Robot Control in Situated Instruction Following

Yoav Artzi

Advances of Language & Vision Research Workshop
ACL 2020

Cornell CIS
Computer Science

CORNELL
TECH

Mapping Instructions to Actions

Goal: model and learn a mapping

$$f(\text{instruction}, \text{context}) = \text{actions}$$

Today

Mapping instructions to continuous control

- Generating and executing interpretable plans
- Jointly learning in real life and a simulation
- Training for test-time exploration

Task

- Navigation between landmarks
- Agent: quadcopter drone
- Context: pose and RGB camera image

Task

after the blue bale take a right towards the small white bush before the white bush take a right and head towards the right side of the banana

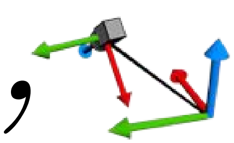
Mapping Instructions to Control

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

$$(v, \omega)$$

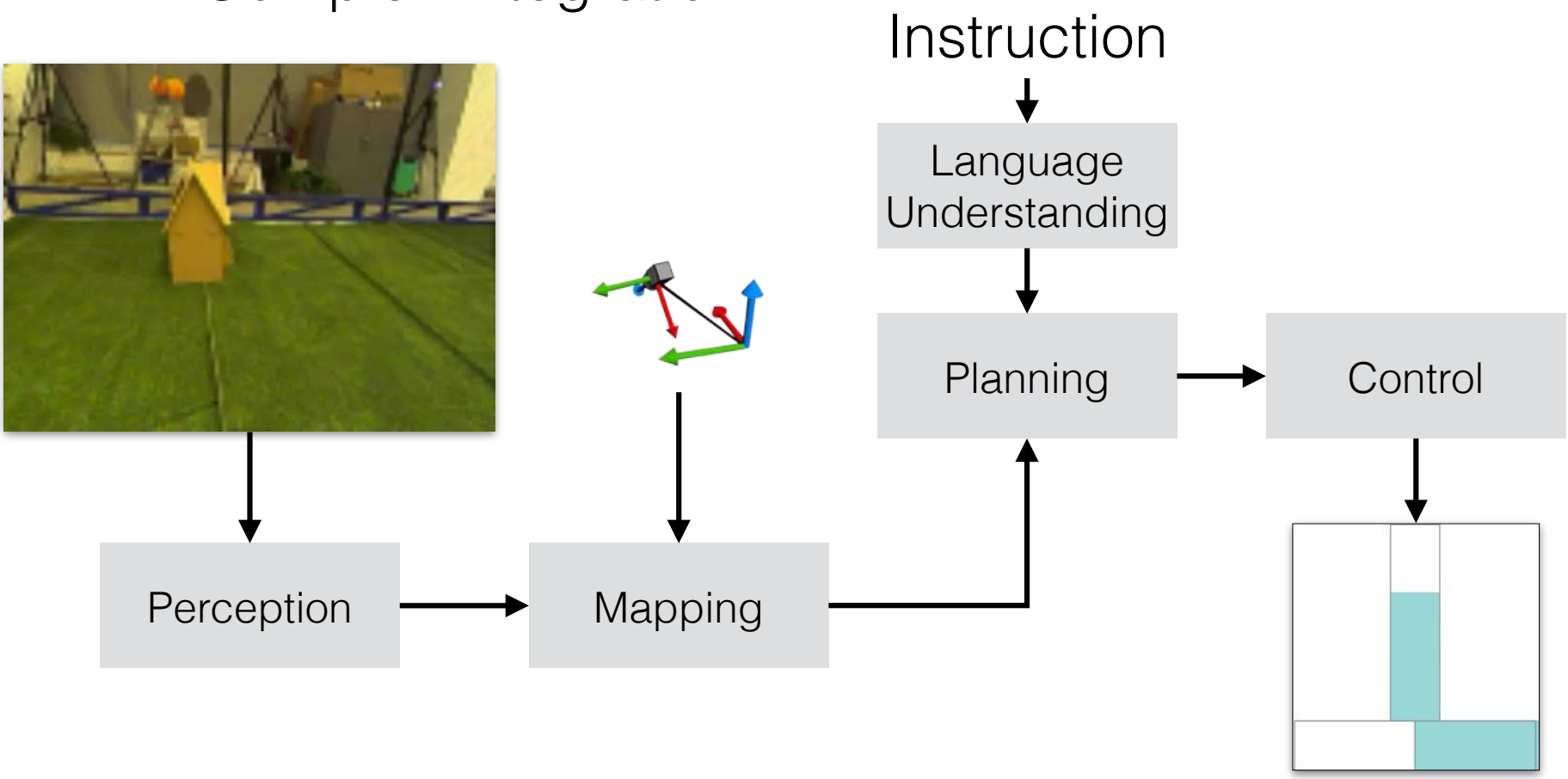
- Each action updates the configuration or stops

- Goal: learn a mapping from inputs to configuration updates

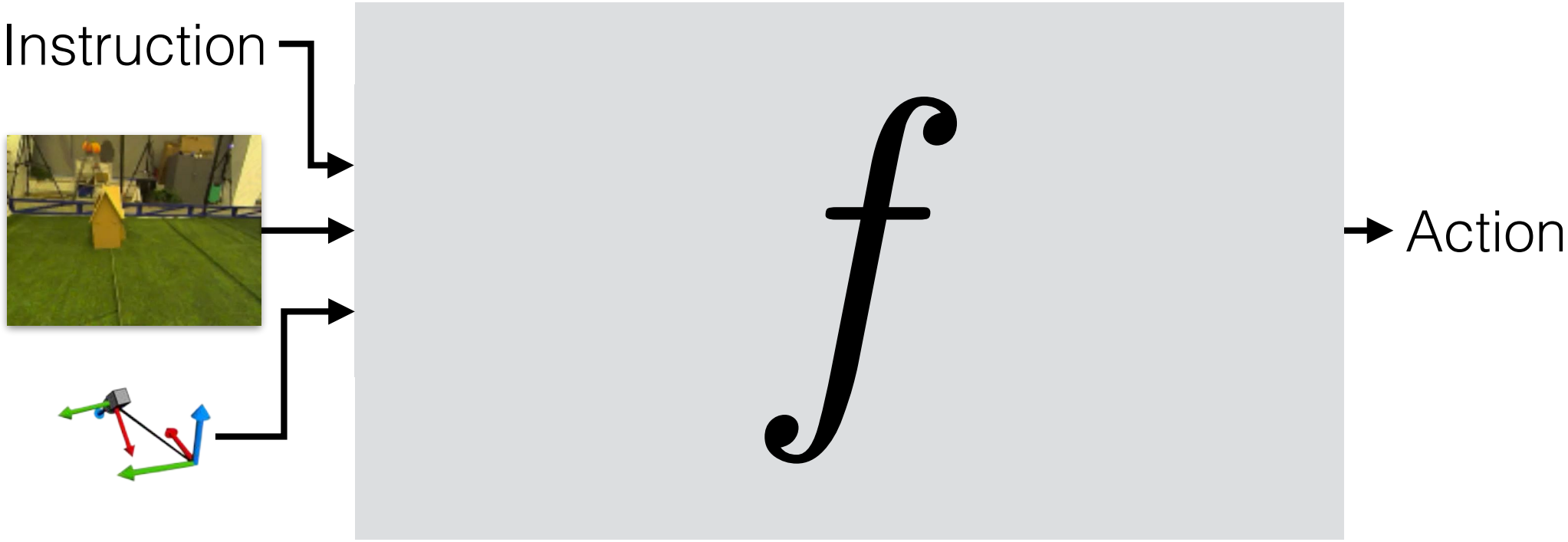
$f($ *after the blue bale take a right towards the small white bush before the white bush …* $,$  $,$  $) =$ 

Modular Approach

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

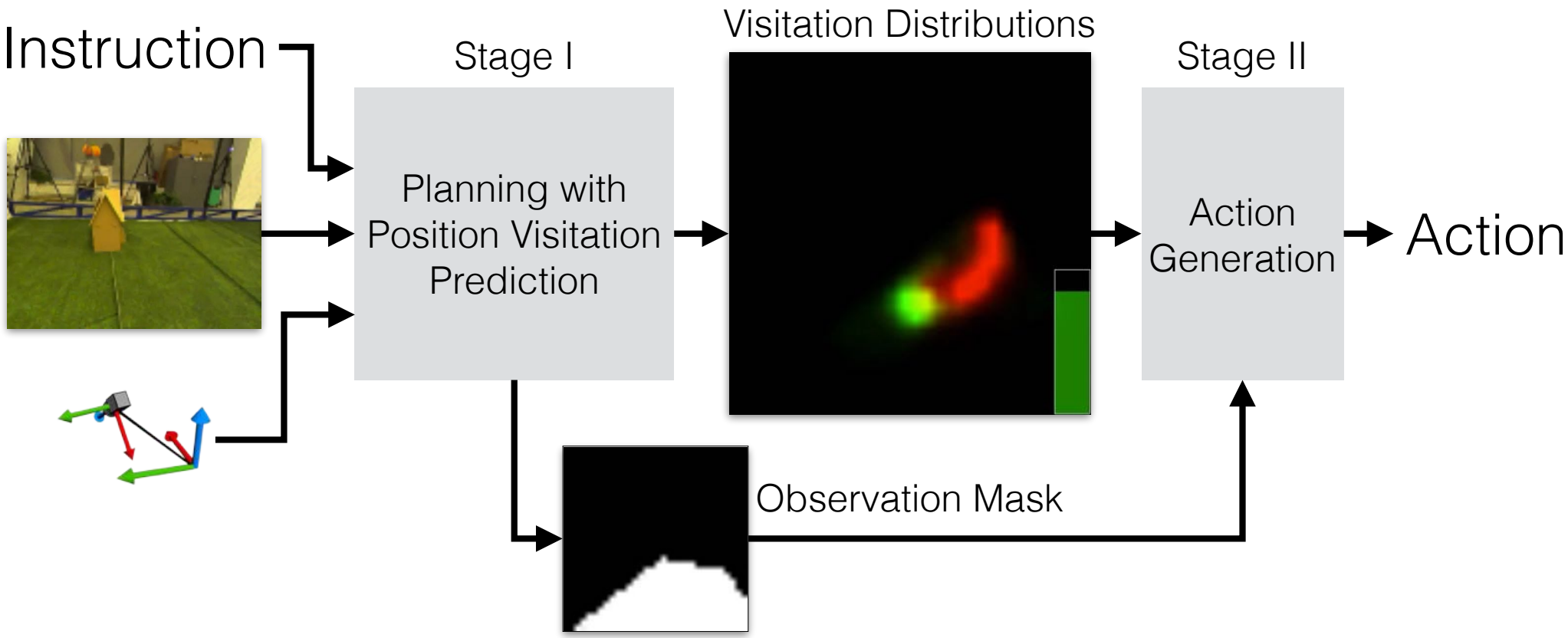


Single-model Approach



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

Single-model Approach



1. Predict states likely to visit and track accumulated observability

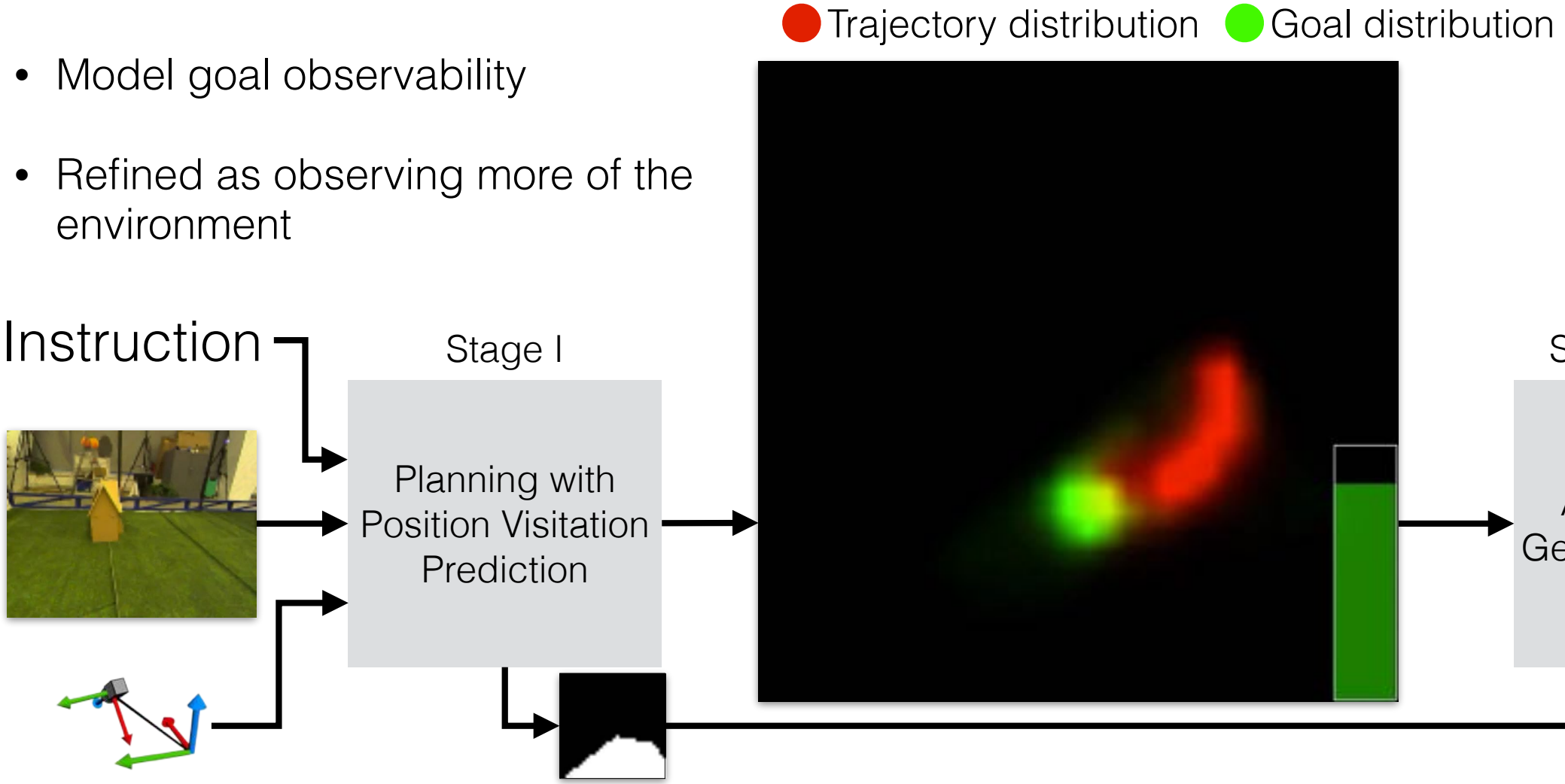
2. Generate actions to visit high-probability states and explore

Visitation Distribution

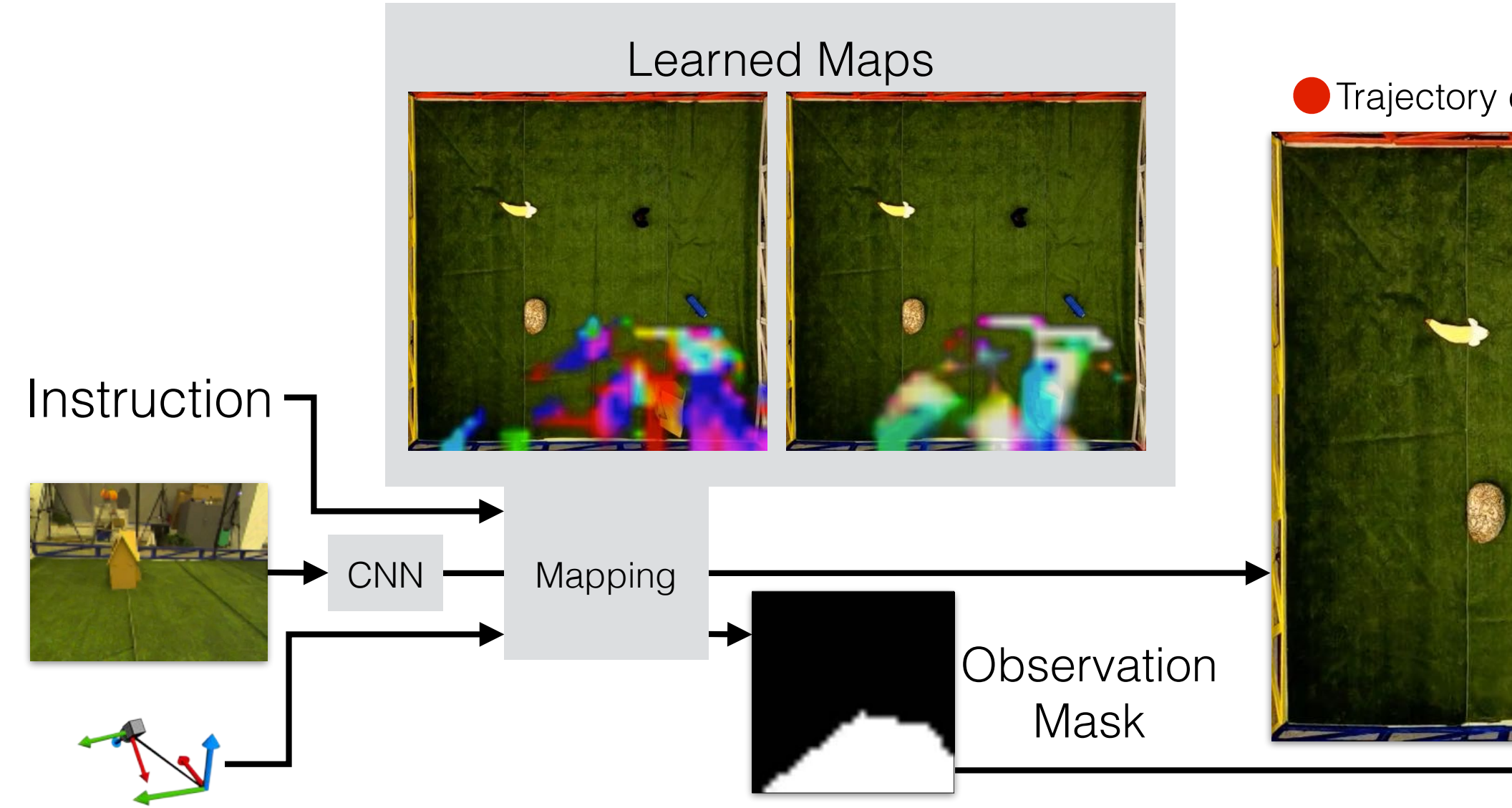
- The state-visitation distribution $d(s; \pi, s_0)$ is the probability of visiting state s following policy π from start state s_0
- Predicting $d(s; \pi^*, s_0)$ for an expert policy π^* tells us the states to visit to complete the task
- We compute two distributions: **trajectory-visitation** and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

Visitation Distribution for Navigation

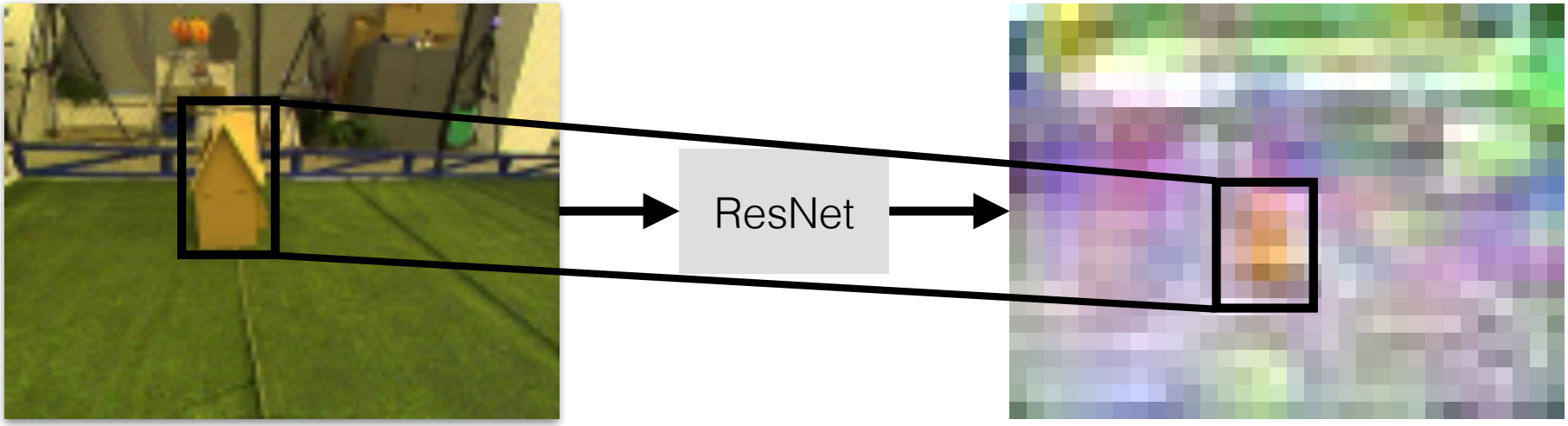
- Distributions reflect the agent plan
- Model goal observability
- Refined as observing more of the environment



Stage I: Planning with Position Visitation Prediction



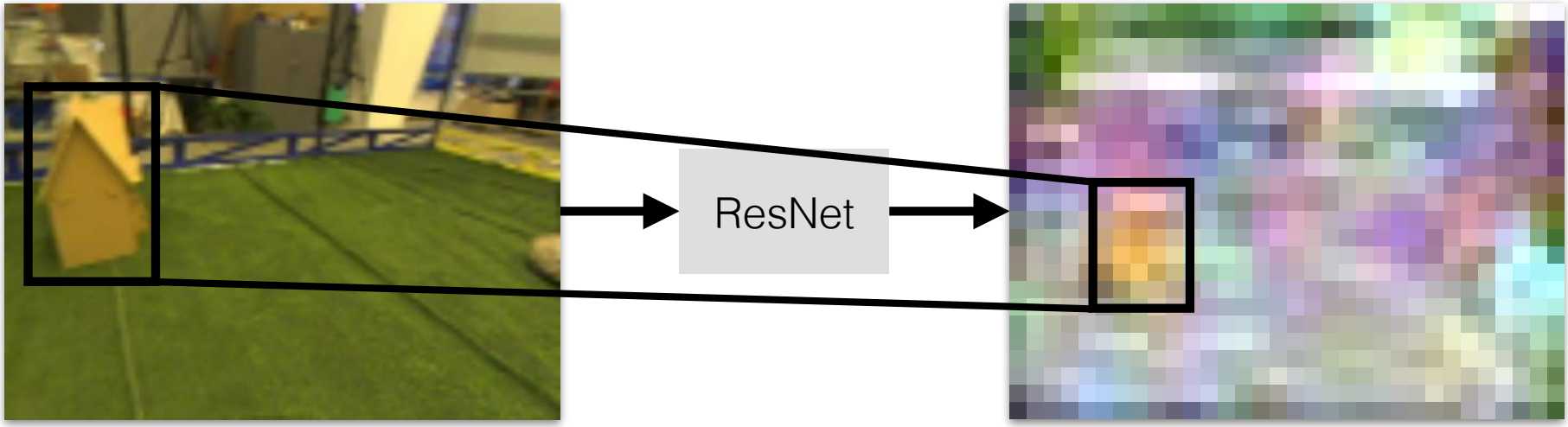
Differentiable Mapping
Step 1: Feature Extraction



- Extract features with a ResNet
- Recover a low resolution semantic view

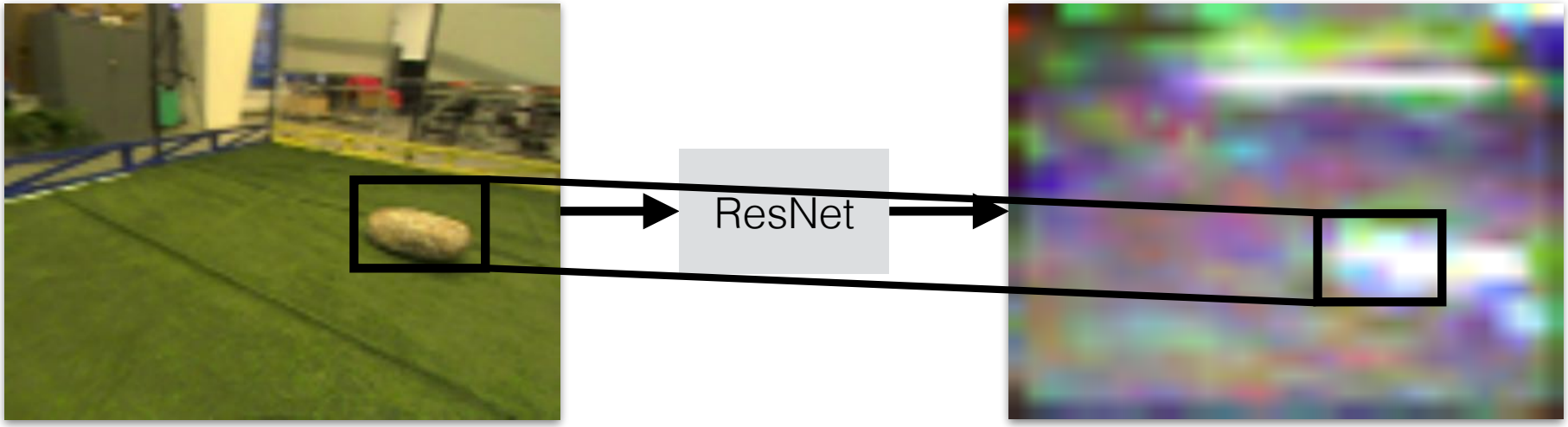
Differentiable Mapping

Step 1: Feature Extraction



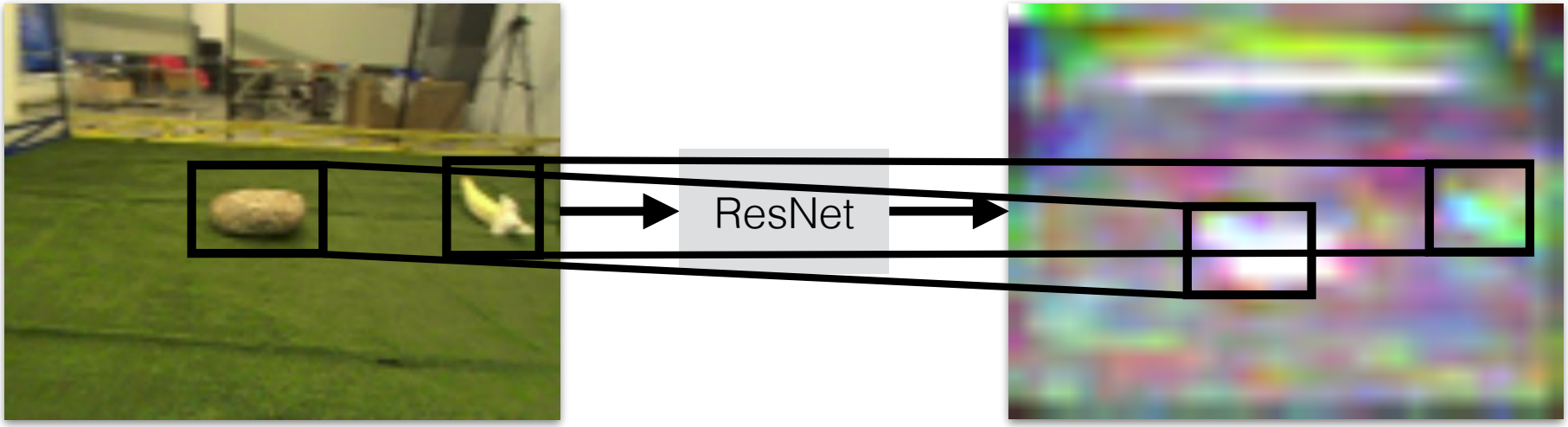
- Extract features with a ResNet
- Recover a low resolution semantic view

Differentiable Mapping
Step 1: Feature Extraction



- Extract features with a ResNet
- Recover a low resolution semantic view

Differentiable Mapping
Step 1: Feature Extraction

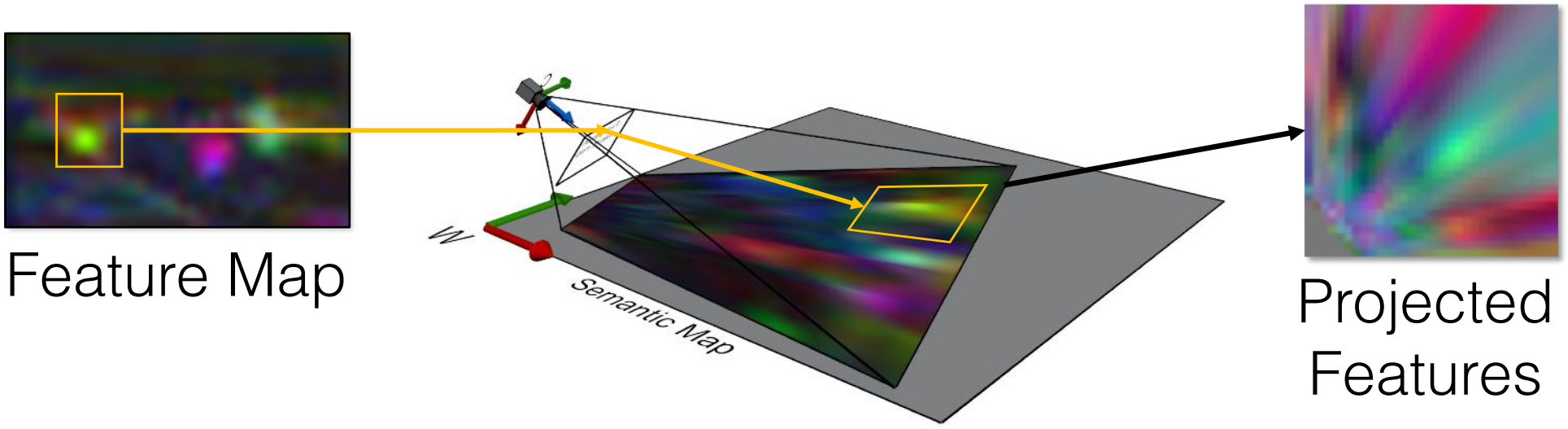


- Extract features with a ResNet
- Recover a low resolution semantic view

Differentiable Mapping

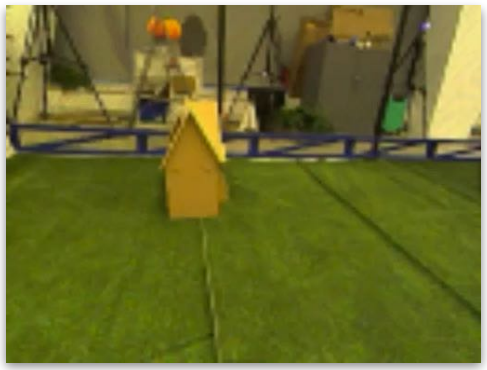
Step 2: Projection

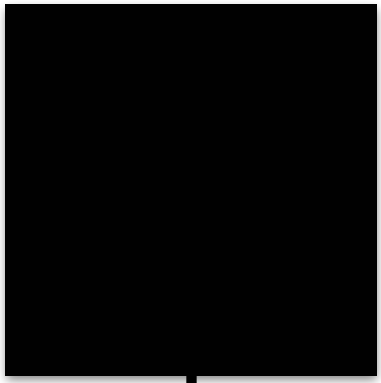
- Deterministic projection from camera image plane to environment ground with pinhole camera model
- Transform from first-person to third-person



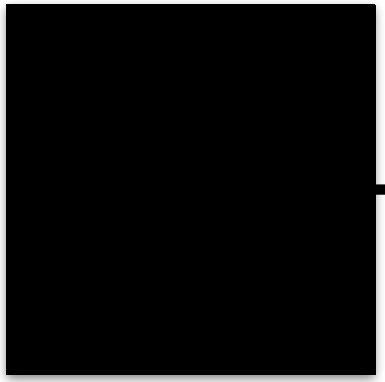
Differentiable Mapping

Step 3: Accumulation





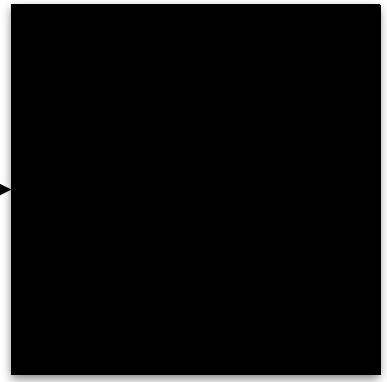
Semantic Map
(time t-1)



Projected Features
(time t)

Σ

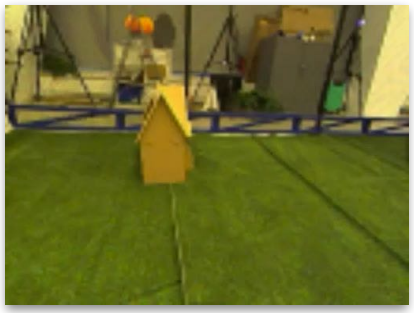
Leaky
Integrator

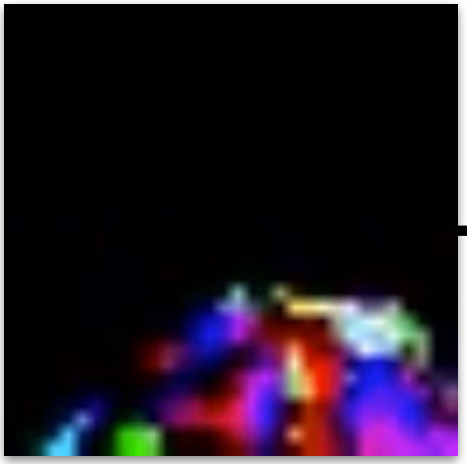


Semantic Map
(time t)

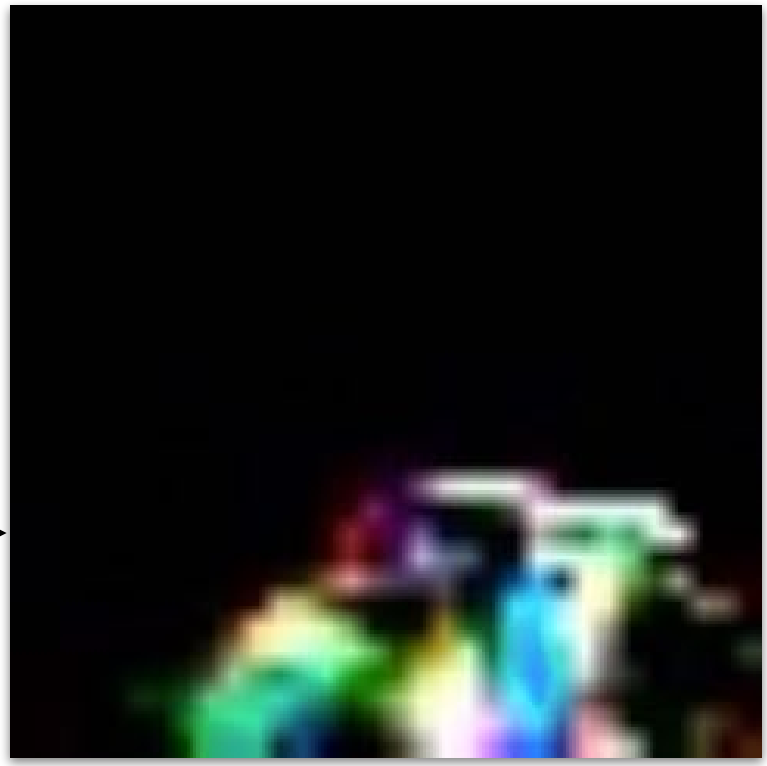
Differentiable Mapping

Step 4: Grounding





1x1 Filter

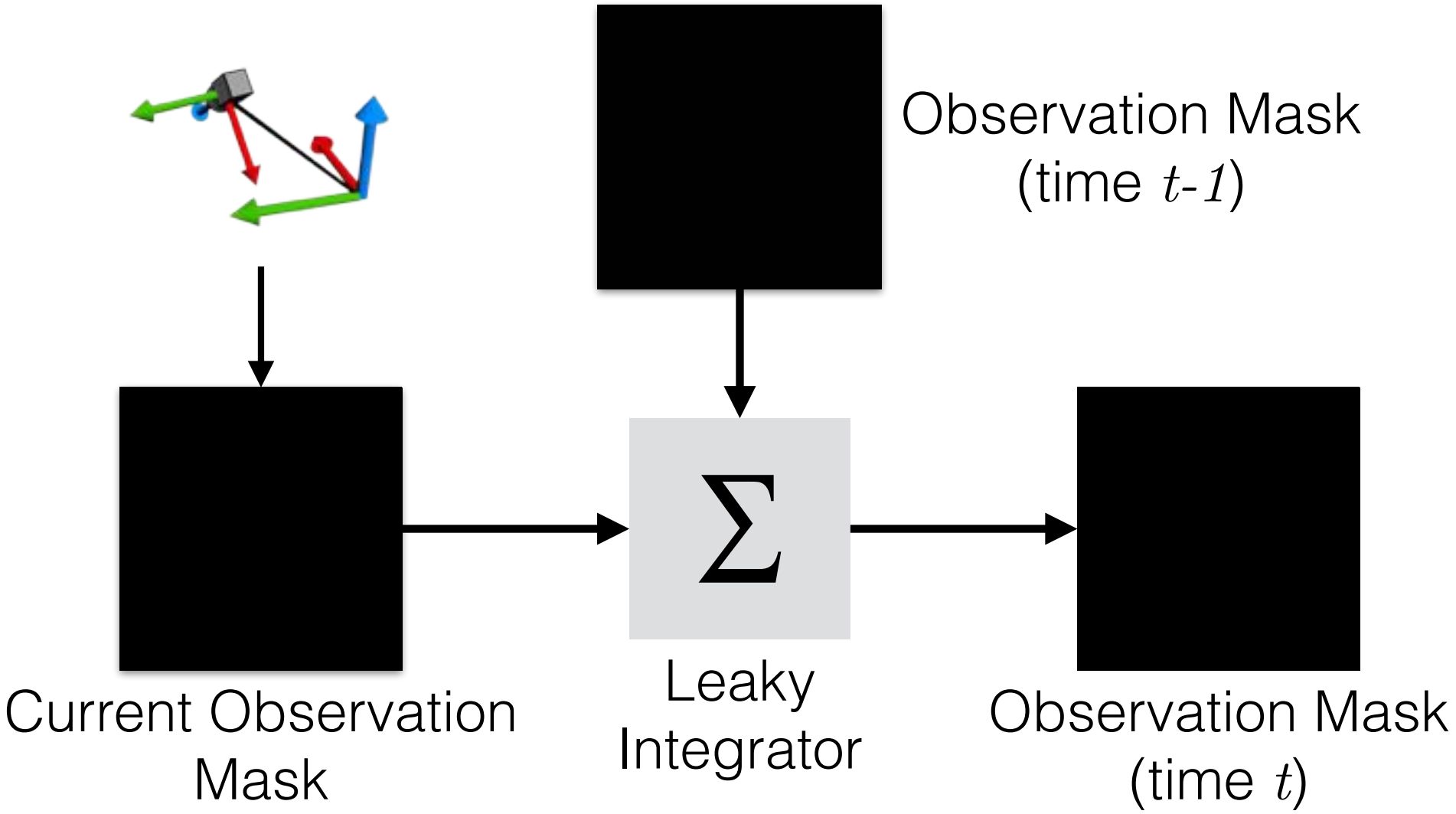


Semantic Map

Grounding Map

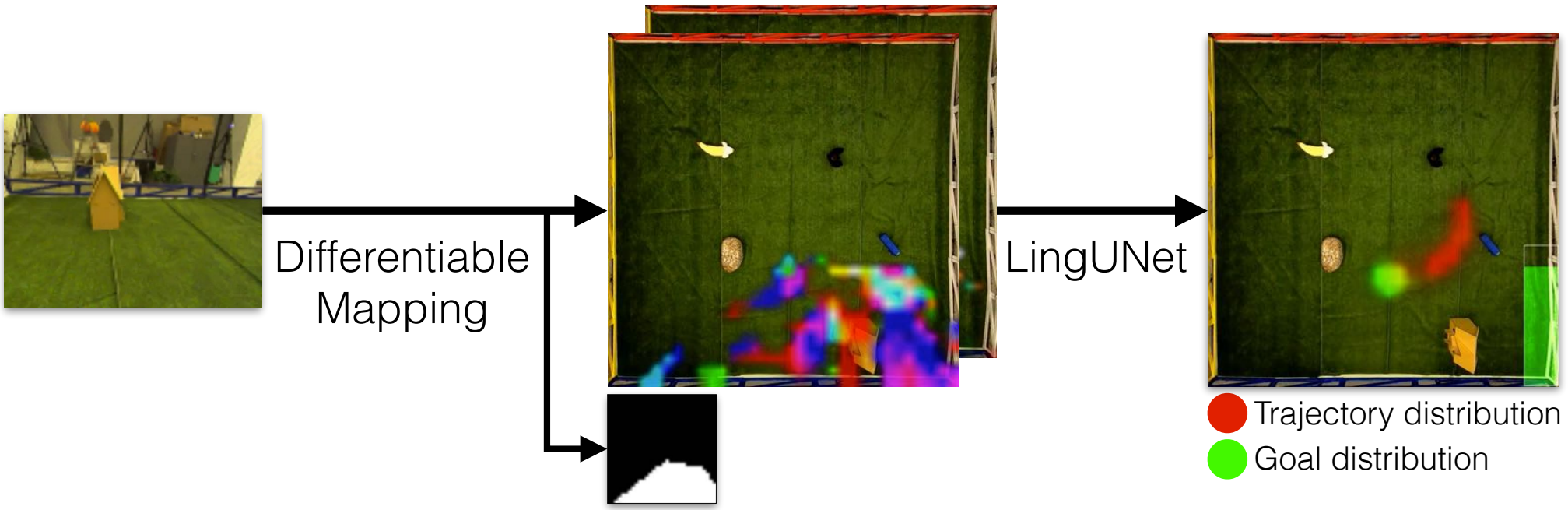
after the blue bale take a right towards the small white bush before the white bush …

Observability Mask



Stage I: Planning with Position Visitation Prediction

- ✓ Extract visual features and construct maps
- • Compute visitation distributions over the maps

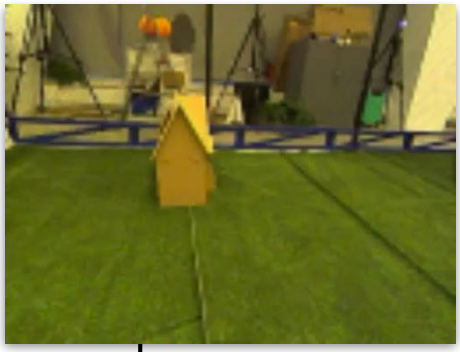


Predicting Visitation Distributions

- We compute two distributions: **trajectory-visitation** and **goal-visitation**
- Cast distribution prediction as image generation

LingUNet

- Image-to-image encoder-decoder
- Visual reasoning at multiple image scales
- Conditioned on language input at all levels of reasoning using text-based convolutions

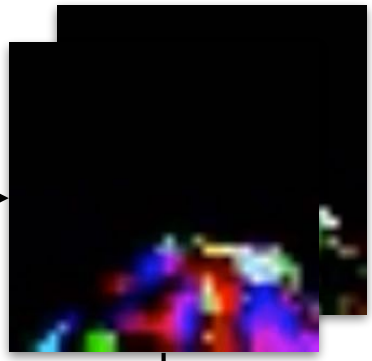


LingUNet

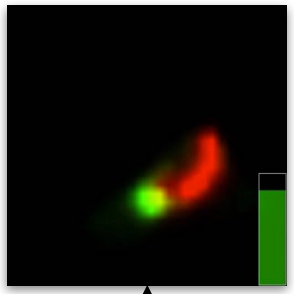
Semantic Maps

Instruction

Visitation Distributions



RNN



SoftMax

Text Kernels

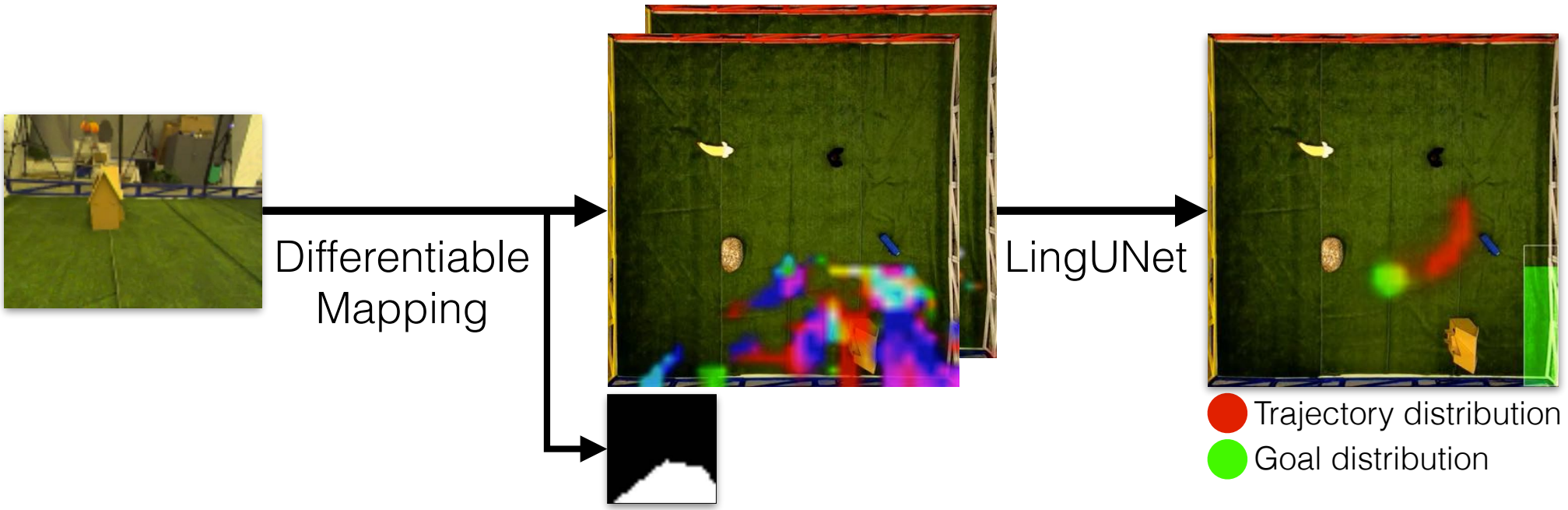
Convolutions

Deconvolutions

Text Convolutions

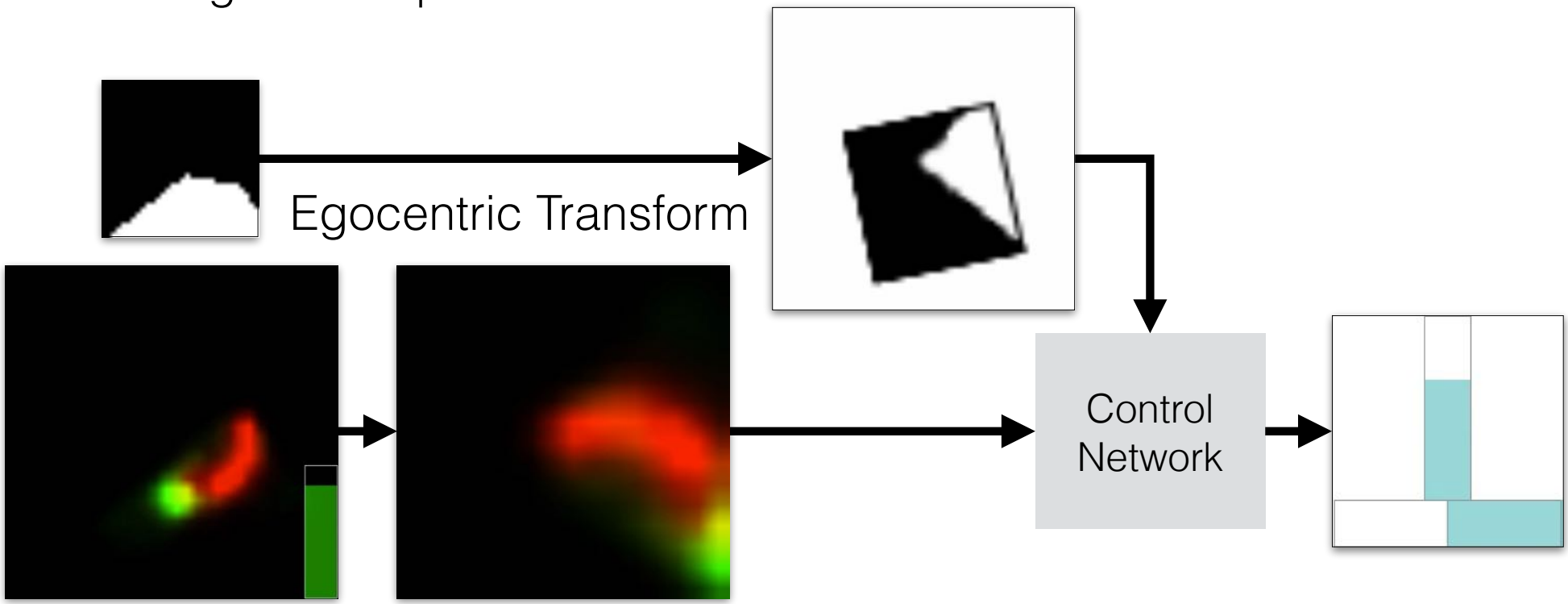
Stage I: Planning with Position Visitation Prediction

- ✓ Extract visual features and construct maps
- ✓ Compute visitation distributions over the maps



Stage II: Action Generation

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



Learning vs. Engineering

Learned 👨‍🎓

- Visual features (ResNet)
- Text representation (RNN)
- Image generation (LingUNet)
- Control (control network)

Engineered 👷

- Feature projections (pinhole camera model)
- Map accumulation (leaky integrator)
- Egocentric transformation (matrix rotations)

Complete network remains fully differentiable

Simulation-Reality Joint Learning

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

after the blue bale take a right towards the small white bush before the white bush take a right and head towards the right side of the banana

Training Data

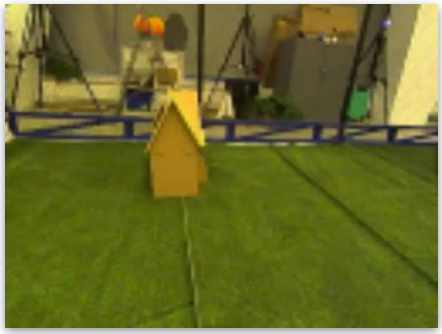
Simulator

Demonstrations and simulator

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

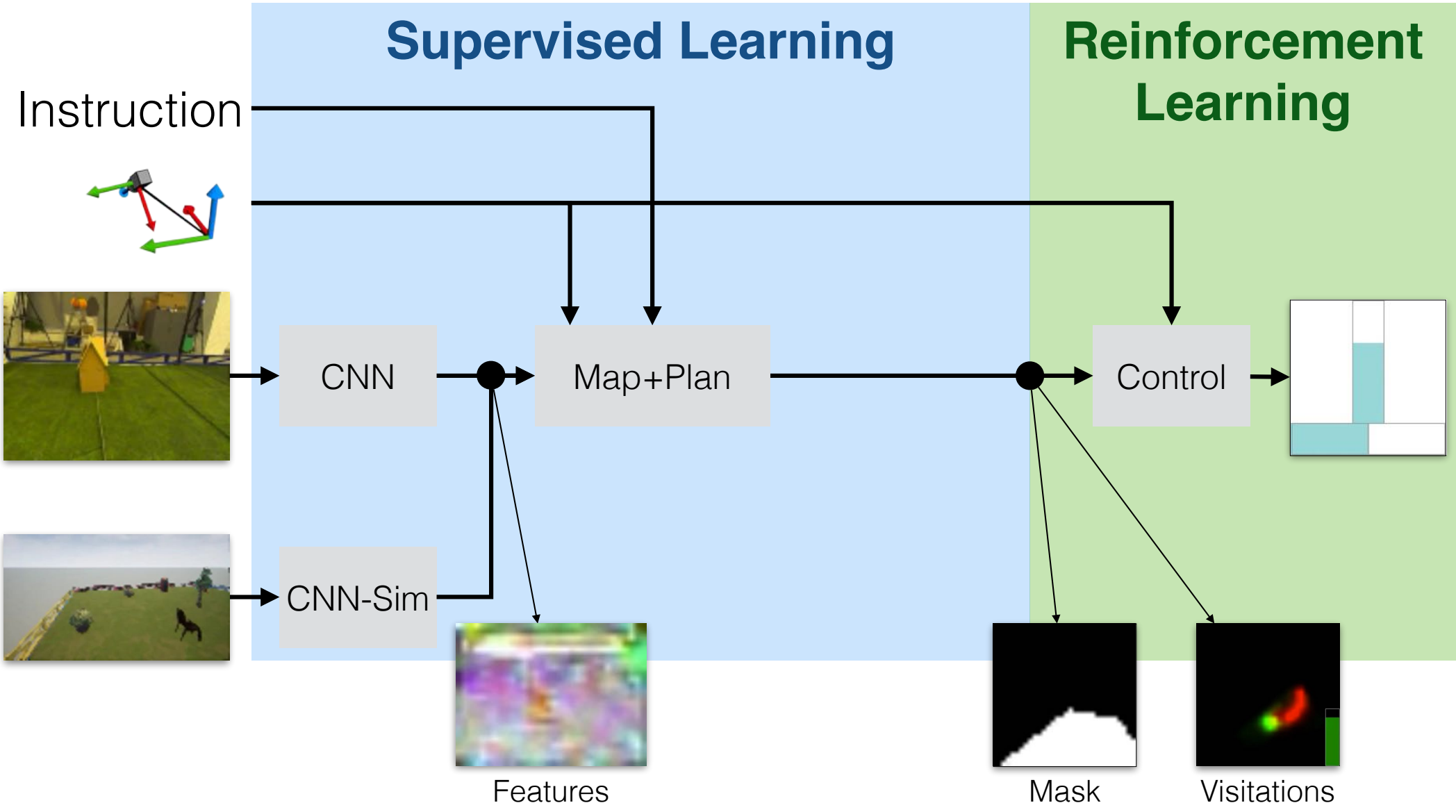
Physical Environment

Demonstrations only



after the blue bale take a right towards the small white bush before the white bush take a right and head towards the right side of the banana

Learning Architecture

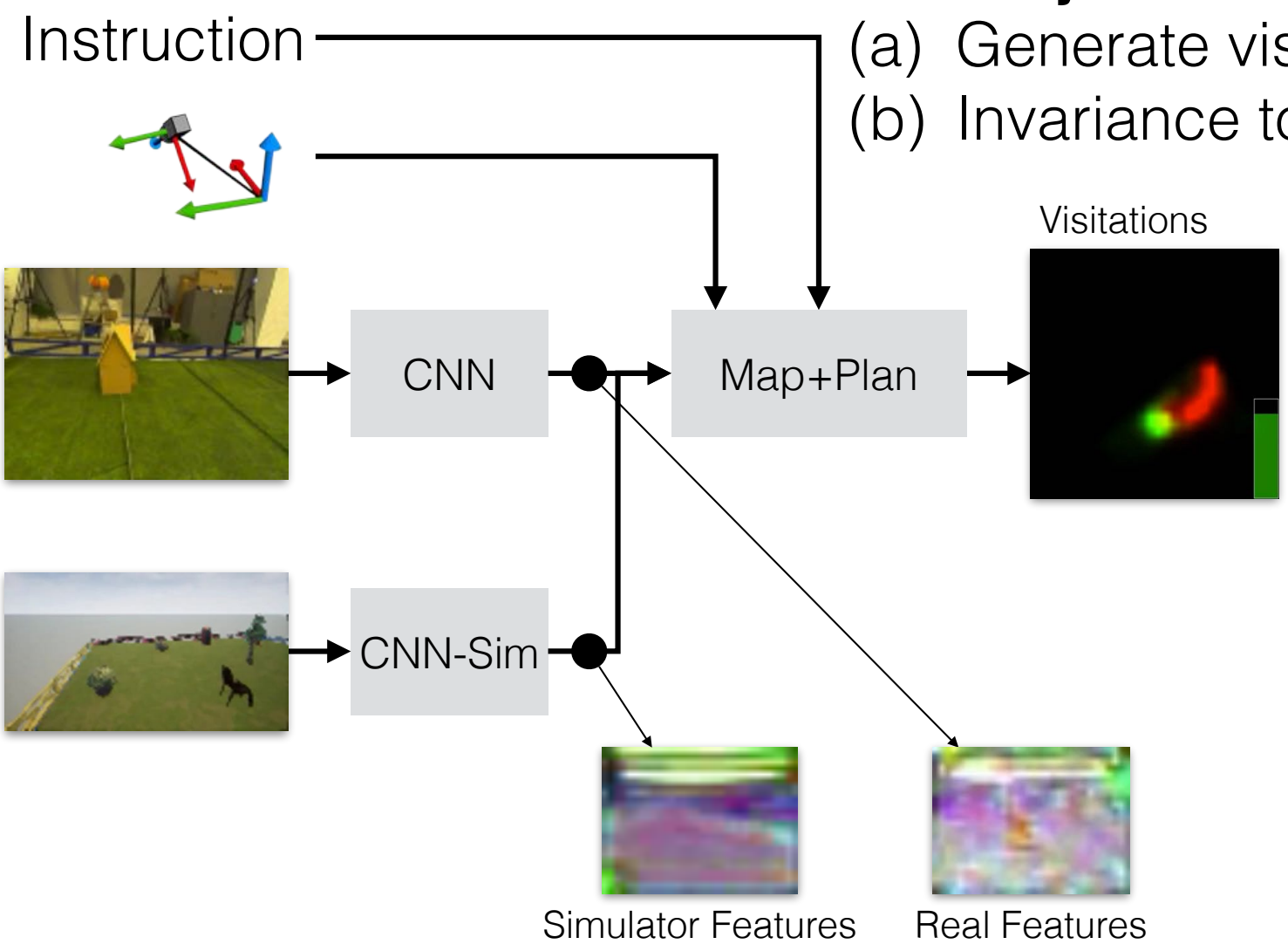


Supervised Plan Learning

Two objectives:

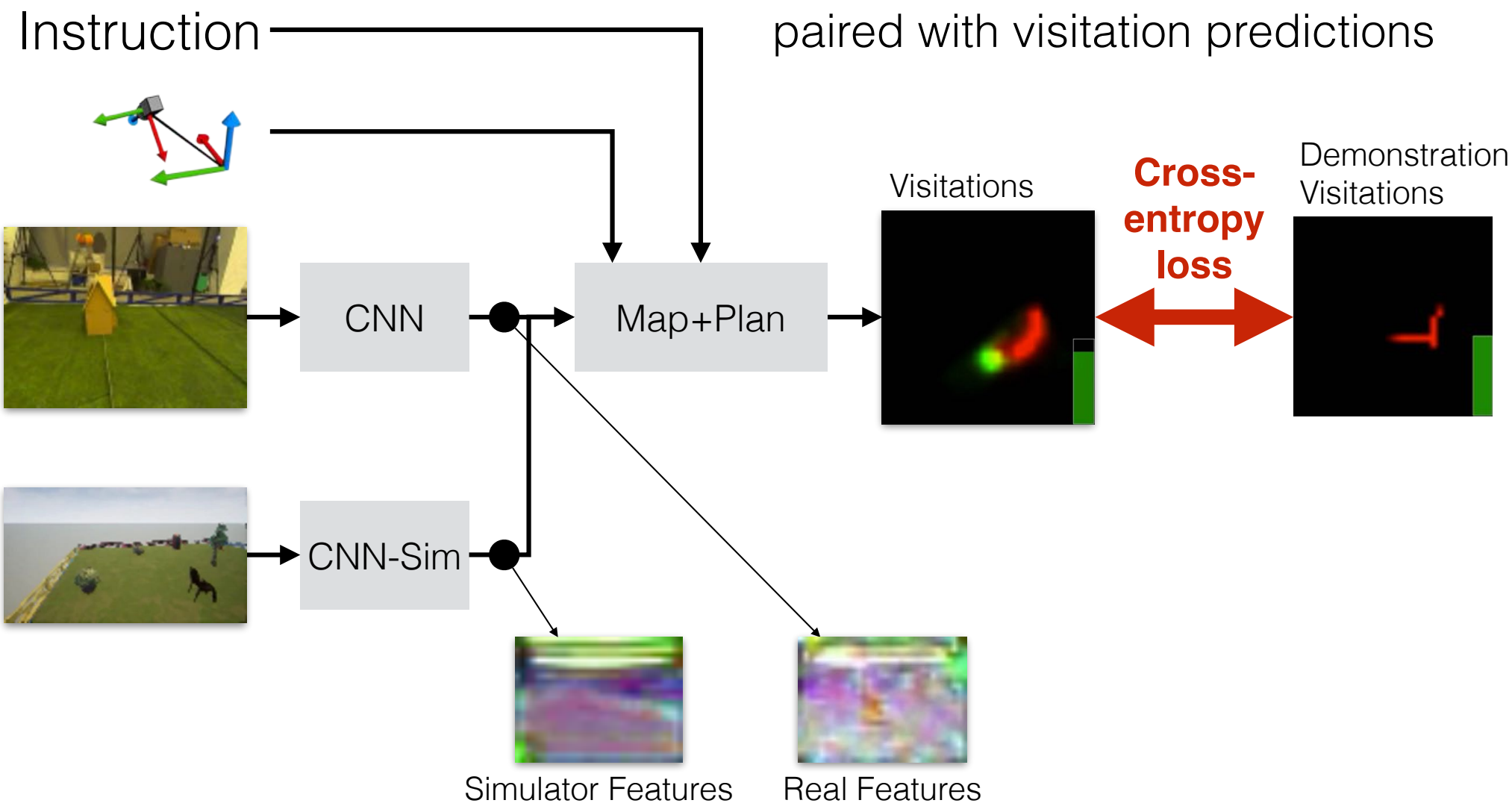
(a) Generate visitation distributions

(b) Invariance to input environment



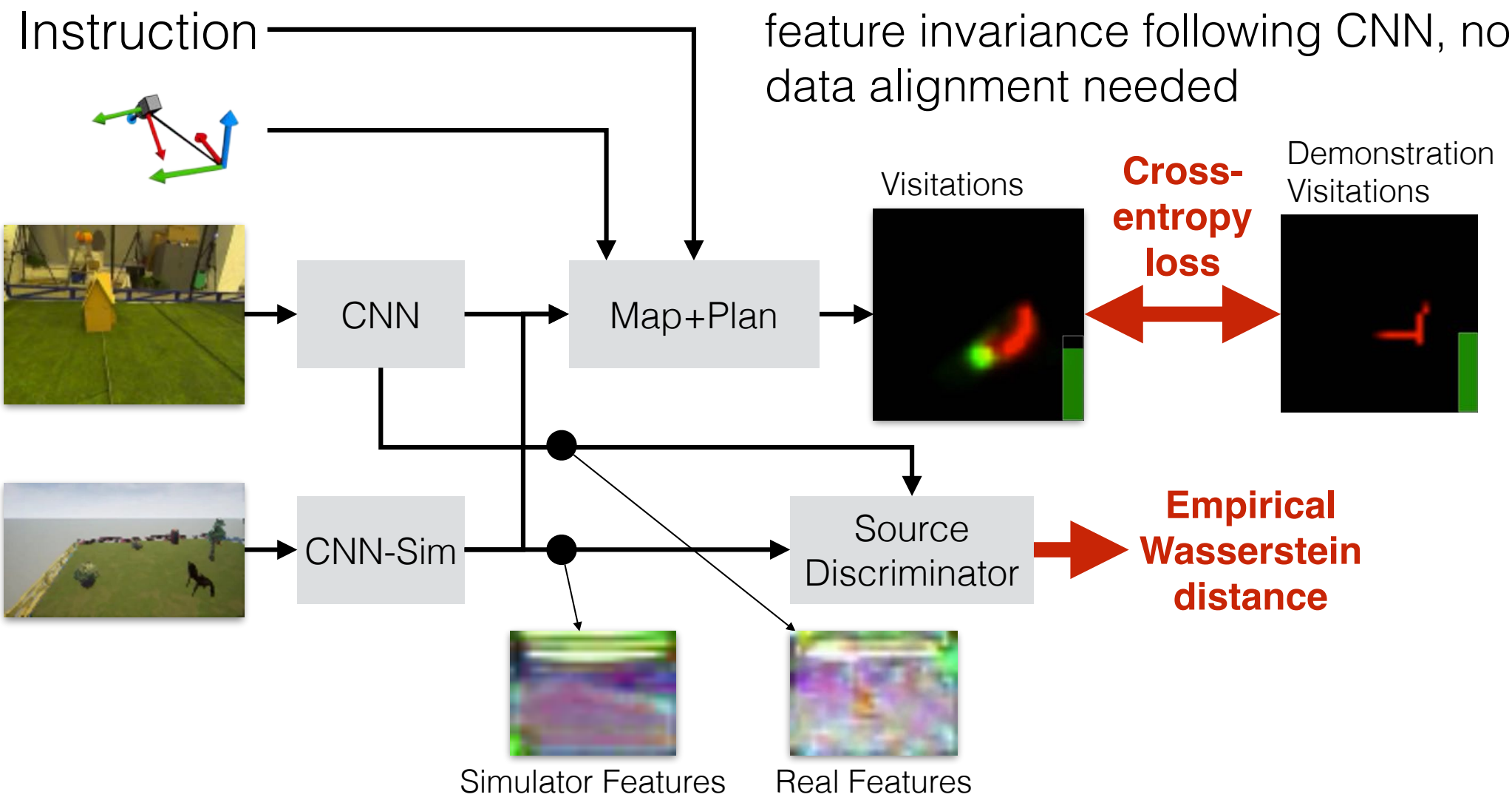
Supervised Plan Learning

Data: real and simulated states paired with visitation predictions

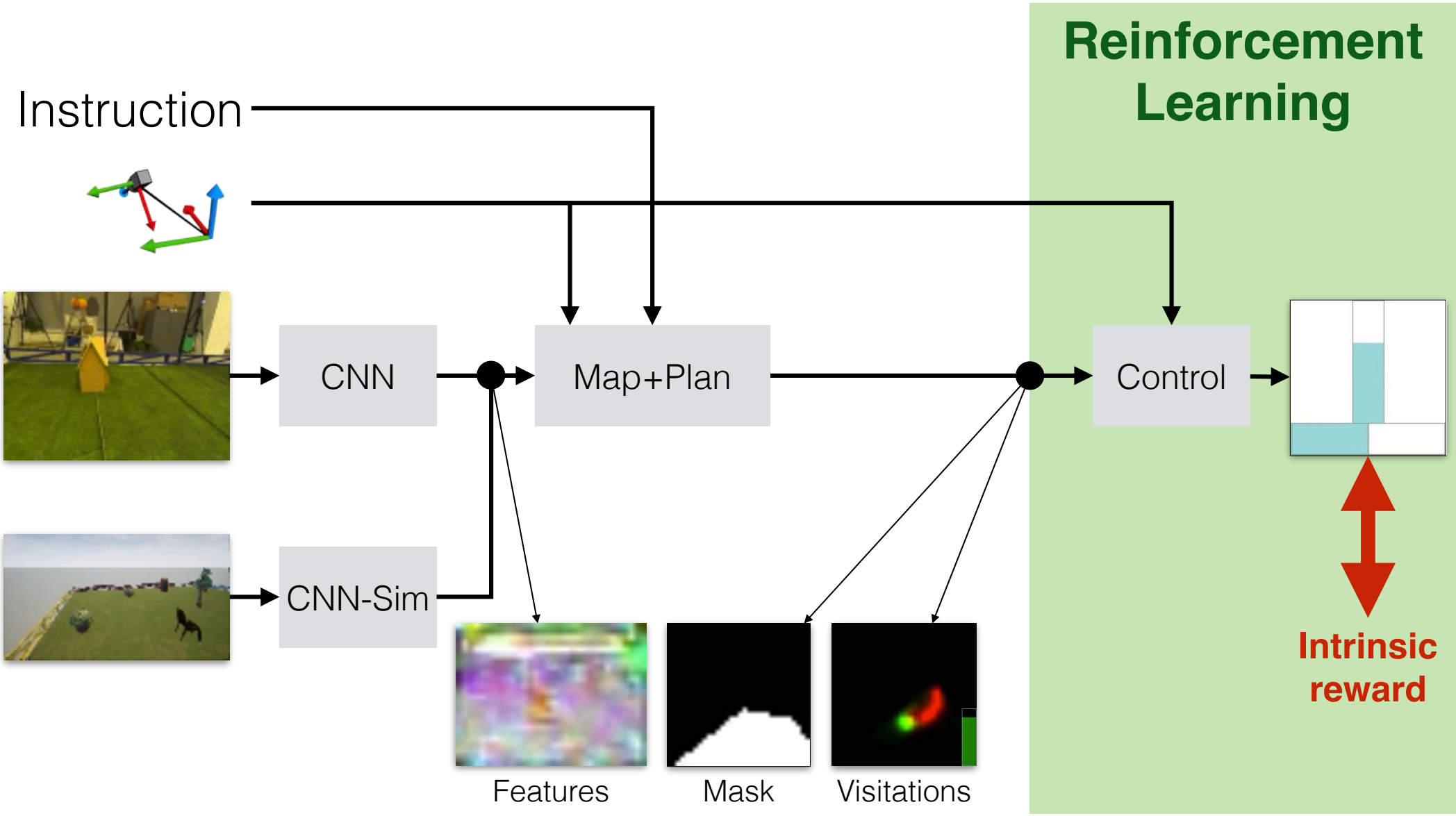


Supervised Plan Learning

Adversarial discriminator to force feature invariance following CNN, no data alignment needed

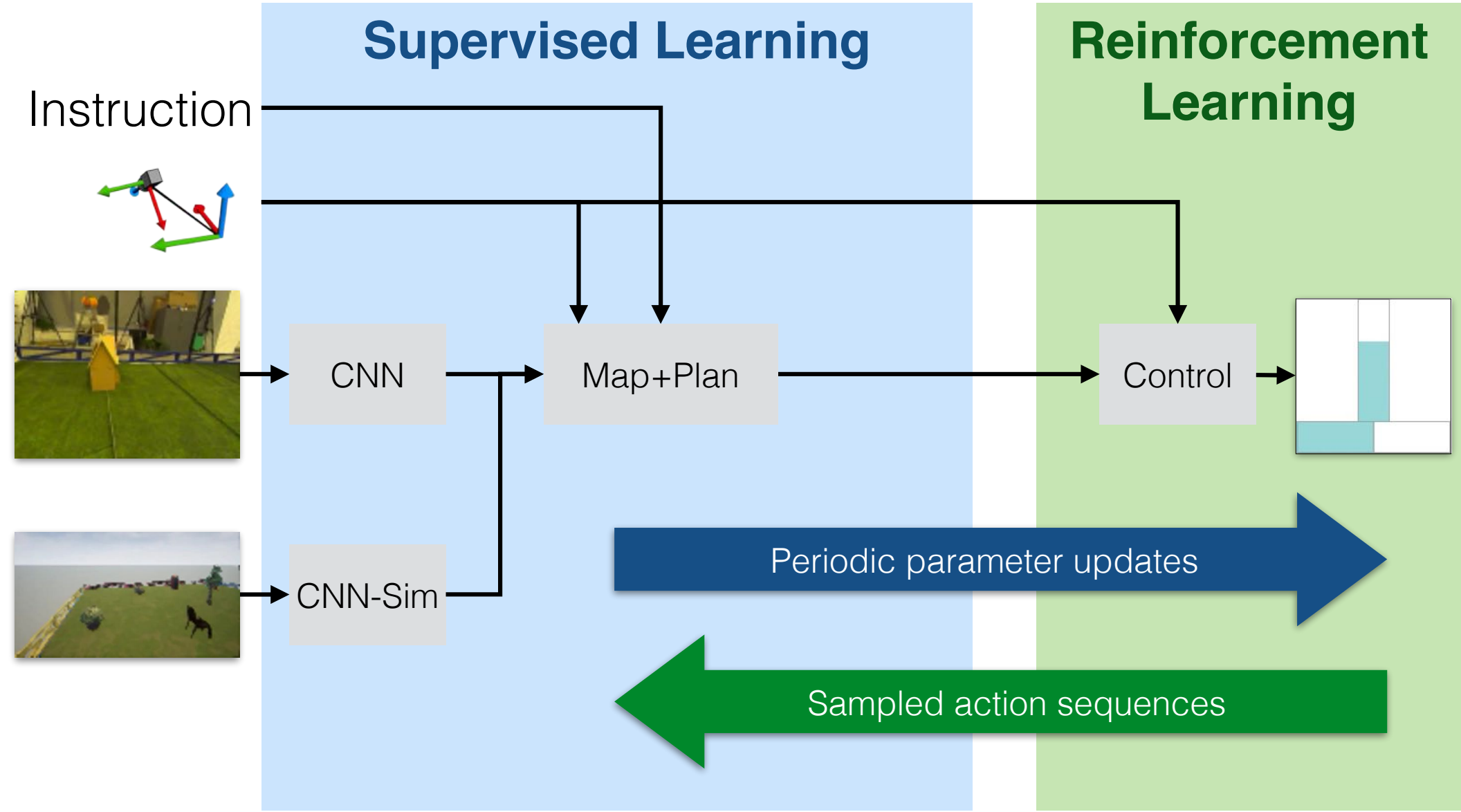


RL for Control



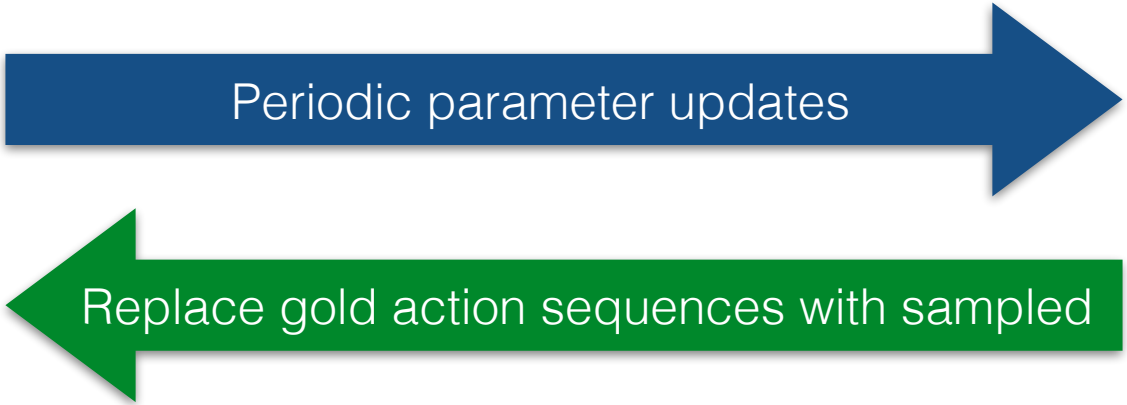
SuReAL

Supervised and Reinforcement Asynchronous Learning



SuReAL

Supervised and Reinforcement Asynchronous Learning

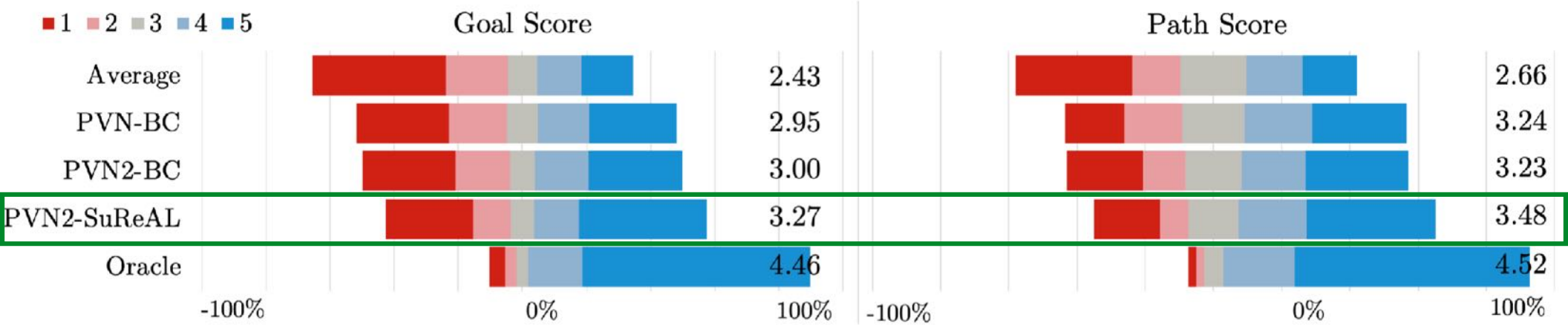


- **Stage I:** learn to predict visitation distributions based on noisy predicted execution trajectories
- **Stage II:** learn to predict actions using predicted visitation distributions

Experimental Setup

- Intel Aero quadcopter
- Vicon motion capture for pose estimate
- Simulation with Microsoft AirSim
- Drone cage is 4.7x4.7m
- Roughly 1.5% of training data in physical environment (402 vs. 23k examples)

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

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

The Papers

- **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.

Today

Mapping instructions to continuous control

- Generating and executing interpretable plans
  Casting planning as visitation distribution prediction and decomposing the policy to tie action generation to spatial reasoning

- Jointly learning in real life and a simulation
  Learning with an adversarial discriminator to train domain-invariant perception

- Training for test-time exploration
  SuReAL training to combine the benefits of supervised and reinforcement learning

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]

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

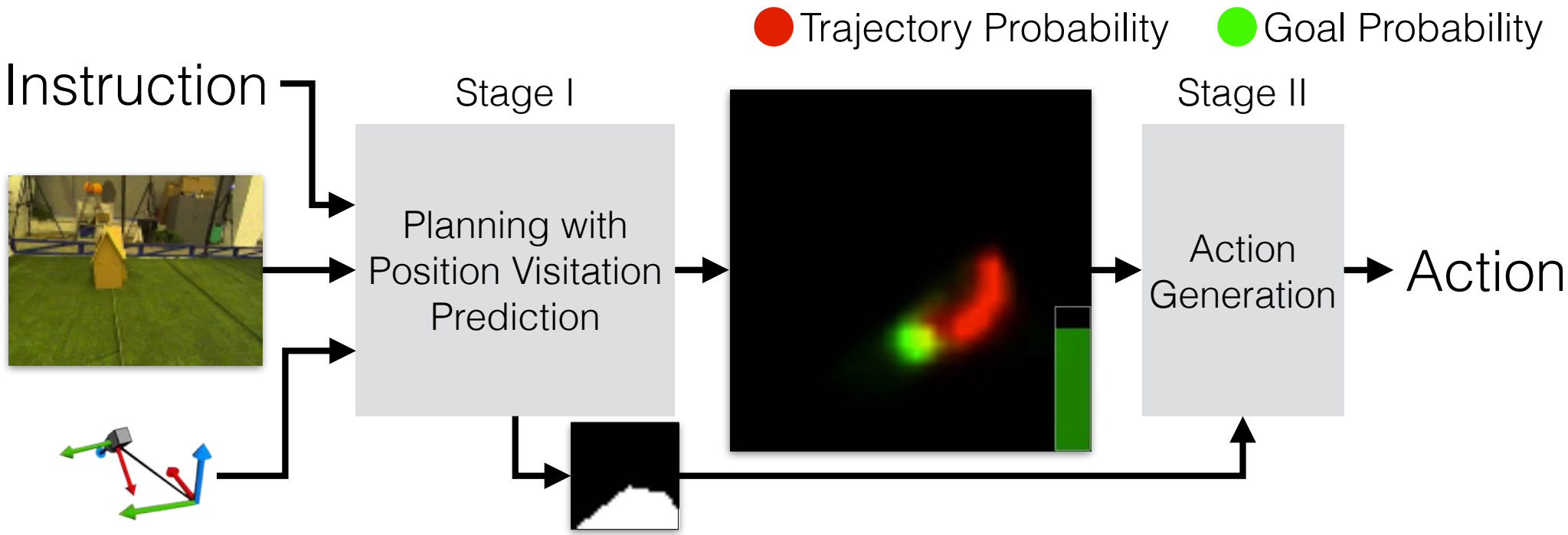
MDP S States A Actions R Reward H Horizon

- 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

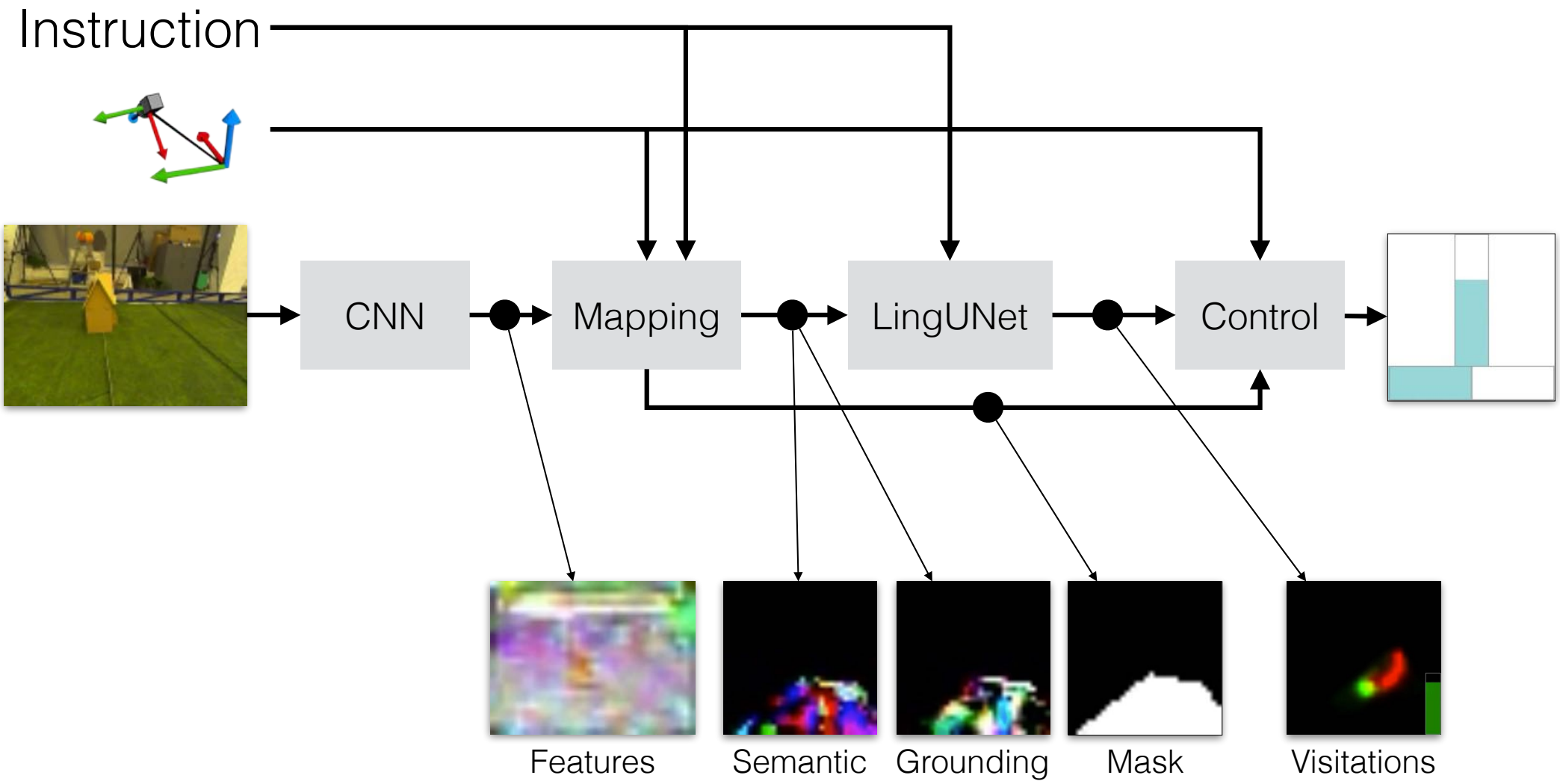
- $\tilde{S}$ is a set of discrete positions in the world
- We compute two distributions: **trajectory-visitation** and **goal-visitation**



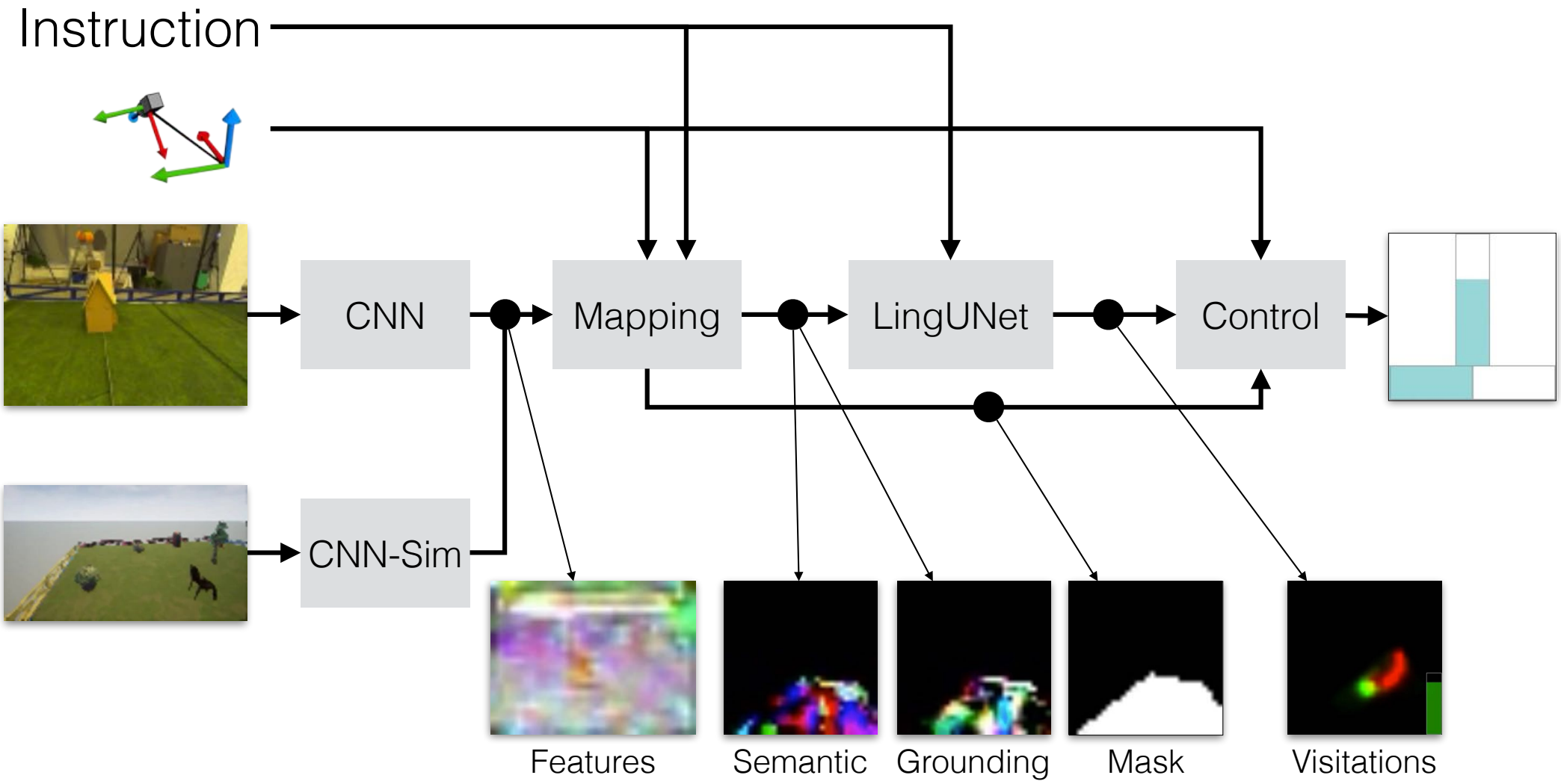
Drone Model

Drone Learning

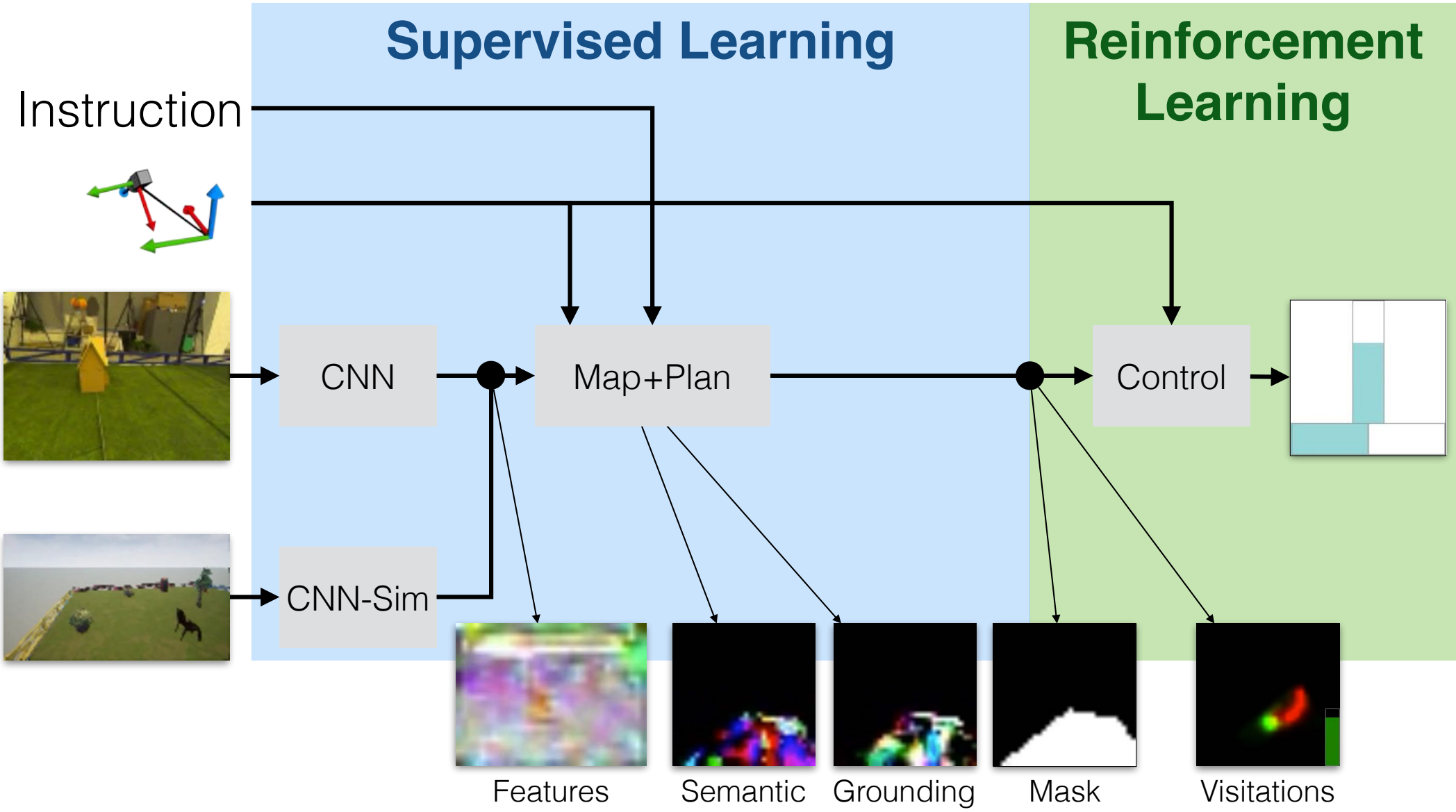
Learning Architecture



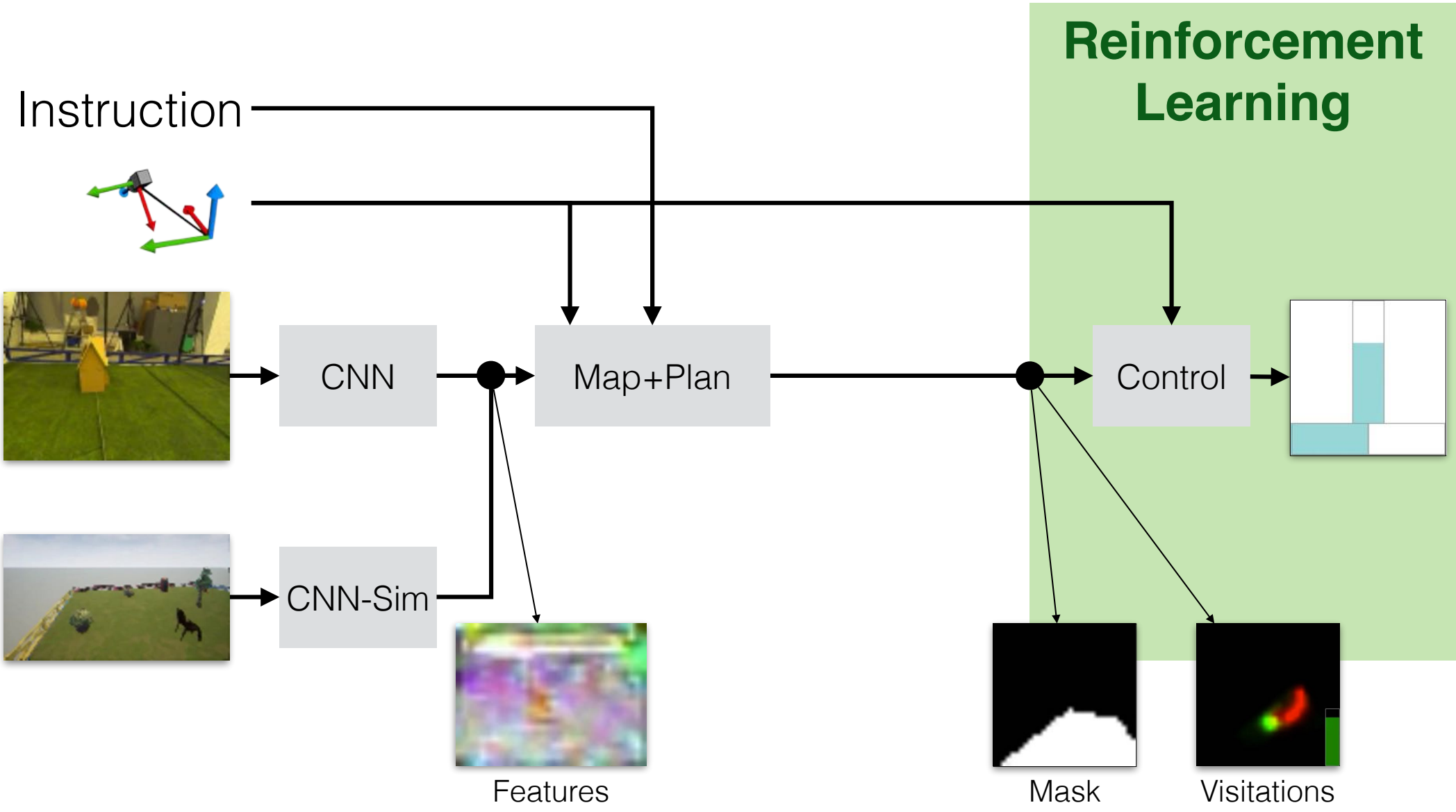
Learning Architecture



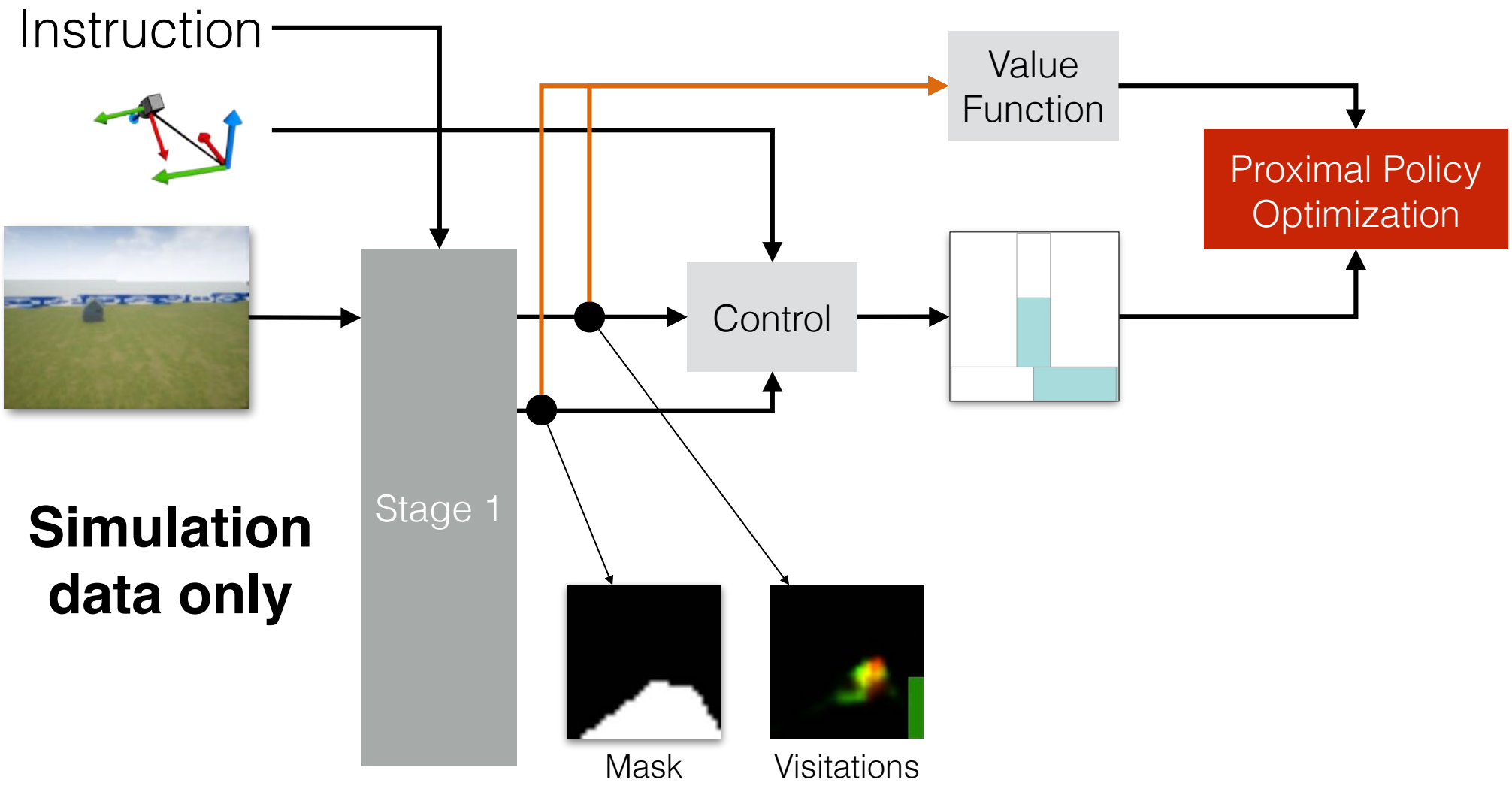
Learning Architecture



RL for Control



RL for Control



Reward Goals

- We want the agent to:
  - Follow the plan
  - Explore to find the goal if not observed
  - Only select feasible actions
  - Be efficient

Reward

- **Visitation:** reduction in earth mover's distance between the predicted trajectory distribution and what has been done so far
- **Stop:** if stopping, the earth mover's distance between the stop location and the predicted goal distribution
- **Exploration:** reward reducing the probability that the goal has not been observed, and penalize stopping when reward not observed
- **Action:** penalize actions outside of controller range
- **Step:** constant step verbosity term to encourage efficient execution

Drone Related Work

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

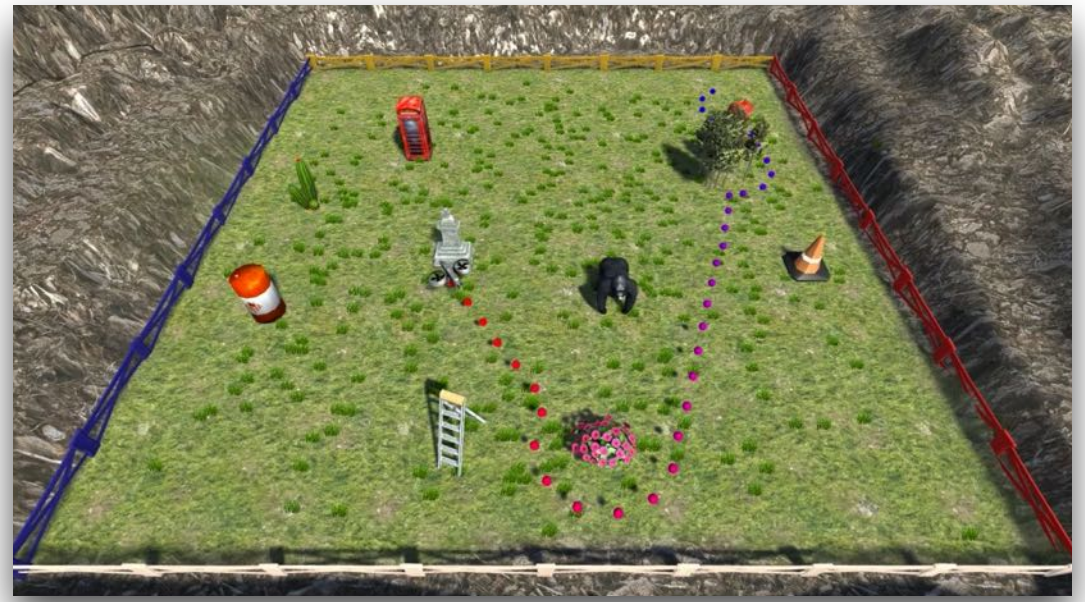
- Learning to explore

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

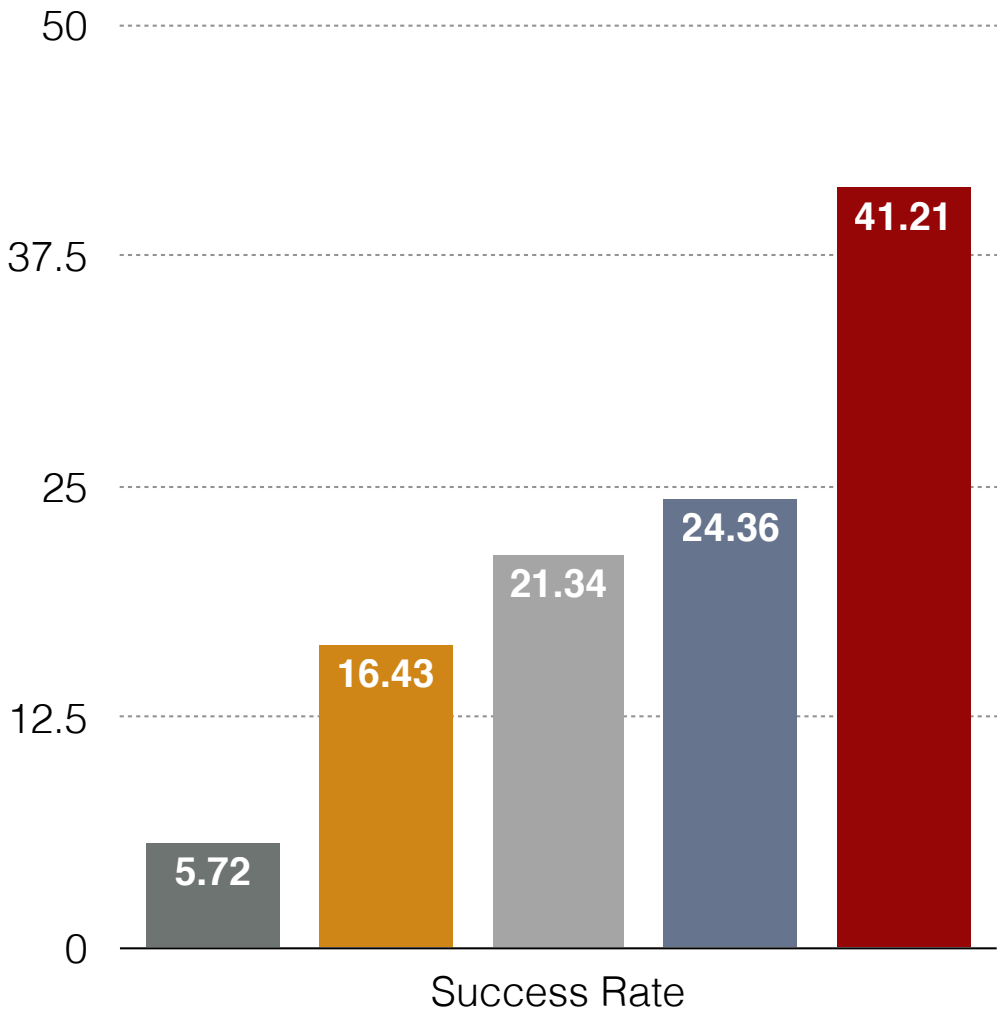
Go towards the pink flowers and pass them on your left, between them and the ladder. Go left around the flower until you're pointed towards the bush, going between the gorilla and the traffic cone. Go around the bush, and go in between it and the apple, with the apple on your right. Turn right and go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

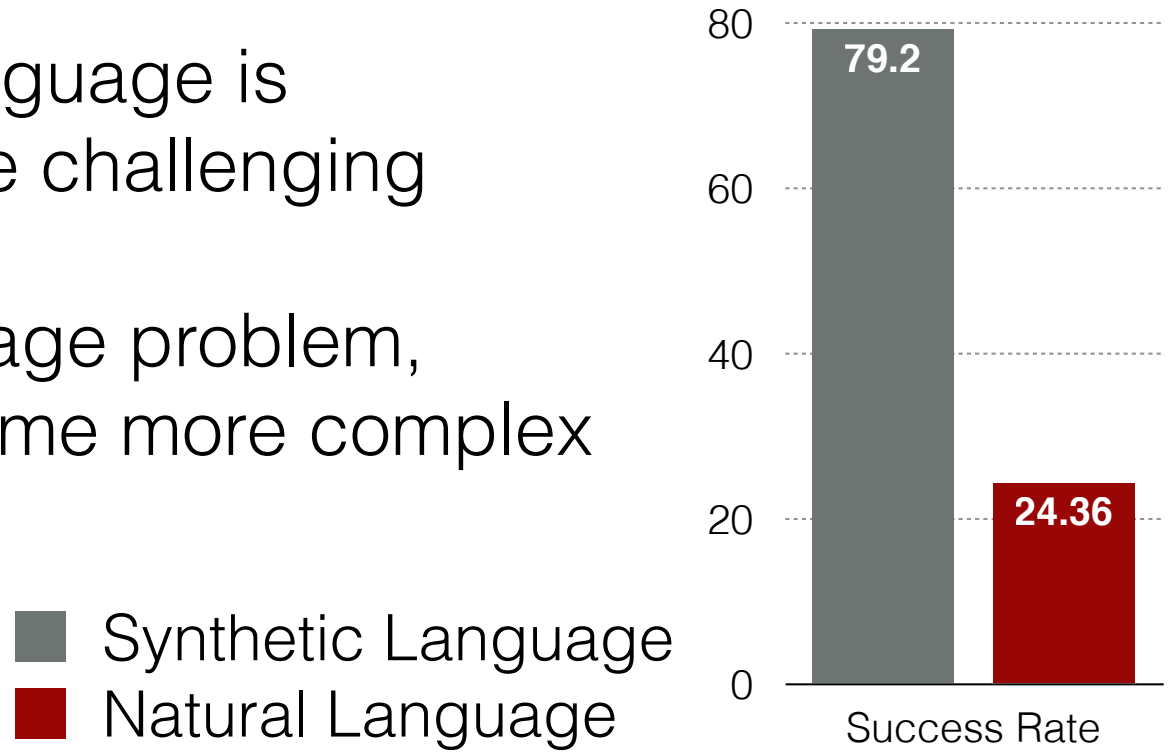


STOP
Average
Chaplot et al. 2018
Blukis et al. 2018
Our Approach

- Explicit mapping helps performance
- Explicit planning further improves performance

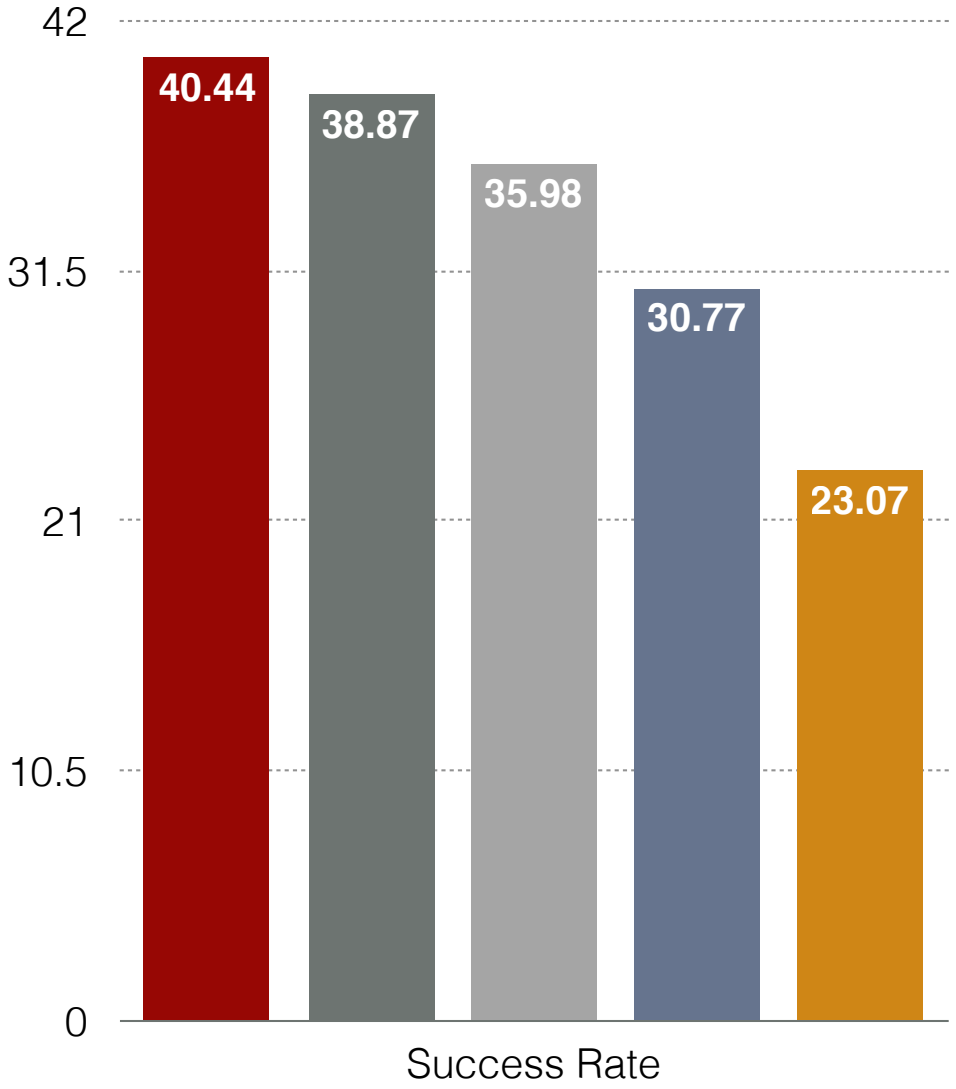
Synthetic vs. Natural Language

- 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

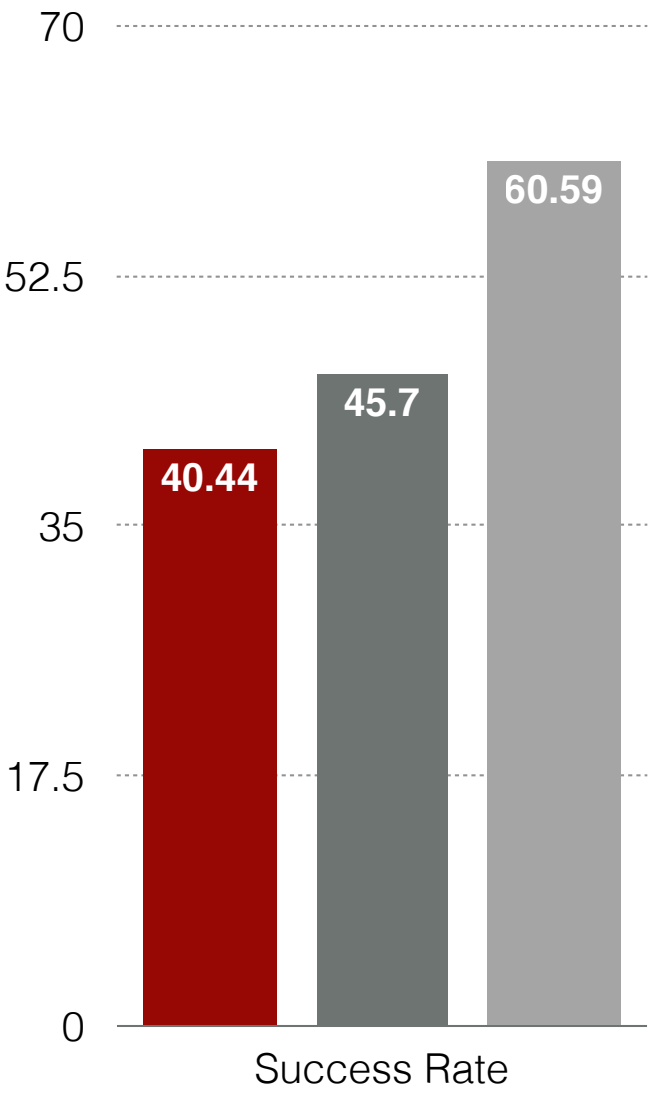


- Our Approach
- w/o imitation learning
- w/o goal distribution
- w/o auxiliary objectives
- w/o language

- 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

Drone 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

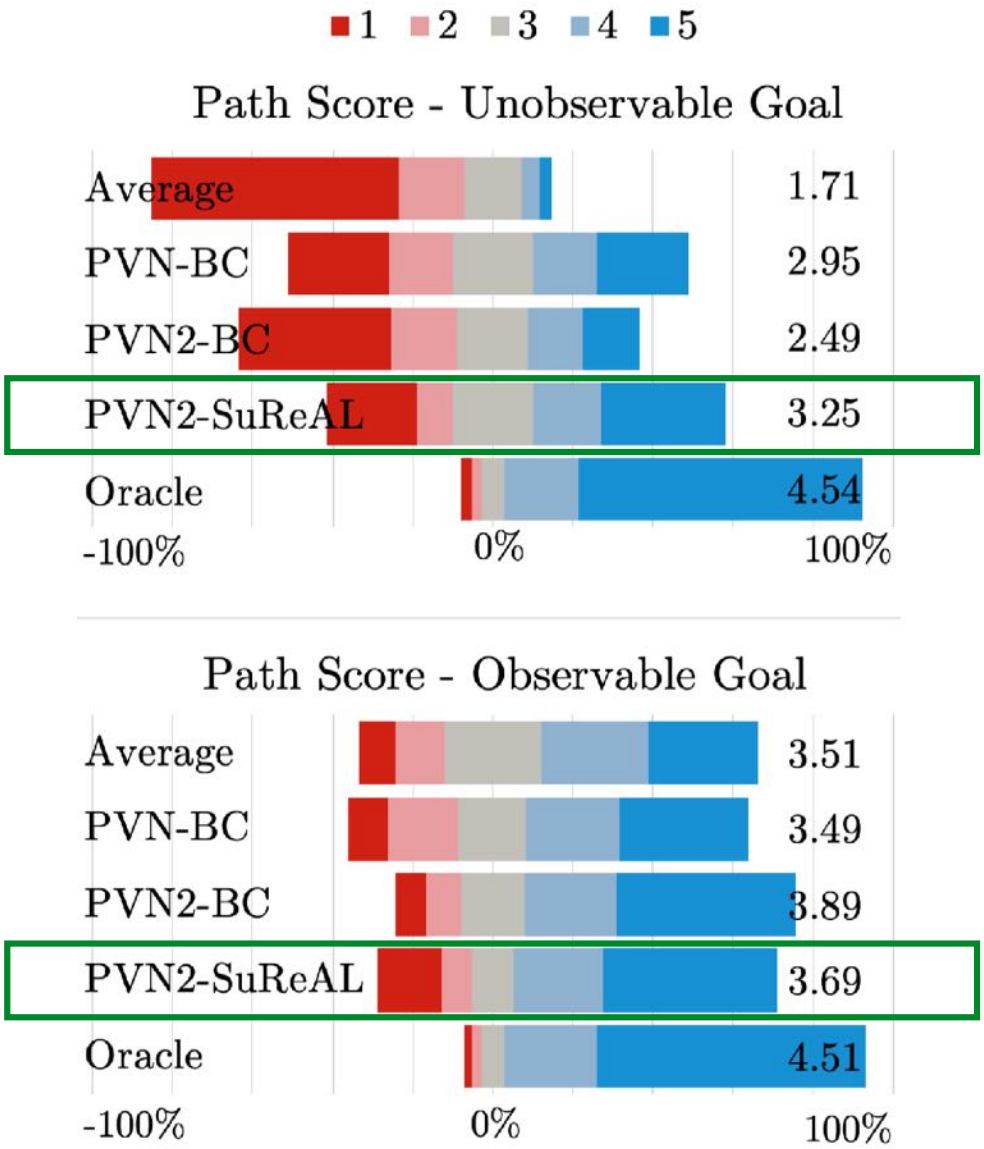
Data

- 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

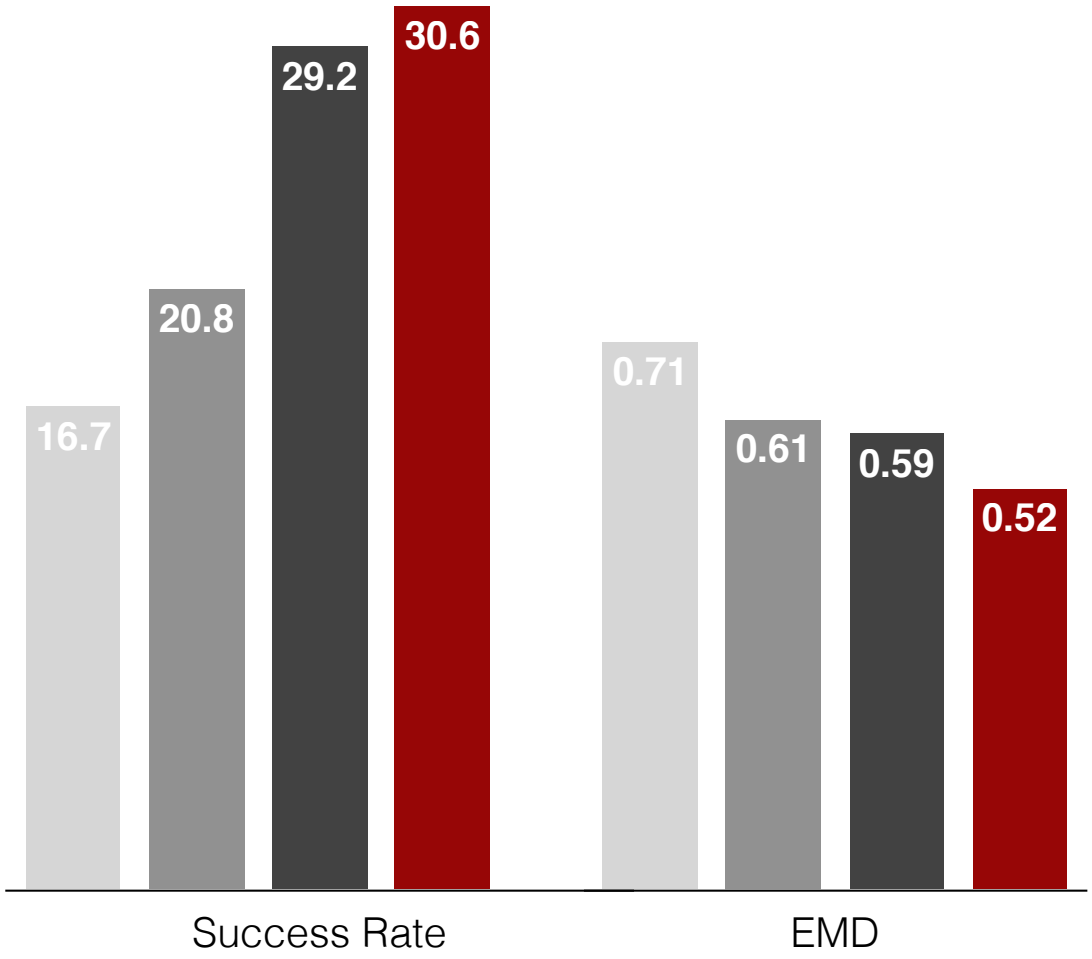
- Two automated metrics
  - SR: success rate
  - EMD: path earth's move distance
- Human evaluation: score path and goal on a 5-point Likert scale

Observability



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

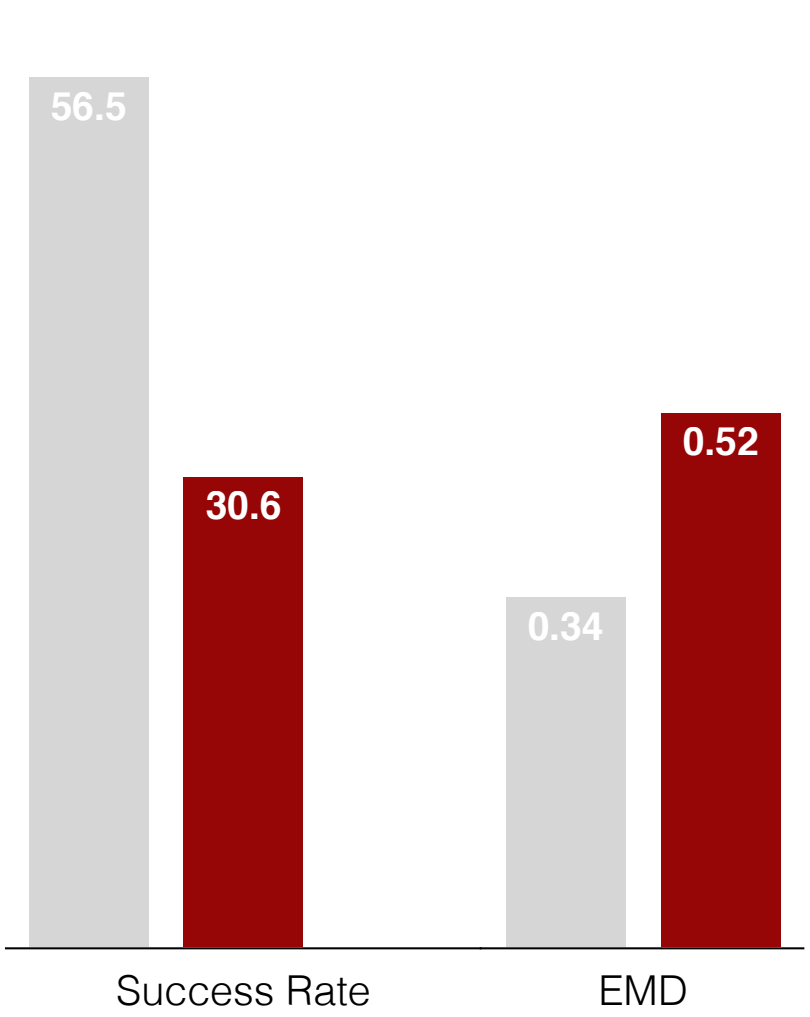
Test Results



Average
PVN-BC
PVN2-BC
Our Approach

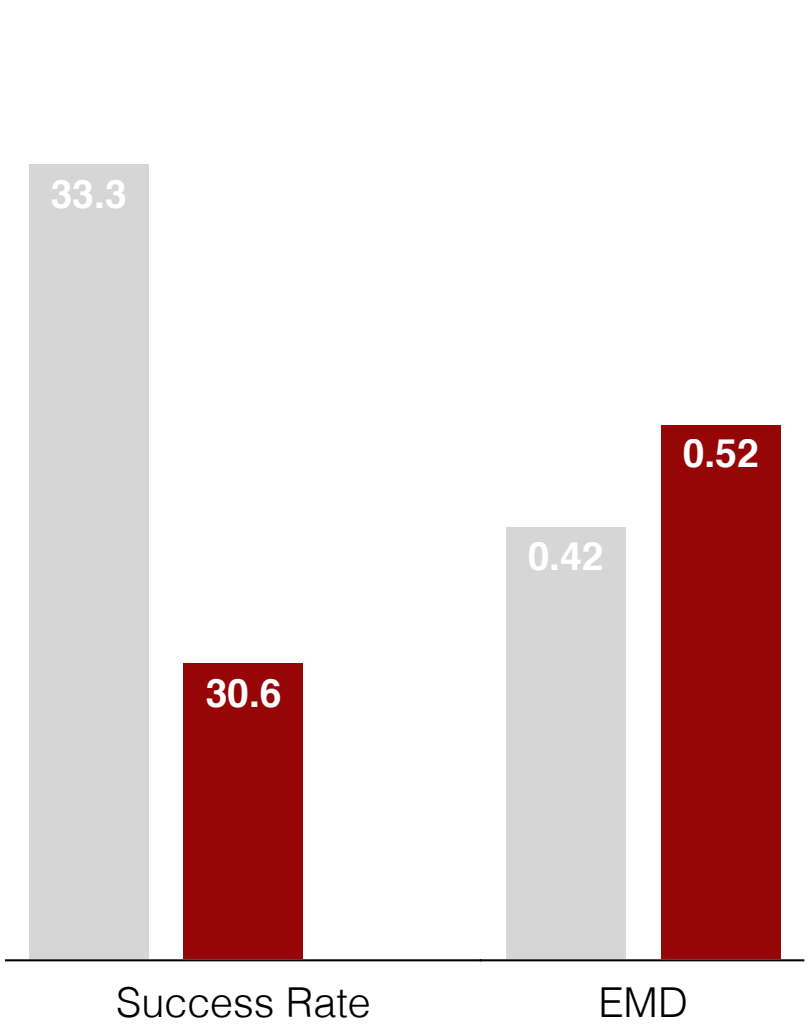
- SR often too strict: 30.6% compared to 39.7% five-points on goal
- EMD performance generally more reliable, but still fails to account for semantic correctness

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

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