Chasing Ghosts : Instruction Following as Bayesian State Tracking

Peter Anderson*[1], Ayush Shrivastava*[1], Devi Parikh[1,2], Dhruv Batra[1,2], Stefan Lee[3]

Georgia Institute of Technology[1] Facebook AI Research[2] Oregon State University[3] * denotes equal contribution

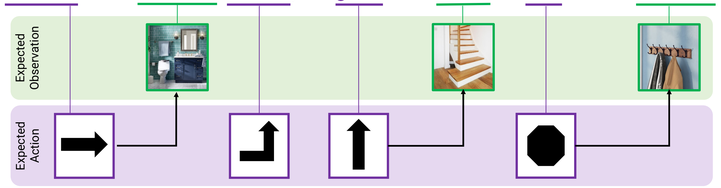
arXiv | Video | Code

NEURAL INFORMATION PROCESSING SYSTEMS

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**.

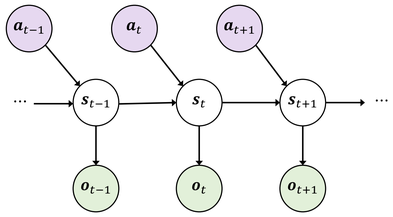


2 Background: Bayesian State Tracking

Given a sequence of observations $\boldsymbol{o}_{1:T}$ and actions $\boldsymbol{a}_{1:T}$, how should we determine the final location $\boldsymbol{s}_T$?

Use **Bayes filter** to estimate probability distribution over latent state $\boldsymbol{s}_T$ given $\boldsymbol{o}_{1:T}$ and $\boldsymbol{a}_{1:T}$.

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



Motion Update: (Motion Model)

$$\overline{bel}(\boldsymbol{s}_t) = \int p(\boldsymbol{s}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_t)\, bel(\boldsymbol{s}_{t-1})\, d\boldsymbol{s}_{t-1}$$

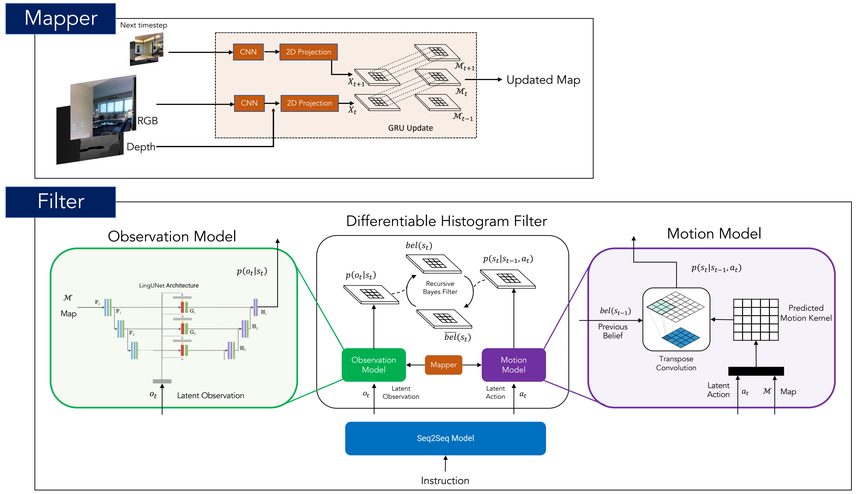
Observation Update:

$$bel(\boldsymbol{s}_t) = \eta\, p(\boldsymbol{o}_t|\boldsymbol{s}_t)\, \overline{bel}(\boldsymbol{s}_t)$$

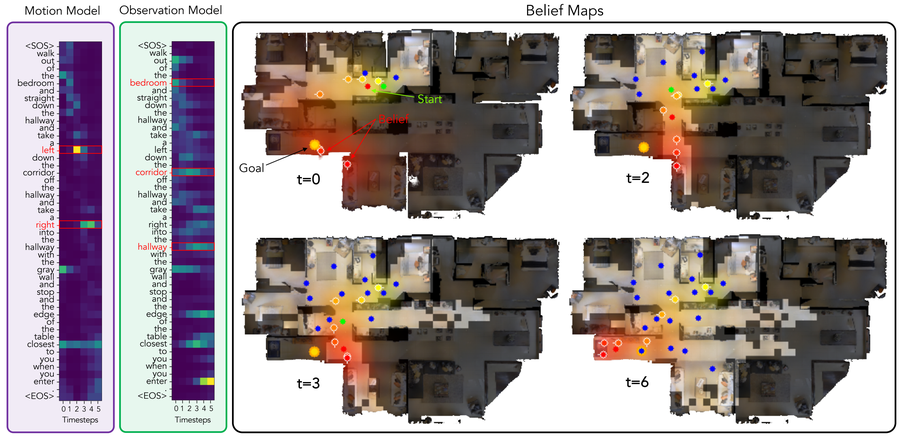
(Observation Model)

- Recent work show Bayes filter can be embedded into deep neural networks

3 Agent Model

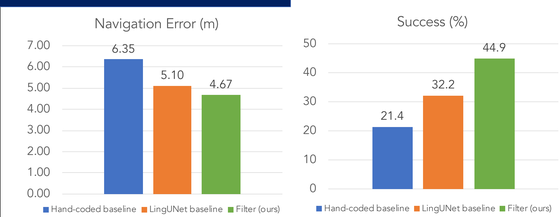


4 Interpretability of Model [Change this Heading]



5 Results

Goal Location Prediction



- Our model outperforms strong baseline (LingUNet)

Vision and Language Navigation (VLN) Task

			Val-Seen					Val-Unseen					Test				
Model	RL	Aug	TL	NE	OS	SR	SPL	TL	NE	OS	SR	SPL	TL	NE	OS	SR	SPL
RPA	✓		8.46	5.56	0.53	0.43	-	7.22	7.65	0.32	0.25	-	9.15	7.53	0.32	0.25	0.23
Speaker-Follower		✓	-	3.36	0.74	0.66	-	-	6.62	0.45	0.36	-	14.82	6.62	0.44	0.35	0.28
RCM	✓		10.65	3.53	0.75	0.67	-	11.46	6.09	0.50	0.43	-	11.97	6.12	0.50	0.43	0.38
Self-Monitoring		✓	-	3.18	0.77	0.68	0.58	-	5.41	0.59	0.47	0.34	18.04	5.67	0.59	0.48	0.35
Regretful Agent		✓	-	3.23	0.77	0.69	0.63	-	5.32	0.59	0.50	0.41	13.69	5.69	0.56	0.48	0.40
FAST	✓		-	-	-	-	-	21.1	4.97	-	0.56	0.43	22.08	**5.14**	**0.64**	**0.54**	0.41
Back Translation	✓	✓	11.0	3.99	-	0.62	0.59	10.7	5.22	-	0.52	0.48	11.66	5.23	0.59	0.51	**0.47**
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	10.03	7.83	0.42	0.33	0.30

- Our model achieves credible results on the full VLN task.

6 Conclusion

1. Instruction following can be formulated as **Bayesian State Tracking** where model maintains
   - a **semantic map** of the environment,
   - an **explicit probability distribution** over **alternative possible trajectories** in the map.
2. Our approach outperforms strong baseline for **goal location prediction**.
3. Credible results on the full VLN task without using RL or data augmentation.