Vision Language Navigation

- **Introduction**
- Dataset & Platform
- Models
- Future Works

Language Empowering Intelligent Agents

Recommender

Dialog system

Translator

Adapting Agents to Physical Environments

Self-driving car

Intelligent nursing robot

Industrial robot

Embodied AI

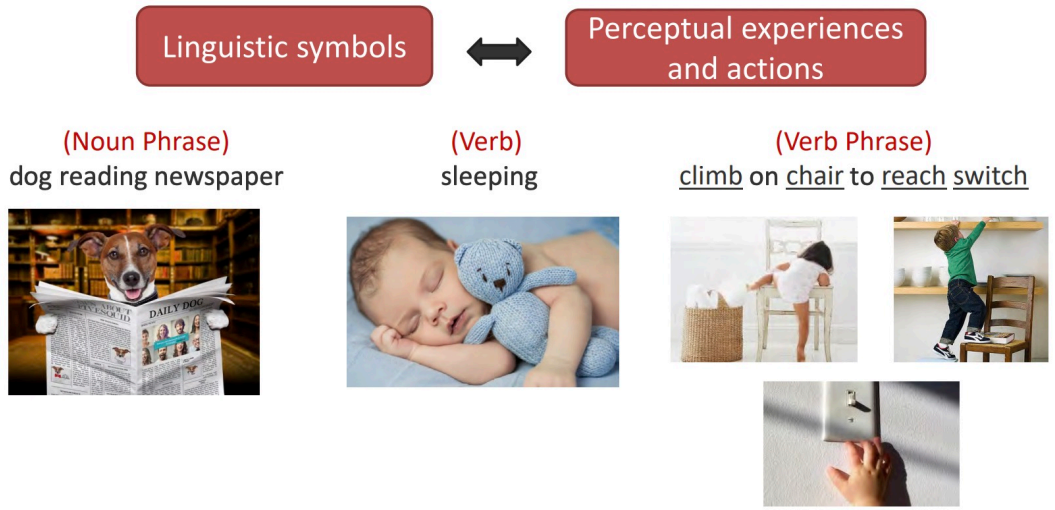
Goal : Build intelligent agents

– Communicate with people

- Follow natural language instructions

– Understand the dynamics of the perceptual environment

– Alignment between the two



Task: Vision & Language Navigation

Navigating an agent inside real 3D environments by following natural language instructions

The evolution of the task

1. go vertically down until you're underneath eh diamond mine
2. then eh go right until you're
3. you're between springbok and highest viewpoint

Figure 2: The instruction giver and instruction follower face each other, and cannot see each others maps.

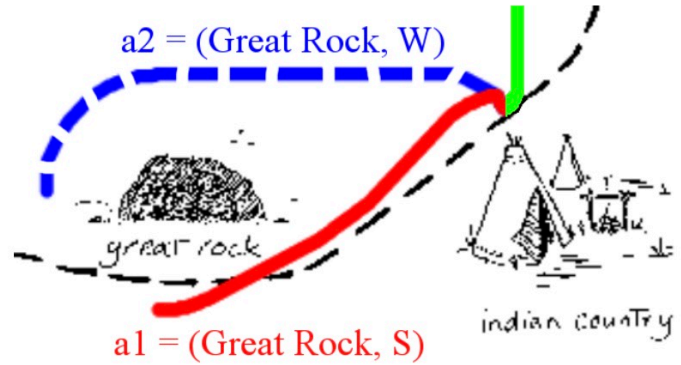


Figure 3: Sample state transition. Both actions get credit for visiting the great rock after the indian country. Action a1 also gets credit for passing the great rock on the correct side.

Vogel, Adam, and Dan Jurafsky. "Learning to follow navigational directions." *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010.

Task: Vision & Language Navigation

(a) Robotic forklift

Commands from the corpus
- Go to the first crate on the left and pick it up.
- Pick up the pallet of boxes in the middle and place them on the trailer to the left.
- Go forward and drop the pallets to the right of the first set of tires.
- Pick up the tire pallet off the truck and set it down

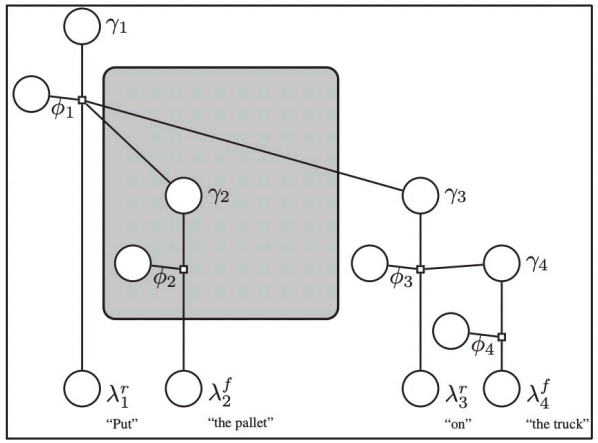
(b) Sample commands

Figure 1: A target robotic platform for mobile manipulation and navigation (Teller et al. 2010), and sample commands from the domain, created by untrained human annotators. Our system can successfully follow these commands.

EVENT An action sequence that takes place (or should take place) in the world (e.g. "Move the tire pallet").
OBJECT A thing in the world. This category includes people and the robot as well as physical objects (e.g. Forklift, the tire pallet, the truck, the person).
PLACE A place in the world (e.g. "on the truck," or "next to the tire pallet").
PATH A path or path fragment through the world (e.g. "past the truck" or "toward receiving").

$$EVENT_1(r = \texttt{Put}, \; l = OBJ_2(f = \texttt{the pallet}), \; l2 = PLACE_3(r = \texttt{on}, \; l = OBJ_4(f = \texttt{the truck})))$$

(a) SDC tree

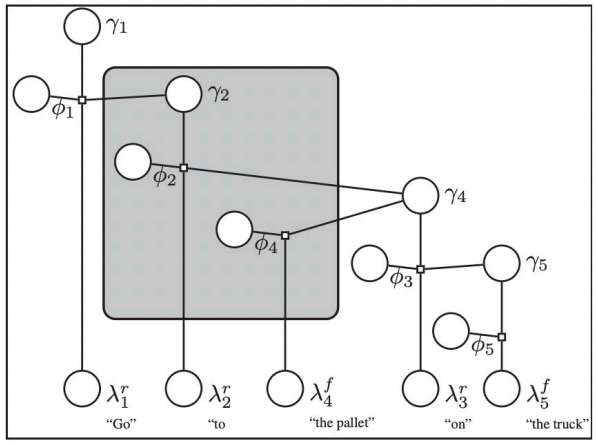


(b) Induced model

Figure 2: (a) SDC tree for "Put the pallet on the truck." (b) Induced graphical model and factorization.

$$EVENT_1(r = \texttt{Go} \; l = PATH_2(r = \texttt{to}, \; l = OBJ_3(f = OBJ_4(f = \texttt{the pallet}), \; r = \texttt{on}, \; l = OBJ_5(f = \texttt{the truck}))))$$

(a) SDC tree

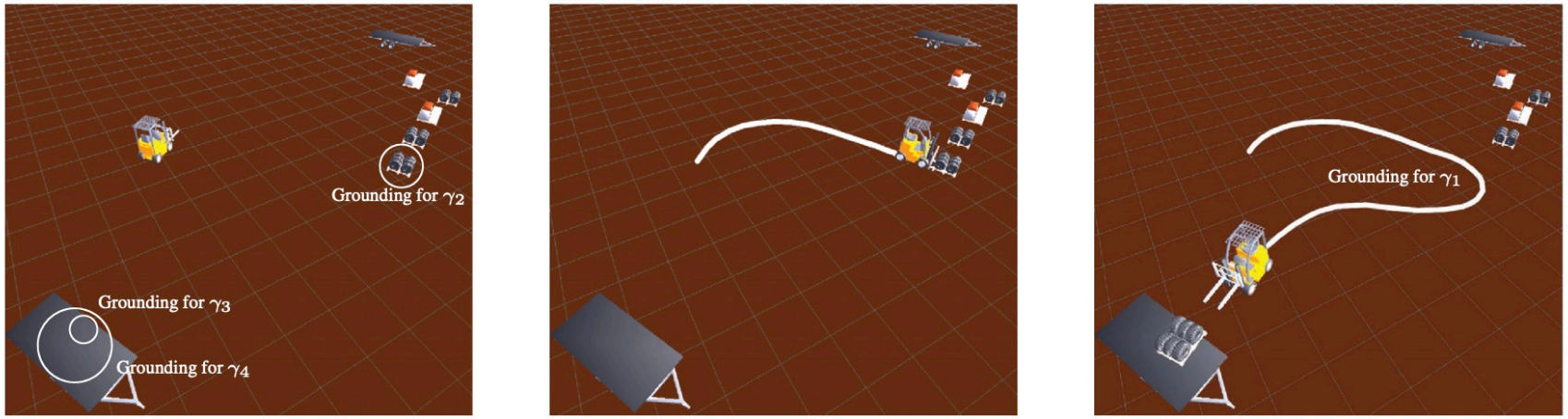


(b) Induced model

Figure 3: (a) SDC tree for "Go to the pallet on the truck." (b) A different induced factor graph from Figure 2. Structural differences between the two models are highlighted in gray.

Tellex, Stefanie, et al. "Understanding natural language commands for robotic navigation and mobile manipulation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 25. No. 1. 2011.

Task: Vision & Language Navigation



(a) Object groundings (b) Pick up the pallet (c) Put it on the truck

Figure 4: A sequence of the actions that the forklift takes in response to the command, "Put the tire pallet on the truck." (a) The search grounds objects and places in the world based on their initial positions. (b) The forklift executes the first action, picking up the pallet. (c) The forklift puts the pallet on the trailer.

Tellex, Stefanie, et al. "Understanding natural language commands for robotic navigation and mobile manipulation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 25. No. 1. 2011.

Task: Vision & Language Navigation

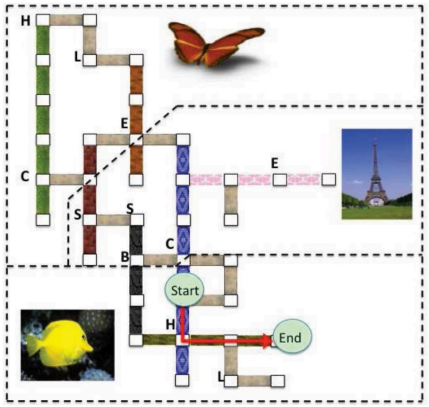


Figure 1: This is an example of a route in our virtual world. The world consists of interconnecting hallways with varying floor tiles and paintings on the wall (butterfly, fish, or Eiffel Tower.) Letters indicate objects (e.g. 'C' is a chair) at a location.

*"Go towards the **coat rack** and take a left at the coat rack. go all the way to the end of the hall and this is 4."*

*"Position 4 is a dead end of the **yellow floored hall** with fish on the walls."*

*"Turn so that the wall is on your right side. **walk forward once**. turn left. walk forward twice."*

"Foward to the fish. first left. go tot [sic] the end."

*"Place your back to the wall of the ' T' intersection. Turn right. Go forward one segment to the intersection with the yellow-tiled hall. This intersection contains a **hatrack**. Turn left. Go forward two segments to the end of the hall. This is Position 4."*

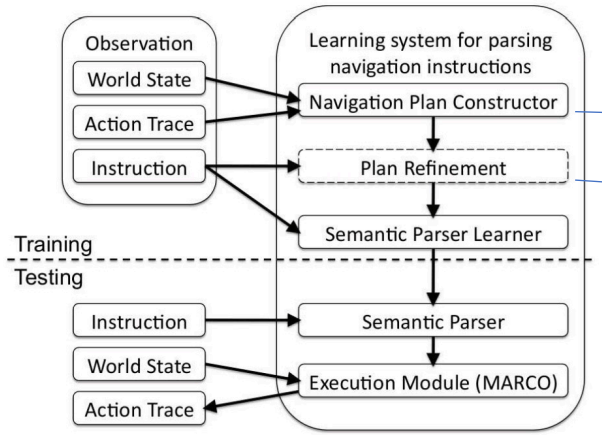


Figure 2: An overview of our system

Instruction: "Go away from the lamp to the intersection of the red brick and wood"

Basic:
Turn ( ) ,
Travel (steps: 1)

Landmarks:
Turn ( ) ,
Verify (left: WALL , back: LAMP , back: HATRACK , front: BRICK HALL) ,
Travel (steps: 1) ,
Verify (side: WOOD HALL)

Figure 3: Examples of automatically generated plans.

Chen, David, and Raymond Mooney. "Learning to interpret natural language navigation instructions from observations." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 25. No. 1. 2011.

Task: Vision & Language Navigation

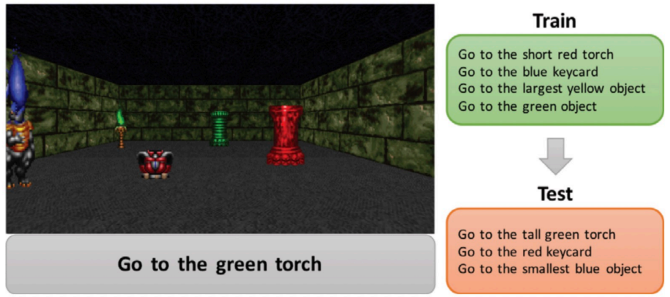


Figure 1: An example of task-oriented language grounding in the 3D Doom environment with sample instructions. The test set consists of unseen instructions.

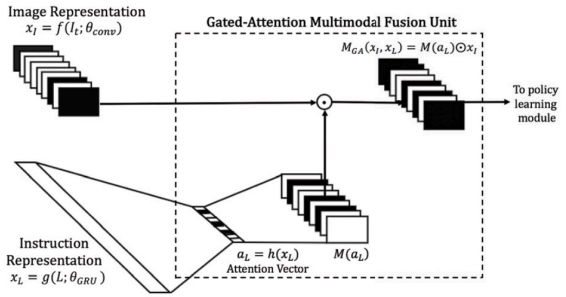


Figure 3: Gated-Attention unit architecture.

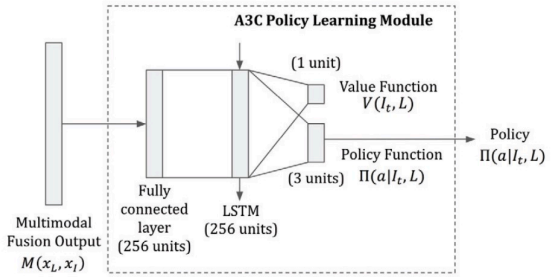
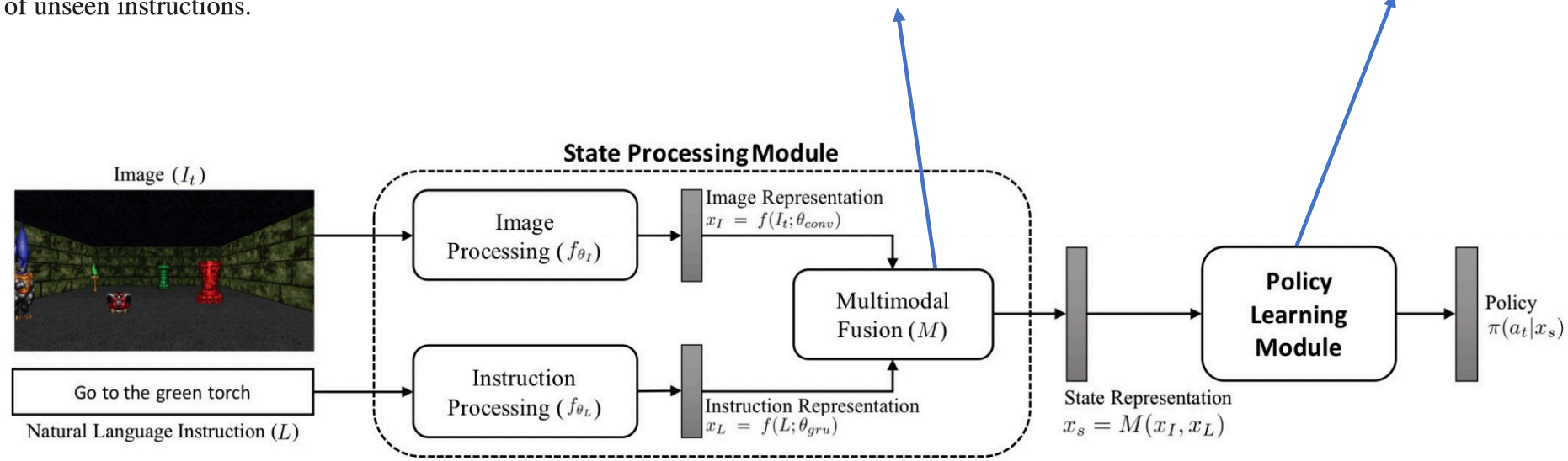
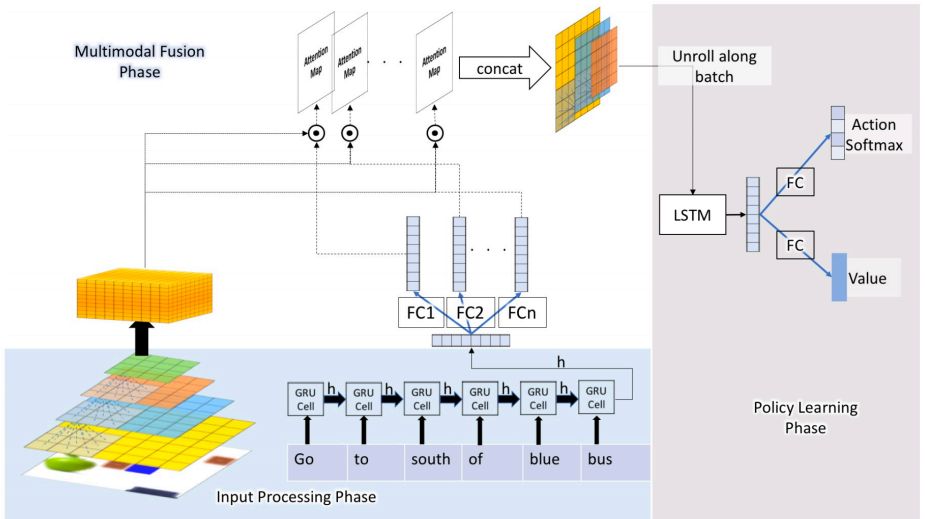


Figure 4: A3C policy model architecture.



Task: Vision & Language Navigation



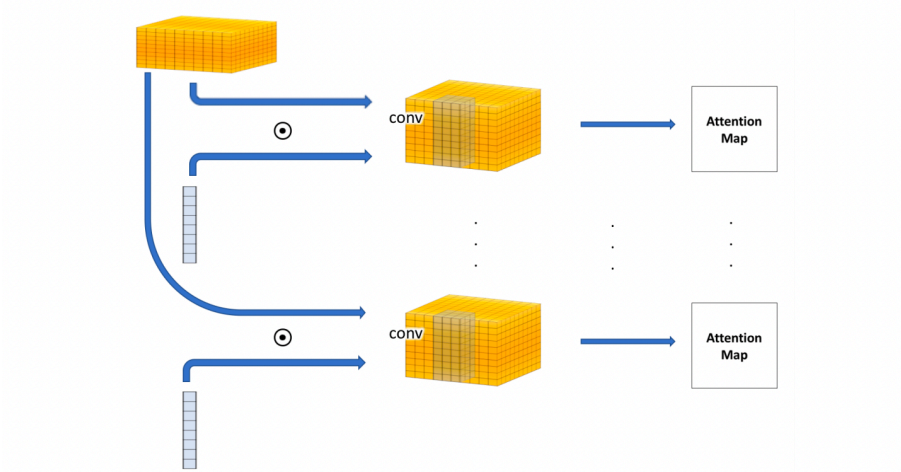
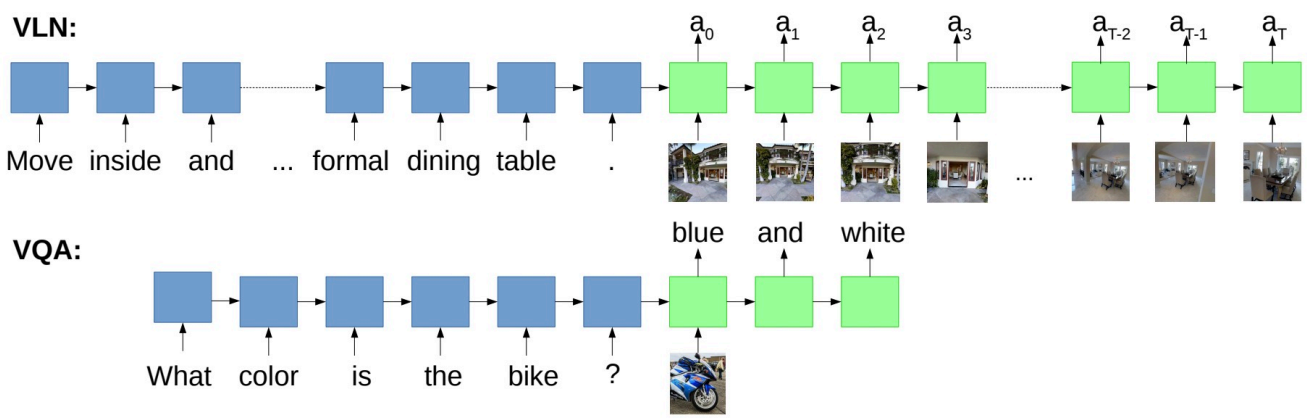


Figure 4. Multimodal fusion phase: each vector obtained from FC layer is used as 1x1 filter to perform convolution over visual representation

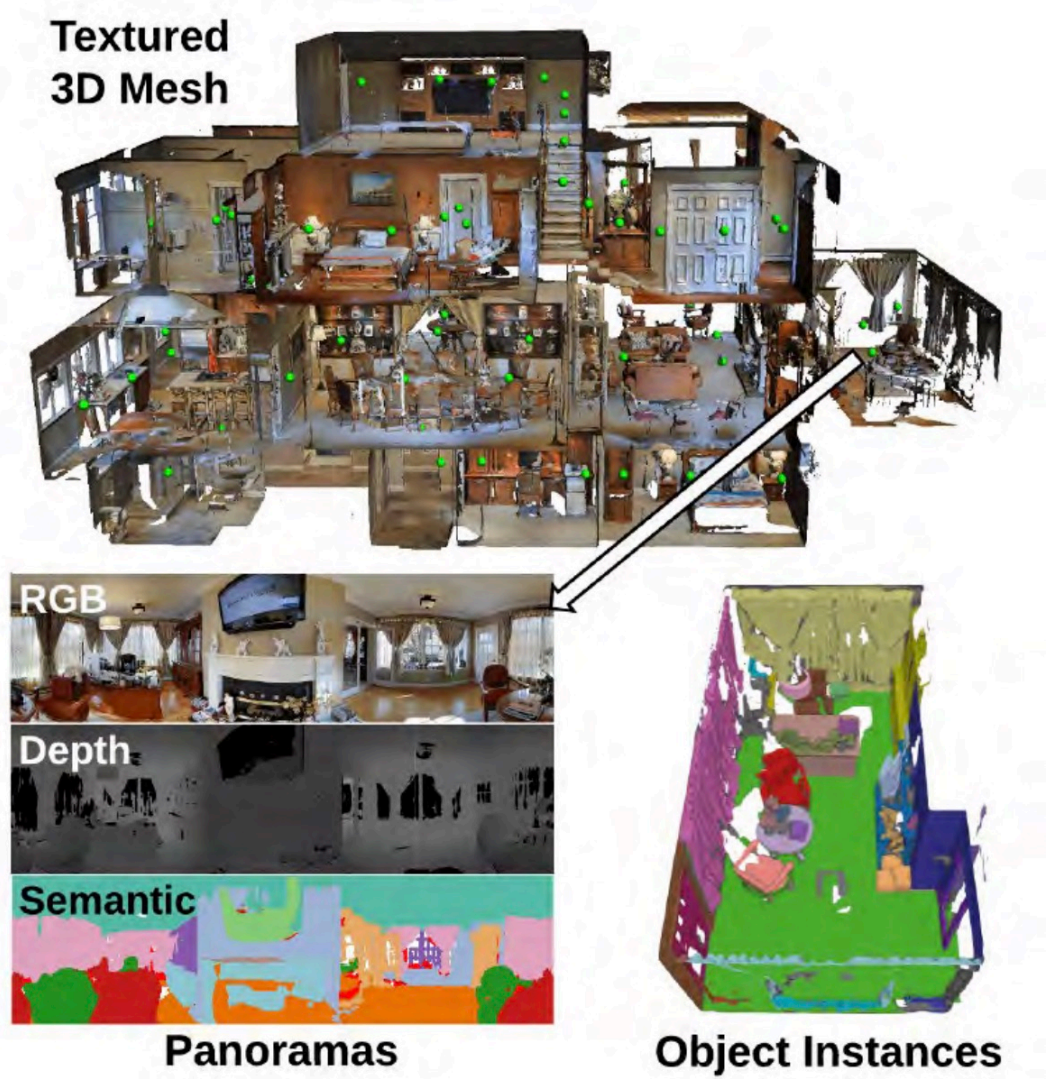
Compared with the VQA



Chaplot, Devendra Singh, et al. "Gated-attention architectures for task-oriented language grounding." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

- Introduction
- **Dataset & Platform**
- Models
- Future Works

Dataset & Platform



Matterport 3D Dataset

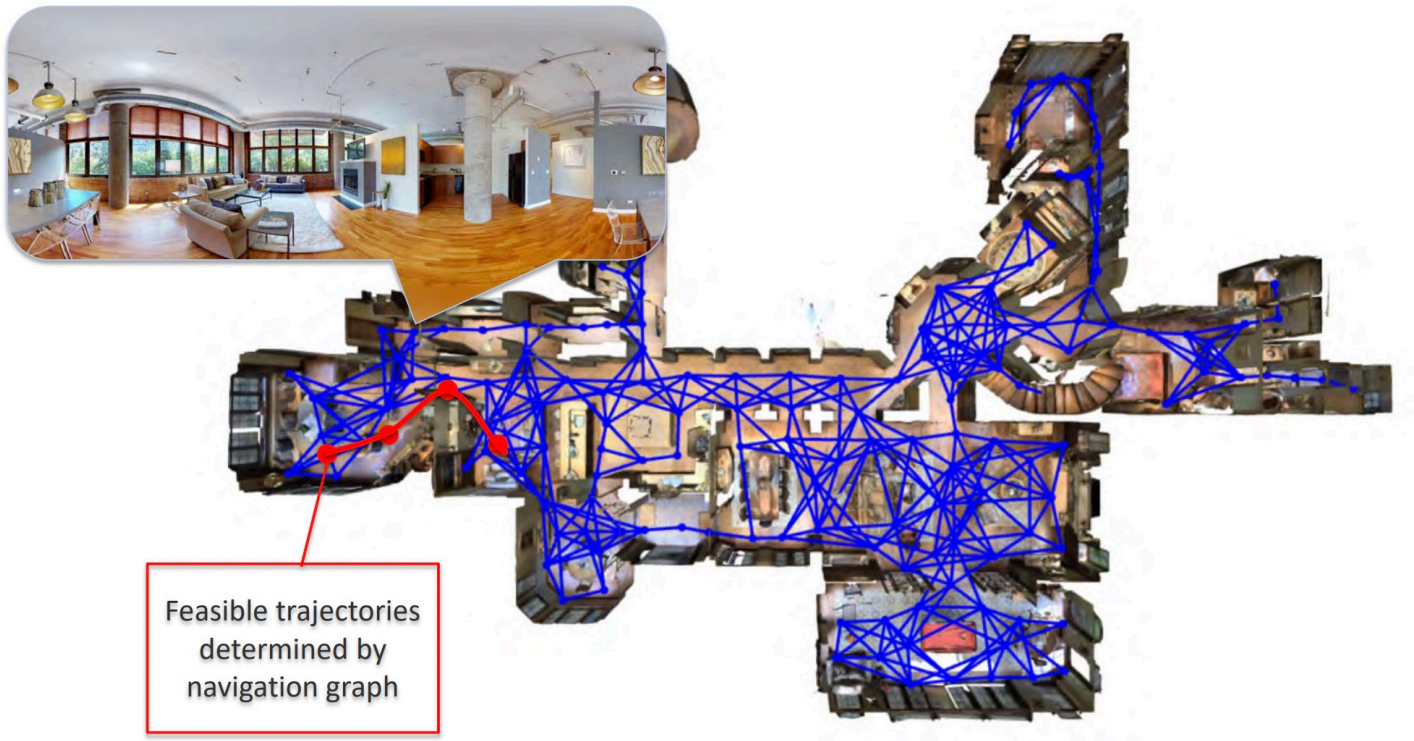
- 10,800 panoramic views based on 194K RGB-D images
- 90 building-scale scenes (avg. 23 rooms each)
- Includes textured 3D meshes with object segmentations
- Largest RGB-D dataset

Panoramas are captured from viewpoints (green spheres) on average 2.25m apart

Anderson, Peter, et al. "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Dataset & Platform

Observations



Nav-Graph

Matterport 3D Simulator for VLN Task

Room-to-Room (R2R) Dataset

- ~7K shortest paths
- 3 instructions for each path
  - Average instruction length 29 words
  - Average trajectory length is 10 meters
- **Task:**

given natural language instructions, find the goal location

Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

Anderson, Peter, et al. "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

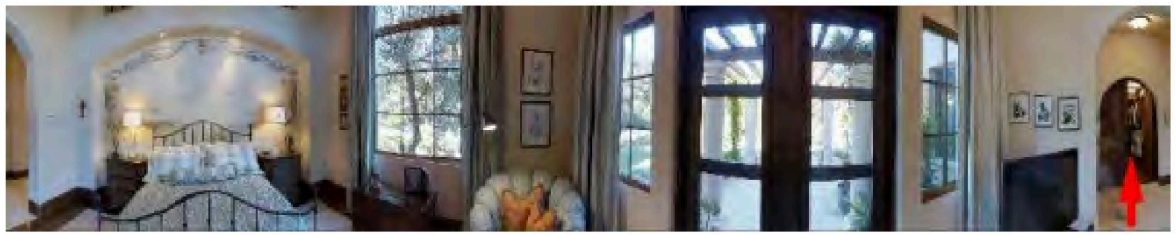
Room-to-Room Dataset Examples

Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

Input: Instruction

turn completely around until you face an open door with a window to the left and a patio to the right, walk forward, … …

Input: Panoramic View



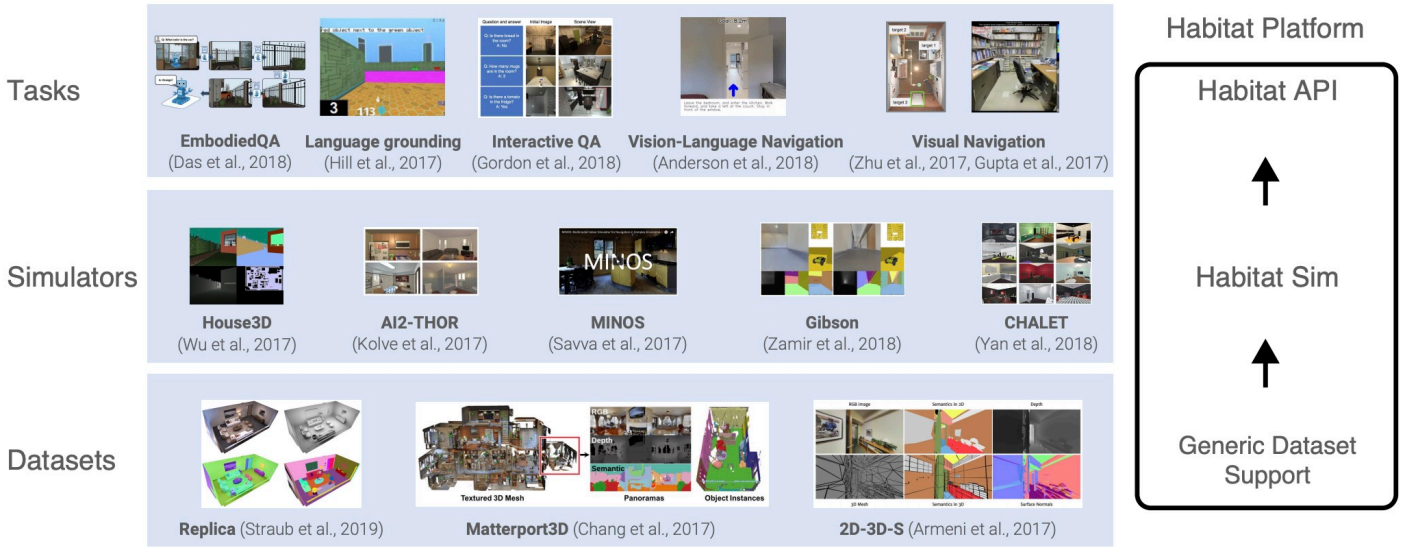
Output

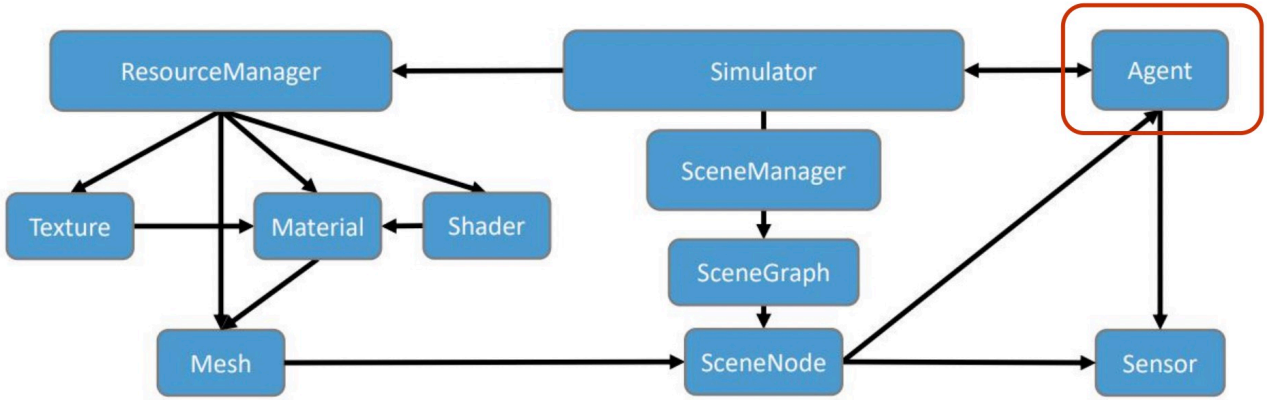
$a_t \in A$

Anderson, Peter, et al. "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Dataset & Platform

Habitat





Savva, Manolis, et al. "Habitat: A platform for embodied ai research." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

Dataset & Platform

Trajectory Length (length): The agent's **mean trajectory length** in metres. This figure provides some context regarding the amount of exploring that an agent is undertaking (relative to the mean trajectory length of the shortest path oracle, for example), and is used in the calculation of SPL.

Navigation Error (error): The agent's mean navigation error in metres. We define navigation error as the shortest path **distance** in the simulator's navigation graph **between the agent's final position** (i.e., disregarding heading and elevation) **and the goal location**. The agent is expected to identify the goal location and stop as close as possible to it. We do not evaluate the agent's entire trajectory as many instructions describe the location of the goal without specifying a particular path that must be taken to reach it.

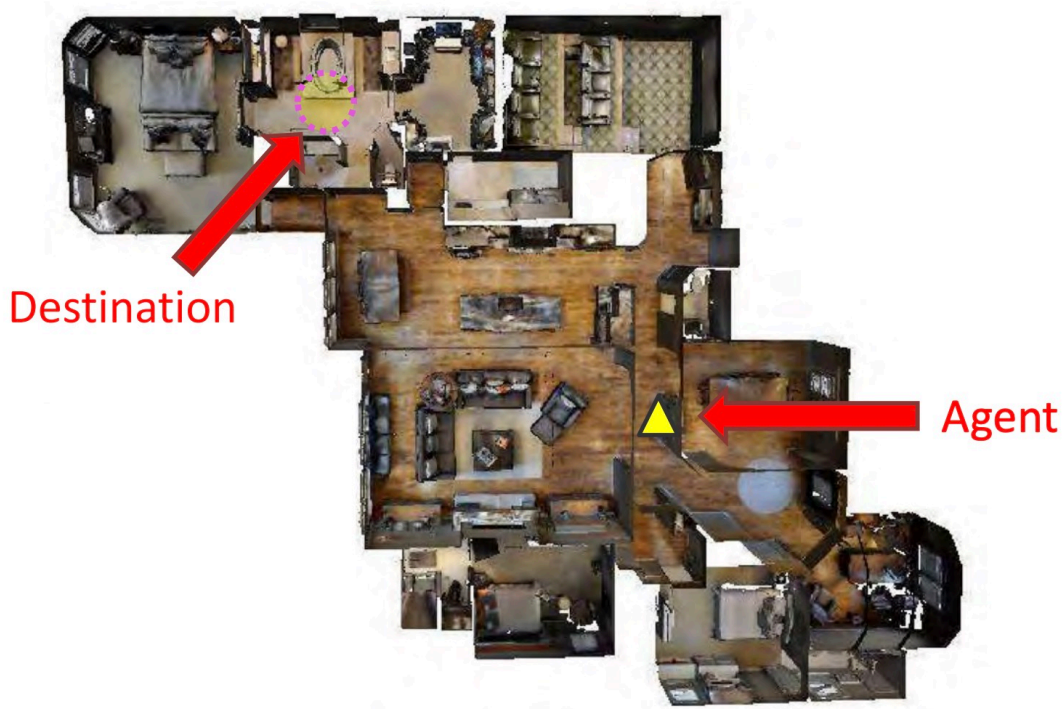
Oracle Success Rate (oracle success): The agent's mean success rate in the presence of an **oracle stopping** rule, i.e. if the agent magically **stopped at the closest point to the goal** on its trajectory. We consider stopping to be a fundamental aspect of completing the navigation task, demonstrating understanding, but also freeing the agent to potentially undertake further tasks at the goal. Nevertheless, we include this evaluation to disentangle the problem of recognizing the goal location from the other aspects of the task.

Success Rate (success): The agent's **mean success rate** in terms of reaching the goal location. We consider an episode to be a success if the agent's navigation error **is less than 3m**. This threshold allows for a reasonable margin of error in the context of an imprecise natural language navigation instruction, yet it is comfortably below the minimum starting error in the R2R dataset (which is 5m).

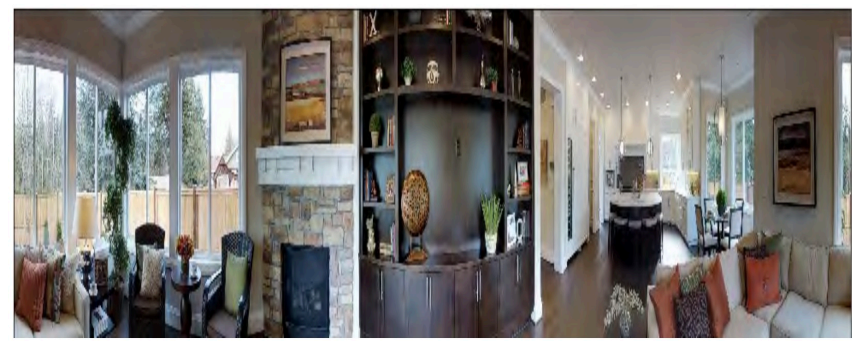
Success rate weighted by (normalized inverse) Path Length (SPL): It is generally possible to improve the agent's Success Rate at reaching the goal by exploring more of the environment before committing to a decision. However, in a robotics context, **longer trajectories have costs** (battery, wear, delays for the user, etc). Further, humans can solve the VLN task with very short trajectories. Therefore, SPL trades-off Success Rate against Trajectory Length.

(1) cross-modal grounding

Instruction: Go towards the *living room* and then turn right to the *kitchen*. Then turn left, pass a *table* and enter the *hallway*. Walk down the *hallway* and turn into the *entry way* to your right. Stop in front of the toilet.



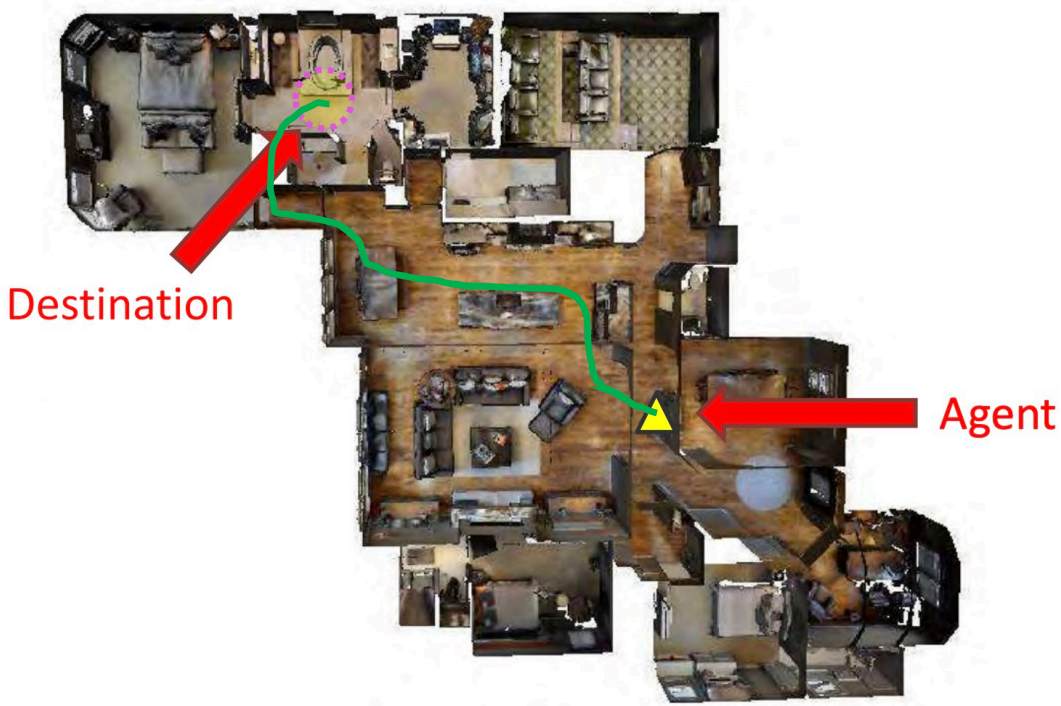
Global trajectory in top-down view (NOT visible to the agent)



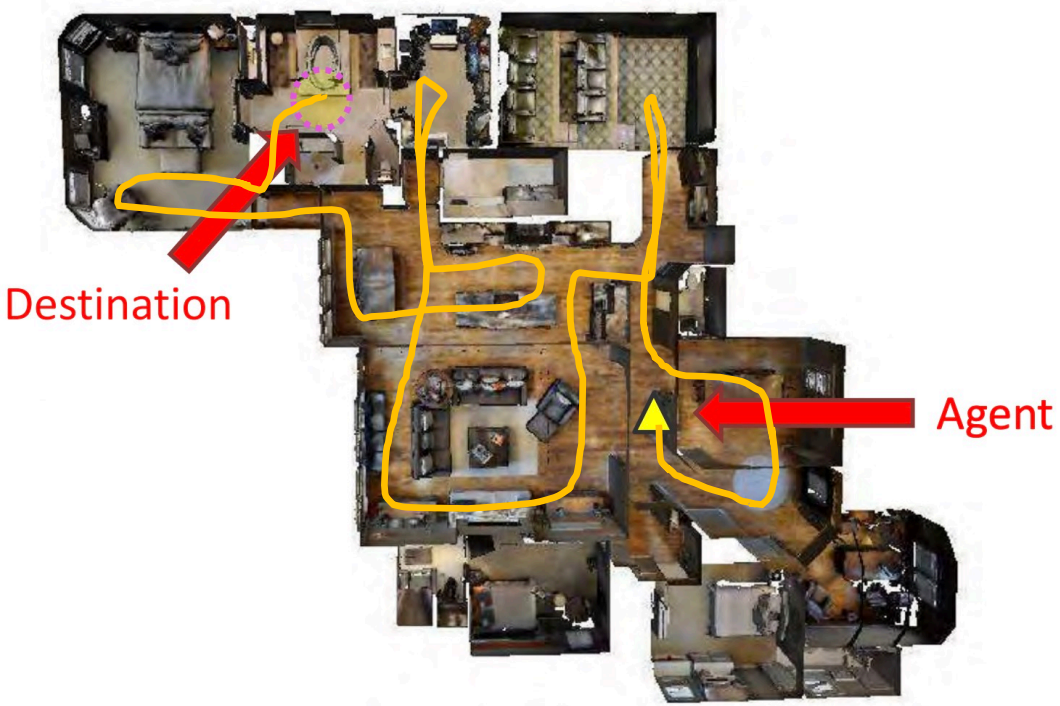
Local visual scene

(2) ill-posed feedback

Instruction: Go towards the *living room* and then turn right to the *kitchen*. Then turn left, pass a *table* and enter the *hallway*. Walk down the *hallway* and turn into the *entry way* to your right. Stop in front of the toilet.



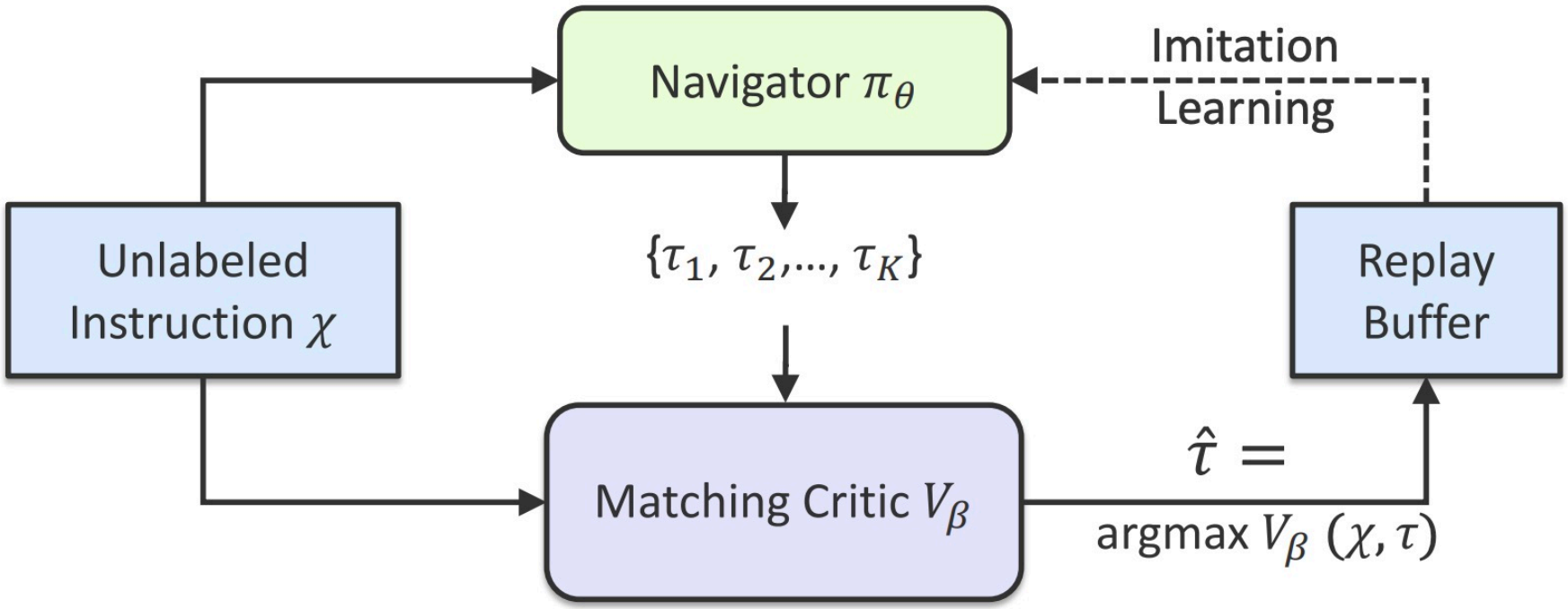
Agent #1 follows the instruction and reaches the destination.



Agent #2 randomly walks insides the house and reaches the destination.

Both trajectories are considered same in terms of the success signal.

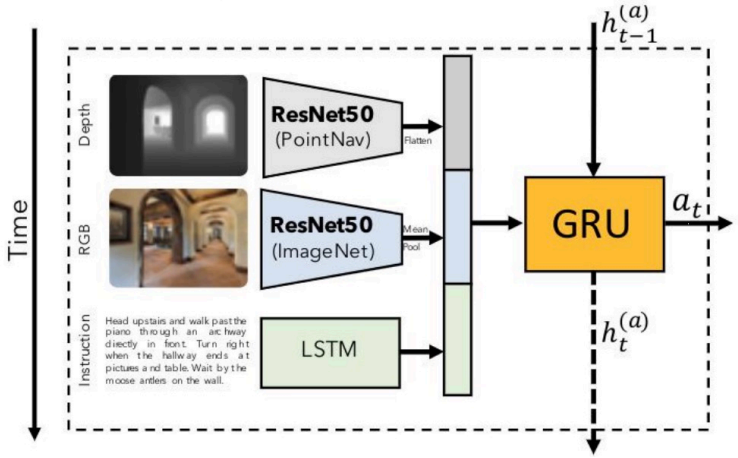
(3) generalization

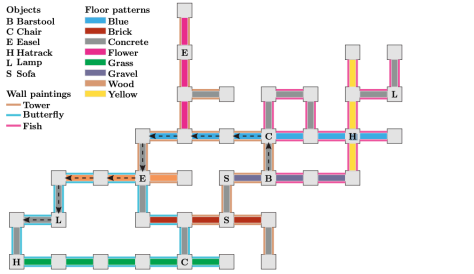


Learning from its previous good behaviors → better policy that adapts to new environments

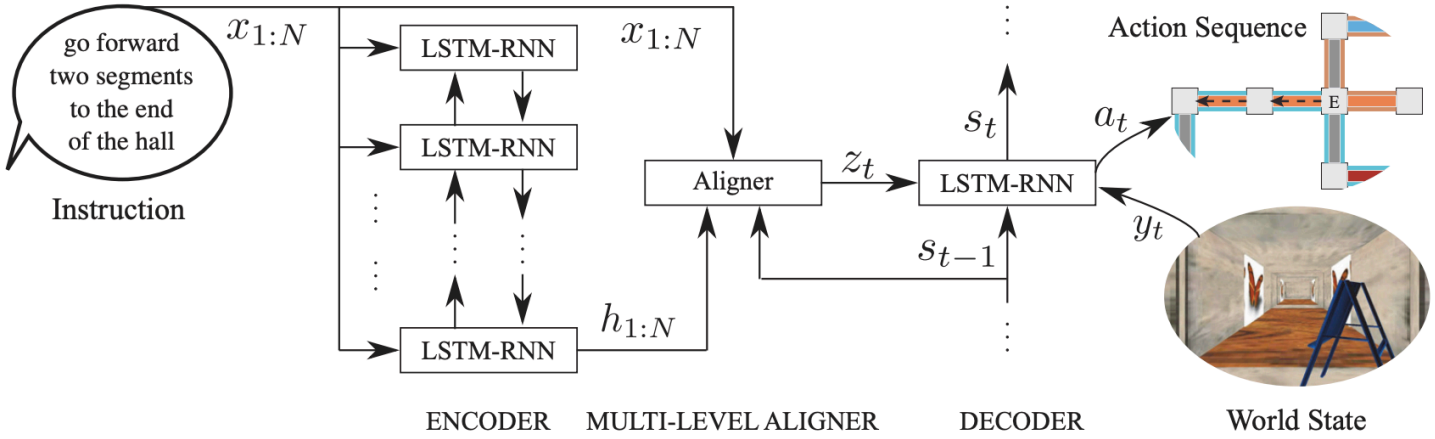
- Introduction
- Dataset & Platform
- **Models**
- Future Works

Baseline Seq2Seq





Place your back against the wall of the "T" intersection. Go forward one segment to the intersection with the blue-tiled hall. This interesction [sic] contains a chair. Turn left. Go forward to the end of the hall. Turn left. Go forward one segment to the intersection with the wooden-floored hall. This intersection conatains [sic] an easel. Turn right. Go forward two segments to the end of the hall. Turn left. Go forward one segment to the intersection containing the lamp. Turn right. Go forward one segment to the empty corner.

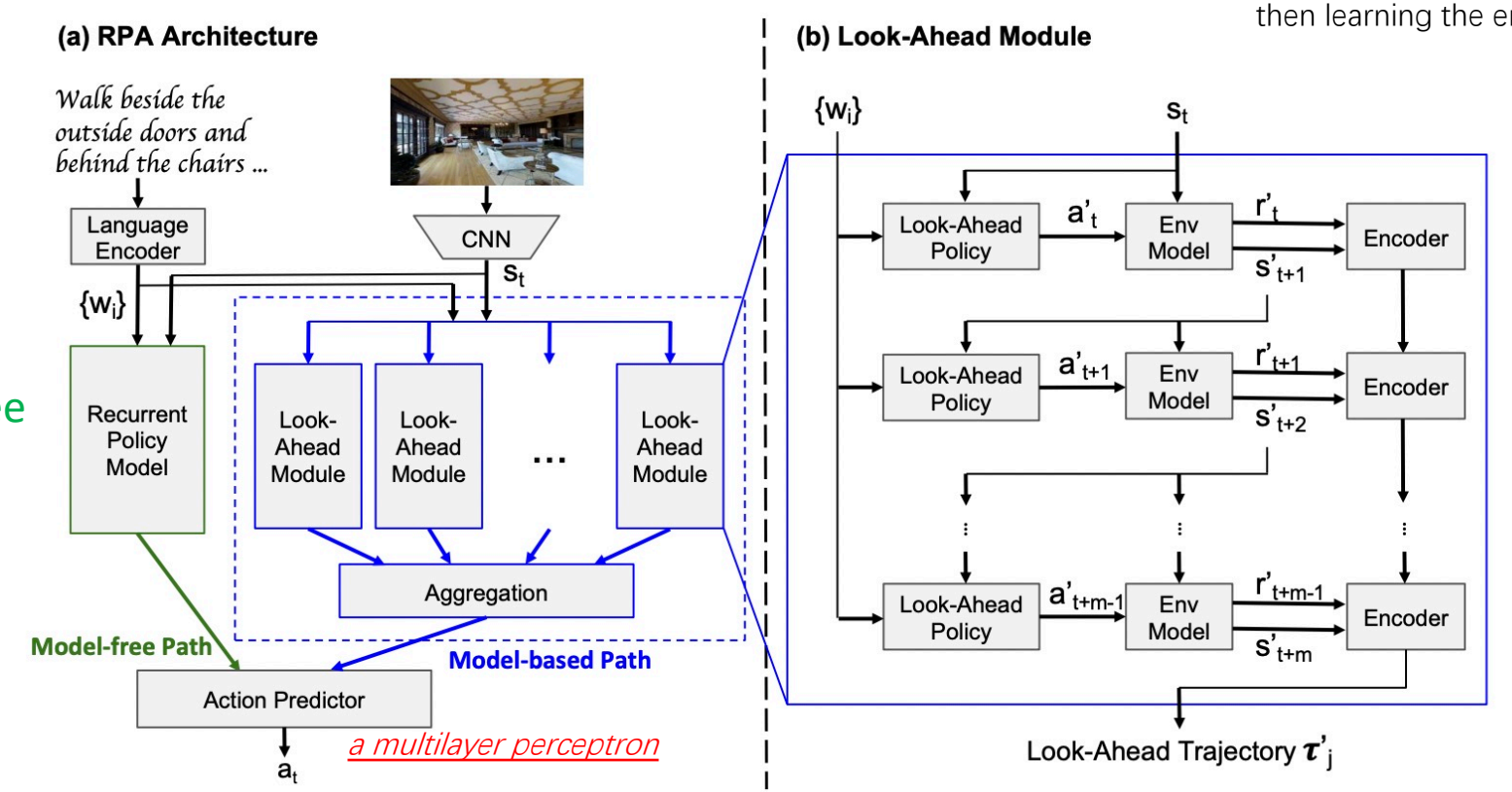


Mei, Hongyuan, Mohit Bansal, and Matthew Walter. "Listen, attend, and walk: Neural mapping of navigational instructions to action sequences." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 30. No. 1. 2016.

Look before you leap

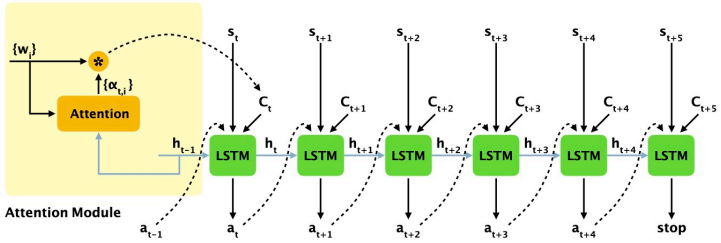
Training: learning the environment model first, then learning the enhanced policy model

Model-Free
RNN+CNN



Modeling the environment is very difficult

Model-Based

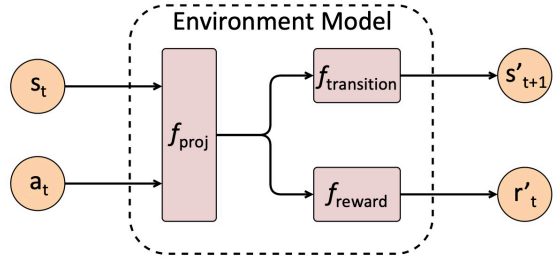


$$c_t = \sum \alpha_{t,i} w_i$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{n} \exp(e_{t,k})} \quad , \quad \text{where } e_{t,i} = h_{t-1}^{\top} w_i$$

$$h_t = LSTM(h_{t-1}, [c_t, s_t, a_{t-1}])$$

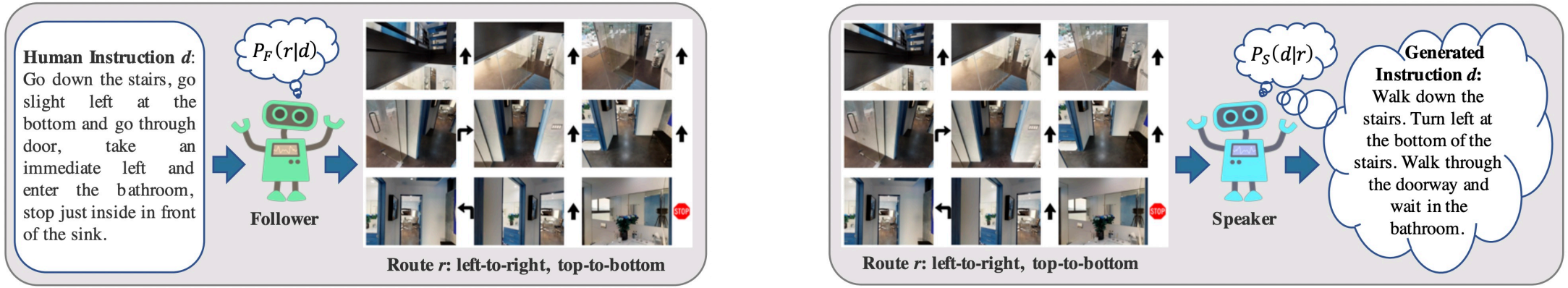
$$s'_{t+1} = f_{transition}(f_{proj}(s_t, a_t))$$
$$r'_t = f_{reward}(f_{proj}(s_t, a_t)) \quad ,$$

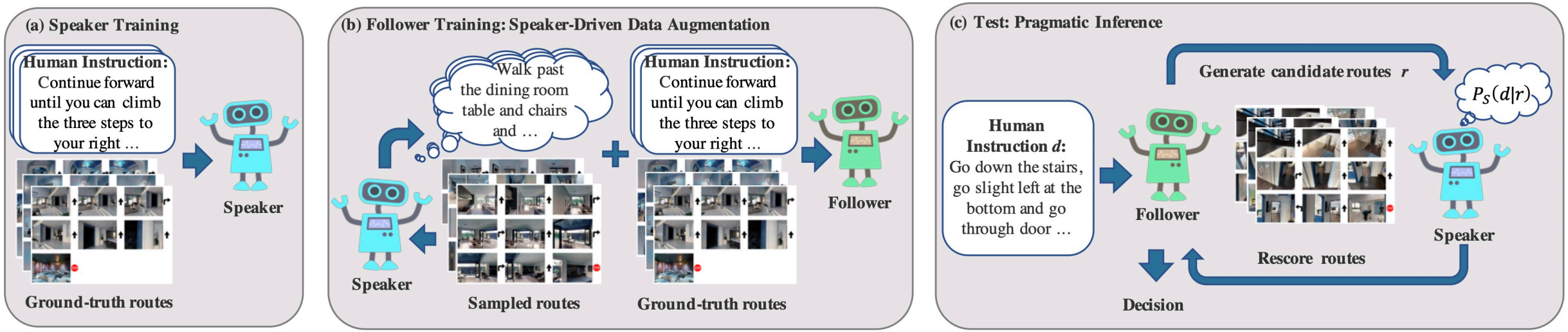


Wang, Xin, et al. "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation." Proceedings of the European Conference on Computer Vision (ECCV). 2018.

Speaker-Follower

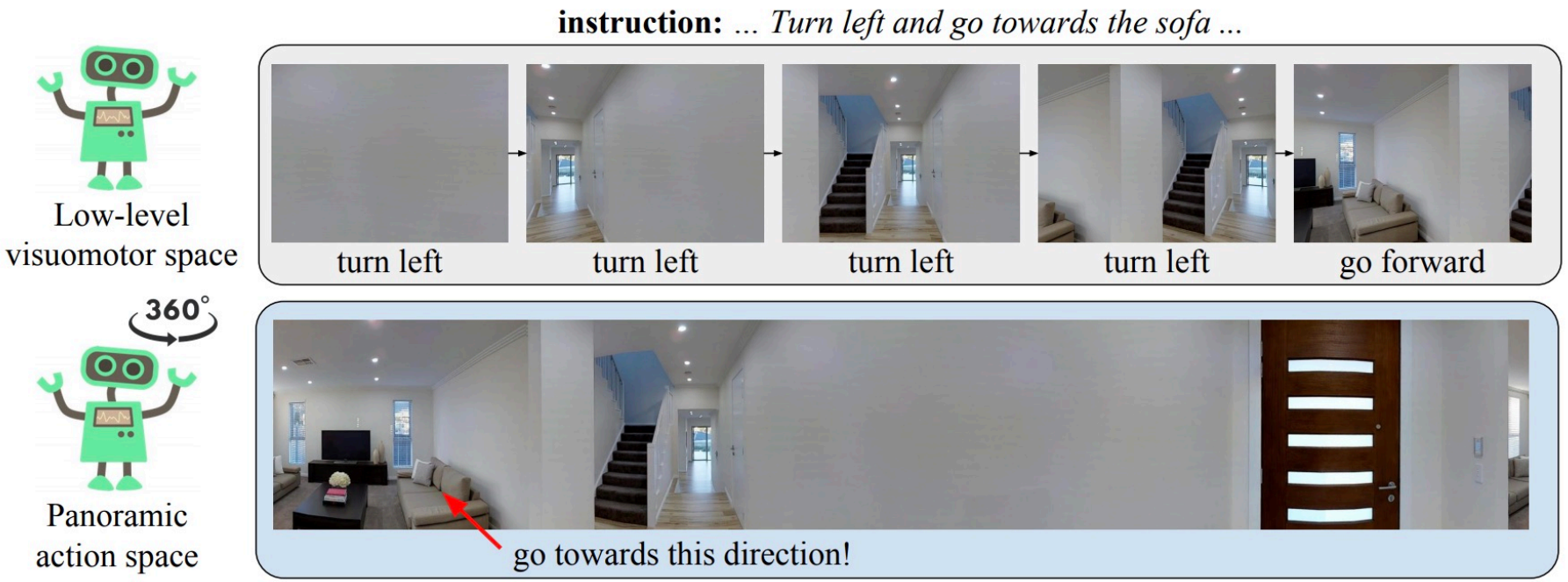
Raw Date: The trajectories are sampled the environment, and the natural language is obtained from the manual annotation.



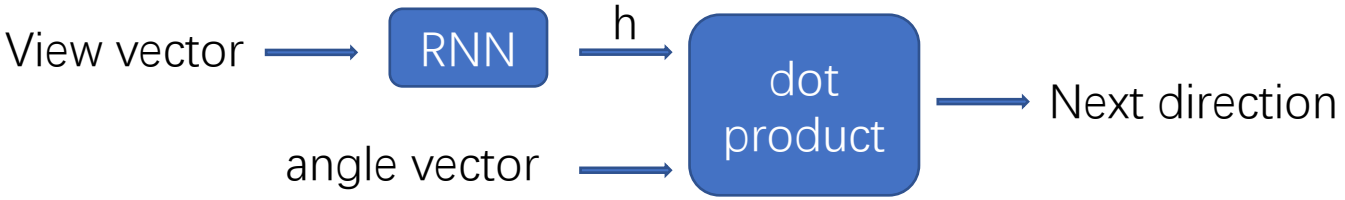


Fried, Daniel, et al. "Speaker-follower models for vision-and-language navigation." arXiv preprint arXiv:1806.02724 (2018).

Speaker-Follower



such as turning left or right by 30 degrees



Fried, Daniel, et al. "Speaker-follower models for vision-and-language navigation." arXiv preprint arXiv:1806.02724 (2018).

Speaker-Follower

Method	Validation-Seen			Validation-Unseen			Test (unseen)			
	NE ↓	SR ↑	OSR ↑	NE ↓	SR ↑	OSR ↑	NE ↓	SR ↑	OSR ↑	TL ↓
Random	9.45	15.9	21.4	9.23	16.3	22.0	9.77	13.2	18.3	9.89
Student-forcing [1]	6.01	38.6	52.9	7.81	21.8	28.4	7.85	20.4	26.6	8.13
RPA [55]	5.56	42.9	52.6	7.65	24.6	31.8	7.53	25.3	32.5	9.15
ours	**3.08**	**70.1**	**78.3**	**4.83**	**54.6**	**65.2**	**4.87**	**53.5**	63.9	11.63
ours (challenge participation)*	–	–	–	–	–	–	**4.87**	**53.5**	**96.0**	1257.38
Human	–	–	–	–	–	–	1.61	86.4	90.2	11.90

Fried, Daniel, et al. "Speaker-follower models for vision-and-language navigation." arXiv preprint arXiv:1806.02724 (2018).

Reinforced Cross-Modal Match

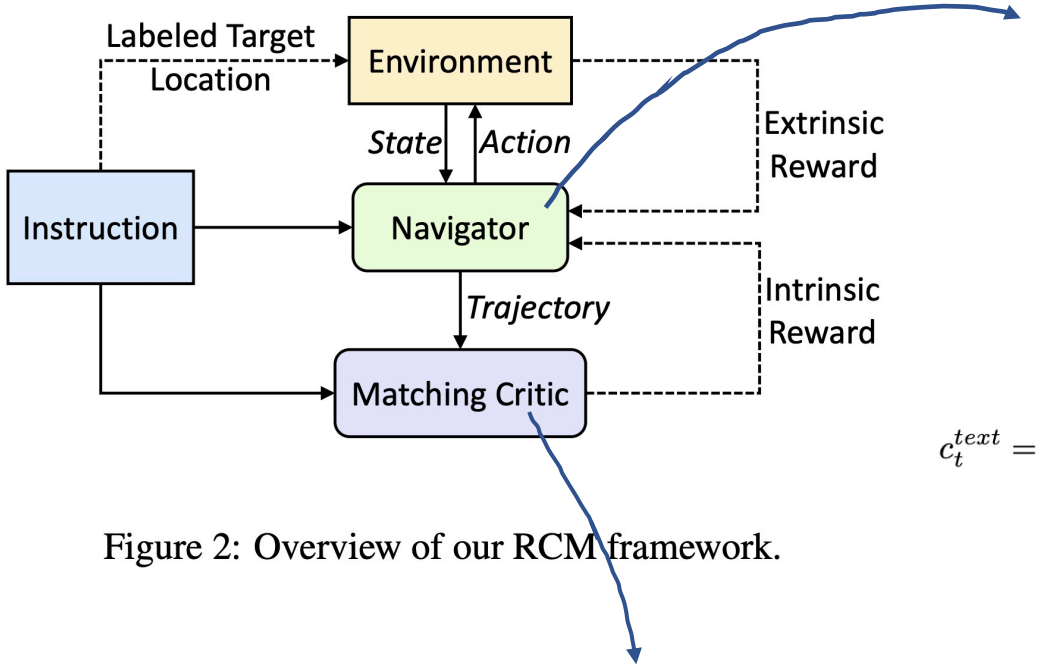


Figure 2: Overview of our RCM framework.

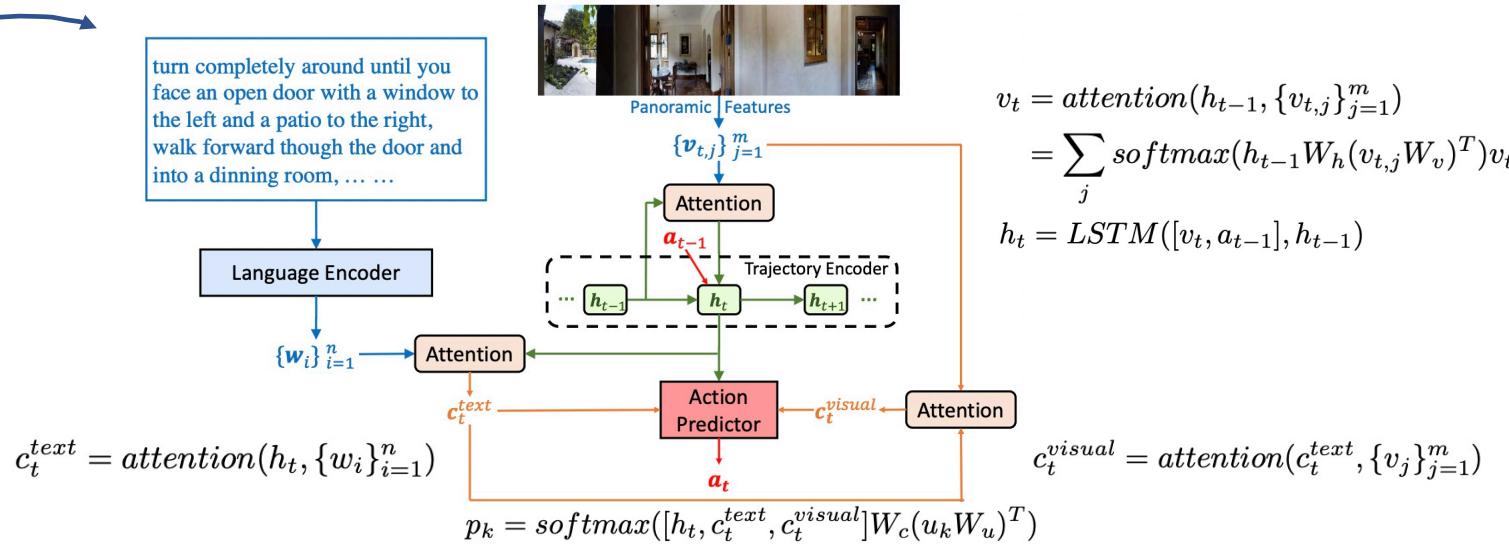
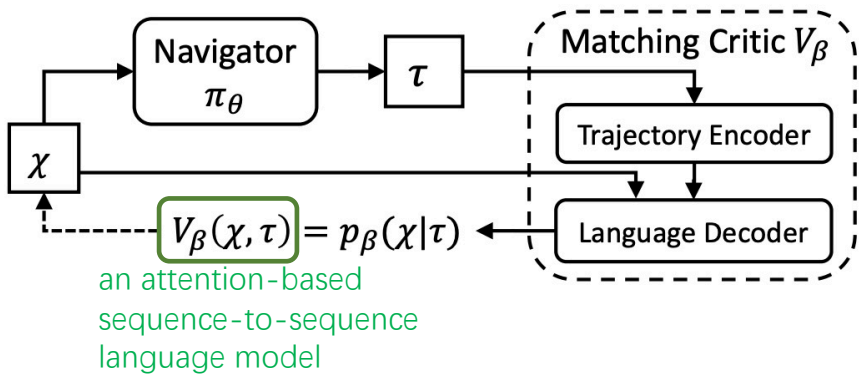


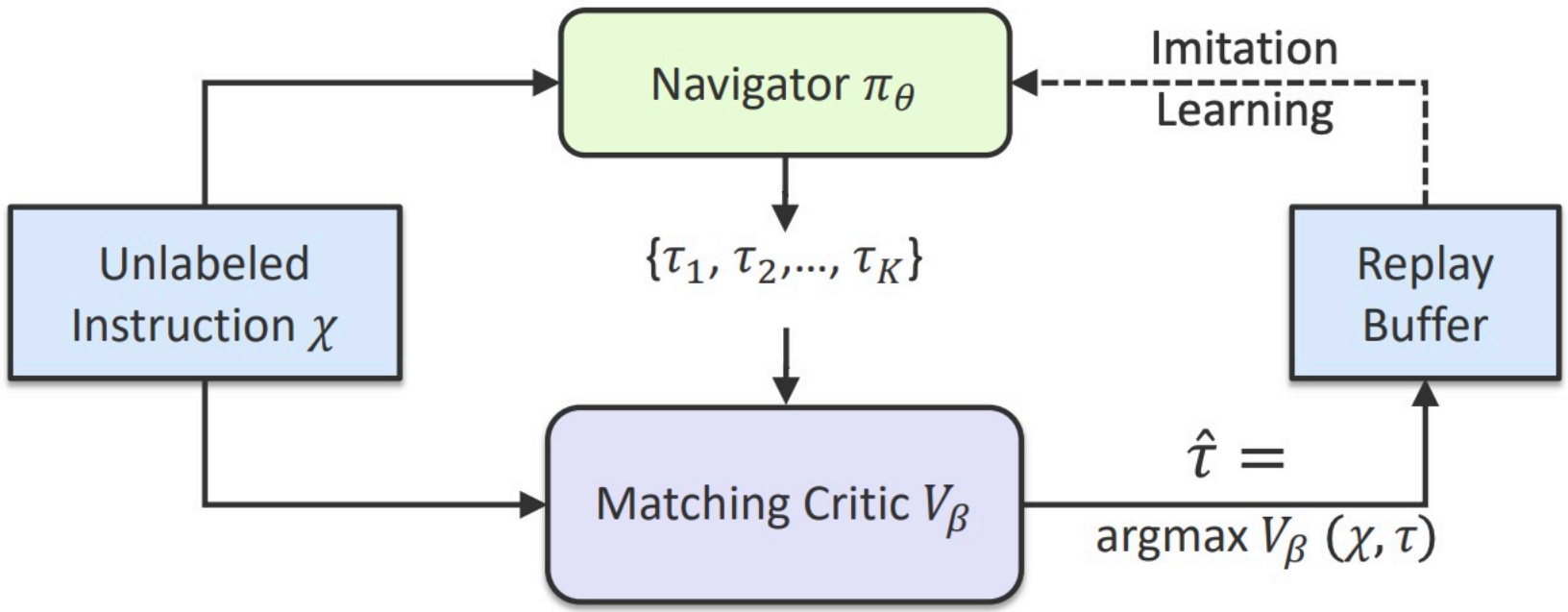
Figure 3: Cross-modal reasoning navigator at step t.



Reconstruct the instruction to encourage global matching

Wang, Xin, et al. "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Self-Supervised Imitation Learning



Learning from its previous good behaviors → better policy that adapts to new environments

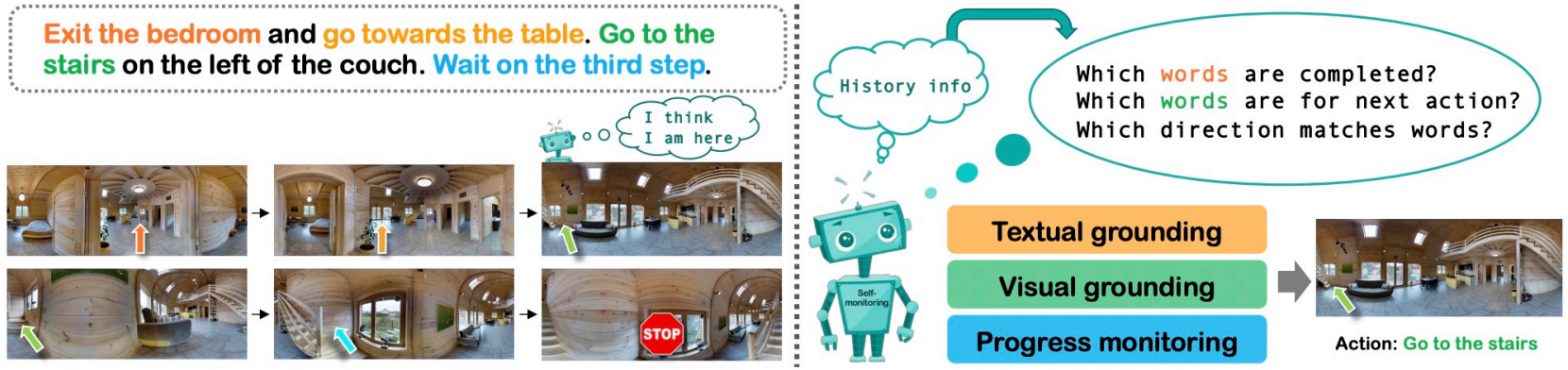
Wang, Xin, et al. "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Reinforced Cross-Modal Match

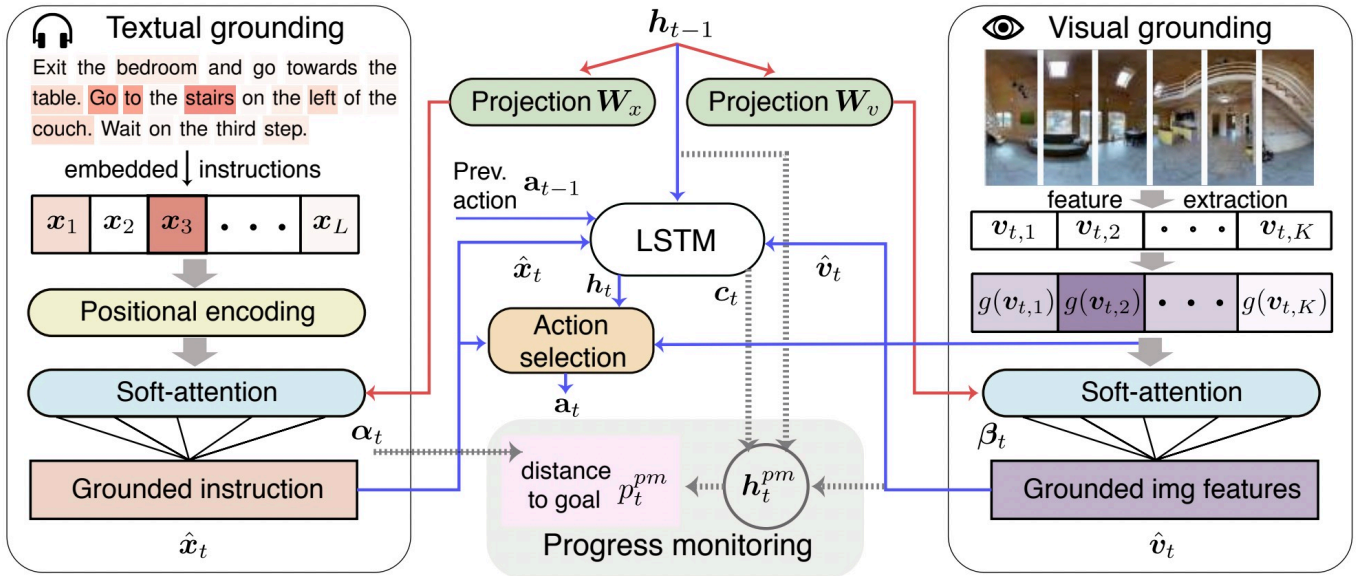
#	Model	Seen Validation				Unseen Validation			
		PL ↓	NE ↓	OSR ↑	SR ↑	PL ↓	NE ↓	OSR ↑	SR ↑
0	Speaker-Follower (no beam search) [11]	-	3.36	73.8	66.4	-	6.62	45.0	35.5
1	RCM + SIL (train)	**10.65**	3.53	75.0	66.7	**11.46**	6.09	50.1	**42.8**
2	RCM	11.92	3.37	76.6	67.4	14.84	**5.88**	**51.9**	42.5
3	— intrinsic reward	12.08	3.25	**77.2**	**67.6**	15.00	6.02	50.5	40.6
4	— extrinsic reward = pure SL	11.99	3.22	76.7	66.9	14.83	6.29	46.5	37.7
5	— cross-modal reasoning	11.88	**3.18**	73.9	66.4	14.51	6.47	44.8	35.7
6	RCM + SIL (unseen)	**10.13**	**2.78**	**79.7**	**73.0**	**9.12**	**4.17**	**69.31**	**61.3**

Wang, Xin, et al. "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Self Monitoring



$$\boldsymbol{h}_t = LSTM([\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{v}}_t, \mathbf{a}_{t-1}])$$



$$z_{t,l}^{\text{textual}} = (\boldsymbol{W}_x \boldsymbol{h}_{t-1})^\top PE(\boldsymbol{x}_l), \quad \text{and} \quad \boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{z}_t^{\text{textual}}),$$

$$z_{t,k}^{\text{visual}} = (\boldsymbol{W}_v \boldsymbol{h}_{t-1})^\top g(\boldsymbol{v}_{t,k}), \quad \text{and} \quad \boldsymbol{\beta}_t = \text{softmax}(\boldsymbol{z}_t^{\text{visual}}),$$

PROGRESS MONITOR

$$\boldsymbol{h}_t^{pm} = \sigma(\boldsymbol{W}_h([\boldsymbol{h}_{t-1}, \hat{\boldsymbol{v}}_t]) \otimes tanh(\boldsymbol{c}_t))$$

$$p_t^{pm} = tanh(\boldsymbol{W}_{pm}([\boldsymbol{\alpha}_t, \boldsymbol{h}_t^{pm}]))$$

completeness of instruction-following

Ma, Chih-Yao, et al. "Self-monitoring navigation agent via auxiliary progress estimation." arXiv preprint arXiv:1901.03035 (2019).

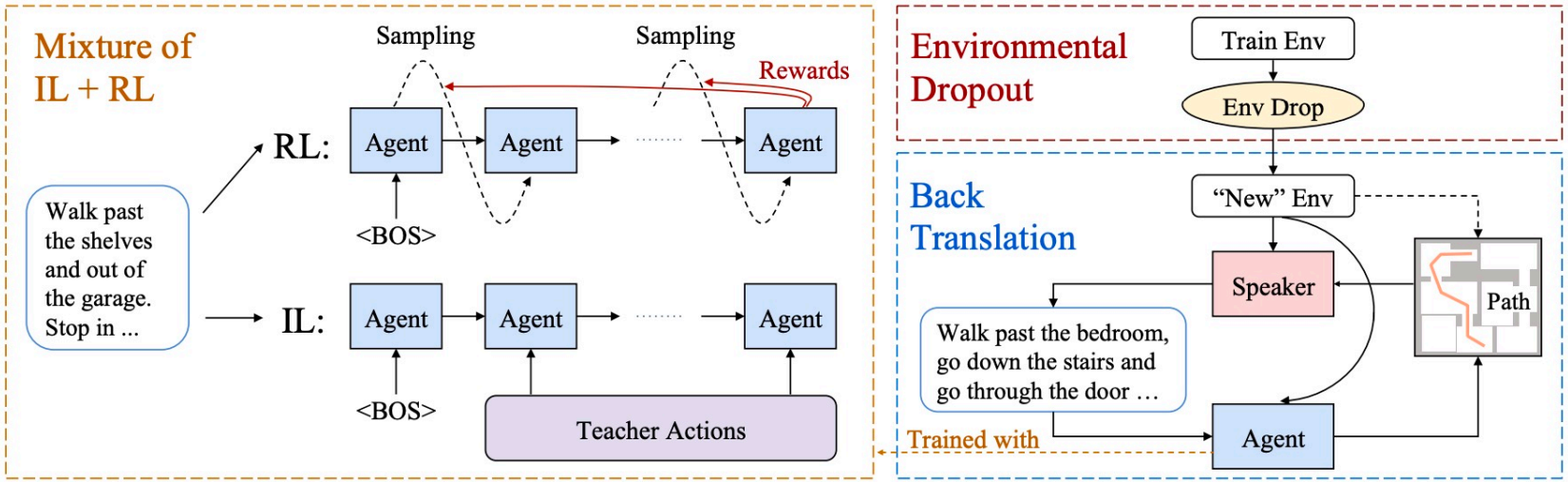
Self Monitoring

Method	Validation-Seen				Validation-Unseen				Test (unseen)			
	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑	NE ↓	SR ↑	OSR ↑	SPL ↑
Random	9.45	0.16	0.21	-	9.23	0.16	0.22	-	9.77	0.13	0.18	-
Student-forcing	6.01	0.39	0.53	-	7.81	0.22	0.28	-	7.85	0.20	0.27	-
RPA	5.56	0.43	0.53	-	7.65	0.25	0.32	-	7.53	0.25	0.33	-
Speaker-Follower	3.88	0.63	0.71	-	5.24	0.50	0.63	-	-	-	-	-
Speaker-Follower*	3.08	0.70	**0.78**	-	4.83	0.55	0.65	-	4.87	0.53	0.64	-
(leaderboard)	-	-	-	-	-	-	-	-	4.87	0.53	0.96	0.01
Ours (beam search)	3.23	0.70	**0.78**	0.66	5.04	0.57	**0.70**	0.51	4.99	0.57	0.68	0.51
(leaderboard)	-	-	-	-	-	-	-	-	4.99	0.57	0.95	0.02
Ours* (beam search)	**3.04**	**0.71**	**0.78**	**0.67**	**4.62**	**0.58**	0.68	**0.52**	**4.48**	**0.61**	0.70	**0.56**
(leaderboard)	-	-	-	-	-	-	-	-	**4.48**	**0.61**	**0.97**	0.02

*: with data augmentation.

Ma, Chih-Yao, et al. "Self-monitoring navigation agent via auxiliary progress estimation." arXiv preprint arXiv:1901.03035 (2019).

Self Monitoring



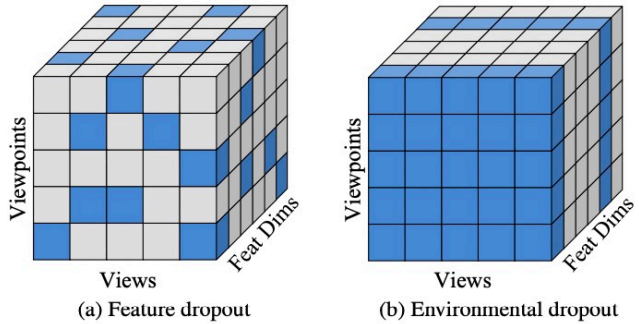


Figure 4: Comparison of the two dropout methods (based on image features).

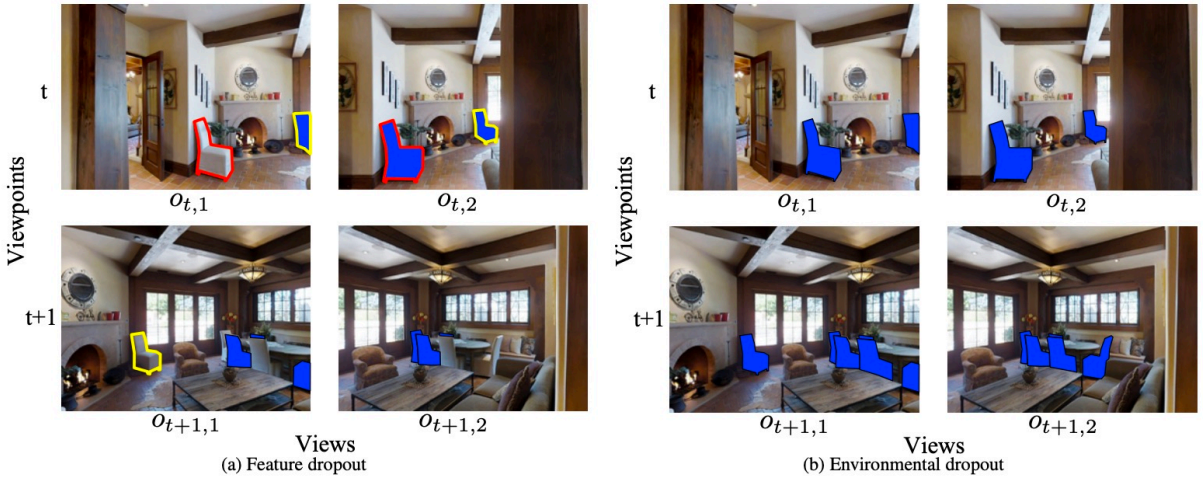


Figure 3: Comparison of the two dropout methods (based on an illustration on an RGB image).

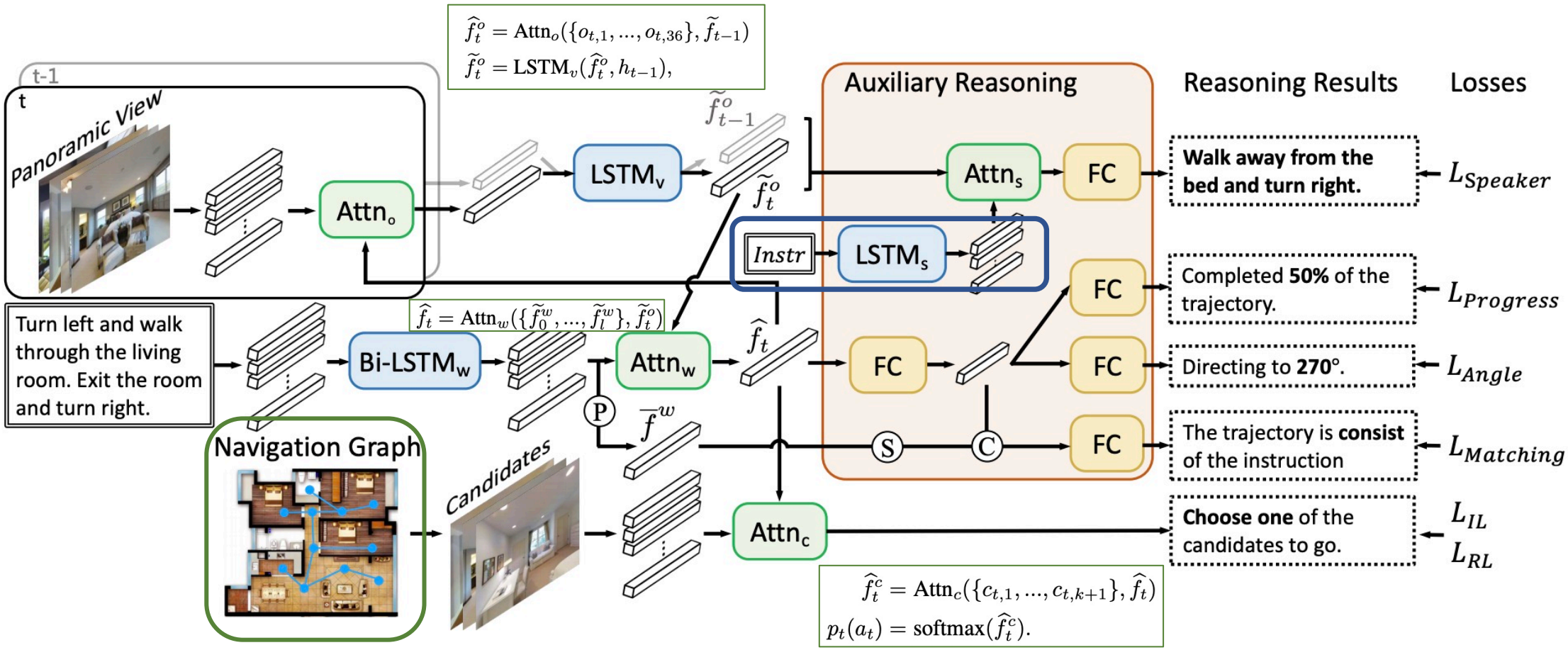
Tan, Hao, Licheng Yu, and Mohit Bansal. "Learning to navigate unseen environments: Back translation with environmental dropout." arXiv preprint arXiv:1904.04195 (2019).

Self Monitoring

Models	Test Unseen (Leader-Board)								
	Single Run			Beam Search			Pre-Explore		
	NL	SR(%)	*SPL*	NL	*SR*(%)	SPL	NL	SR(%)	*SPL*
Random (Anderson et al., 2018b)	9.89	13.2	0.12	-	-	-	-	-	-
Seq-to-Seq (Anderson et al., 2018b)	8.13	20.4	0.18	-	-	-	-	-	-
Look Before You Leap (Wang et al., 2018a)	9.15	25.3	0.23	-	-	-	-	-	-
Speaker-Follower (Fried et al., 2018)	14.8	35.0	0.28	1257	53.5	0.01	-	-	-
Self-Monitoring (Ma et al., 2019)	18.0	48.0	0.35	373	61.0	**0.02**	-	-	-
Reinforced Cross-Modal (Wang et al., 2019)	12.0	43.1	0.38	358	63.0	**0.02**	9.48	60.5	0.59
Ours	11.7	**51.5**	**0.47**	687	**68.9**	0.01	9.79	**63.9**	**0.61**

Tan, Hao, Licheng Yu, and Mohit Bansal. "Learning to navigate unseen environments: Back translation with environmental dropout." arXiv preprint arXiv:1904.04195 (2019).

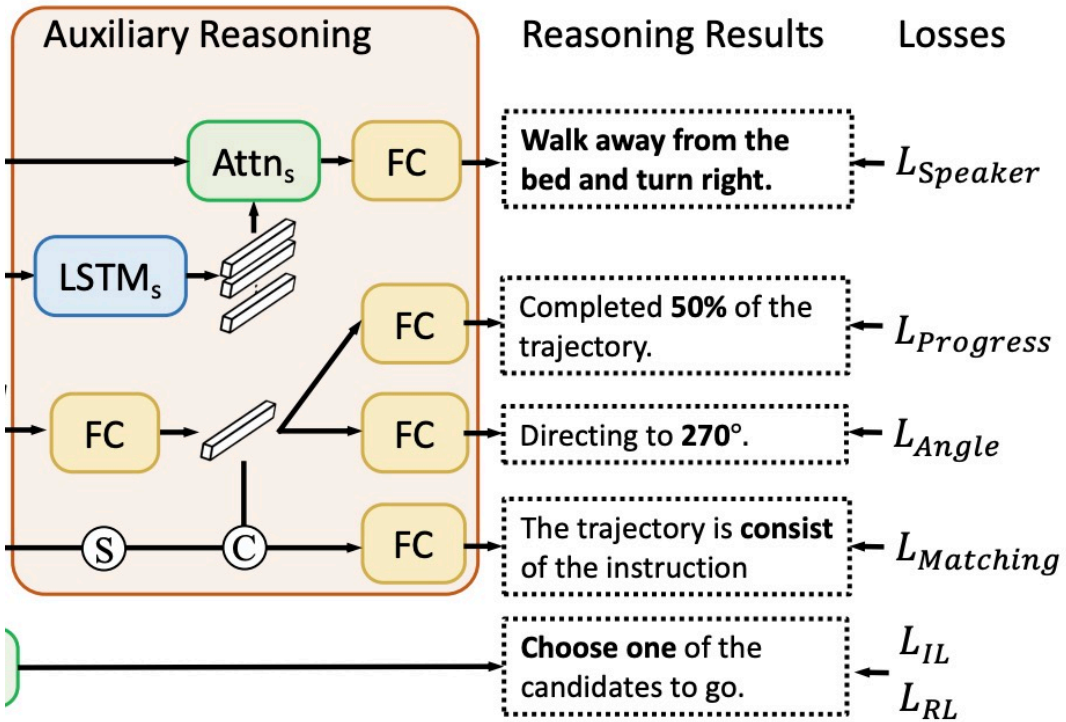
Self Monitoring



obtain the reachable candidates(nodes) from the navigation graph

Zhu, Fengda, et al. "Vision-language navigation with self-supervised auxiliary reasoning tasks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

Self Monitoring



$$L_{Speaker} = -\frac{1}{l}\sum_{i=1}^{l}\log p(w_i|\widehat{f}_i^s).$$

use the percentage of steps r_t, noted as a soft label $\{\frac{t}{T}, 1-\frac{t}{T}\}$ to represent the progress:

$$L_{progress} = -\frac{1}{T}\sum_{t=1}^{T} r_t \log \sigma(W_r \widehat{f}_t)$$

$$L_{angle} = -\frac{1}{T}\sum_{t=1}^{T} \parallel e_t - W_e \widehat{f}_t \parallel,$$

$$L_{Matching} = -\frac{1}{T}\sum_{t=1}^{T} m_t \log \sigma(W_m[\widehat{f}_t, \overline{f'}^w]),$$

m_t: a binary label indicating whether the feature has been shuffled or remains unchanged

Zhu, Fengda, et al. "Vision-language navigation with self-supervised auxiliary reasoning tasks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

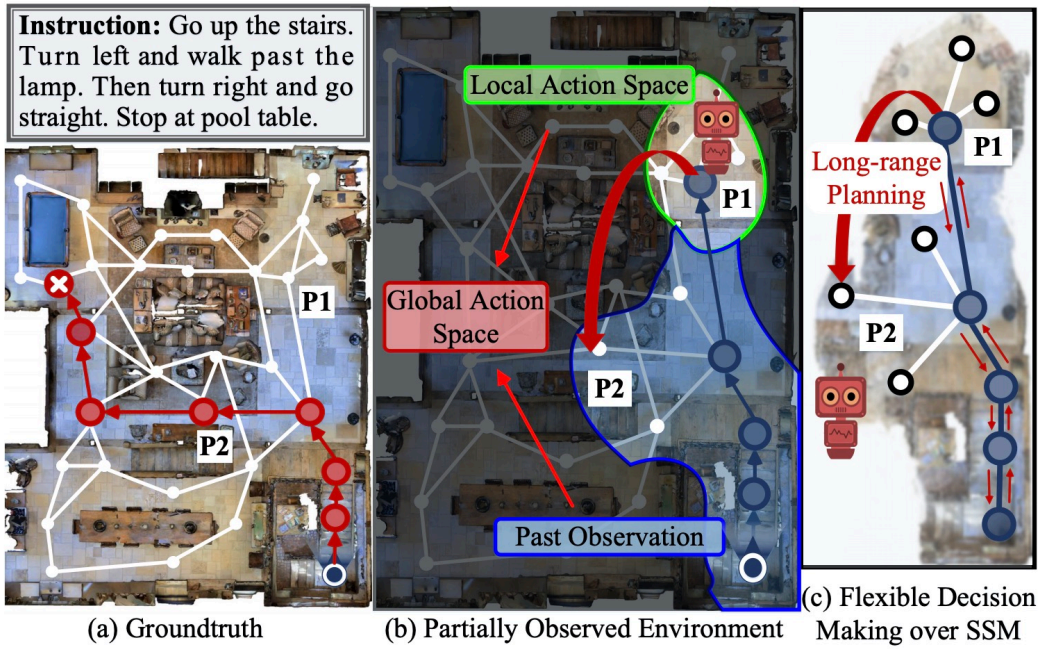
Self Monitoring

Leader-Board (Test Unseen)	Single Run				Pre-explore				Beam Search		
Models	NE	OR	SR	*SPL*	NE	OR	SR	*SPL*	TL	SR	*SPL*
Random [5]	9.79	0.18	0.17	0.12	-	-	-	-	-	-	-
Seq-to-Seq [5]	20.4	0.27	0.20	0.18	-	-	-	-	-	-	-
Look Before You Leap [42]	7.5	0.32	0.25	0.23	-	-	-	-	-	-	-
Speaker-Follower [10]	6.62	0.44	0.35	0.28	-	-	-	-	1257	0.54	0.01
Self-Monitoring [23]	5.67	0.59	0.48	0.35	-	-	-	-	373	0.61	0.02
The Regretful Agent [48]	5.69	0.48	0.56	0.40	-	-	-	-	13.69	0.48	0.40
FAST [49]	5.14	-	0.54	0.41	-	-	-	-	196.53	0.61	0.03
Reinforced Cross-Modal [41]	6.12	0.50	0.43	0.38	4.21	0.67	0.61	0.59	358	0.63	0.02
ALTR [51]	5.49	-	0.48	0.45	-	-	-	-	-	-	-
Environmental Dropout [37]	5.23	0.59	0.51	0.47	3.97	0.70	0.64	0.61	687	0.69	0.01
AuxRN(Ours)	**5.15**	**0.62**	**0.55**	**0.51**	**3.69**	**0.75**	**0.68**	**0.65**	41	**0.71**	**0.21**

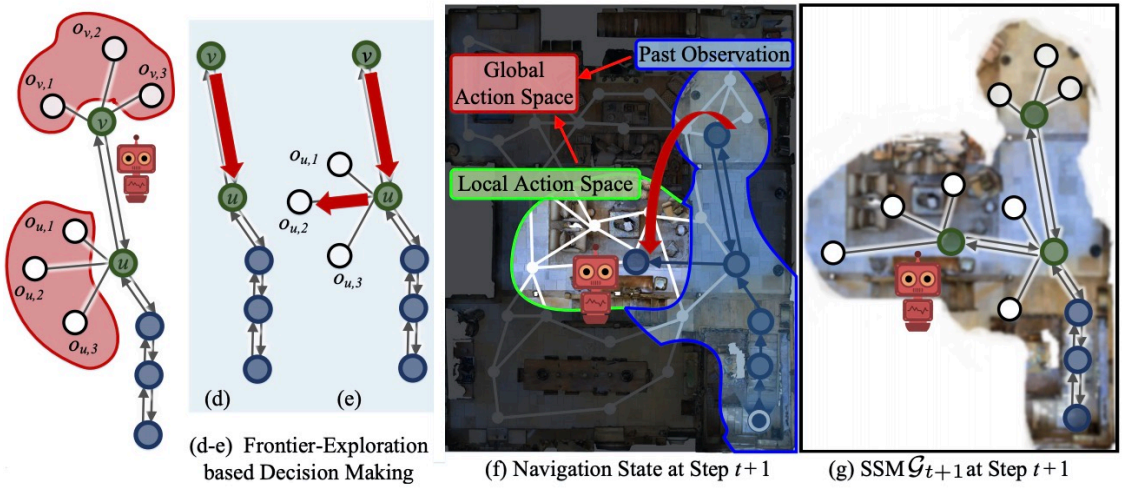
Zhu, Fengda, et al. "Vision-language navigation with self-supervised auxiliary reasoning tasks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

- Introduction
- Dataset & Platform
- Models
- **Future Works**

Continuous Environments



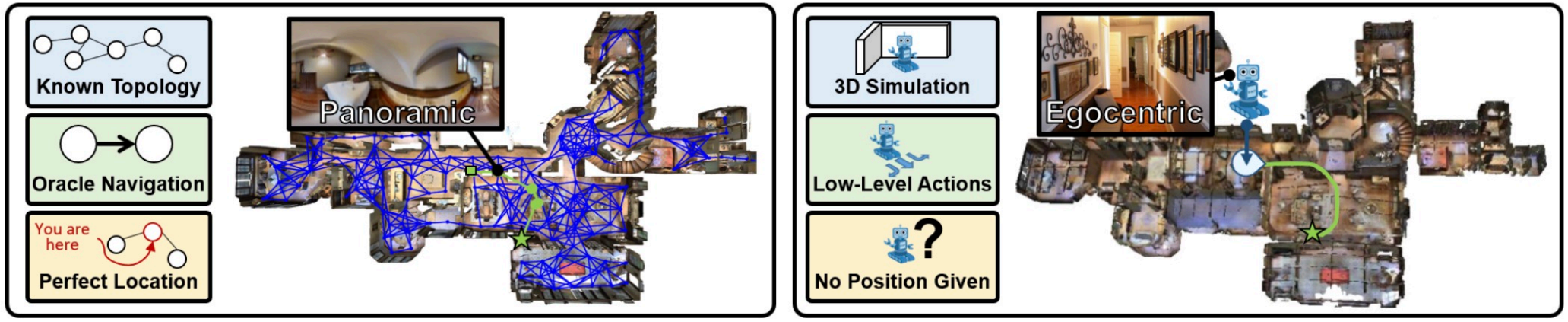
(a) Groundtruth (b) Partially Observed Environment (c) Flexible Decision Making over SSM



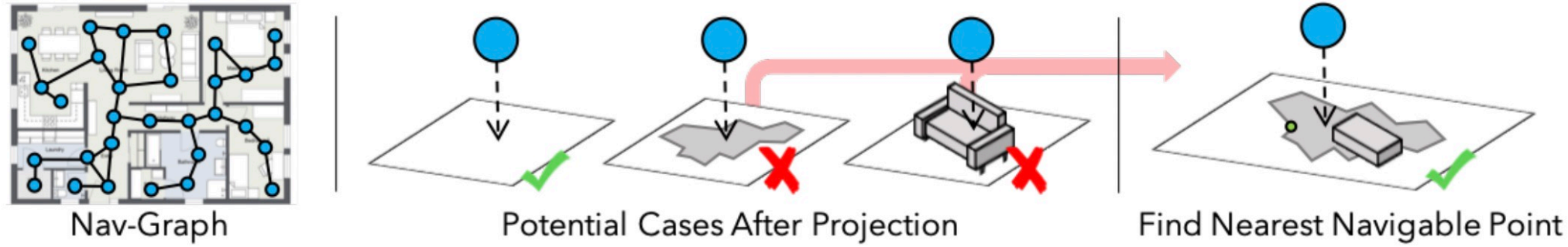
(d-e) Frontier-Exploration based Decision Making (f) Navigation State at Step $t+1$ (g) SSM $\mathcal{G}_{t+1}$ at Step $t+1$

Krantz, Jacob, et al. "Beyond the nav-graph: Vision-and-language navigation in continuous environments." European Conference on Computer Vision. Springer, Cham, 2020.

Continuous Environments

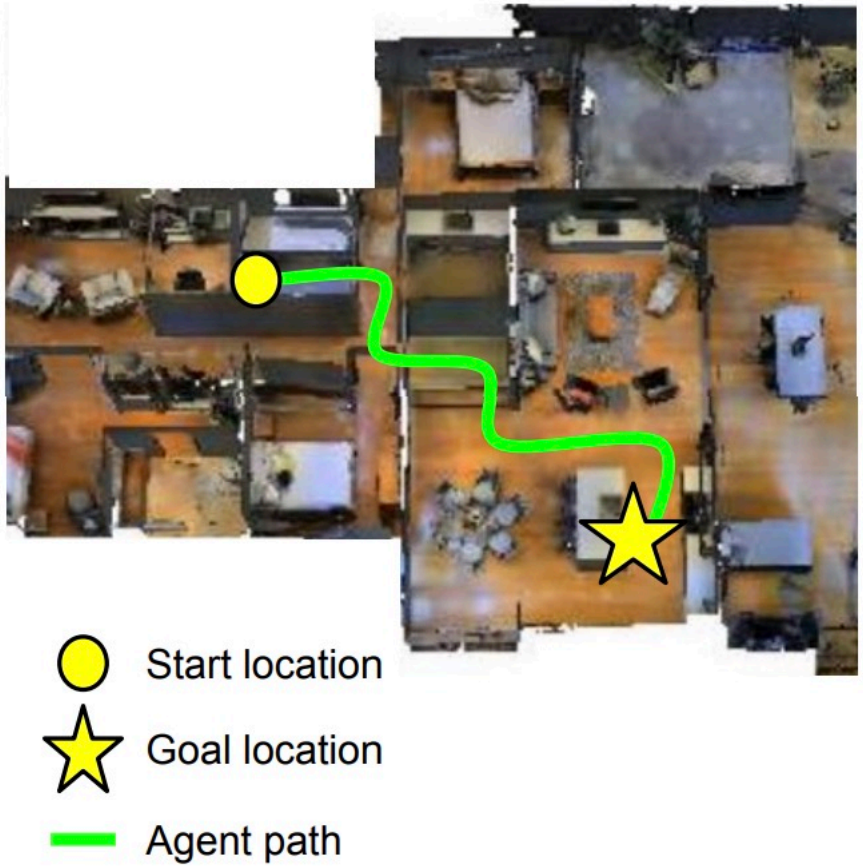


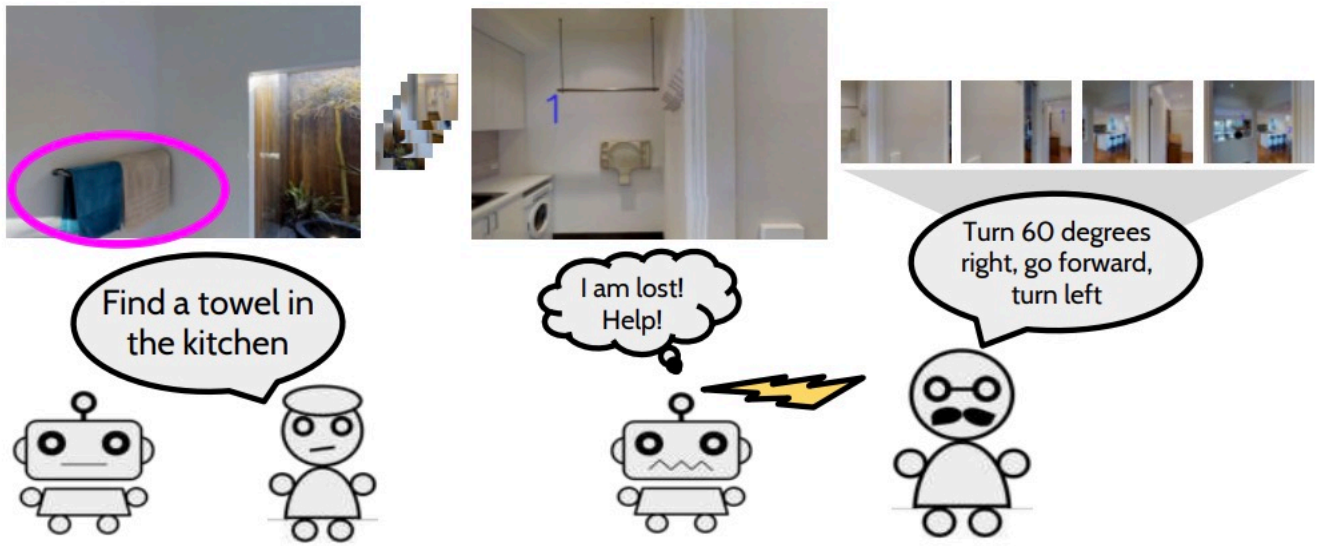
(a) Vision-and-Language Navigation (VLN) (b) VLN in Continuous Environments (VLN-CE)



Krantz, Jacob, et al. "Beyond the nav-graph: Vision-and-language navigation in continuous environments." European Conference on Computer Vision. Springer, Cham, 2020.

Grounding via Interaction





Nguyen, Khanh, et al. "Vision-based navigation with language-based assistance via imitation learning with indirect intervention." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Thanks