are equivariant to 90° rotations of the workspace around Z

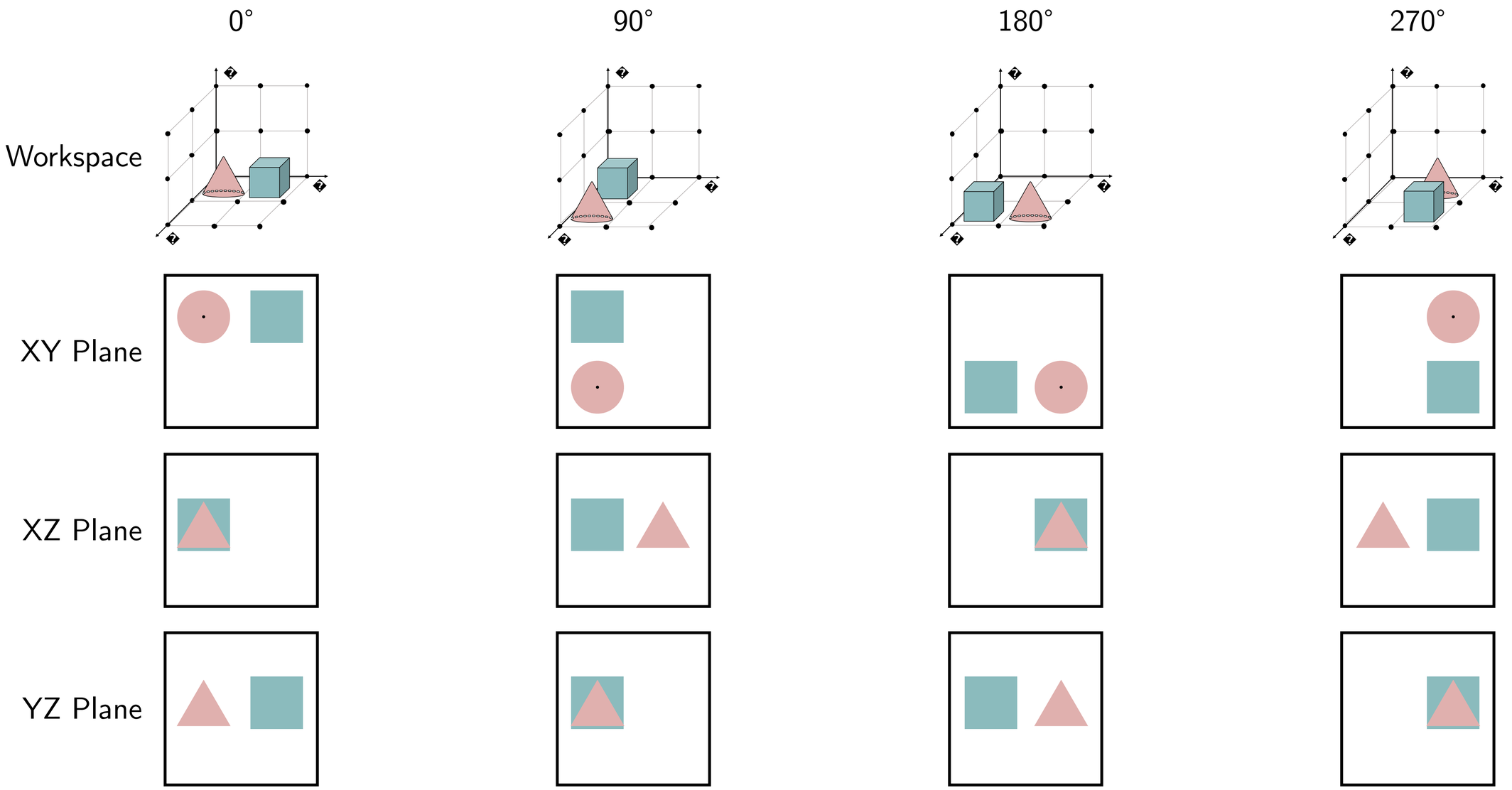


We must design planar features that are equivariant to 90° rotations of the workspace around Z

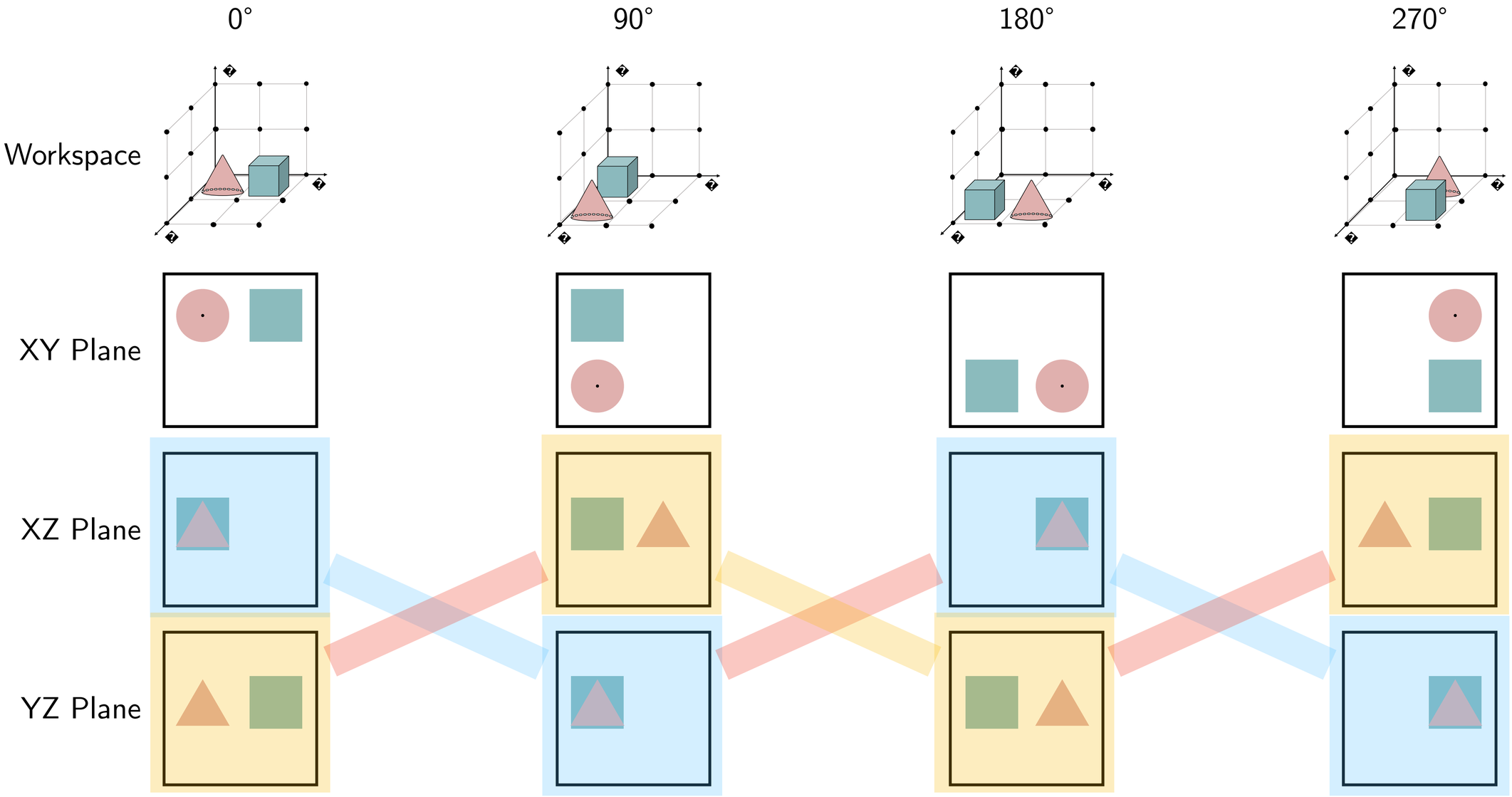


- For the XY plane, equivariance to a 90° rotation around Z is equivalent to equivariance to in-plane rotations.
- It can be achieved by equipping the XY UNet with C_4-equivariant steerable convolutions.

For XZ and XZ, equivariance to 90° rotations around Z is not equivalent to in-plane rotations

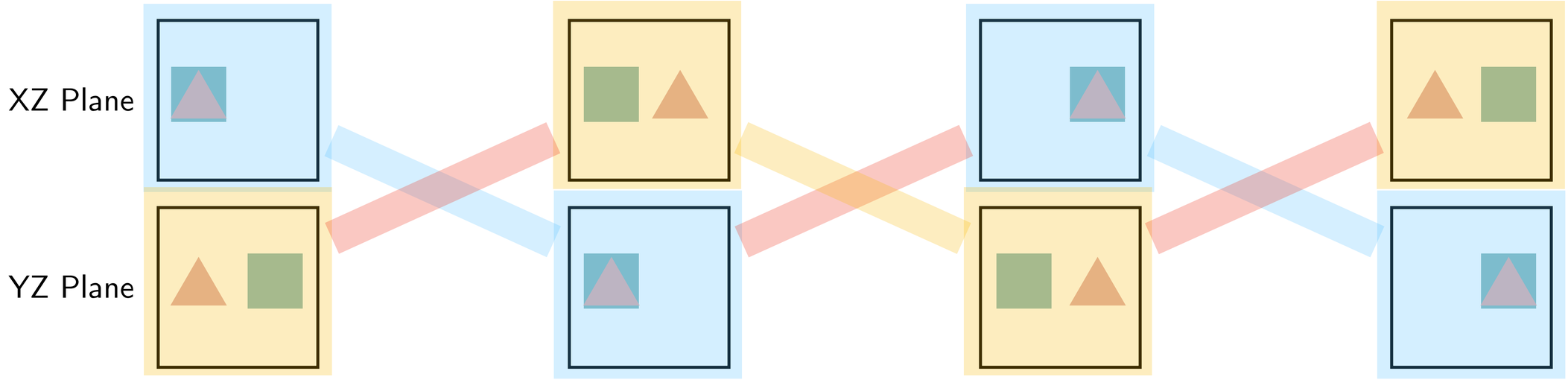


The predictable effect on XZ and YZ of a 90° Z-rotations is a permutation between XZ and YZ, occasionally accompanied by a reflection



The predictable effect on XZ and YZ of a 90° Z-rotations is a permutation between XZ and YZ, occasionally accompanied by a reflection

- **If** we design the XZ and YZ UNets for reflection invariance,
- **Then** the pairwise sum of reflection-invariant XZ and YZ features is invariant to the permutations induced by 90° Z-rotations.
- **Conclusion**: Downstream tasks (a grasp planner) will use the sum of reflection-invariant XZ/YZ features.



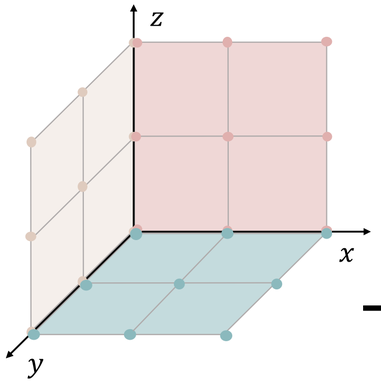
Equivariant to in-plane translation and invariant to reflections and XY-YZ permutations

Volumetric Input

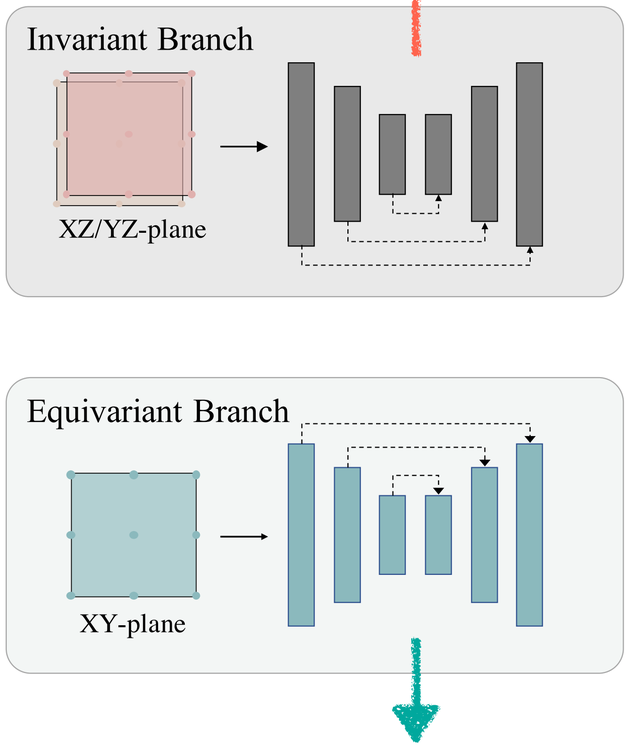
Lifting Conv

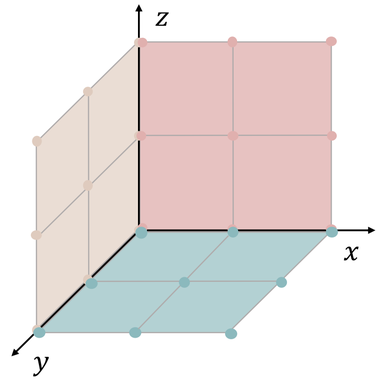
Projection

Aggregation



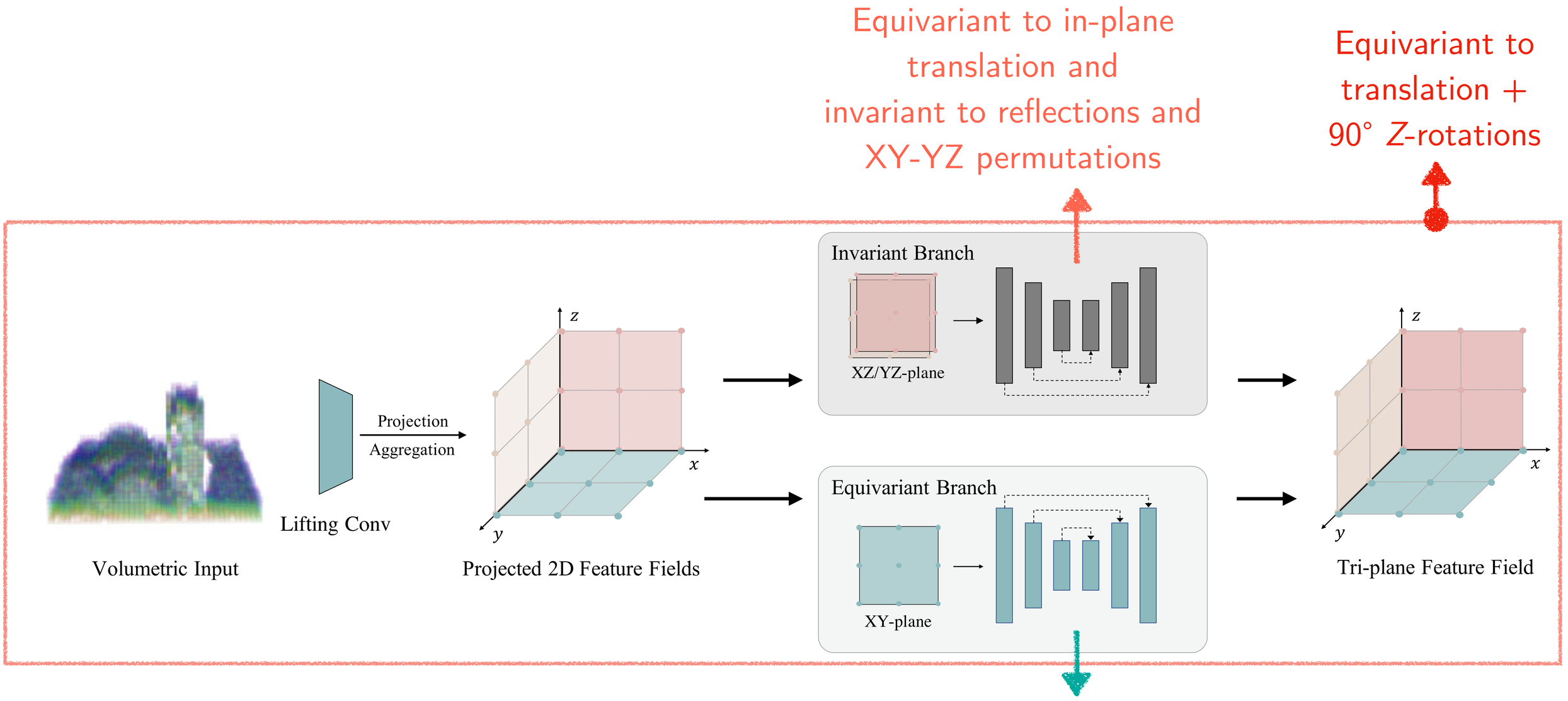
Projected 2D Feature Fields



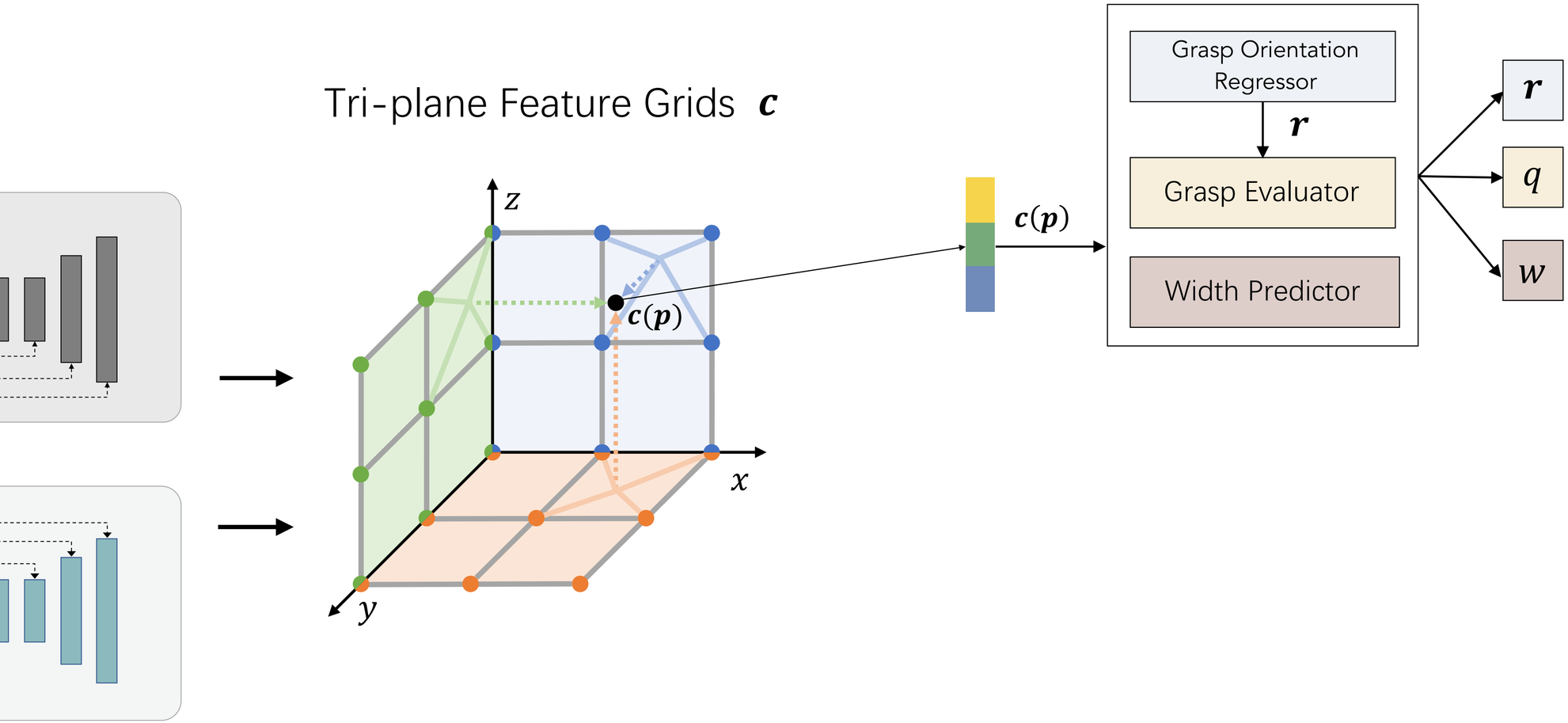


Tri-plane Feature Field

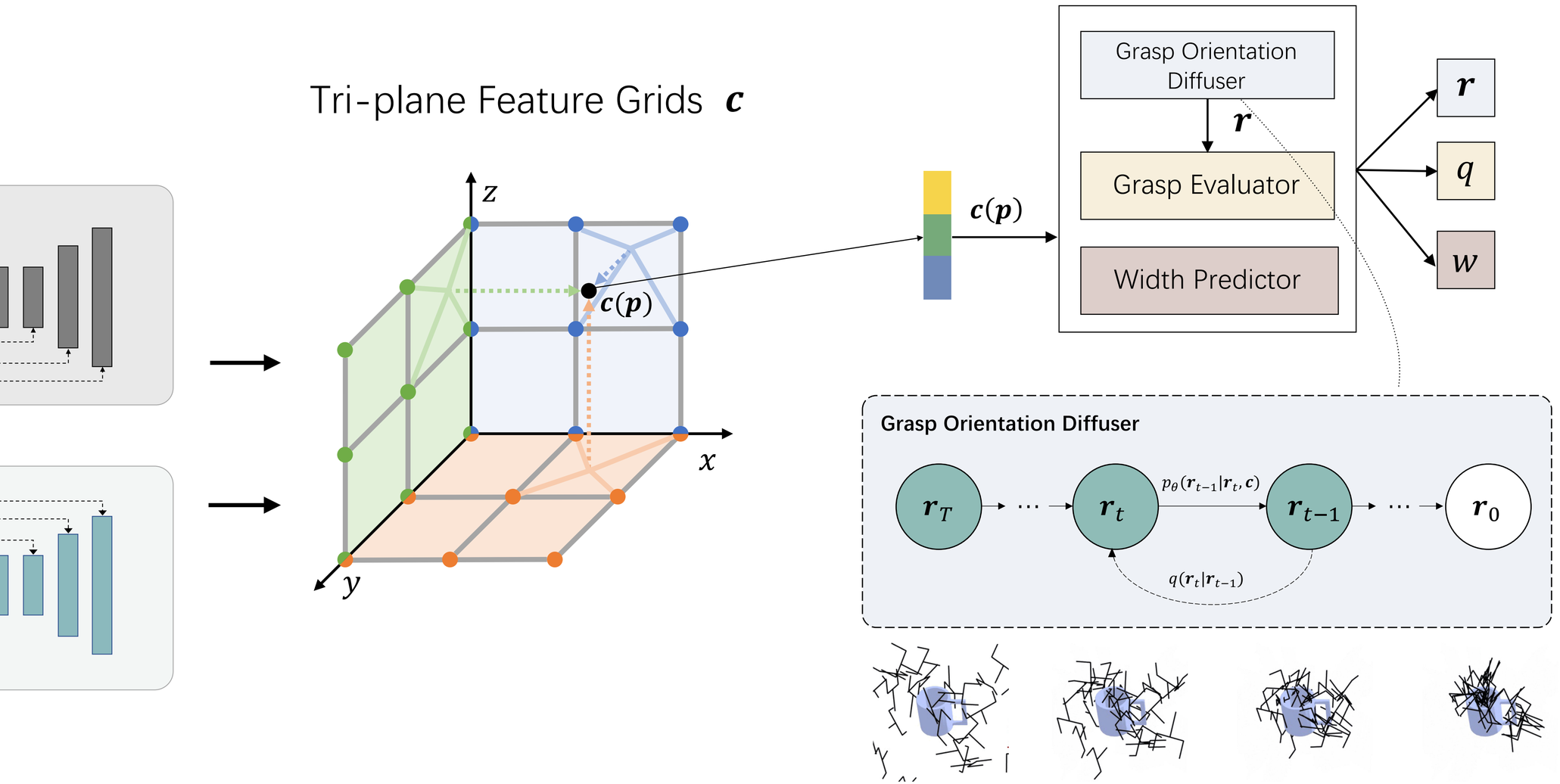
Equivariant to in-plane translation/rotation


Equivariant to in-plane translation and invariant to reflections and XY-YZ permutations
Equivariant to translation + 90° Z-rotations
Invariant Branch
XZ/YZ-plane
Equivariant Branch
XY-plane
Projection
Aggregation
Lifting Conv
Volumetric Input
Projected 2D Feature Fields
Tri-plane Feature Field
Equivariant to in-plane translation/rotation

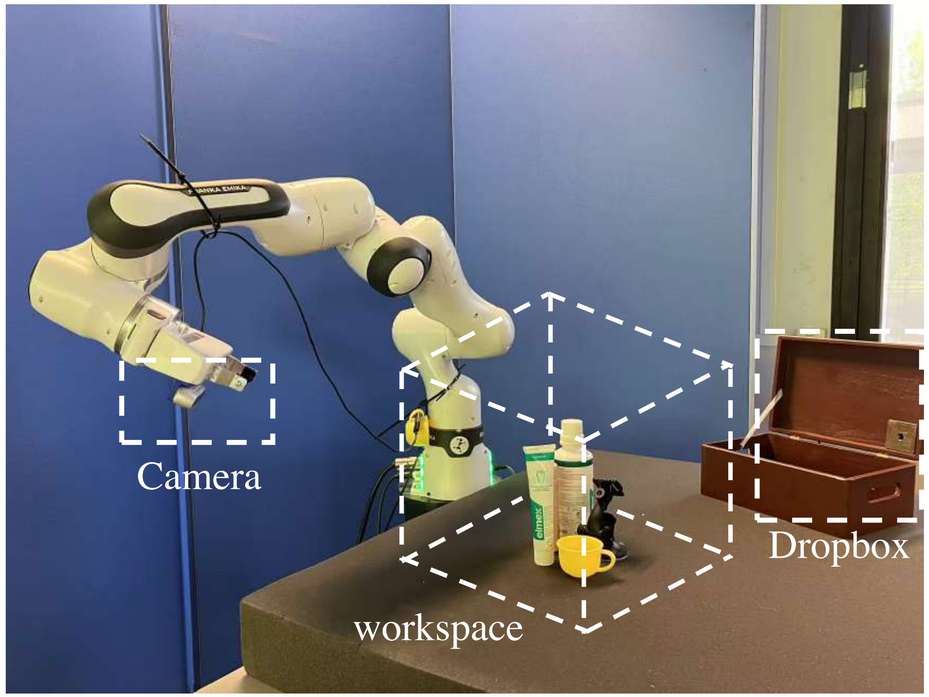
EquiGIGA: Given a grasp location *p*, we ground models of gripper rotation *r*, grasp quality *q* and gripper width *w* in the tri-plane feature *c(p)*



EquiIGD: Grasp rotations are encoded with a diffusion model, which effectively captures multi-modal rotation distributions

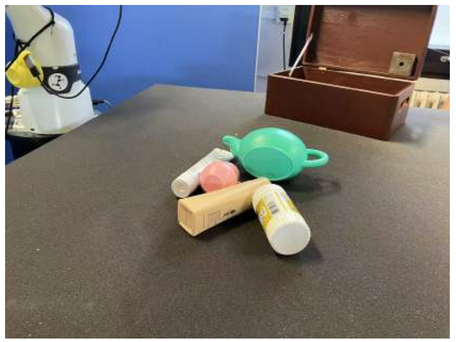


Method	Packed		Pile		Latency (ms)
	GSR (%)	DR (%)	GSR (%)	DR (%)	
VGN* [1]	72.5±2.6	76.7±1.7	59.3±2.9	43.5±2.9	**9**
GIGA* [3]	84.8±2.2	85.1±2.5	69.5±1.3	49.0±3.4	**24**
GraspNet-1B Baselines* [12]	49.9±2.3	40.1±2.2	50.2±4.2	30.0±2.3	77
GSNet* [11]	67.8±2.5	60.1±3.2	58.3±3.8	51.3±4.6	156
GPD* [44]	41.8±2.9	34.1±3.4	22.7±1.1	9.0±0.7	2138
6DoF-GraspNet* [16]	17.9±0.8	11.9±0.9	15.5±2.9	6.9±1.1	2232
SE(3)-Dif* [15]	7.2±1.5	4.3±1.0	7.6±1.8	3.0±0.8	5691
EdgeGraspNet† [13]	54.1±2.1	54.0±2.7	50.5±3.7	43.0±4.8	843/685
VN-EdgeGraspNet† [13]	60.6±2.2	60.1±3.8	55.0±2.1	50.1±4.0	1174/953
ICGNet† [20]	60.3±4.1	64.5±5.9	57.3±1.5	51.7±3.3	806
DexGraspNet2† [21]	51.6±2.5	53.9±4.3	39.7±1.3	30.9±2.2	2781
OrbitGrasp† [6]	71.1±1.8	72.8±1.6	69.3±2.1	64.7±3.3	3193
IGD* (N=1) [2]	92.9±1.8	86.7±1.8	68.2±1.9	50.6±1.5	217
IGD* (N=11) [2]	91.2±0.9	88.8±1.5	71.8±2.2	55.7±2.6	1823
EquiGIGA	**96.8±1.0**	88.6±1.3	76.6±2.5	**76.4±2.9**	65
EquiGIGA (HR)	93.1±1.2	**91.8±1.3**	**78.6±1.0**	75.5±1.3	200
EquiIGD	**97.4±1.6**	91.4±1.4	**78.6±2.1**	**78.0±3.0**	147
EquiIGD (HR)	96.0±0.8	**92.4±1.4**	74.9±1.2	73.0±0.8	240

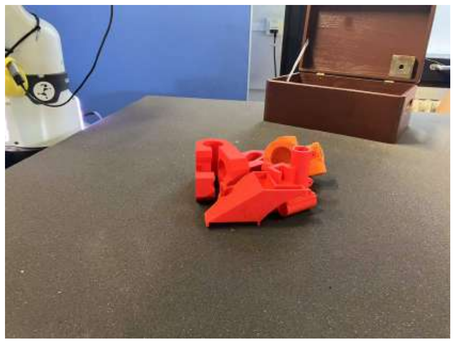


(a) Experimental setup

(b) Packed scene



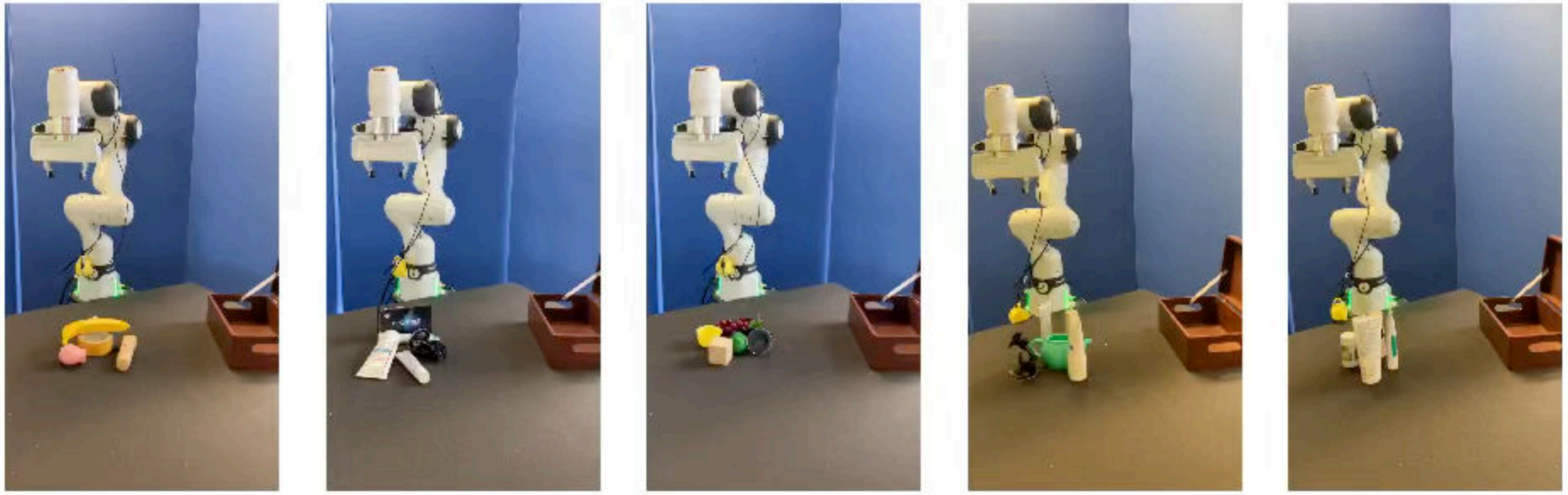
(c) Pile scene



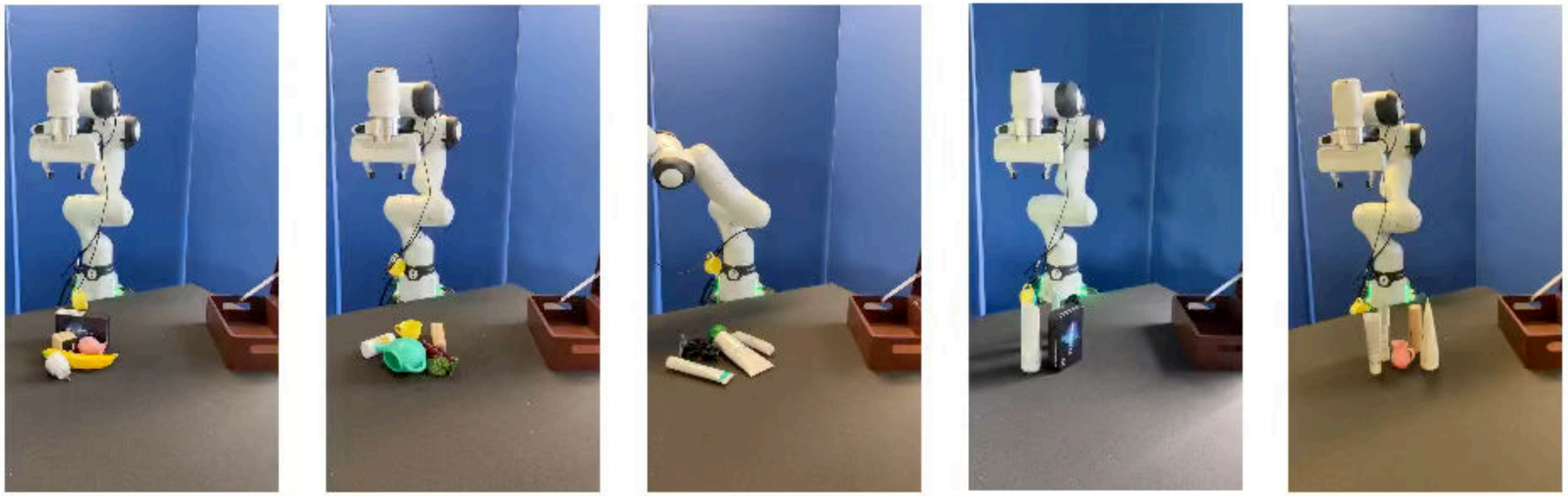
(d) Adv scene

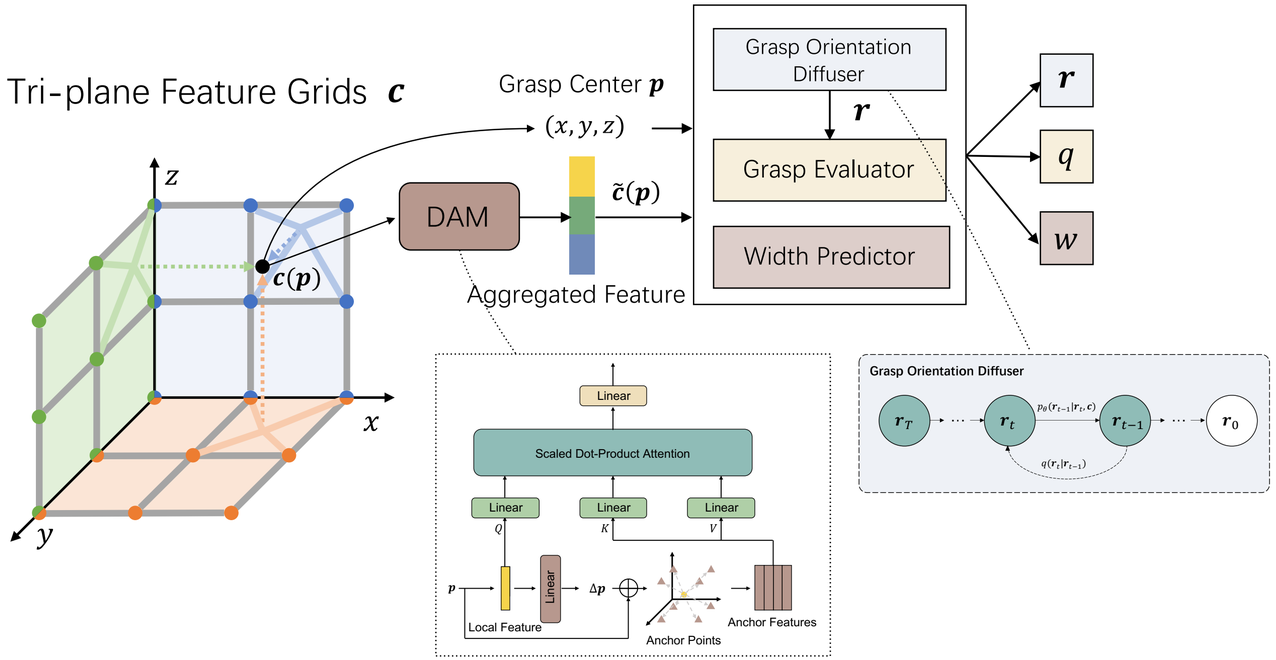
Method	Packed		Pile		Adv	
	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)
GIGA [3]	76.7 (66/86)	88.0	61.1 (44/72)	58.7	72.5 (66/99)	88.0
EdgeGraspNet [13]	73.4 (58/79)	77.3	62.1 (41/66)	54.7	72.2 (57/79)	76.0
VN-EdgeGraspNet [13]	71.3 (57/80)	76.0	67.7 (44/65)	58.7	79.5 (58/73)	77.3
IGD [2]	78.0 (64/82)	85.3	63.0 (51/88)	68.0	78.2 (61/78)	81.3
ICGNet [20]	72.2 (57/79)	76.0	71.1 (54/76)	72.0	69.9 (51/73)	68.0
EquiGIGA	82.7 (67/81)	89.3	79.3 (65/82)	86.7	85.6 (71/83)	94.7
EquiIGD	89.9 (71/79)	94.7	77.0 (67/87)	89.3	88.1 (74/84)	98.7

EquiGIGA



EquiIGD

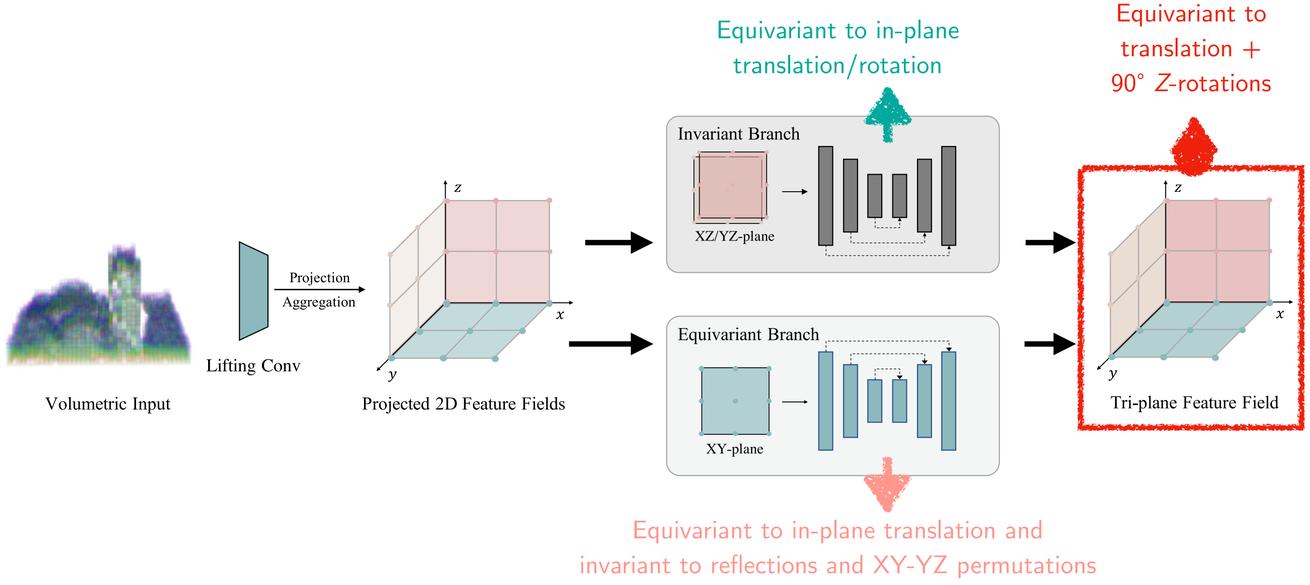




Implicit grasp diffusion: Bridging the gap between dense prediction and sampling-based grasping.

P. Song, P. Li, and R. Detry.

CoRL 2024



Equivariant volumetric grasping.

P. Song, Y. Hu, P. Li, and R. Detry.

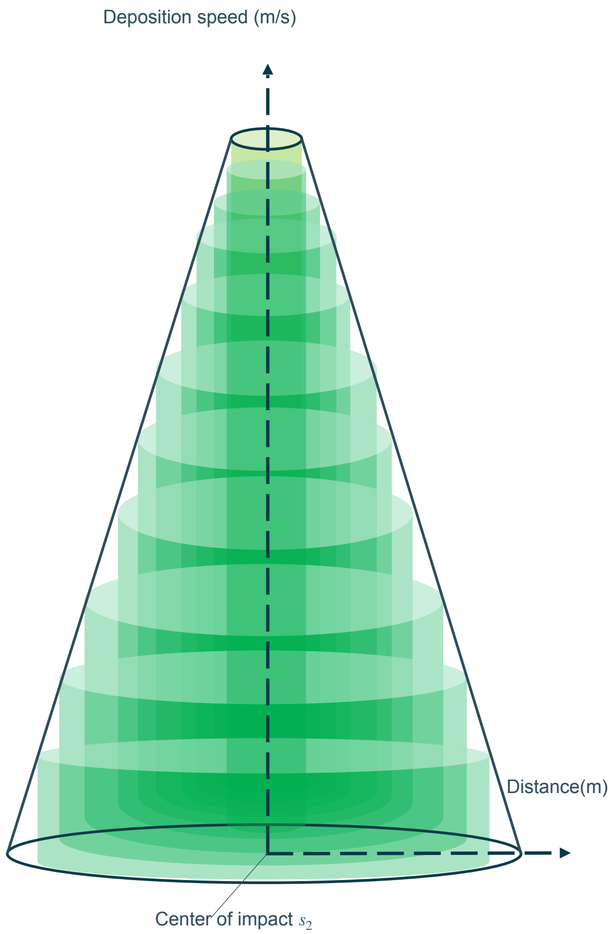
AREPO: Uncertainty-Aware Robot Ensemble Learning Under Extreme Partial Observability

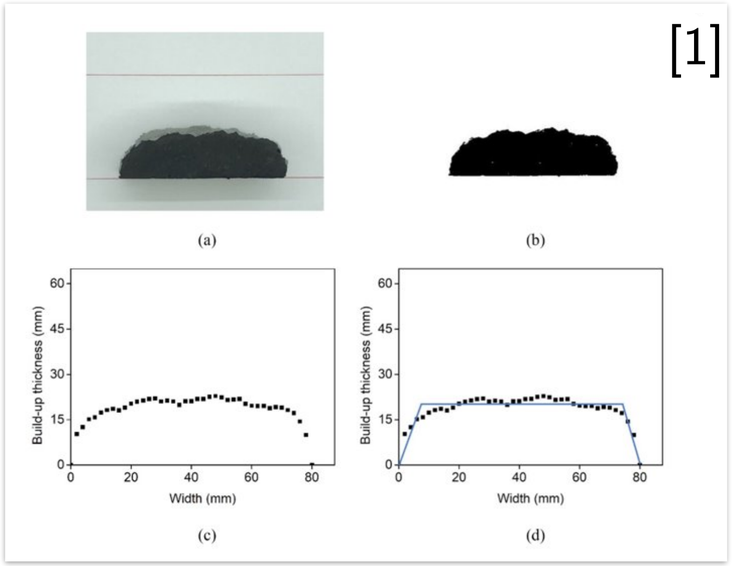
Yurui Du, Louis Hanut,
Herman Bruyninckx,
Renaud Detry

KU LEUVEN

ROMANDIC – Feb 9, 2025

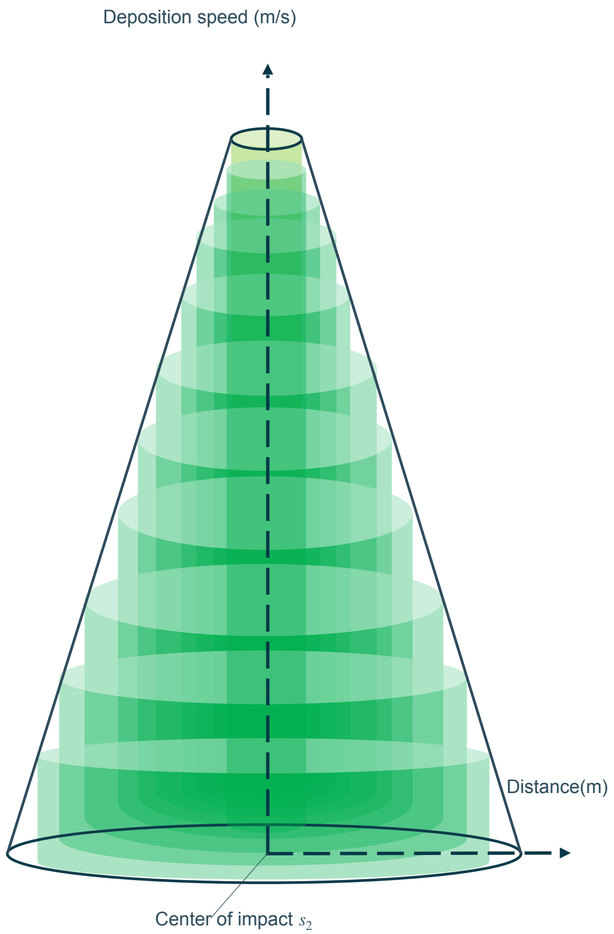
To adequately model shotcreting with MPC, we fit a parametric forward model to tens of minutes of field footage

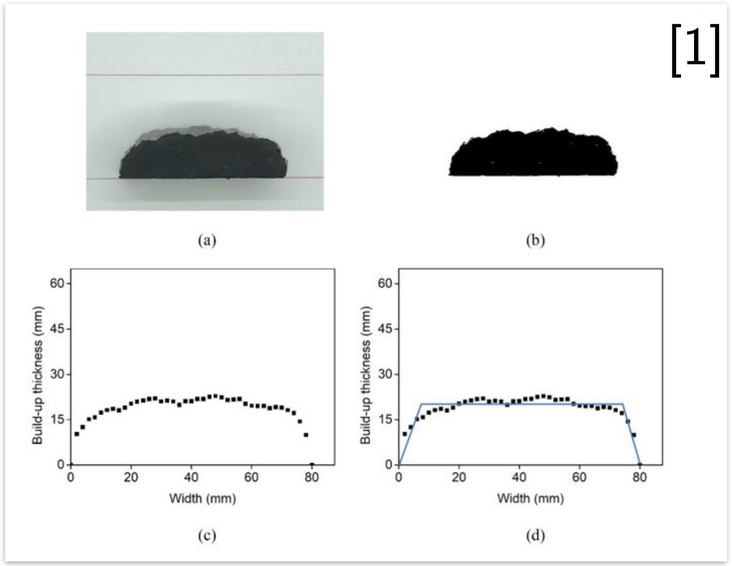


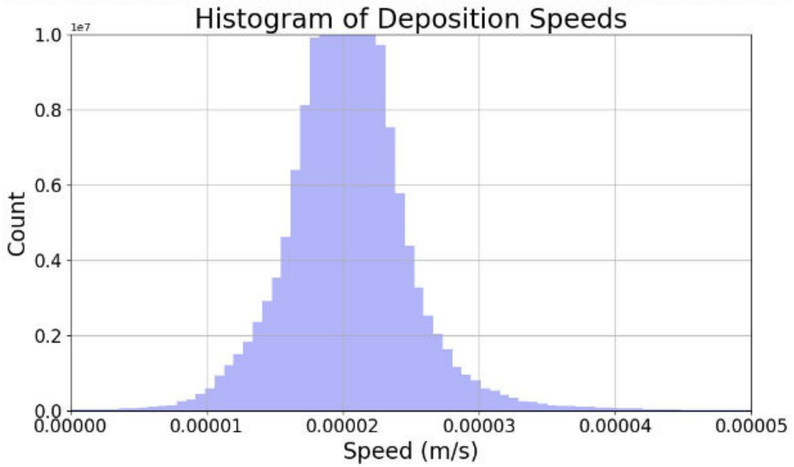


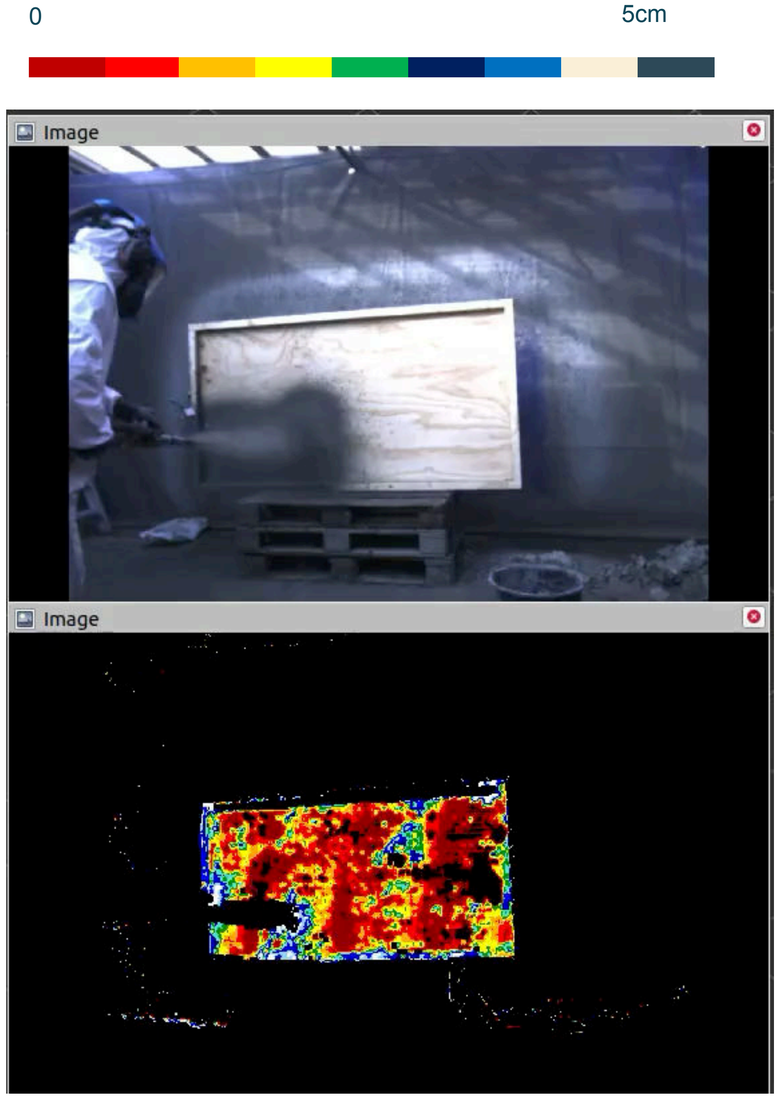
[1] B. Lu, M. Li, T. N. Wong and S. Qian, "Effect of printing parameters on material distribution in spray-based 3D concrete printing (S-3DCP)," 2021.

To adequately model shotcreting with MPC, we fit a parametric forward model to tens of minutes of field footage







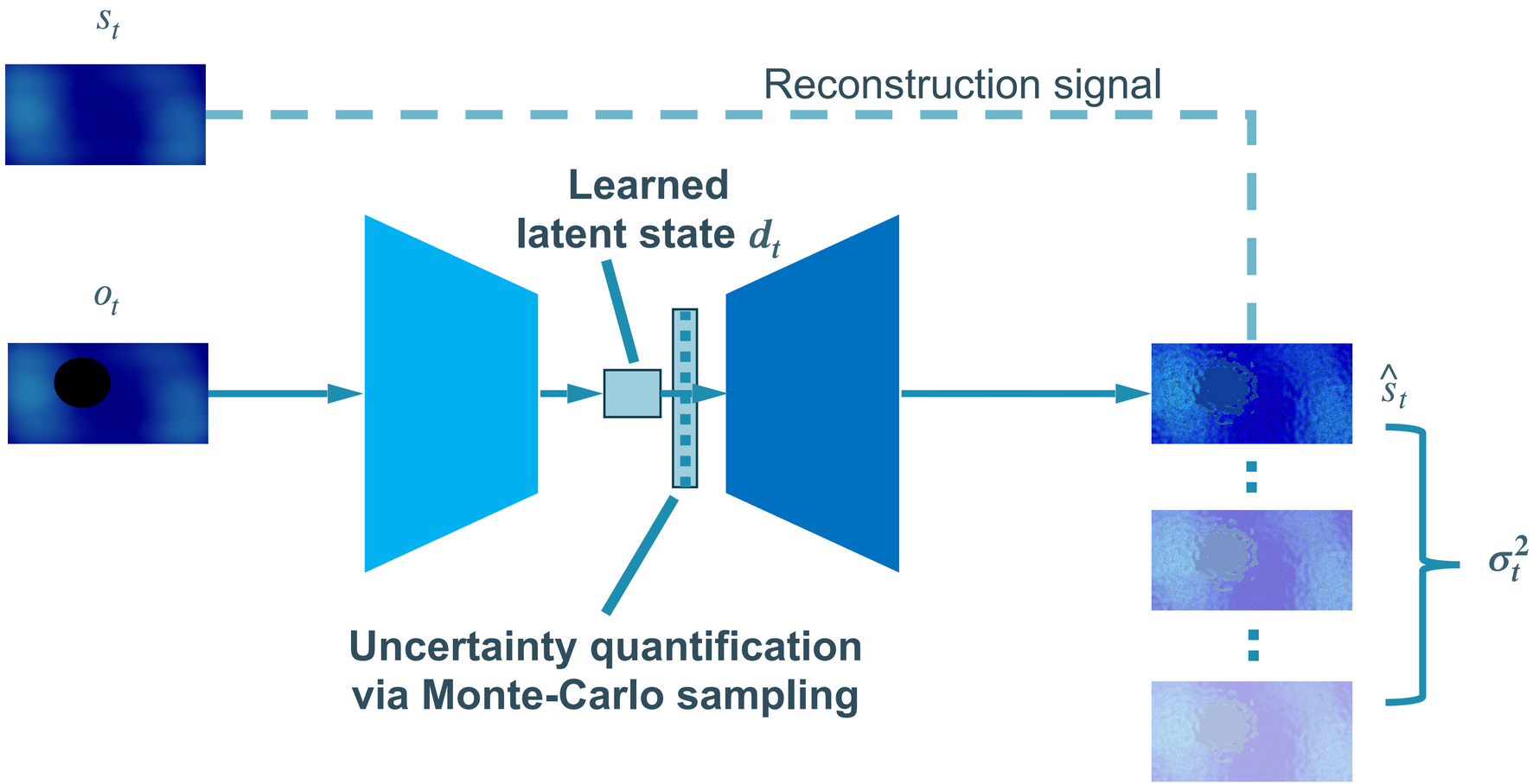


[1] B. Lu, M. Li, T. N. Wong and S. Qian, "Effect of printing parameters on material distribution in spray-based 3D concrete printing (S-3DCP)," 2021.

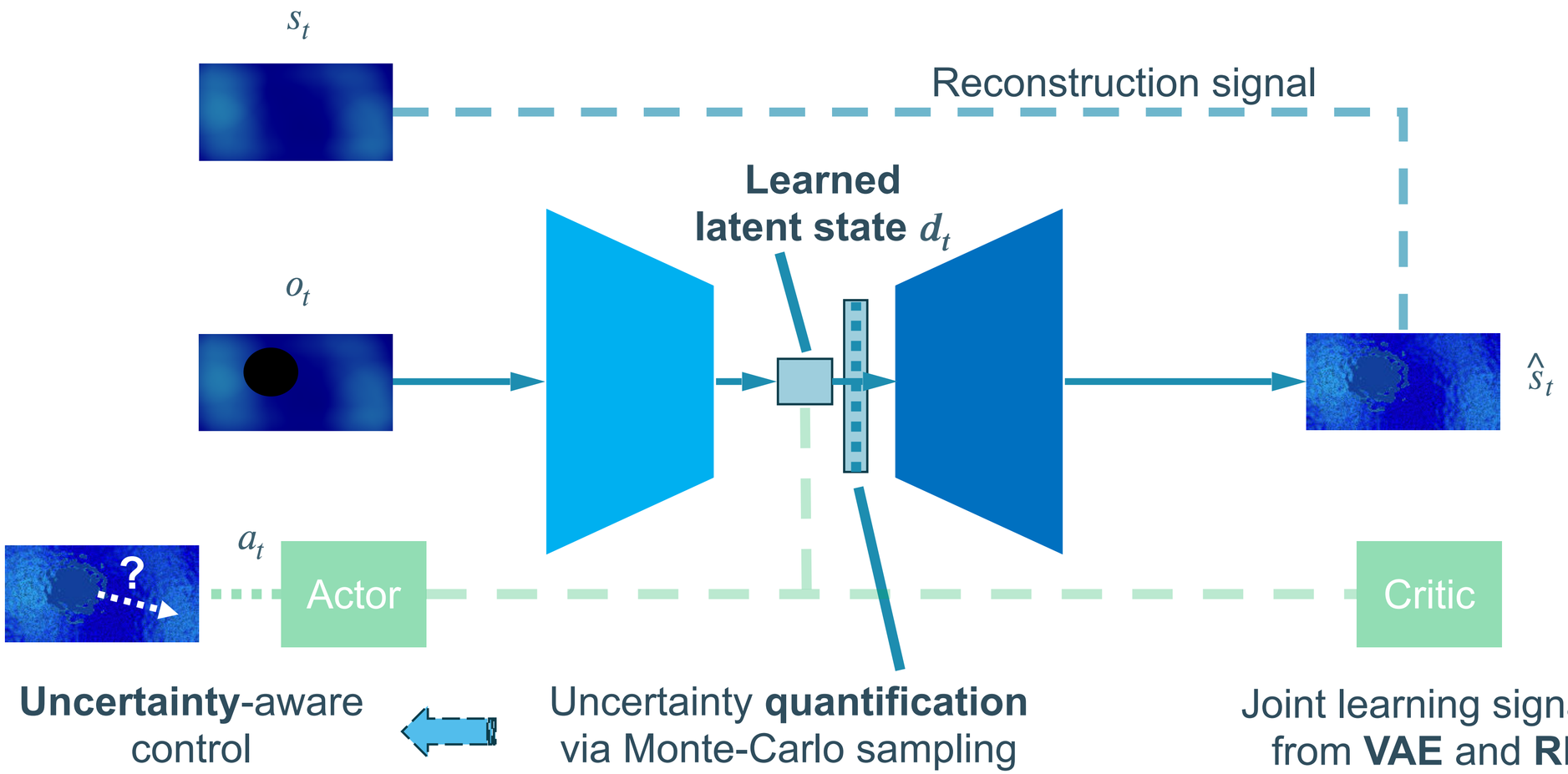
Our MPC planner makes reasonable decisions and achieves real-time control, despite noisy observations and high occlusion

1. **State s_t:**
   - $\text{height\_map}_{h \times w}, \text{nozzle\_location}_{x,y}$
2. **Action a_t:**
   - $\text{nozzle\_velocity}_{v_x, v_y}$
3. **Dynamics:**
   - $F(s_{t+1}) = F(s_t, a_t)$
   - Dynamics is parameterized by a fitted deposition model.
4. **Cost:**
   - $\text{Cost} = \text{MSE}(\text{target}_{h \times w} - \text{height\_map}_{h \times w})$
5. **Constraints:**
   - $0 < x < \text{height}, 0 < y < \text{width}$
   - $-0.5 < v_x < 0.5, -0.5 < v_y < 0.5$

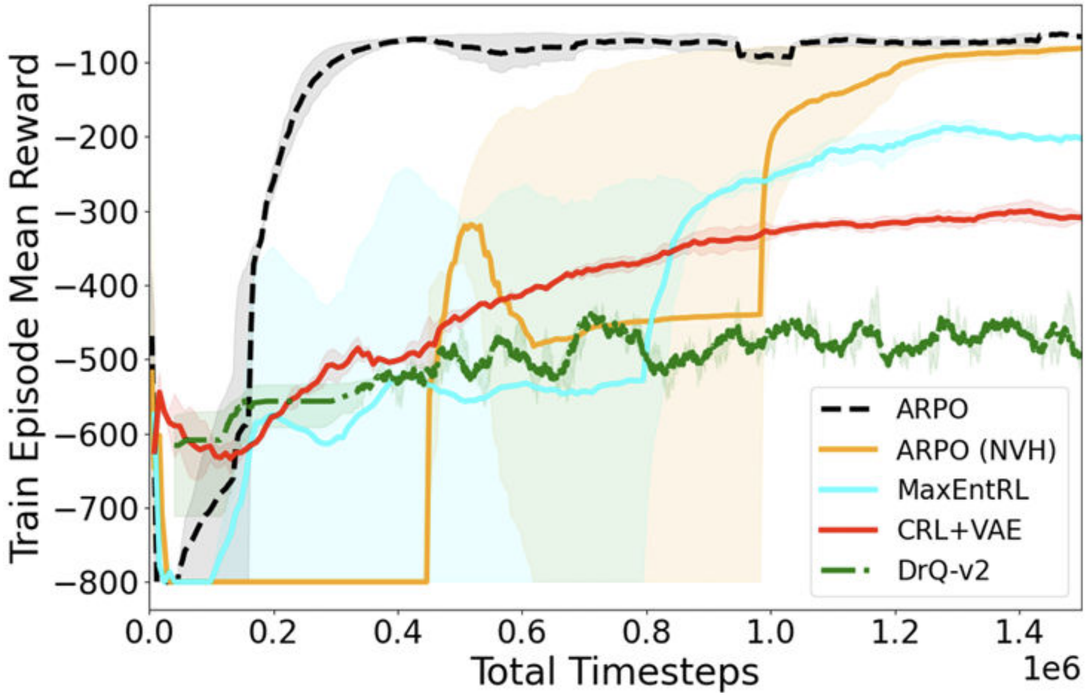
To handle the noise that affects visual data, we model a denoised state with a VAE



The empirical uncertainty that affects the state shapes the agent's reward function



Our RL solution demonstrated results superior than MPC or other RL baselines

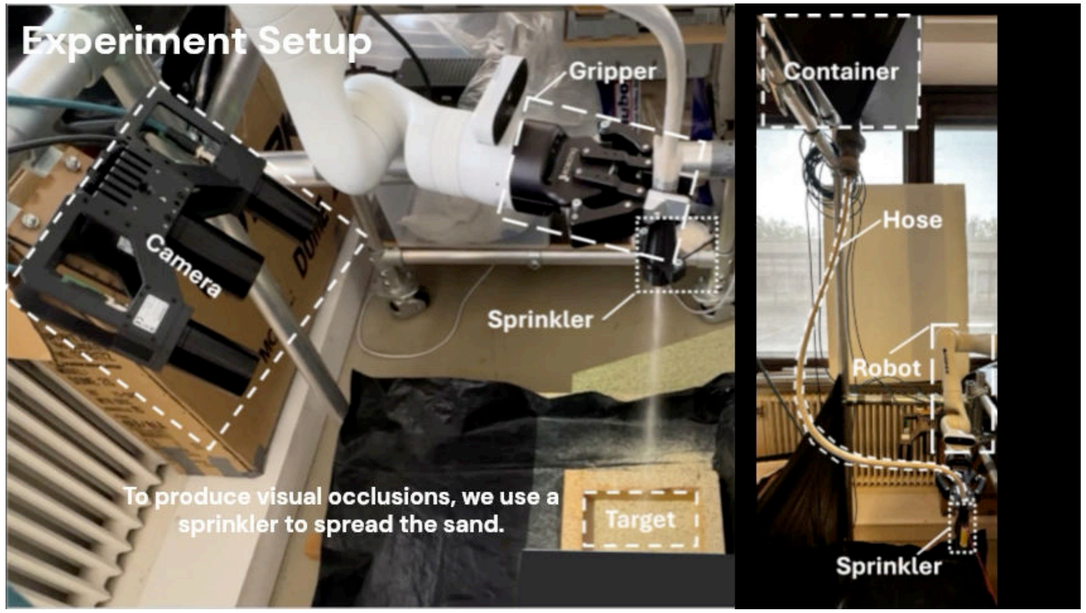


The performance of the agents are compared with 4 metrics: i) *root-mean-square roughness* R_{rms}: the square root of the mean of the squares of the deviations of the surface height values from the mean surface height, ii) *peak-to-valley roughness* R_t: the difference in height between the highest point and the lowest point on a surface, iii) *waste volume ratio* r_{wv}: the ratio between the wasted volume and the desired volume to be fulfilled. The wasted volume is defined as the material volume that has been sprayed outside the target surface or that exceeds the target thickness. iv) *average inference time* t_{avg}: the average time the agent takes to compute a_t given $o_{t-N:t}$.

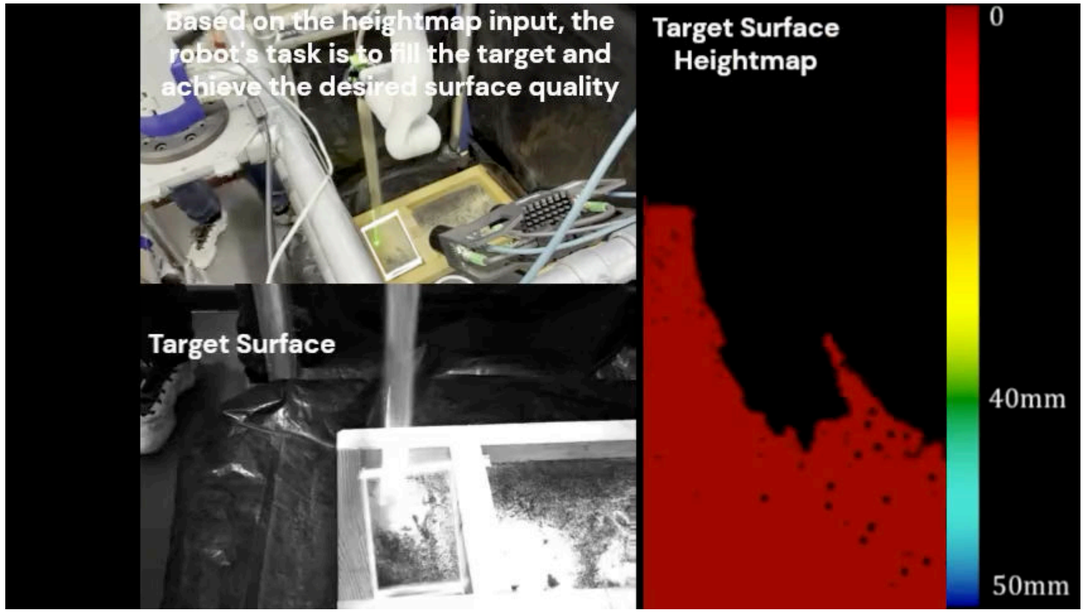
Metrics	R_{rms} (mm)	R_t (mm)	r_{wv} (%)	t_{avg} (ms)
ARPO	$\mathbf{0.6 \pm 0.2}$	$\mathbf{3.9 \pm 1.4}$	$\mathbf{23.8 \pm 0.01}$	17.2 ± 1.4
CRL+VAE	1.7 ± 0.9	8.3 ± 4.9	32.5 ± 0.02	16.6 ± 2.3
MaxEntRL	1.3 ± 0.6	5.5 ± 2.5	33.8 ± 0.01	$\mathbf{7.7 \pm 0.5}$
MPC	2.8 ± 0.4	13.5 ± 1.6	31.3 ± 0.05	59.2 ± 2.5
Vanilla	10.2 ± 0.4	40.6 ± 1.5	53.8 ± 0.02	50.1 ± 1.2

Compared to SoA RL, AREPO improves stability and sampling efficiency

Our proxy sand-sprinkling task confirms simulation results



The agent needs to fill the target volume with sand while achieving smooth surface quality.

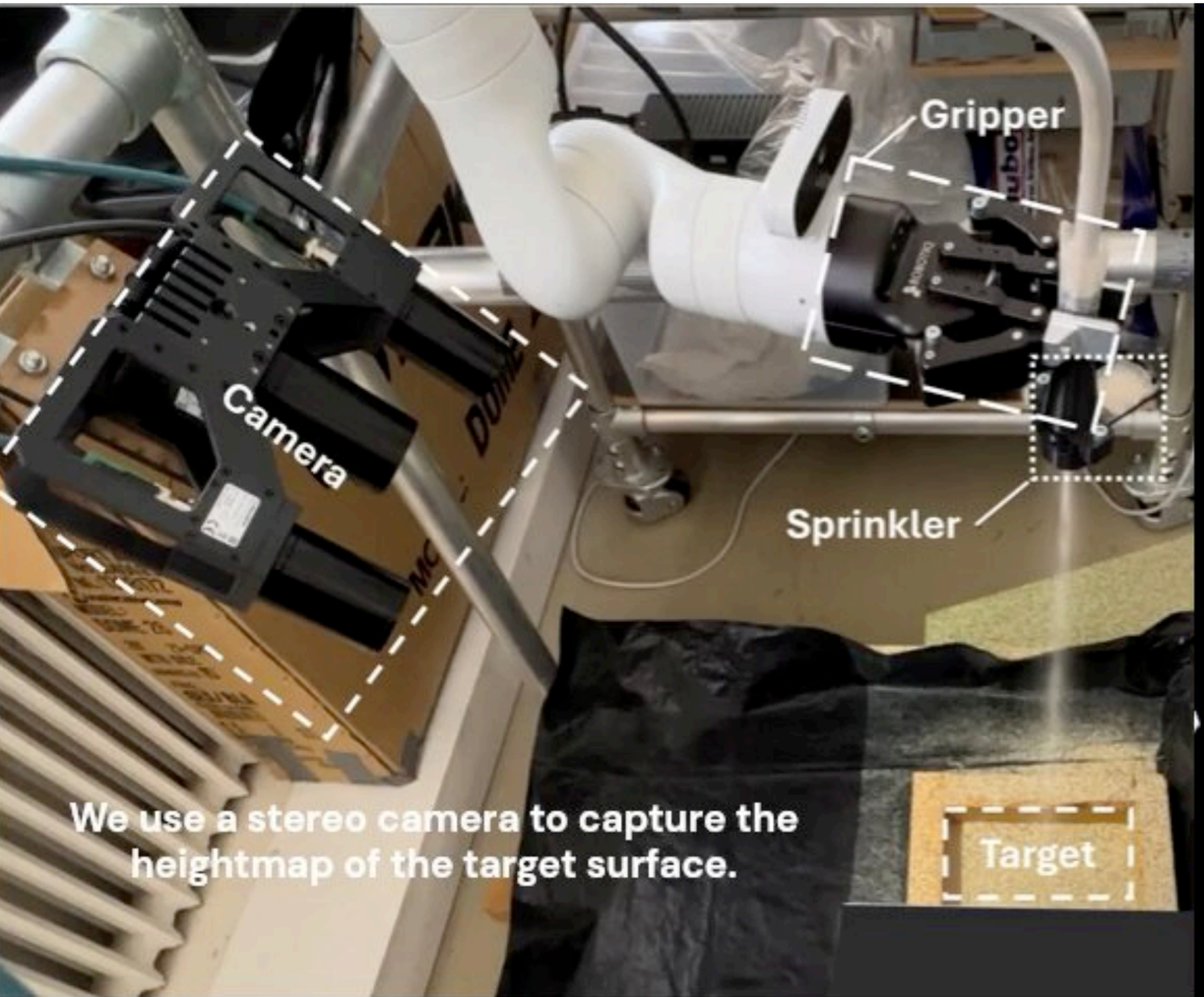


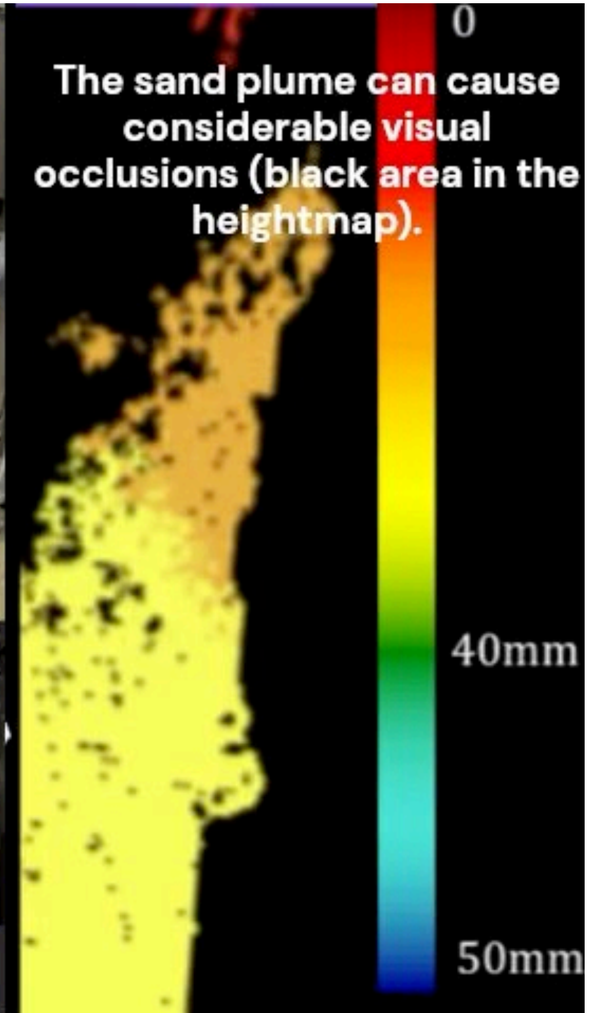
An example of poor performance due to visual occlusions and sim-to-real gap

AREPO: Uncertainty-Aware Robot Ensemble Learning Under Extreme Partial Observability

Yurui Du, Louis Hanut, Herman Bruyninckx, Renaud Detry

Robotics and Automation Letters, 2025



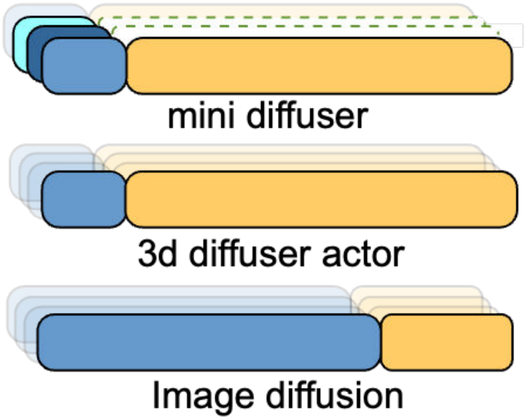


Take-home's:

- Mini-diffuser cuts compute and memory by an order of magnitude. Use it to accelerate the prototyping of your deformable diffusion models!
- Equivariant modeling requires delicate trade-offs.
  - Consider non-trivial equivariance formulations!
  - ... including trip-plane feature projection, which lends itself to C_4 Z-rotation equivariance.

Mini Diffuser: Fast Multi-task Diffusion Policy Training Using Two-level Mini-batches

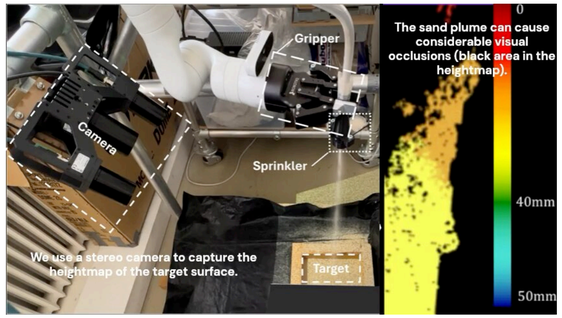
Yutong Hu, Pinhao Song, Kehan Wen, and Renaud Detry



AREPO: Uncertainty-Aware Robot Ensemble Learning Under Extreme Partial Observability

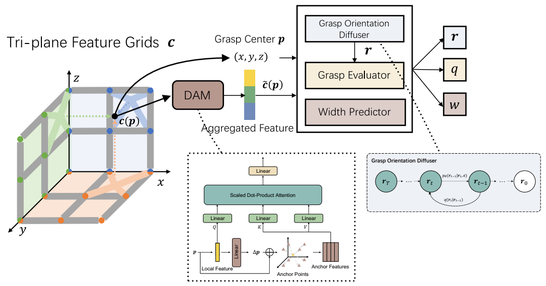
Yurui Du, Louis Hanut, Herman Bruyninckx, Renaud Detry

Robotics and Automation Letters, 2025



Implicit grasp diffusion: Bridging the gap between dense prediction and sampling-based grasping

P. Song, P. Li, and R. Detry, CoRL 2024



Equivariant volumetric grasping

P. Song, Y. Hu, P. Li, and R. Detry