Robot Control in Situated Instruction Following

Yoav Artzi

Advances of Language & Vision Research Workshop
ACL 2020
Mapping Instructions to Actions

Goal: model and learn a mapping

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

\[ f(\ldots, v_t, \omega_t, \ldots) = \text{STOP} \]

\textit{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

Instruction

Language Understanding

Planning

Control

Perception

Mapping
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 $\pi$ from start state $s_0$.
- Predicting $d(s; \pi^*, s_0)$ for an expert policy $\pi^*$ 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

**Instruction**

**Stage I**

- Planning with Position Visitation Prediction
- Trajectory distribution
- Goal distribution

**Stage II**

- Action Generation
Stage I: Planning with Position Visitation Prediction

Instruction → CNN → Mapping → Observation Mask → Trajectory Prediction

Learned Maps
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

\[ \sum \]

Semantic Map (time $t-1$)

Projected Features (time $t$)

Leaky Integrator

Semantic Map (time $t$)
Differentiable Mapping

Step 4: Grounding

after the blue bale take a right towards the small white bush before the white bush ...
Observability Mask

Observation Mask (time \( t-1 \))

Current Observation Mask

Leaky Integrator

Observation Mask (time \( t \))
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

Convolutions  Text Kernels  Deconvolutions

Text Convolutions
Stage I: Planning with Position Visitation Prediction

✓ Extract visual features and construct maps

✓ Compute visitation distributions over the maps

Differentiable Mapping

LingUNet

Trajectory distribution
Goal distribution
Stage II: Action Generation

- Relatively simple control problem without language
- Transform and crop to agent perspective and generate configuration update
## Learning vs. Engineering

<table>
<thead>
<tr>
<th>Learned 🧕</th>
<th>Engineered 🧥</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Visual features (ResNet)</td>
<td>• Feature projections (pinhole camera model)</td>
</tr>
<tr>
<td>• Text representation (RNN)</td>
<td>• Map accumulation (leaky integrator)</td>
</tr>
<tr>
<td>• Image generation (LingUNet)</td>
<td>• Egocentric transformation (matrix rotations)</td>
</tr>
<tr>
<td>• Control (control network)</td>
<td></td>
</tr>
</tbody>
</table>

**Complete network remains fully differentiable**
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 Learning

Instruction

CNN

Map+Plan

Control

Reinforcement Learning

CNN-Sim

Features

Mask

Visitations
Supervised Plan Learning

Two objectives:
(a) Generate visitation distributions
(b) Invariance to input environment
Supervised Plan Learning

Instruction

Data: real and simulated states paired with visitation predictions

CNN

Map+Plan

Visitations

Cross-entropy loss

Demonstration Visitations

CNN-Sim

Simulator Features

Real Features

Data:

- real and simulated states paired with visitation predictions

- Simulator Features

- Real Features
Supervised Plan Learning

- Instruction
- CNN
- CNN-Sim
- Map+Plan
- Demonstration Visitations
- Cross-entropy loss
- Empirical Wasserstein distance

Adversarial discriminator to force feature invariance following CNN, no data alignment needed.
RL for Control

Instruction

CNN

Map+Plan

Features

Mask

Visitations

Control

Intrinsic reward

Reinforcement Learning
SuReAL
Supervised and Reinforcement Asynchronous Learning

Instruction

Supervised Learning

CNN

Map+Plan

CNN-Sim

Reinforcement Learning

Control

Periodic parameter updates

Sampled action sequences
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

Periodic parameter updates
Replace gold action sequences with sampled
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)
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 illustrate 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
Thank you! Questions?

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

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

Valts Blukis
[fin]
Visitation Distributions
Visitation Distribution

• Given a Markov Decisions Process:

  \[
  \pi^d(s; \pi, s_0) \]

  is the probability of visiting state \( s \) following policy \( \pi \) from start state \( s_0 \)

  \[
  d(s; \pi^*, s_0) \]

  for an expert policy \( \pi^* \) 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

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 $\phi$, a policy $\pi$ with a state-visitation distribution close to $d(\tilde{s}; \pi^*, \tilde{s}_0)$ has bounded sub-optimality
Visitation Distribution for Navigation

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

**MDP**

- $S$ States
- $A$ Actions
- $R$ Reward
- $H$ Horizon

---

**Instruction**

**Stage I**

Planning with Position Visitation Prediction

**Stage II**

Action Generation

- Trajectory Probability
- Goal Probability
Drone Model
Drone Learning
Learning Architecture

Instruction

CNN → Mapping → LingUNet → Control

Features → Semantic → Grounding → Mask → Visitations
Learning Architecture
Learning Architecture

Supervised Learning

Instruction

CNN

Map+Plan

Control

Reinforcement Learning

CNN-Sim

Features

Semantic

Grounding

Mask

Visitations
RL for Control

Instruction

CNN

Map+Plan

Control

Reinforcement Learning

CNN-Sim

Features

Mask

Visitations
RL for Control

Instruction

Simulation data only

Stage 1

Mask

Visitations

Value Function

Proximal Policy Optimization
Reward Goals

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

- 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
  
## 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**

- **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

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

- Our Approach
- Ideal Actions
- Fully Observable
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

- 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 from 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
make a right at the rock and head towards the banana