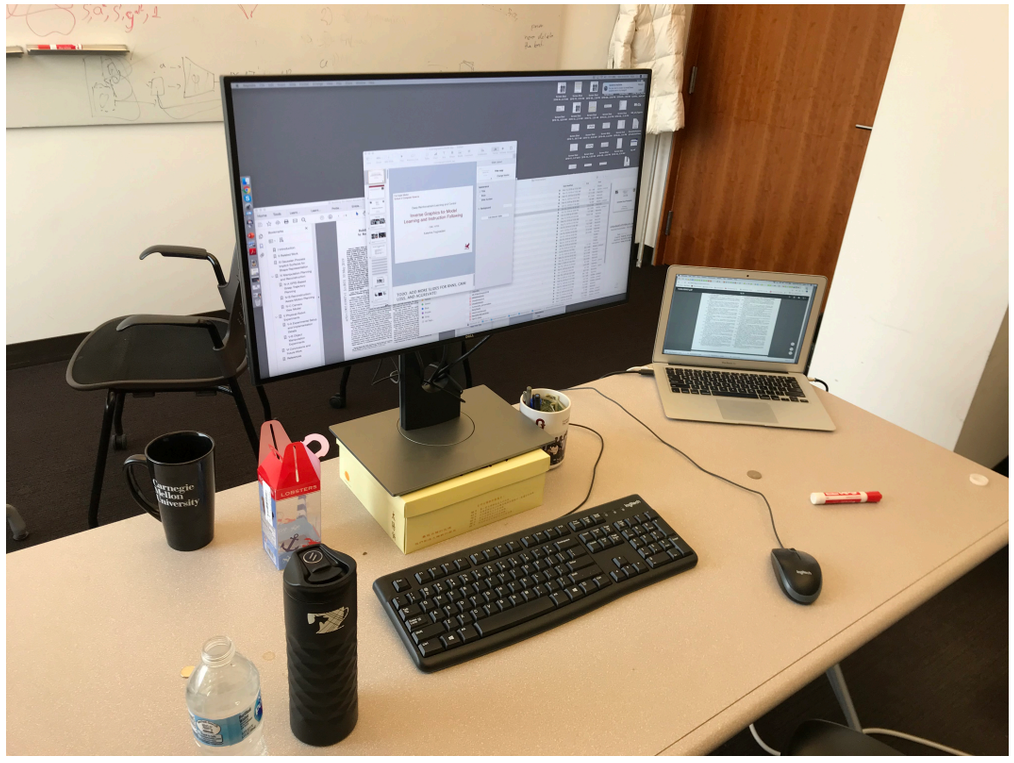
Carnegie Mellon
School of Computer Science

Deep Reinforcement Learning and Control

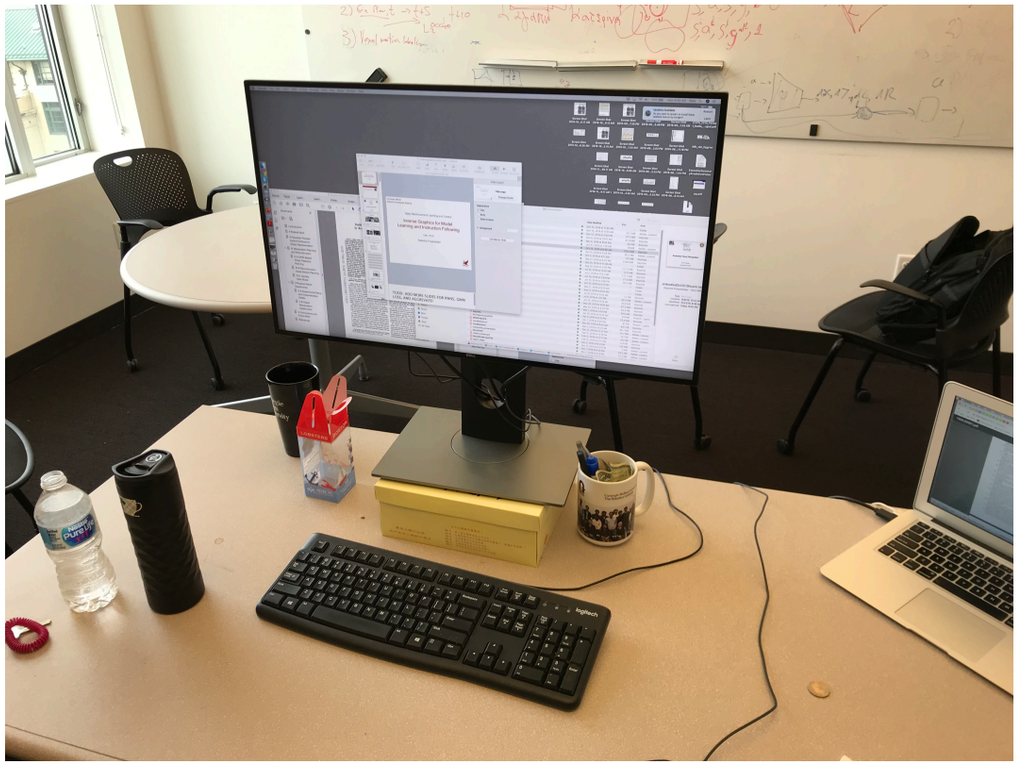
Inverse Graphics for Behaviour learning

CMU 10703

Katerina Fragkiadaki

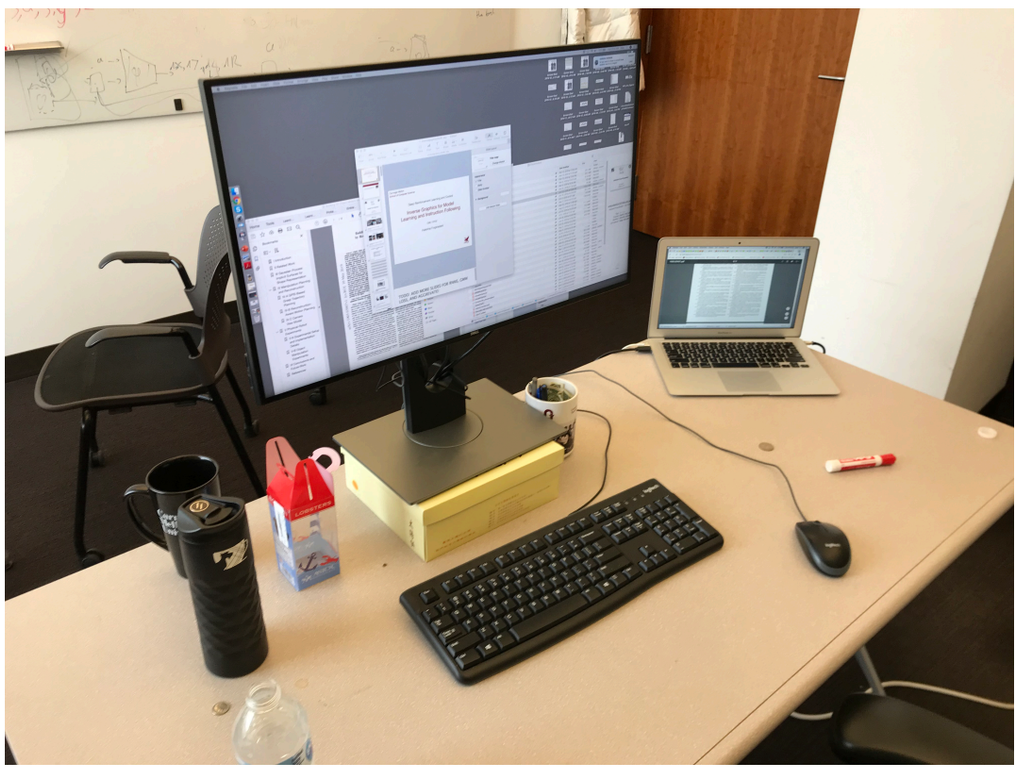


2 pictures depict identical scenes, and one picture depicts a different scene.



``It is easy to infer the dynamics of a scene, to build an approximate mental simulator for it”

Chris Atkeson



``It is easy to infer the dynamics of a scene, to build an approximate mental simulator for it”

Chris Atkeson

(I’m showing the depthmaps but please imagine full 3D meshed object models)

This means you can do (without learning):

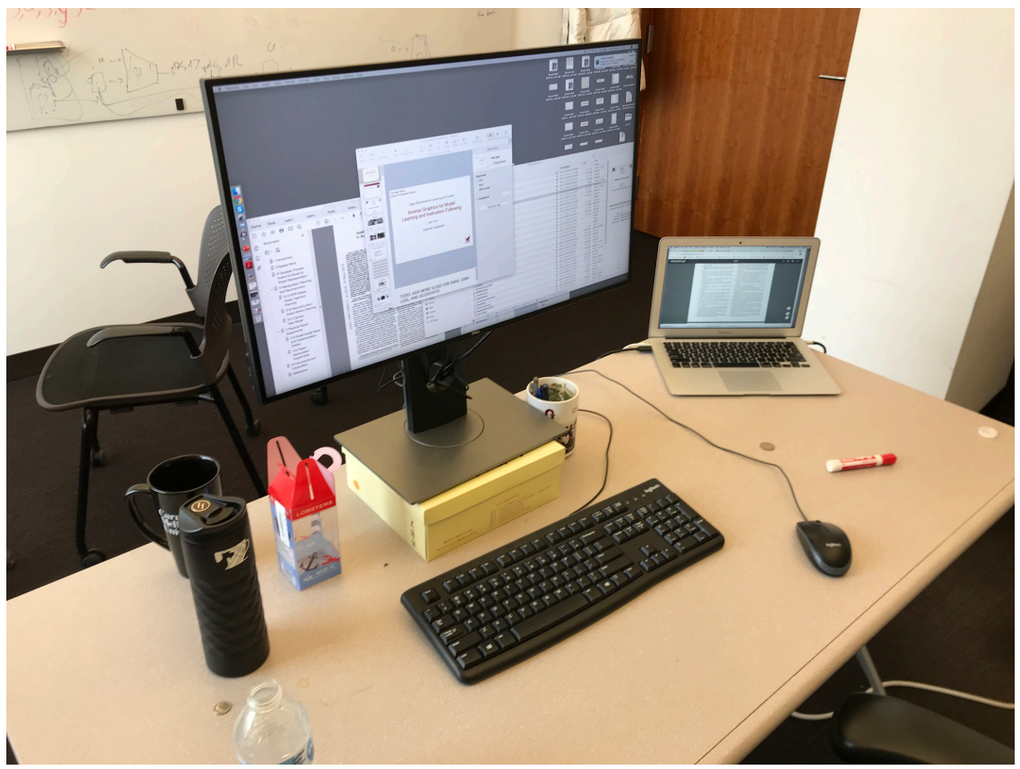
- *Object detection: infer where the objects are*
- *Free-space inference: find collision-free trajectories*
- *Affordance inference: imagine where objects can appear and where they cannot,*
- *Plan: find intermediatewaypoints to take you half way to your desired goals.*

What you cannot do (without learning):

- ***Plan/reason about contacts***

``It is easy to infer the dynamics of a scene, to build an approximate mental simulator for it”

Chris Atkeson



What is though trivial to infer from a 3 dimensional representation of the scene, it requires lots of labelled data to learn if you are in 2D…

Learning/reasoning about space from 2D images is hard because:

- There are projection artifacts: foreshortening (it’s not easy to know which object is close to which object and how much free space is there between them)
- There is no object permanence: objects disappear at occlusions
- Objects ``move” with camera motion
- Objects change size during camera zoom in / zoom out motion

Roboticists like 3D representations

ORB-SLAM 2.0

- Scene and camera motion are disentangled
- Object permanence: objects do not disappear from the map as the camera moves around

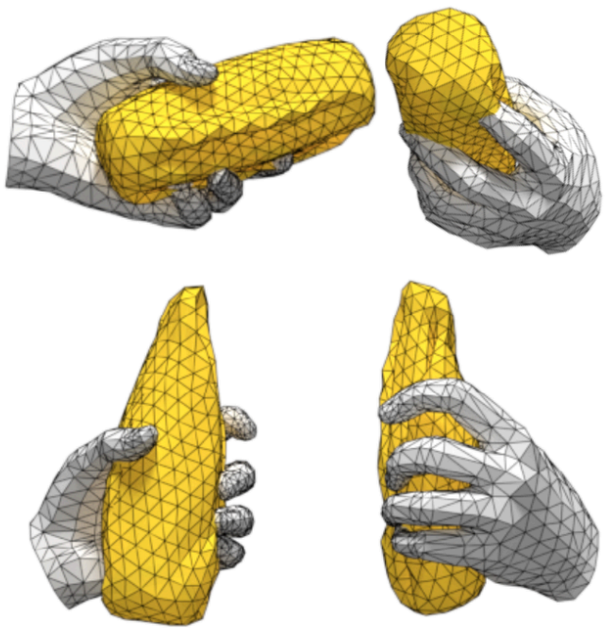
Self-driving uses 3D representations

Why don’t we try to map input images to 3D object models?

For self-driving we have done that, why not for object manipulation?

3D meshing the world

Why don’t we try to map input images to 3D object models?



Do we know now how to place the bottle on the table?

*Beyond free-space, there is a lot of knowledge about contacts that 3D meshed versions of the world cannot **readily** provide, we need to learn those from 2D or 3D meshed data!*

(while we do not need to learn free space in general in 3D meshed data: it comes from free)

And self-driving cars care about free space and intended motion for objects

Learning Joint Reconstruction of Hands and Manipulated Objects, CVPR 2019

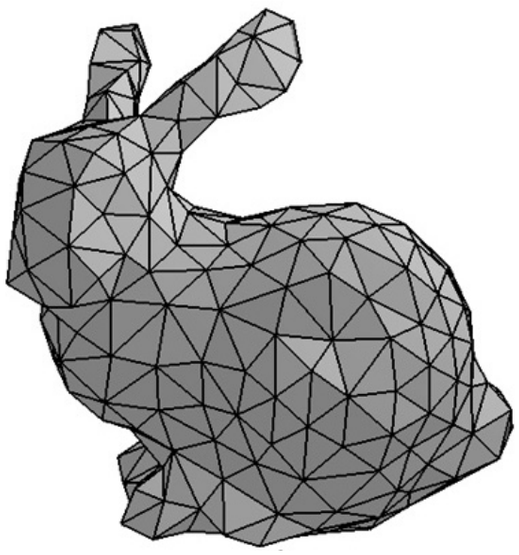
3D models are impossible and unecessary

``Internal world models which are complete representations of the external environment, besides being impossible to obtain, are not at all necessary for agents to act in a competent manner.”

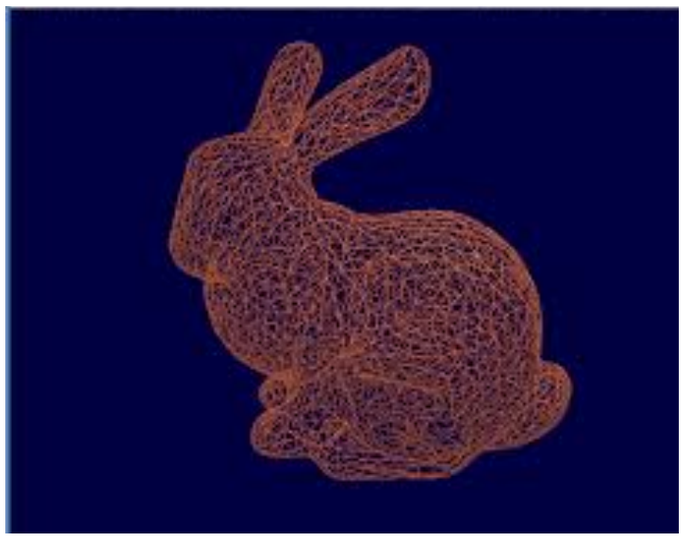
Intelligence without reason, IJCAI, Rodney Brooks (1991)

3D models are impossible and unecessary

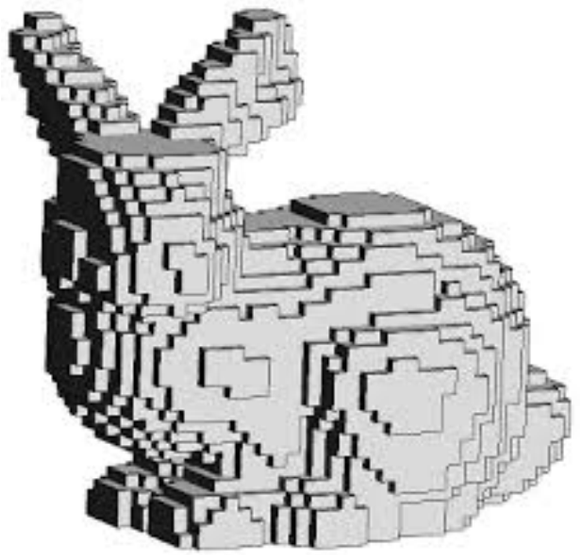
3D mesh



3D pointcloud



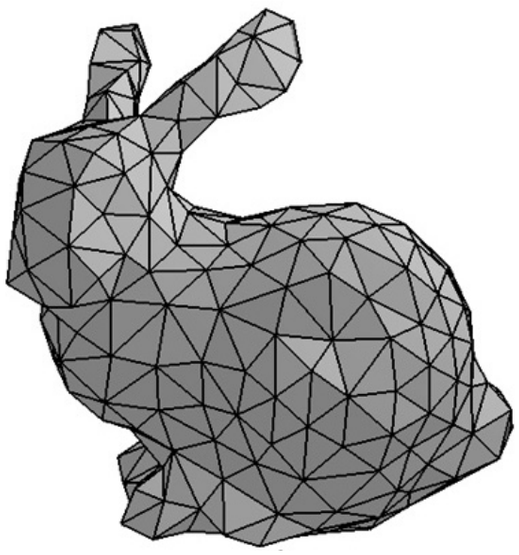
3D voxel occupancy



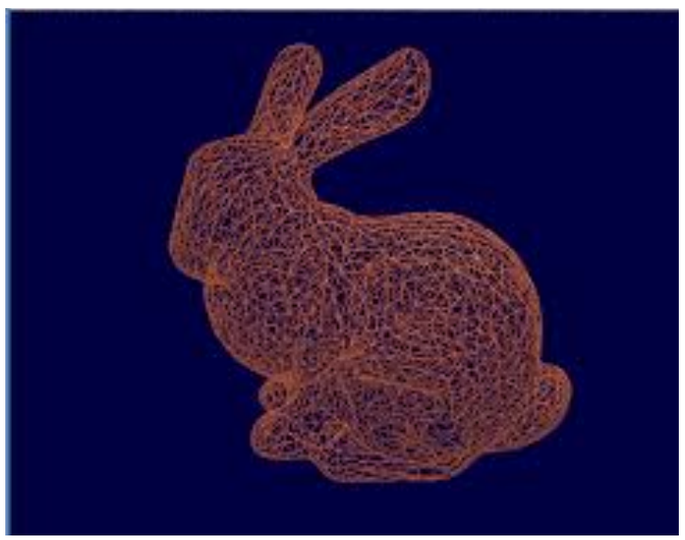
- They do not optimize the right end task: catching the rabbit.
- They optimize for 3D reconstruction quality
- 2D image to 3D mesh reconstruction requires a lot of human labelled data

What we want

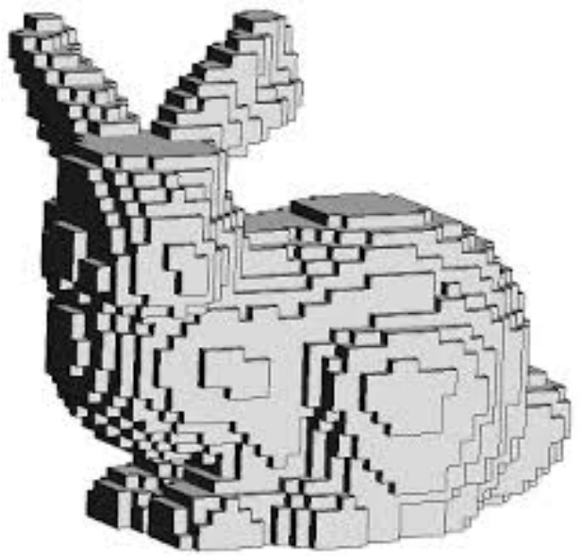
3D mesh



3D pointcloud



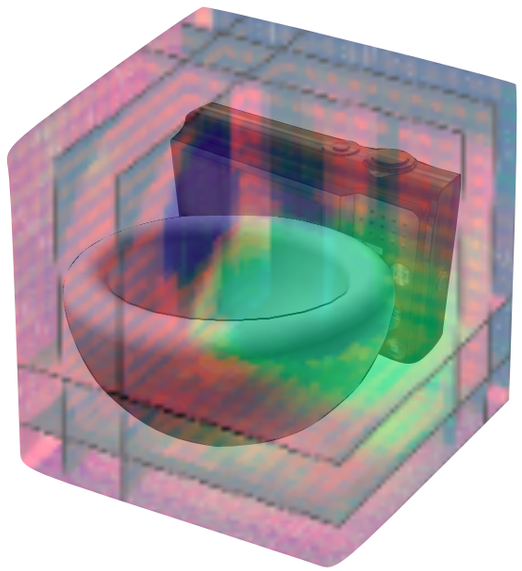
3D voxel occupancy



- We need to link 3D representations with the end-task of behaviour learning.
- We should be able to learn without any human supervision.

To 3D or not to 3D?

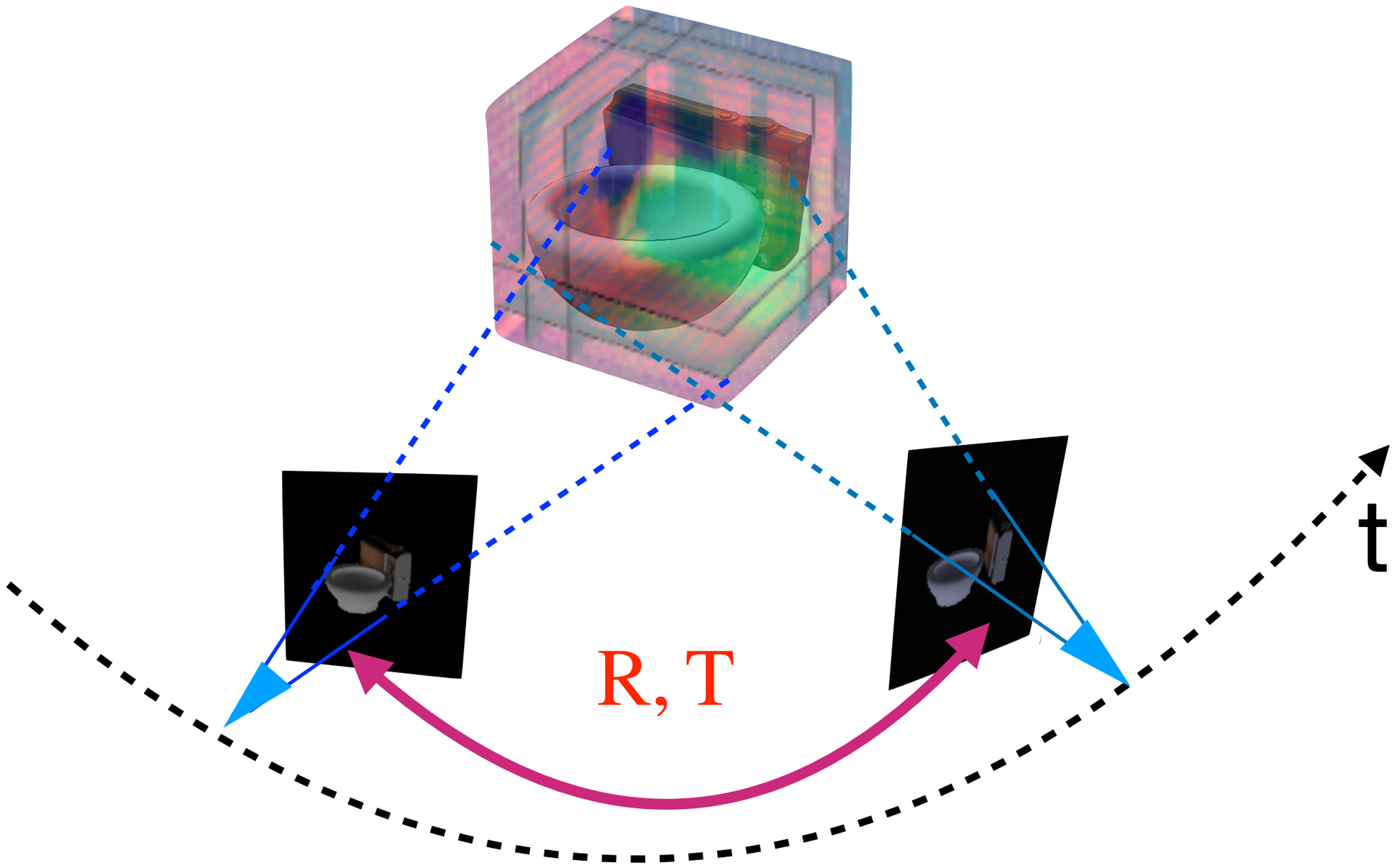
3D feature maps



$$H \times W \times D \times C$$

3 spatial dimensions, multiple feature dimensions

Geometry-Aware Recurrent Networks

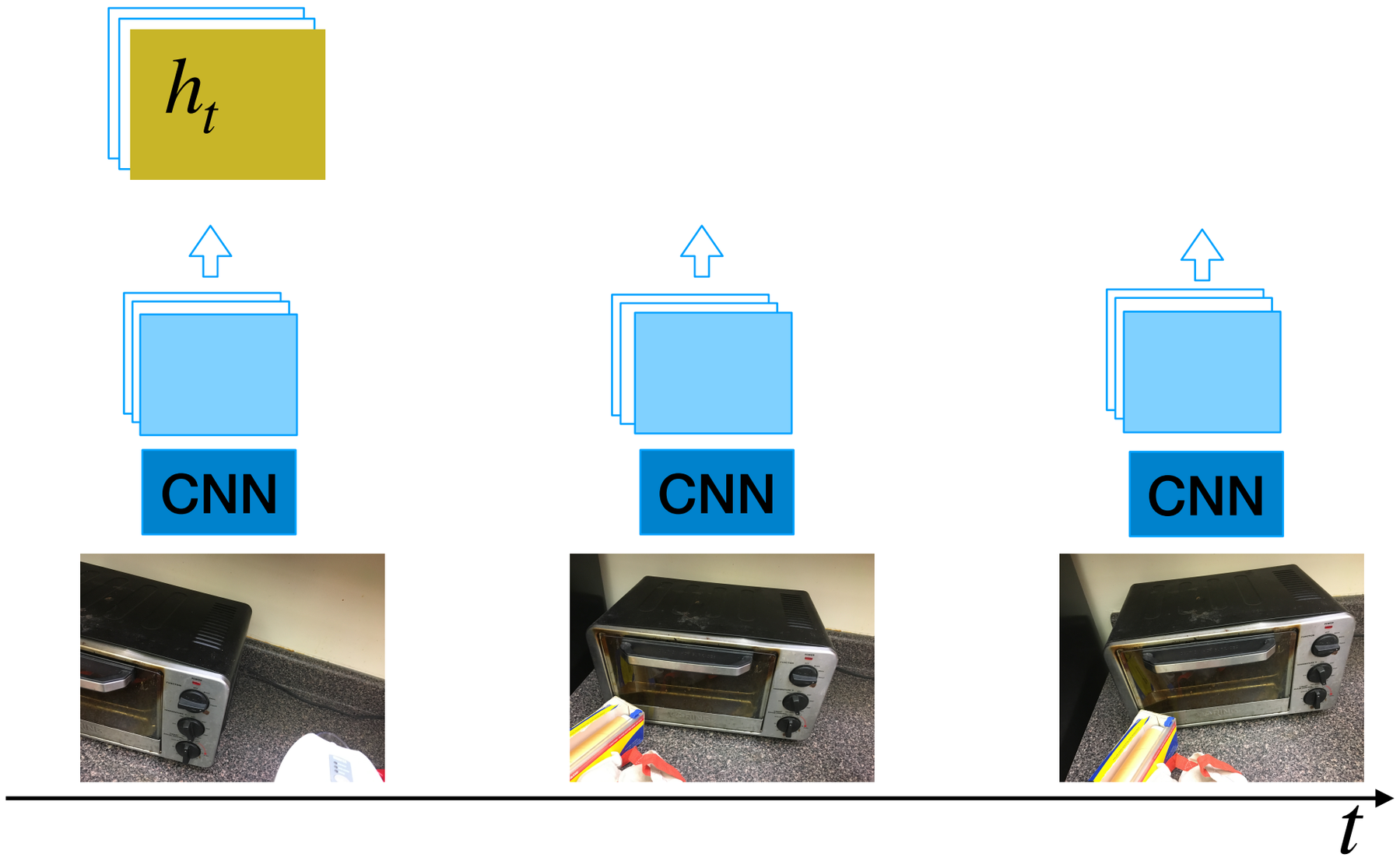


1. Hidden state: 3D feature maps
2. Egomotion-stabilized hidden state updates

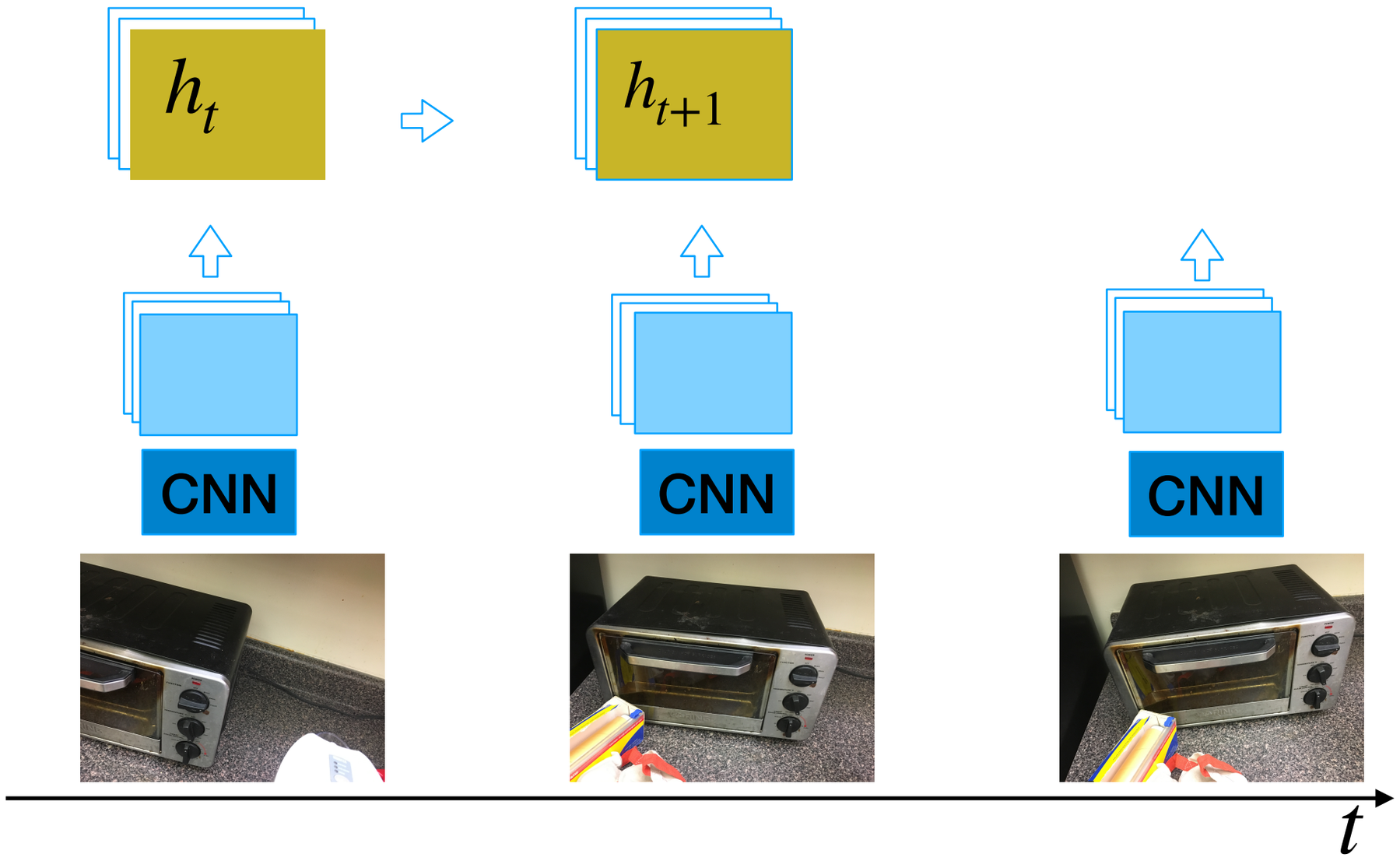
2D RNNs (conv-LSTMs/GRUs)

2D RNNs (conv-LSTMs/GRUs)

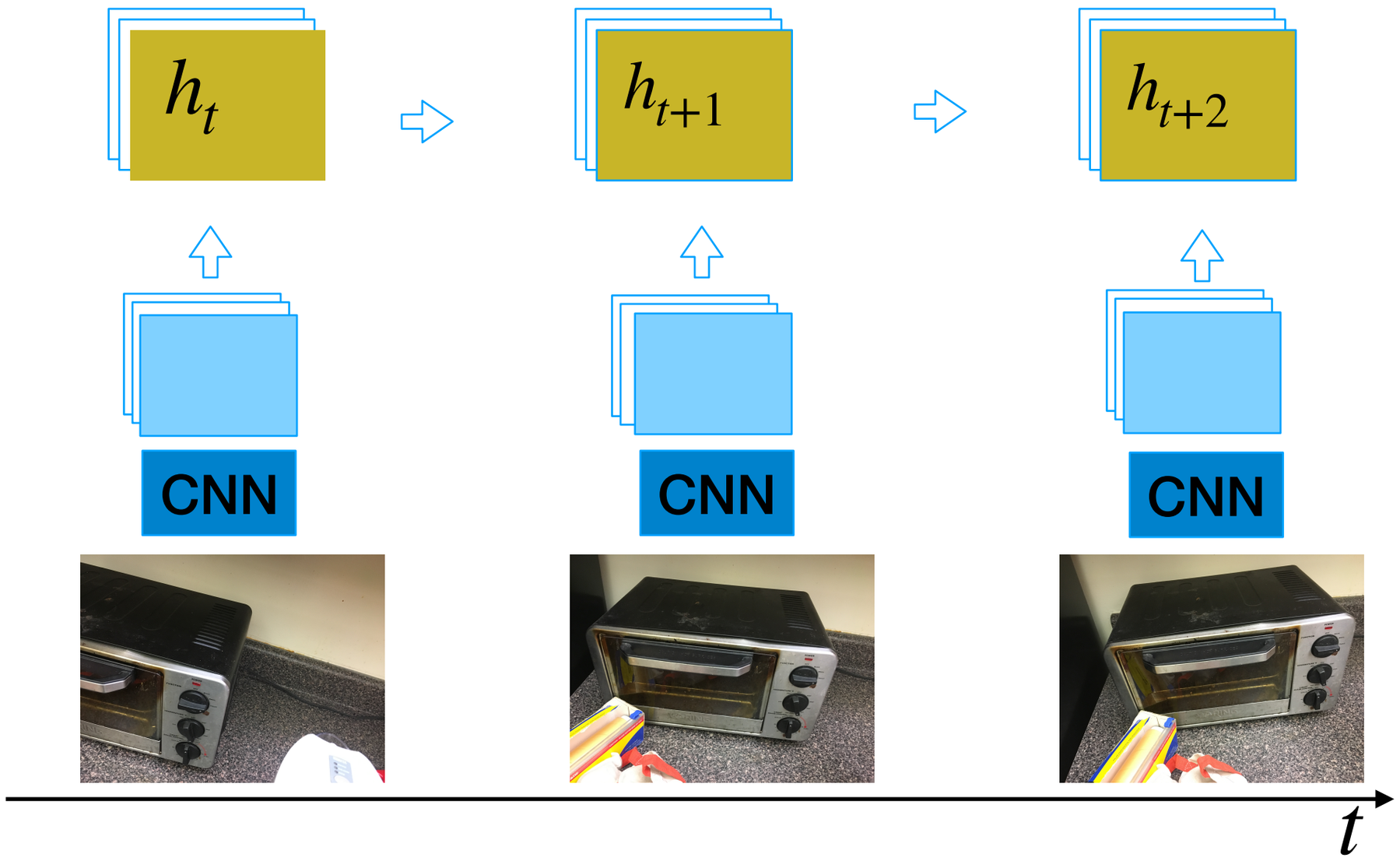
2D RNNs (conv-LSTMs/GRUs)



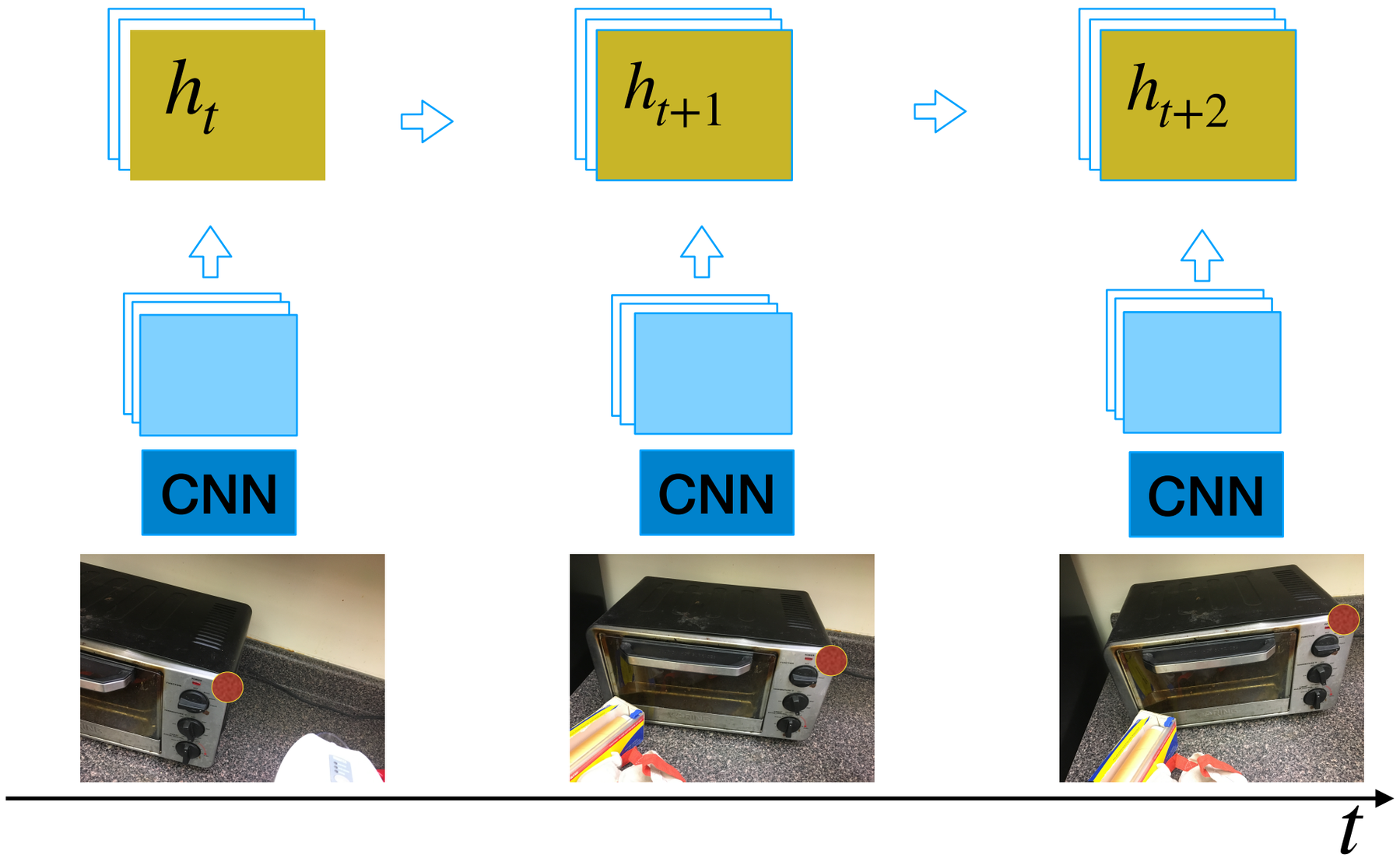
2D RNNs (conv-LSTMs/GRUs)



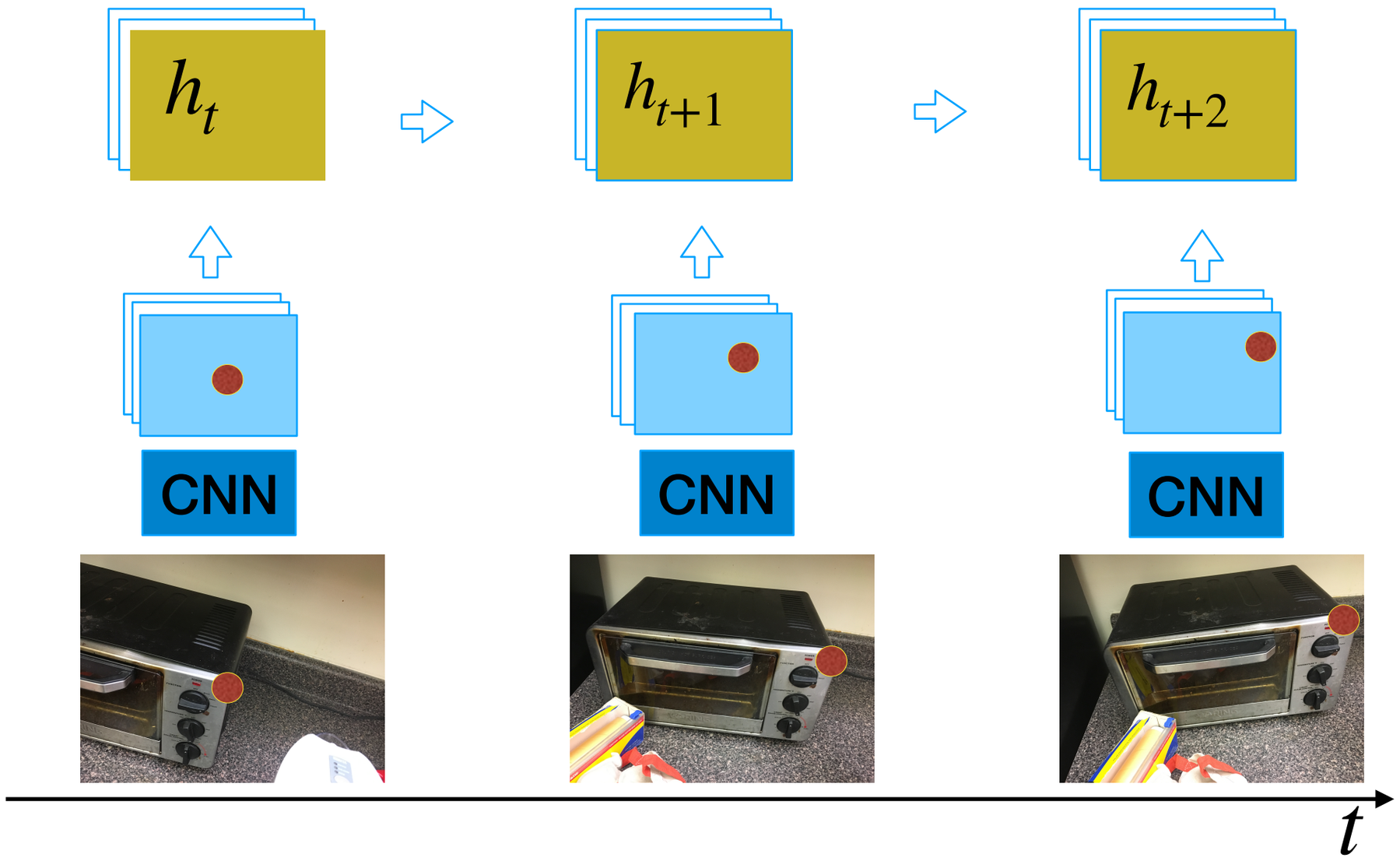
2D RNNs (conv-LSTMs/GRUs)



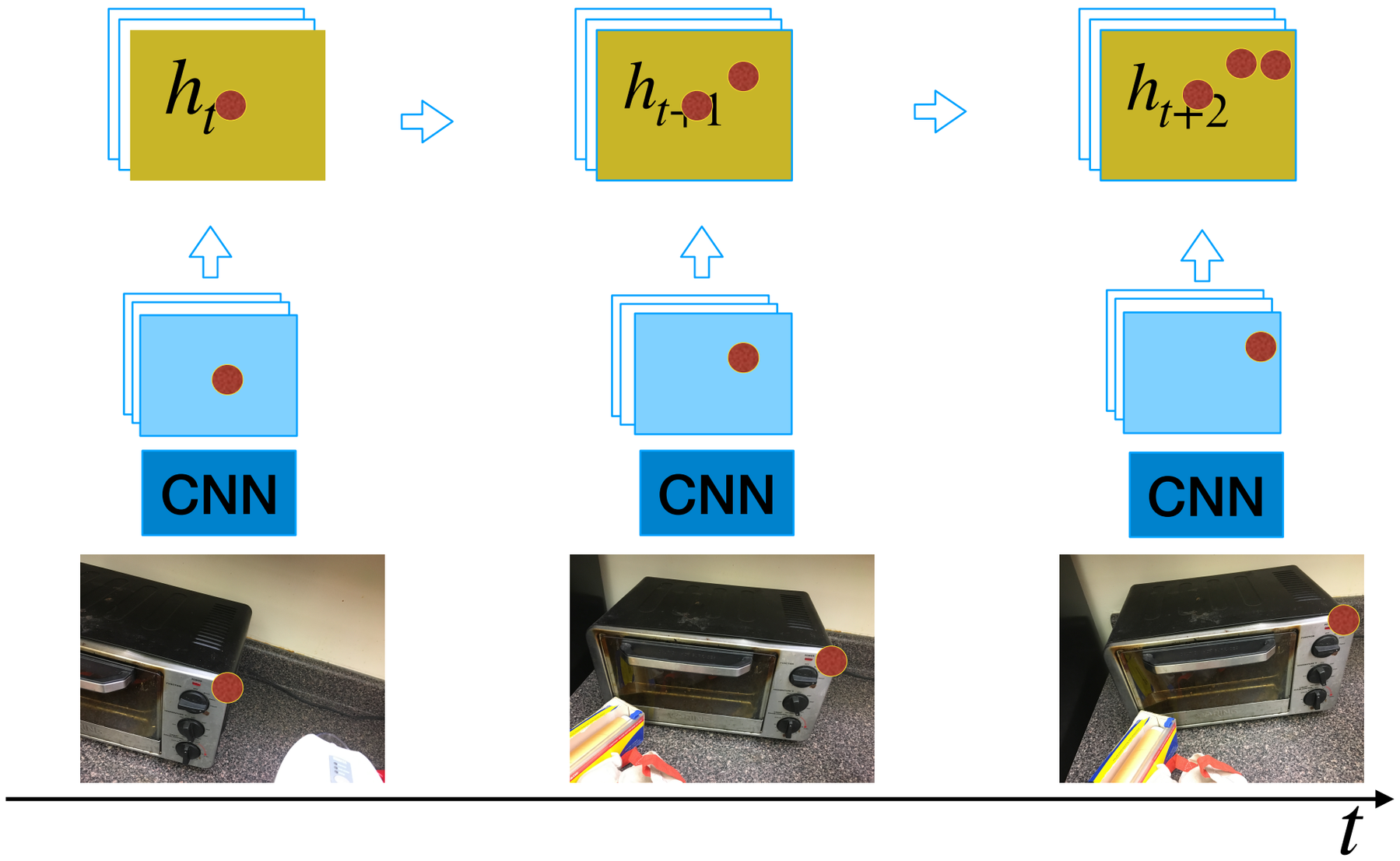
2D RNNs (conv-LSTMs/GRUs)



2D RNNs (conv-LSTMs/GRUs)



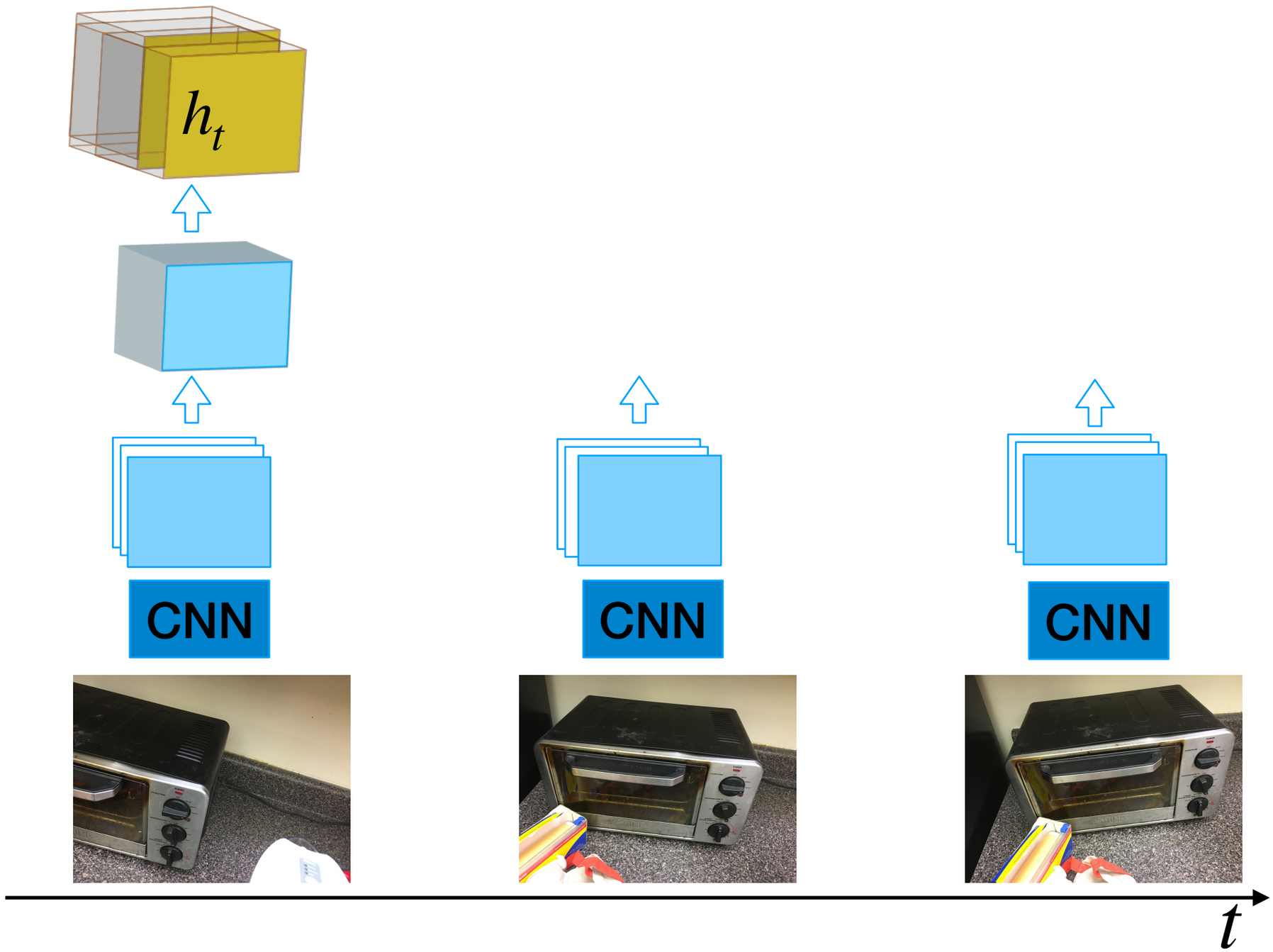
2D RNNs (conv-LSTMs/GRUs)



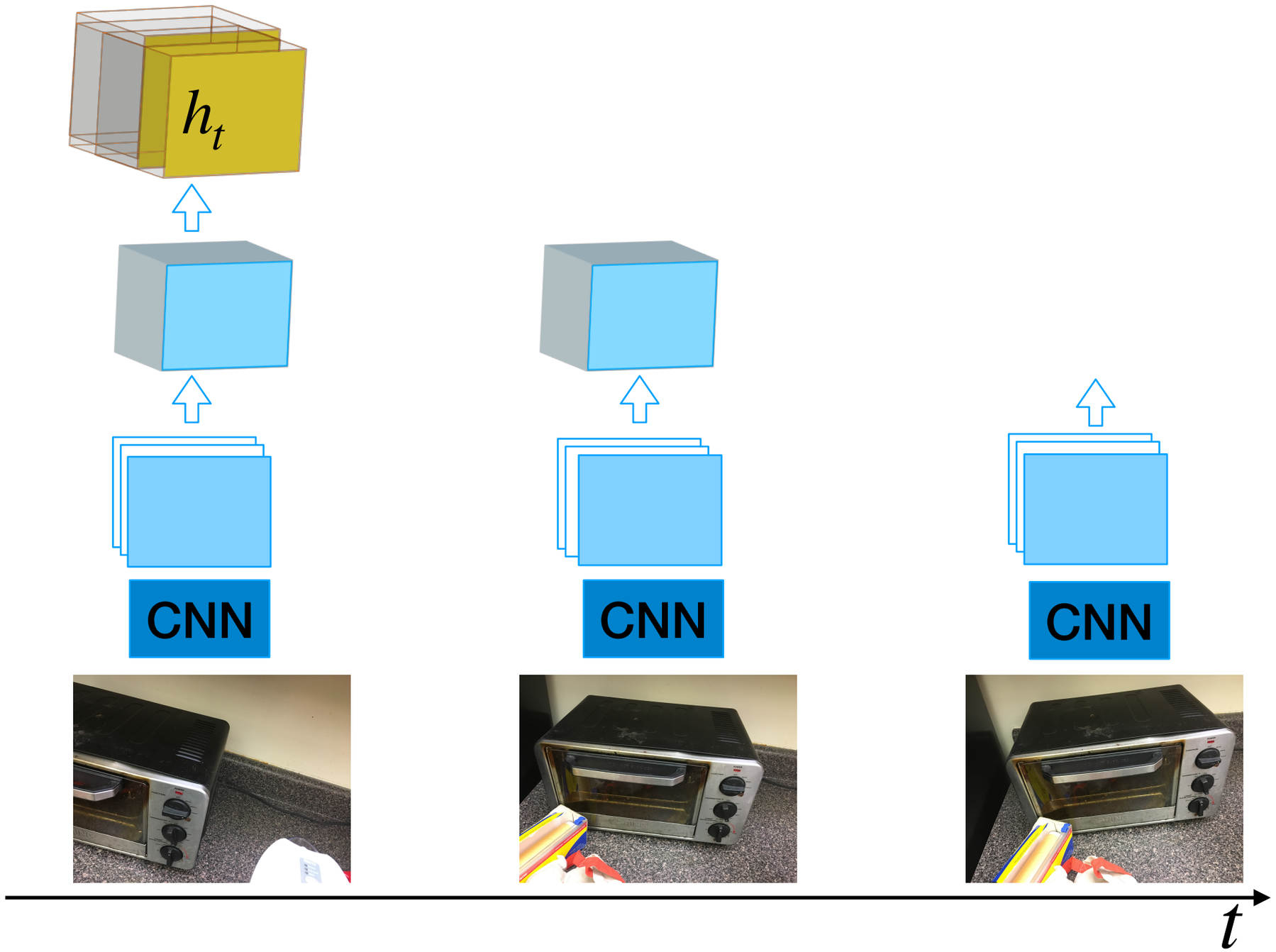
Geometry-Aware Recurrent Networks

Geometry-Aware Recurrent Networks

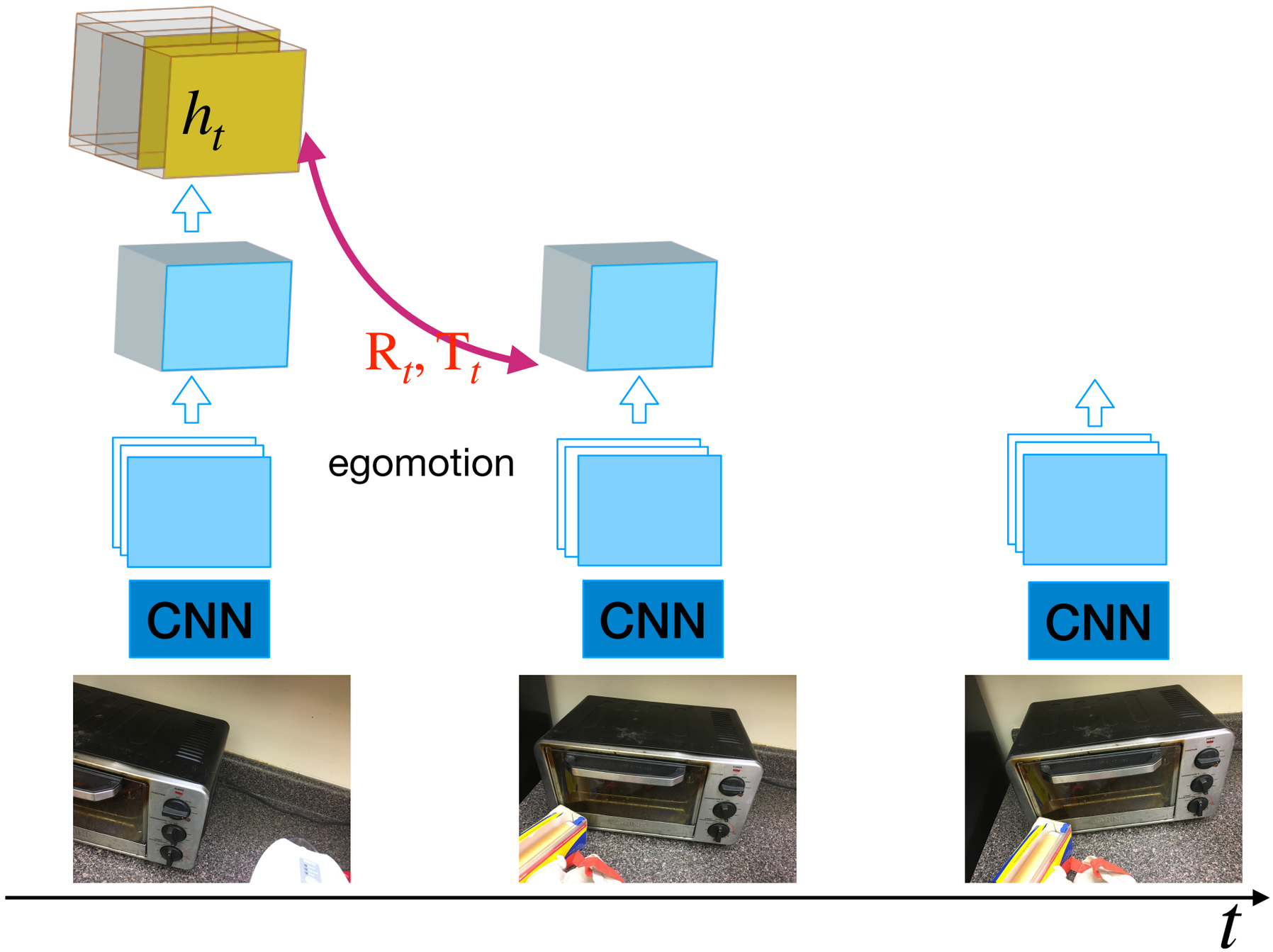
Geometry-Aware Recurrent Networks



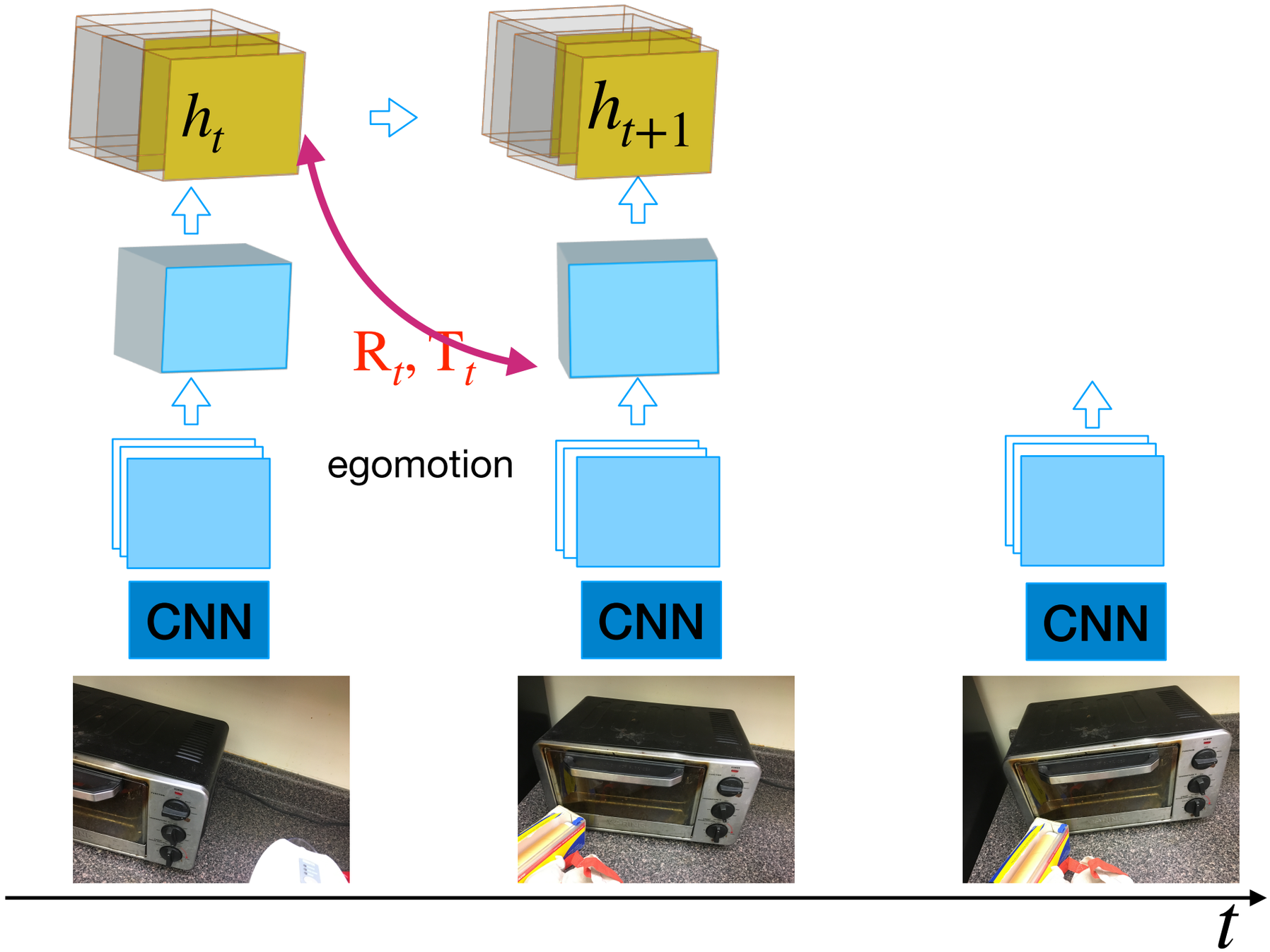
Geometry-Aware Recurrent Networks



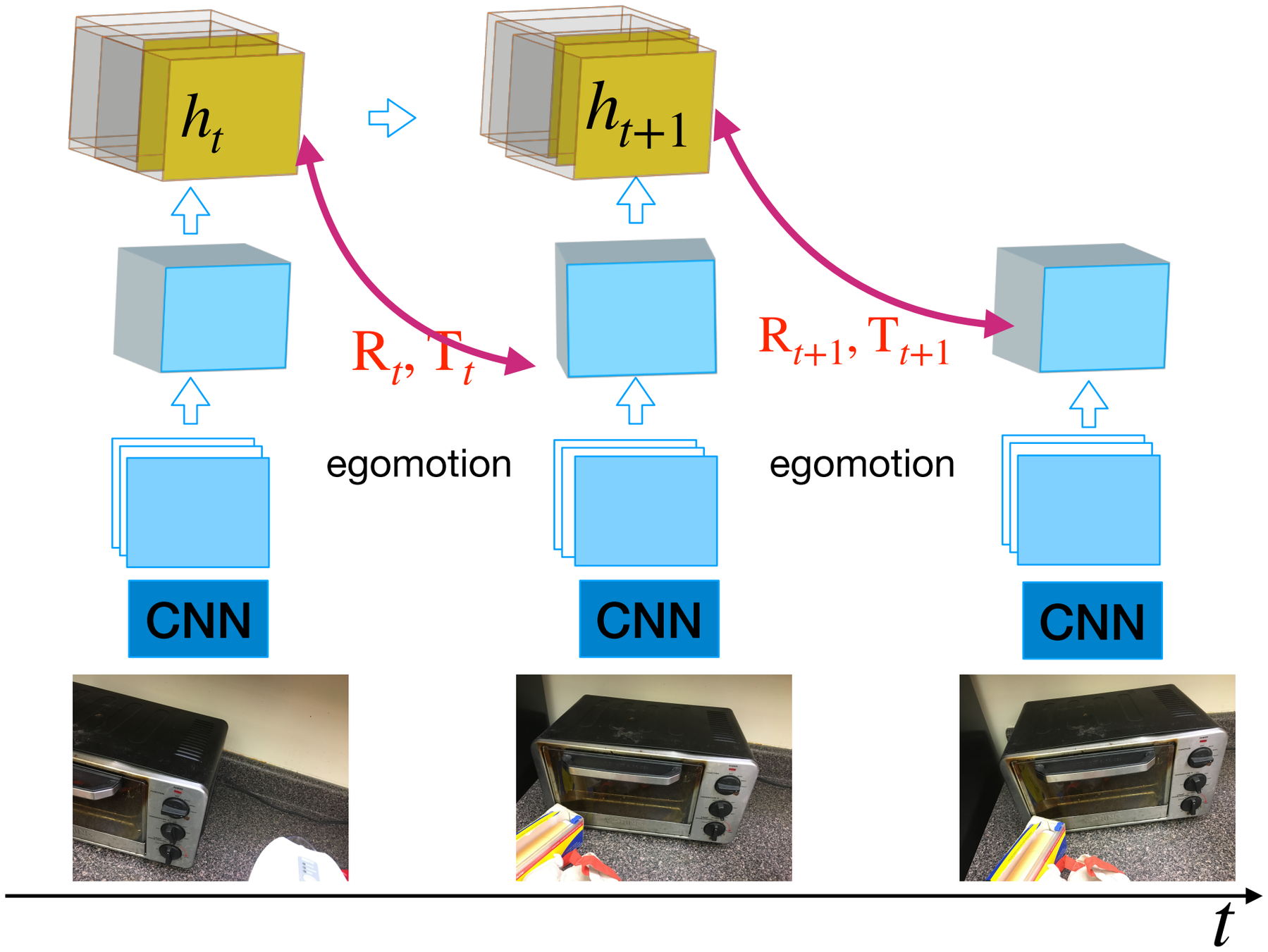
Geometry-Aware Recurrent Networks



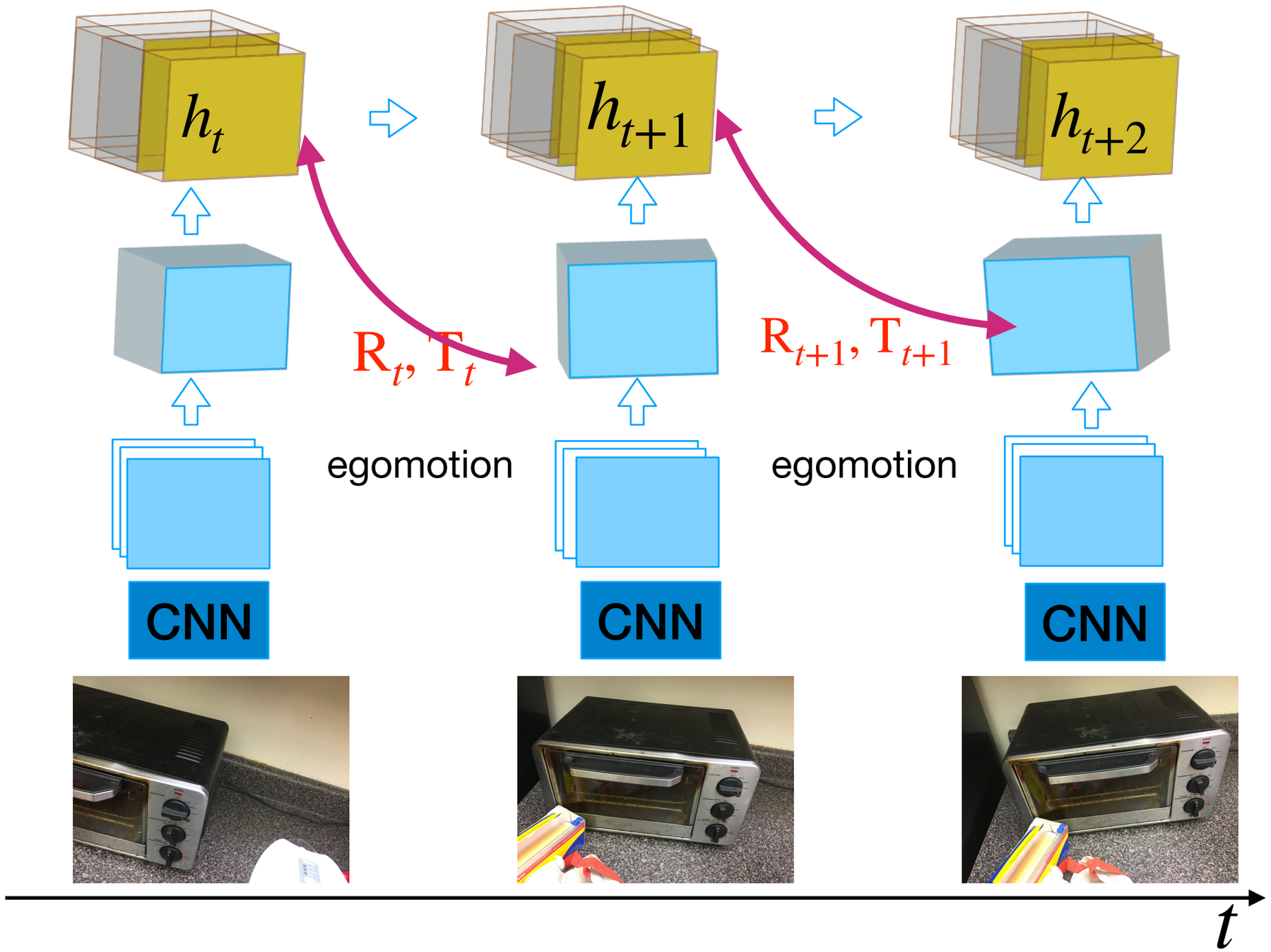
Geometry-Aware Recurrent Networks



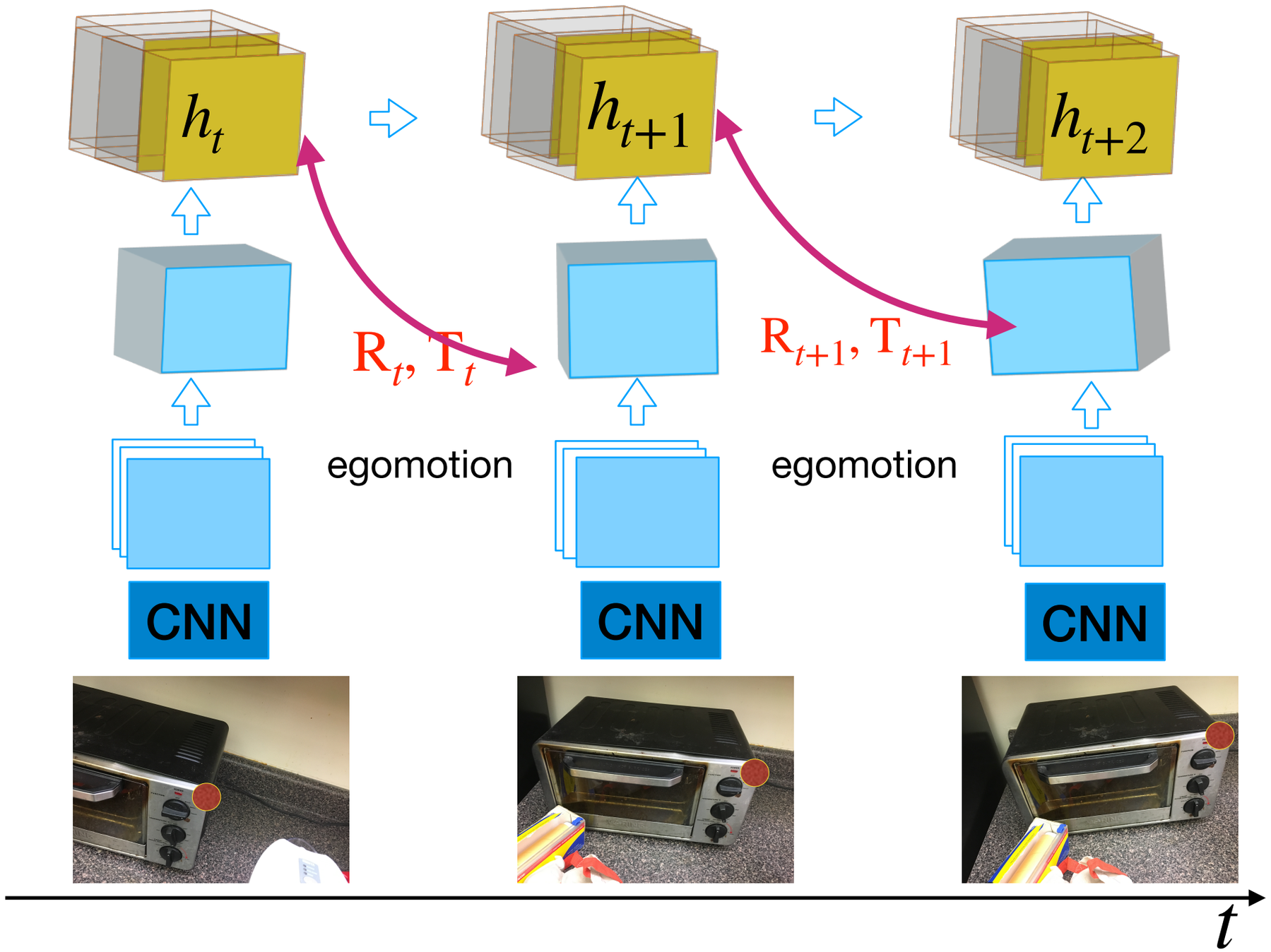
Geometry-Aware Recurrent Networks



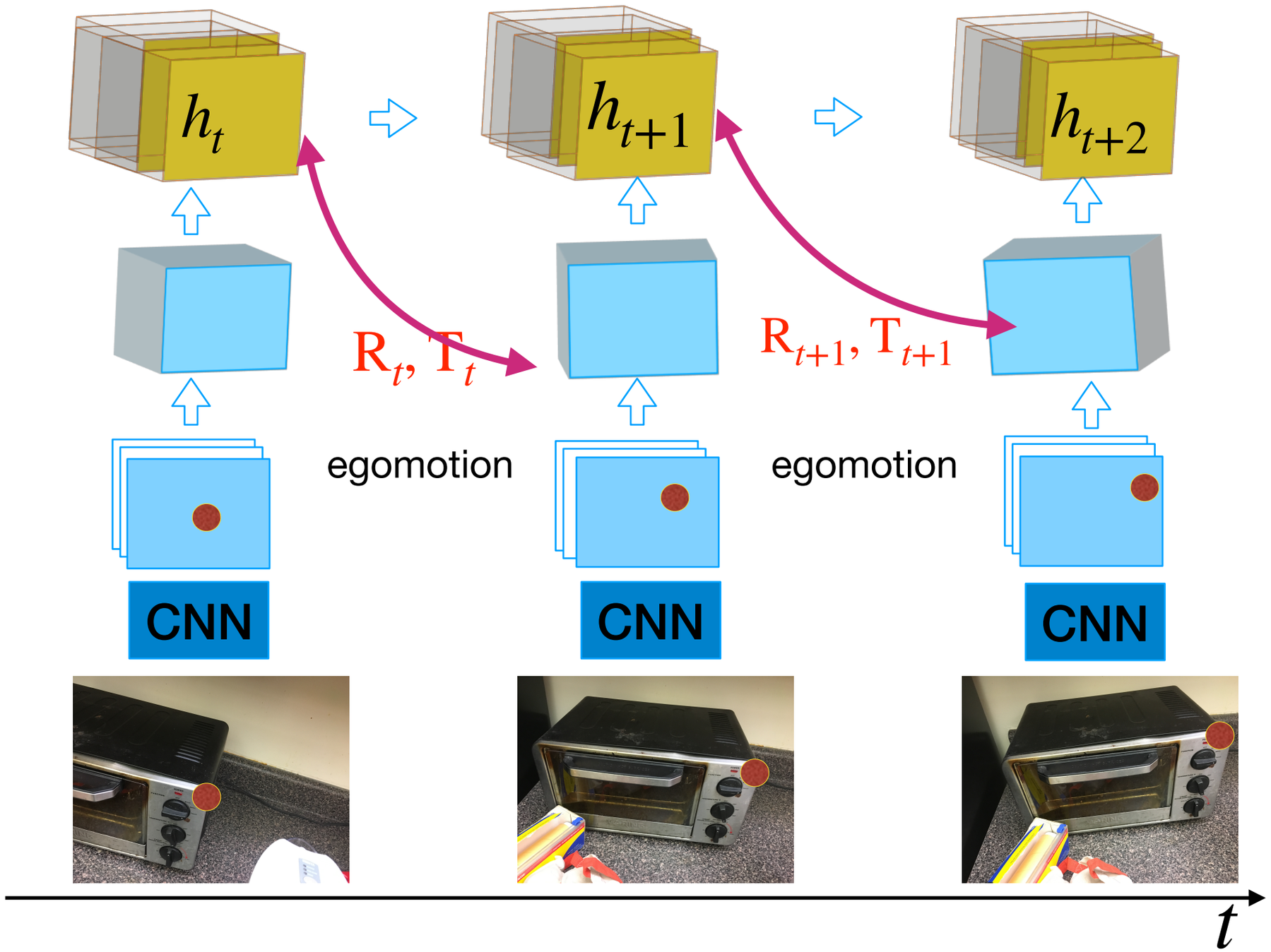
Geometry-Aware Recurrent Networks



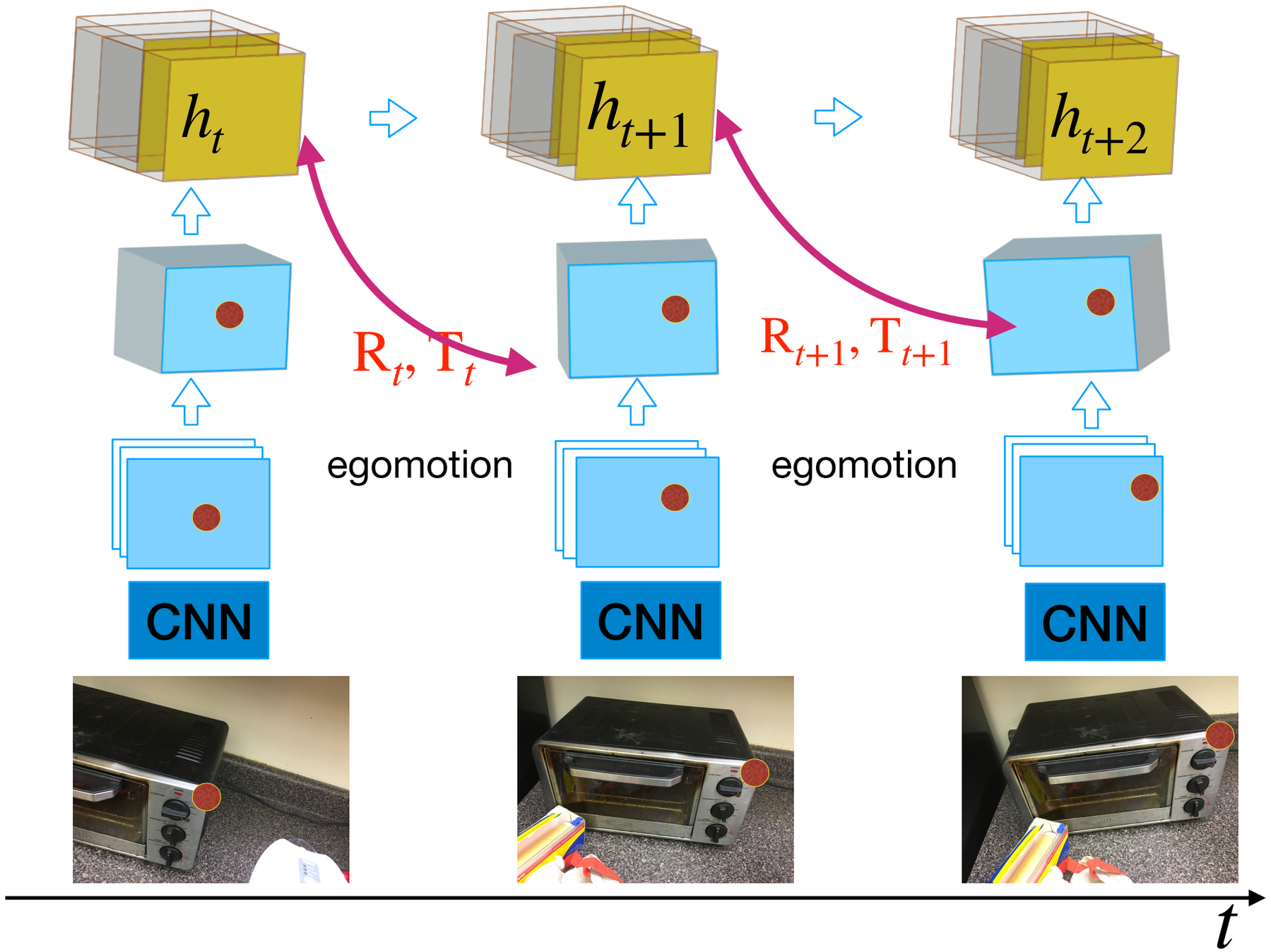
Geometry-Aware Recurrent Networks



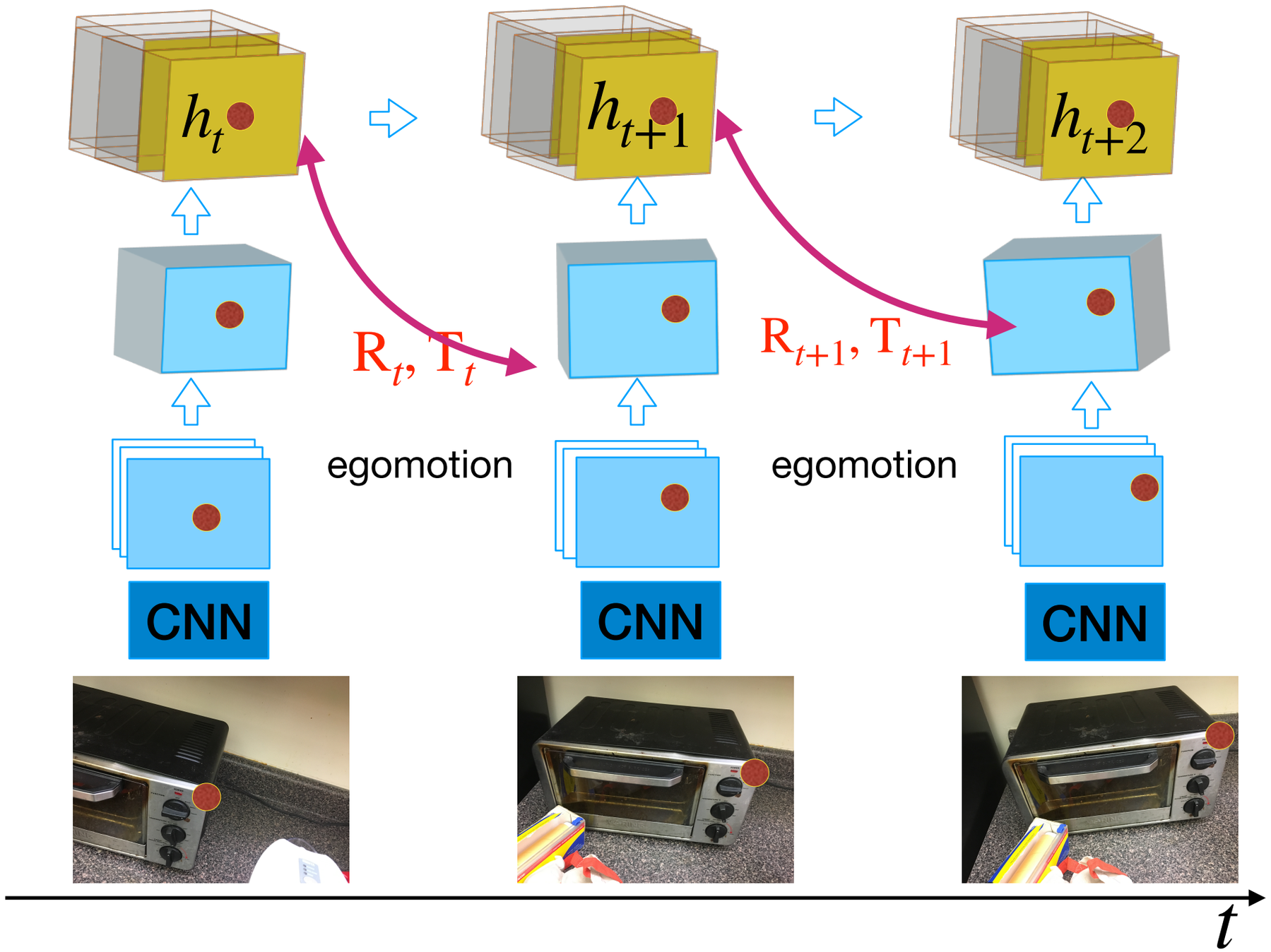
Geometry-Aware Recurrent Networks



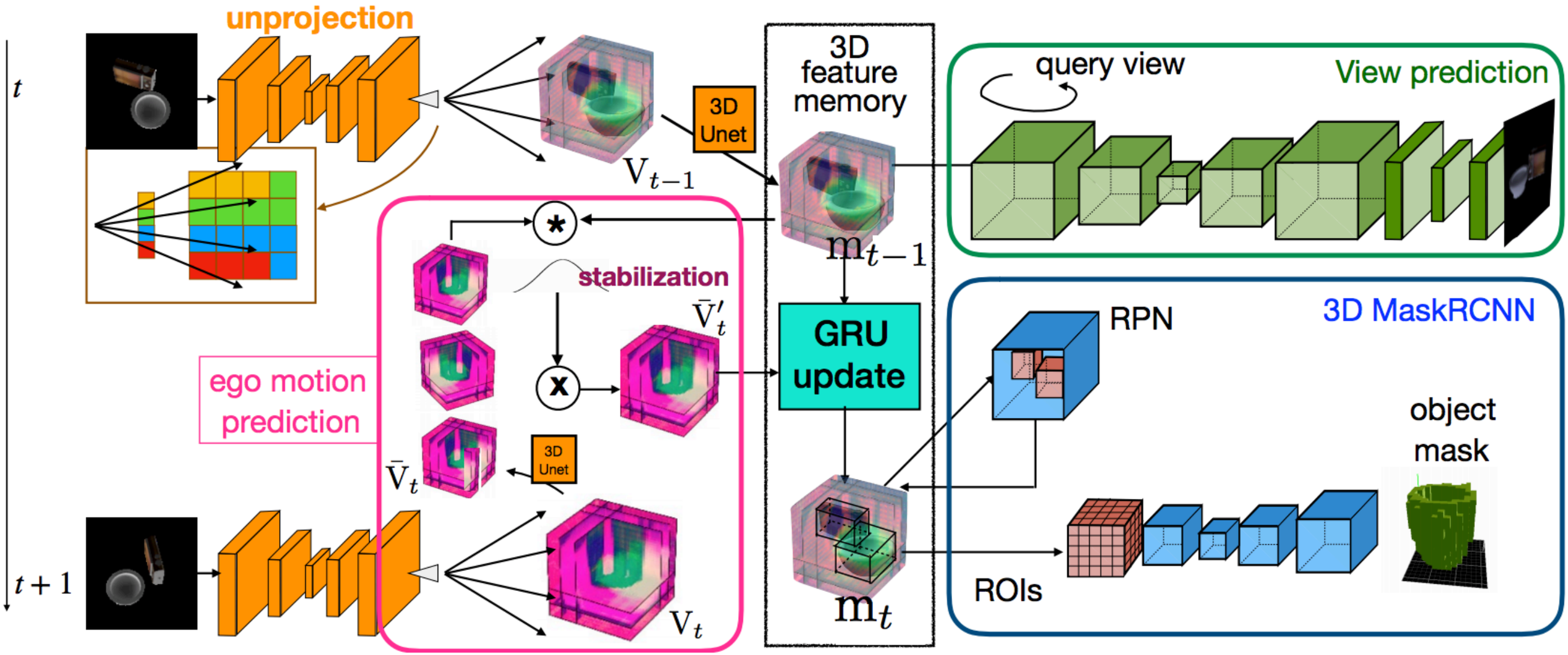
Geometry-Aware Recurrent Networks



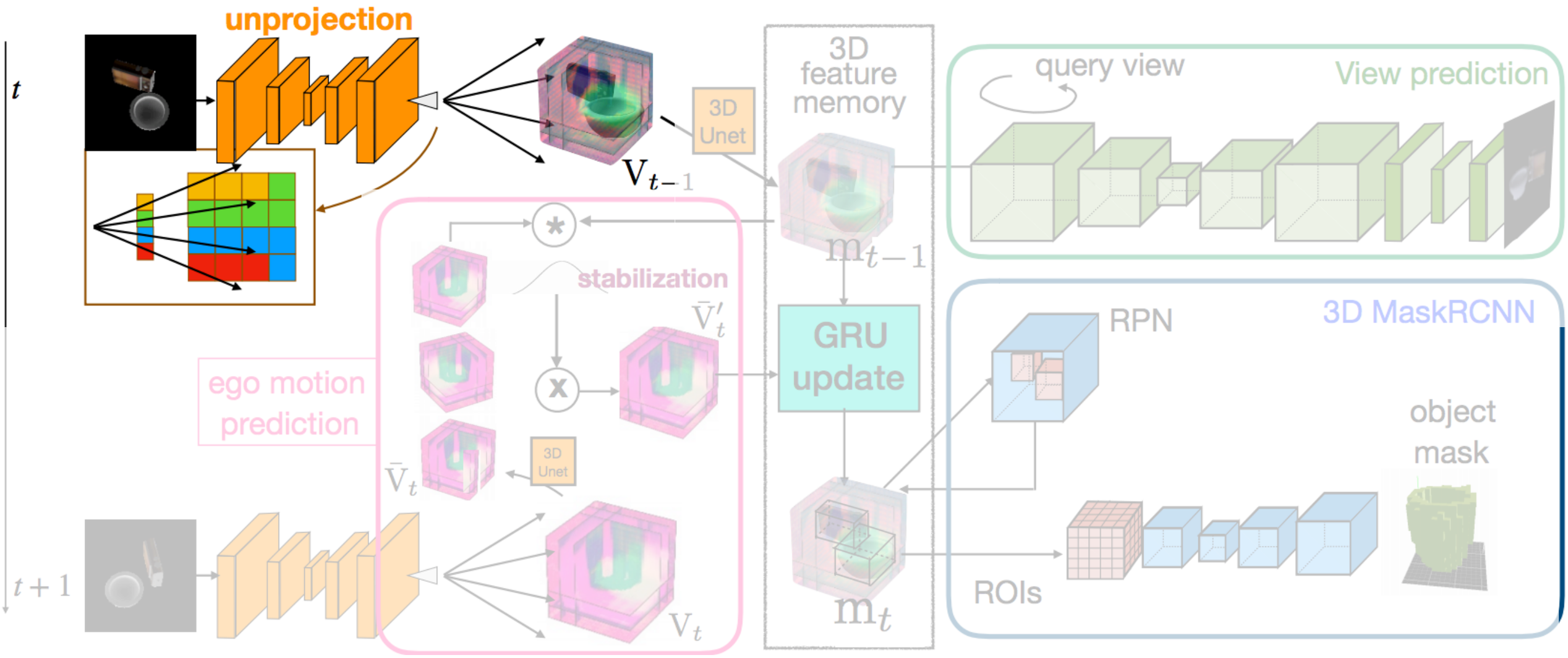
Geometry-Aware Recurrent Networks



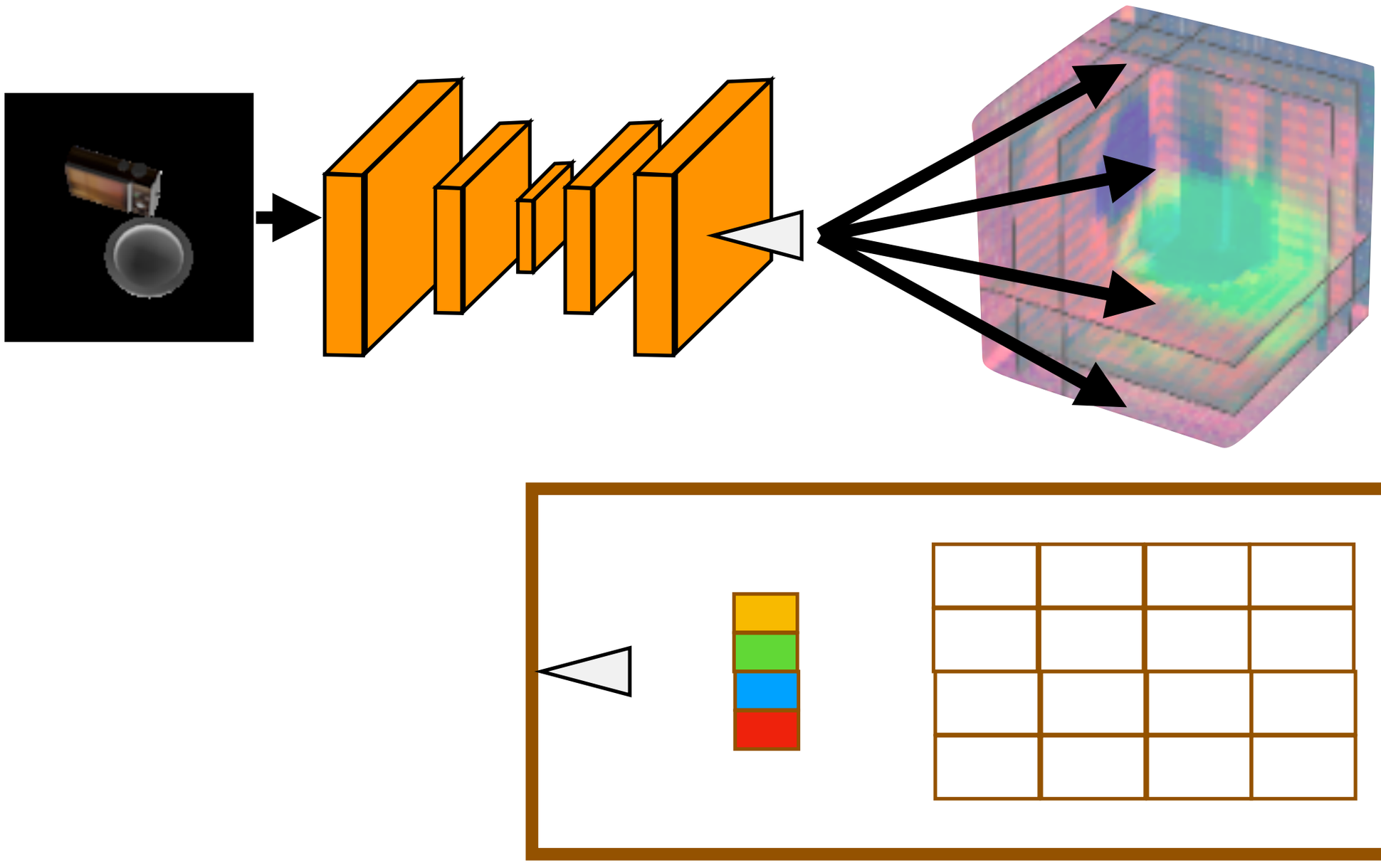
Architecture



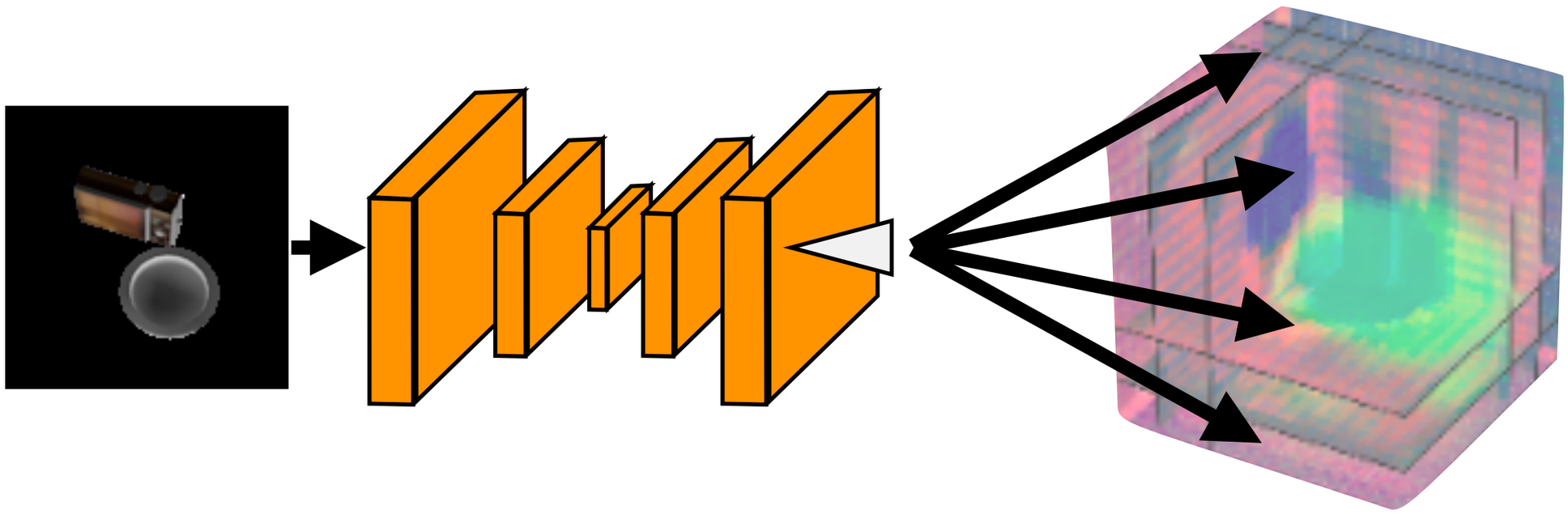
Architecture

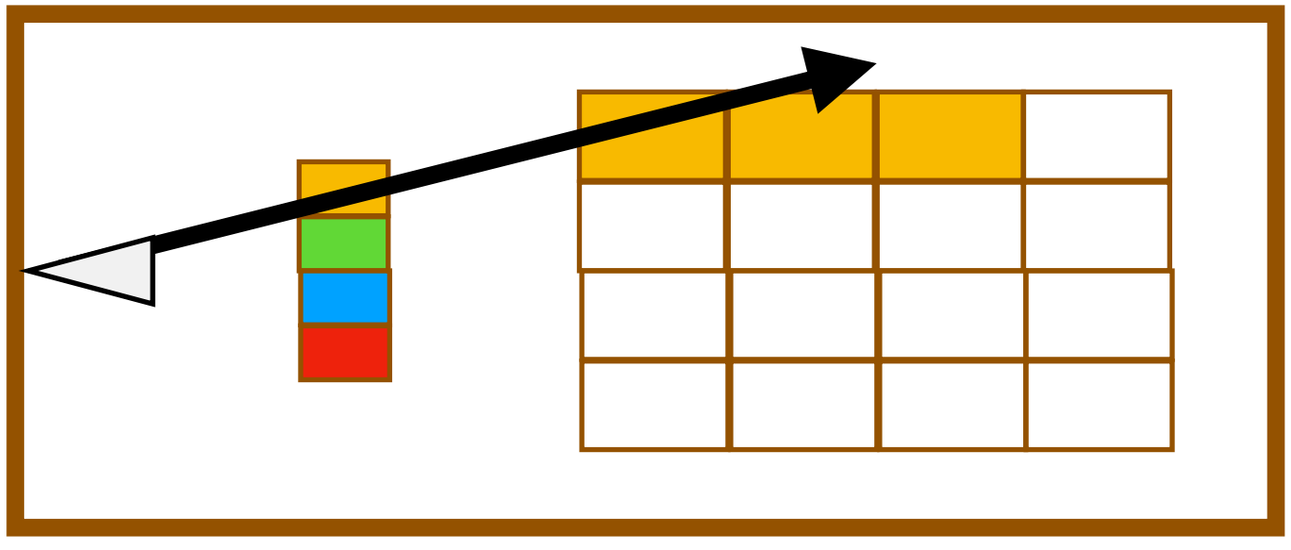


Unprojection (2D to 3D)

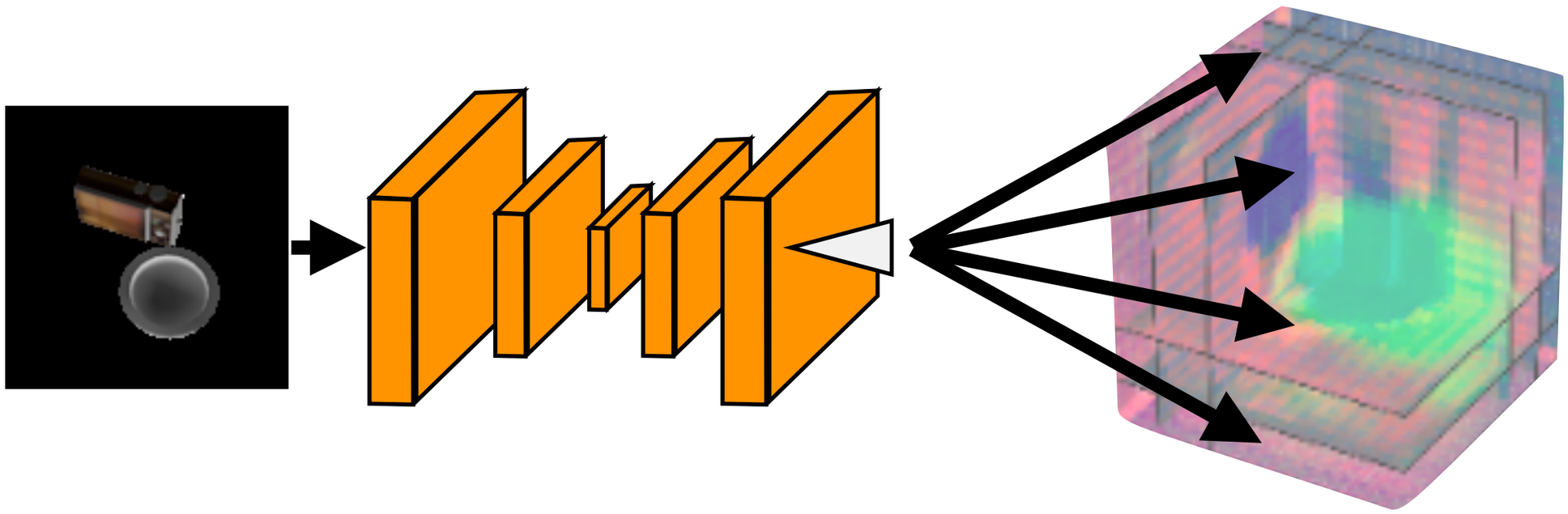


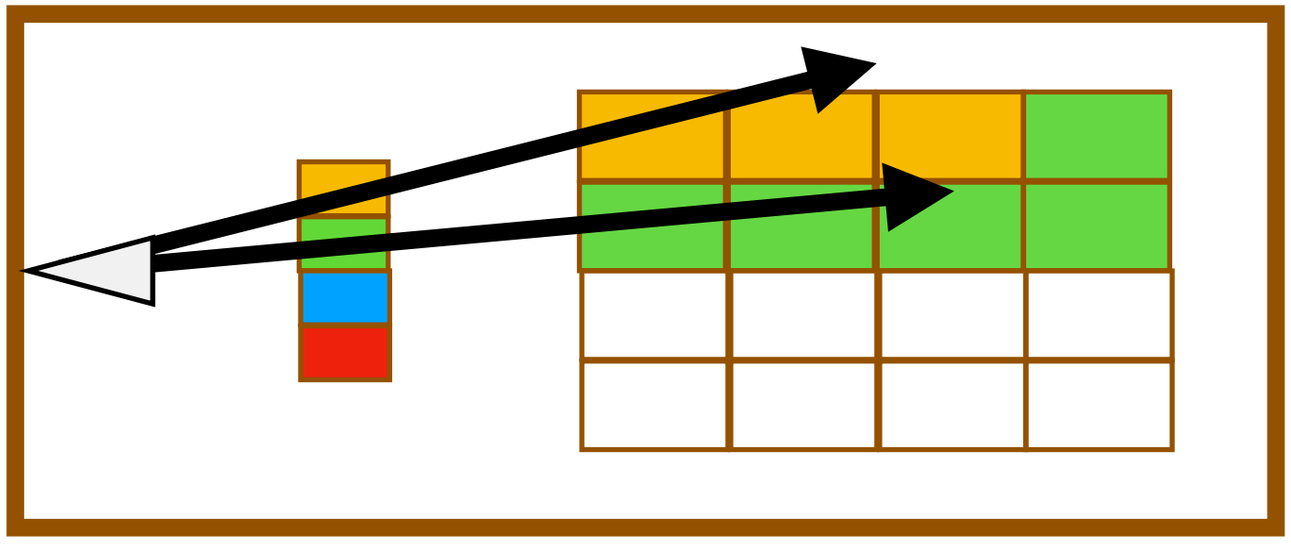
Unprojection (2D to 3D)



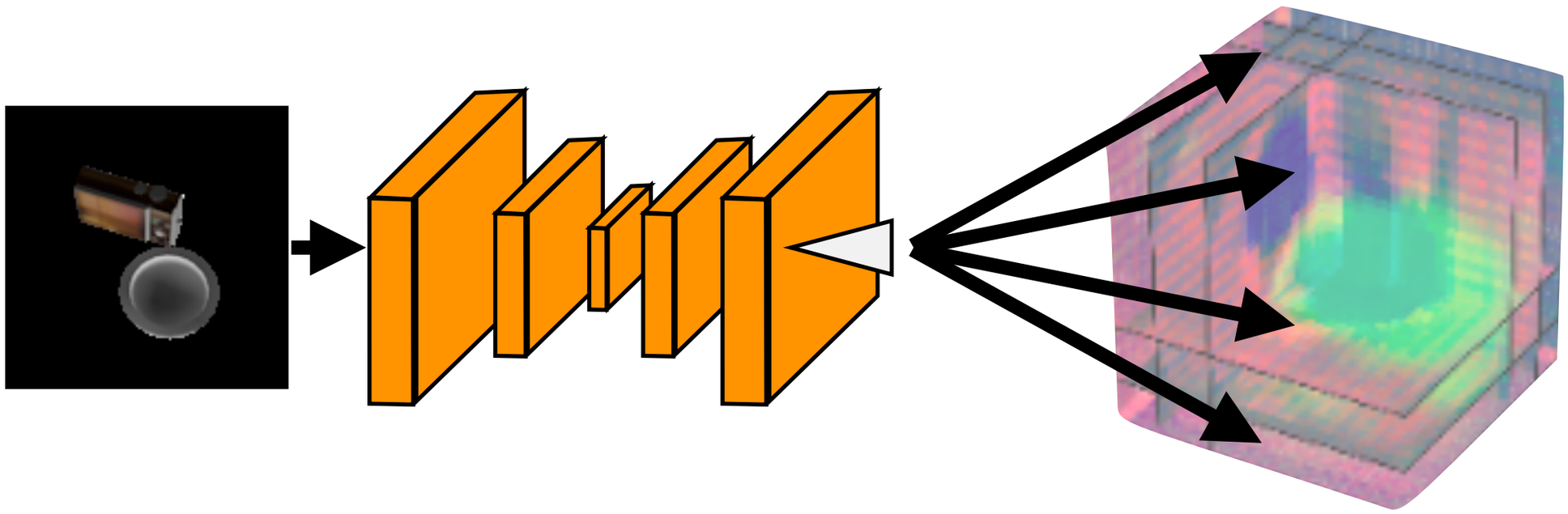


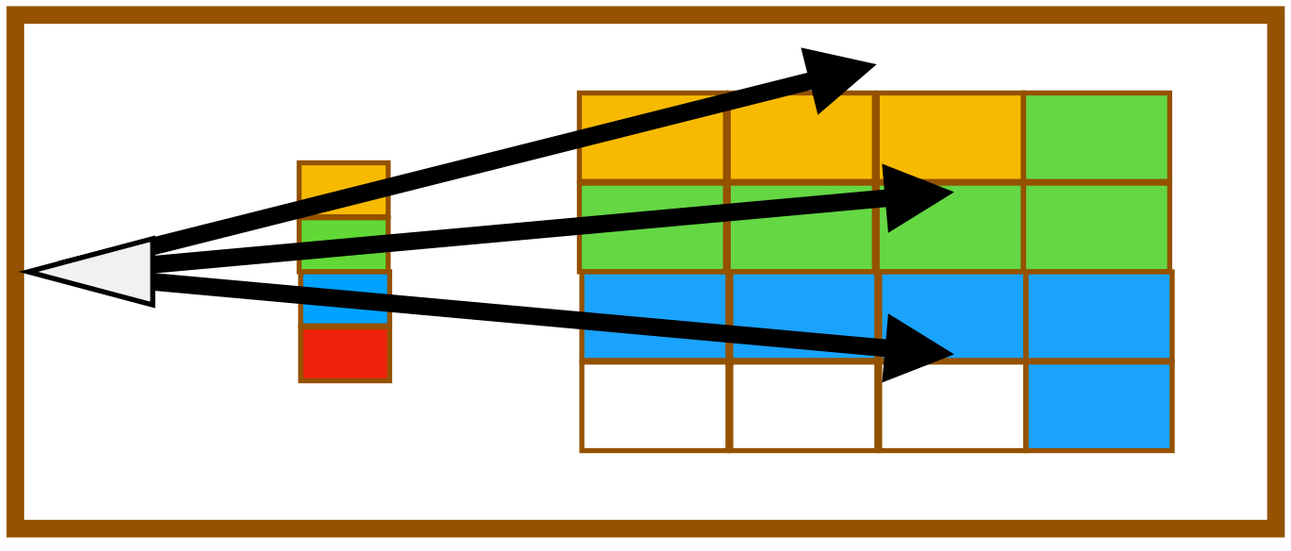
Unprojection (2D to 3D)



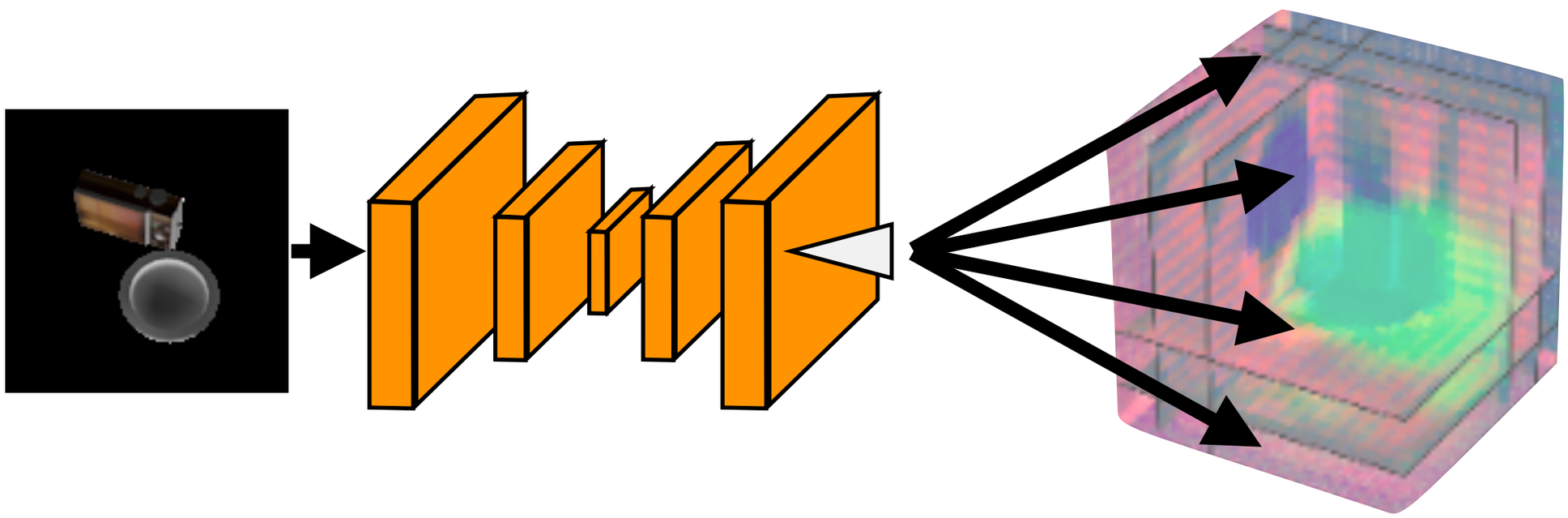


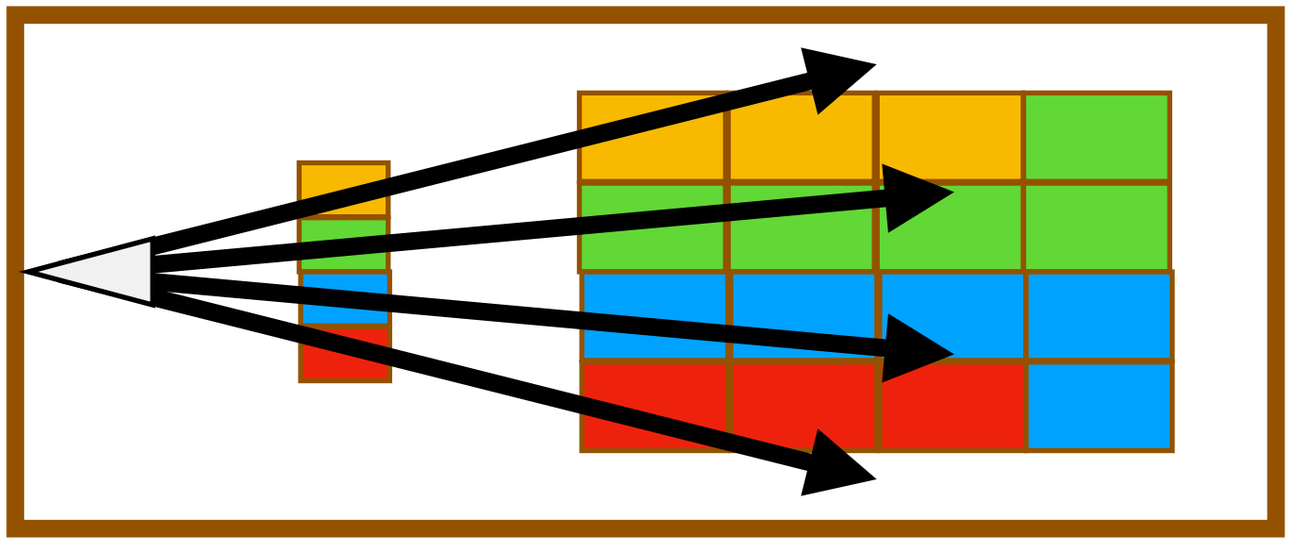
Unprojection (2D to 3D)



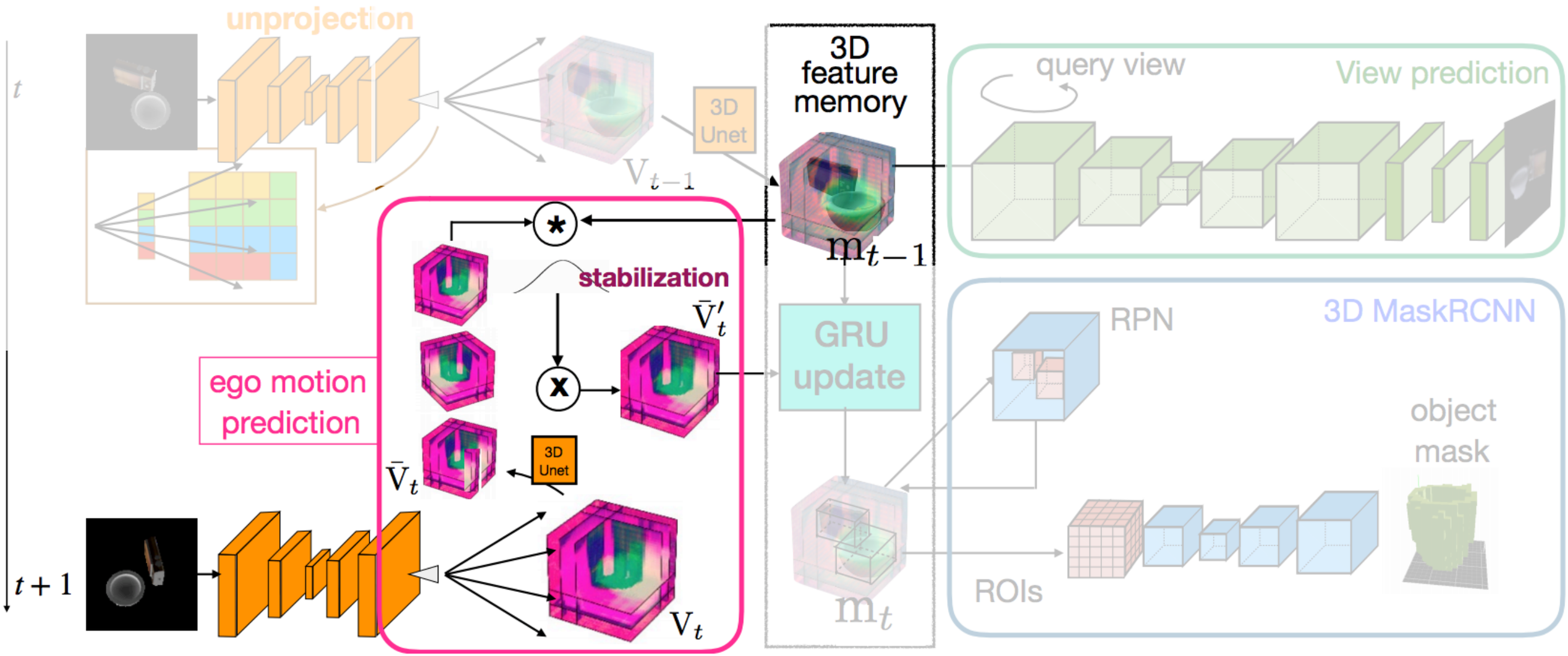


Unprojection (2D to 3D)



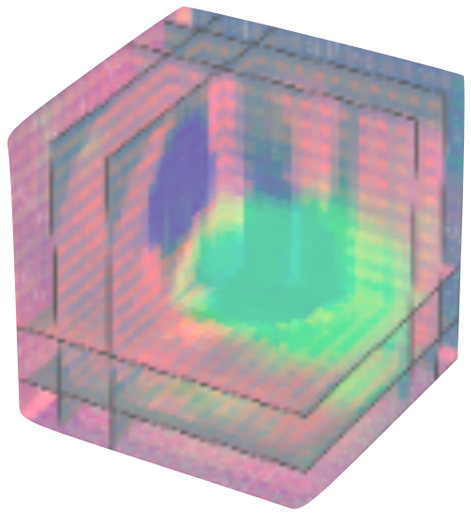


Architecture



Egomotion-stabilized memory update

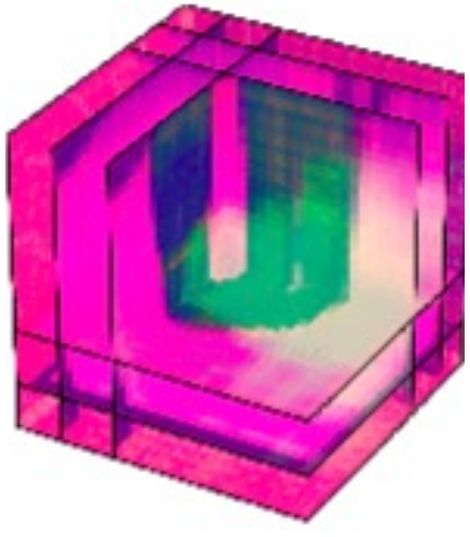
3D feature memory



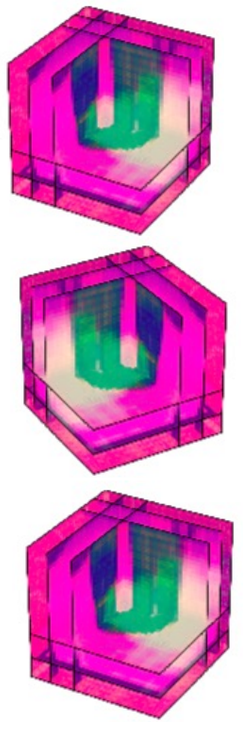
Relative Rotation R

cross convolution

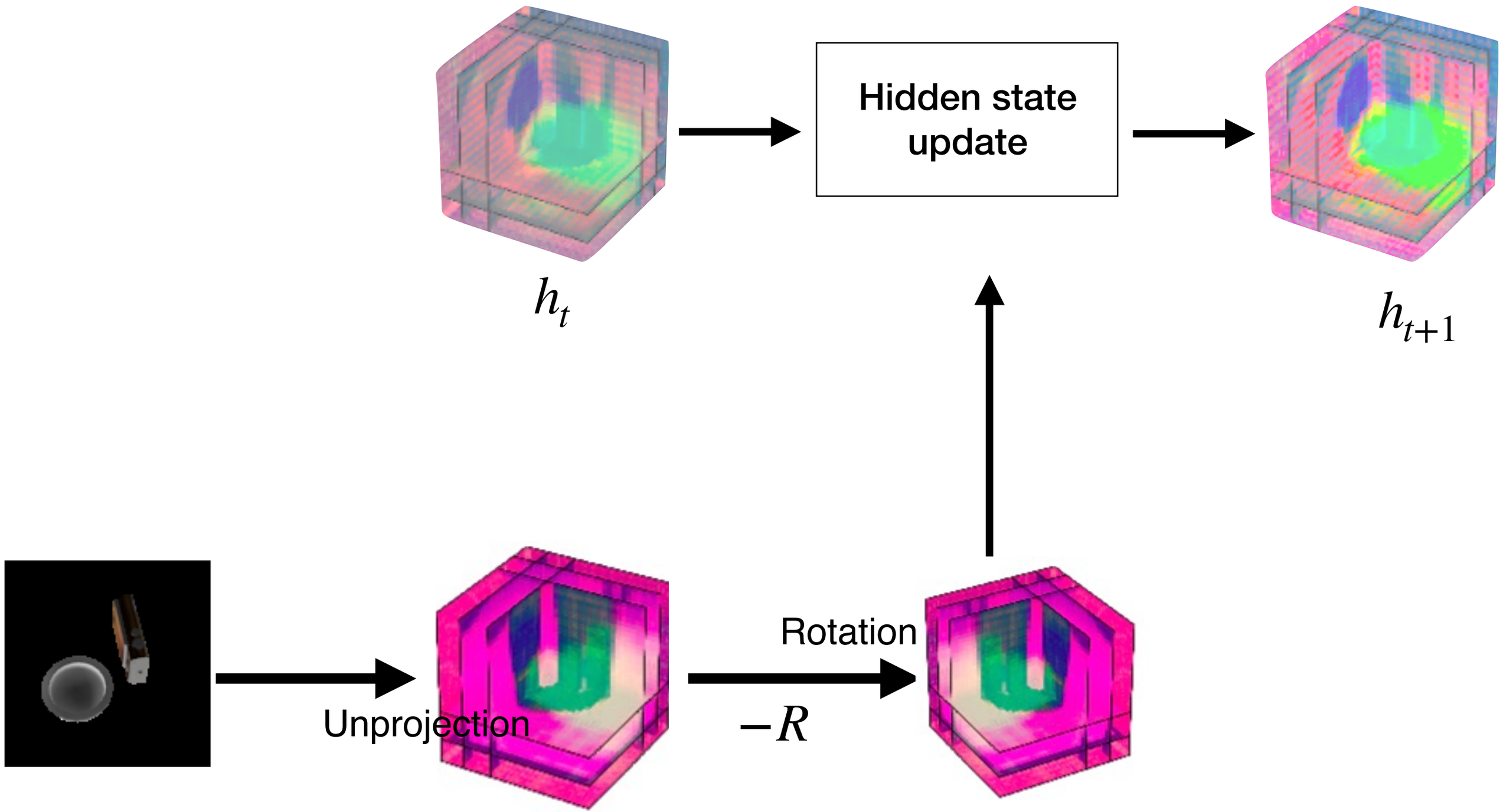
Unprojection



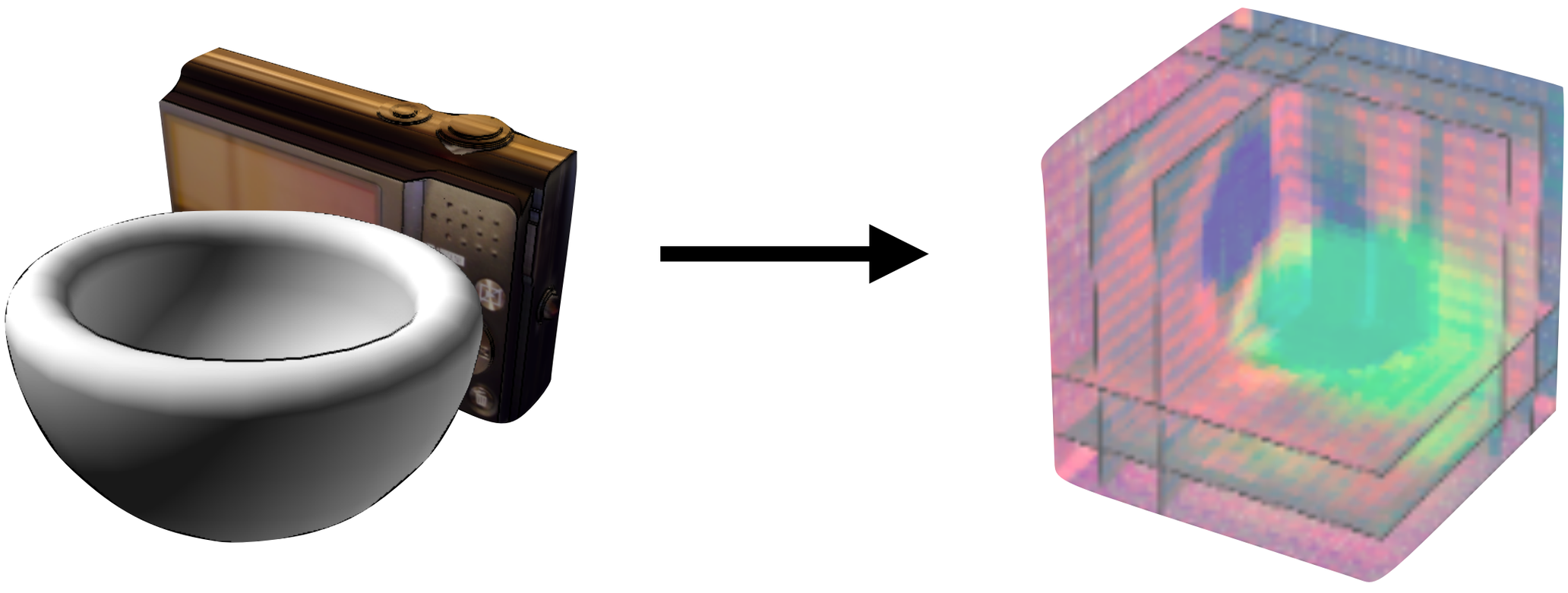
Rotation



Egomotion-stabilized memory update

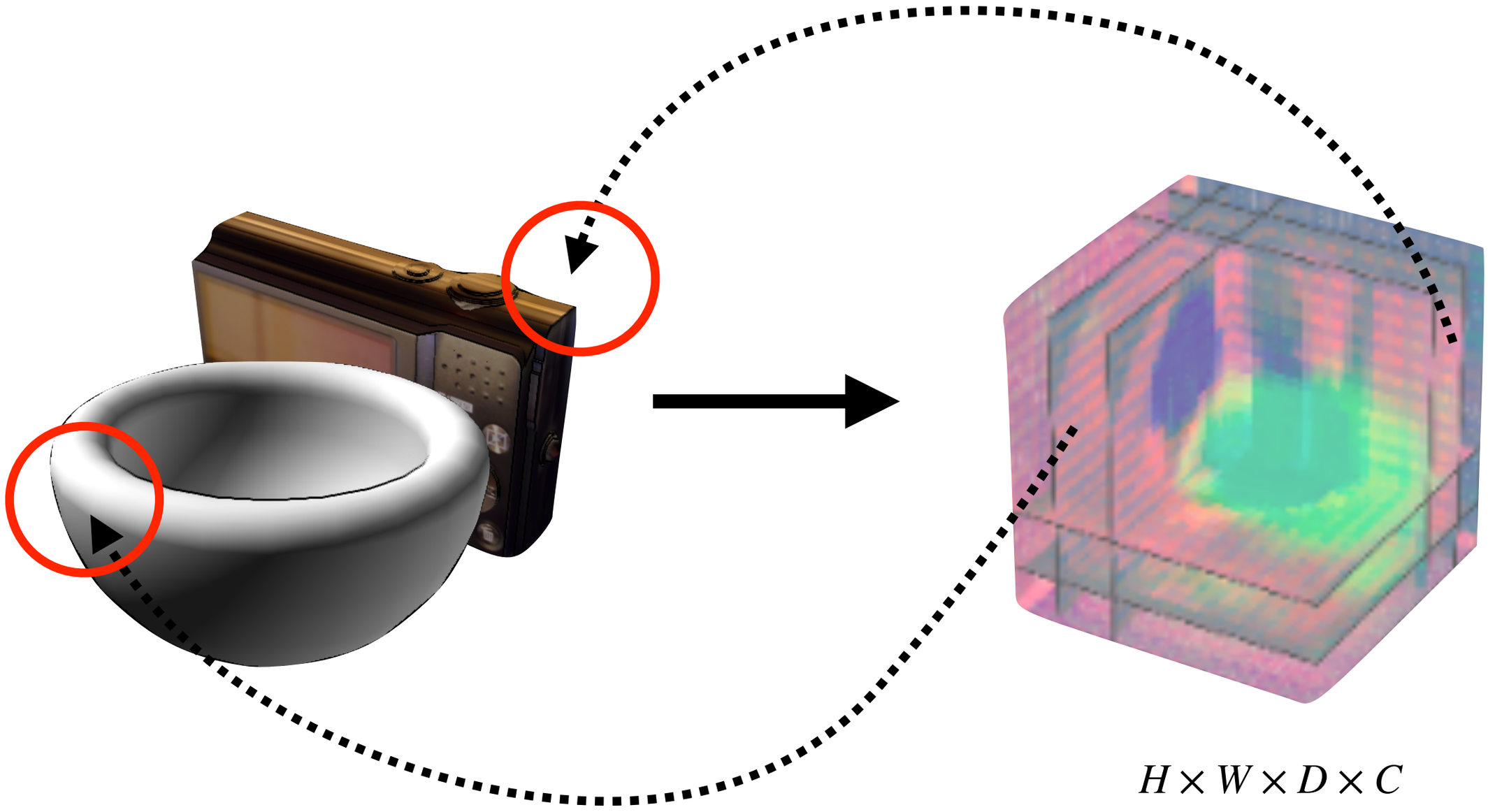


Geometry-Aware Recurrent Networks (GRNNs)

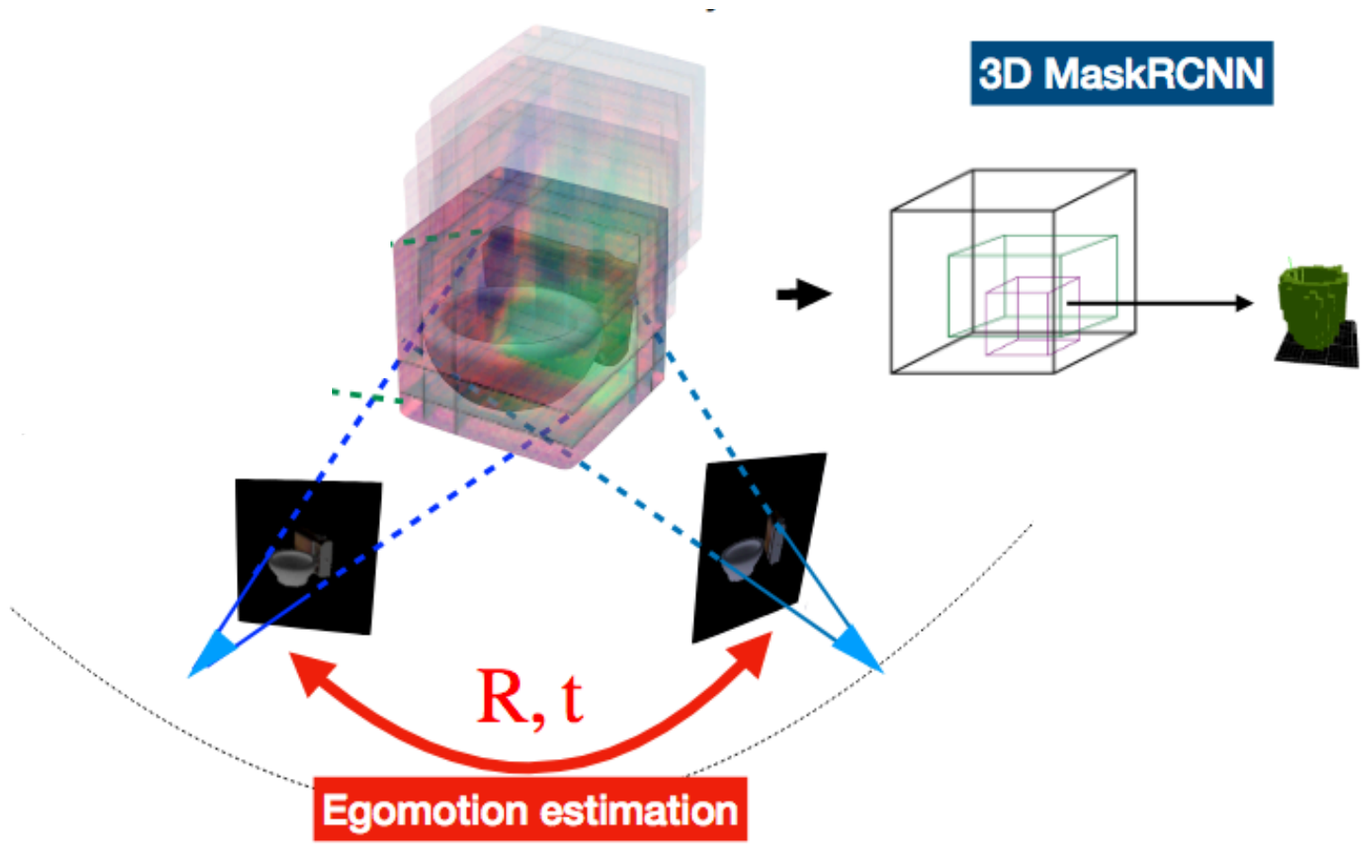


$H \times W \times D \times C$

Geometry-Aware Recurrent Networks (GRNNs)

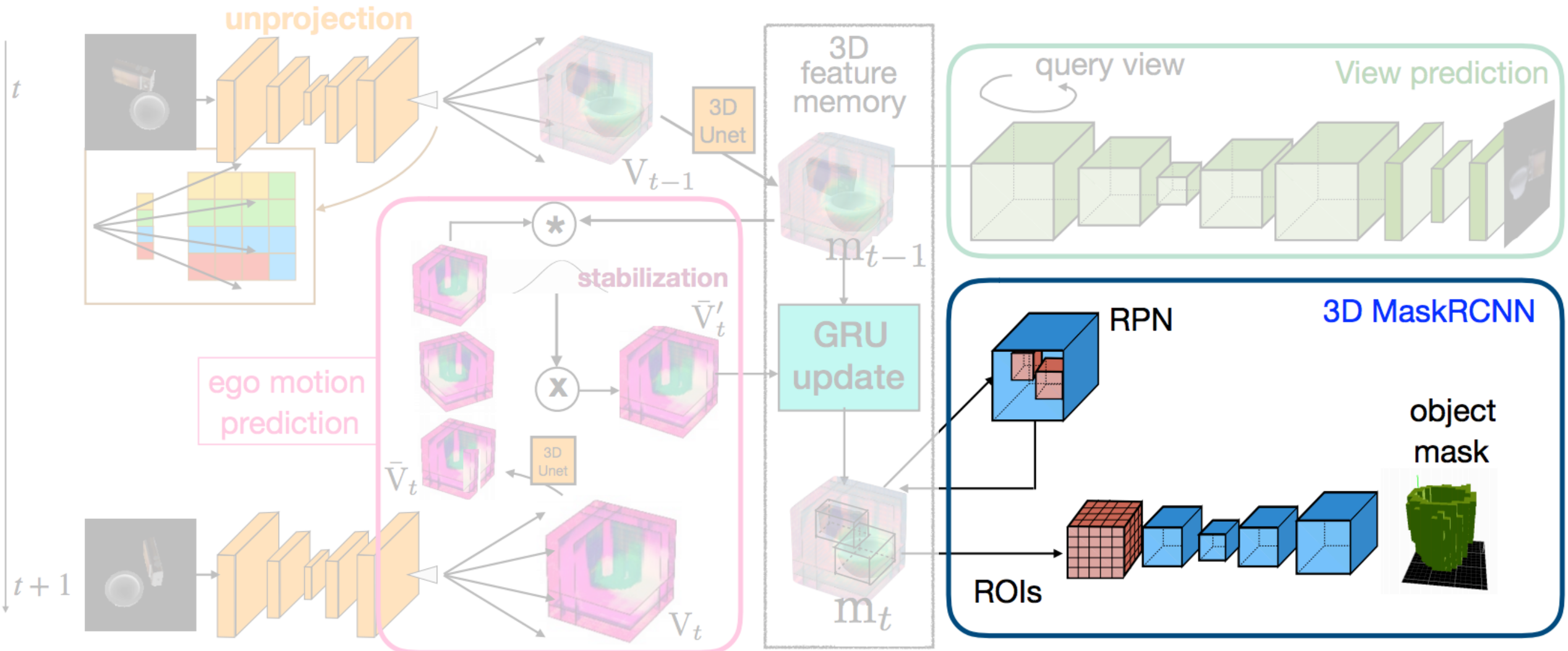


Training GRNNs



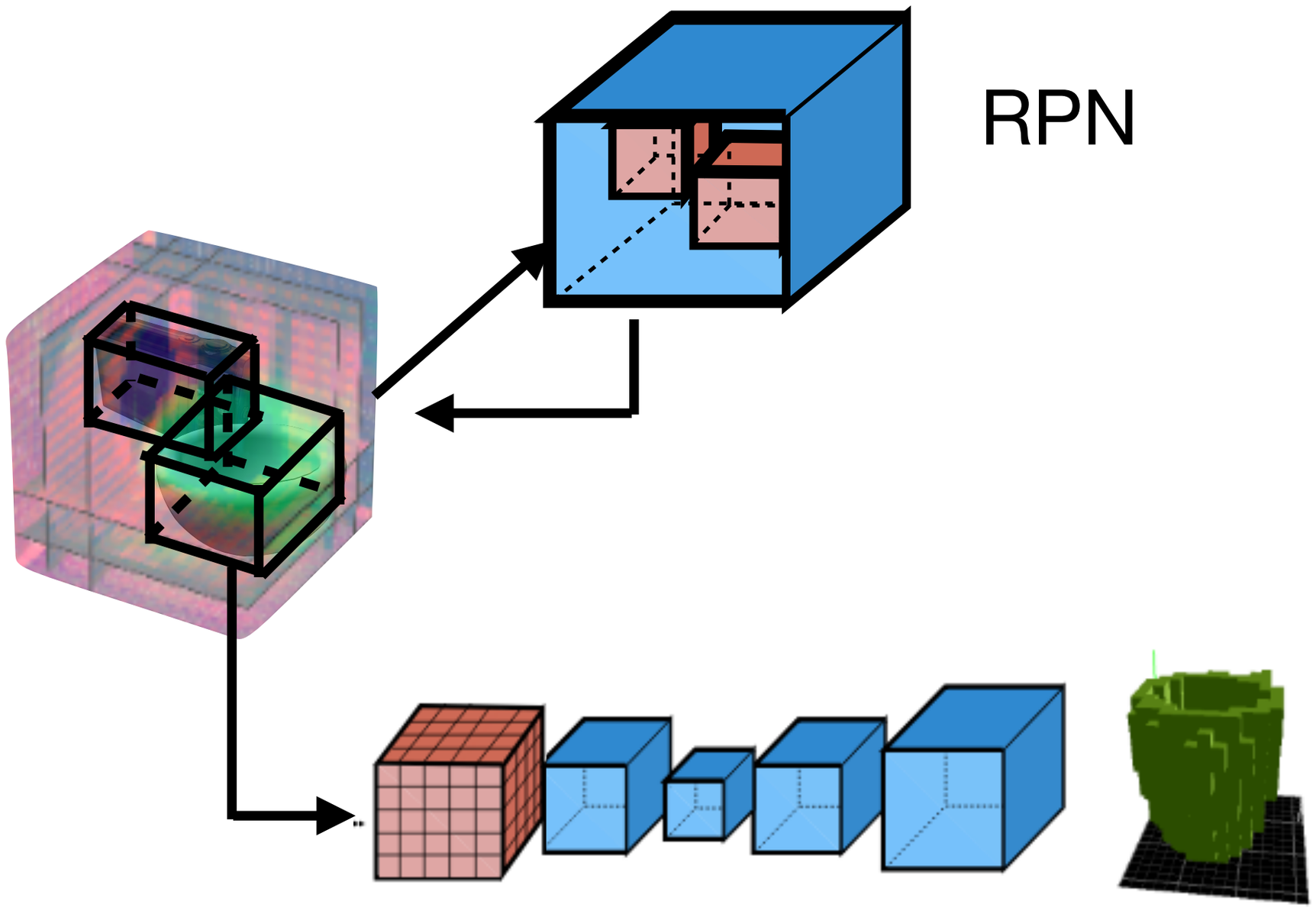
1. Supervised for 3D object detection
2. Self-supervised for view prediction

Architecture



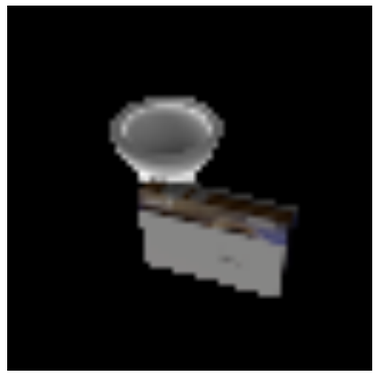
3D Object Detection

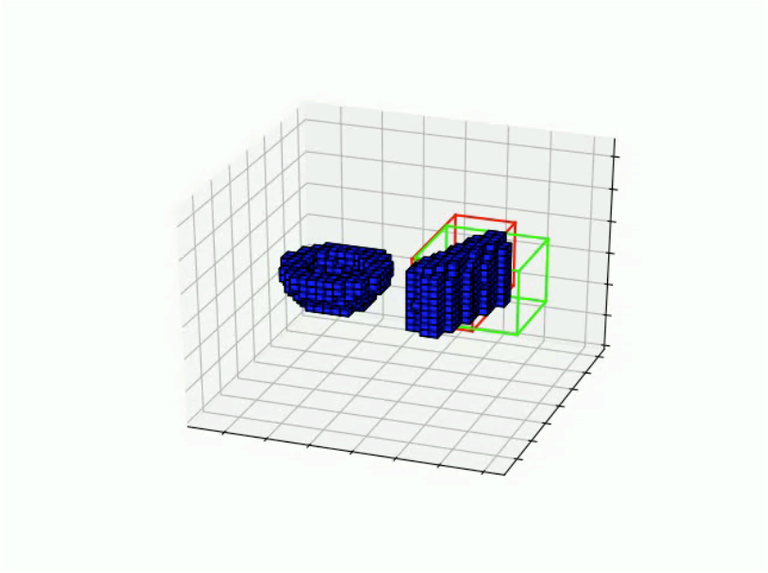
Input: the 3D latent feature map
Output: 3D boxes and segmentations for the objects



Results - 3D object detection

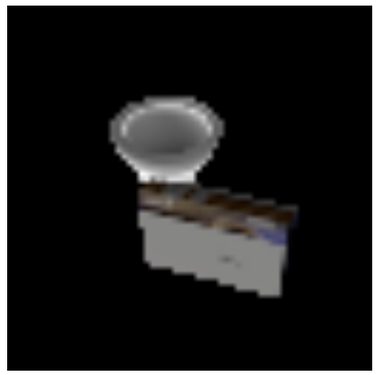
of input views

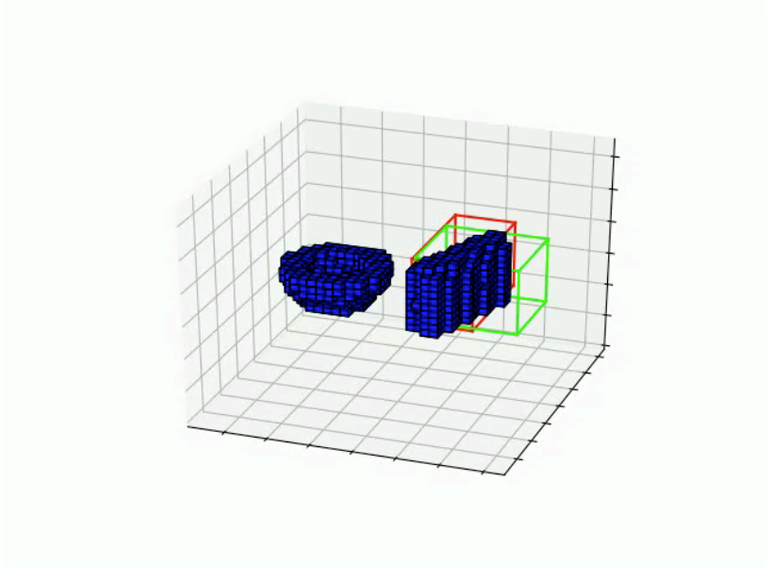


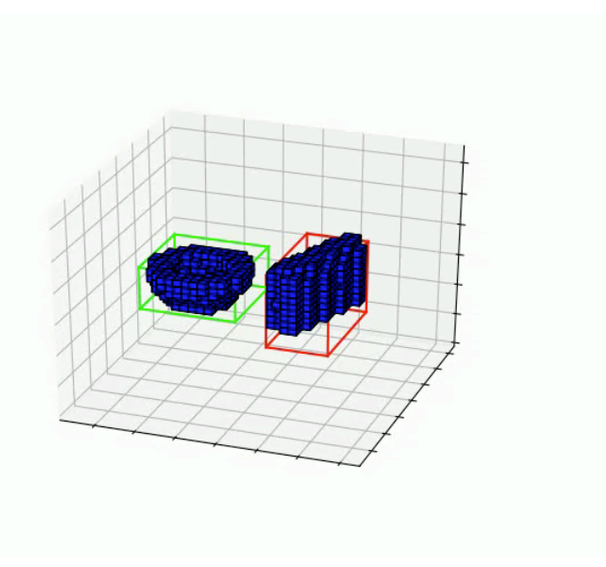


Results - 3D object detection

of input views

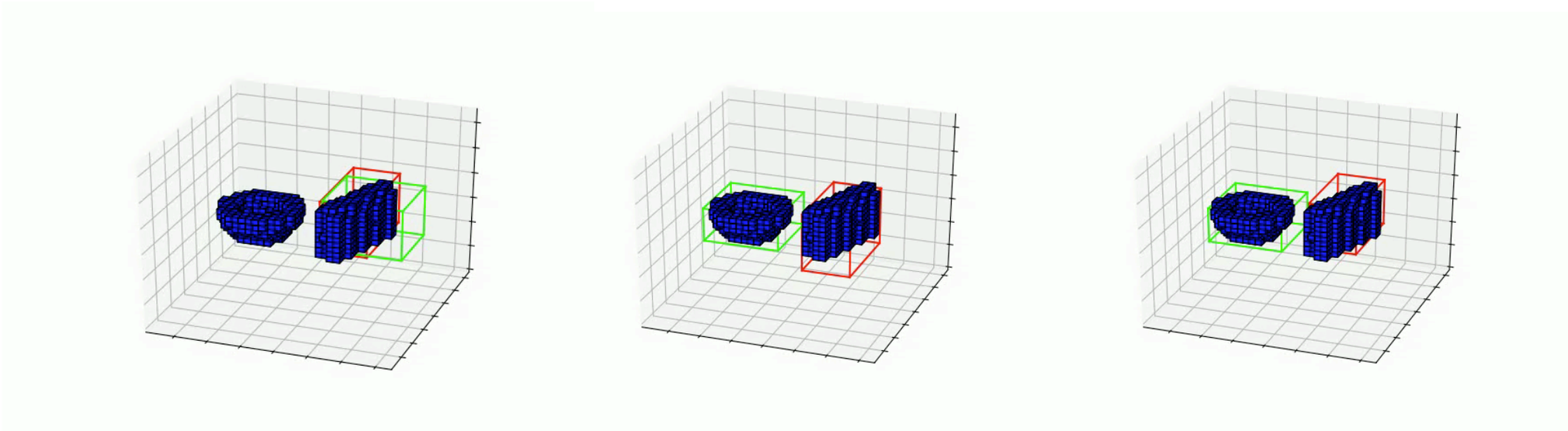




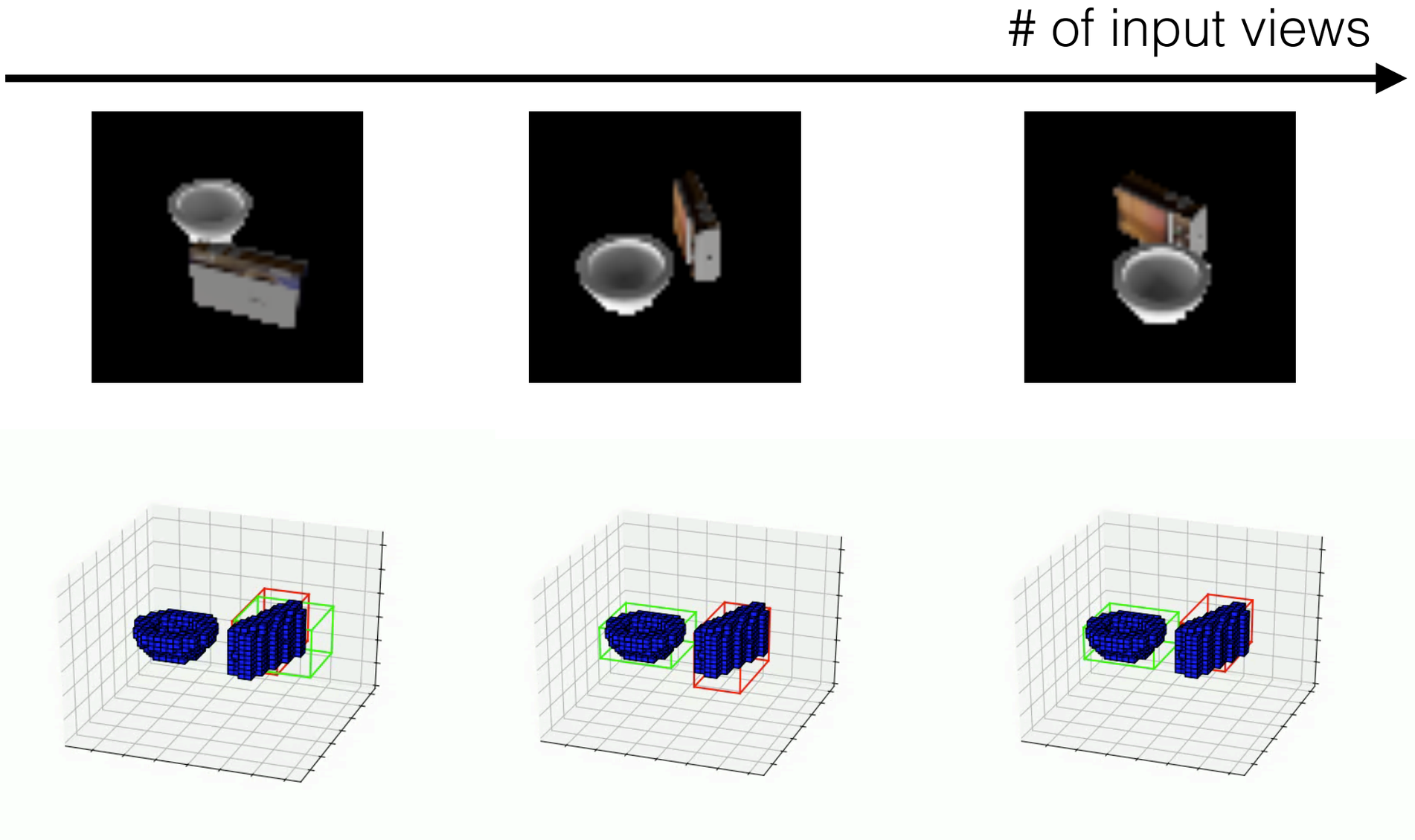


Results - 3D object detection

of input views

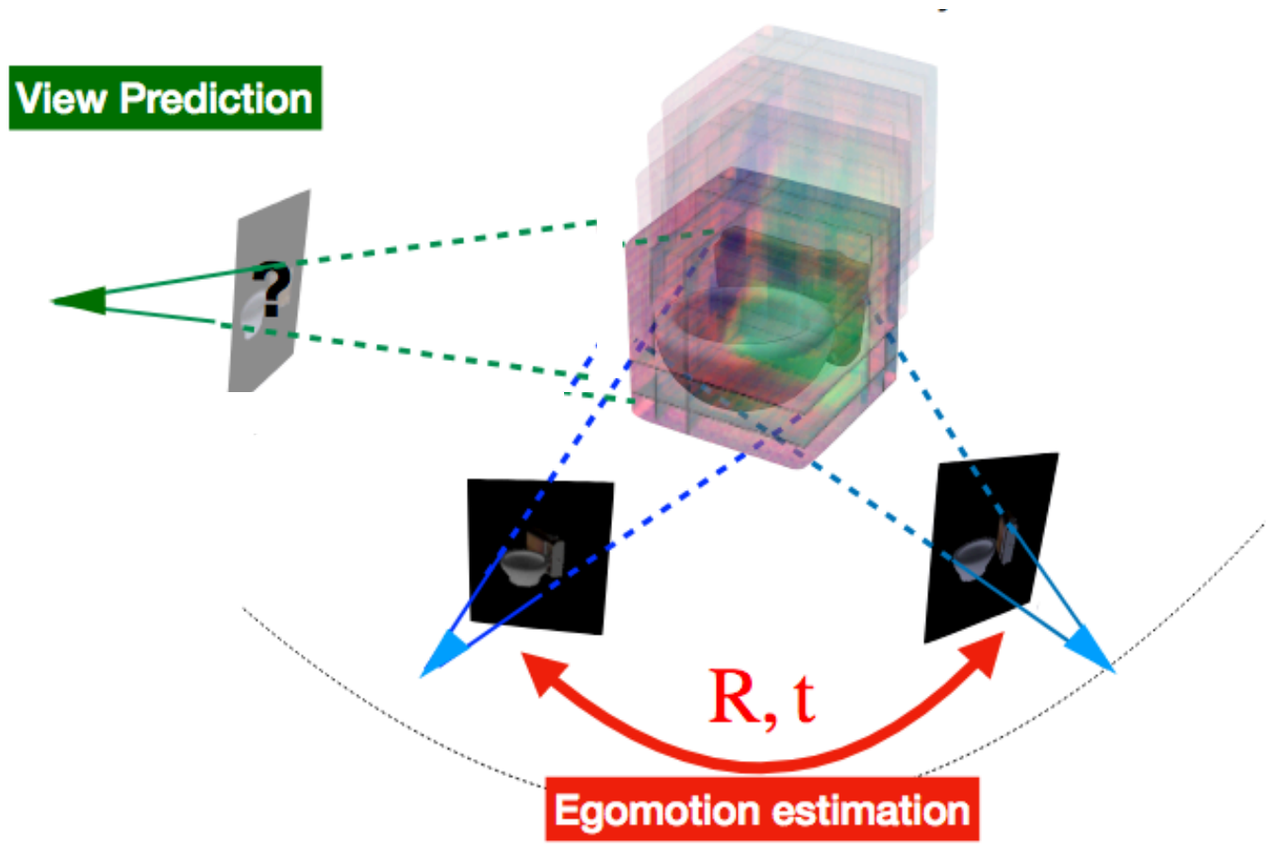


Object permanence emerges



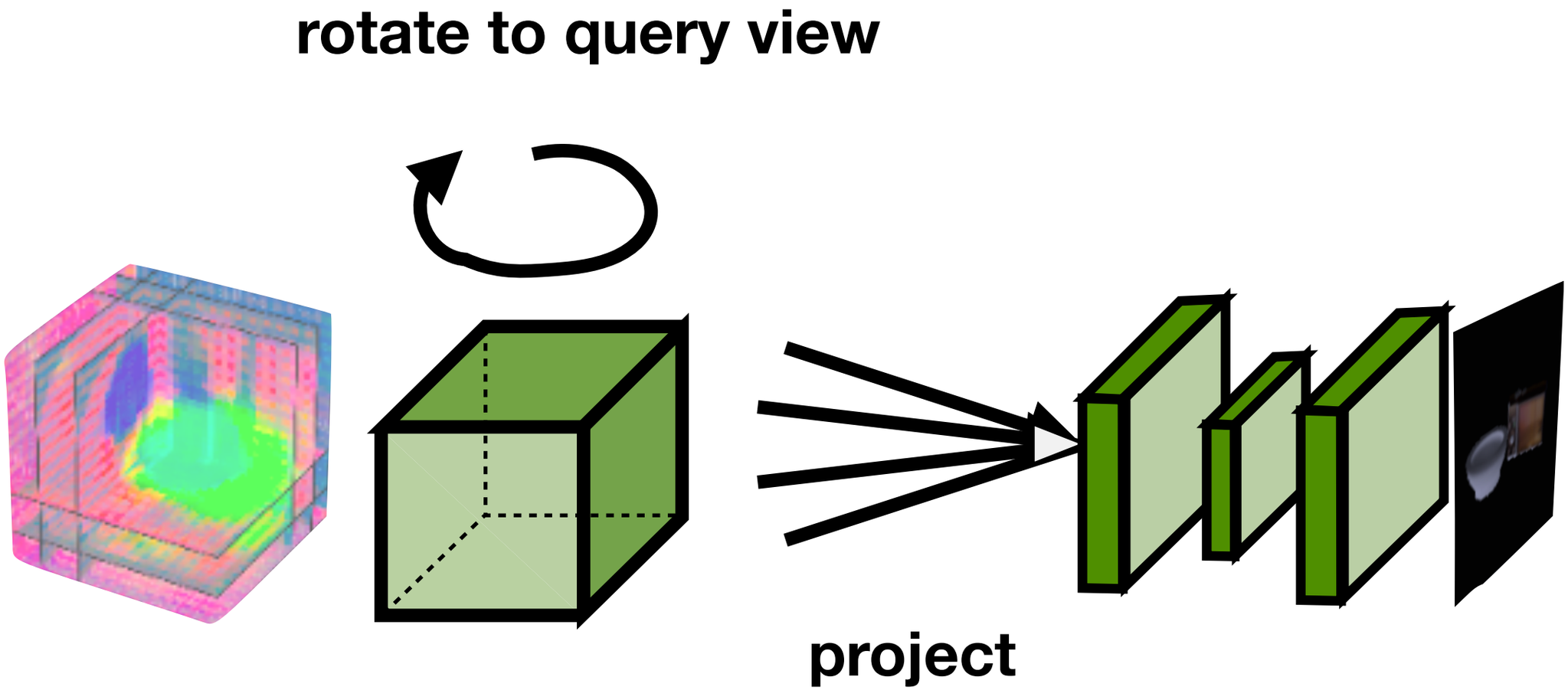
- Objects persist over time, objects have 3D extent

Training GRNNs



1. Supervised for 3D object detection
2. Self-supervised for view prediction

View prediction

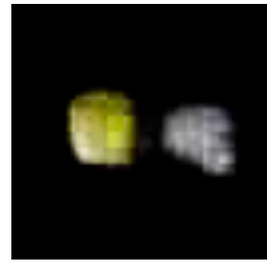


Results - view prediction

Input views

GRNNs

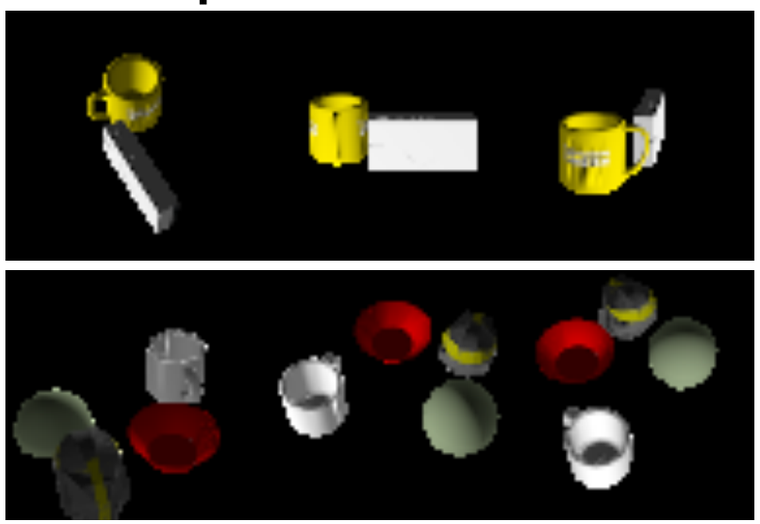
GQN [1]



1. Neural scene representation and rendering Deep

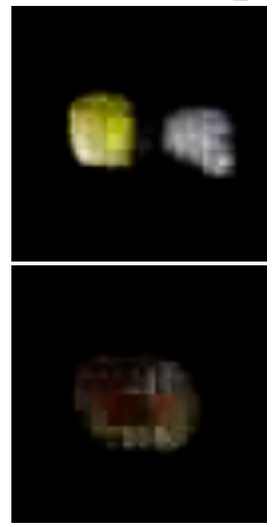
Results - view prediction

Input views



GRNNs

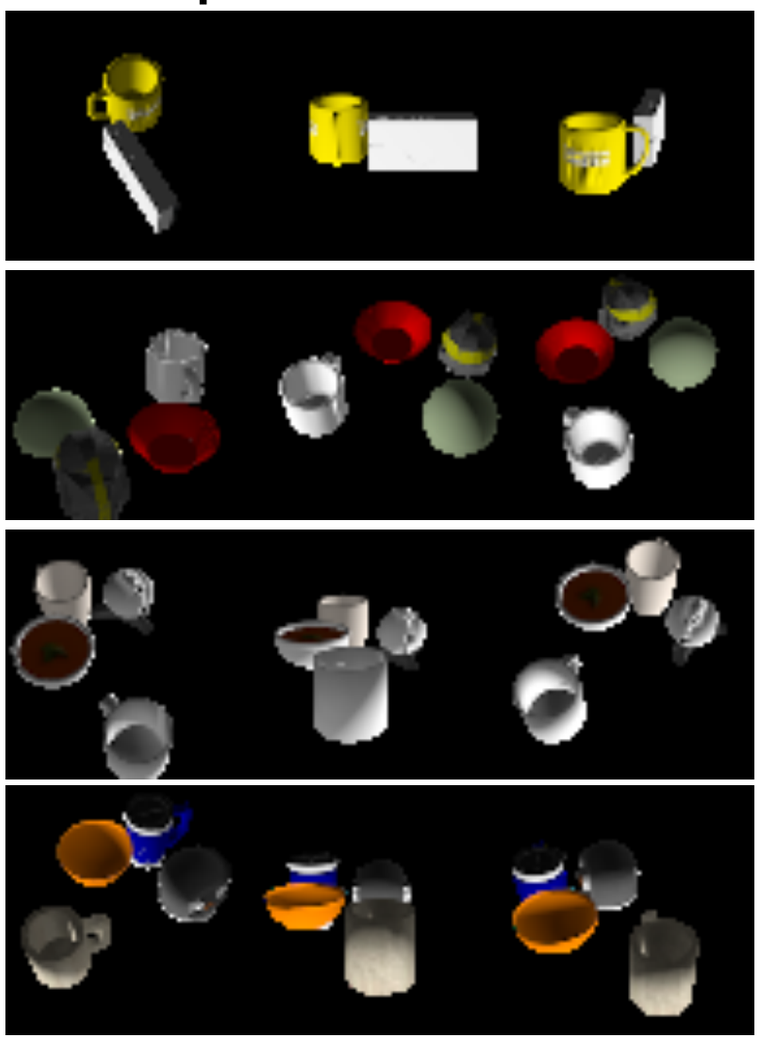
GQN [1]



1. Neural scene representation and rendering Dee

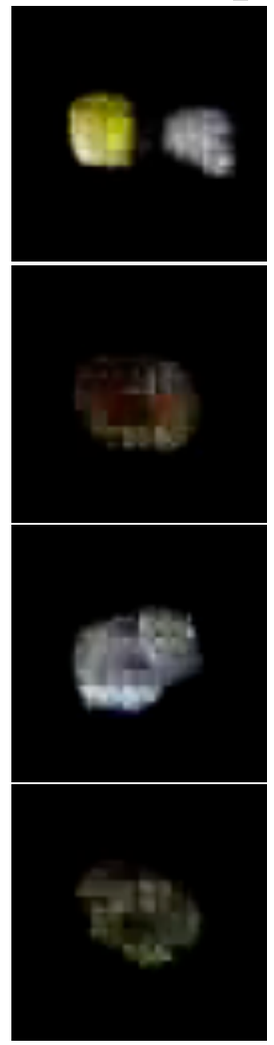
Results - view prediction

Input views



GRNNs

GQN [1]



1. Neural scene representation and rendering Dee

Results - view prediction

Geometry-aware RNN

GQN [1]

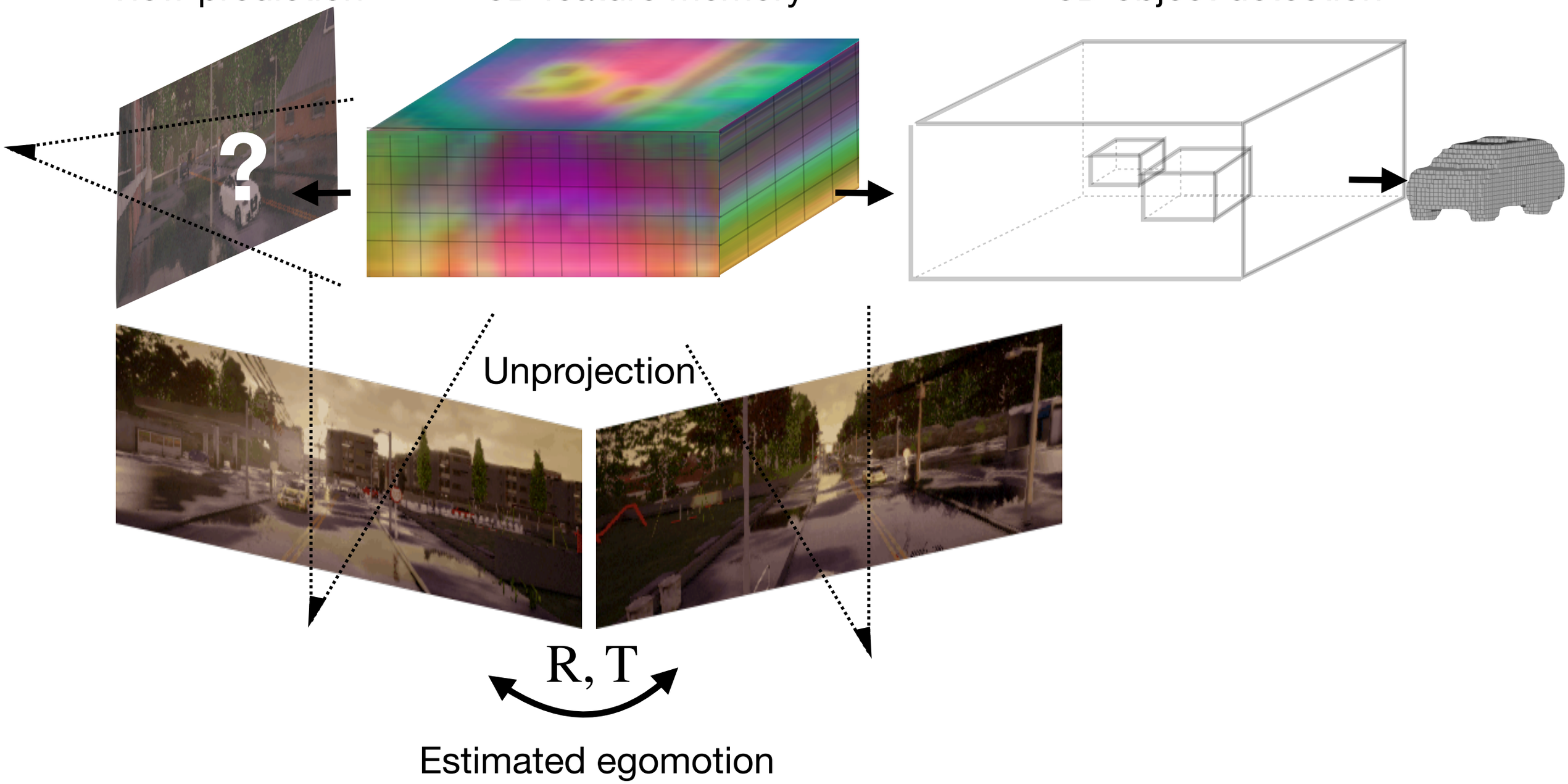
1. *Neural scene representation and rendering Deep*

Embodied visual recognition

- Can view prediction work beyond the toy simulation worlds we have just showed?
- Can view prediction learn features useful for object detection?

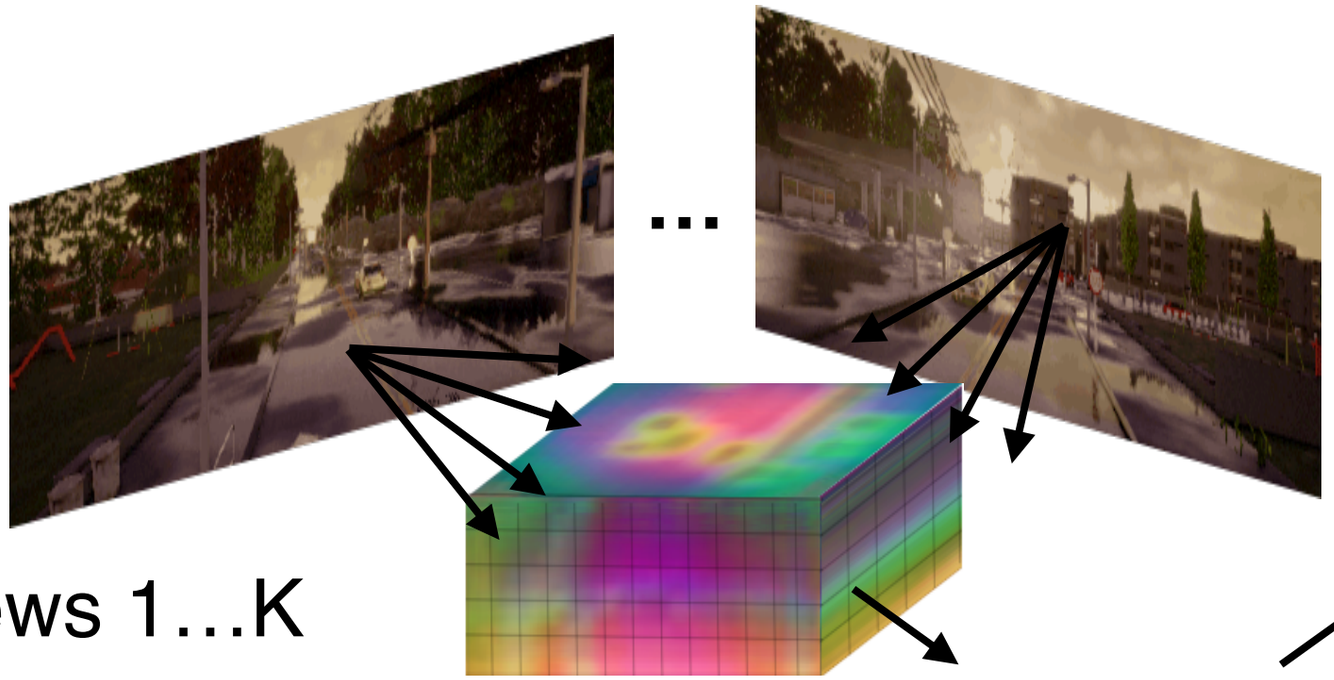
Yes, with a change in the loss function…

GRNNs in CARLA



View-contrastive prediction

Adam Harley



Embodied View-Contrastive 3D Feature Learn

View-contrastive prediction

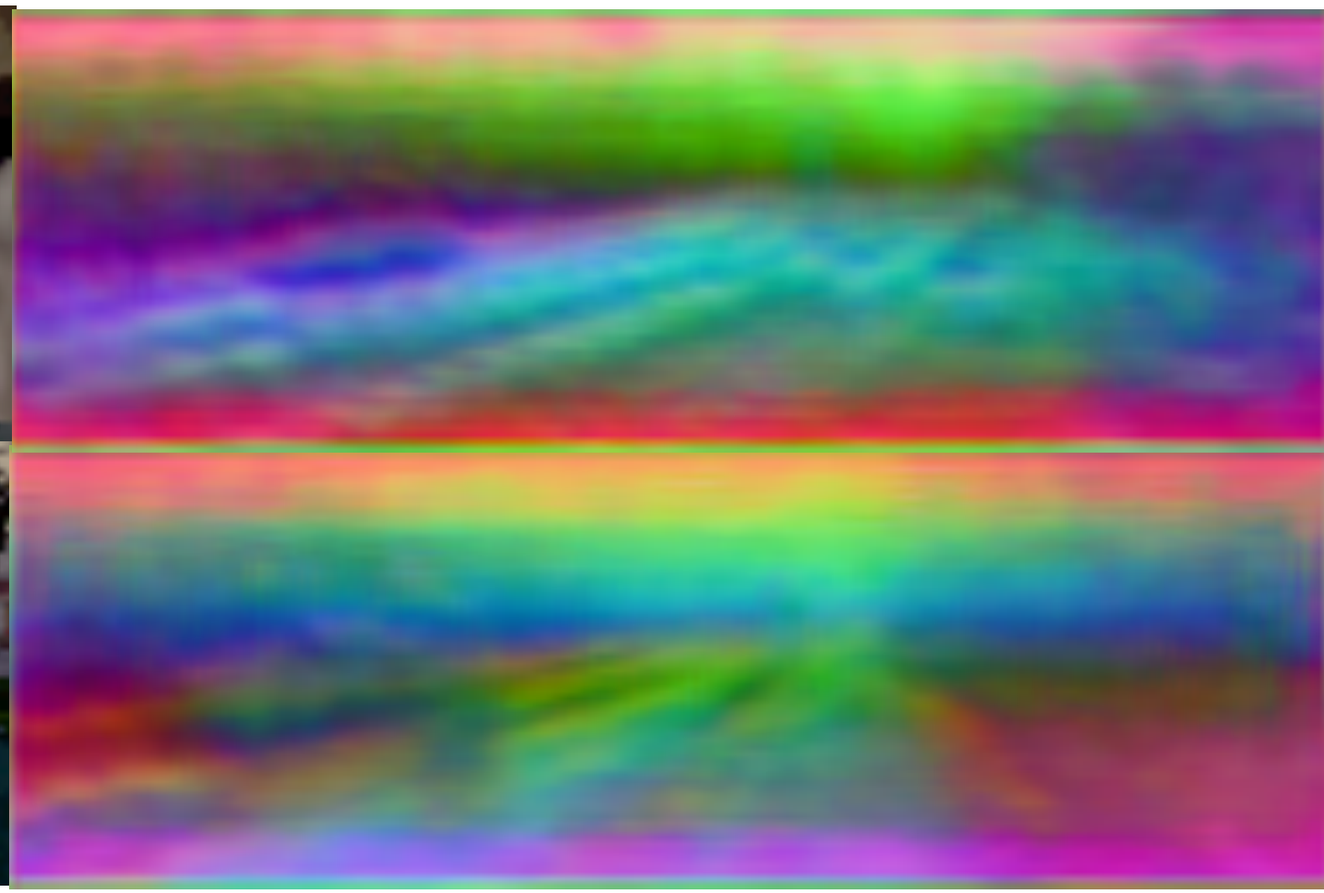
Target view

RGB estimates

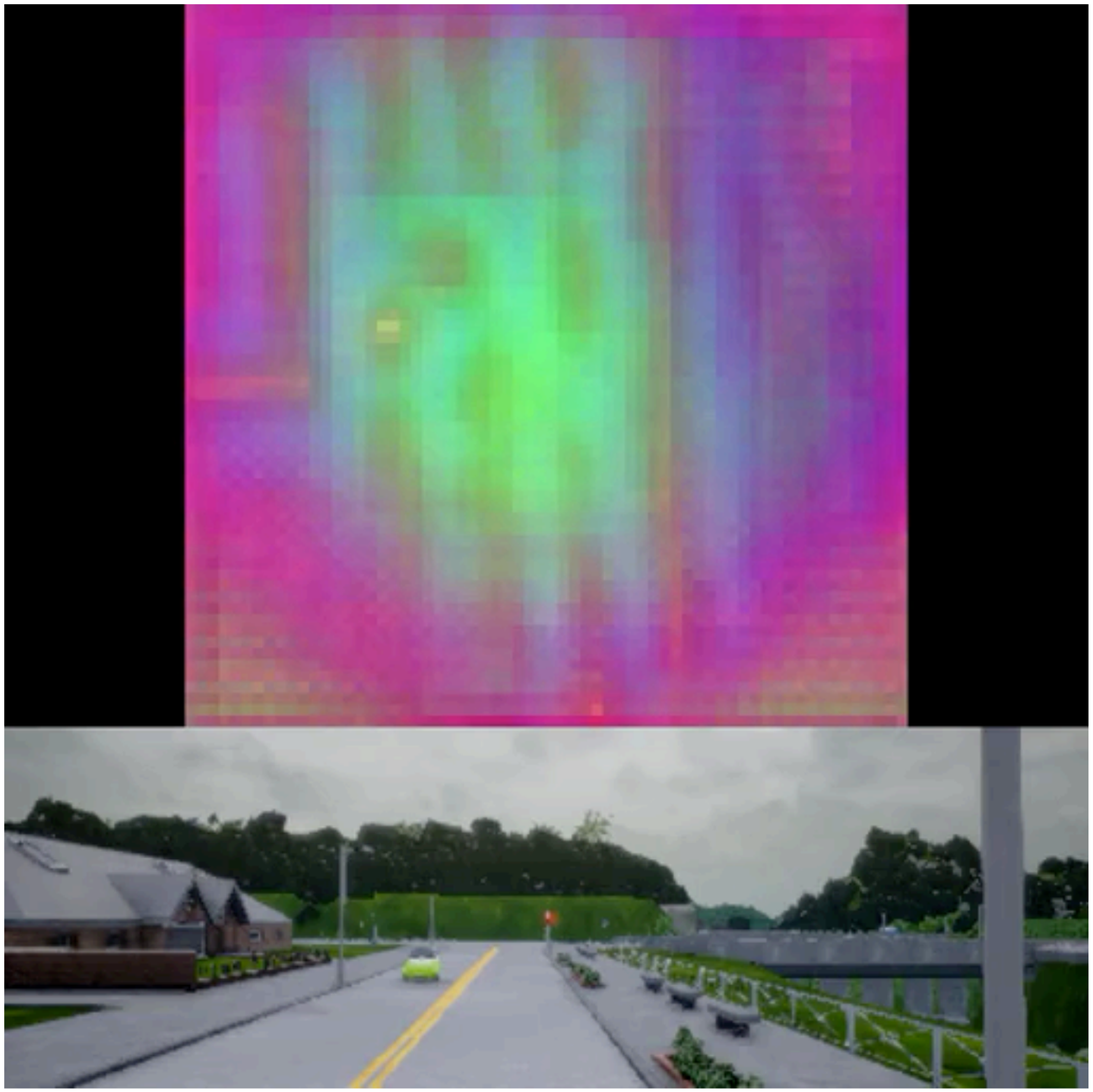
View-contrastive prediction

Target view

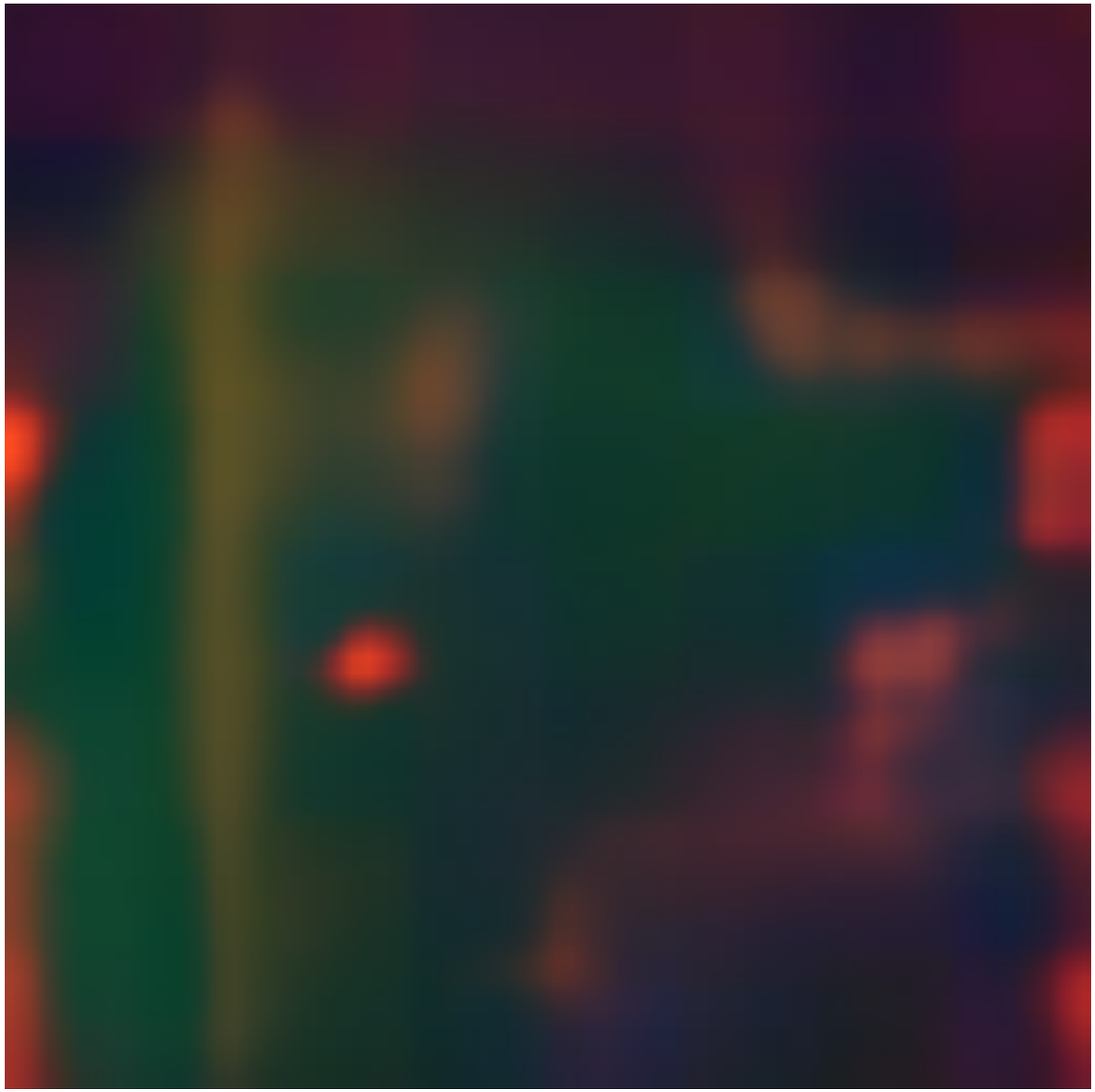
Embeddings



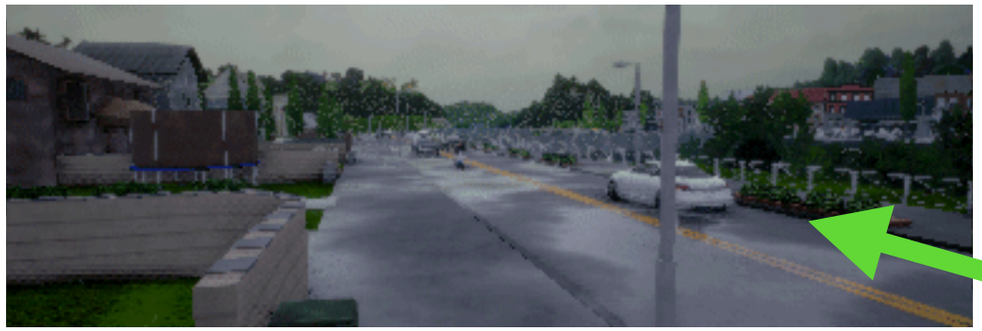
View-contrastive prediction

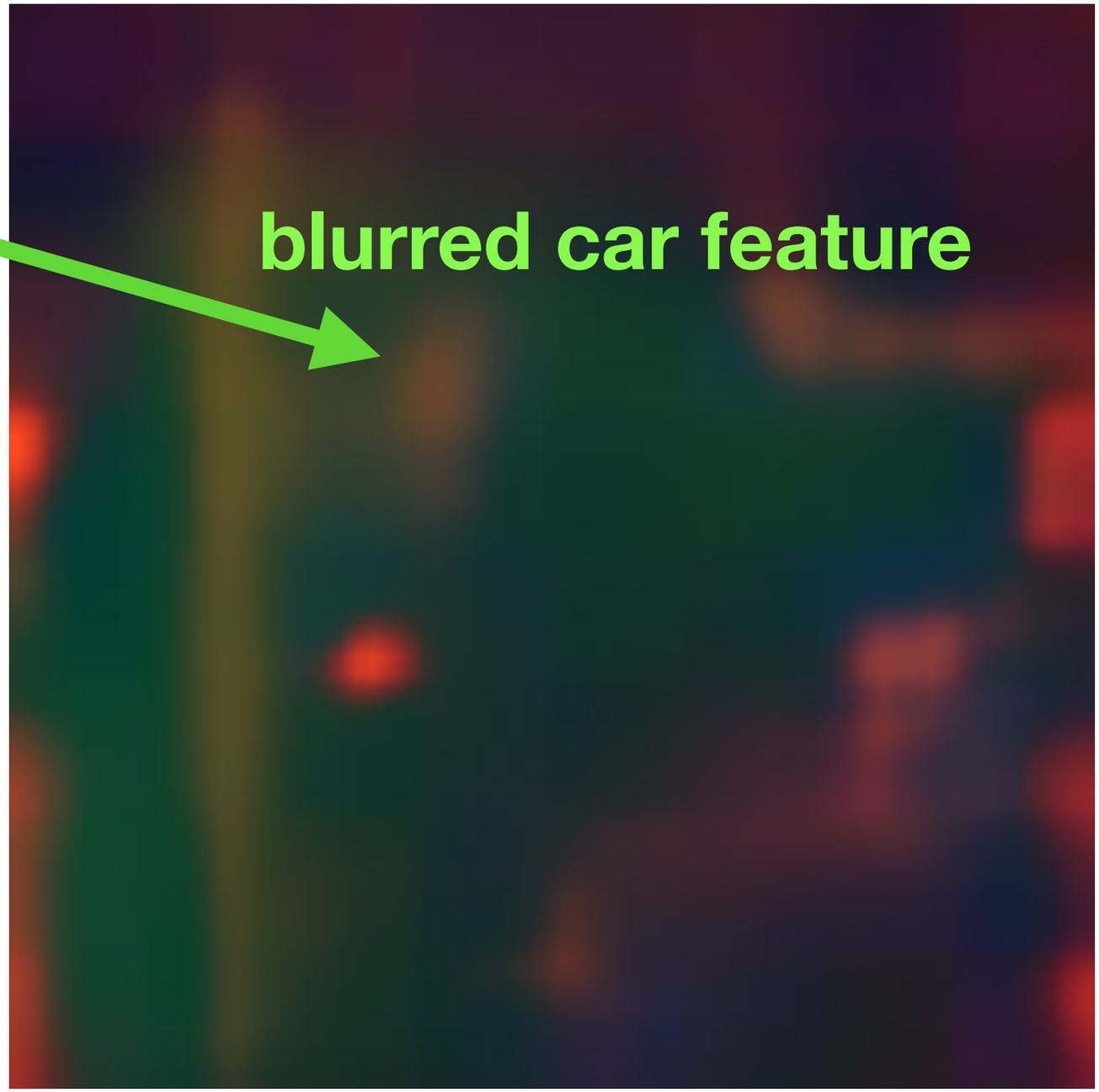


Imagination after 1 frame

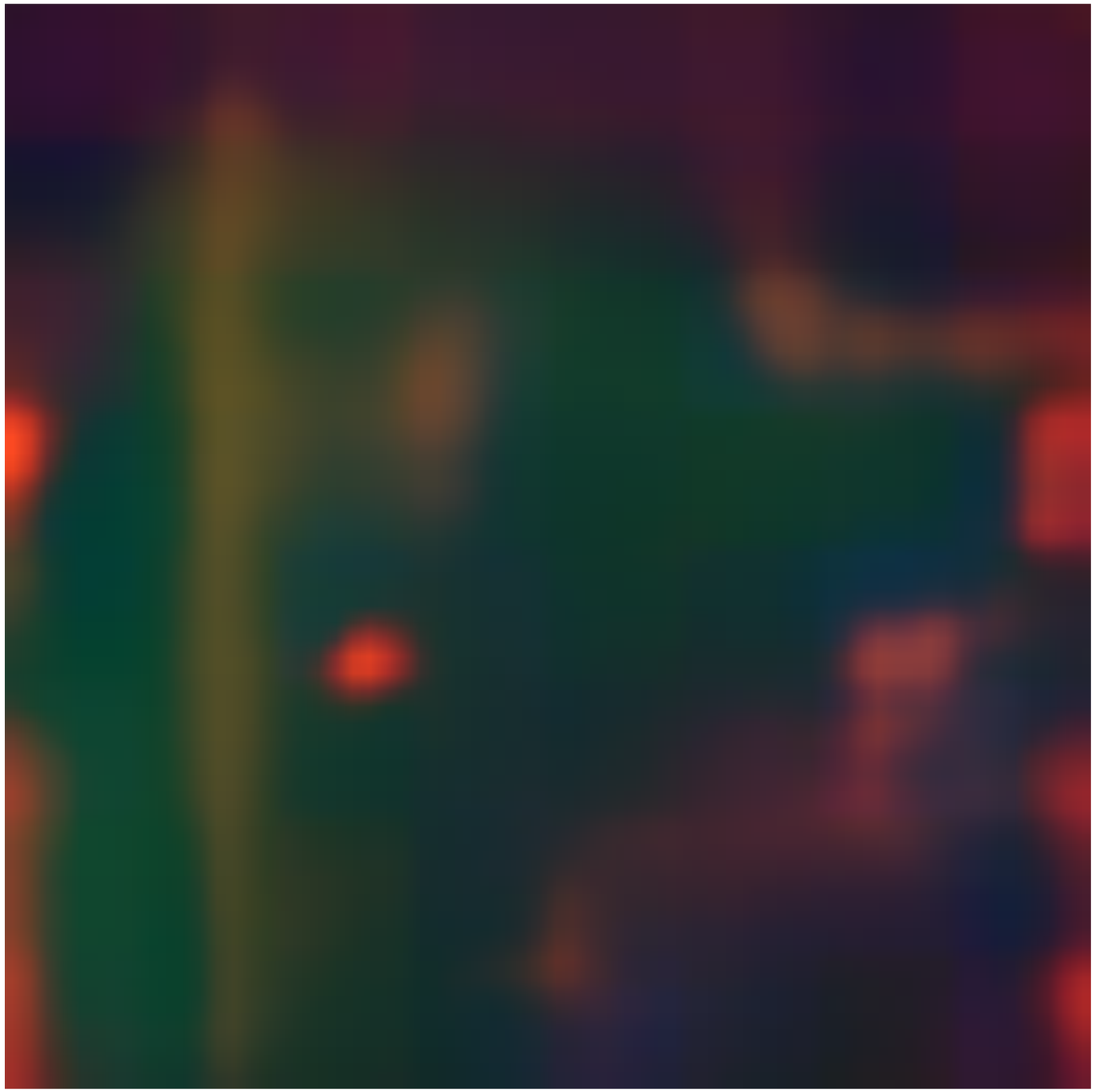


Imagination after 1 frame

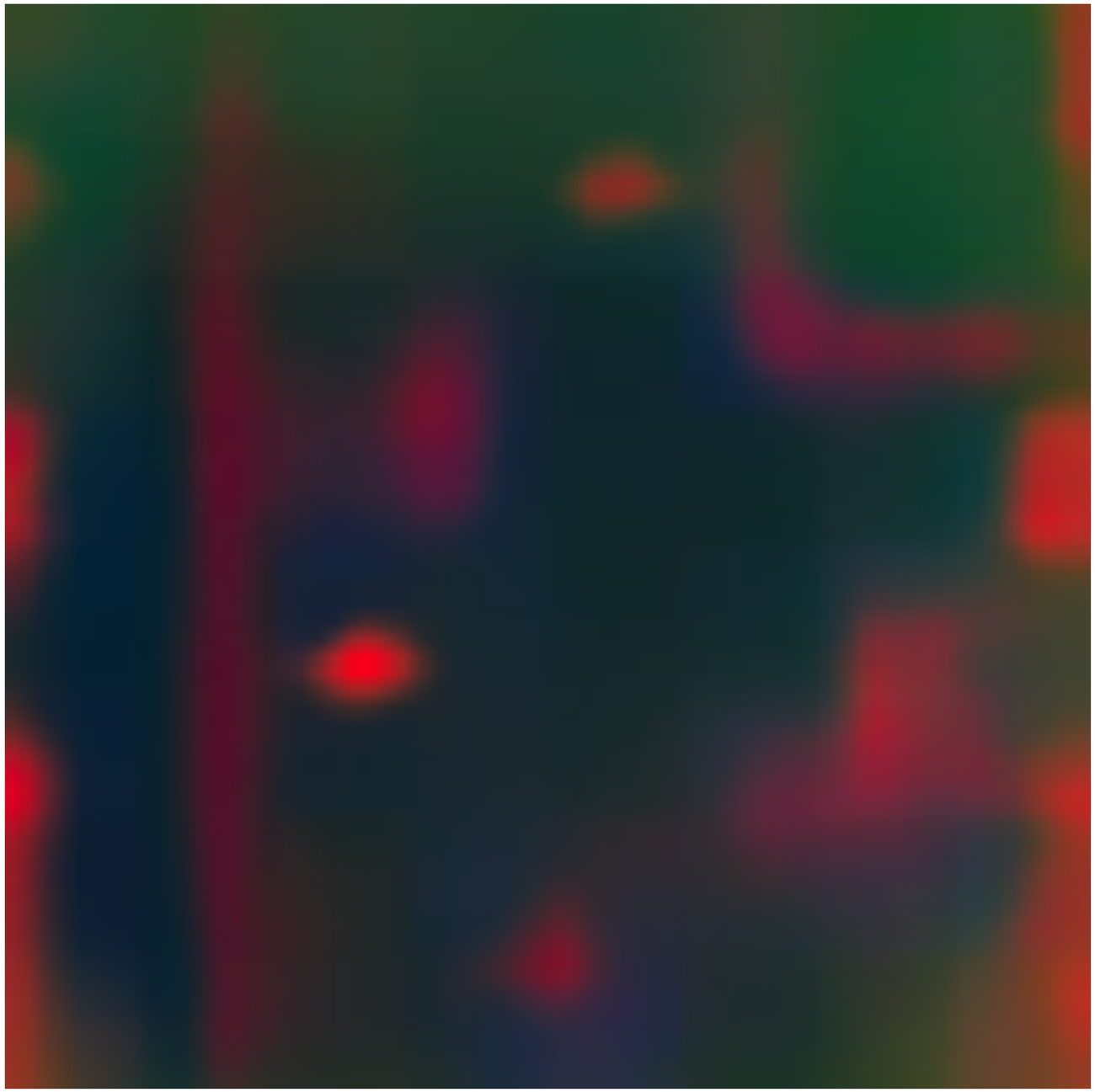




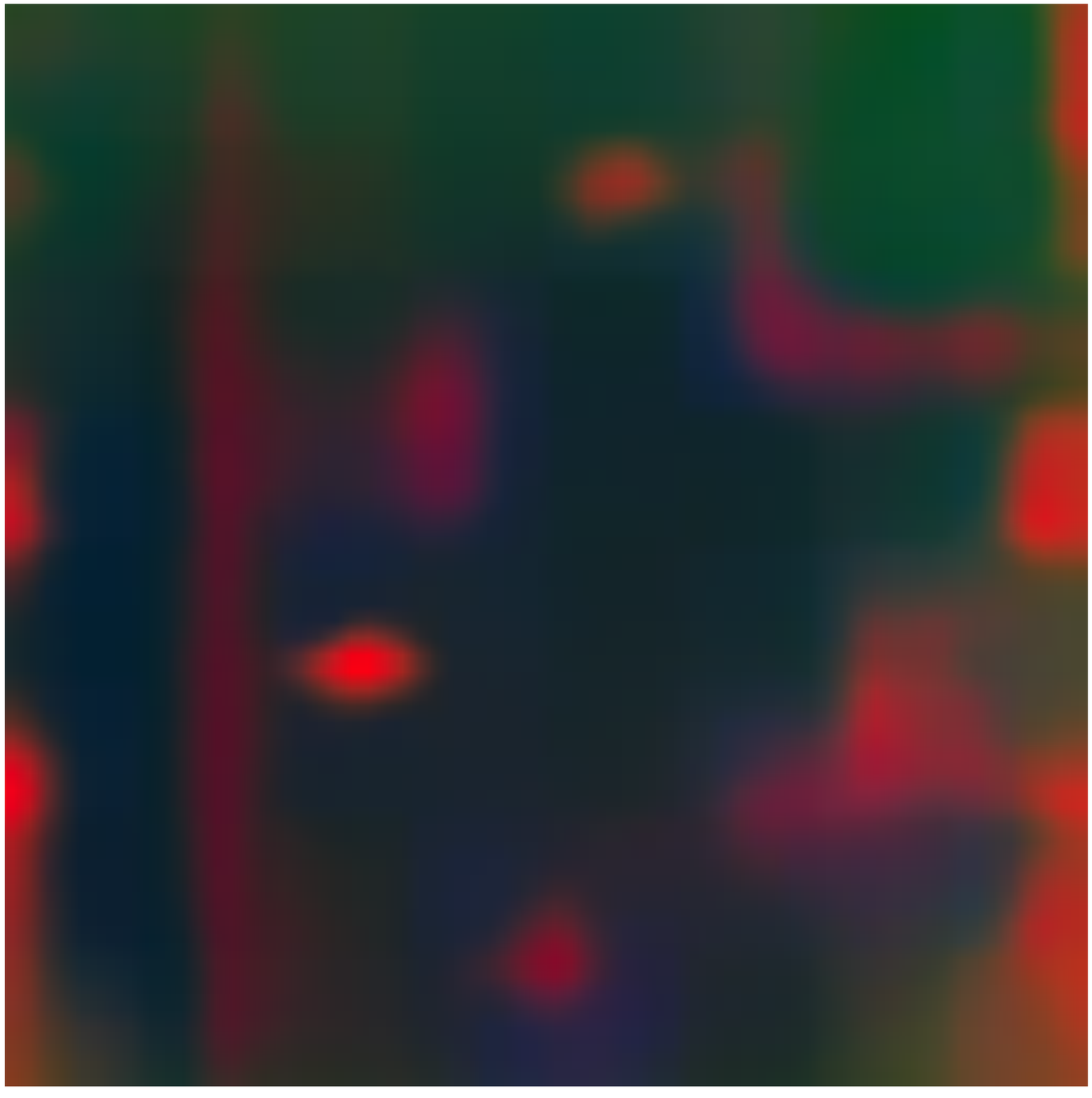
Imagination after 1 frame



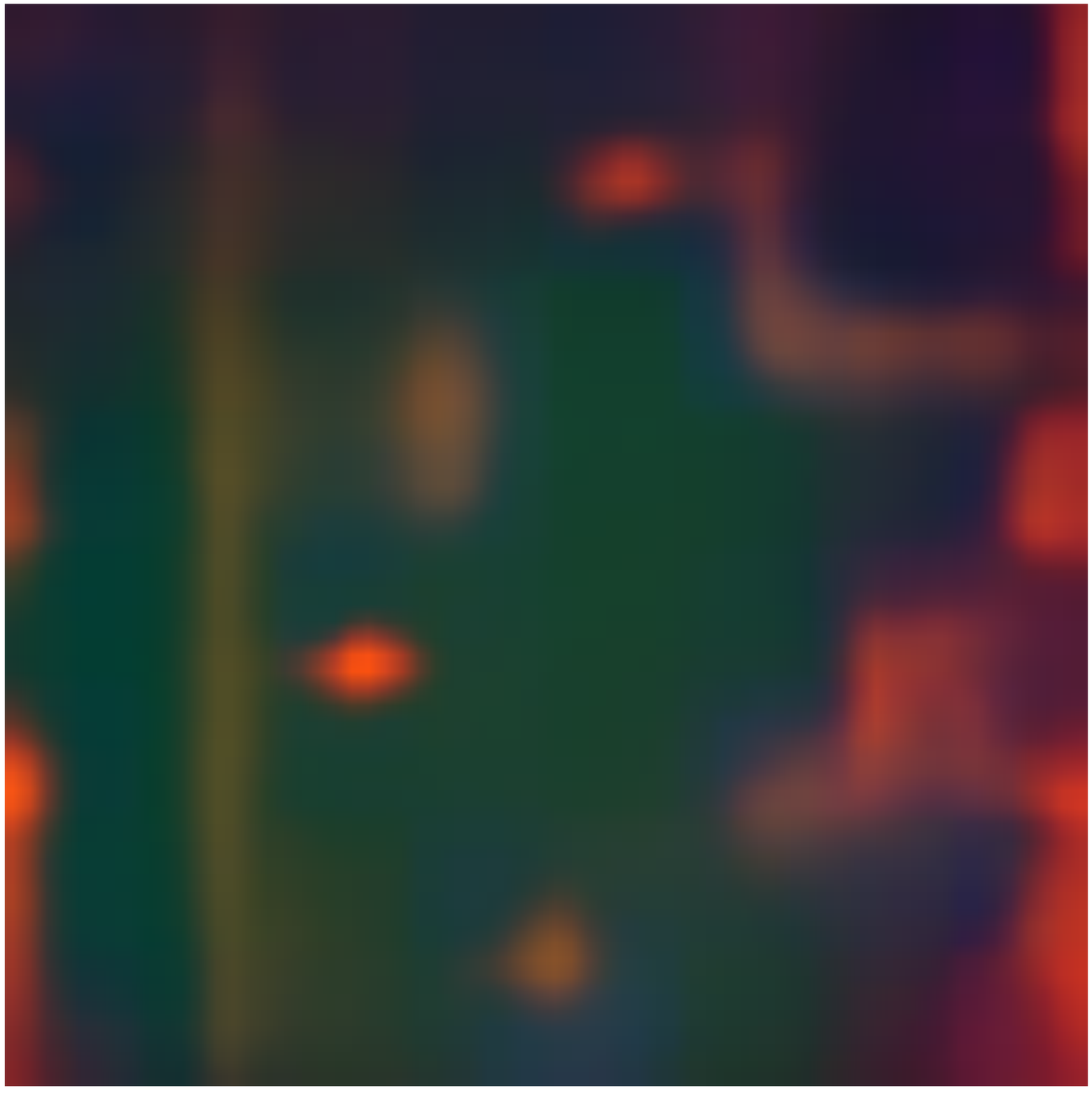
Imagination after 2 frames



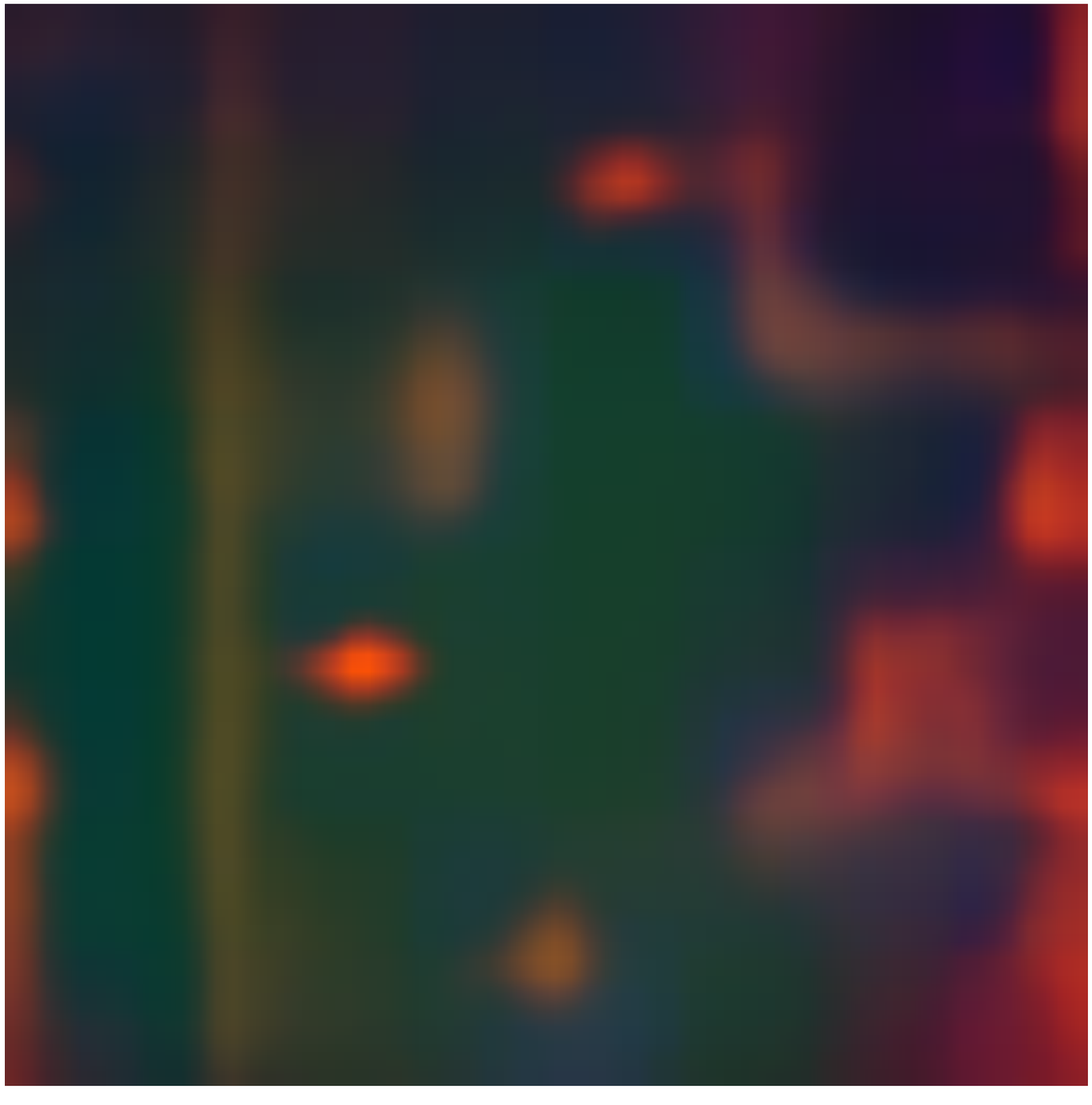
Imagination after 3 frames



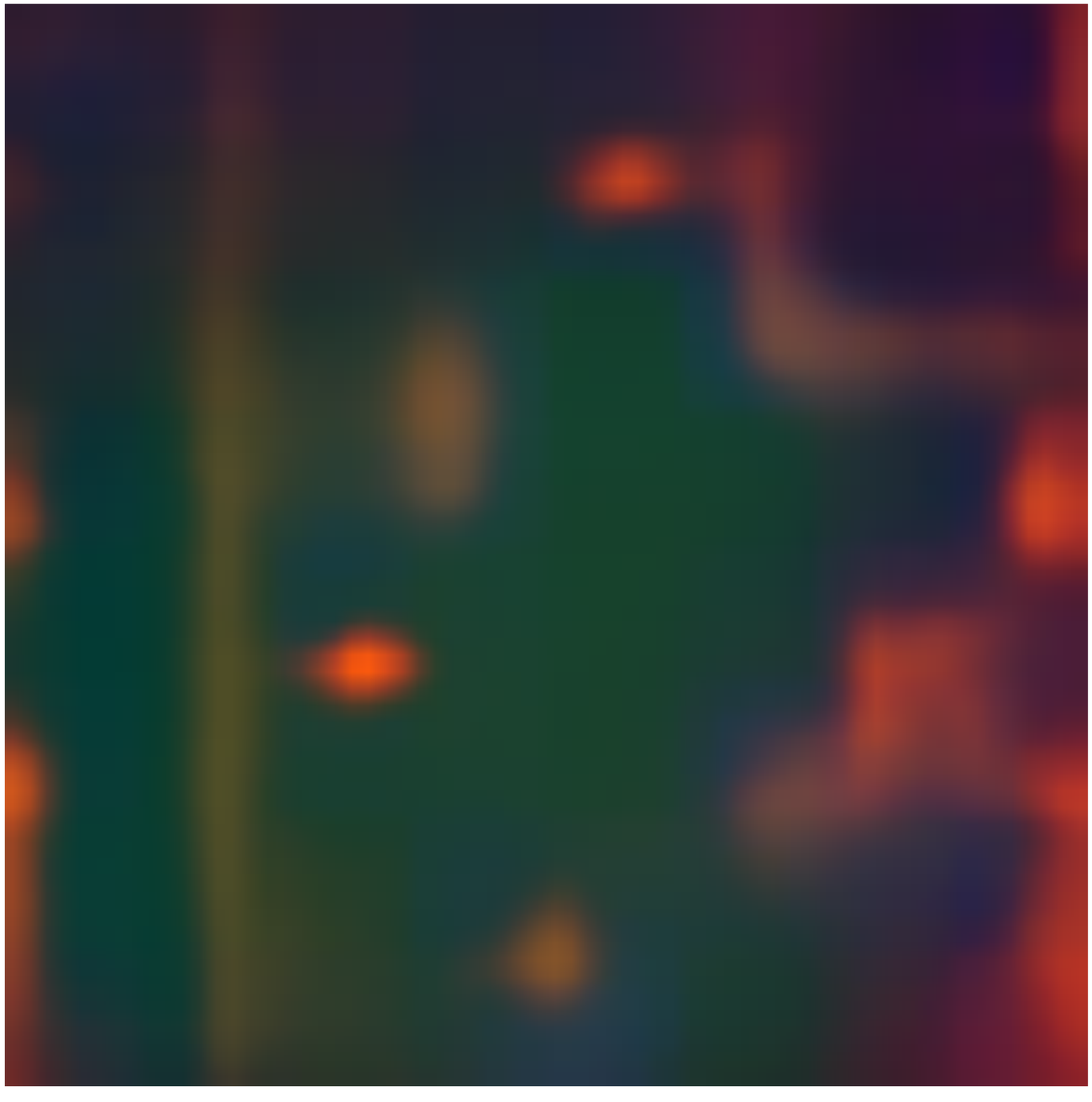
Imagination after 4 frames



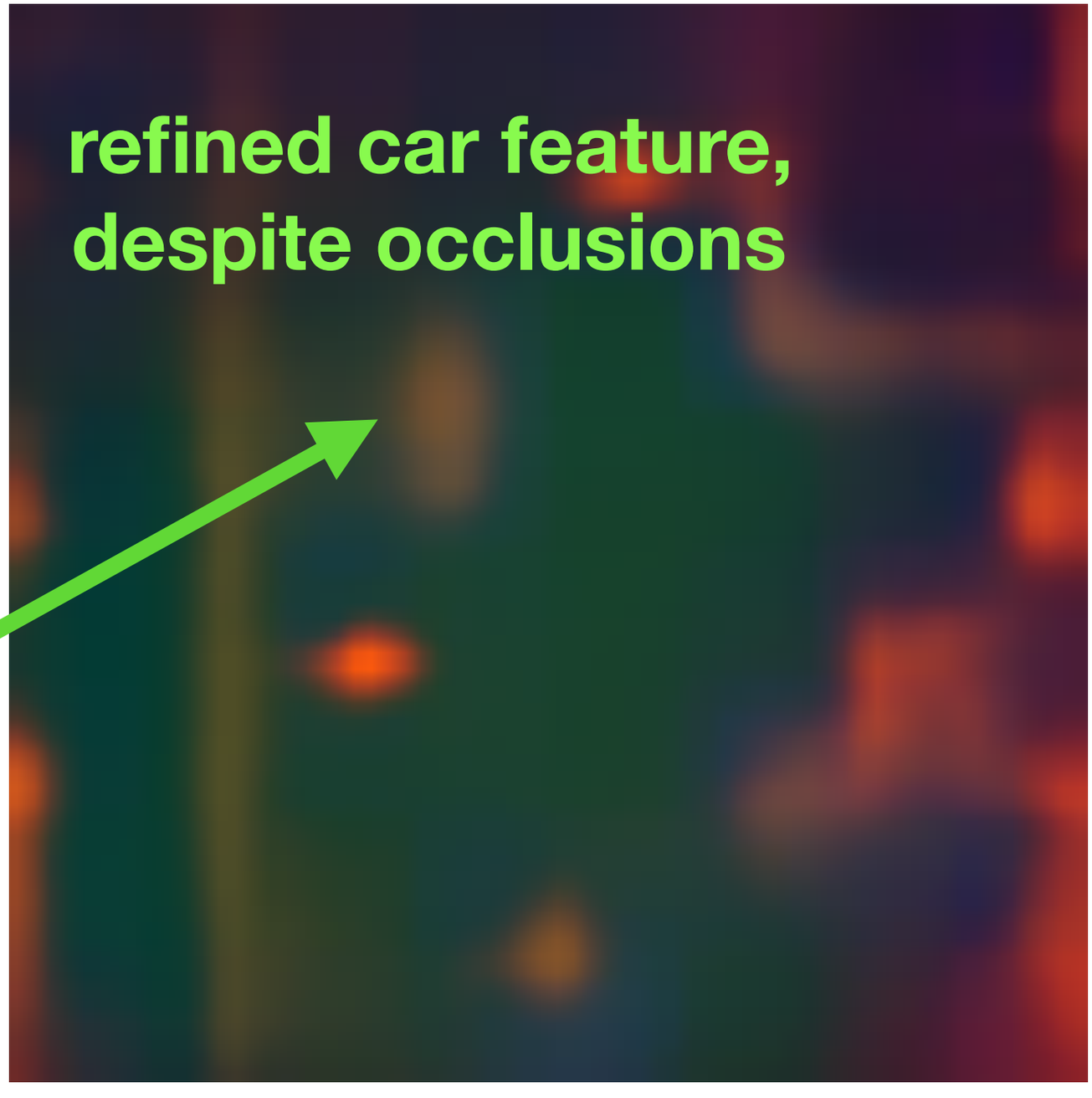
Imagination after 5 frames



Imagination after 6 frames

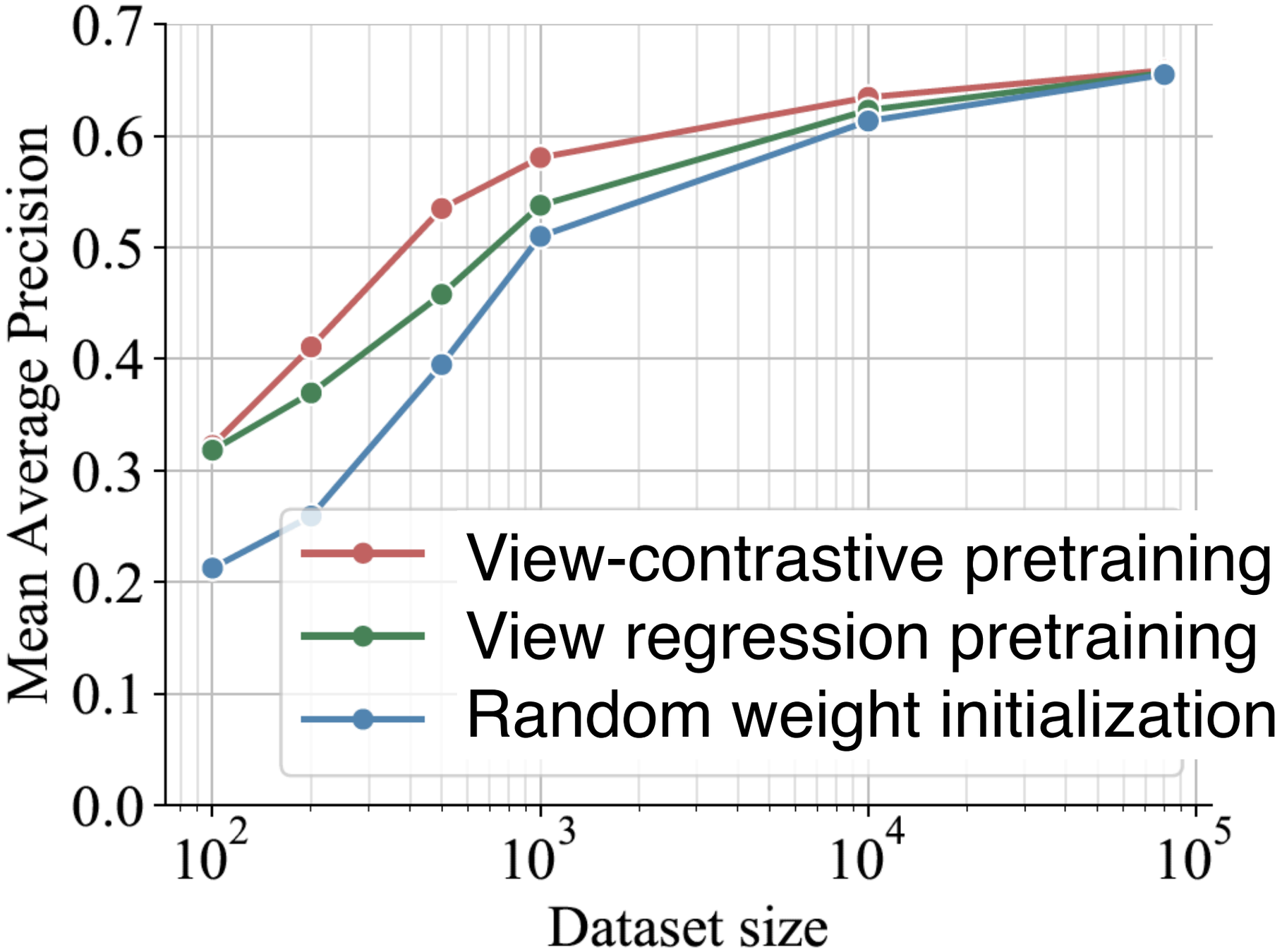


Imagination after 6 frames



View-contrastive GRNN training helps 3D object detection

3D object detection in the CARLA simulator



Embodied View-Contrastive 3D Feature Learni

People detect and segment objects in 3D without even been supplied a 3D box or 3D segmentation mask

Can machines similarly learn to see without any annotations?

3D moving object discovery

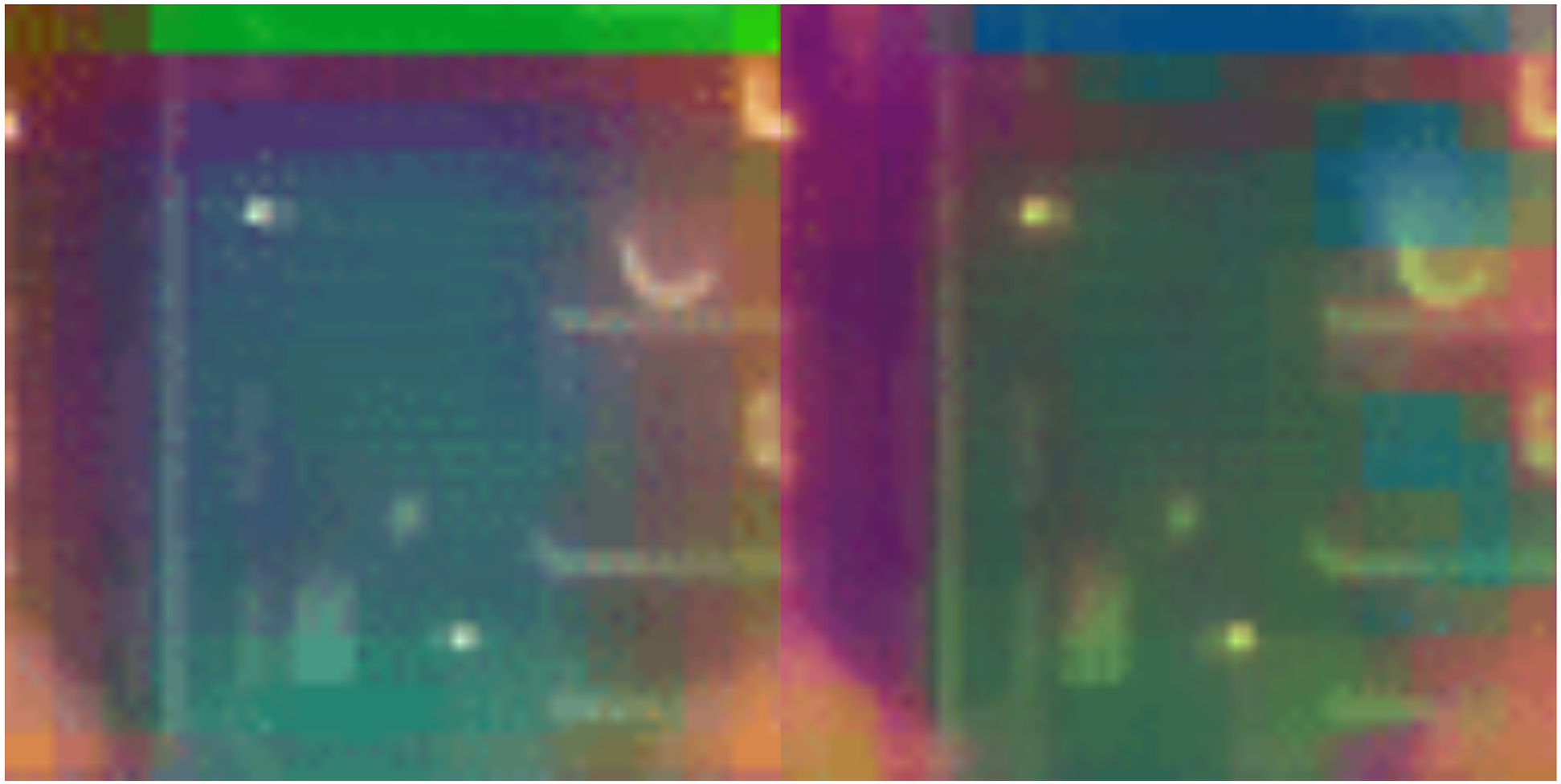
Input: pair of RGB-D frames

3D moving object discovery

Input: pair of RGB-D frames

PCA encoding of 3D feature maps overhead view

3D moving object discovery



Unstabilized

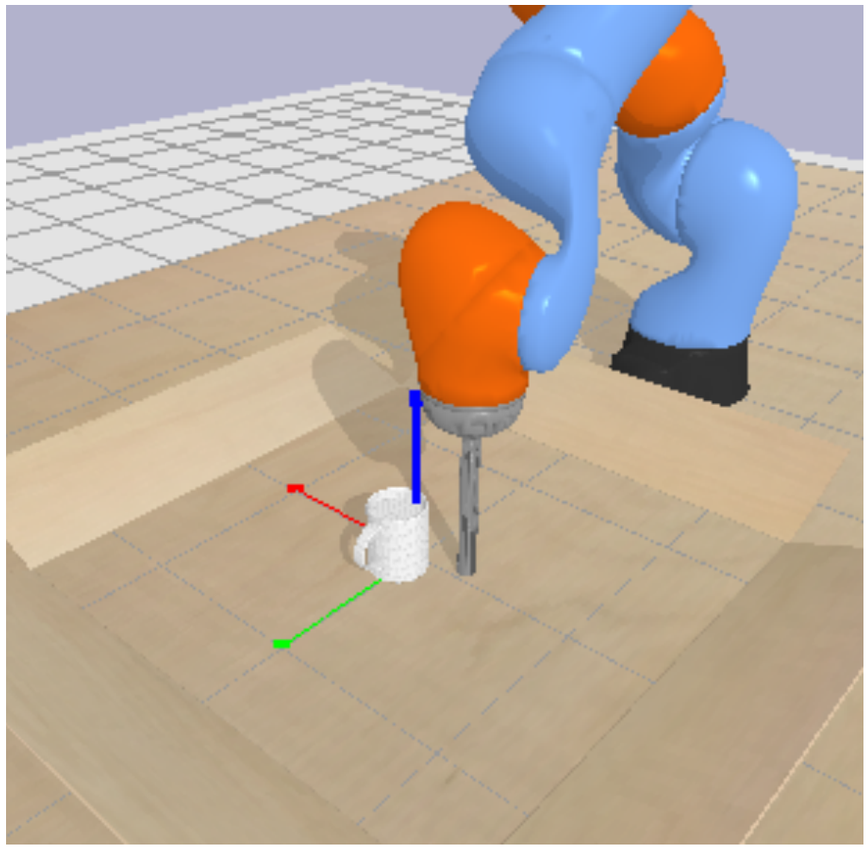
Stabilized

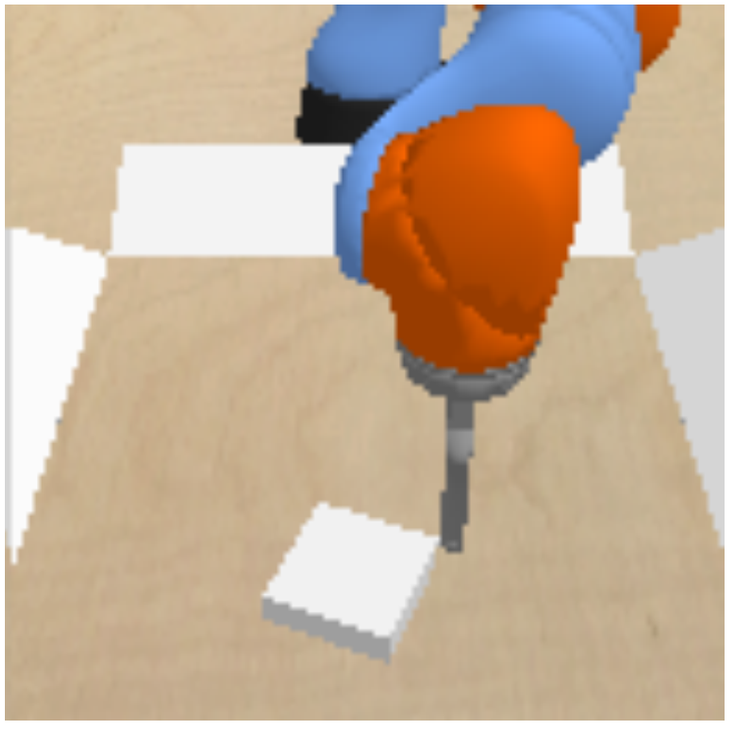
3D moving object discovery

Object proposals

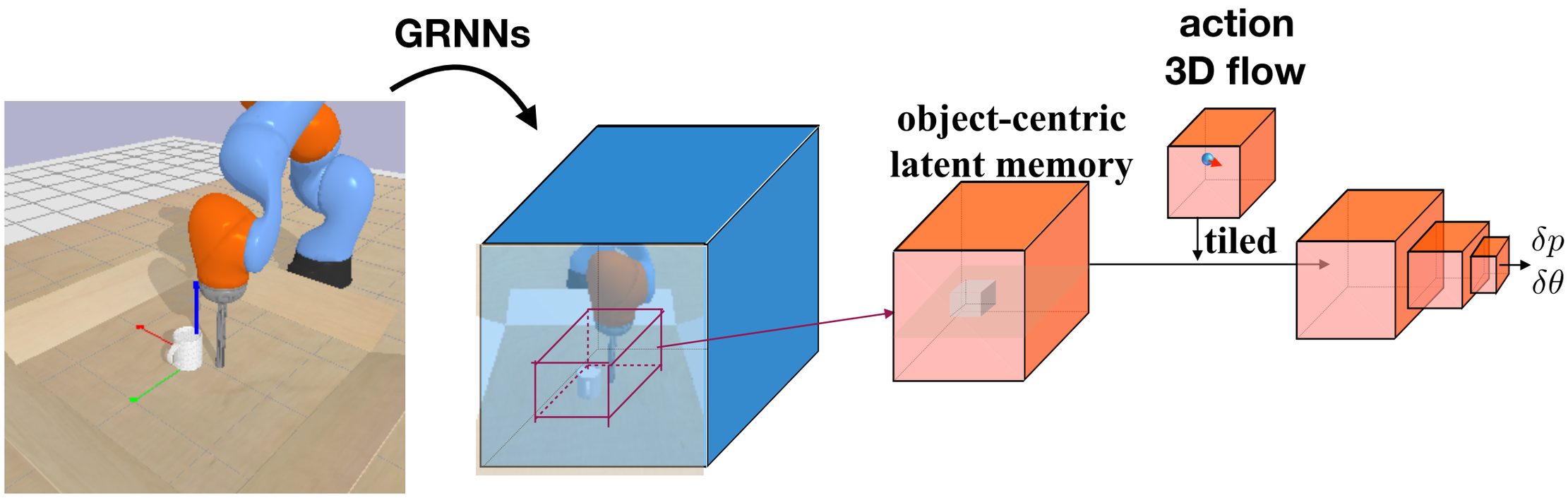
3D objects emerge without any annotations

Learning object 3D motion dynamics

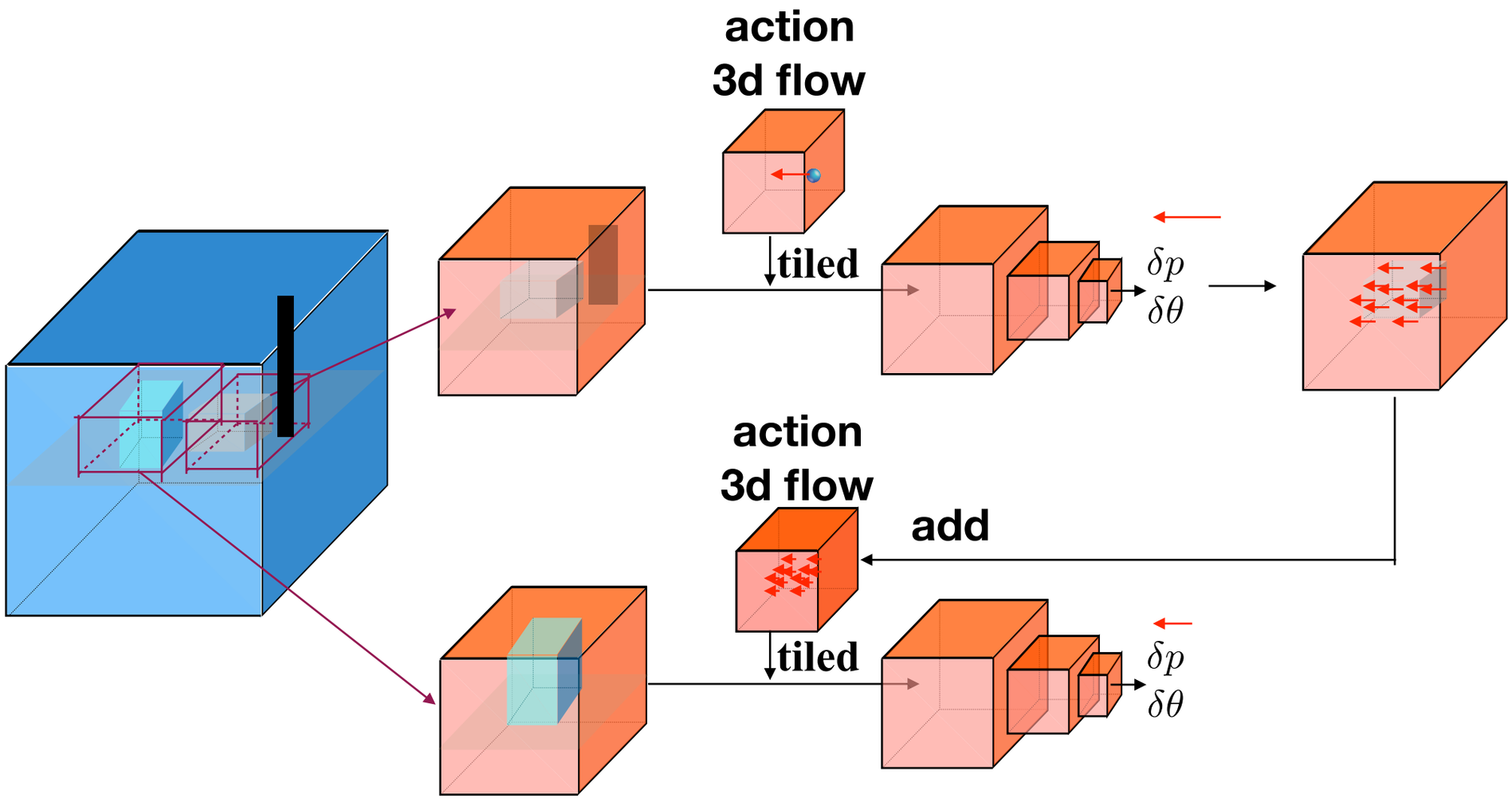




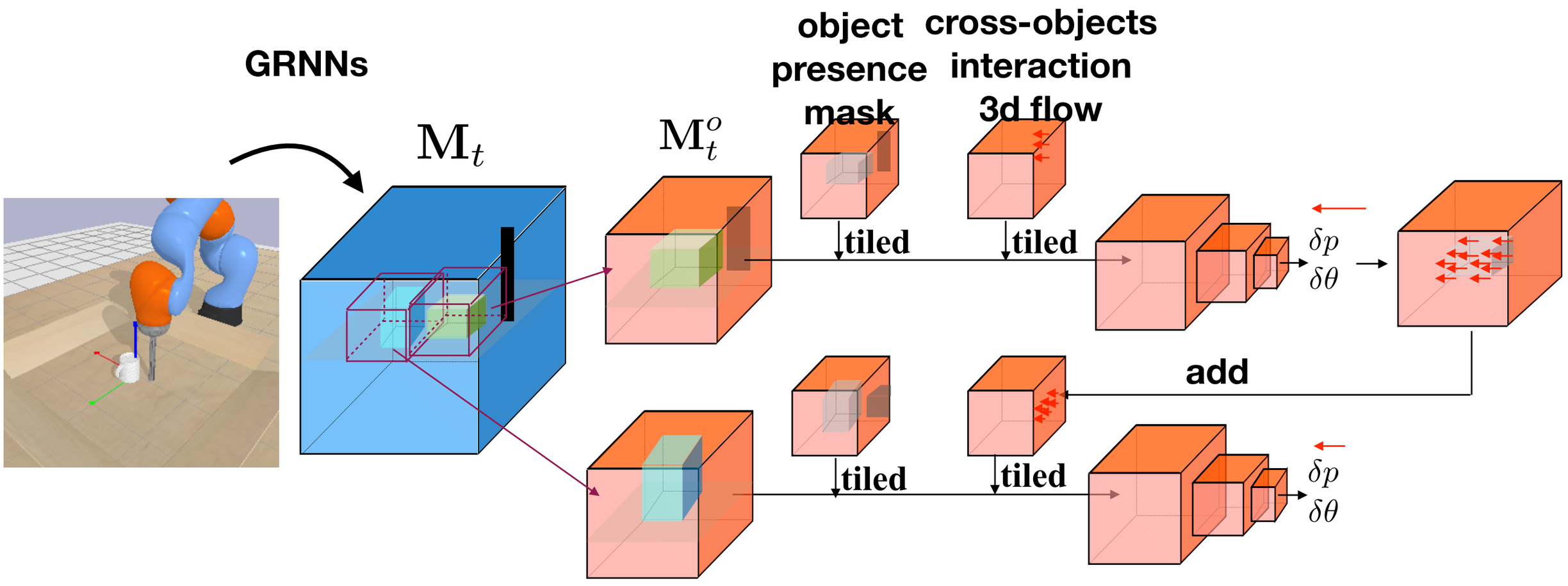
Learning object 3D motion dynamics



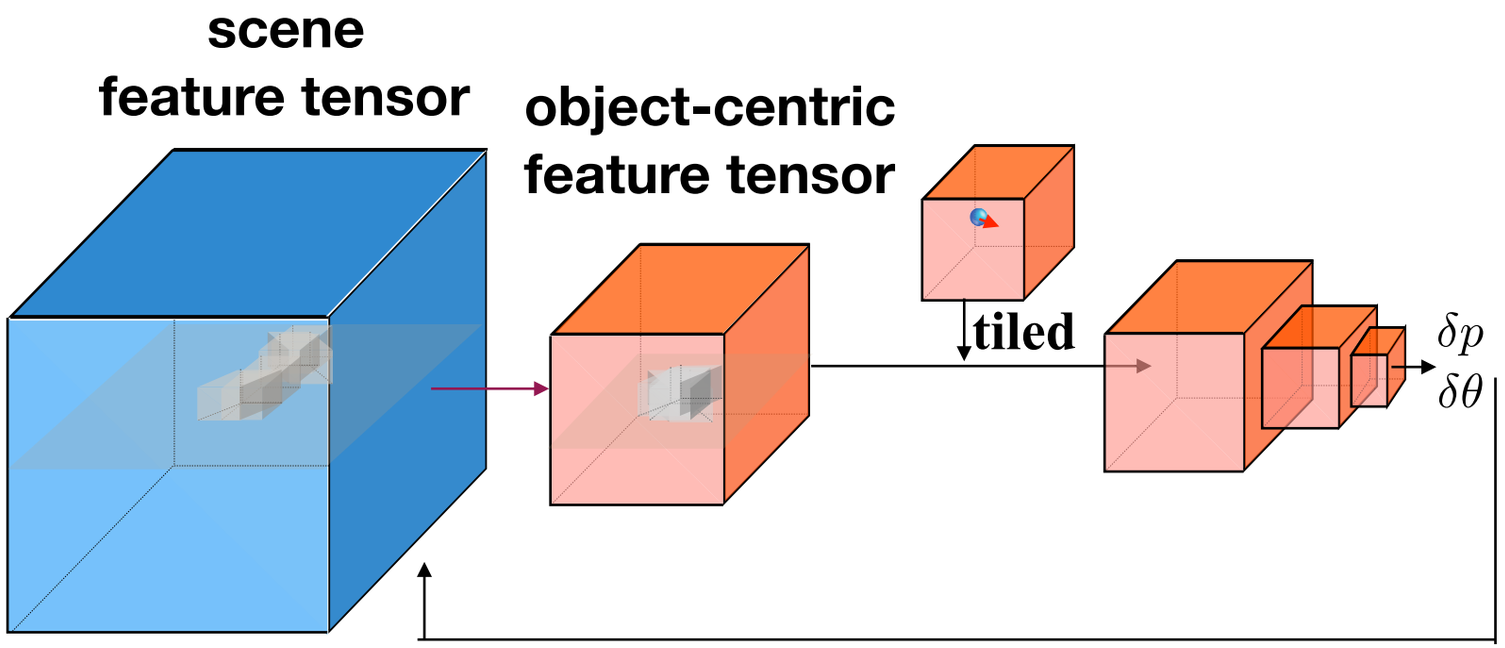
Learning object 3D motion dynamics



Learning object 3D motion dynamics

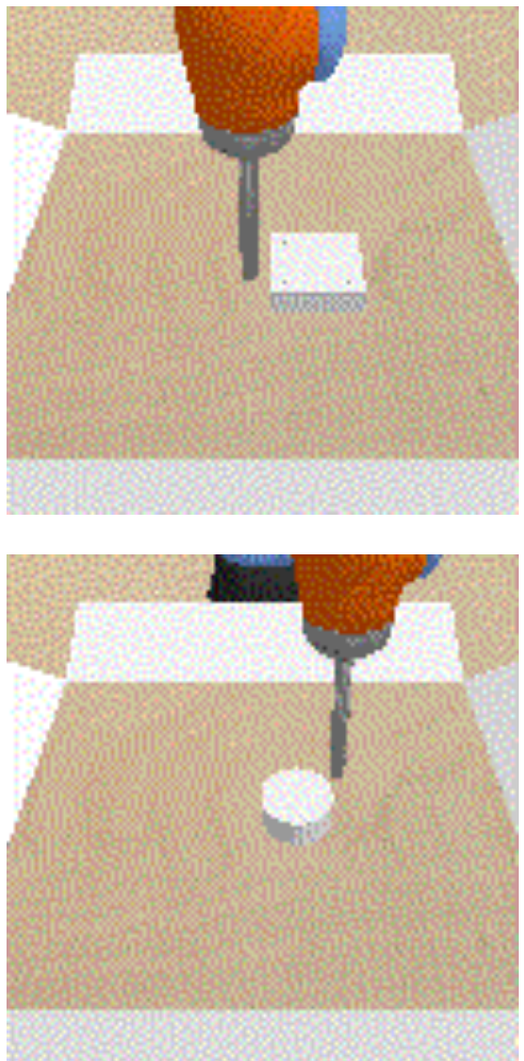


Model forward unrolling temporal skip connections

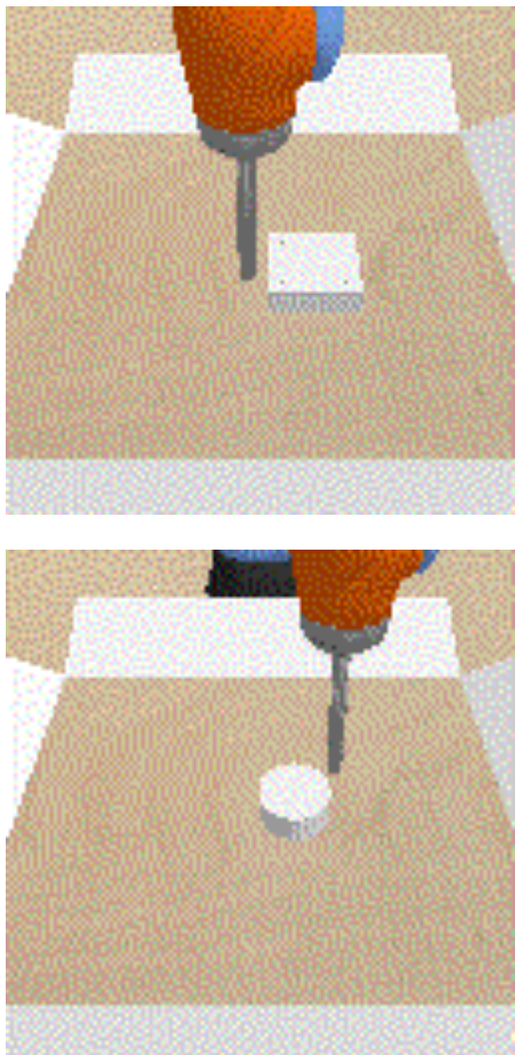


results—dynamics rollout

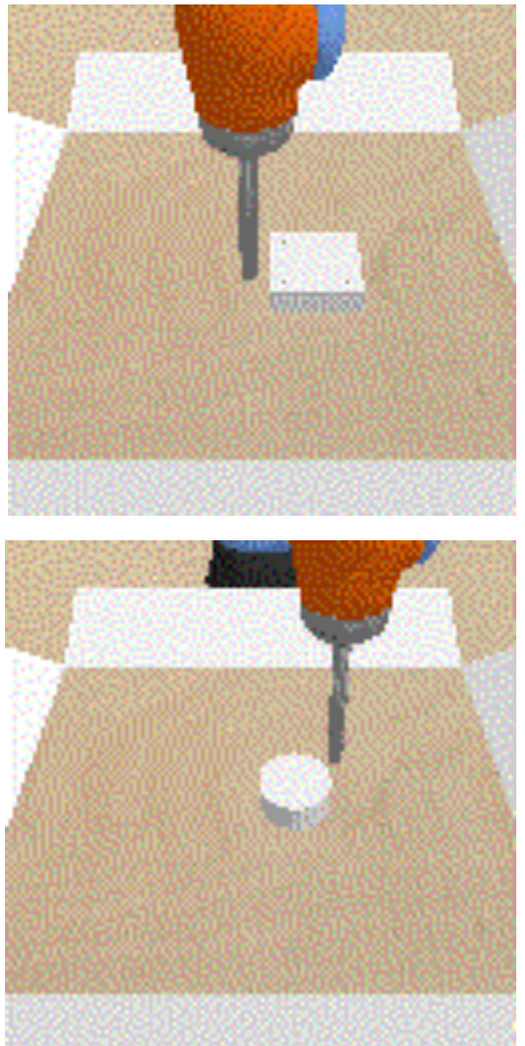
GT



XYZ- Baseline

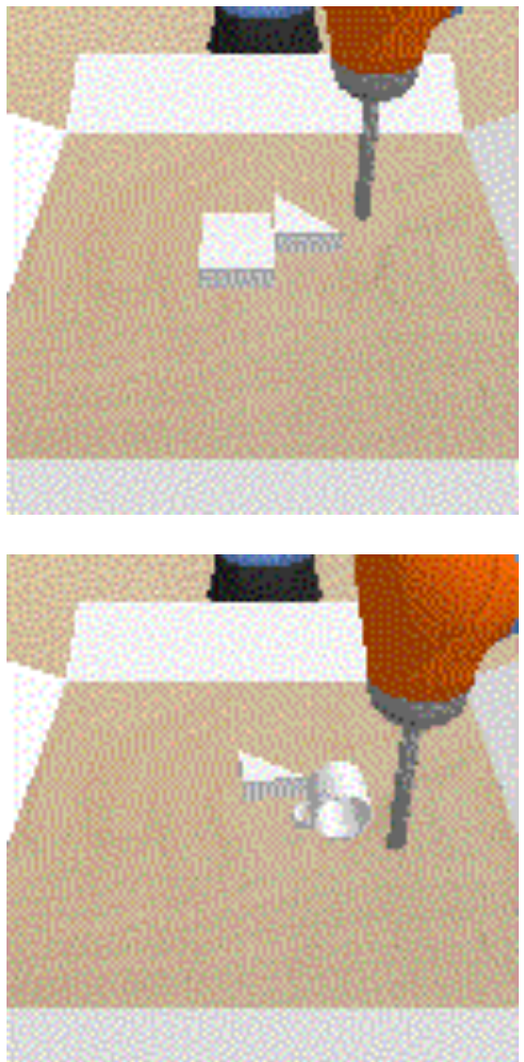


Ours

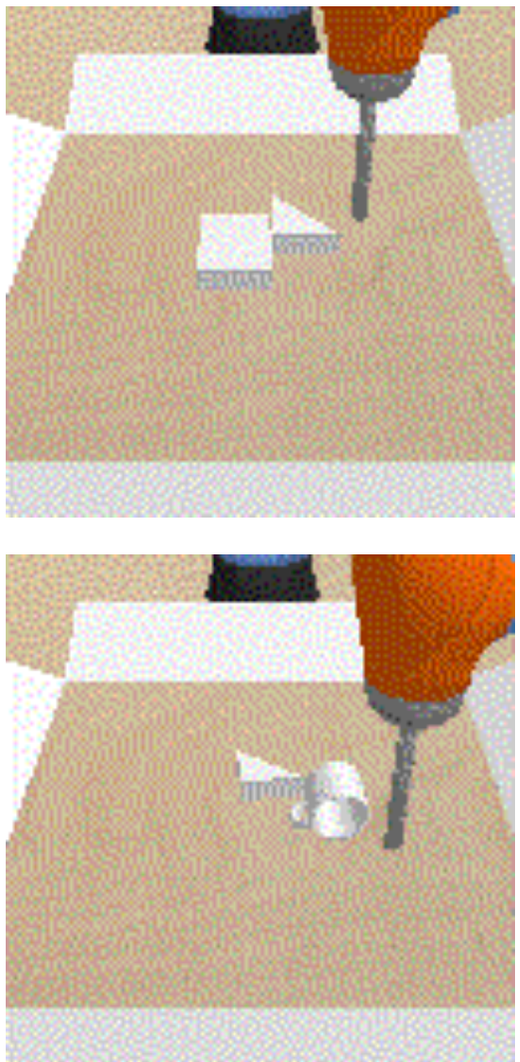


results—dynamics rollout (multiple objects)

GT



XYZ- Baseline

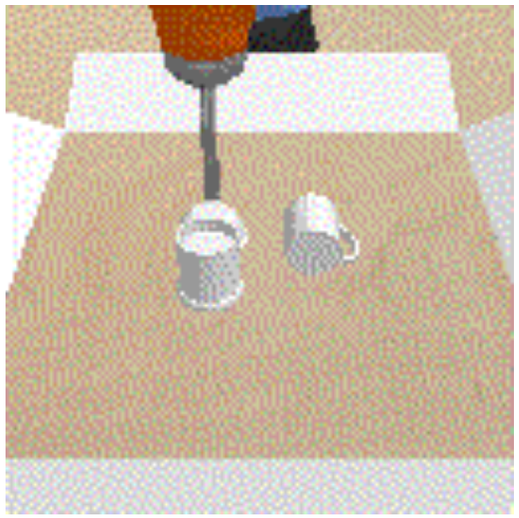


Ours

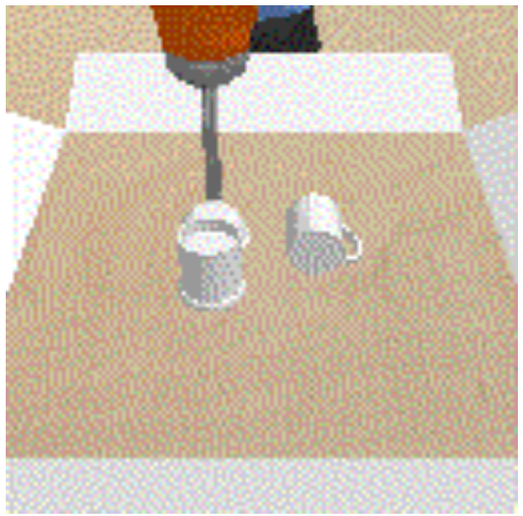
results—dynamics rollout

GT

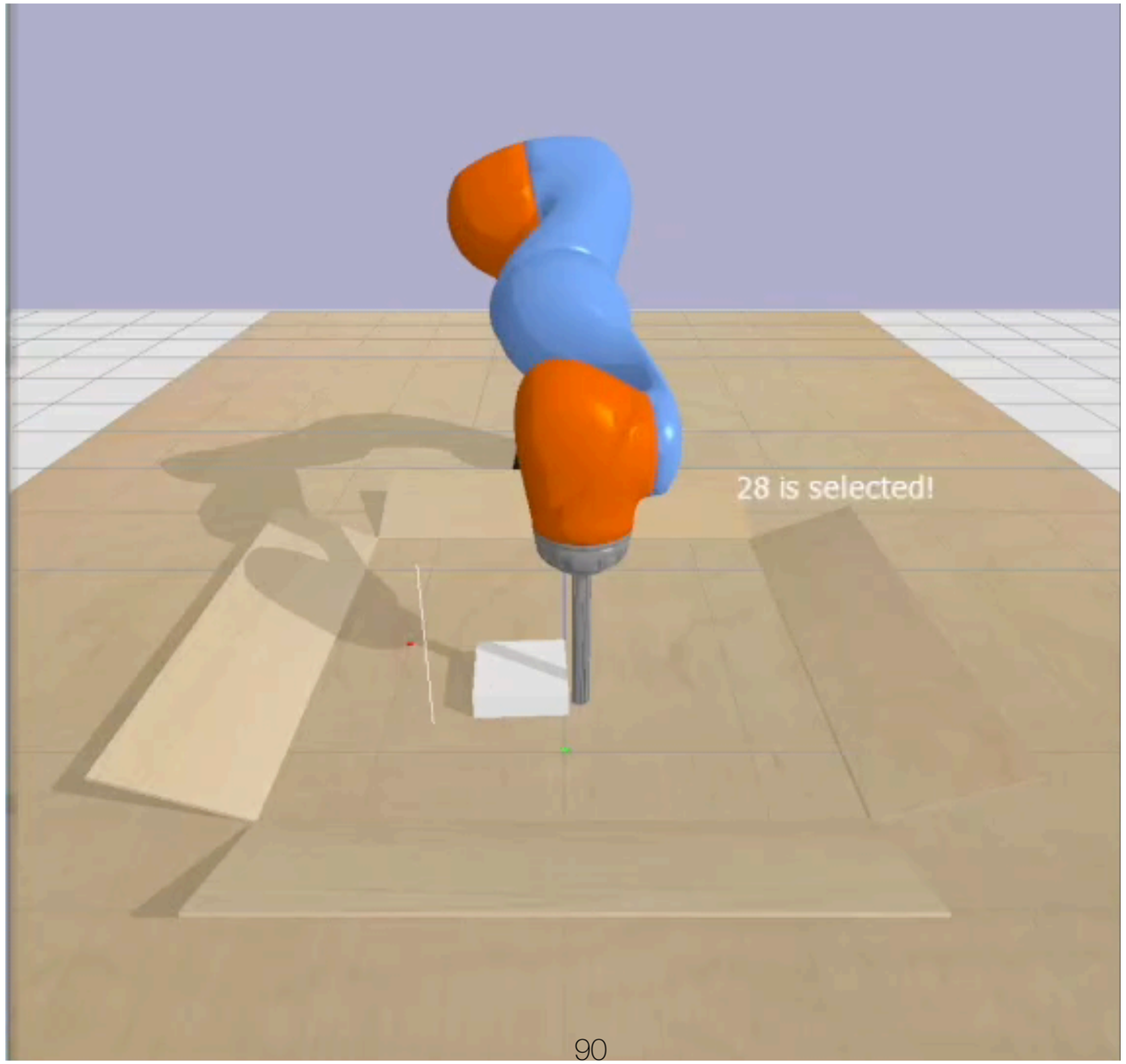
XYZ- Baseline



Ours



Model predictive control



Carnegie Mellon
School of Computer Science

Deep Reinforcement Learning and Control

Inverse Graphics for Instruction Following

CMU 10703

Katerina Fragkiadaki

Describing goals in natural language

Describe goals/subgoals for a robot in natural language, as opposed to:

- hard coding them in the environment in the form of rewards, or
- providing a goal image

“C*an is to the right of the bowl*”

Describe goals/subgoals for a robot in natural language, as opposed to:

- hard coding them in the environment in the form of rewards, or
- providing a goal image

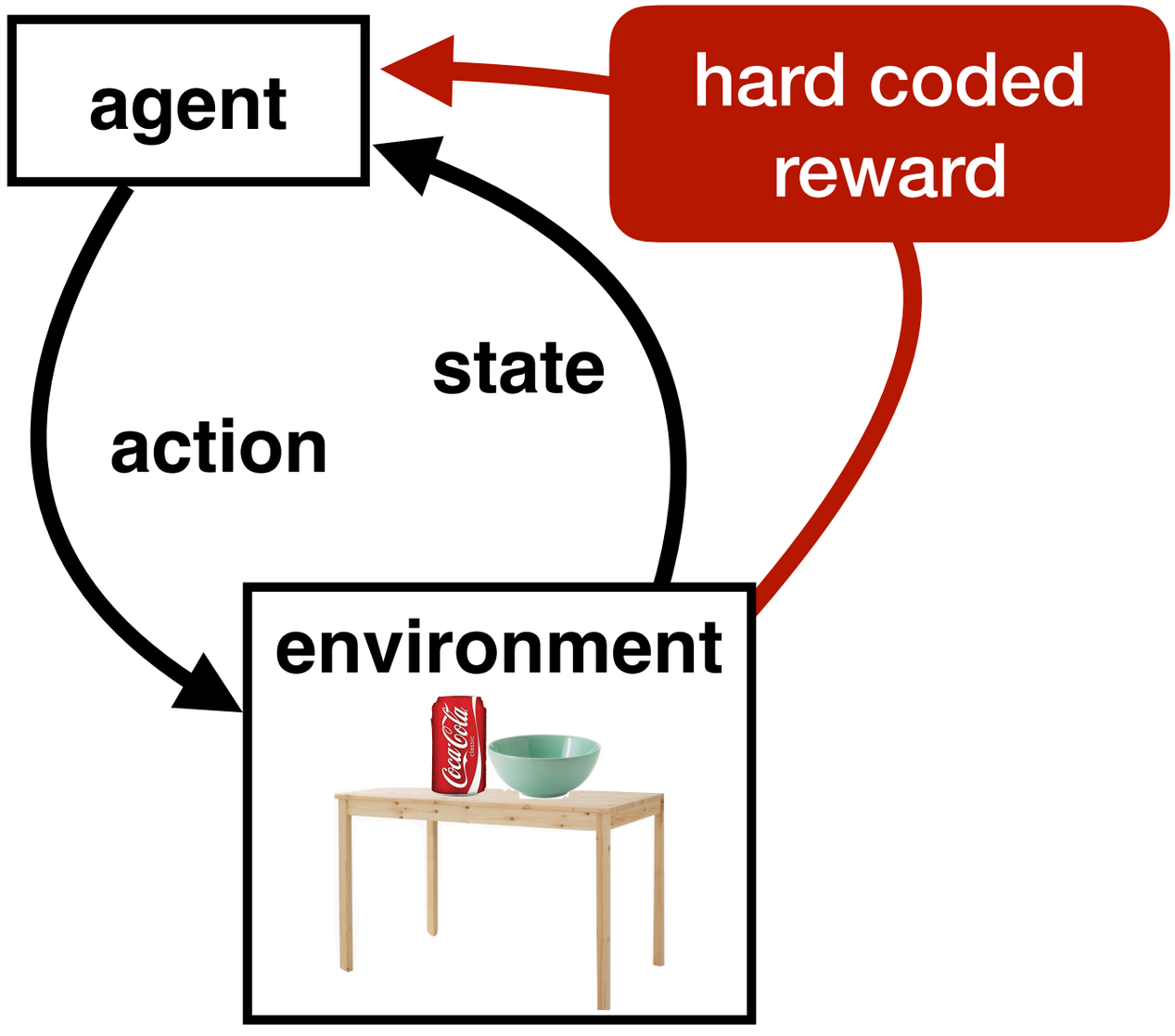
"Can is to the right of the bowl"

Use the learned visual detector to get rewards for policy learning

Tung et al. CVPR 2018

Reward learning using natural language

Goal: place *the coca-cola to the right of the bowl*

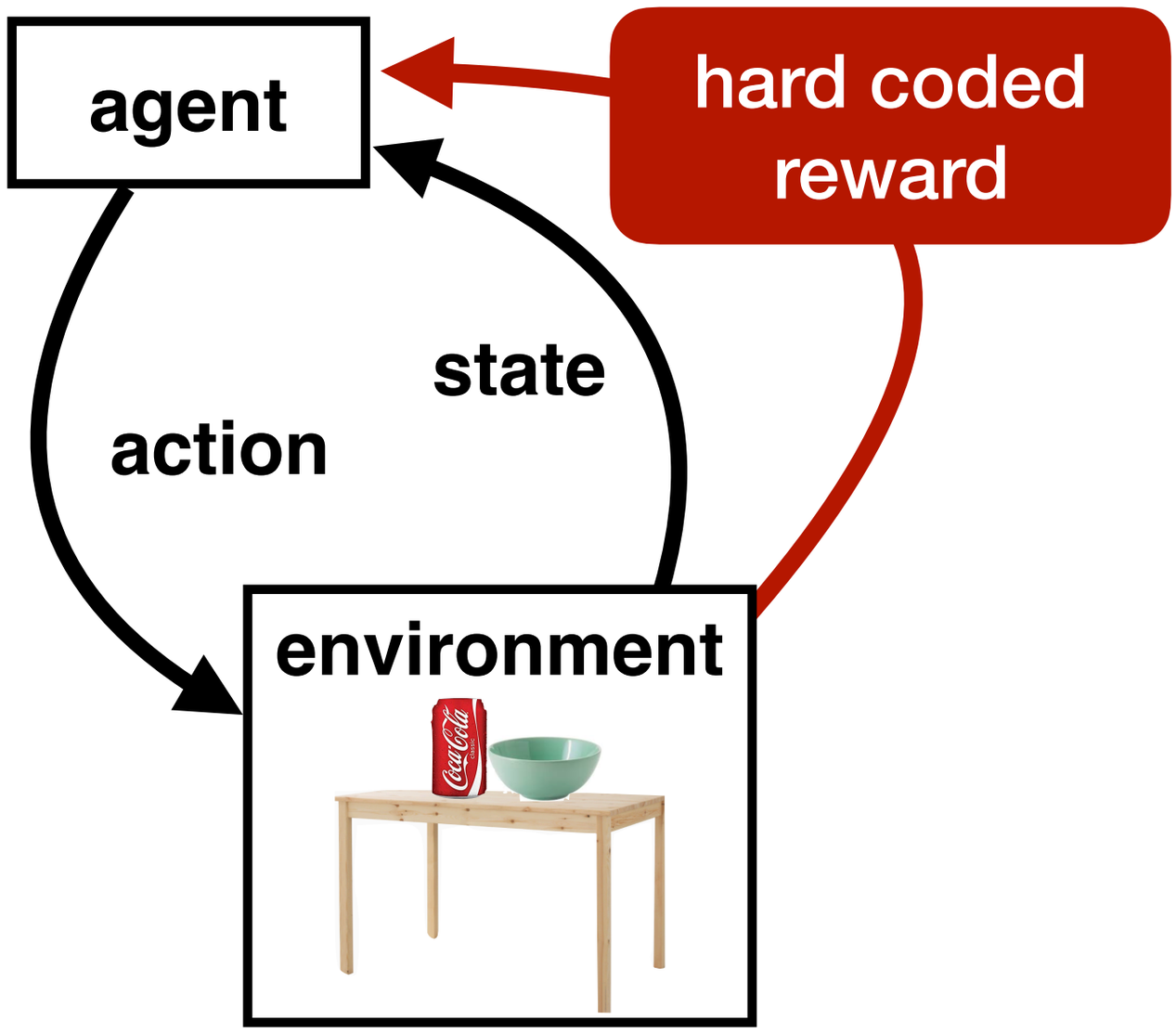


Manually code the reward in a simulated or instrumented environment

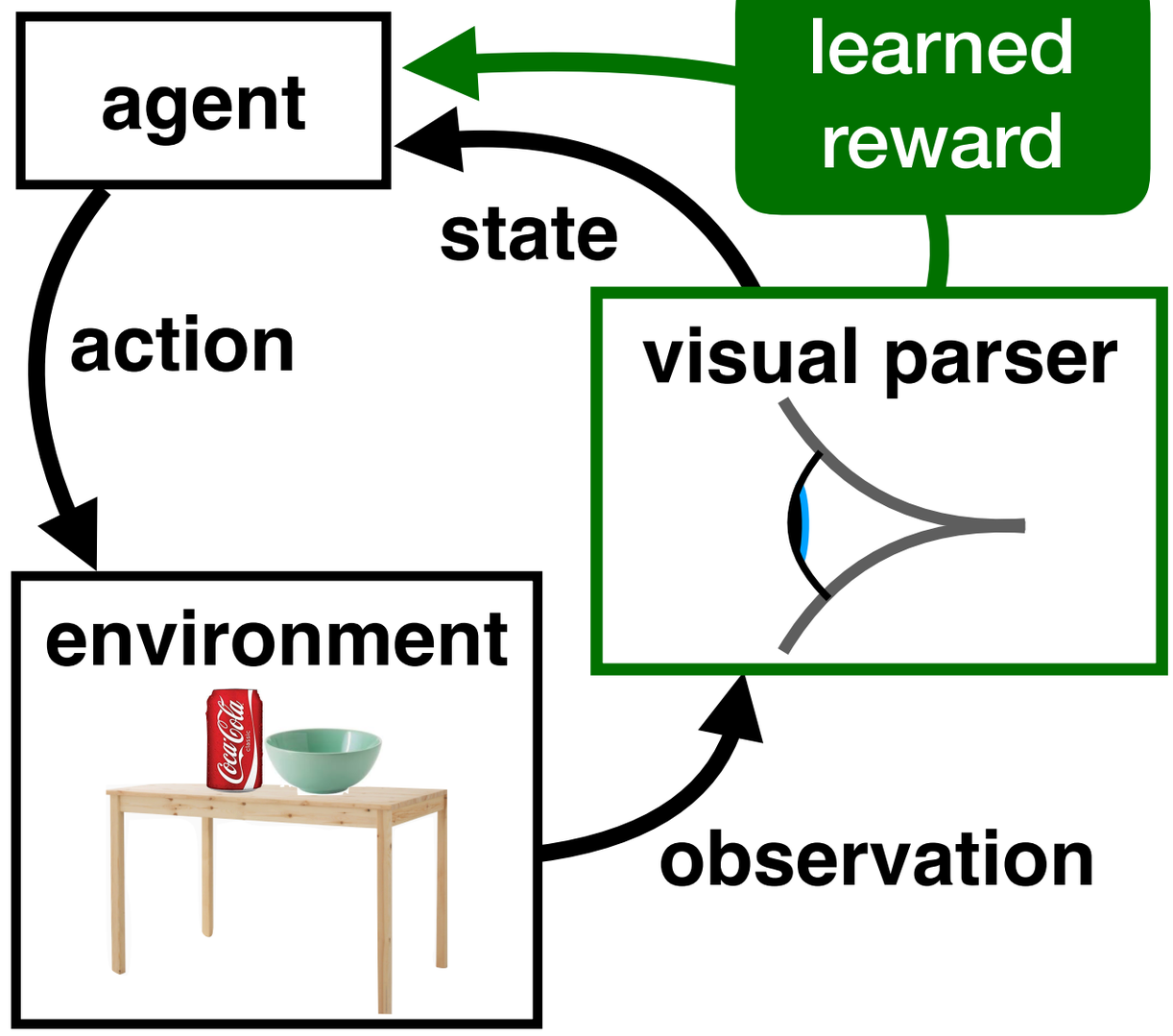
Tung et al. CVPR 2018

Reward learning using natural language

Goal: place *the coca-cola to the right of the bowl*



Manually code the reward in a simulated or instrumented environment

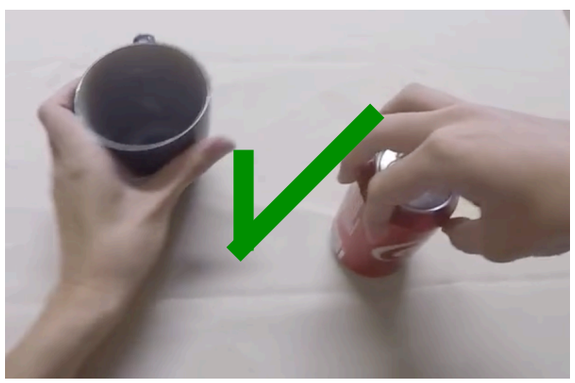


Learn to detect from an RGB image when the goal is achieved

Tung et al. CVPR 2018

Reward learning using natural language

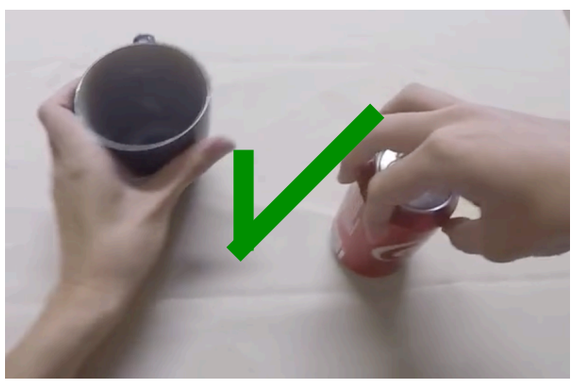
“*Can is to the right of the mug*”



Modeling Relationships in Referential Expressions with Compositional Modular Networks, Ronghang et al.

Reward learning using natural language

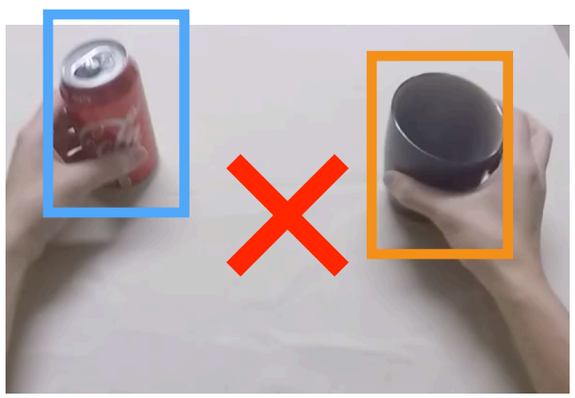
"Can *is to the right of the mug*"



Modeling Relationships in Referential Expressions with Compositional Modular Networks, Ronghang et al.

Reward learning using natural language

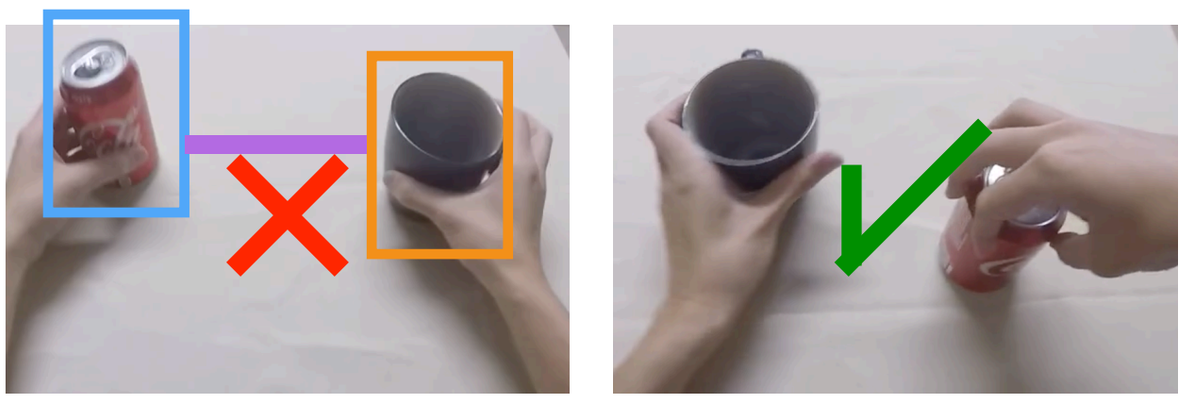
"*Can is to the right of the mug*"



Modeling Relationships in Referential Expressions with Compositional Modular Networks, Ronghang et al.

Reward learning using natural language

"*Can is to the right of the mug*"

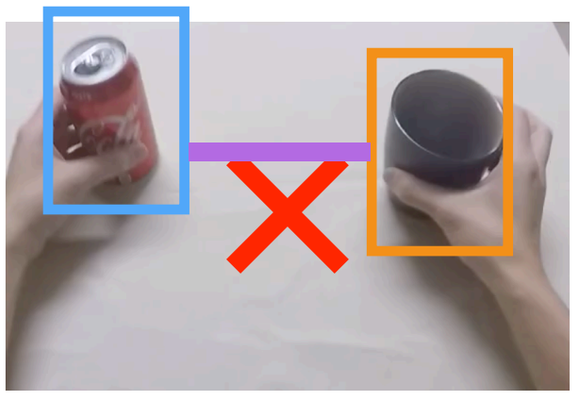


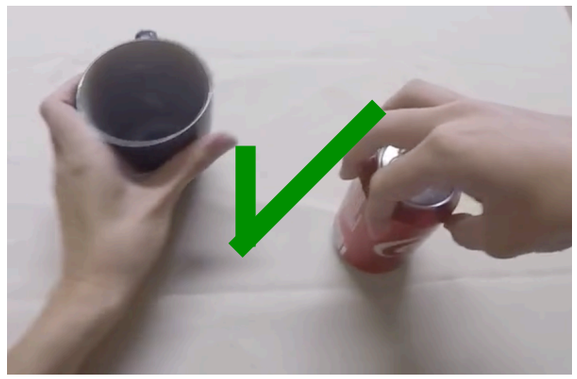
reward detector

Modeling Relationships in Referential Expressions with Compositional Modular Networks, Ronghang et al.

Reward learning using natural language

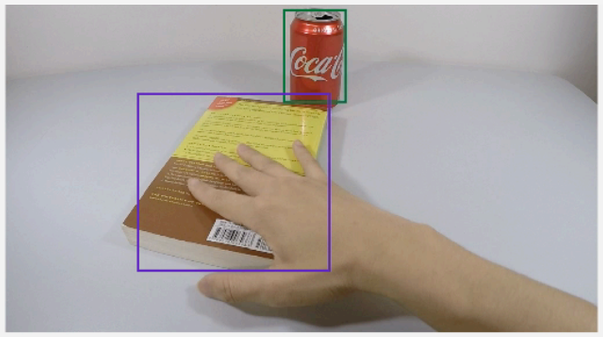
"*Can is to the right of the mug*"



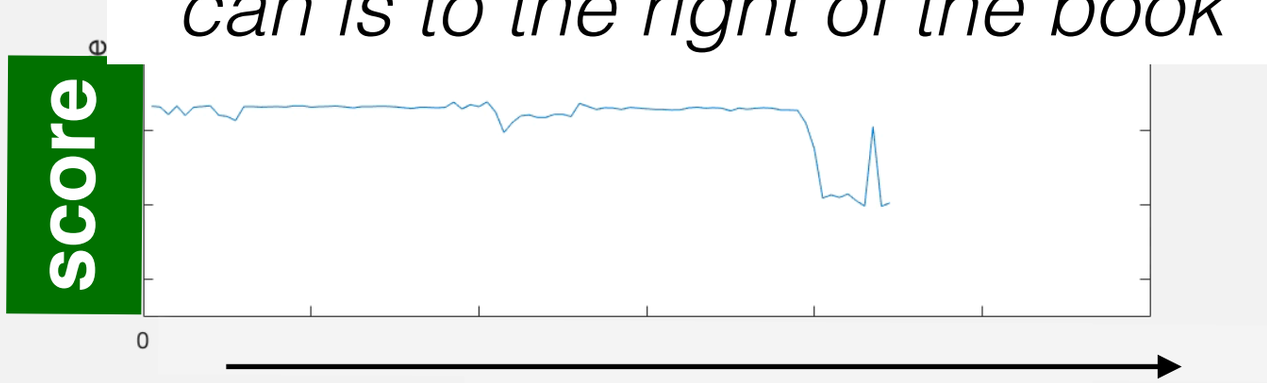


reward detector

Learned reward detector



``*can is to the right of the book*"



Learned policy

Reward learning using natural language

“Can is to the right of the mug”

Our conclusions:

- the reward detector could not effectively generalize across camera placements
- could not provide shaped rewards
- could not discerne impossible goals for possible ones, e.g., “*the mug inside the coca cola”* versus “*the coca cola inside the mug”*

Tung et al. CVPR 2018

Affordandable visual representations

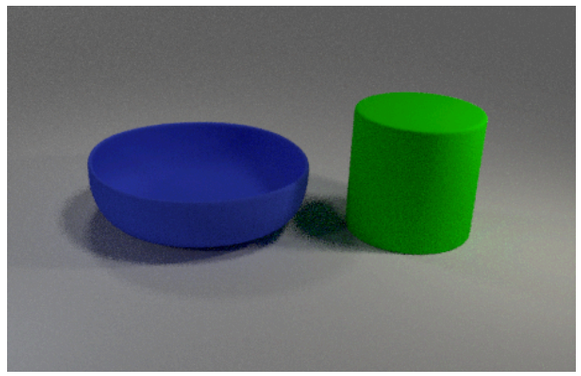
We seek visual feature representations to ground NL onto that obey basic spatial common sense constraints:

- Objects have 3D extent
- Objects do not interpenetrate in 3D
- Objects come in regular sizes
- Objects persist over time

Embodied language grounding

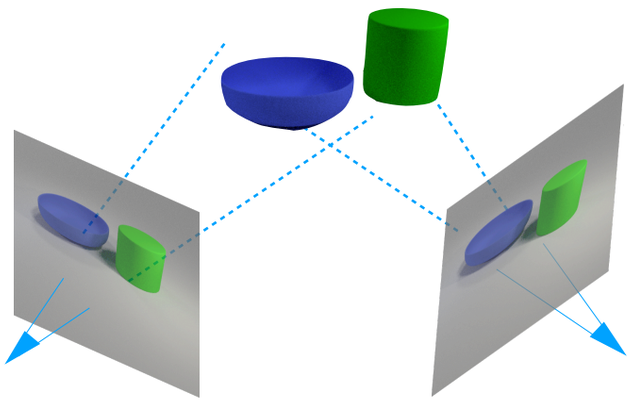
Learn to associate natural language utterances with 3D feature representations of the scene described.

“The green rubber cylinder is on the right of the blue bowl”



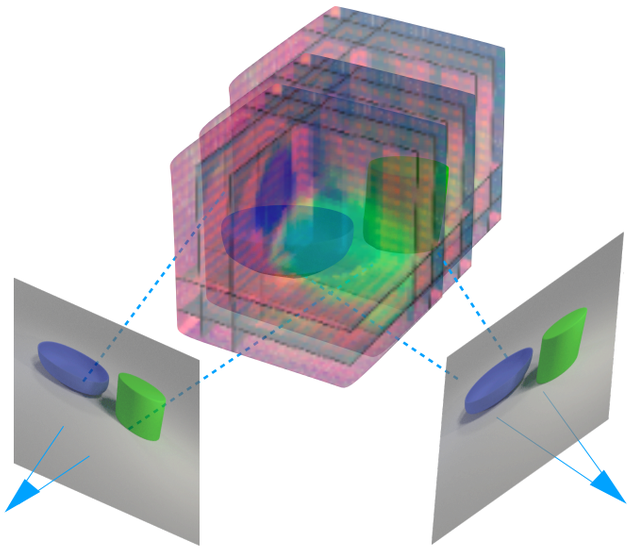
1. We consider an embodied agent that can see a scene from multiple viewpoints

“The green rubber cylinder is on the right of the blue bowl”



1. We consider an embodied agent that can see a scene from multiple viewpoints

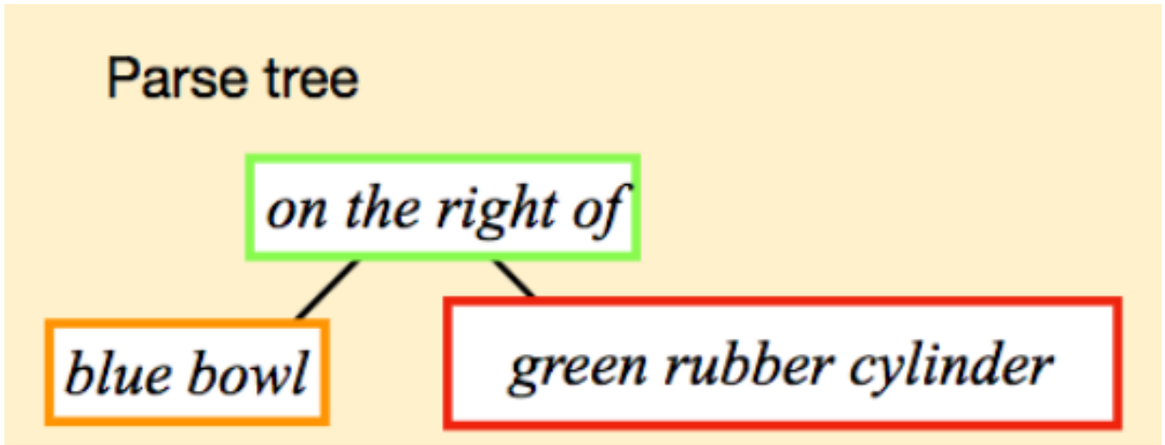
“The green rubber cylinder is on the right of the blue bowl”



2. Our agent learns to map an RGB image to a set of 3D feature maps by training GRNNs to predict views

“The green rubber cylinder is on the right of the blue bowl”

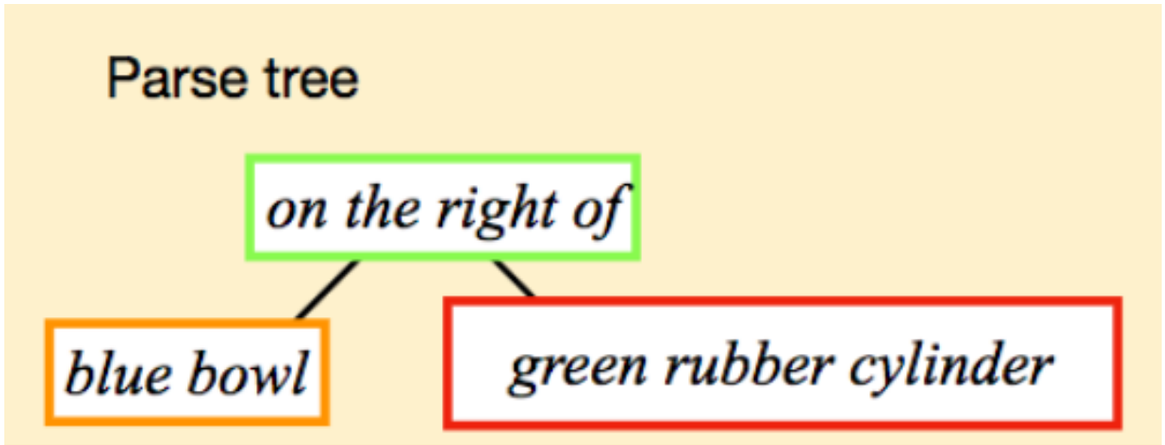
“The green rubber cylinder is on the right of the blue bowl”



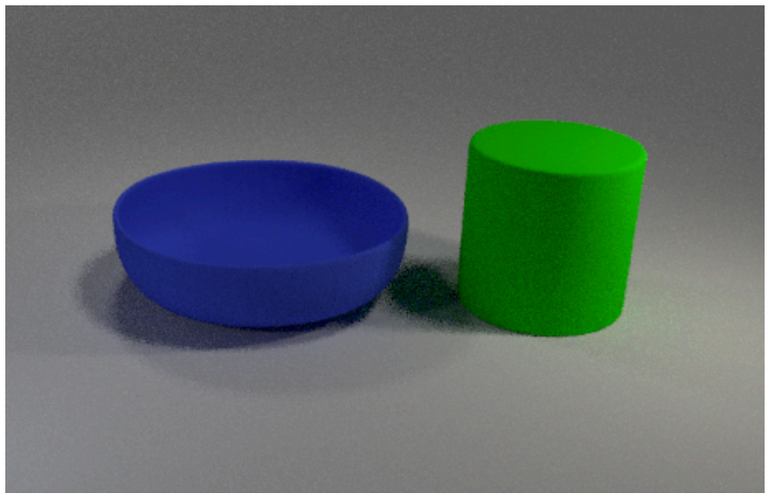
“The green rubber cylinder is on the right of the blue bowl”

Where:

What:

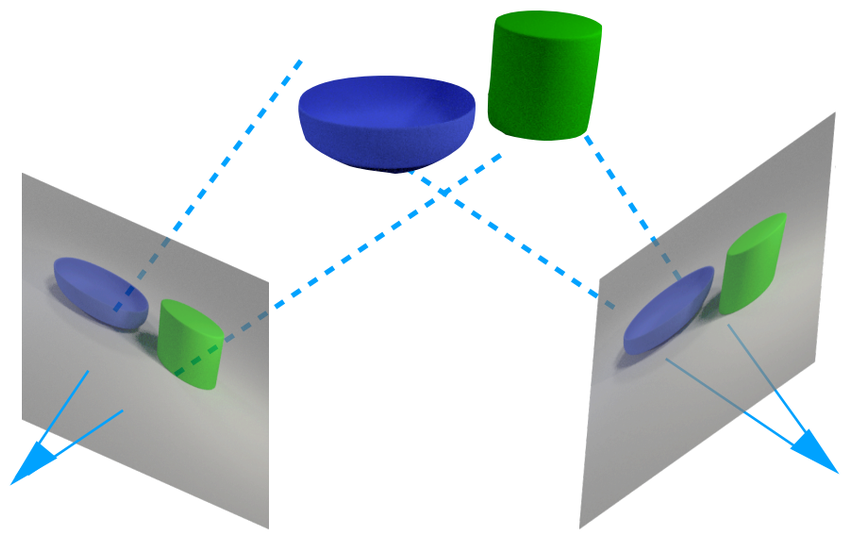


“The green rubber cylinder is on the right of the blue bowl”



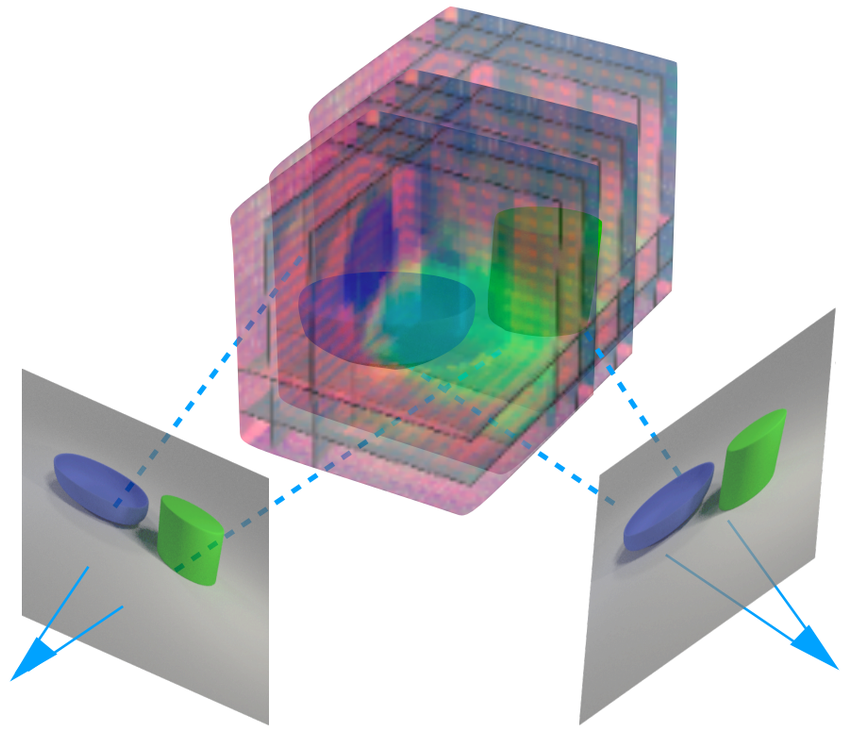
1. We consider an embodied agent that can see a scene from multiple viewpoints

“The green rubber cylinder is on the right of the blue bowl”



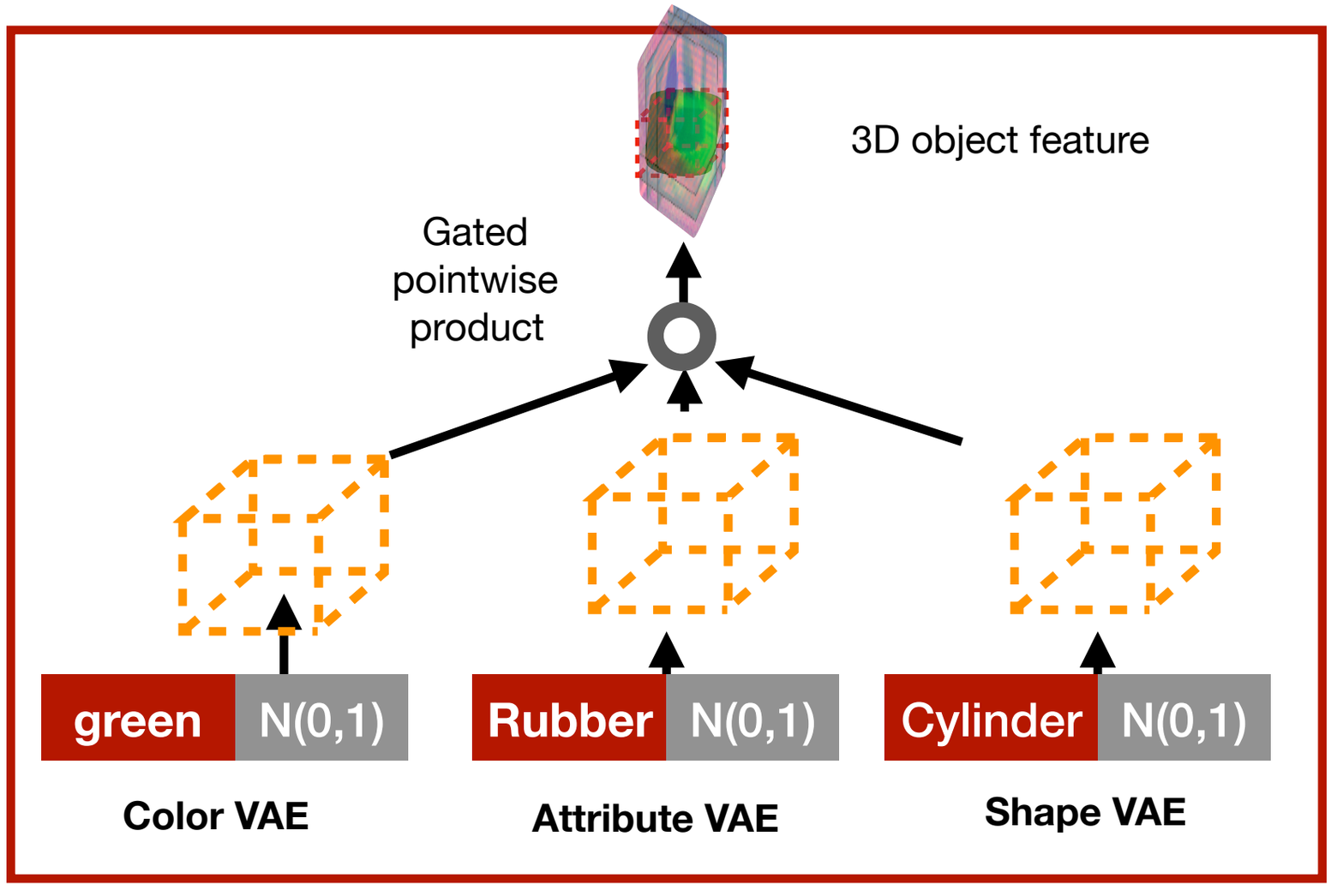
1. We consider an embodied agent that can see a scene from multiple viewpoints

“The green rubber cylinder is on the right of the blue bowl”



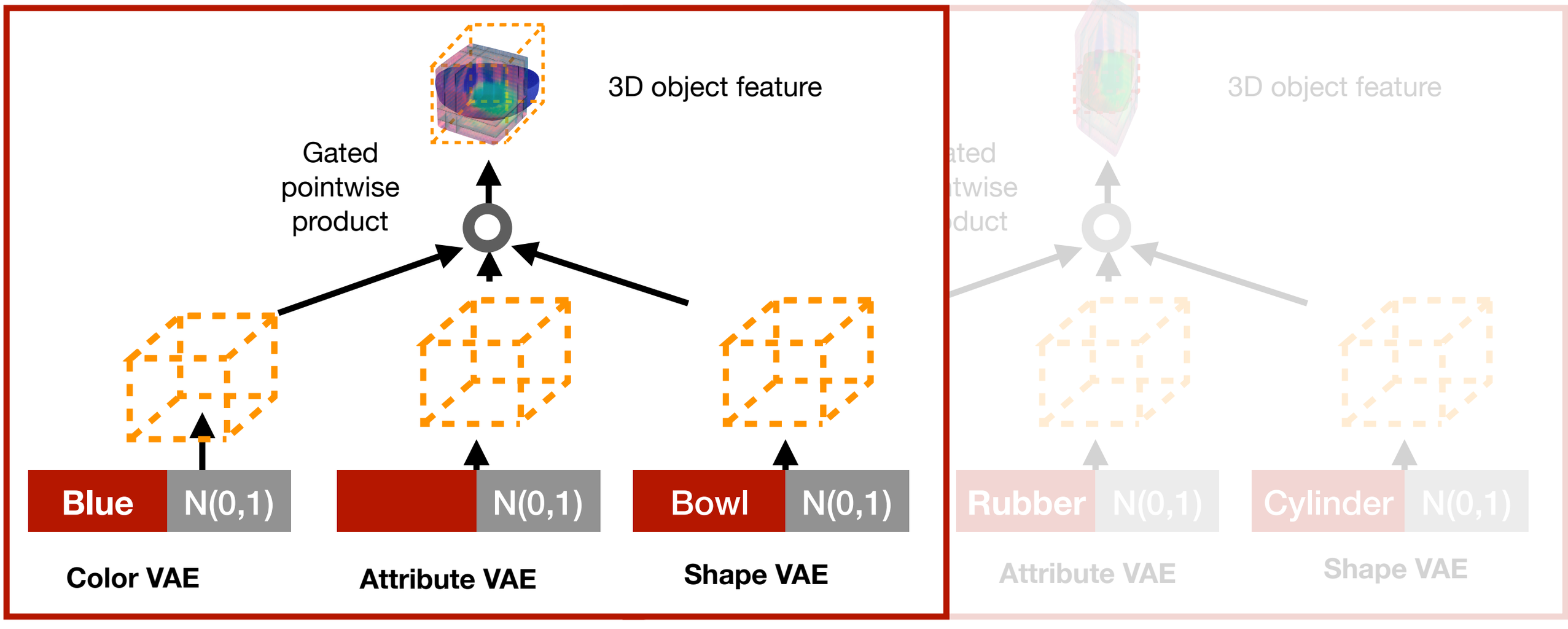
2. Our agent learns to map an RGB image to a set of 3D feature maps by training GRNNs to predict views

*"The **green rubber cylinder** is on the right of the blue bowl"*



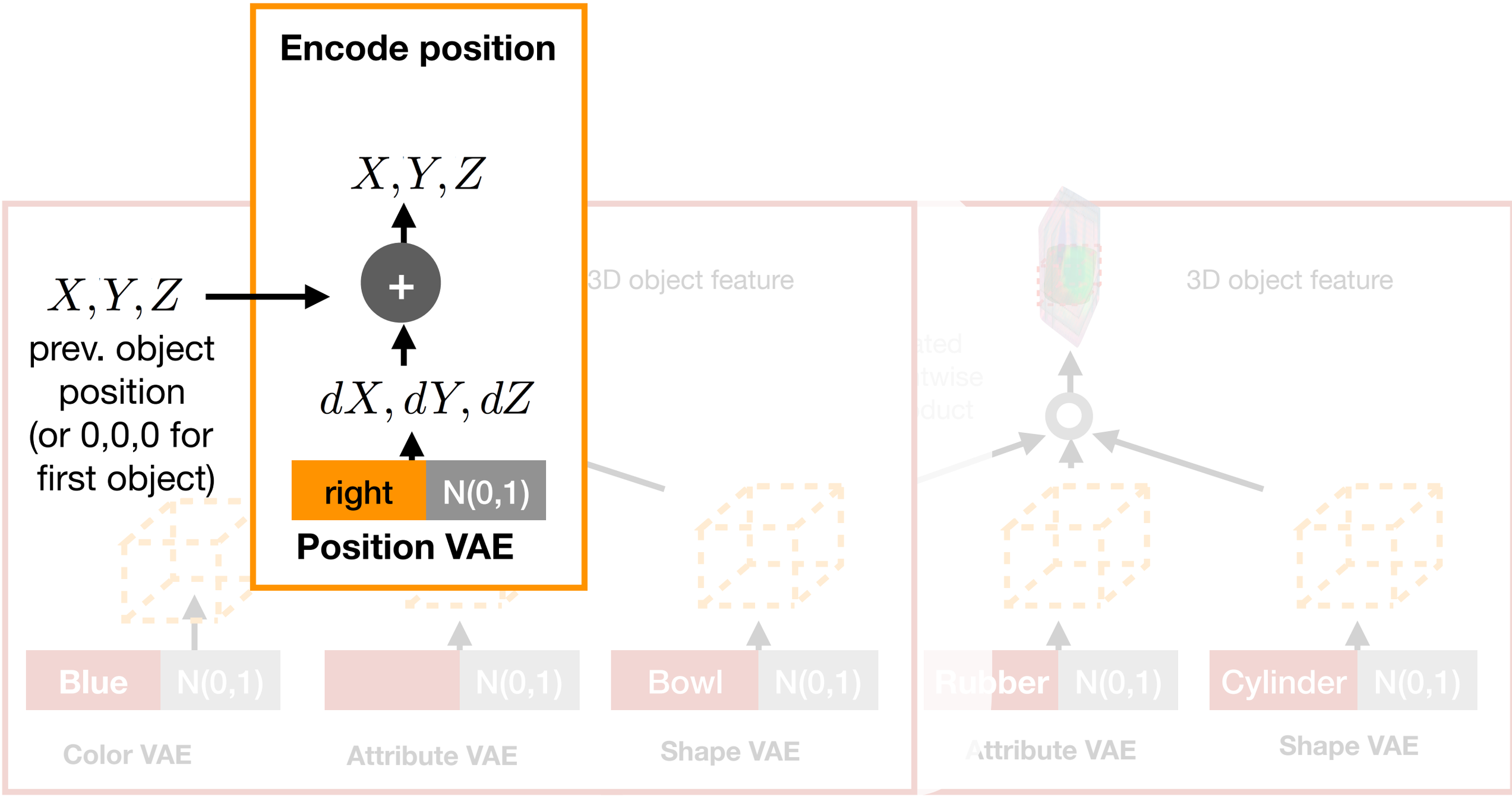
3. Our agent maps noun phrases to object-centric 3D feature maps

*"The green rubber cylinder is on the right of the **blue bowl**"*



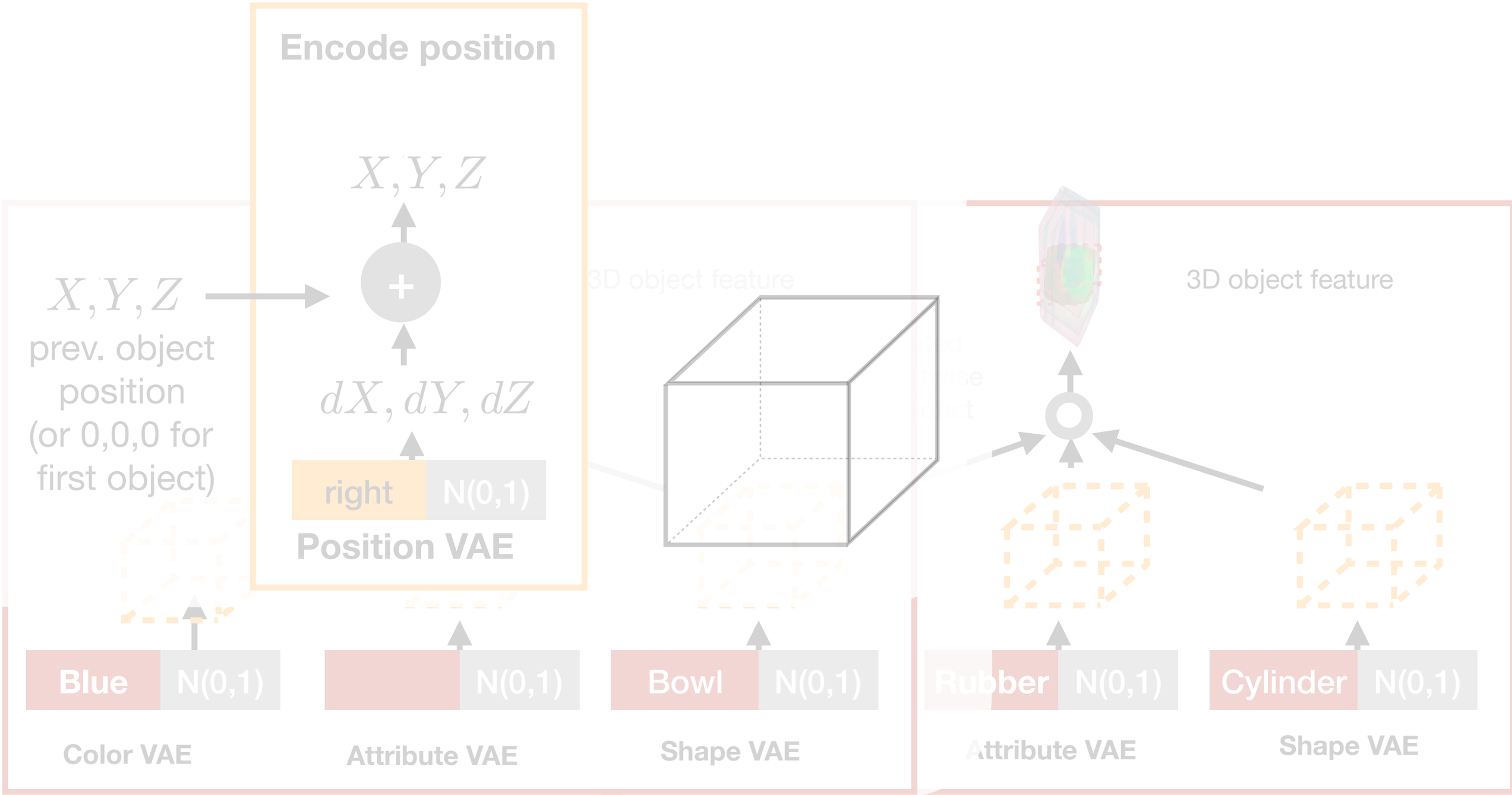
3. Our agent maps noun phrases to object-centric 3D feature maps

"The green rubber cylinder is
***on the right of** the blue bowl"*



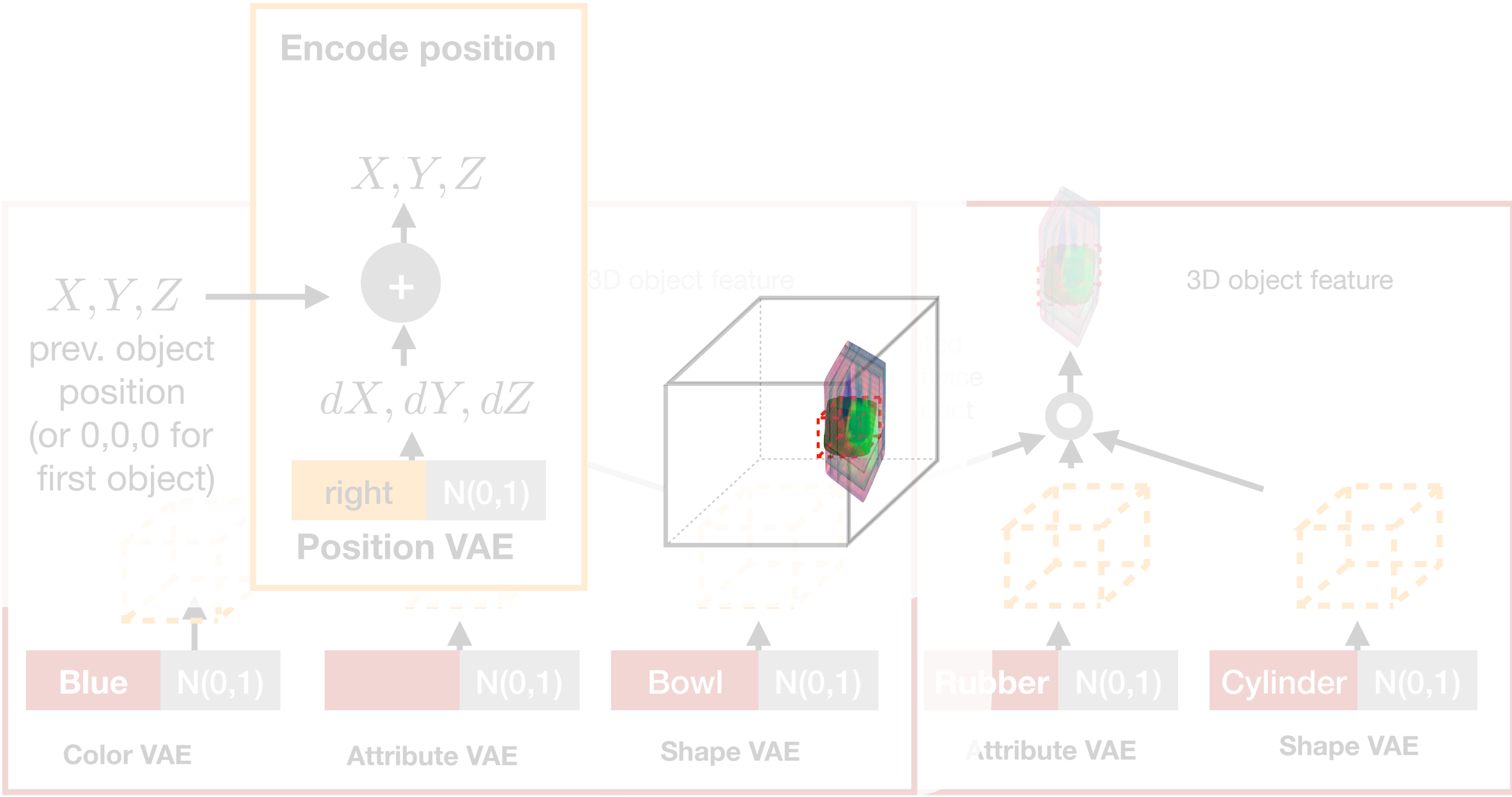
4. Our agent maps spatial expressions to relative 3D offsets

“The green rubber cylinder is on the right of the blue bowl”



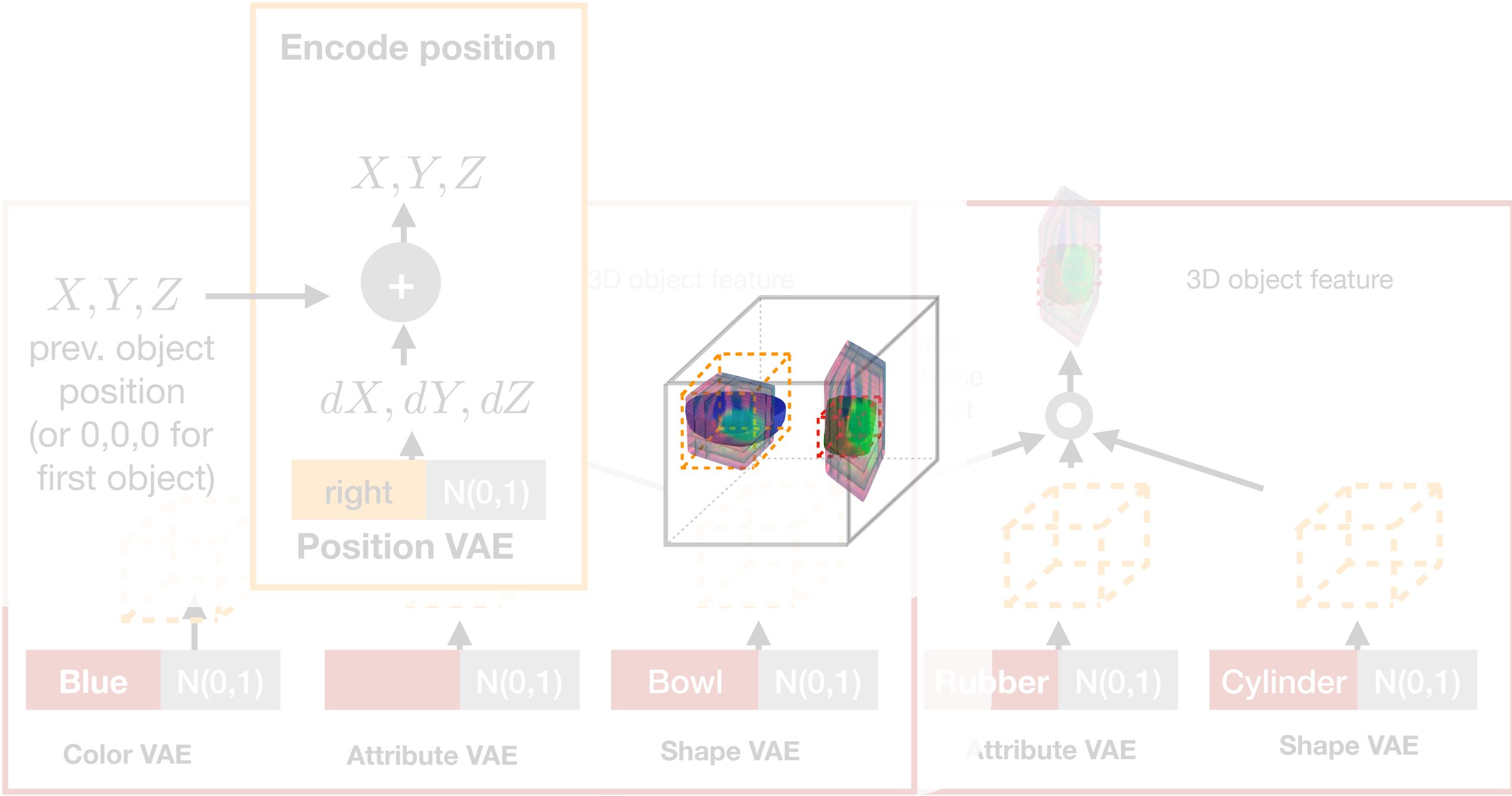
5. Our agent populates a 3D canvas with the predicted object tensors adn their relative offsets

"The green rubber cylinder is on the right of the blue bowl"



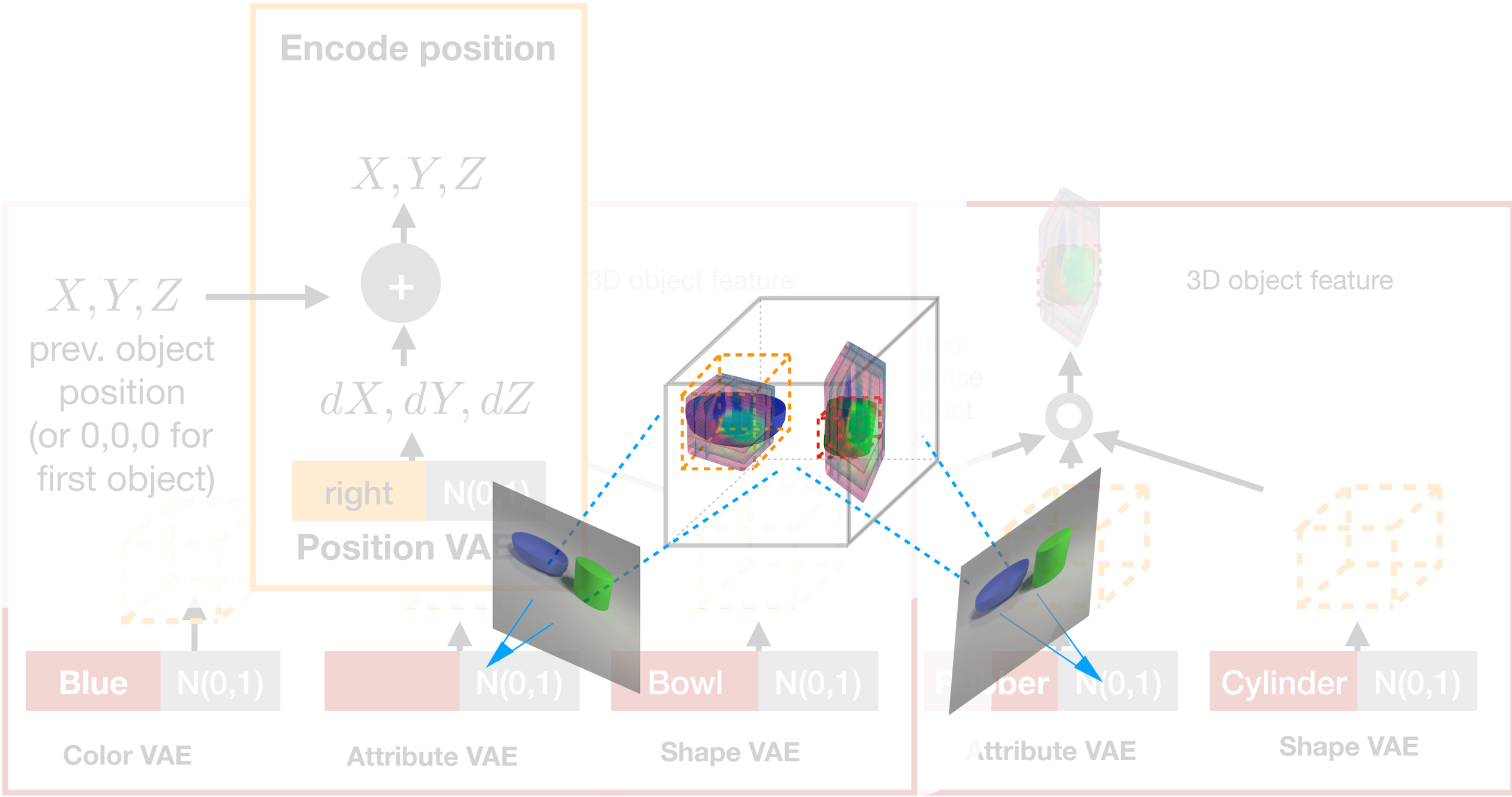
5. Our agent populates a 3D canvas with the predicted object tensors adn their relative offsets

“The green rubber cylinder is on the right of the blue bowl”



5. Our agent populates a 3D canvas with the predicted object tensors adn their relative offsets

"The green rubber cylinder is on the right of the blue bowl"



6. The generated canvas when projected should match the RGB image views

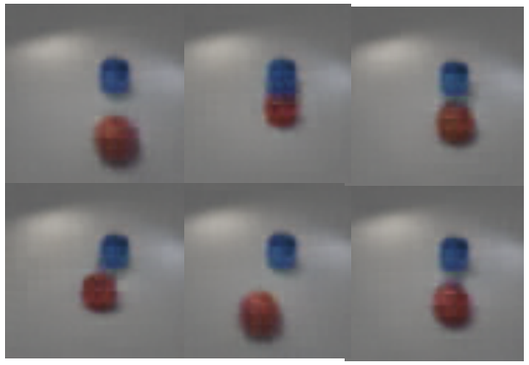
Multimodality in appearance

cylinder

sphere

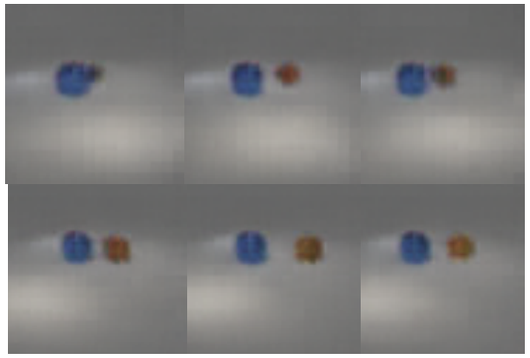
Multimodality in spatial arrangements

“red sphere **front left** of blue cylinder”



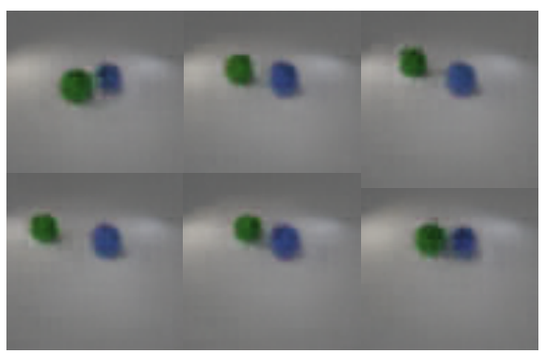
View Angle 1

View Angle 2

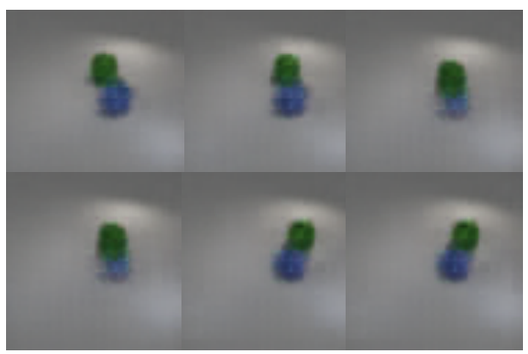


View Angle 3

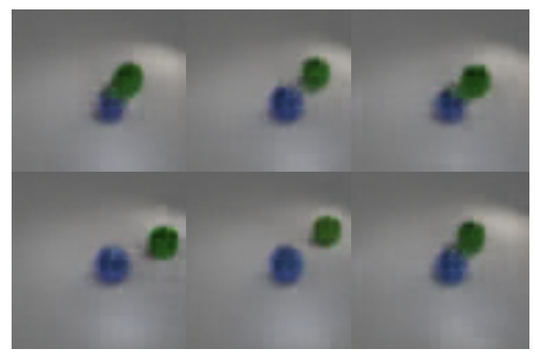
“green sphere t**o the left behind** of blue sphere”



View Angle 1



View Angle 2



View Angle 3

Scene imagination

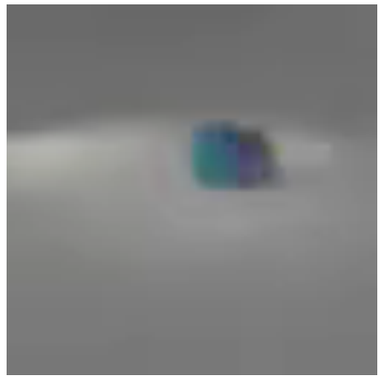
“cyan sphere to the left of red cube”

“red cylinder to the front of red sphere to the left-front of blue sphere”

“cyan cylinder to the left of red sphere to the front of green sphere”

“blue sphere to the left front of green cube”

“cyan cylinder to the front of yellow cube”



“cyan cylinder to the left front of yellow sphere to the behind of green sphere to the front of blue sphere to the front of gray cylinder to the behind of red sphere”

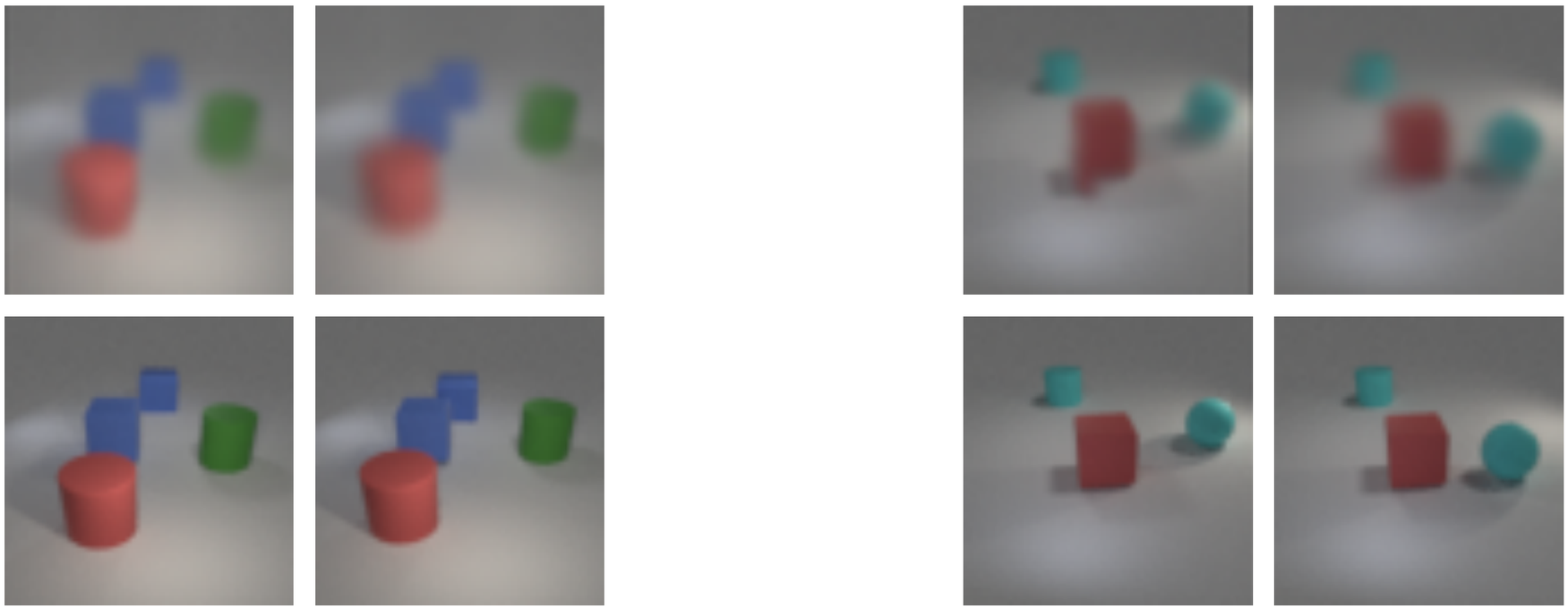
Scene imagination

“Red Rubber Cylinder to the left front of Blue Rubber Cube to the left front of Green Rubber Cylinder to right front of Blue Rubber Cube”

“Red Rubber Cube to the left front of the Blue Rubber Sphere to the right front of Cyan Metal Cylinder”

Neural rendering

Blender rendering



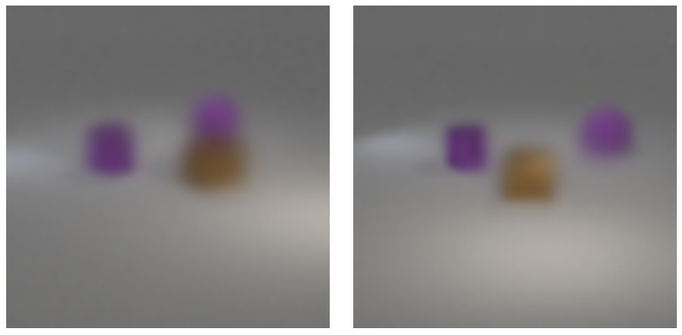
- **Neural rendering**: project the 3D feature maps using our learned project+RGB decoder neural module
- **Blender rendering**: use the object-centric 3D feature maps to retrieve nearest 3D mesh neighbors from a training set, then arrange the retrieved meshes based on predicted 3D spatial offsets

Scene imagination

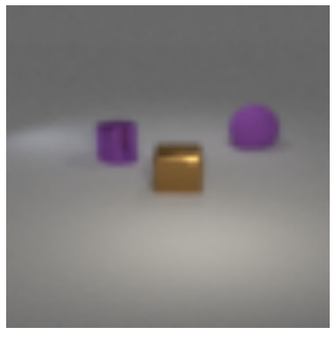
“Purple Cylinder to the left behind of Brown Cube to the left front of Purple Sphere”

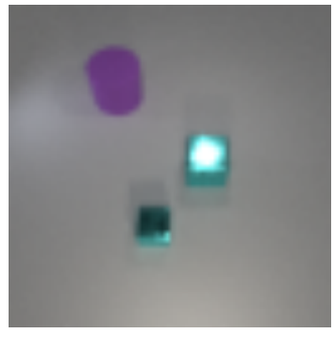
“Purple Cylinder to the left behind of Cyan Cube to the left front of Cyan Cube”

Neural rendering



Blender rendering





- **Neural rendering**: project the 3D feature maps using our learned project+RGB decoder neural module
- **Blender rendering**: use the object-centric 3D feature maps to retrieve nearest 3D mesh neighbors from a training set, then arrange the retrieved meshes based on predicted 3D spatial offsets

Grounding arbitrarily long utterances

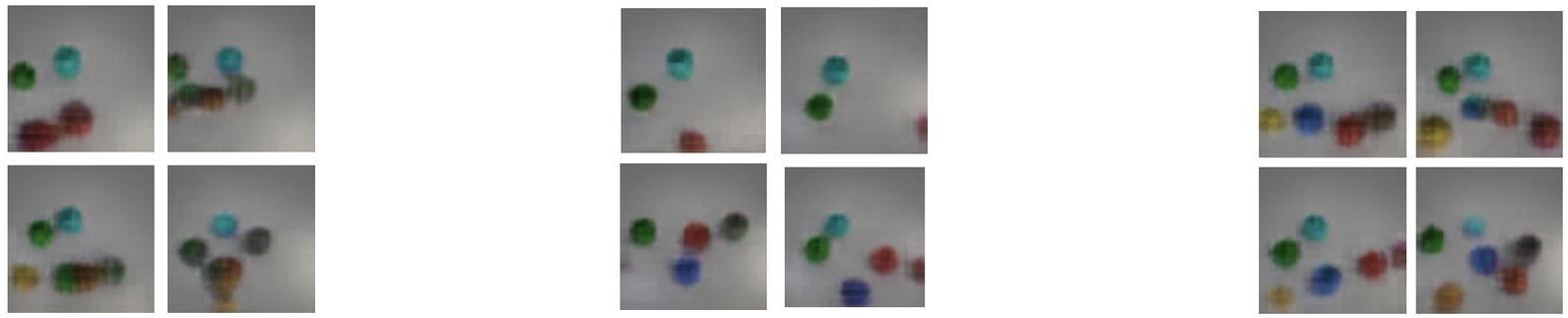
“yellow sphere to the left front of green sphere to the left behind of blue sphere to the left front of blue cylinder to the left behind of red cube to the left front of gray cube”

IOU > 0.1

Object Out of Camera View

IOU= 0

Top View

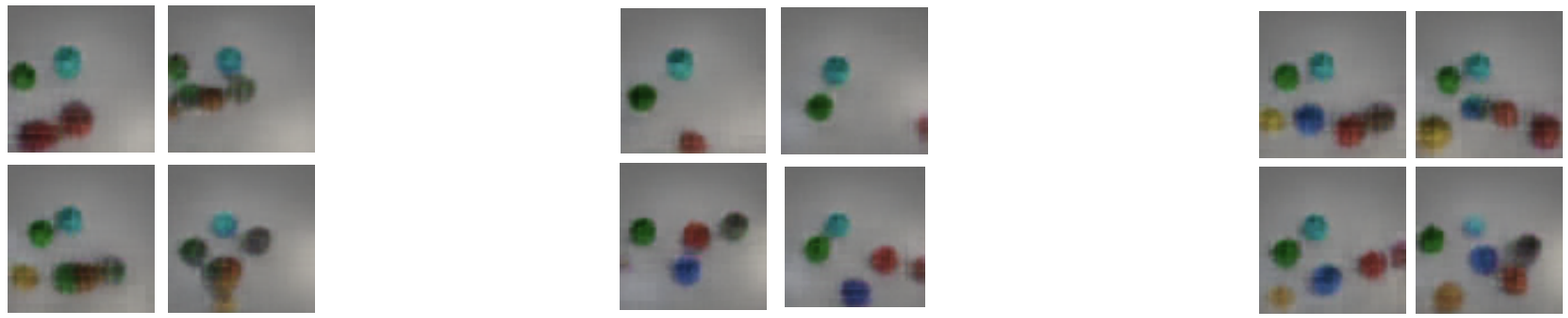


Grounding arbitrarily long utterances

“yellow sphere to the left front of green sphere to the left behind of blue sphere to the left front of blue cylinder to the left behind of red cube to the left front of gray cube”

IOU > 0.1 | Object Out of Camera View | IOU= 0

Top View



“gray sphere to the left front of blue sphere to the left front of red sphere to the left behind of cyan sphere to the left behind of green sphere”

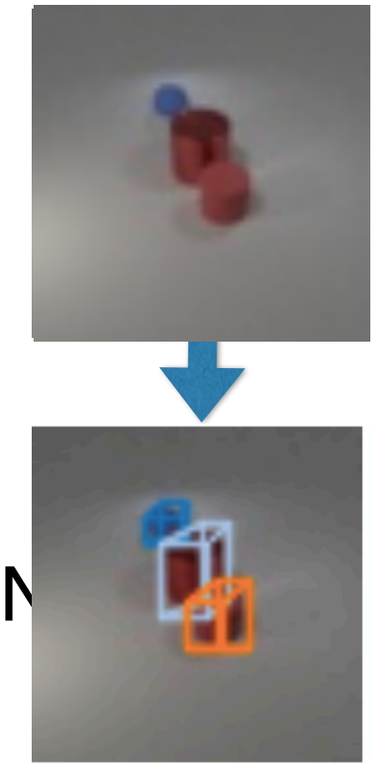
IOU > 0.1 | Object Out of Camera View | IOU= 0

Top View

3D referential object detection

3D referential object detection

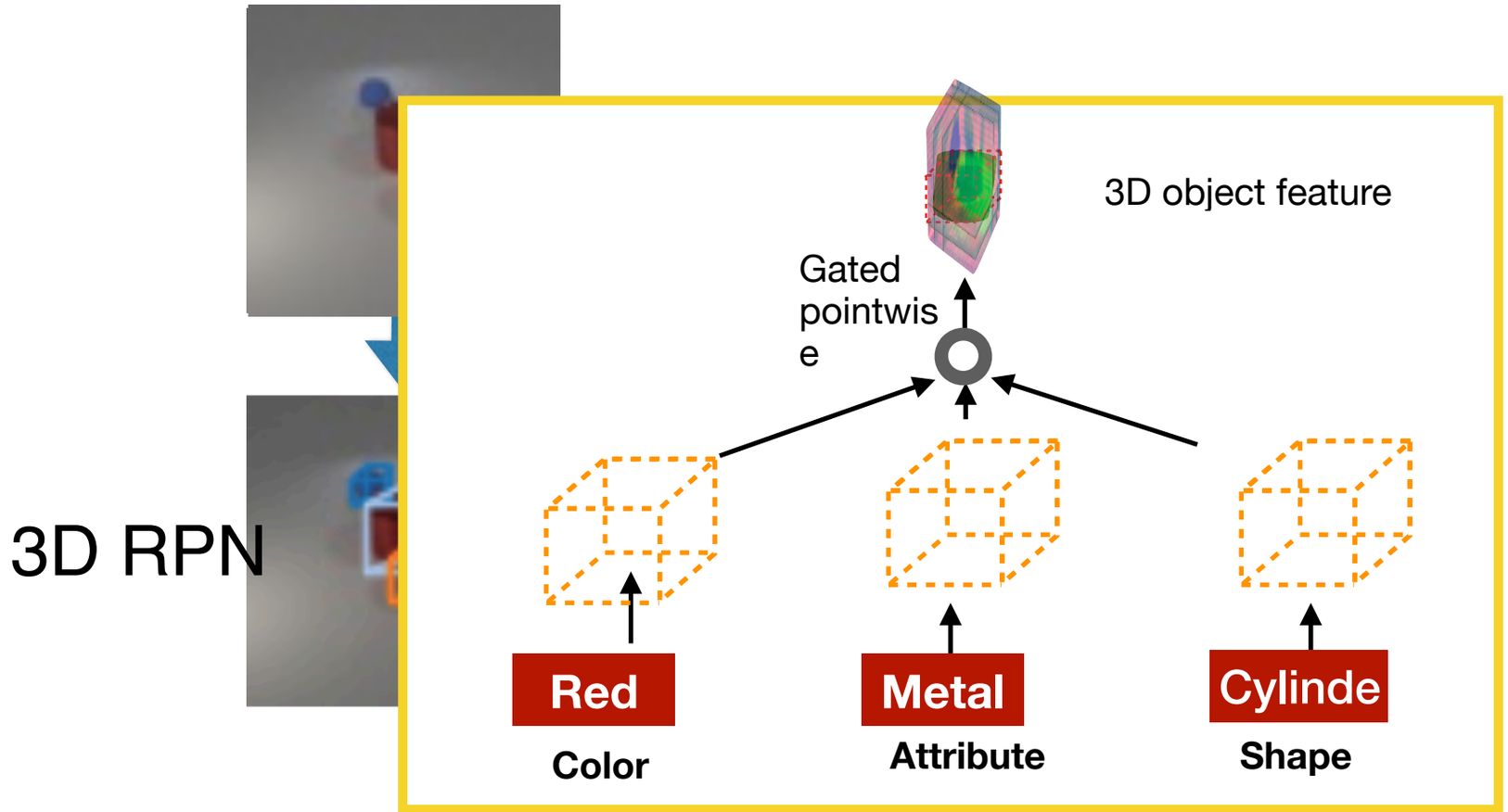
query *"find red metal cylinder to the left behind of red rubber cylinder"*



3D RPN

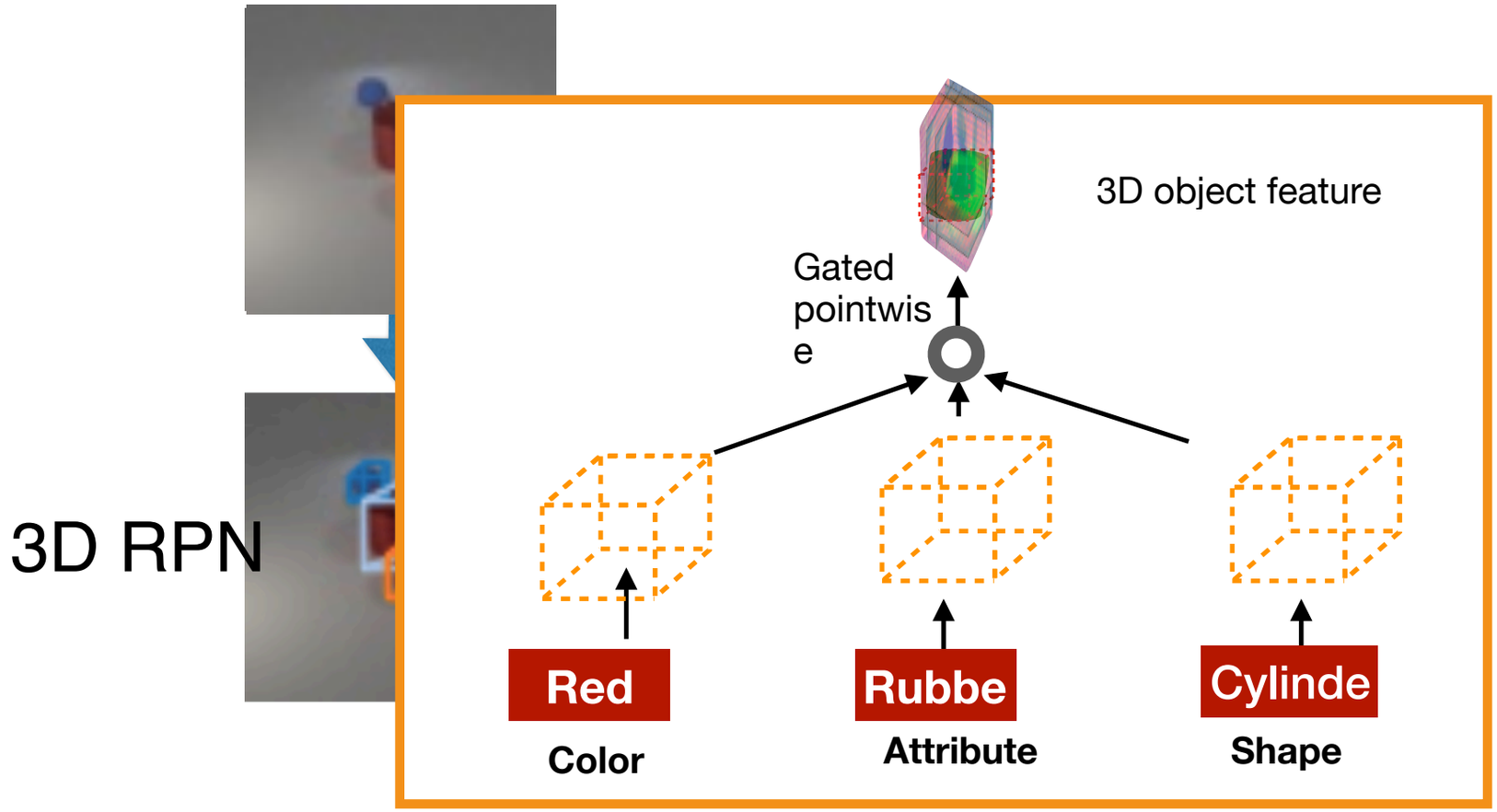
3D referential object detection

query *"find red metal cylinder to the left behind of red rubber cylinder"*

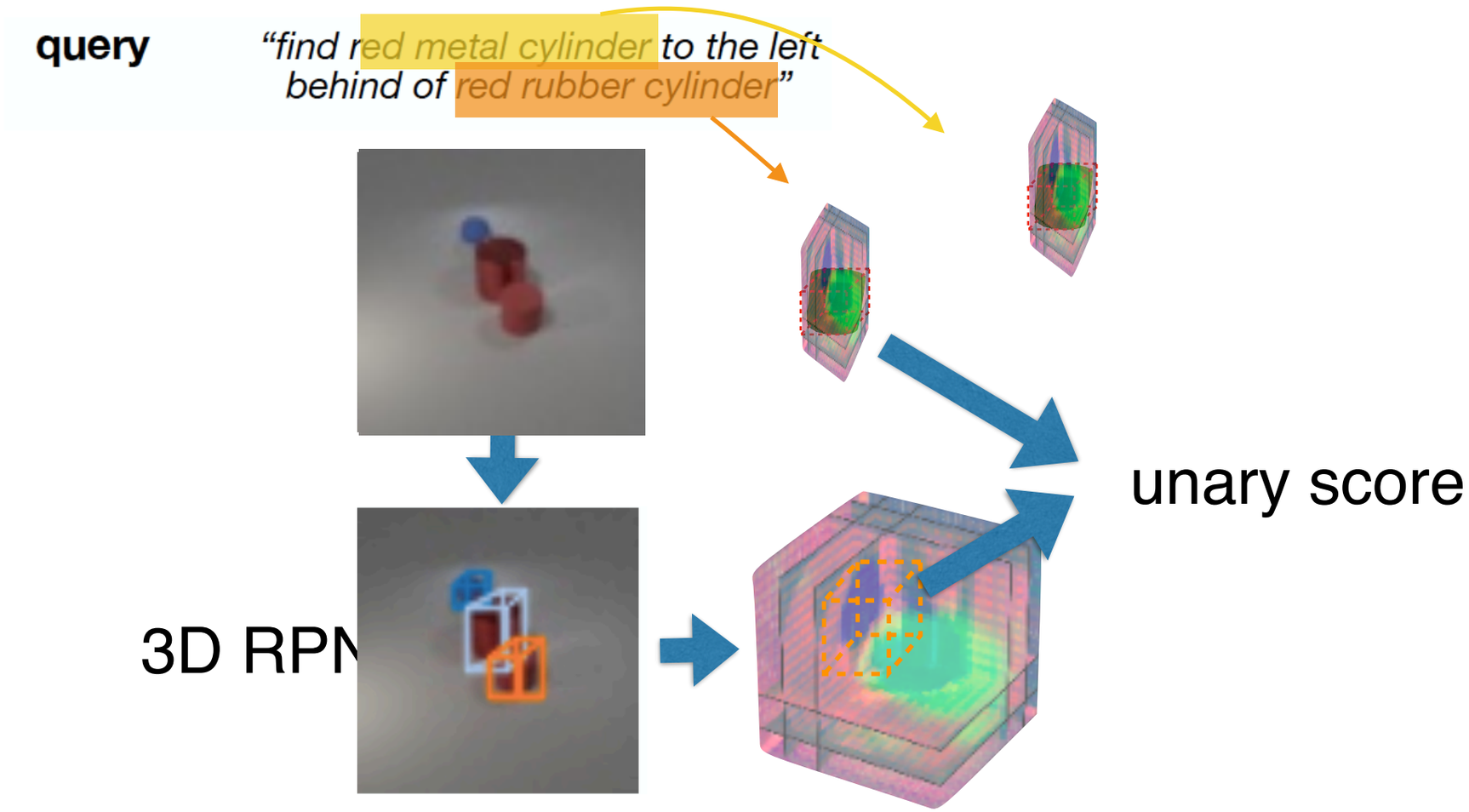


3D referential object detection

query *"find red metal cylinder to the left behind of red rubber cylinder"*

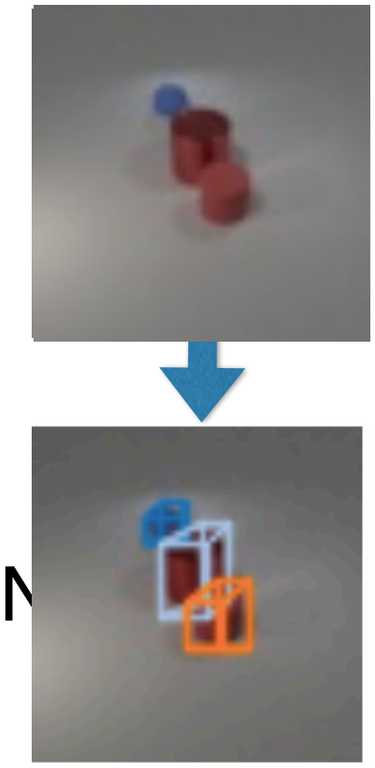


3D referential object detection

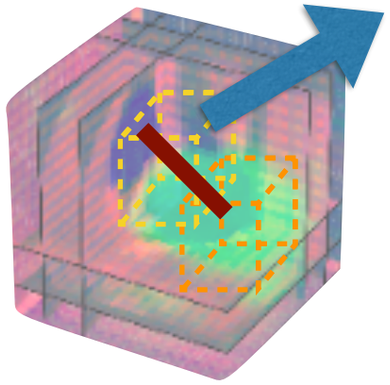


3D referential object detection

query *"find red metal cylinder to the left behind of red rubber cylinder"*



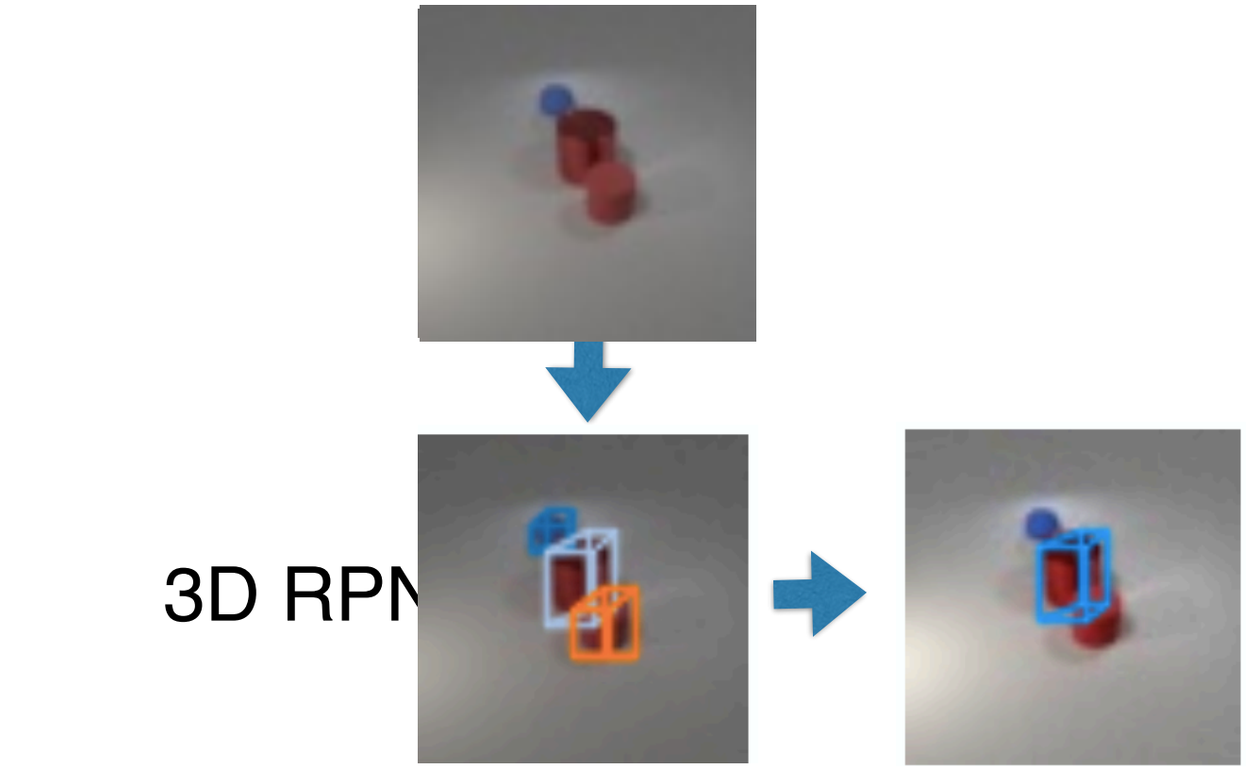
3D RPN



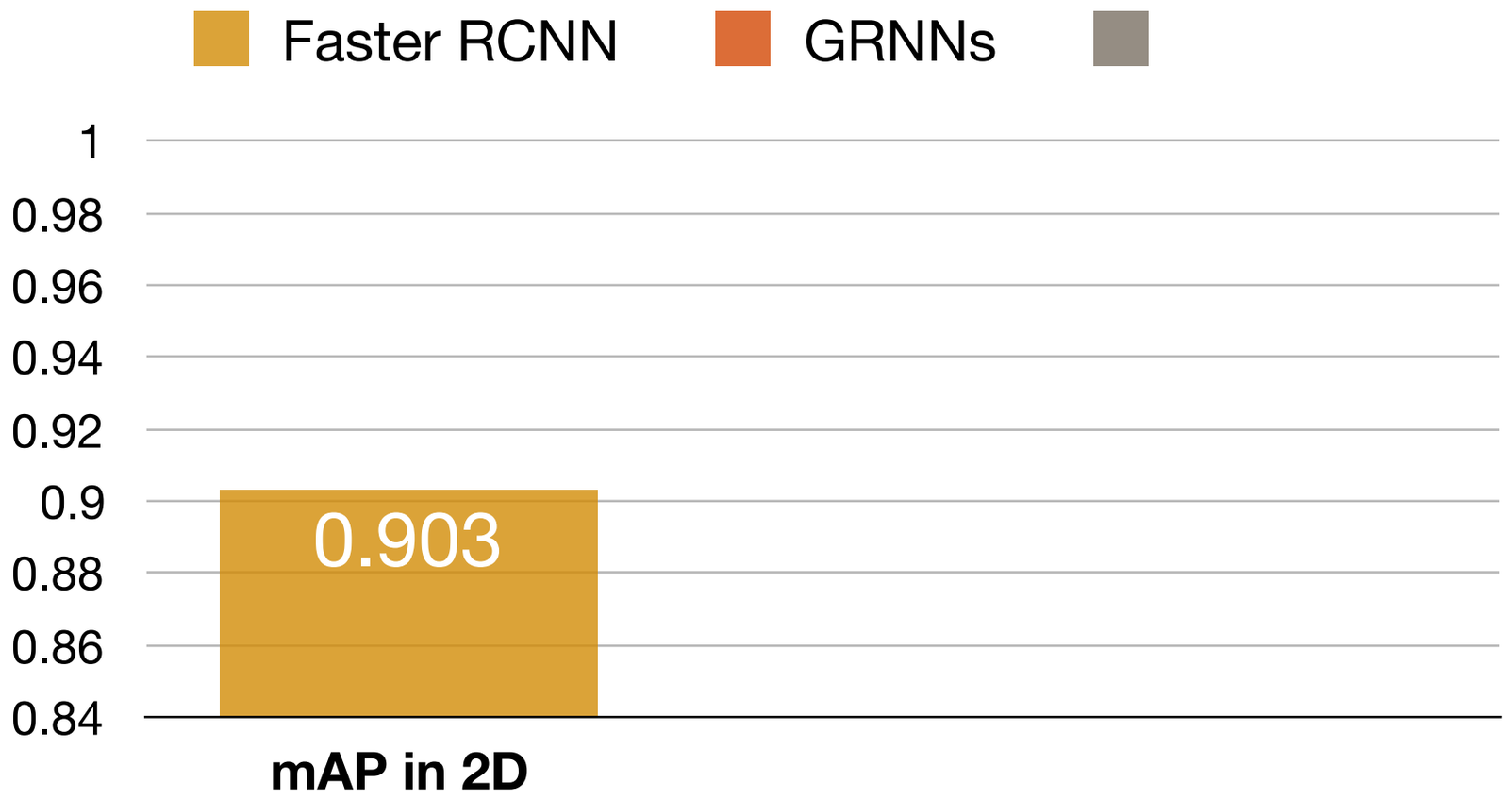
Spatial score

3D referential object detection

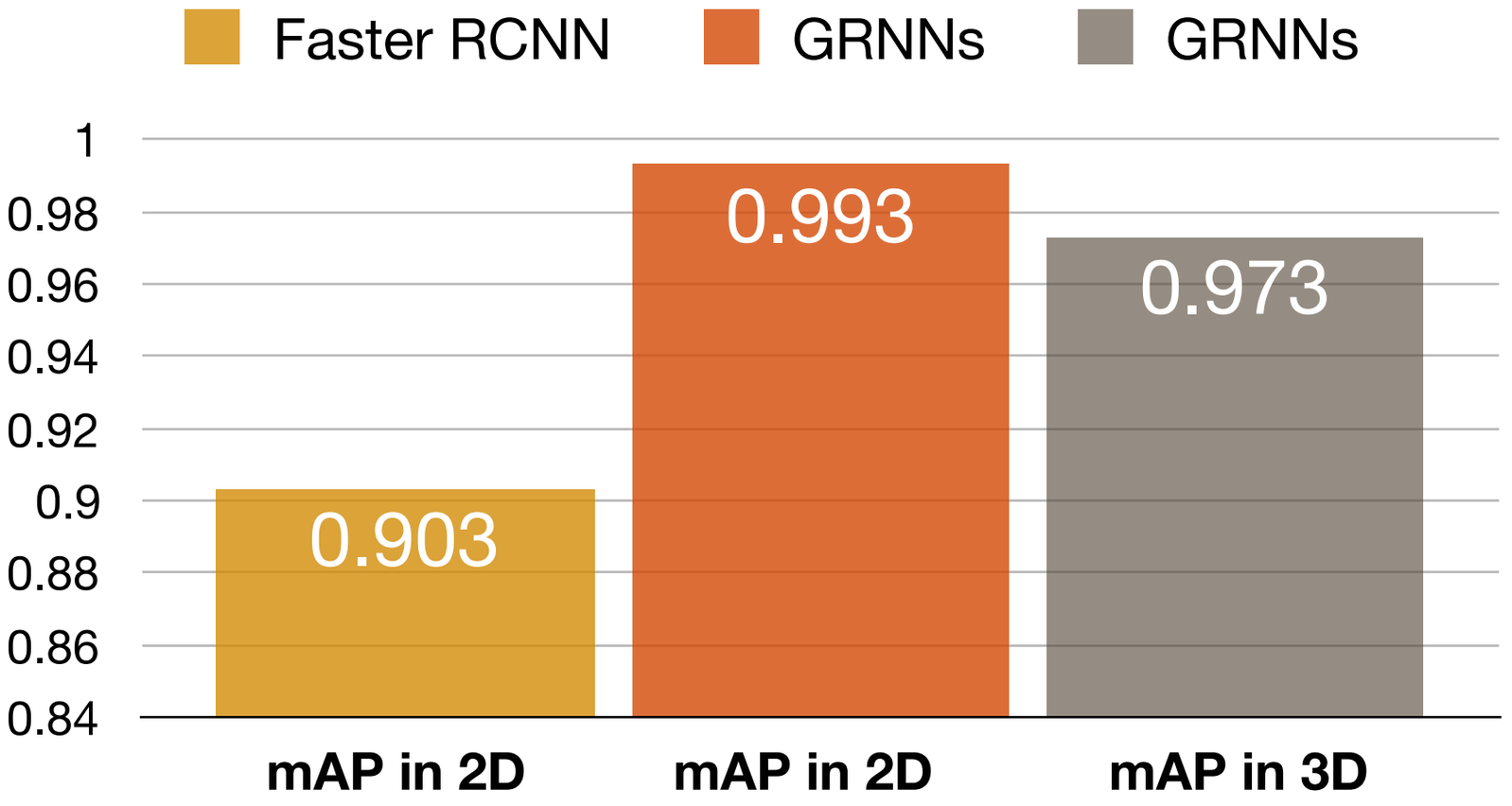
query *"find red metal cylinder to the left behind of red rubber cylinder"*



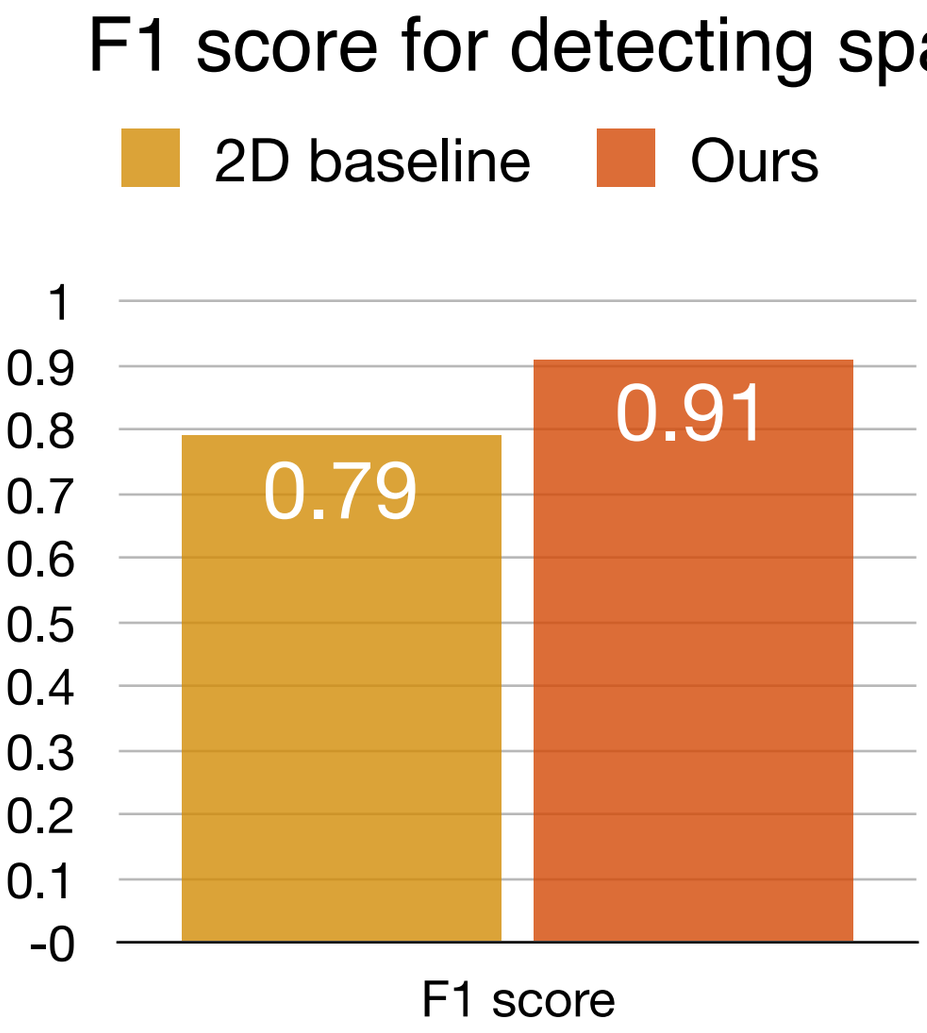
3D referential object detection

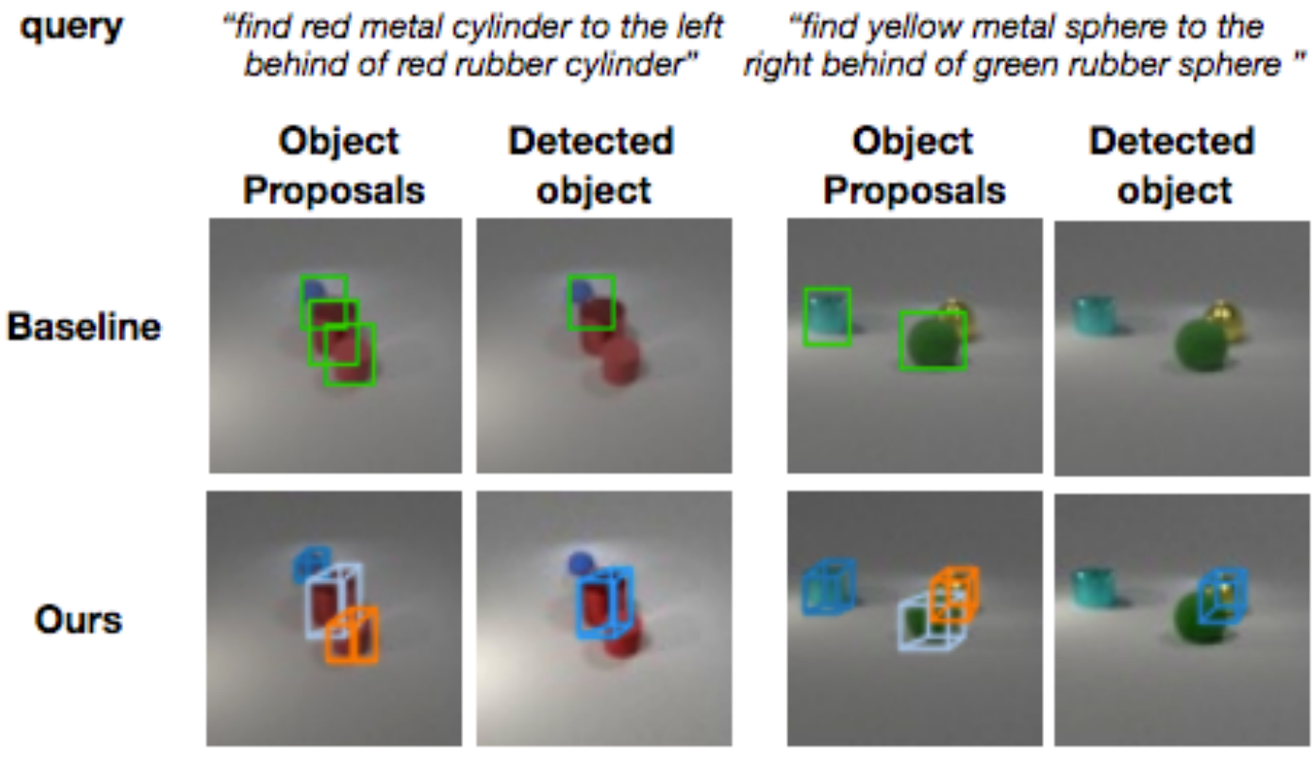


Object region proposals



3D referential object detection

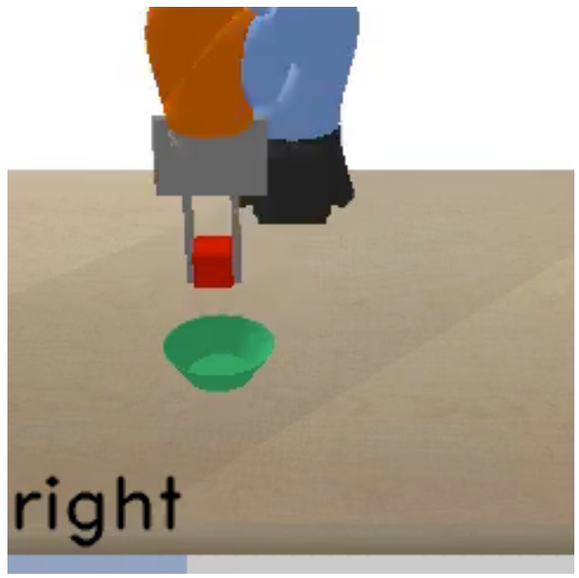




Ronghang Hu et al. Modeling relationships in referential expressions with compositional modular networks

Instruction Following

``*put the cube on the right of the bowl”*



1. Referential 3D object detection
2. Goal generation: Predict relative 3D desired location for the object
3. Use LQR with Euclidean distance of current to goal location as the cost.

Conclusion / Conjecture

Embodiment and egomotion-stable perception is the problem and the solution to visual recognition, common sense learning and learning behaviours from visual streams

Conclusion

`` We must perceive in order to move, but we must also move in order to perceive”

James J. Gibson

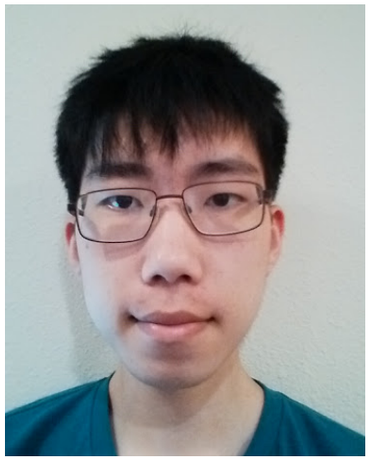
Conclusion

“If we figure out the right way to do 3D perception, no one will use 2D again, the same way when color TV was invented no one used black and white”

Yaser Sheikh

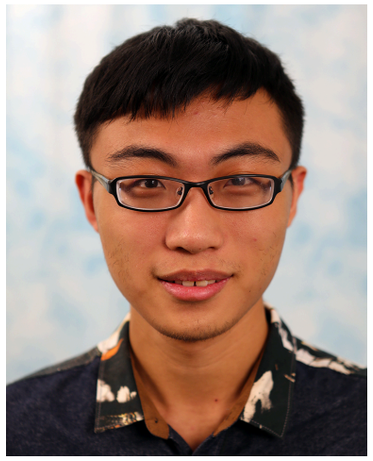
Thank you!

Fish Tung



Ricson Chen

Adam Harley

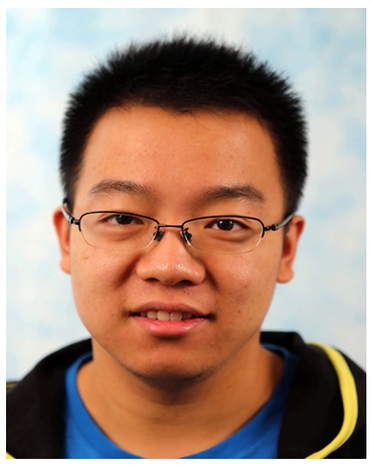


Fangyu Li

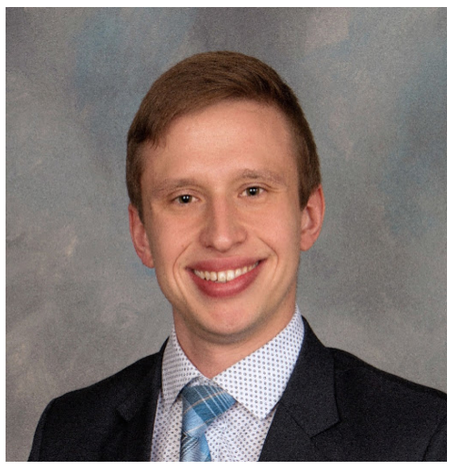
Shrinidhi K. Lakshmikanth

Mihir Prabhudesai

Syed Javed



Xian Zhou



Max Sieb

- **Learning spatial common sense with geometry-aware recurrent networks**, Tung et al., CVPR 2019,
- **Visual Representation Learning with 3D View-Contrastive Inverse Graphics Networks**, Harley et al., arxiv
- **Embodied Language Grounding with Implicit 3D Visual Feature Representations,** Prabhudesai et al., arxiv