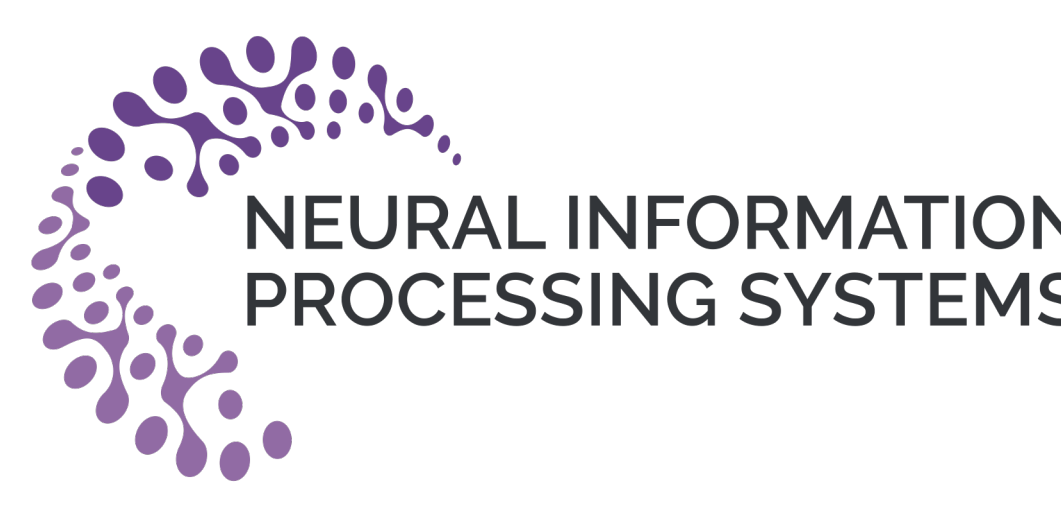
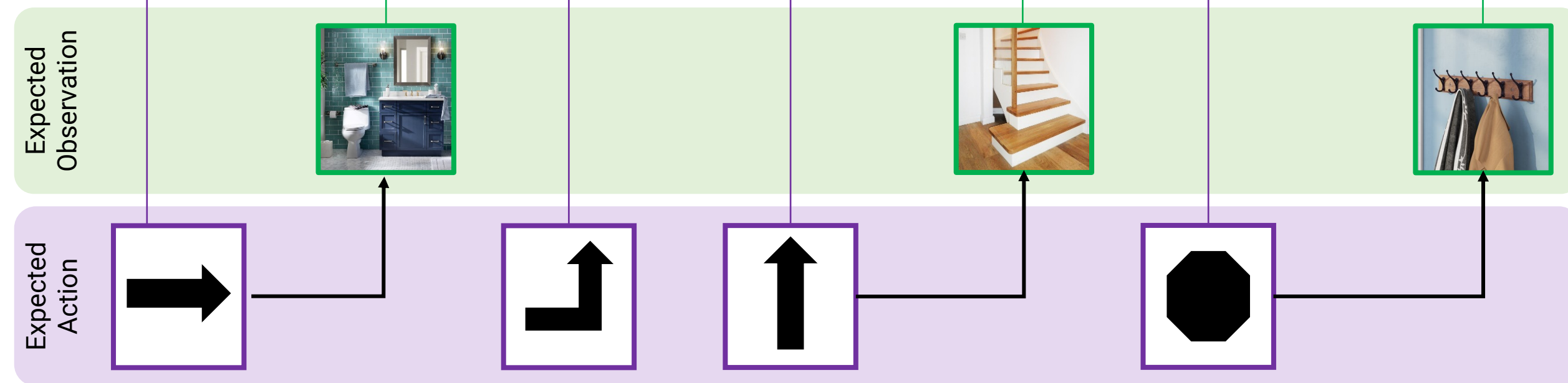




1 INTUITION: UNPACKING A NAVIGATION INSTRUCTION

A **visually-grounded navigation instruction** can be interpreted as a sequence of expected **observations** and **actions** an agent following the correct trajectory would encounter and perform.

Walk out of the **bathroom**, **turn left**, and **go on** to the **stairs** and **wait** near the **coat rack**.

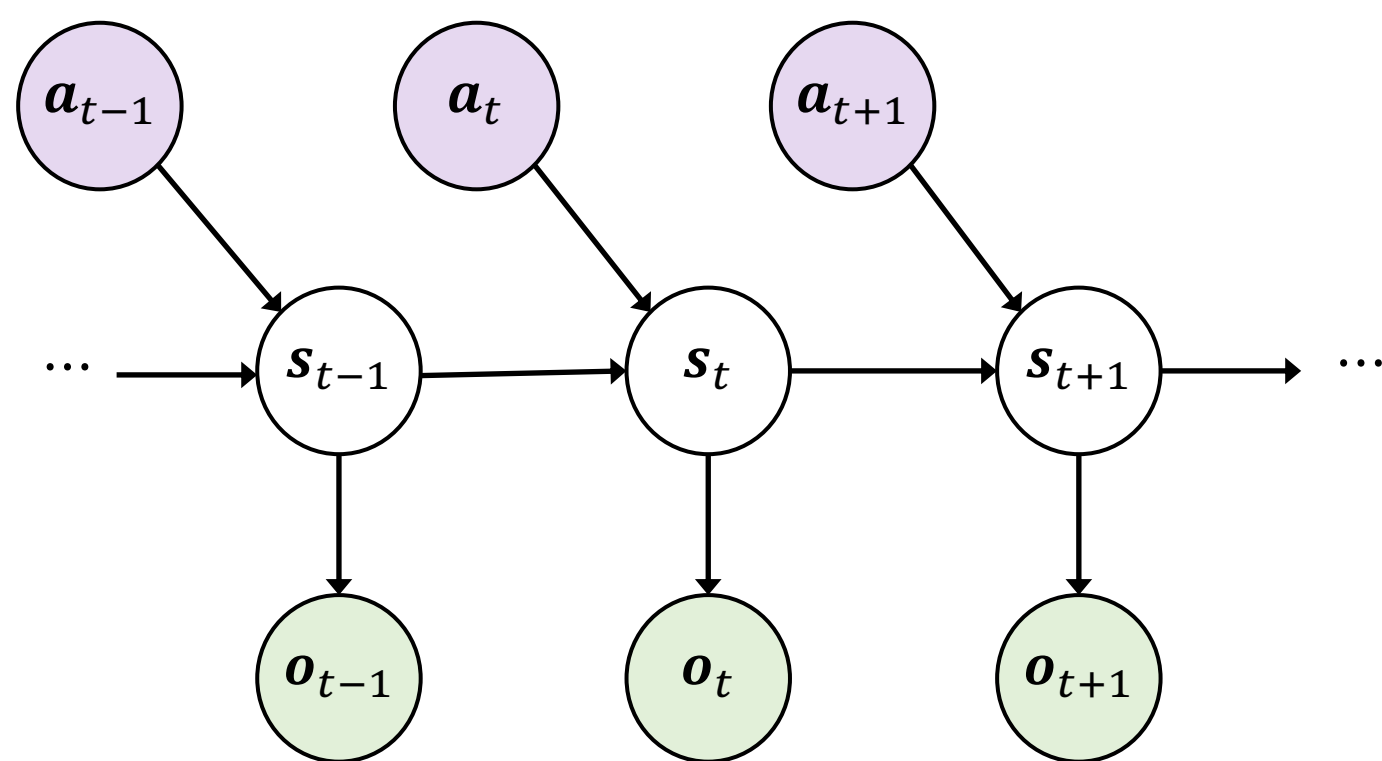


2 BACKGROUND: BAYESIAN STATE TRACKING

Given a sequence of **observations** $\mathbf{o}_{1:T}$ and **actions** $\mathbf{a}_{1:T}$ extracted from a natural language instruction, how should we determine the final (goal) location $\mathbf{s}_T$?

Key idea: Use a **Bayes filter** to track the trajectory to goal, maintaining a probability distribution over the location state $\mathbf{s}_t$ from start $\mathbf{s}_0$ to goal $\mathbf{s}_T$

i.e. at each time step t , compute $bel(\mathbf{s}_t) = p(\mathbf{s}_t | \mathbf{a}_{1:t}, \mathbf{o}_{1:t})$ also called *belief*.



Motion Update: $\bar{bel}(\mathbf{s}_t) = \int p(\mathbf{s}_t | \mathbf{s}_{t-1}, \mathbf{a}_t) \bar{bel}(\mathbf{s}_{t-1}) d\mathbf{s}_{t-1}$

Observation Update: $bel(\mathbf{s}_t) = \eta p(\mathbf{o}_t | \mathbf{s}_t) \bar{bel}(\mathbf{s}_t)$

Recent work [1-3] show Bayes filter can be embedded into deep neural networks.

3 REFERENCES

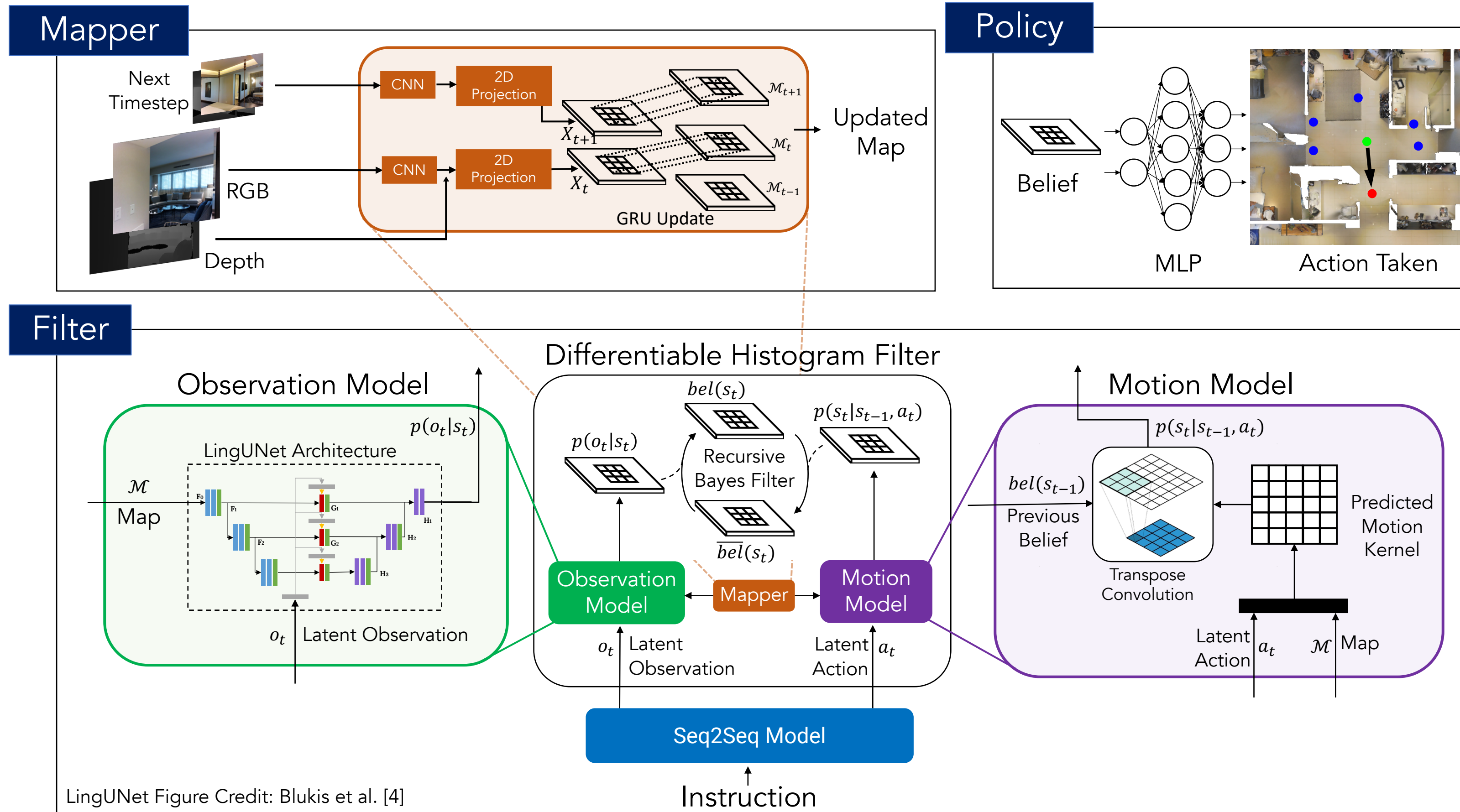
[1] Rico Jonschkowski and Oliver Brock. End-to-end learnable histogram filters. In *In Workshop on Deep Learning for Action and Interaction at the Conference on Neural Information Processing Systems (NIPS)*, 2016.

[2] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

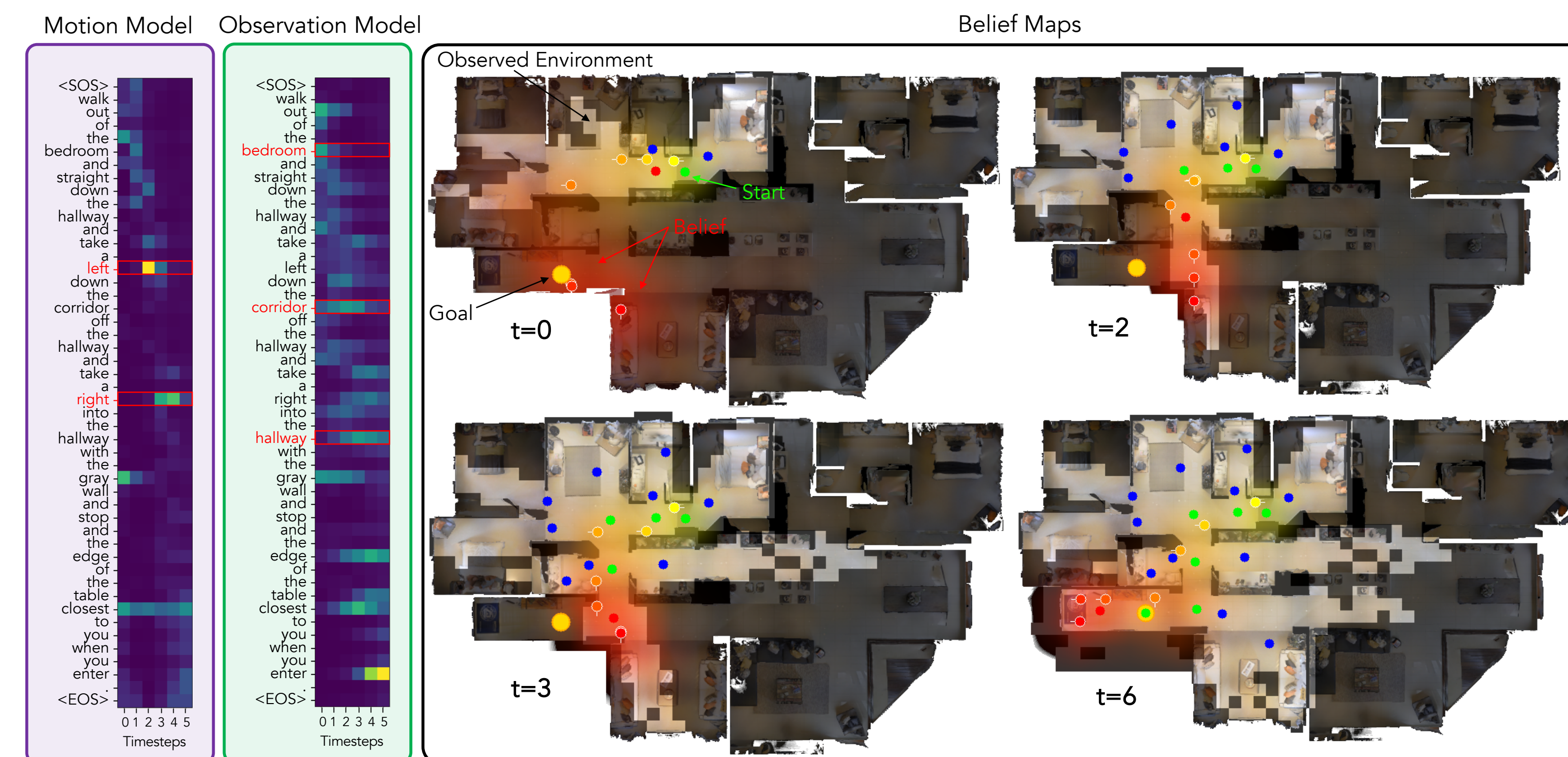
[3] Peter Karkus, David Hsu, and Wee Sun Lee. Particle filter networks with application to visual localization. In *CoRL*, 2018.

[4] Valts Blukis, Dipendra Misra, Ross A Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position-visitation prediction. In *CoRL*, 2018.

4 AGENT MODEL



5 INTERPRETABILITY OF MODEL



6 RESULTS

Goal Location Prediction Task [Mapper + Filter]



- Trained/eval'ed without Policy
- Fixed trajectories move towards goal with 50% probability
- Adding Bayes filter structure improves over just using LingUNet [4]
- Including heading in the state is important for modeling oriented instructions (e.g., "pass the kitchen on your left")

Vision and Language Navigation (VLN) Task [Mapper + Filter + Policy]

Model	RL	Val-Seen						Val-Unseen					
		Aug	TL	NE	OS	SR	SPL	TL	NE	OS	SR	SPL	
Speaker-Follower	✓	-	3.36	0.74	0.66	-	-	6.62	0.45	0.36	-	-	
RCM	✓	-	10.65	3.53	0.75	0.67	-	11.46	6.09	0.50	0.43	-	
Regretful Agent	✓	-	3.23	0.77	0.69	0.63	-	5.32	0.59	0.50	0.41	-	
FAST	✓	-	-	-	-	-	-	21.1	4.97	-	0.56	0.43	
Back Translation	✓	✓	11.0	3.99	-	0.62	0.59	10.7	5.22	-	0.52	0.48	
Speaker-Follower	-	-	4.86	0.63	0.52	-	-	7.07	0.41	0.31	-	-	
Back Translation	-	-	10.3	5.39	-	0.48	0.46	9.15	6.25	-	0.44	0.40	
Ours	-	-	10.15	7.59	0.42	0.34	0.30	9.64	7.20	0.44	0.35	0.31	

- Training trajectories: Sampled from Policy with 50% probability, otherwise GT
- Credible performance on the full VLN task compared to existing models with no RL and no data augmentation
- Improved generalization from seen to unseen environments

7 CONCLUSION

- Instruction following can be formulated as **Bayesian State Tracking** with observations and actions extracted from the instruction.
- Advantages of this approach:
 - **Uncertainty:** an explicit probability for every trajectory the agent could take (naturally handles multimodal hypotheses)
 - **Interpretability:** inspect the predicted goal location distribution
 - **Performance:** Improved goal location prediction
- Ideas for future work:
 - More sophisticated policy module, RL training and data augmentation
 - Reasoning about unseen map regions