Few-shot Object Reasoning for Robot Instruction Following

Yoav Artzi

Workshop on Spatial Language Understanding
EMNLP 2020

Cornell CIS Computer Science | CORNELL TECH

Task

- Navigation between landmarks
- Agent: quadcopter drone
- Inputs: poses, raw RGB camera images, and natural language instructions

Task

go straight and stop before reaching the planter
turn left towards the globe and go forward until just before it

Mapping Instructions to Control

- The drone maintains a **configuration** of target velocities

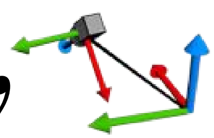
Linear forward velocity   Angular yaw rate

$$(v, \omega)$$

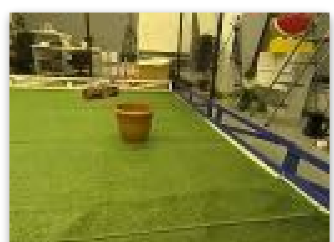
- Each action updates the configuration or stops

- Goal: learn a mapping from inputs to configuration updates

$f($ *go straight and stop before reaching the planter turn left globe …*,



,



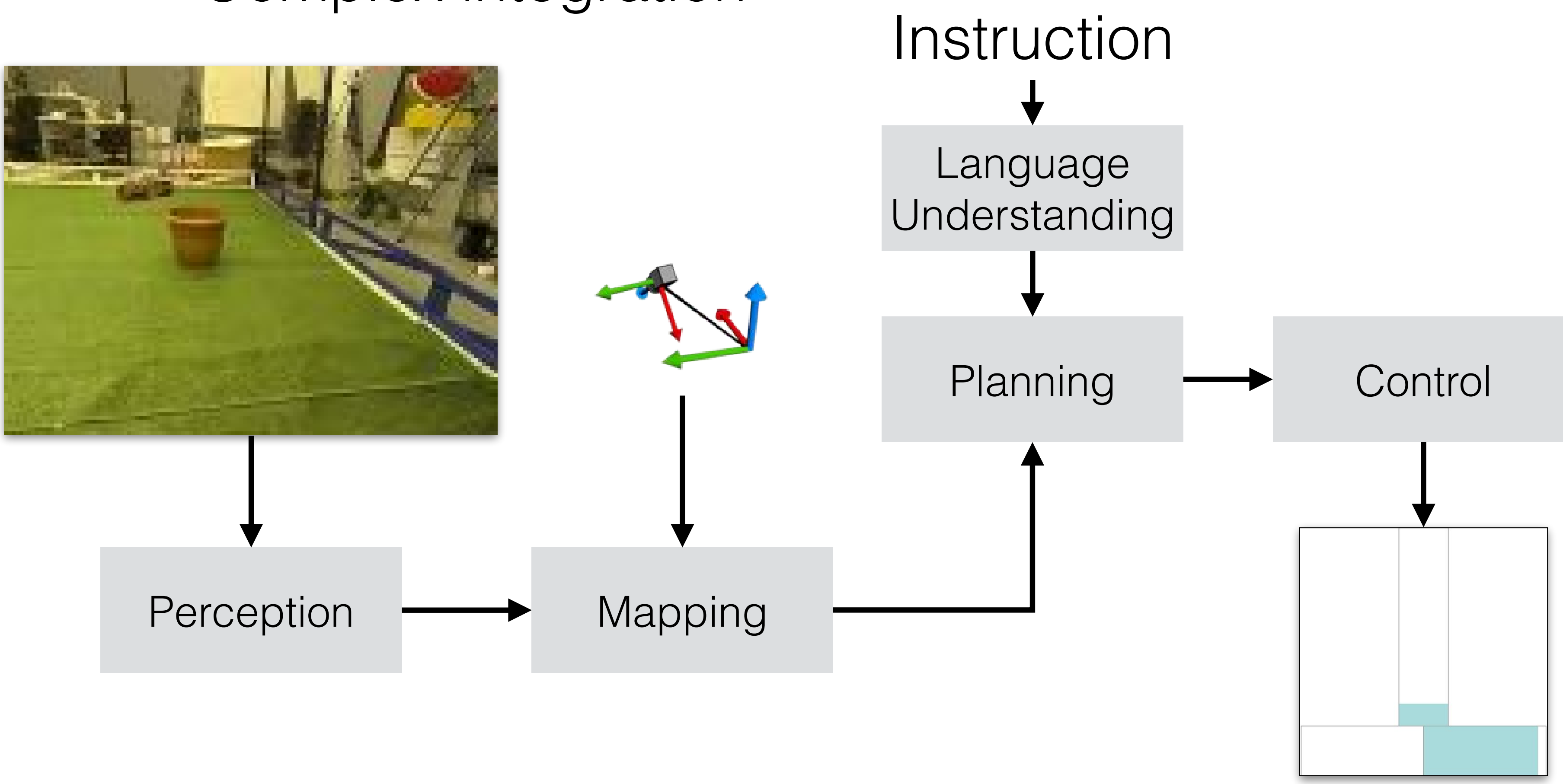
$) = STOP$

v_t

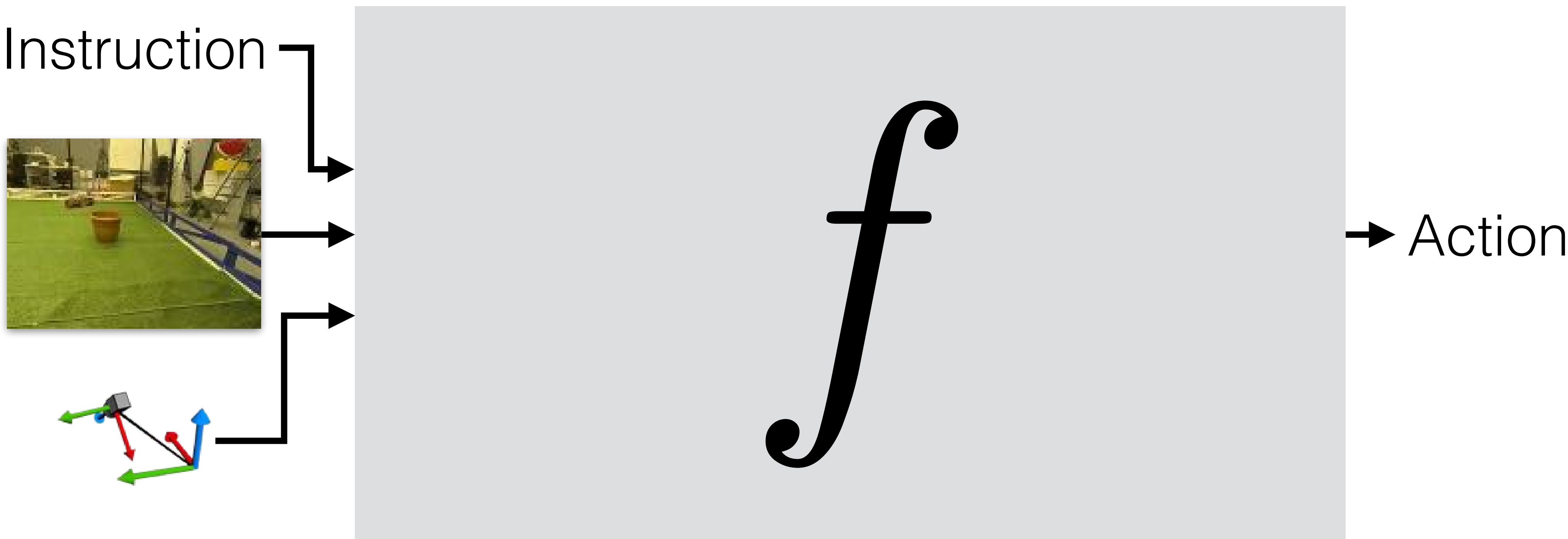
ω_t

Modular Approach

- Build/train separate components
- Symbolic meaning representation
- Complex integration



Single-model Approach (a.k.a end-to-end)



How to think of extensibility, interpretability, and modularity when packing everything in a single model?

Single-model Approach

- Extensibility: extending the model to reason about new object after training
- Interpretability: viewing how the model reasons about object grounding and trajectories
- Modularity: re-using parts of the model

Within a representation learning framework

Representation: Design vs. Learning

- Systems that use symbolic representations are interpretable and (potentially) extensible
- However: representation design of every possible concept is brittle and hard to scale
- Instead: design the most general concepts and let representation learning fill them with content
- Today, two concepts: objects and trajectories

Today

Few-shot instruction following:

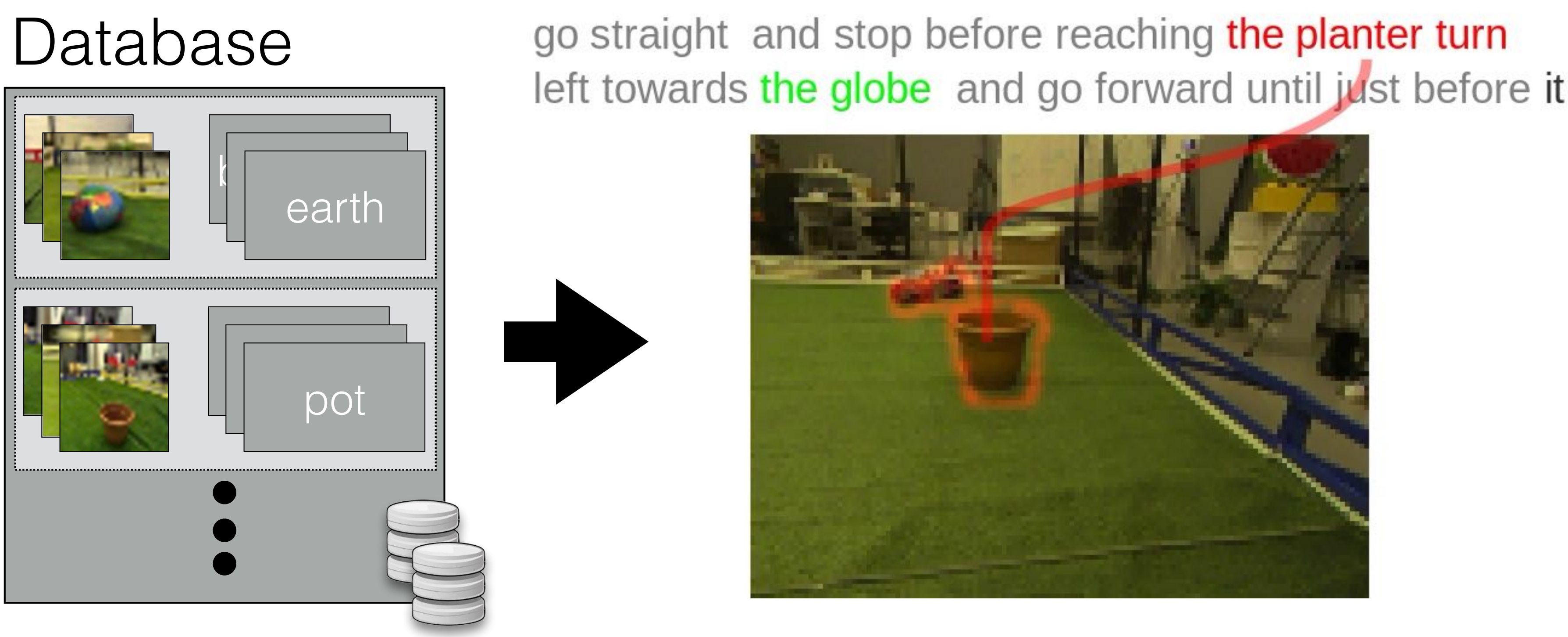
- Few-shot language-conditioned object segmentation
- Object context mapping
- Integration into a visitation-prediction policy for mapping instructions to drone control

Language-conditioned Object Segmentation

- Input: instruction and observation images
- Goal: identify and align objects and references

Few-shot Version

- Input: instruction, observation images, and database
- Goal: identify previously unseen objects and mentions and align them

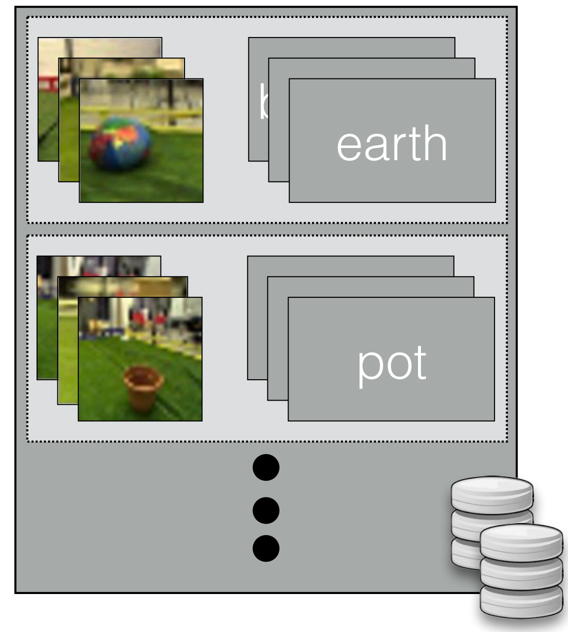


Alignment via a Database

- Approach: align observations and references through the database

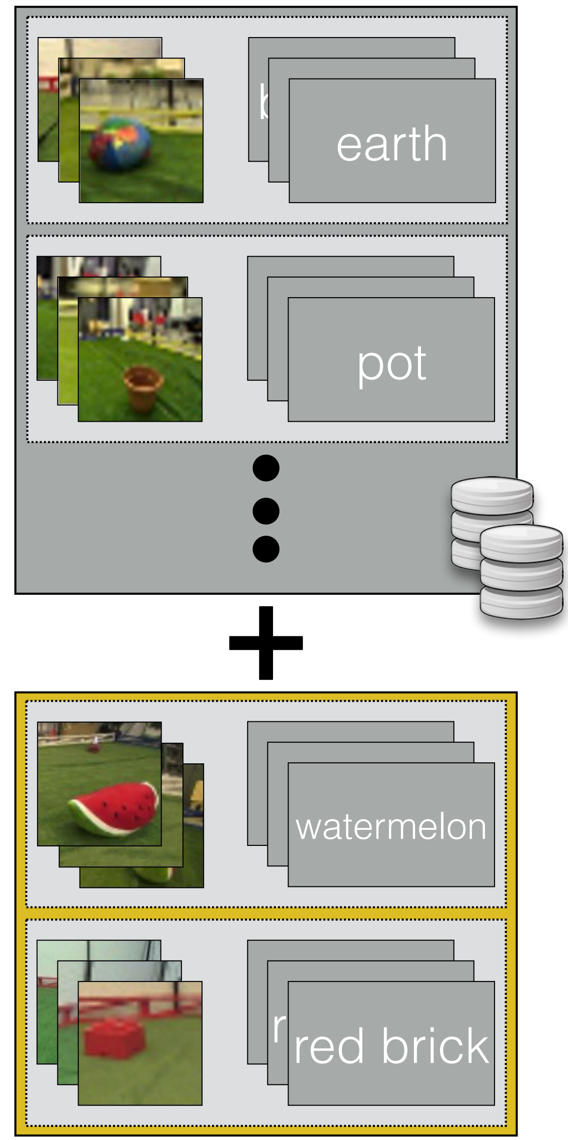
- Adding objects to the database extends the alignment ability

- Requires only adding a few image and language exemplars

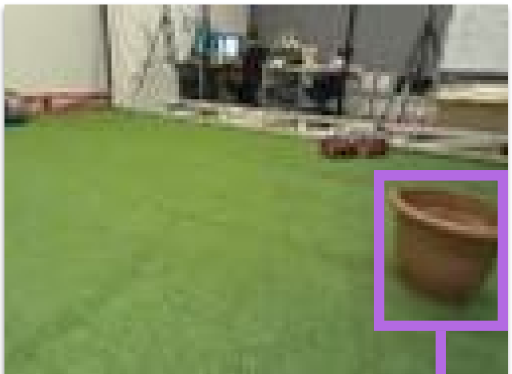


Alignment via a Database

- Approach: align observations and references through the database
- Adding objects to the database extends the alignment ability
- Requires only adding a few image and language exemplars



Alignment Score



go straight and stop before reaching the planter turn left towards the globe and go forward until just before it

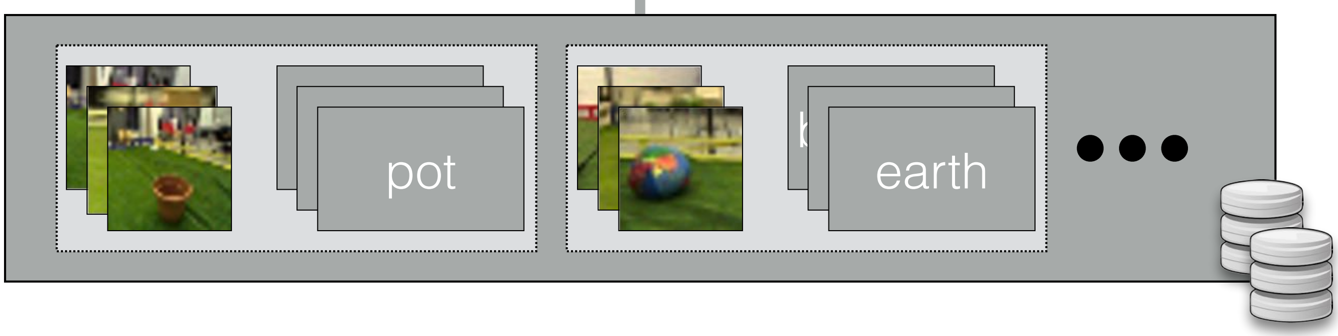
Reference

Bounding box

$$\textsc{Align}(b, r) = \sum_{o} P(b \mid o)\, P(o \mid r)$$

Object record

Database

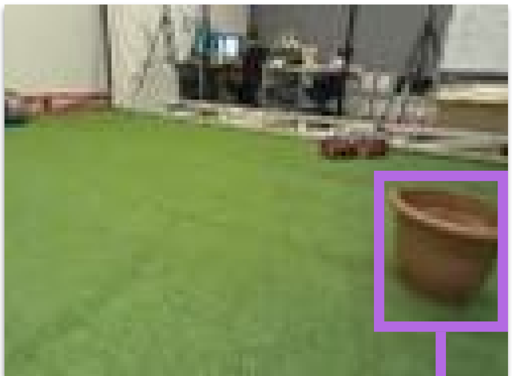


b Bounding box

r Reference

o Database object

Alignment Score



go straight and stop before reaching the planter turn left towards the globe and go forward until just before it

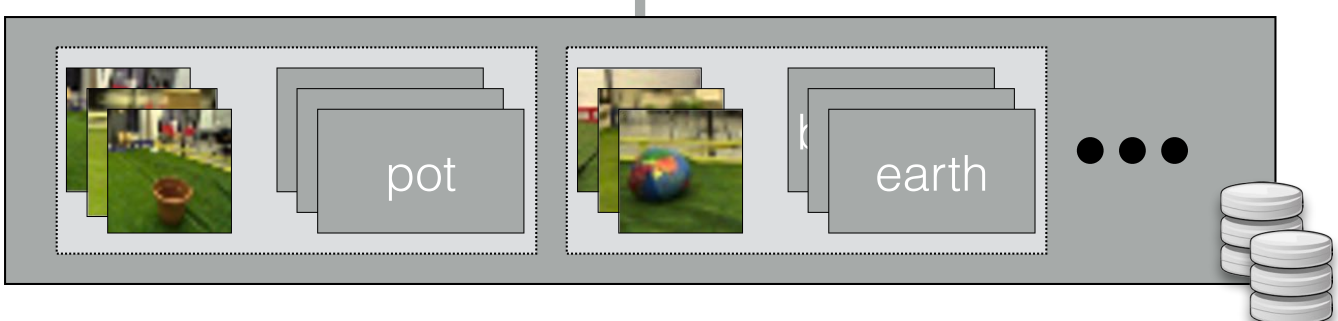
Reference

Bounding box

$$\textsc{Align}(b, r) = \sum_{o} \frac{P(o \mid b)\,P(b)\,P(o \mid r)}{P(o)}$$

Object record

Database



b	Bounding box
r	Reference
o	Database object

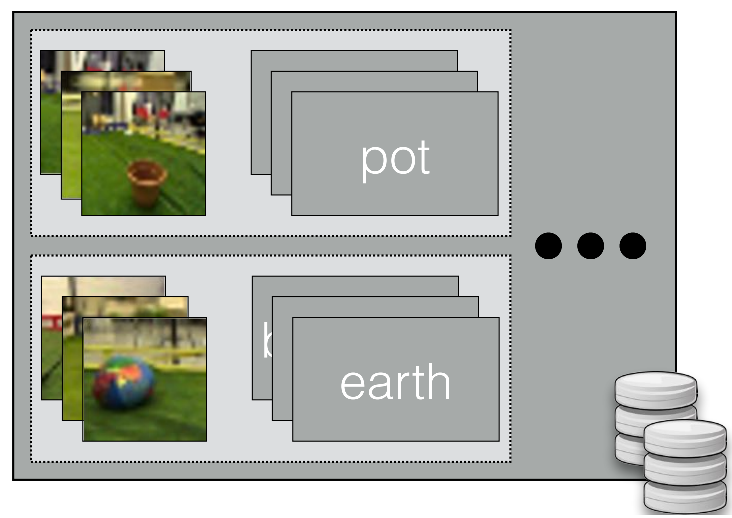
Alignment Score

$$\textsc{Align}(b, r) = \sum_{o} \frac{P(o \mid b)P(b)P(o \mid r)}{P(o)}$$

- Region proposal network gives bounding boxes and $P(b)$

- $P(o)$ is uniform

b	Bounding box
r	Reference
o	Database object

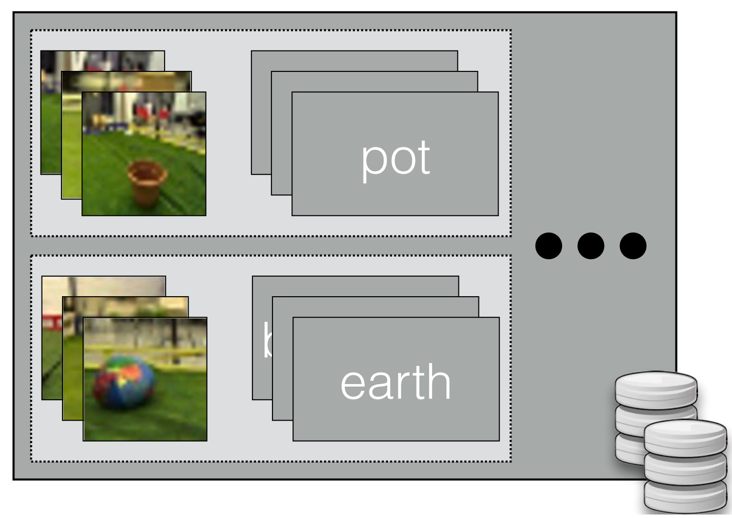


Alignment Score

$$\textsc{Align}(b, r) = \sum_{o} \frac{P(o \mid b)P(b)P(o \mid r)}{P(o)}$$

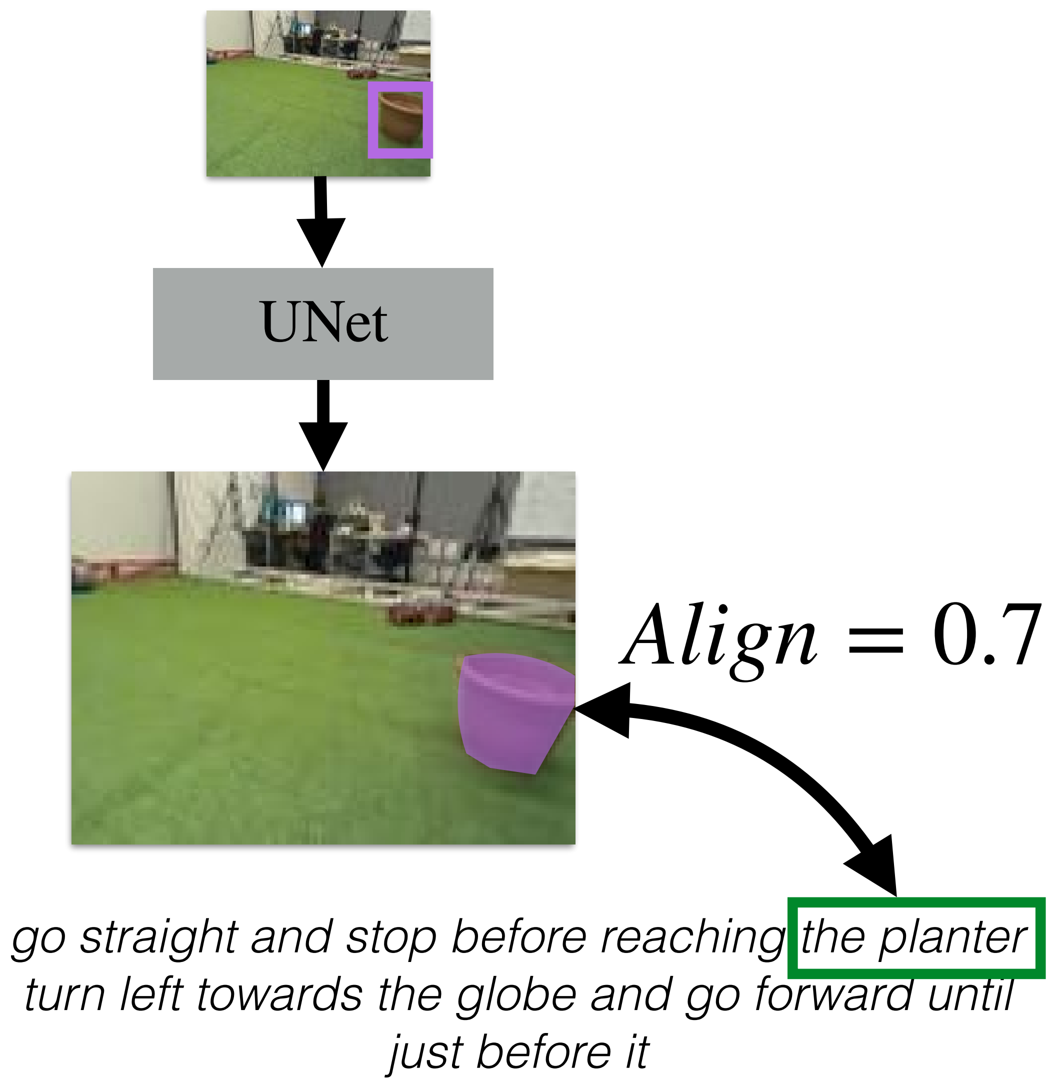
- $P(o \mid b)$ is computed using visual similarity
- Using Kernel Density Estimation with a symmetric multivariate Gaussian kernel
- $P(o \mid r)$ is computed similarly using text similarity with pre-trained embeddings

b	Bounding box
r	Reference
o	Database object



Mask Refinement

- Refine each bounding box with a UNet model
- Gives a tight object mask
- Paired with a bounded alignment score to a reference in the text



Learning

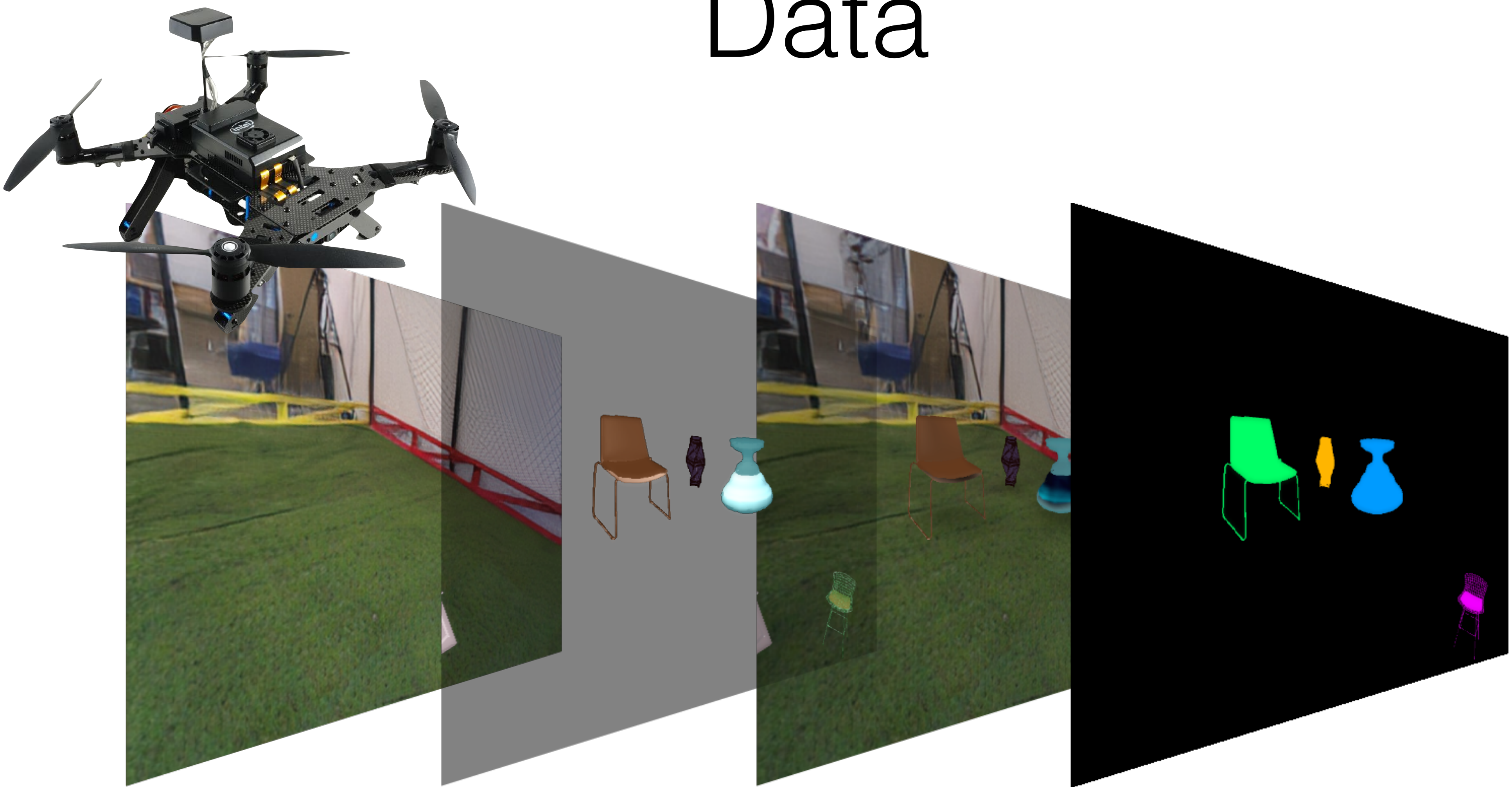
$$\textsc{Align}(b, r) = \sum_{o} \frac{P(o \mid b)P(b)P(o \mid r)}{P(o)}$$

UNet

- Region proposal network parameters for bounding box proposal
- Image similarity measure for $P(o \mid b)$
- UNet parameters for mask refinement
- Text similarity uses pre-trained embeddings
- Challenge: need large-scale heavily annotated visual data

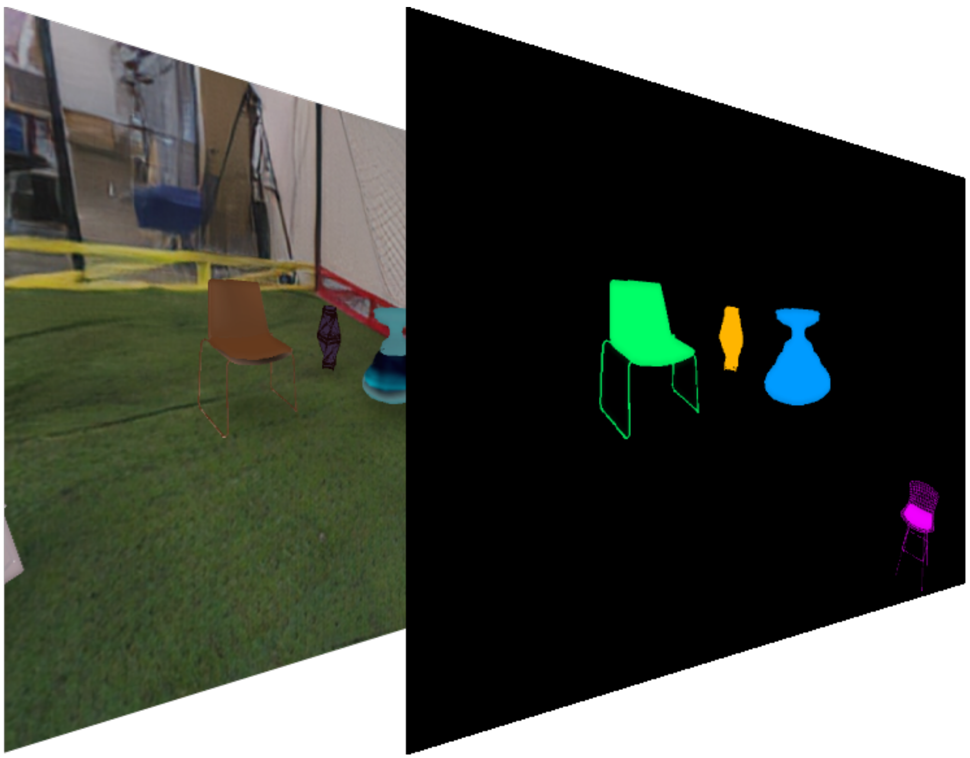
b	Bounding box
r	Reference
o	Database object

Augmented Reality Training Data



Augmented Reality Training Data

Large-scale generation with ShapeNet objects



Composite Mask labels

Learned representations generalize beyond specific objects for:

- Region proposal network for bounding boxes
- Image similarity measure for $P(o \mid b)$
- UNet parameters for mask refinement

Today

Few-shot instruction following:

- Few-shot language-conditioned object segmentation
- Object context mapping
- Integration into a visitation-prediction policy for mapping instructions to drone control

Object Context Mapping

Goal: create maps that capture object location and the instruction behavior around objects

1. Identify and align object mentions to observations

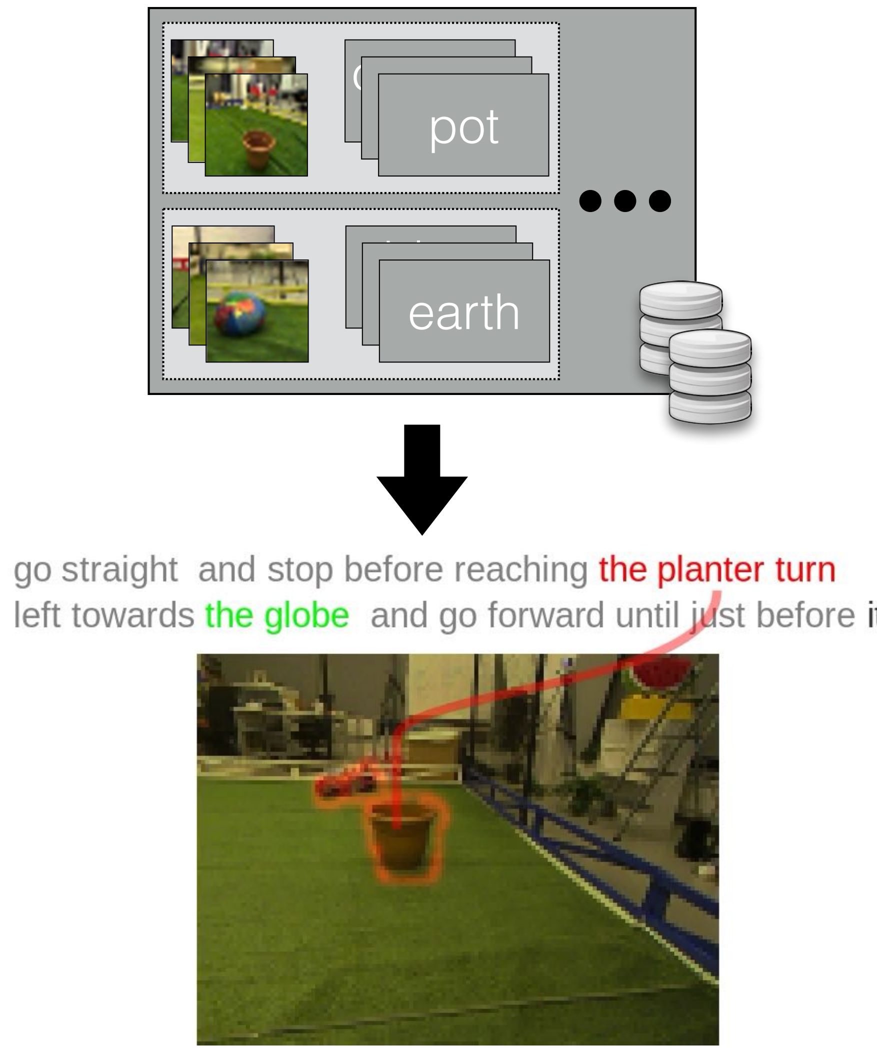
2. Compute abstract contextual representations for object references

3. Project and aggregate masks over time

4. Combine aggregated masks with contextual representations to create a map

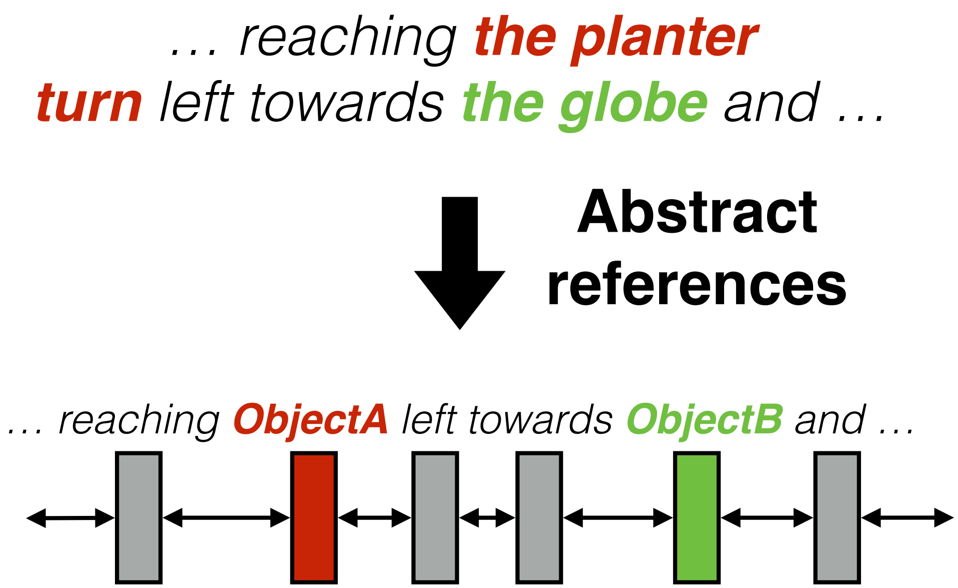
Object Context Mapping
Step I: Identify and Align

- Bounding box proposals from Region Proposal Network
- Object references from tagger
- Align with language-conditioned segmentation and the database
- To compute: first-person masks aligned to instruction references



Object Context Mapping
Step II: Abstract Contextual Representations

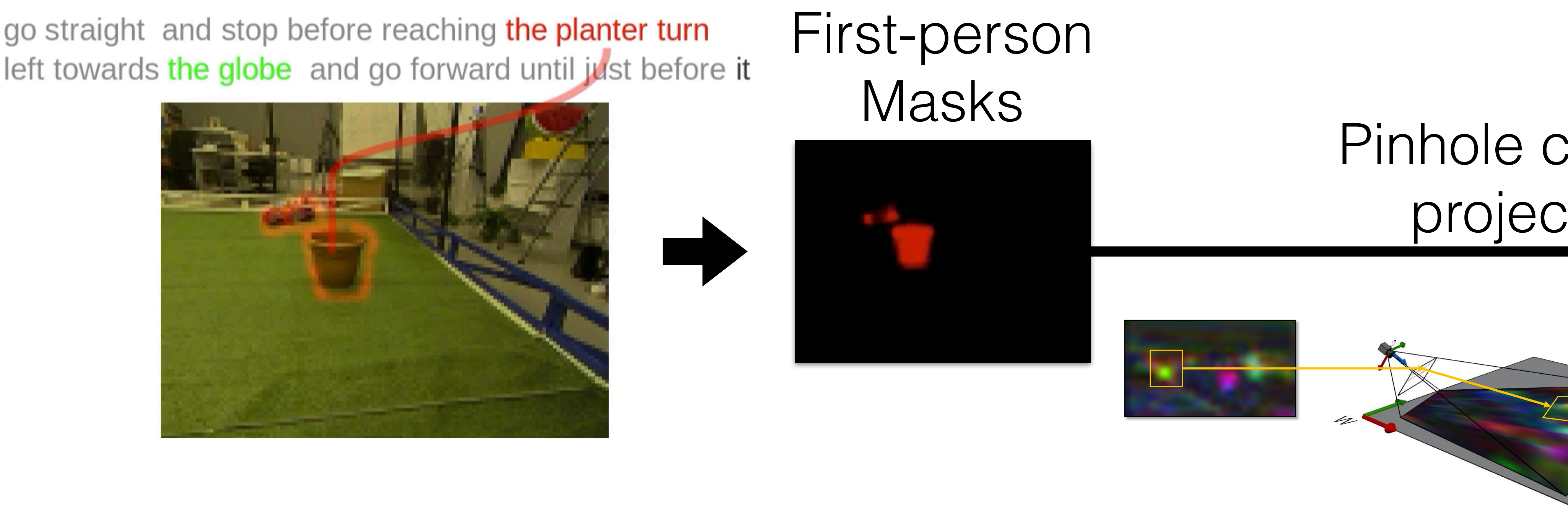
- Replace references with object placeholders
- Compute bi-directional RNN representations for all tokens
- The hidden state for each placeholder is the **object context representation**



Object Context Mapping
Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model

- Deterministic aggregation

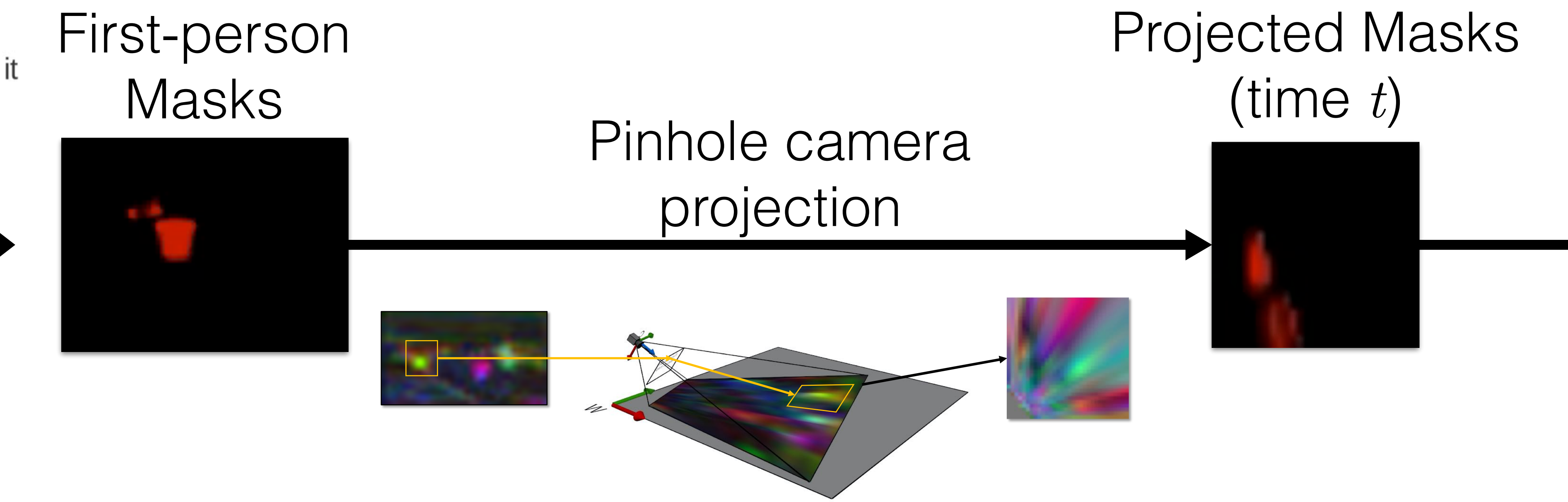


Object Context Mapping

Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model

- Deterministic aggregation

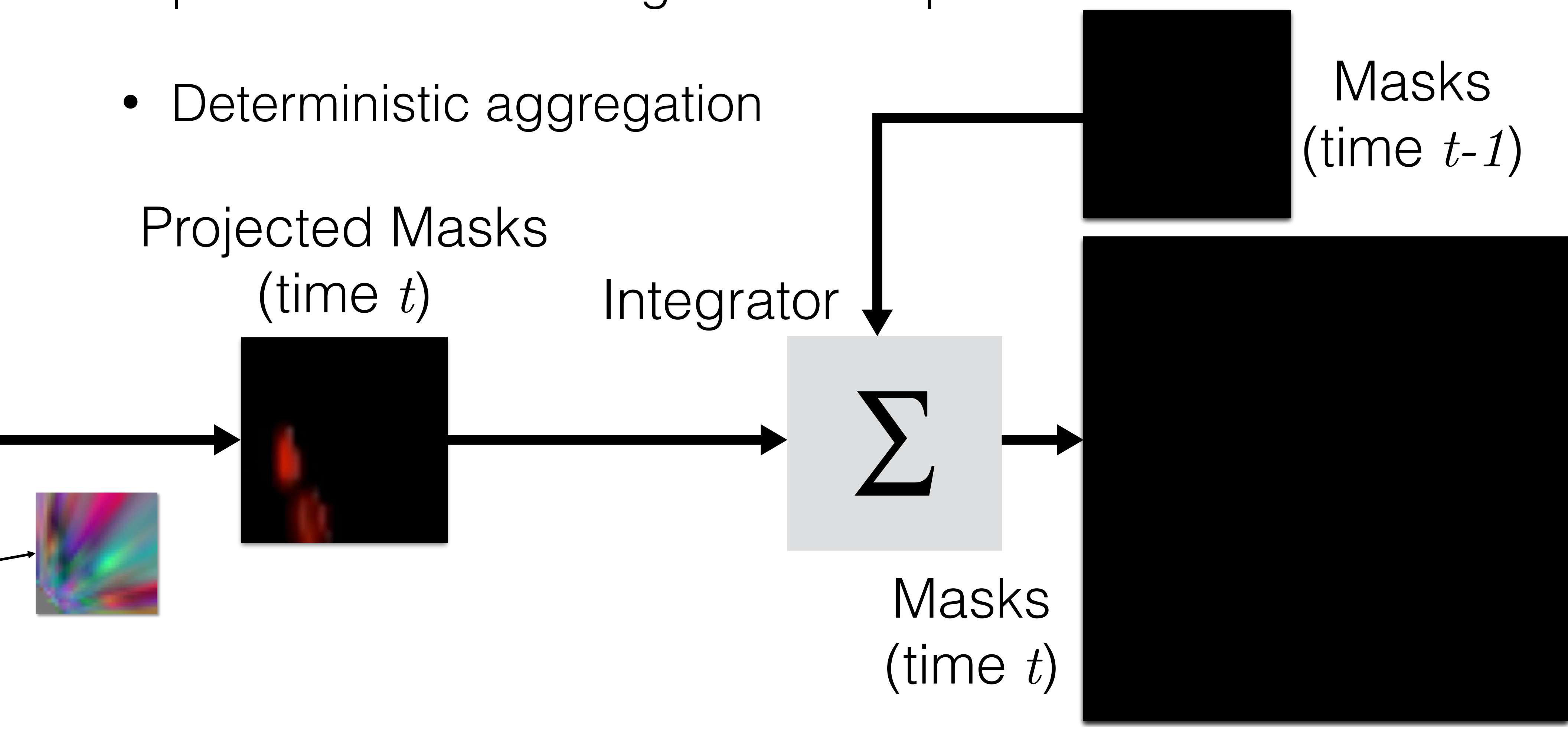


Object Context Mapping

Step III: Projection and Aggregation

- Projection from first-person camera masks to third-person environment ground with pinhole camera model

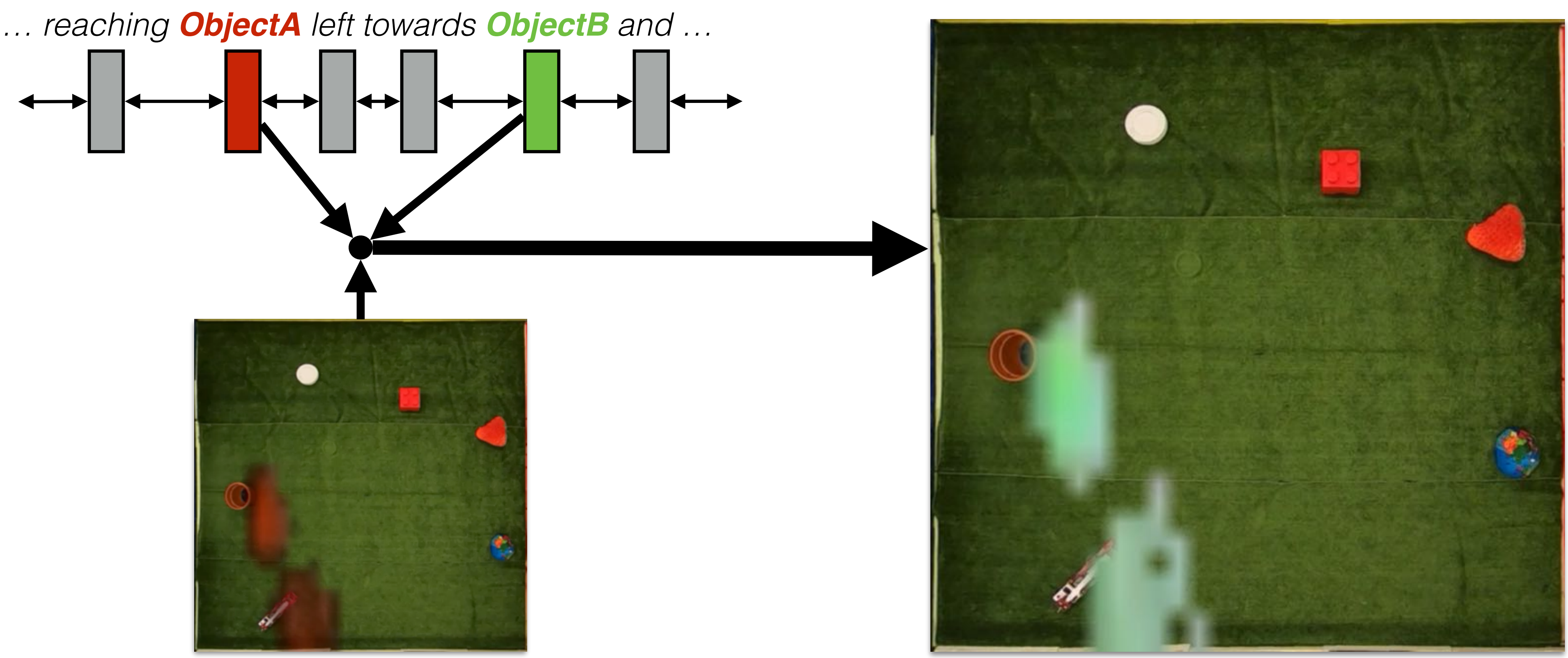
- Deterministic aggregation



Object Context Mapping

Step IV: Combine Object Rpresentations

- Each position is a product of a mask value and its aligned object context representation



Object Context Map

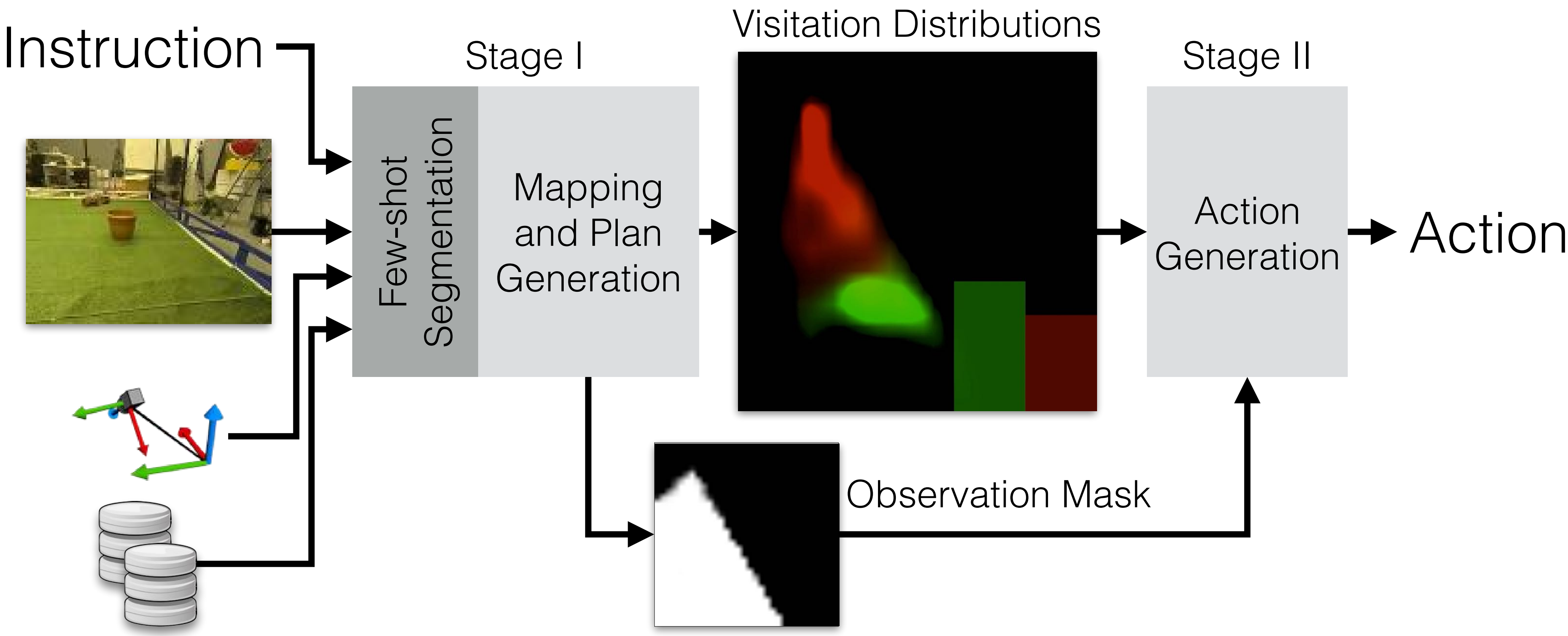
- Map information abstracts over reference content stripped from instruction
- Includes for each object the context of its reference in the instruction
- Tells the agent how to behave around the object
- Policy remains blind to the object itself

Today

Few-shot instruction following:

- Few-shot language-conditioned object segmentation
- Object context mapping
- Integration into a visitation-prediction policy for mapping instructions to drone control

Two-stage Policy



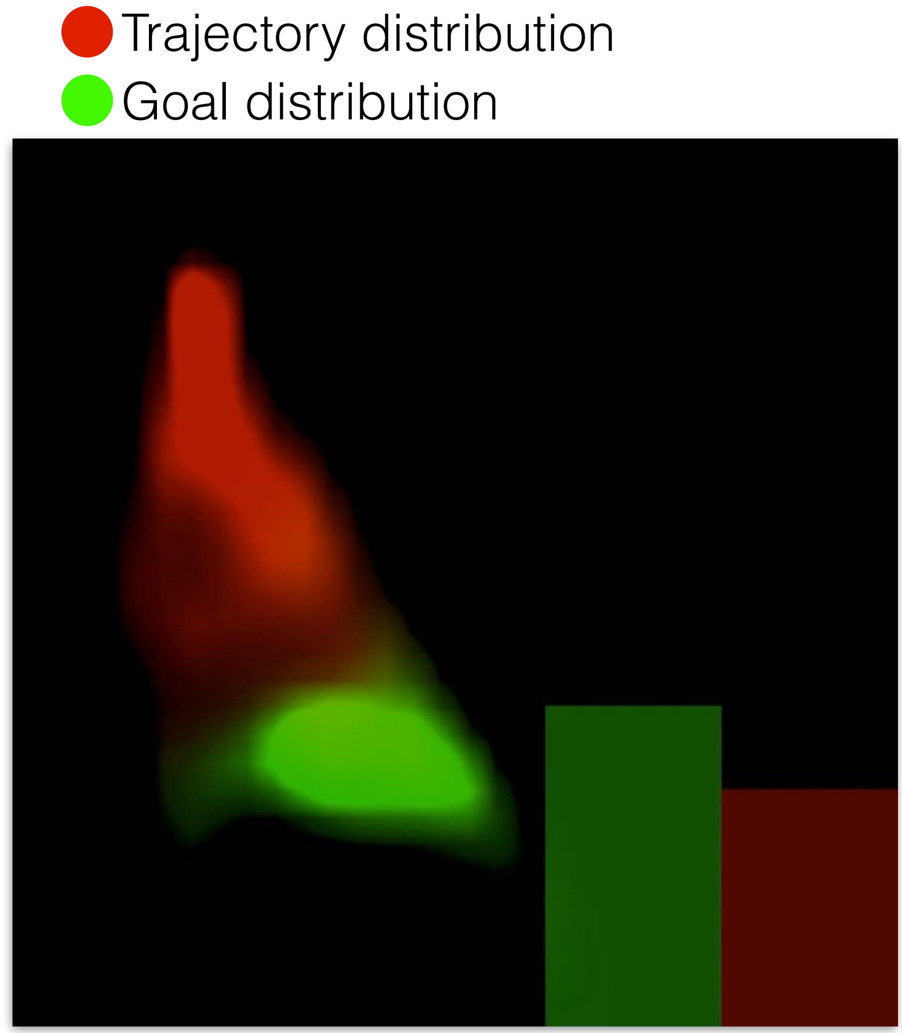
1. Map and predict states likely to visit + track observability

2. Generate actions to visit high-probability states and explore

Visitation Distributions

- The state-visitation distribution $d(s; \pi, s_0)$ is the probability of visiting state s following policy π from start state s_0
- Predicting $d(s; \pi^*, s_0)$ for an expert policy π^* tells us the states to visit to complete the task
- We compute two distributions: **trajectory-visitation** and **goal-visitation**

Visitation Distributions

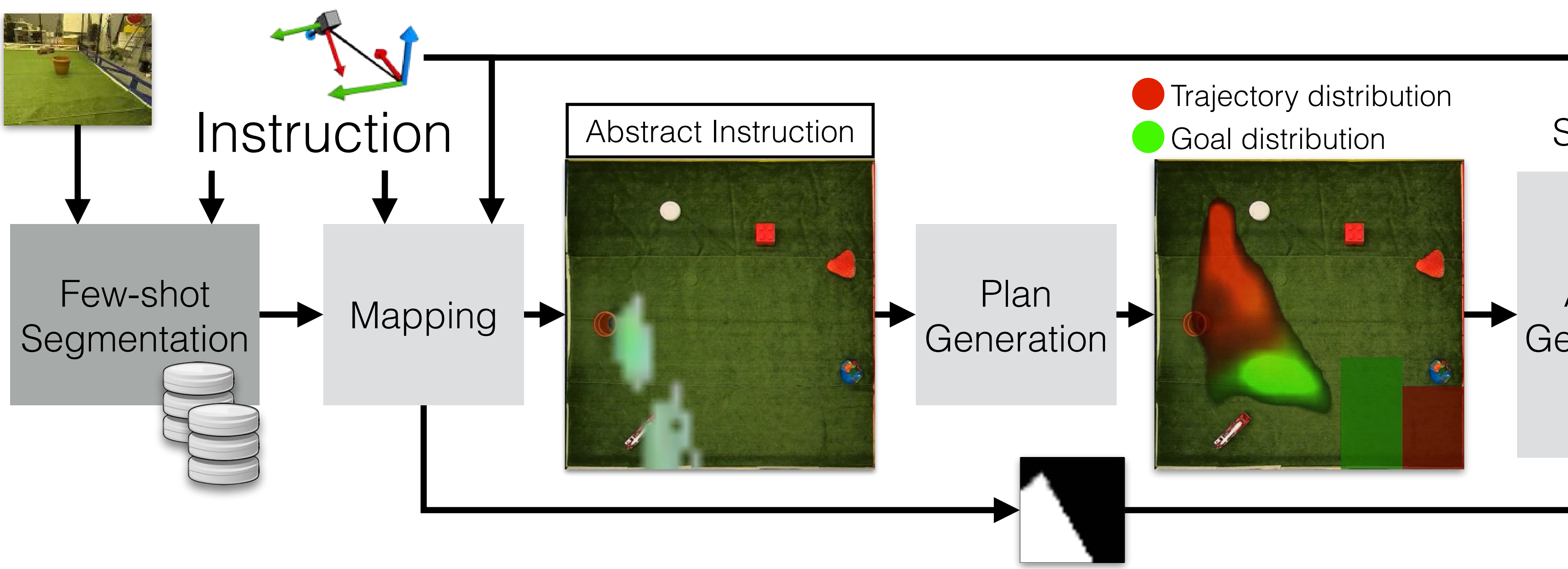


- Distributions reflect the agent plan
- Model path and goal observability
- Refined as observing more of the environment

Stage I: Mapping and Plan Generation

- Few-shot language-conditioned segmentation to construct an object context map

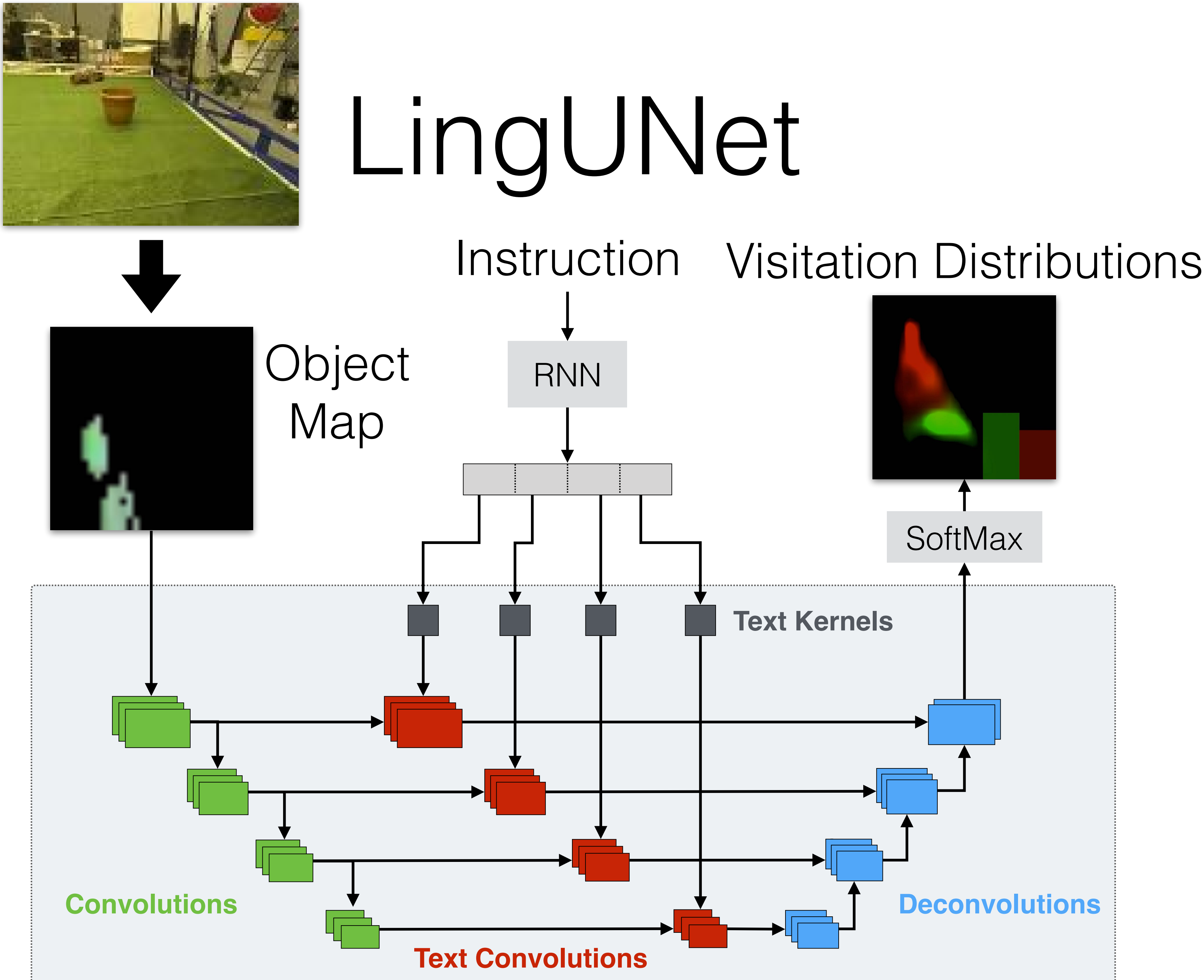
- Predict distribution over map positions



Plan Generation

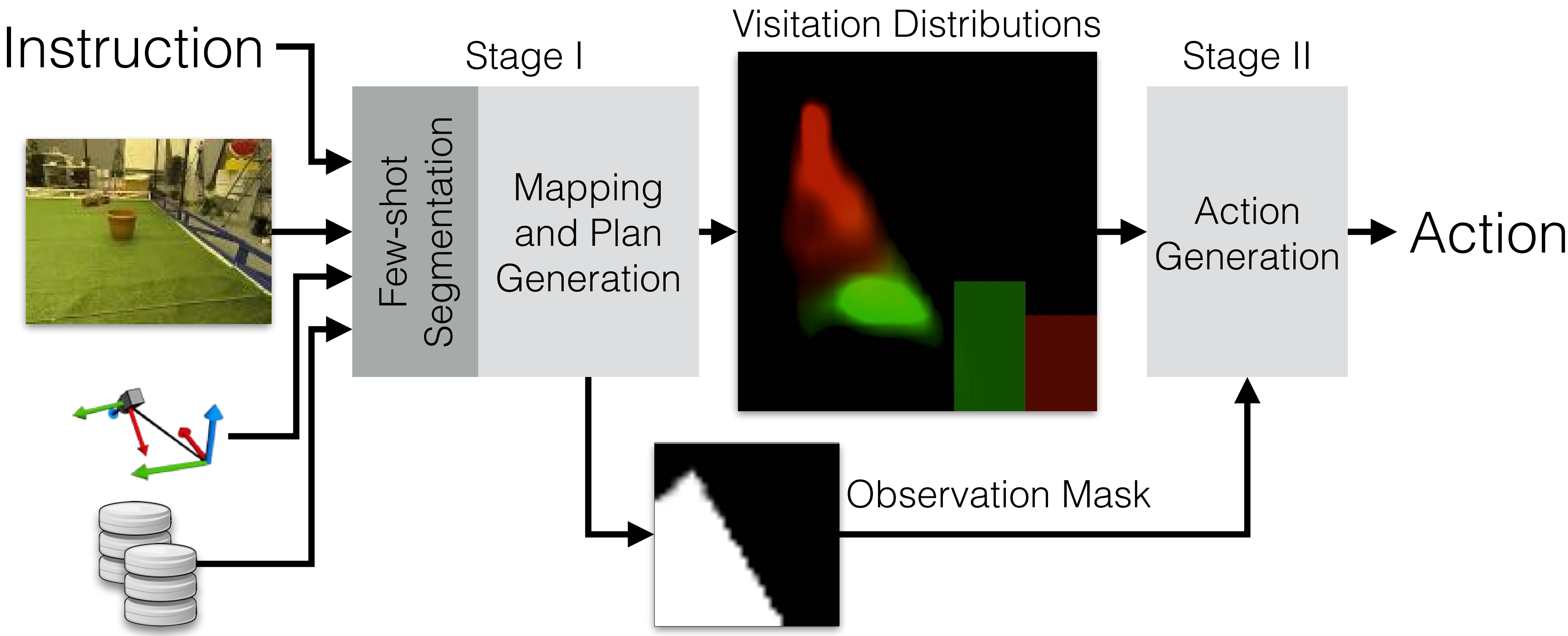
- Cast distribution prediction as image generation
- LingUNet: an image-to-image encoder-decoder
- Visual reasoning at multiple image scales
- Conditioned on language input at all levels of reasoning using text-based convolutions

LingUNet



Instruction
Visitation Distributions
Object
Map
RNN
SoftMax
Text Kernels
Convolutions
Deconvolutions
Text Convolutions

Two-stage Policy

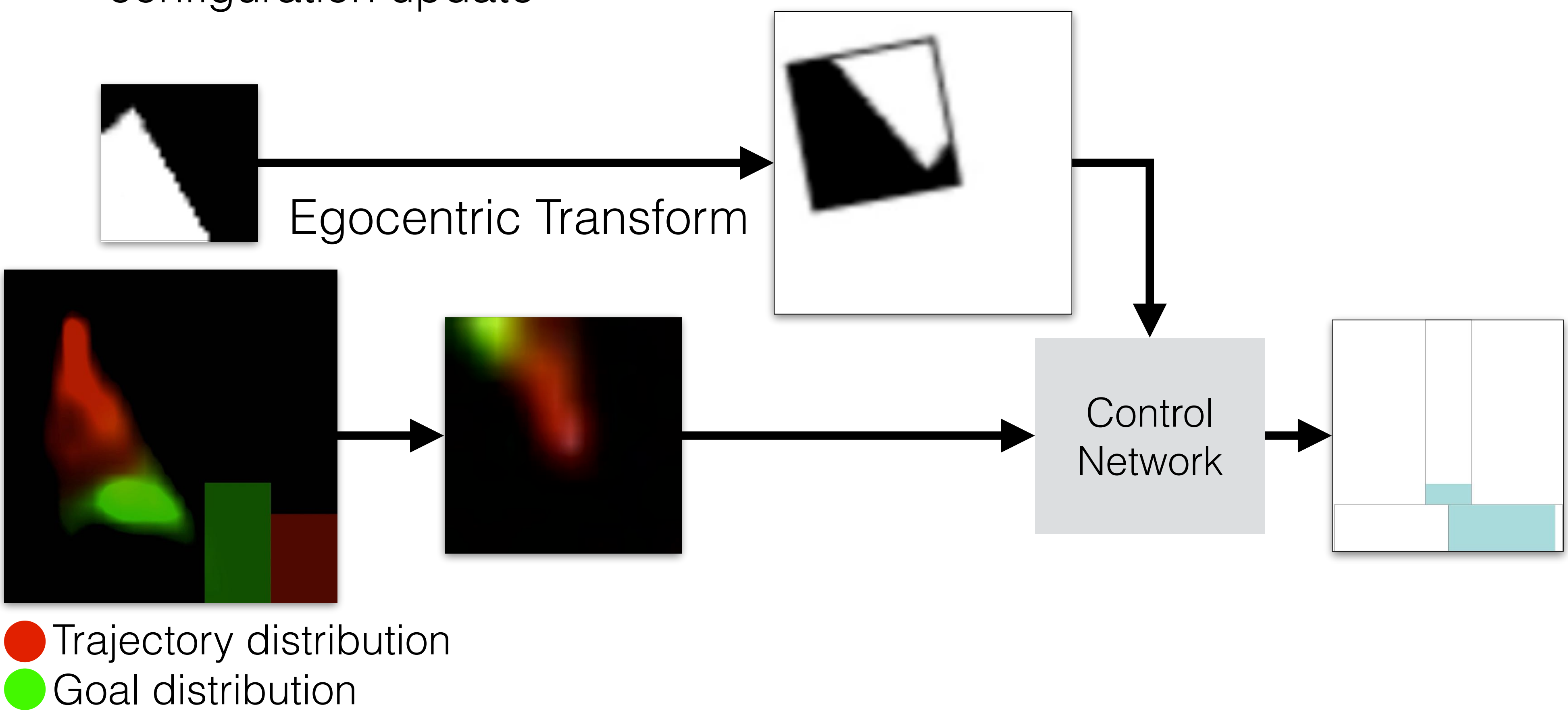


1. Map and predict states likely to visit + track observability

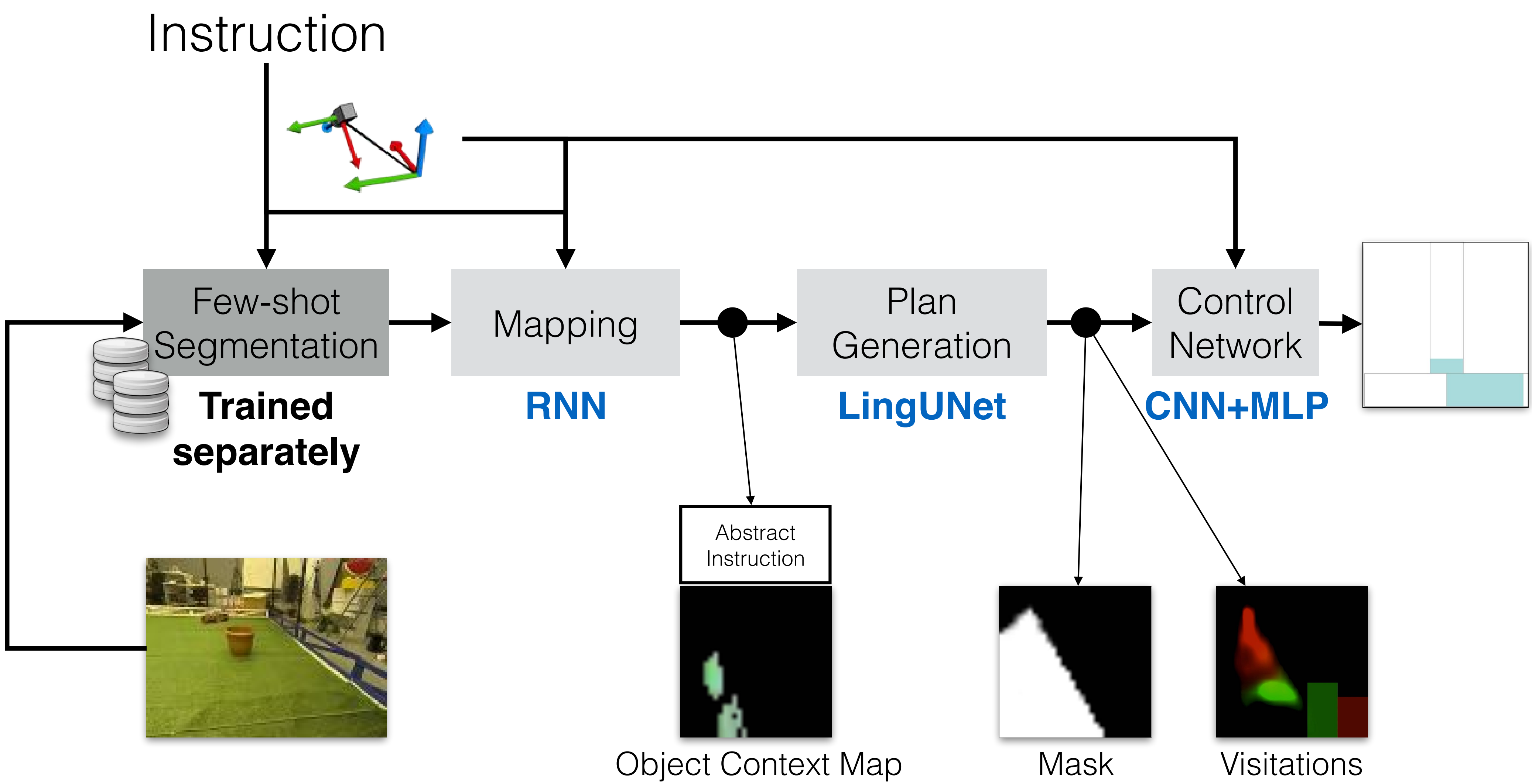
2. Generate actions to visit high-probability states and explore

Stage II: Action Generation

- Relatively simple control problem without language
- Transform and crop to agent perspective and generate configuration update



Training


Instruction
Few-shot Segmentation
Trained separately
Mapping
RNN
Plan Generation
LingUNet
Control Network
CNN+MLP
Abstract Instruction
Object Context Map
Mask
Visitations

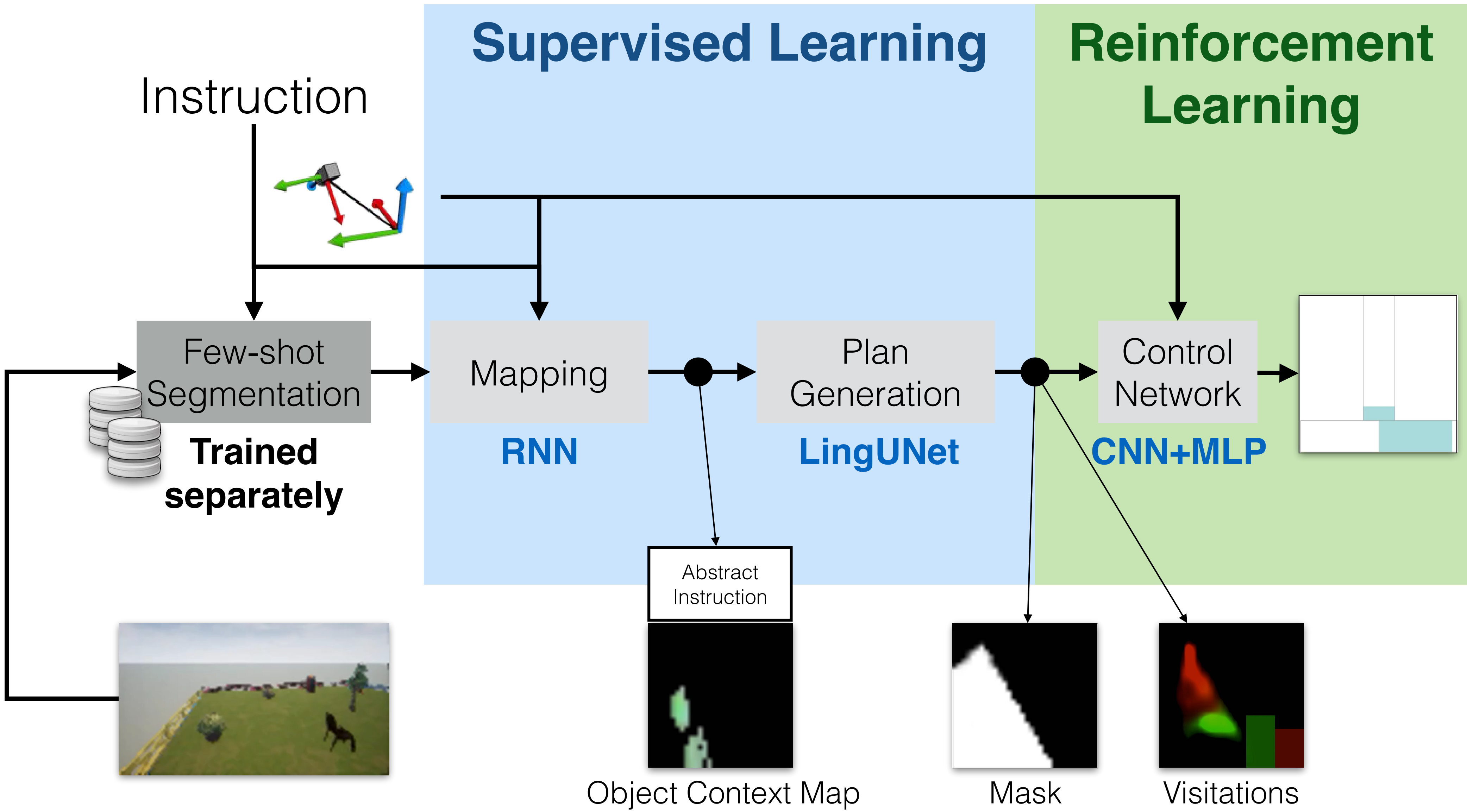
Training in Simulation

- Language-conditioned segmentation trained separately for simulation and real environment
- Policy training does not require access to real world
- After training: swap the segmentation component
- Data: demonstrations and experience

Go between the mushroom and flower chair the tree all the way up to the phone booth

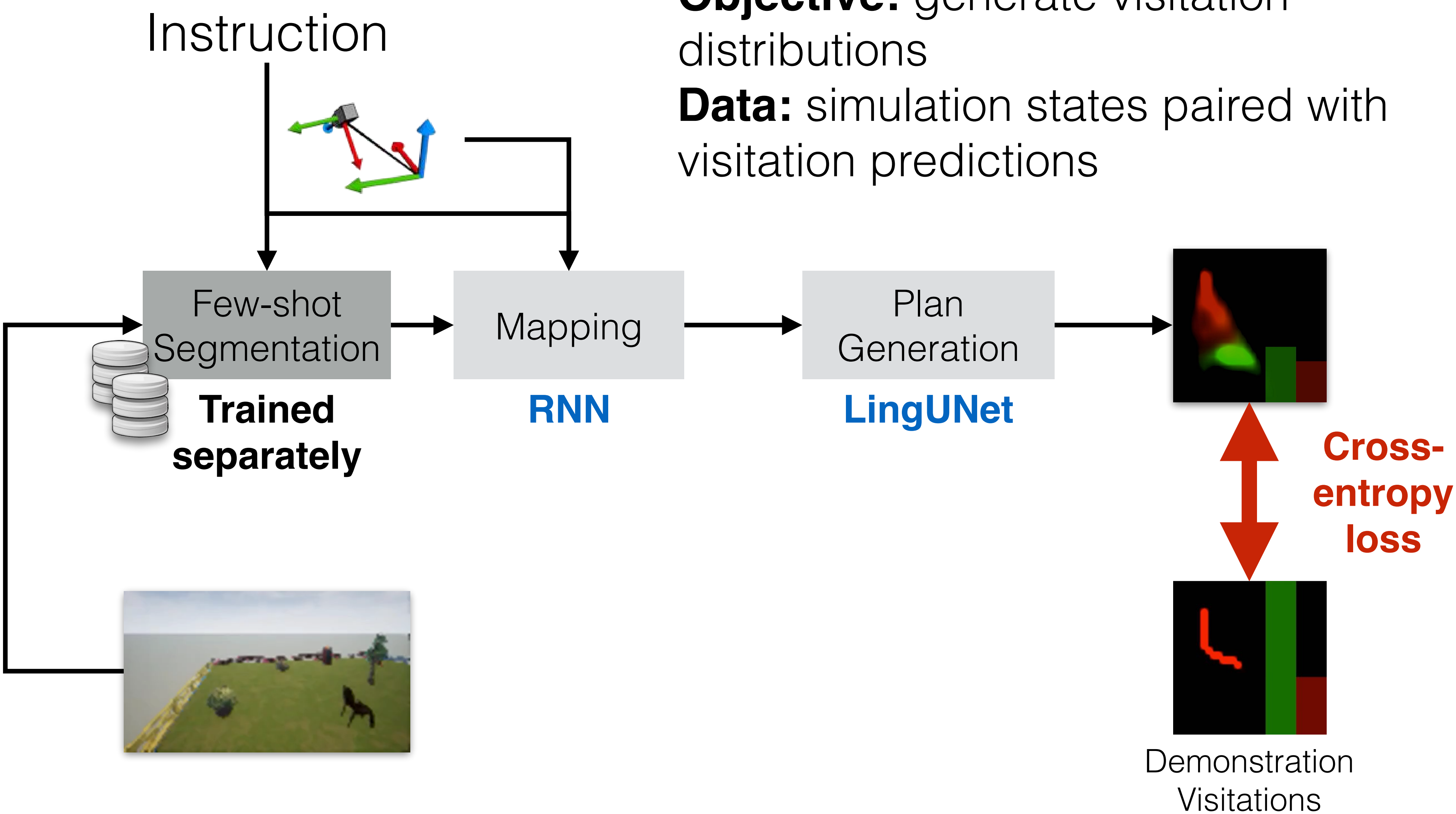
SuReAL

Supervised and Reinforcement Asynchronous Learning

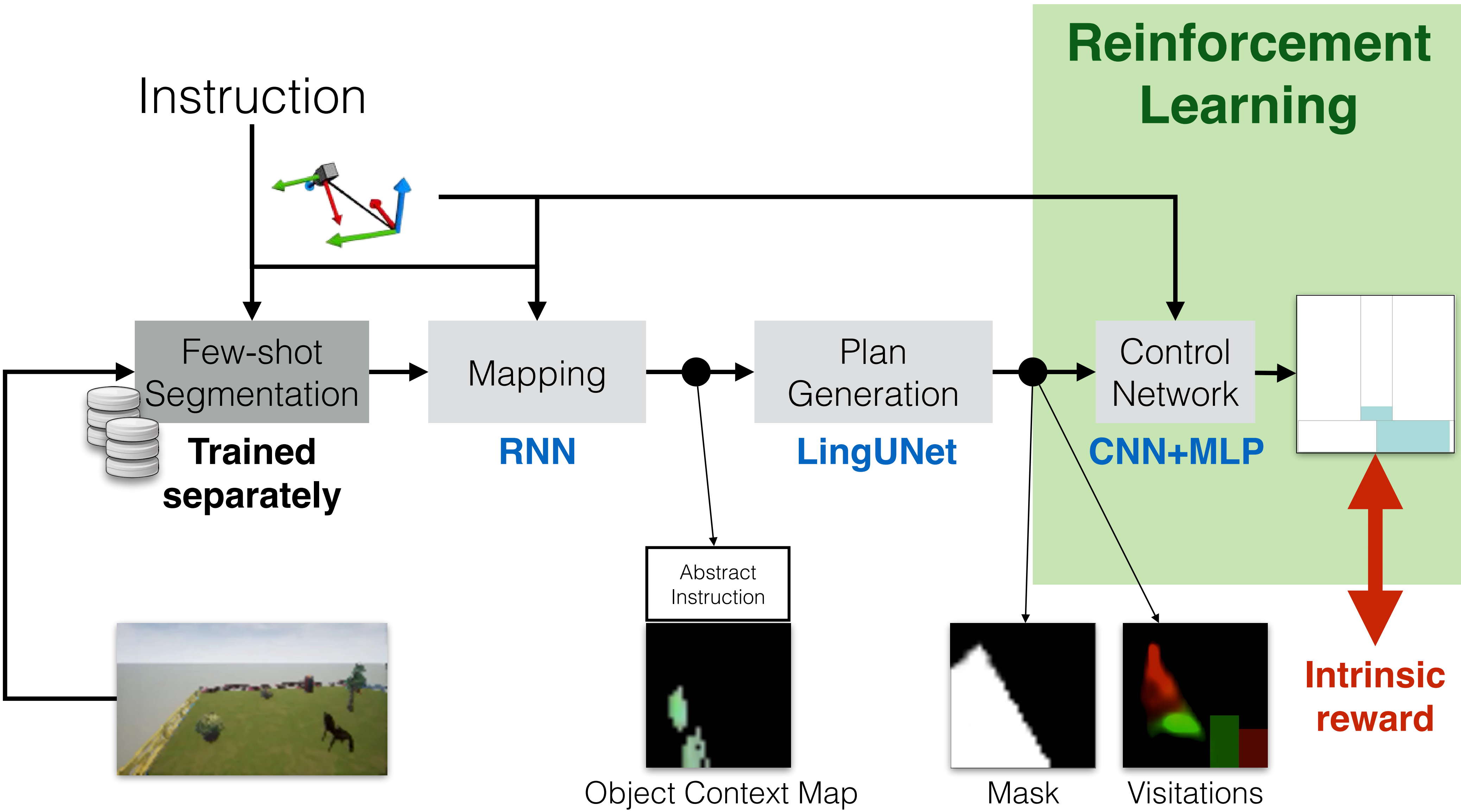


Supervised Learning

Objective: generate visitation distributions
Data: simulation states paired with visitation predictions

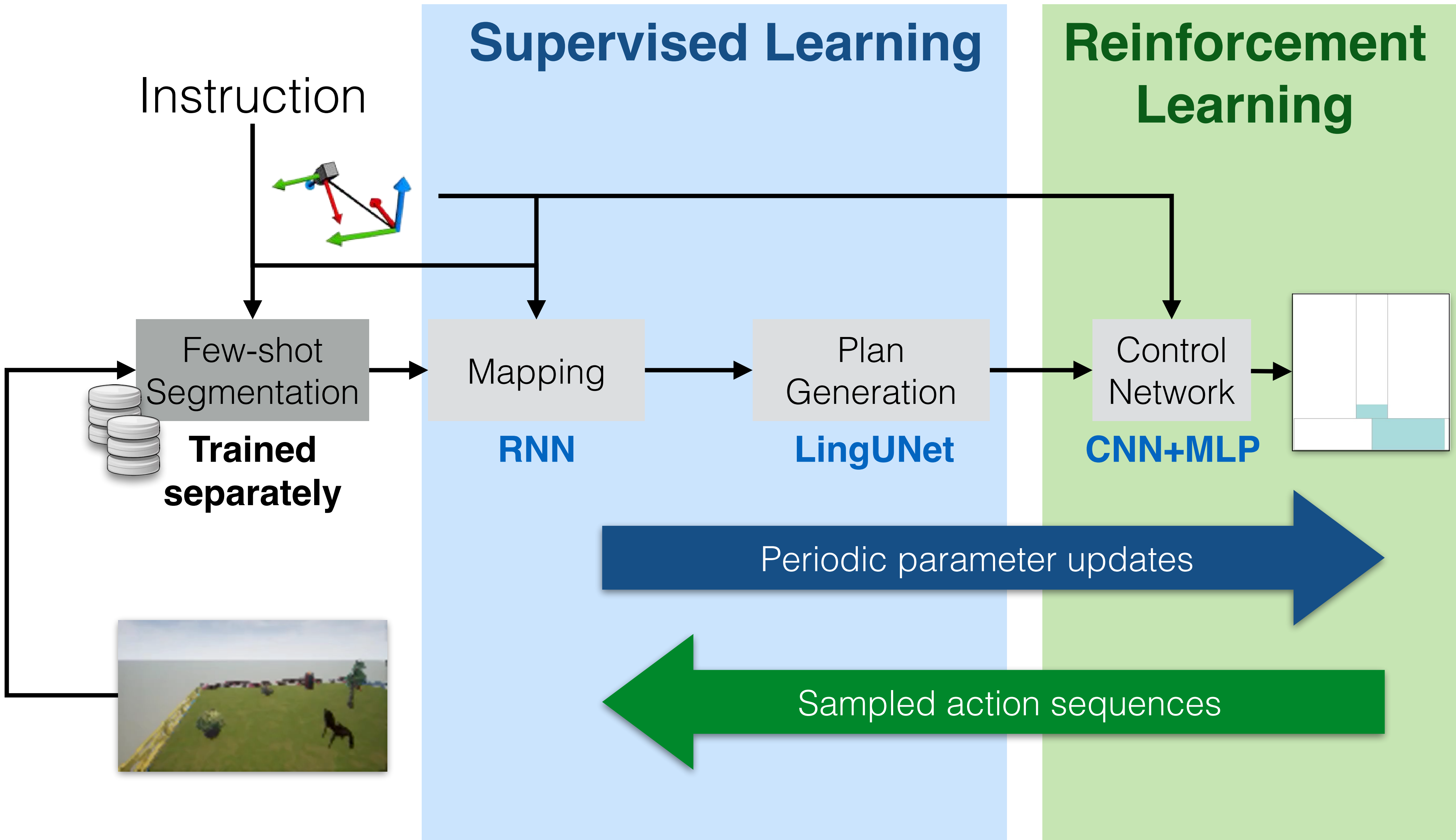


RL for Control



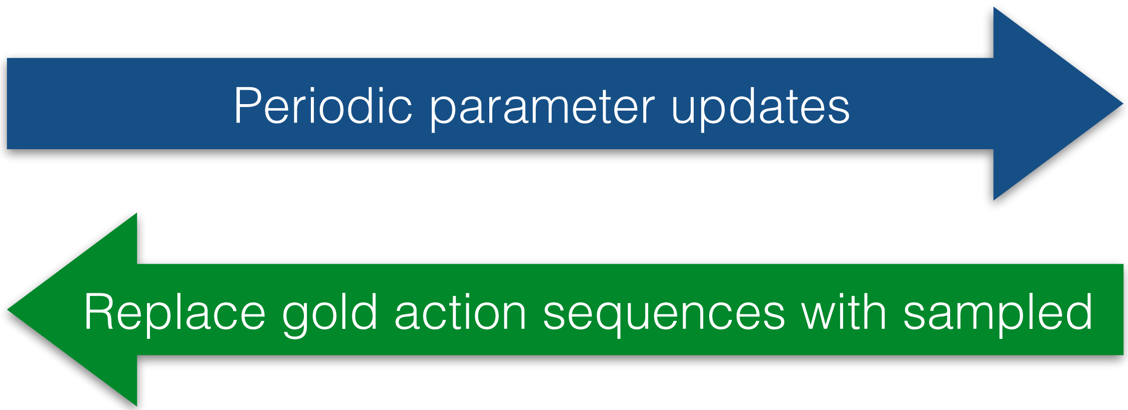
SuReAL

Supervised and Reinforcement Asynchronous Learning



SuReAL

Supervised and Reinforcement Asynchronous Learning

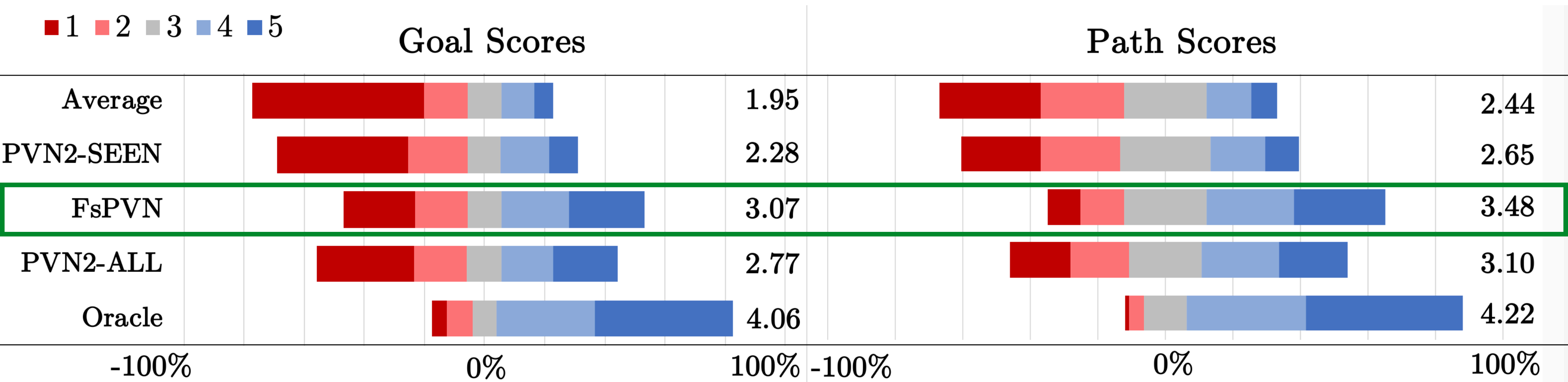


- **Stage I:** learn to predict visitation distributions based on noisy predicted execution trajectories
- **Stage II:** learn to predict actions using predicted visitation distributions

Experimental Setup

- Intel Aero quadcopter
- Vicon motion capture for pose estimate
- Simulation with Microsoft AirSim
- Drone cage is 4.7x4.7m
- All evaluation with eight new objects
- Database includes five images and five phrases for each object
- Training data: 41k instruction-demonstration pairs in simulation, no demonstration data in the real world

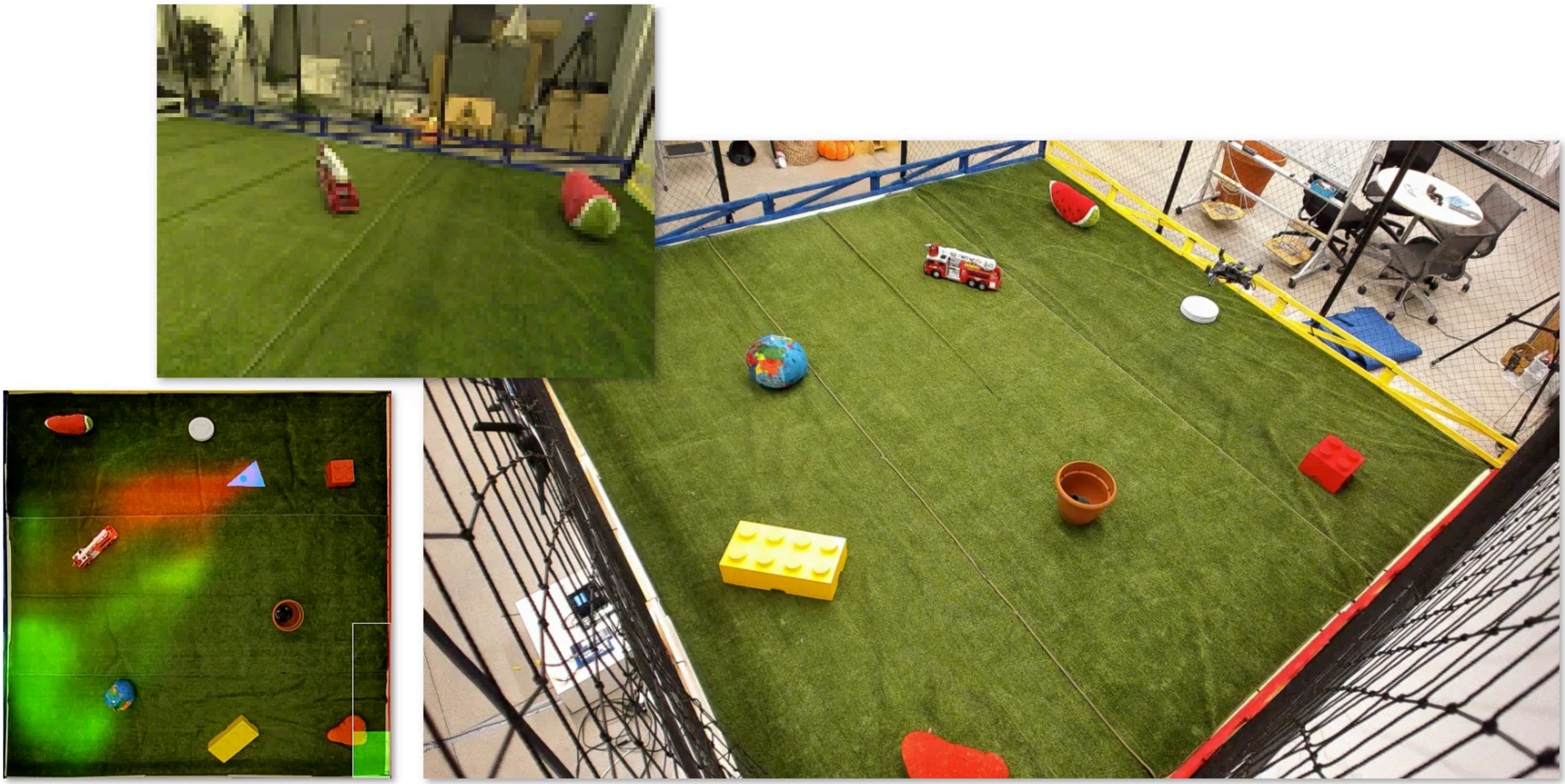
Human Evaluation



- Score path and goal on a 5-point Likert scale for 63 examples
- Our model receives 4-5 path scores 53% of the time, double than PVN2-SEEN, showing effective generalization to unknown objects
- Outperforming PVN2-ALL illustrates the benefit of the object-centric inductive bias

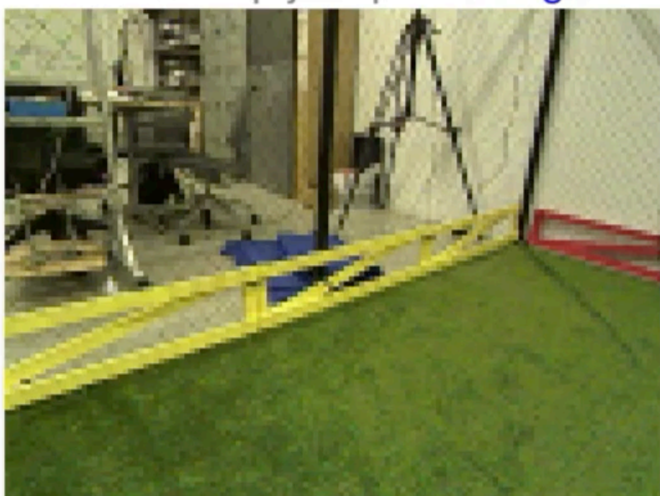
Example

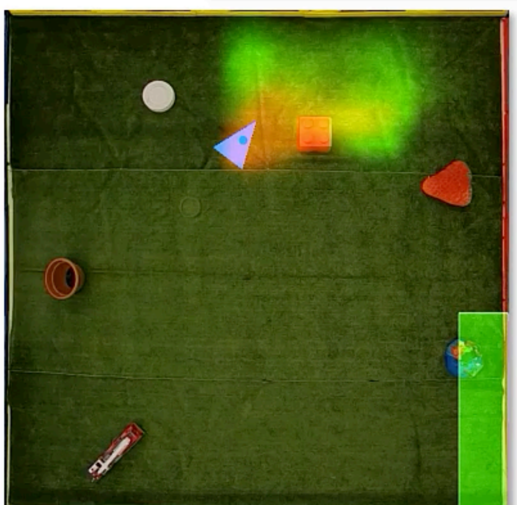
and move to the right side of the firetruck
and looping around it until heading towards the globes
left side

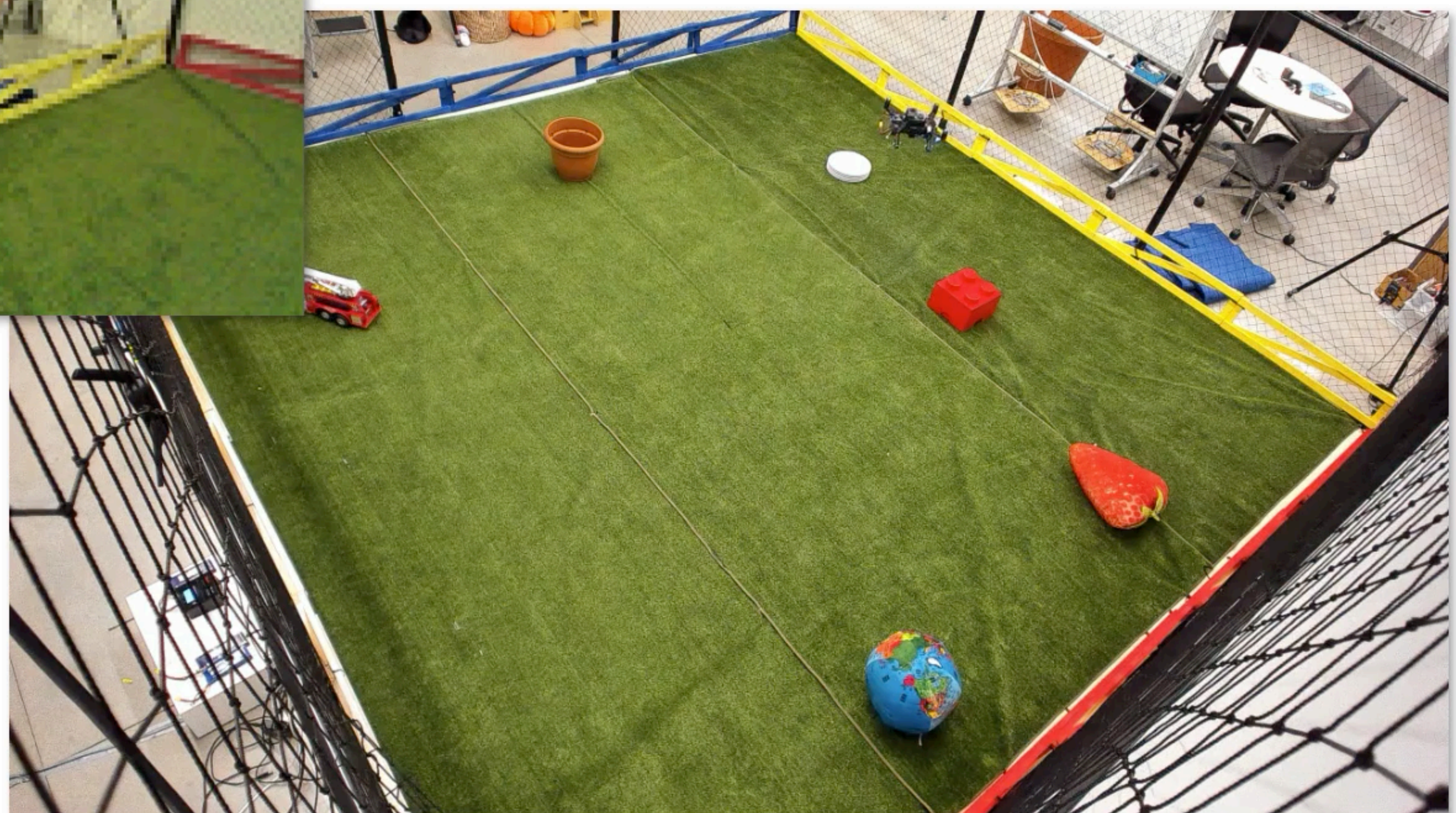


Messy Example

keep the red lego to your right as you to around it
to face the beige fence pass the strawberry on your left
and move forward to stop just past the globe



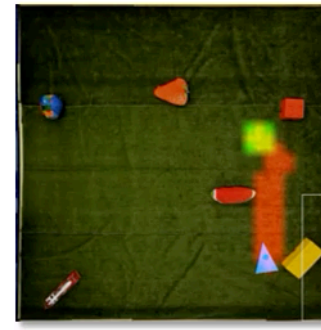


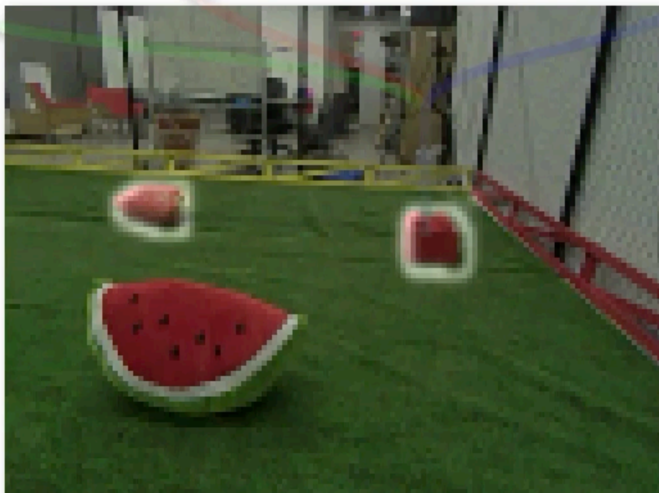


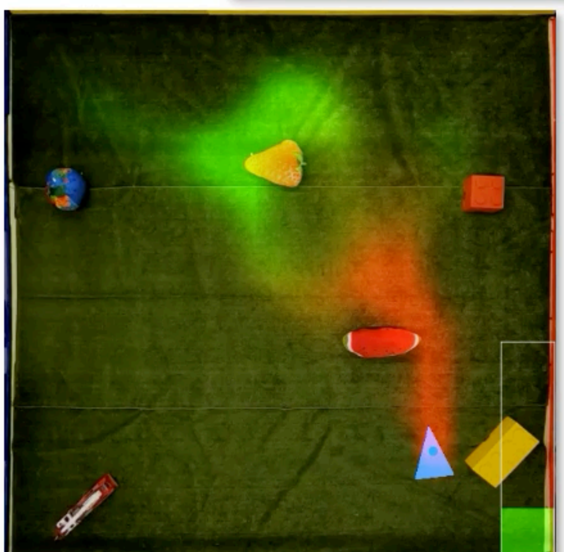
Failure

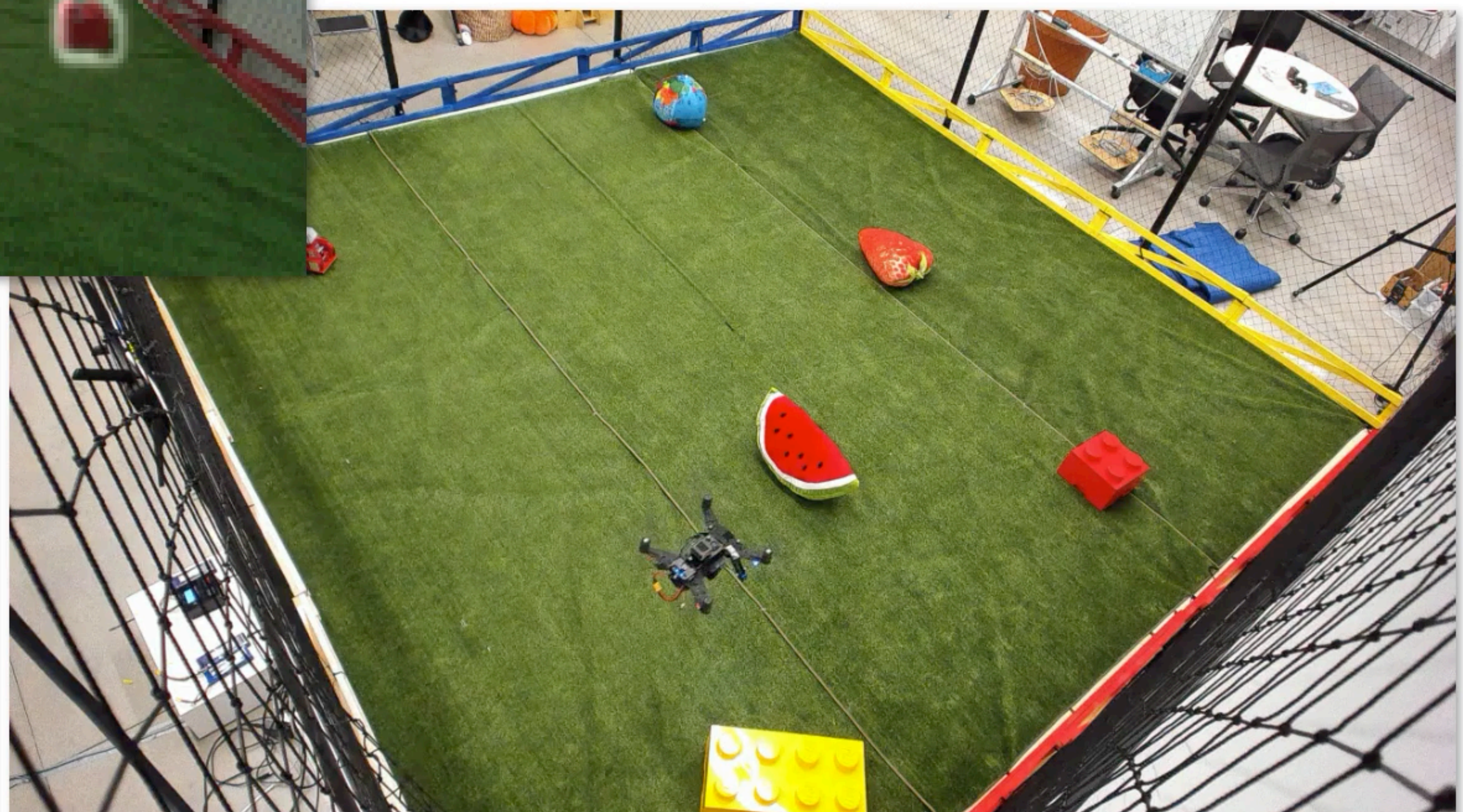
fly straight at the red lego up ahead stop just in front of the red lego and then veer left in front of the red lego

Ground Truth









Today

Few-shot instruction following:

- Few-shot language-conditioned object segmentation
  Modeling objects and aligning their references and observations + training with augmented reality data

- Object context mapping
  Incorporate contextual text information into spatial map without specific object information

- Integration into a visitation-prediction policy for mapping instructions to drone control
  Generate trajectory plans over object context map + train in simulation only by swapping the segmentation component

Some Open Questions

- How to elicit exemplars to add to the database from human users, potentially within interaction?
- How to generalize from objects to more general objects types?
- What other object properties should we model? Such as permanence and reference consistency

The Papers

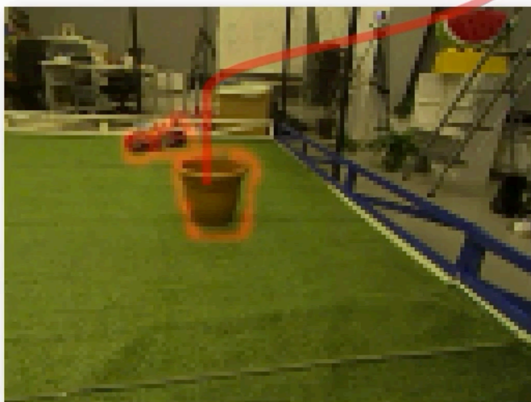
- **Few-shot Object Grounding for Mapping Natural Language Instructions to Robot Control**
  Valts Blukis, Ross A. Knepper, and Yoav Artzi
  CoRL, 2020

- **Learning to Map Natural Language Instructions to Physical Quadcopter Control Using Simulated Flight**
  Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A. Knepper, and Yoav Artzi
  CoRL, 2019

- **Mapping Navigation Instructions to Continuous Control Actions with Position Visitation Prediction**
  Valts Blukis, Dipendra Misra, Ross A. Knepper, and Yoav Artzi
  CoRL, 2018

- **Following High-level Navigation Instructions on a Simulated Quadcopter with Imitation Learning**
  Valts Blukis, Nataly Brukhim, Andrew Bennett, Ross A. Knepper, and Yoav Artzi
  RSS, 2018.

go straight and stop before reaching the planter turn left towards the globe and go forward until just before it



Valts Blukis

And collaborators: Dipendra Misra, Eyvind Niklasson, Nataly Brukhim, Andrew Bennett, and Ross Knepper

https://github.com/lil-lab/drif

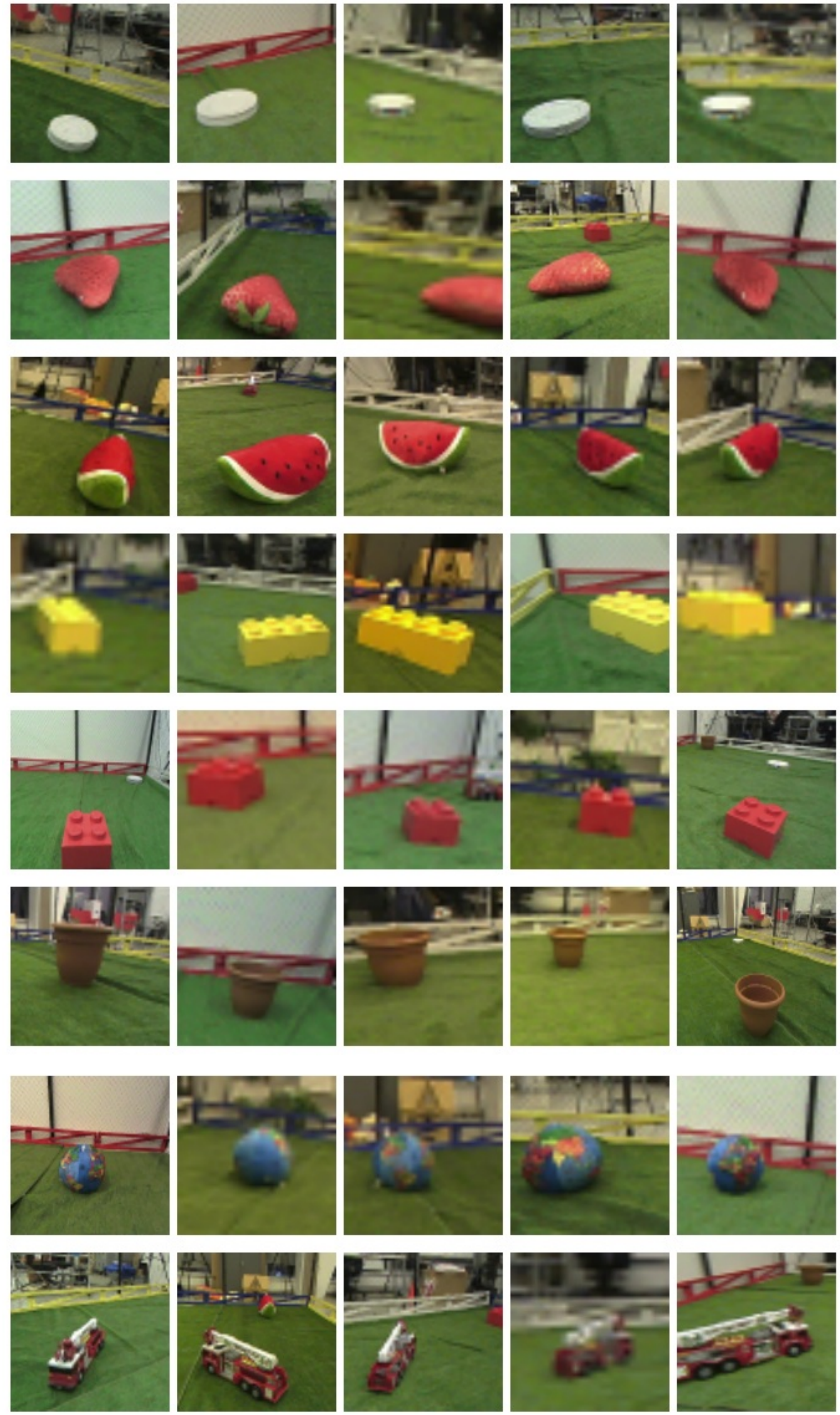
Thank you! Questions?

[fin]

Object Database

The object database used during **development** in the physical environment.

The object database used **during testing**, containing previously unseen physical objects.



the dinner plate
frisbee
white plate
the ceramic
the round white disc

strawberry
the strawberry
a red strawberry
berry
a red strawberry object

watermelon
pink watermelon slice
melon wedge
berry slice
the fruit slice

the yellow lego
yellow brick
yellow block
the yellow rectangle
the yellow lego piece

the red lego
red brick
square lego block
the red building block
red cube

the pot
clay pot
flower pot
the brown planter
terracotta planter
orange cup

the globe
globe
the blue globe
earth
saucer

truck
red firetruck
red and white fire truck
engine
the fire engine

Visitation Distributions

Visitation Distribution

- Given a Markov Decisions Process:

MDP S States A Actions R Reward H Horizon

- The state-visitation distribution $d(s; \pi, s_0)$ is the probability of visiting state s following policy π from start state s_0

- Predicting $d(s; \pi^*, s_0)$ for an expert policy π^* tells us the states to visit to complete the task

- Can learn from demonstrations, but prediction generally impossible: S is very large!

Approximating Visitation Distributions

MDP S States A Actions R Reward H Horizon

- Solution: approximate the state space

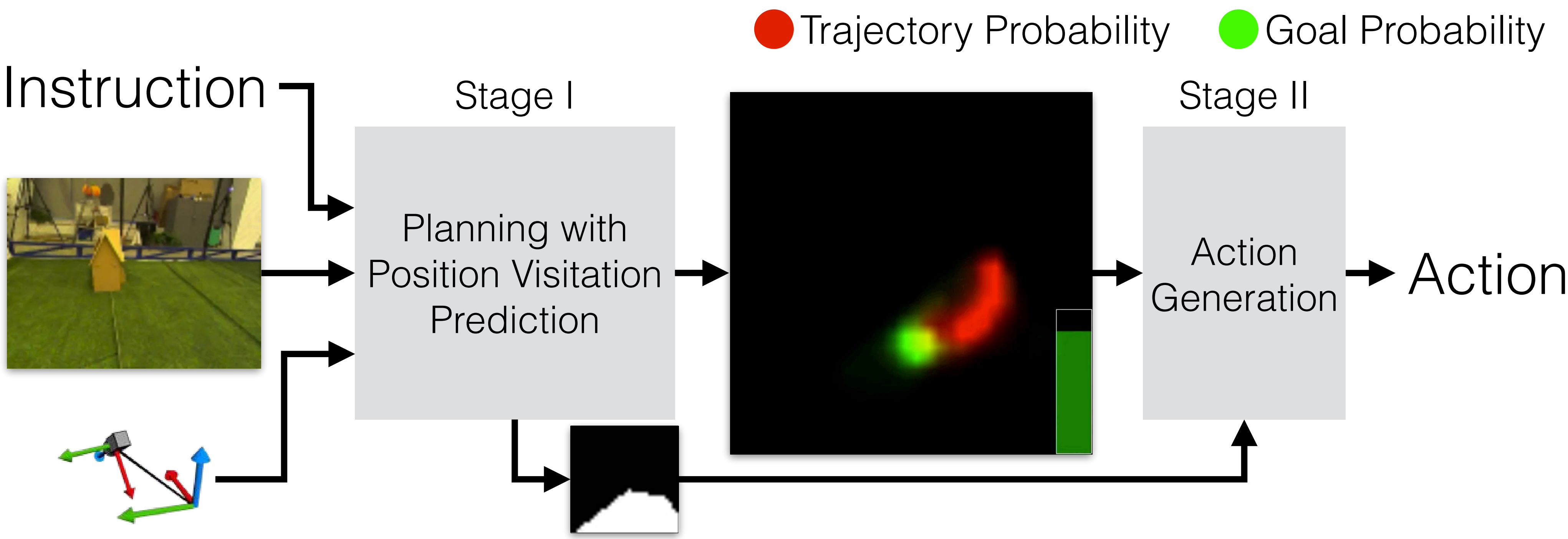
- Use an approximate state space $\tilde{S}$ and a mapping between the state spaces $\phi : S \rightarrow \tilde{S}$

- For a well chosen ϕ, a policy π with a state-visitation distribution close to $d(\tilde{s}; \pi^*, \tilde{s}_0)$ has bounded sub-optimality

Visitation Distribution for Navigation

MDP S States A Actions R Reward H Horizon

- $\tilde{S}$ is a set of discrete positions in the world
- We compute two distributions: **trajectory-visitation** and **goal-visitation**



Drone Related Work

(Somewhat outdated)

Related Work: Task

- Mapping instructions to actions with robotic agents

Tellex et al. 2011; Matuszek et al. 2012; Duvallet et al. 2013; Walter et al. 2013; Misra et al. 2014; Hemachandra et al. 2015; Lignos et al. 2015

- Mapping instruction to actions in software and simulated environments

MacMahon et al. 2006; Branavan et al. 2010; Matuszek et al. 2010, 2012; Artzi et al. 2013, 2014; Misra et al. 2017, 2018; Anderson et al. 2017; Suhr and Artzi 2018

- Learning visuomotor policies for robotic agents

Lenz et al. 2015; Levine et al. 2016; Bhatti et al. 2016; Nair et al. 2017; Tobin et al. 2017; Quillen et al. 2018, Sadeghi et al. 2017

Related Work: Method

- Mapping and planning in neural networks

Bhatti et al. 2016; Gupta et al. 2017; Khan et al. 2018; Savinov et al. 2018; Srinivas et al. 2018

- Model and learning decomposition

Pastor et al. 2009, 2011; Konidaris et al. 2012; Paraschos et al. 2013; Maeda et al. 2017

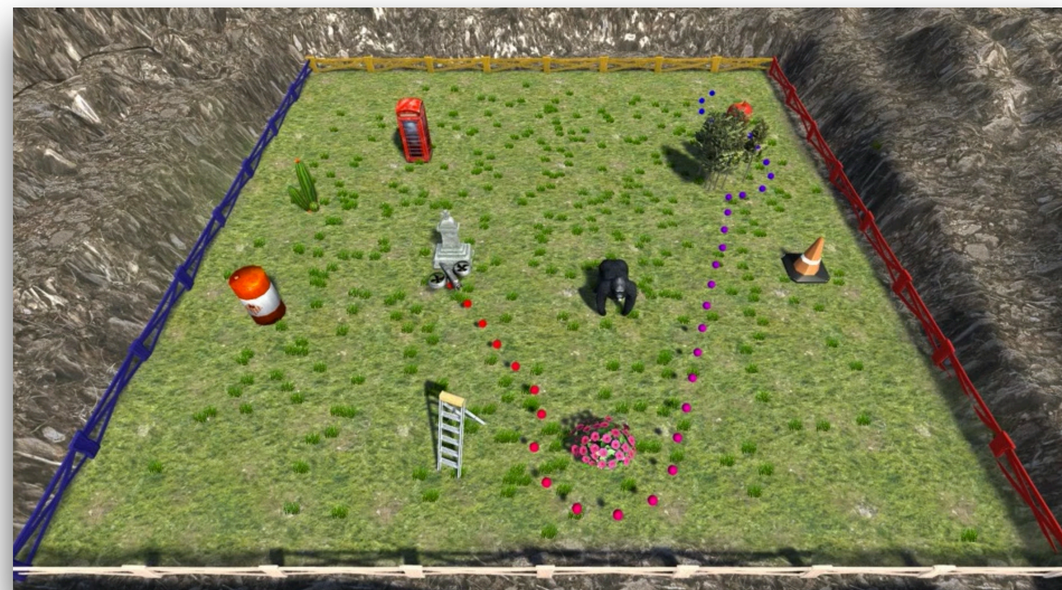
- Learning to explore

Knepper et al. 2015; Nyga et al. 2018

Drone Data Collection

Data

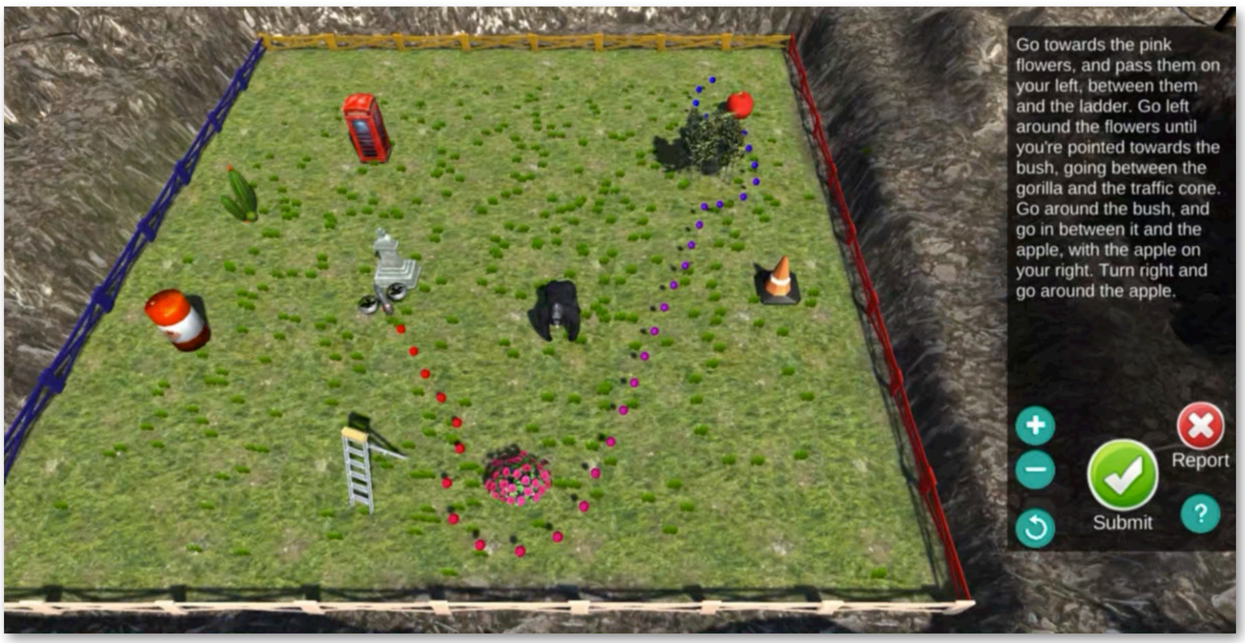
- Crowdsourced with a simplified environment and agent
- Two-step data collection: writing and validation/segmentation



Go towards the pink flowers and pass them on your left, between them and the ladder. Go left around the flower until you're pointed towards the bush, going between the gorilla and the traffic cone. Go around the bush, and go in between it and the apple, with the apple on your right. Turn right and go around the apple.

Data

- Crowdsourced with a simplified environment and agent
- Two-step data collection: writing and validation/segmentation



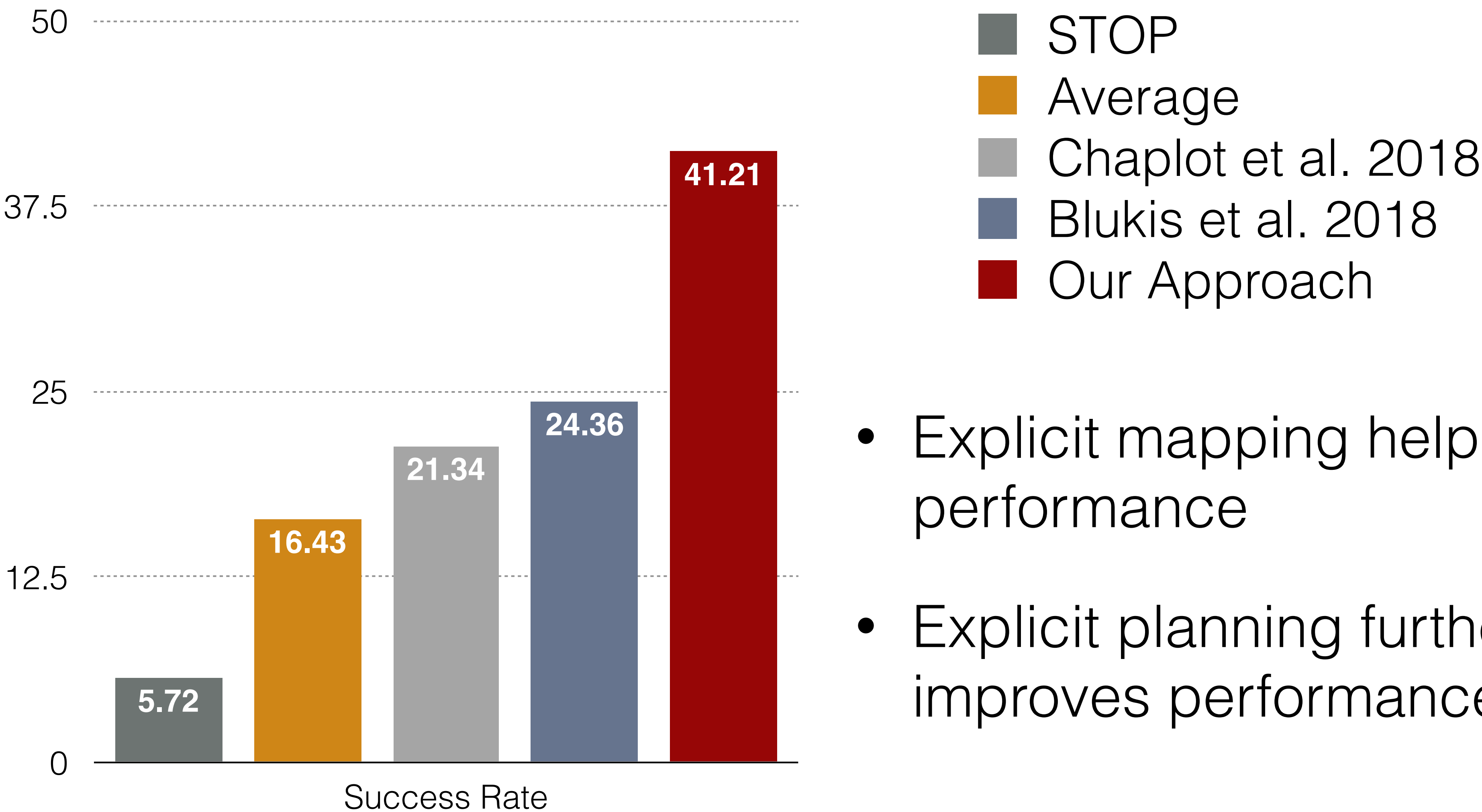
Go towards the pink flowers and pass them on your left, between them and the ladder. Go left around the flower until you're pointed towards the bush, going between the gorilla and the traffic cone. Go around the bush, and go in between it and the apple, with the apple on your right. Turn right and go around the apple.

CoRL 2018 Experiments

Experimental Setup

- Crowdsourced instructions and demonstrations
- 19,758/4,135/4,072 train/dev/test examples
- Each environment includes 6-13 landmarks
- Quadcopter simulation with AirSim
- Metric: task-completion accuracy

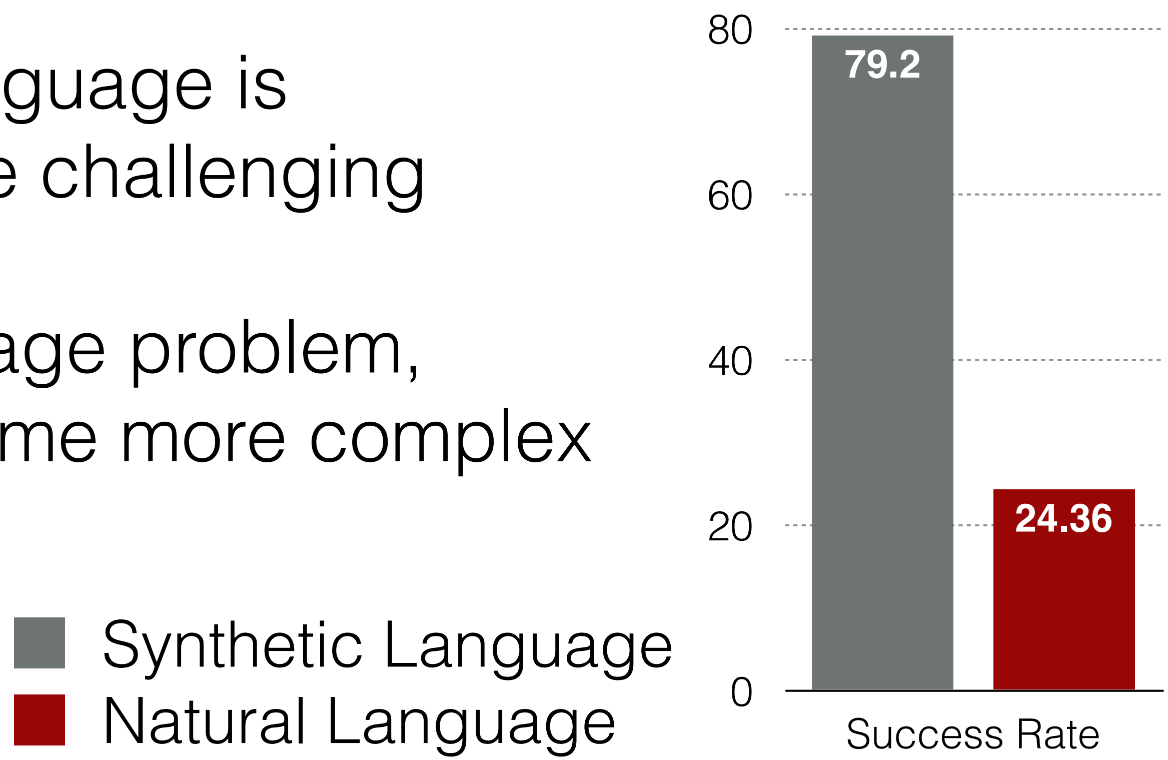
Test Results



- Explicit mapping helps performance
- Explicit planning further improves performance

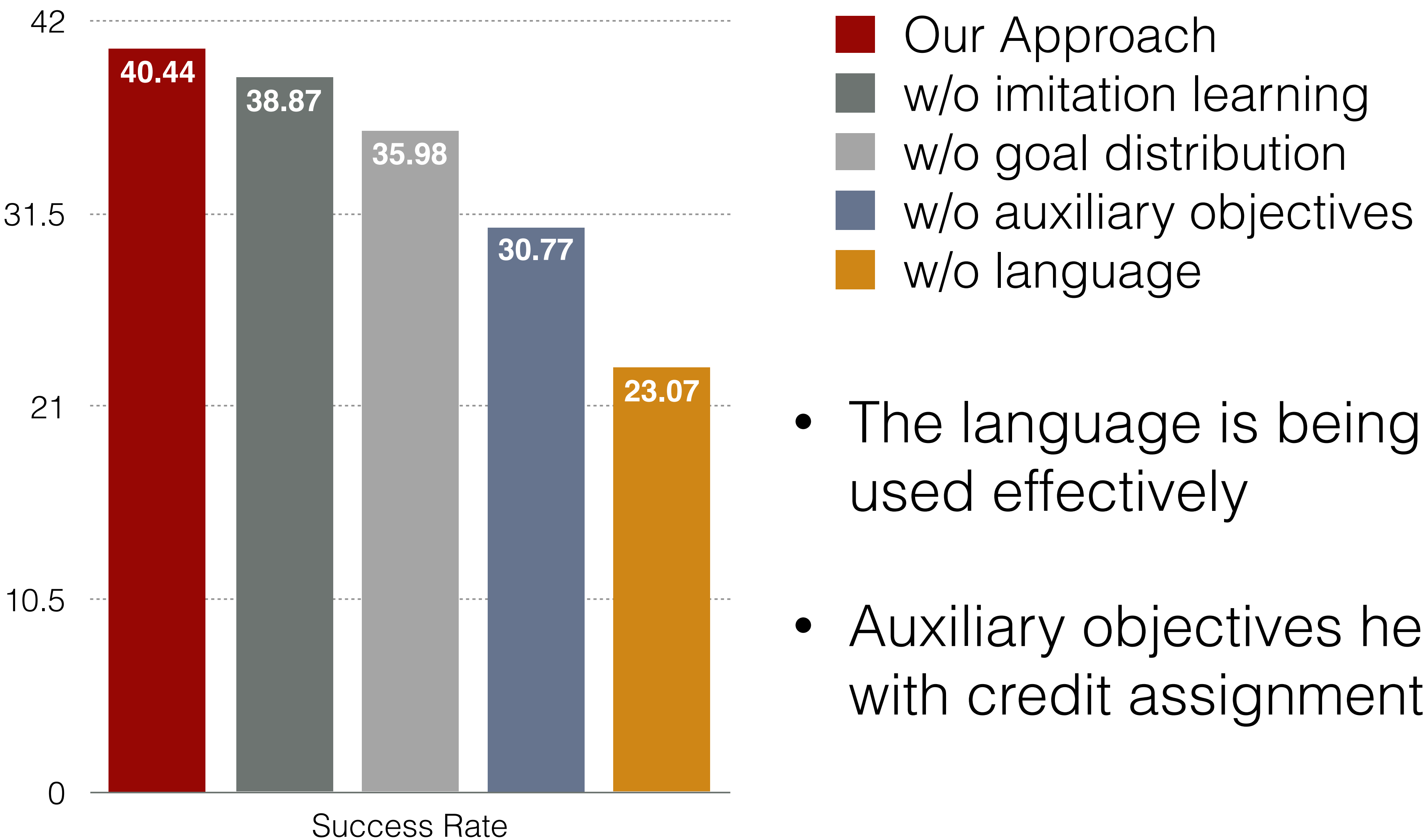
Synthetic vs. Natural Language

- Synthetically generated instructions with templates
- Evaluated with explicit mapping (Blukis et al. 2018)
- Using natural language is significantly more challenging
- Not only a language problem, trajectories become more complex



Ablations

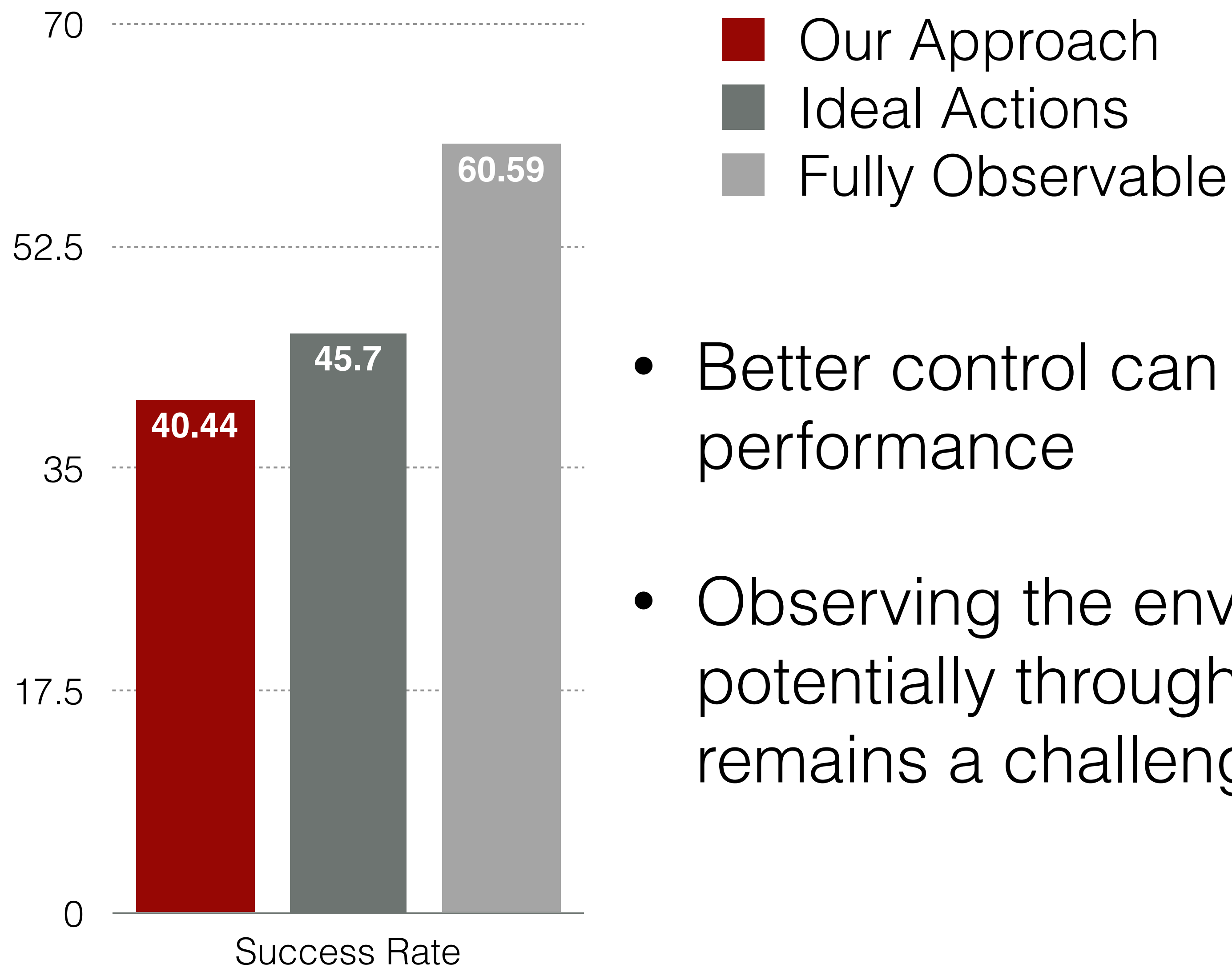
Development Results



- The language is being used effectively
- Auxiliary objectives help with credit assignment

Analysis

Development Results



- Better control can improve performance
- Observing the environment, potentially through exploration, remains a challenge

CoRL 2019 Experiments

Environment

- Drone cage is 4.7x4.7m
- Created in reality and simulation
- 15 possible landmarks, 5-8 in each environment
- Also: larger 50x50m simulation-only environment with 6-13 landmarks out of possible 63

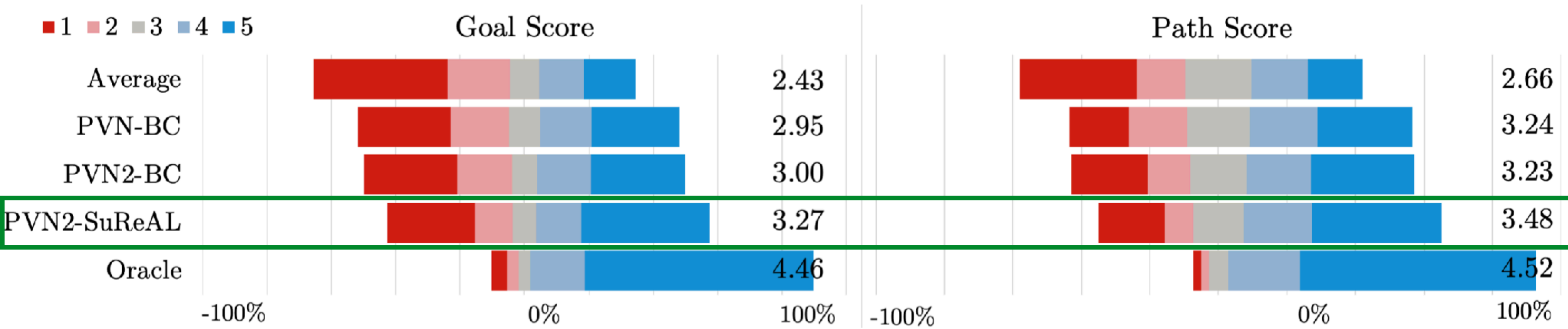
Data

- Real environment training data includes 100 instruction paragraphs, segmented to 402 instructions
- Evaluation with 20 paragraphs
- Evaluate on concatenated consecutive segments
- Oracle trajectories from a simple carrot planner
- Much more data in simulation, including for a larger 50x50m environment

Evaluation

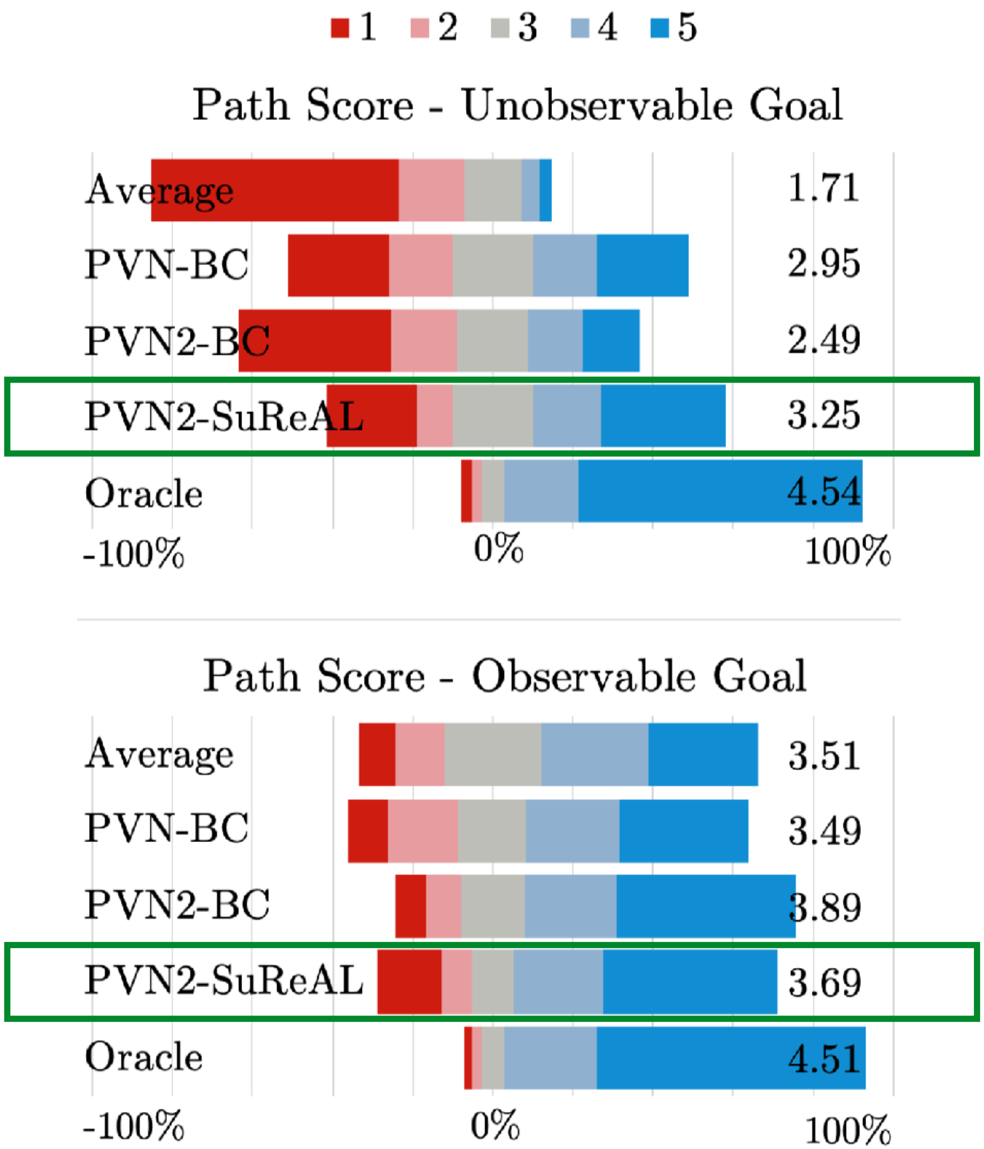
- Two automated metrics
  - SR: success rate
  - EMD: path earth’s move distance
- Human evaluation: score path and goal on a 5-point Likert scale

Human Evaluation



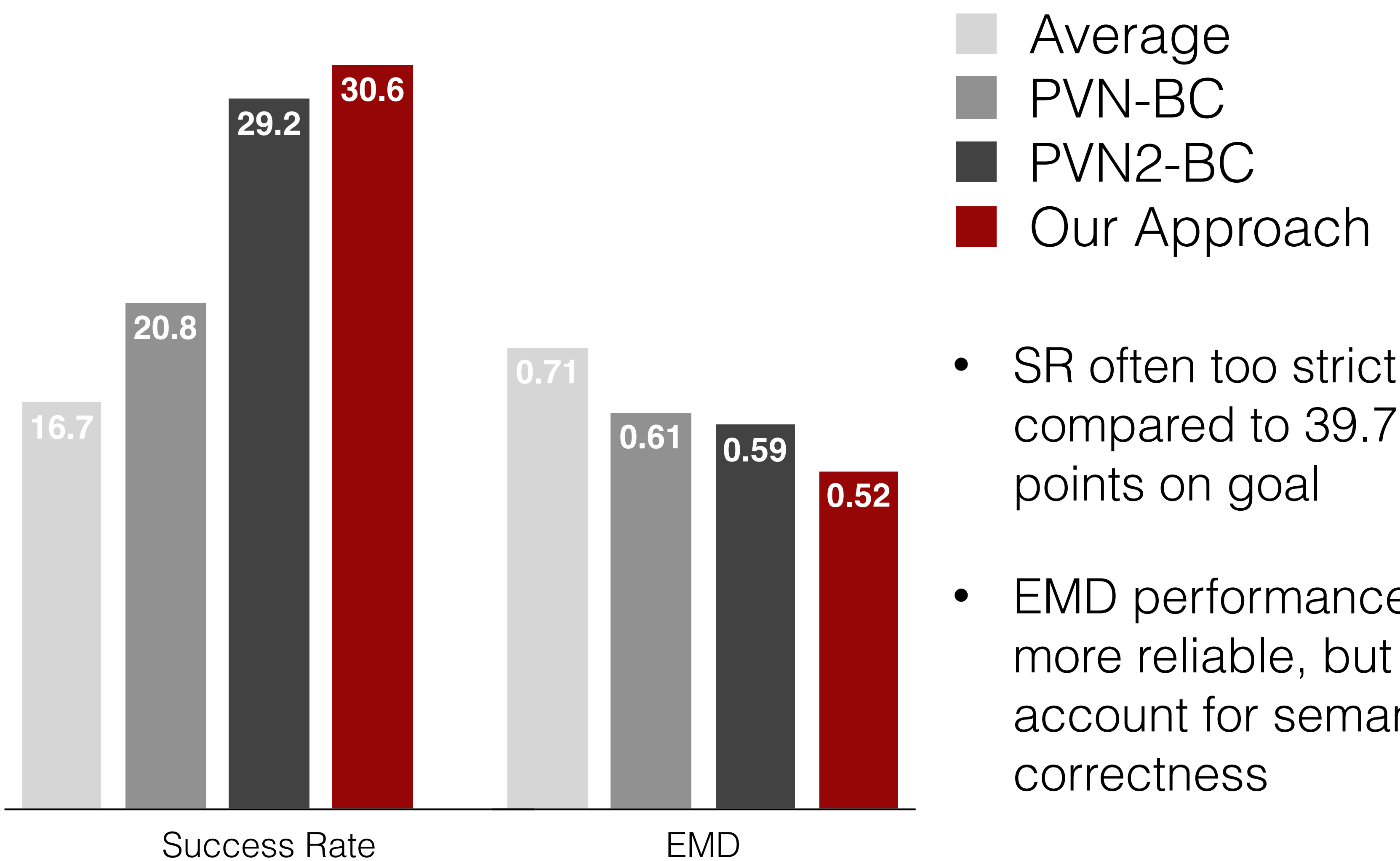
- Score path and goal on a 5-point Likert scale for 73 examples
- Our model receives five-point path scores 37.8% of the time, 24.8% improvement over PVN2-BC
- Improvements over PVN2-BC illustrates the benefit of SuReAL and the exploration reward

Observability



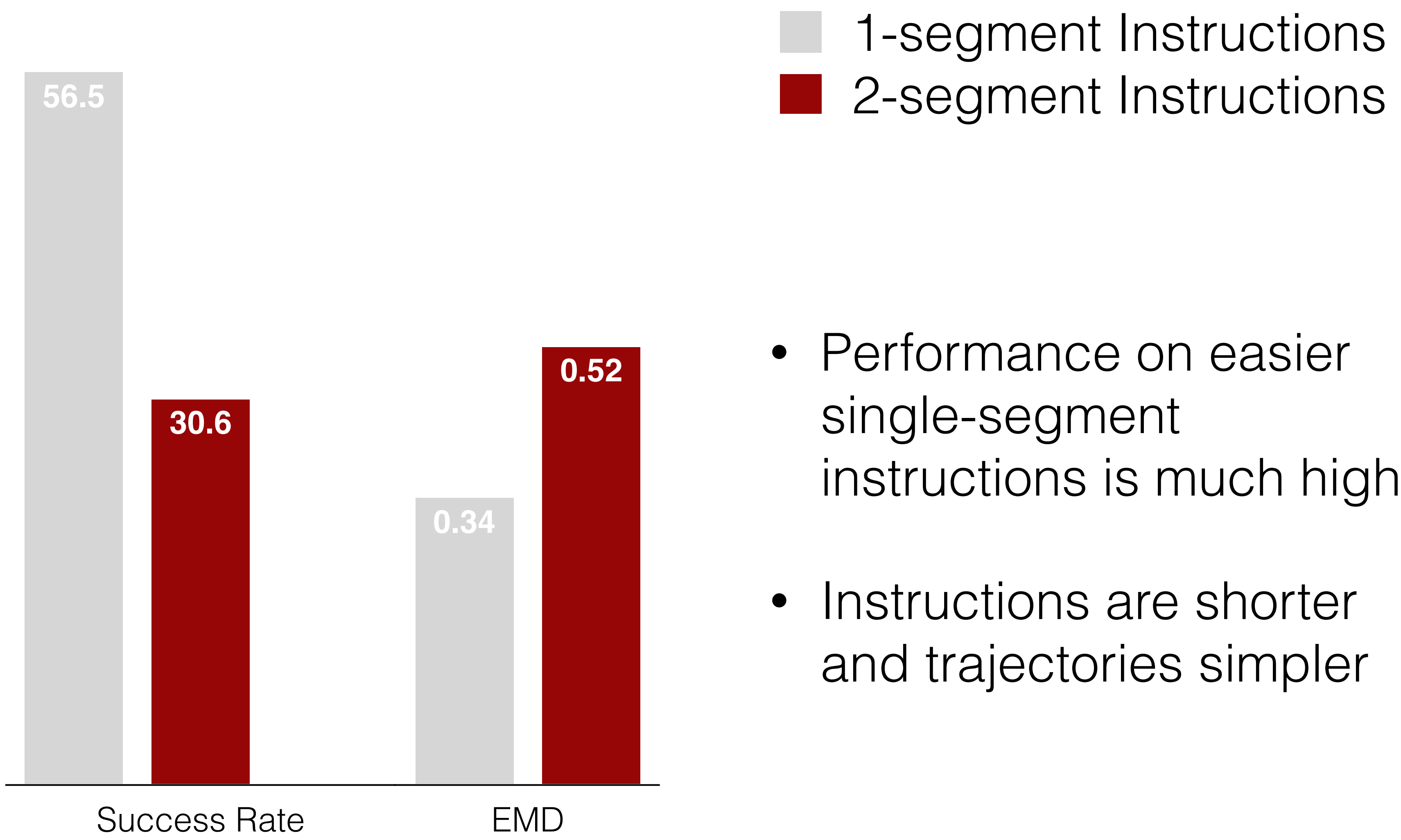
- Big benefit when goal is not immediately observed
- However, complexity comes at small performance cost on easier examples

Test Results



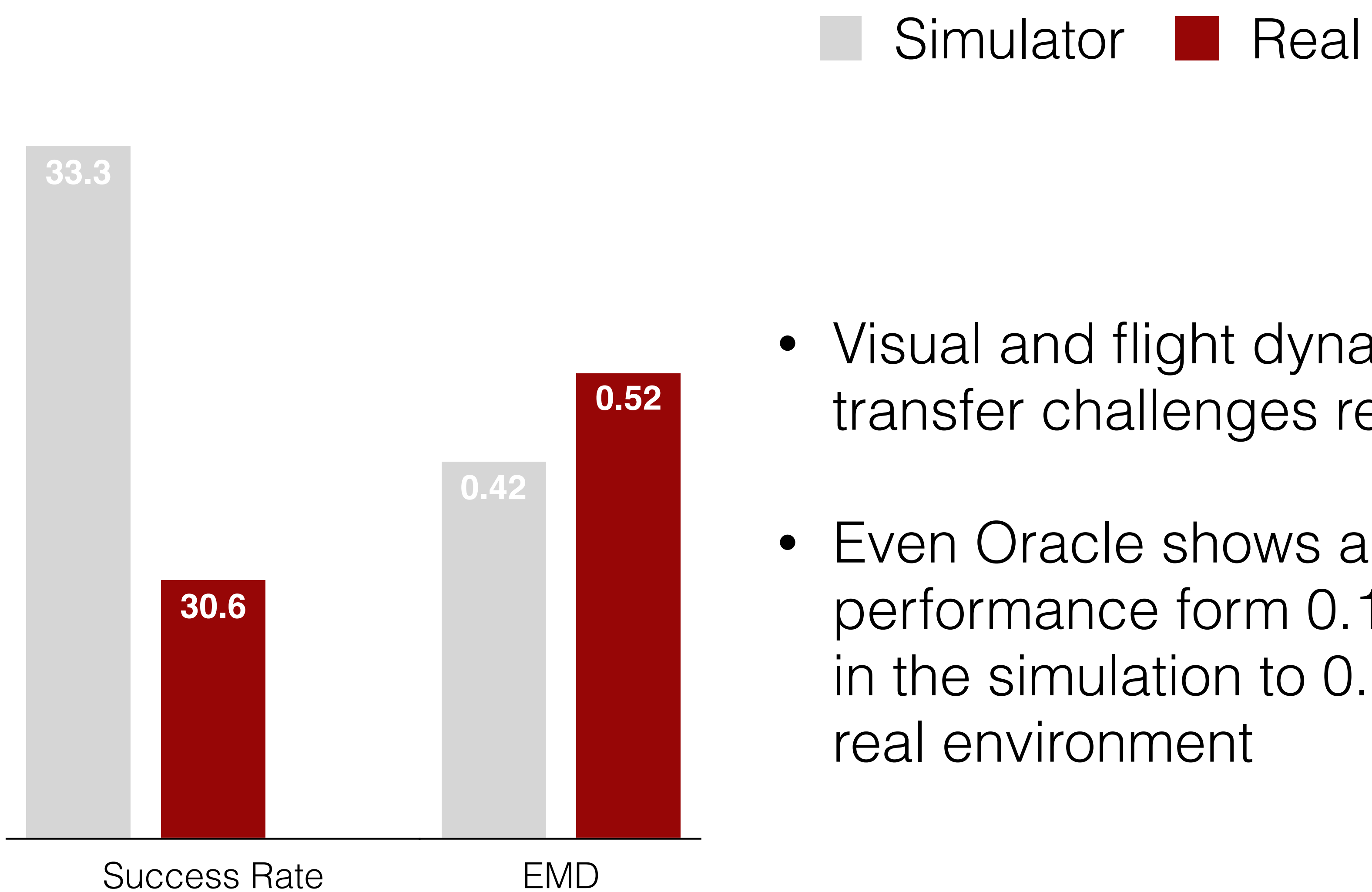
- SR often too strict: 30.6% compared to 39.7% five-points on goal
- EMD performance generally more reliable, but still fails to account for semantic correctness

Simple vs. Complex Instructions



- Performance on easier single-segment instructions is much higher
- Instructions are shorter and trajectories simpler

Transfer Effects



- Visual and flight dynamics transfer challenges remain
- Even Oracle shows a drop in performance form 0.17 EMD in the simulation to 0.23 in the real environment

CoRL 2019 Examples

Cool Example

once near the rear of the gorilla turn right and head towards the rock stopping once near it

Failure

head towards the area just to the left of the mushroom and then loop around it

CoRL 2019 Sim-real Shift Examples

Sim-real Control Shift

when you reach the right of the palm tree take a sharp right when you see a blue box   head toward it

Sim-real Control Shift