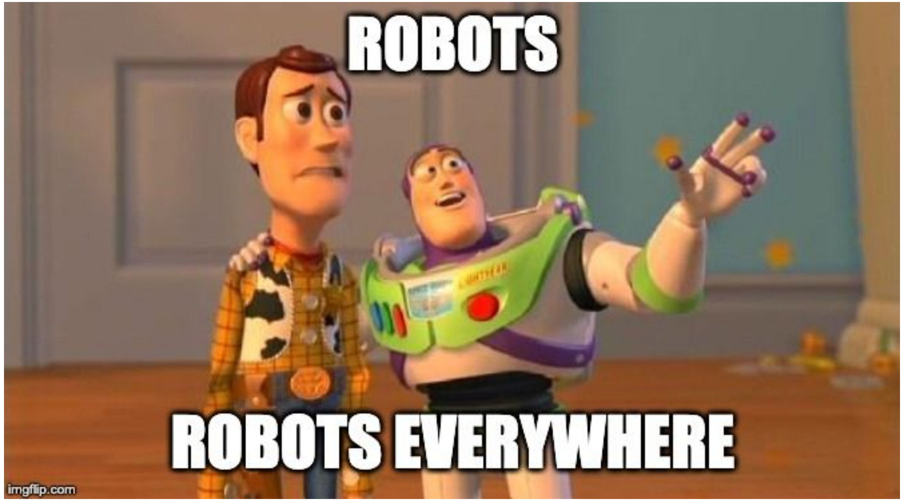
Reinforced Cross-Modal Matching and Self-Supervision Imitation Learning for Vision-Language Navigation

Presented by George Revkov, Yurong Zhuang, Yuchen Zhou

Vision-Language Navigation



Vision-Language Navigation

Task of navigating inside real environments following natural language instructions

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

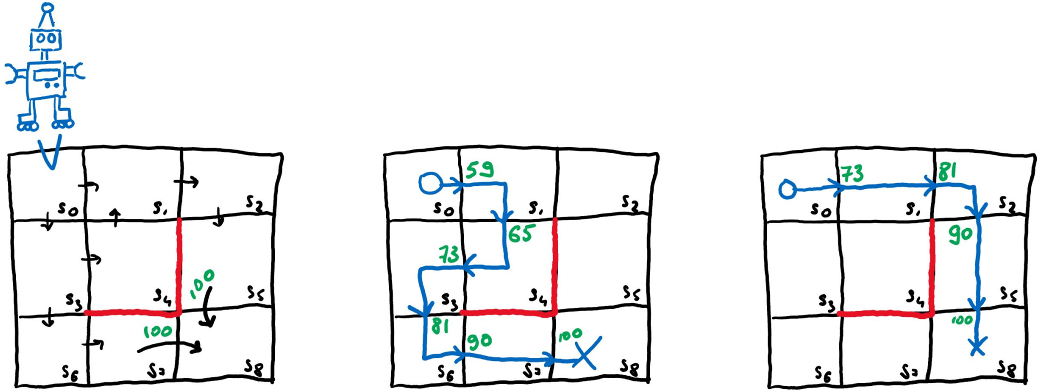
Aside: Reinforcement Learning

- **Agent** observes the **environment**, identifies its current **state s_t ∈ S**, takes an **action a_t ∈ A,** and gets a **reward** (may be = 0)
- Goal 1: from each possible state find a sequence of actions that will maximize total reward
- Goal 2: come up with a **policy π: S -> A**
- **Markov Decision Process (MDP)**

Aside: Reinforcement Learning

- States **S = {s_0 … s_8}**
- Actions **A(S_t) = {U, D, R, L}**
- Transition F-n **T: S, A -> S'**
- Reward F-n **R(S, A)**
  - $r_t = r(s_t, a_t)$
- Policy **π: S -> A**
- **Discounted Cumulative Reward:**

$V^{\pi}(s_t) = r_t + \gamma * r_{t+1} + \gamma^2 * r_{t+2} + \ldots$



- Say γ = .9
- $V^{\pi}(s_0)$ = 0 + 0*.9 + 0*.9^2 + 100*.9^3 = 73
- $\pi(S = s_0)$ = R, $\pi(S = s_1)$ = R,

  $\pi(S = s_2)$ = D, $\pi(S = s_5)$ = D

Challenges

- Reasoning over visual images & natural language instructions is difficult
    - Ground an instruction in the local scene
    - Match the instruction to the visual trajectory over time
- Ill-posed feedback
- Generalization problem

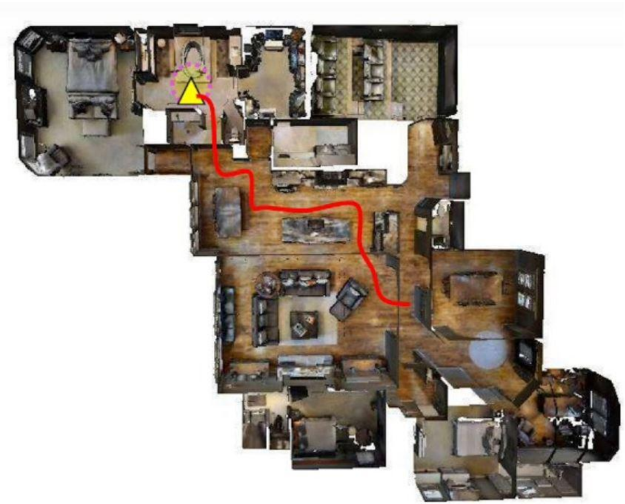
Summary

- Novel Reinforced Cross-Modal Matching approach
  - **Reasoning Navigator** (based on where I am right now / what I’m seeing):
    - What should I be doing?
    - Which sub-instruction should execute?
  - **Matching critic**:
    - What is the probability of reconstructing the instruction based on executed trajectory
    - Penalize the trajectories that don’t match instructions

Matching Critic

Enforces the agent to follow the instruction

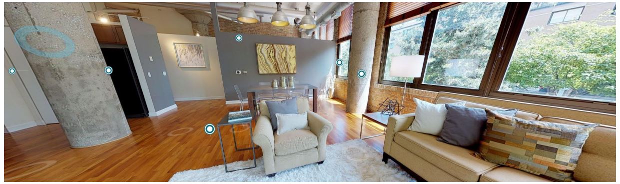
by rewarding trajectories that match the instruction



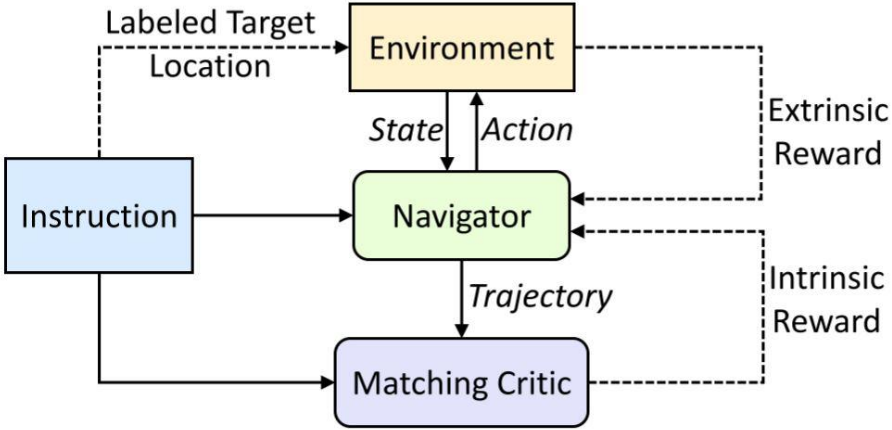
Summary

- Use Reinforcement Learning with two reward functions
- Use Self-Supervised Imitation learning to deal with unseen environments
- *Establishes new state-of-the-art performance* on R2R dataset
- Proposes a new evaluation setting for VLN:
    - Exploring unseen environments prior to testing is allowed

RCM Model



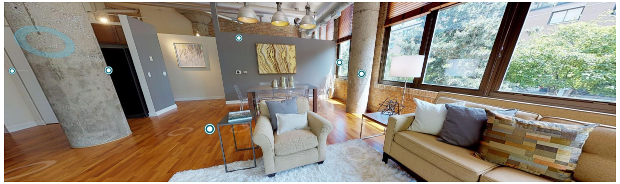
Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front
of the toilet.



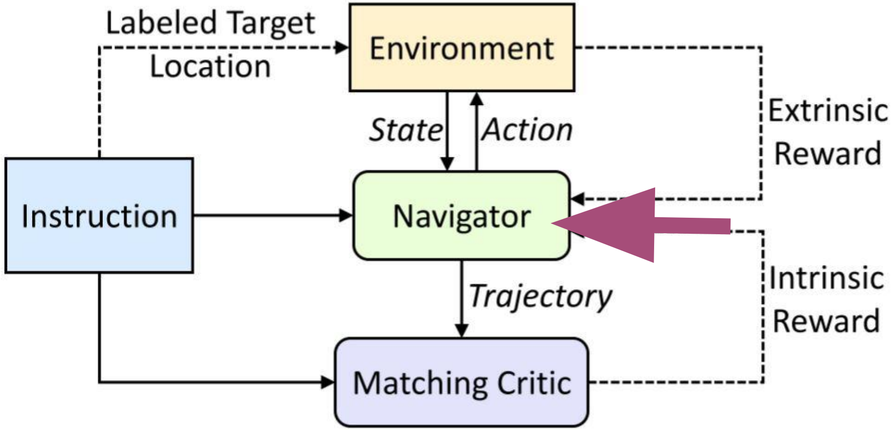
Close to the target?

Follows the instruction?

RCM Model



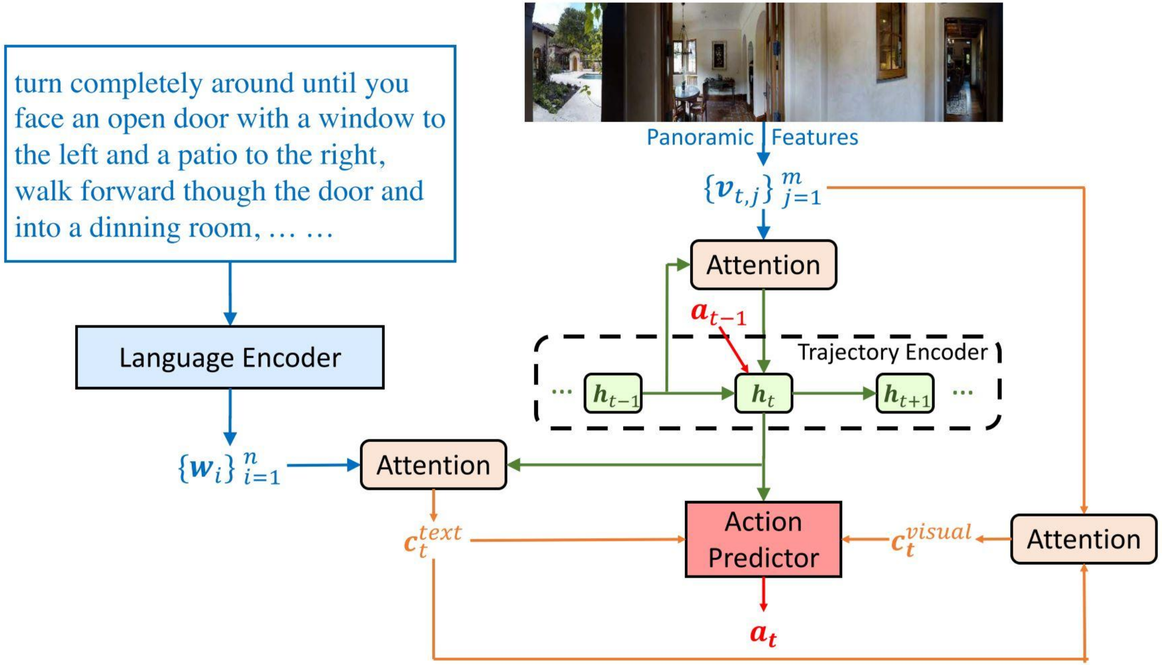
*Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front
of the toilet.*



Close to the target?

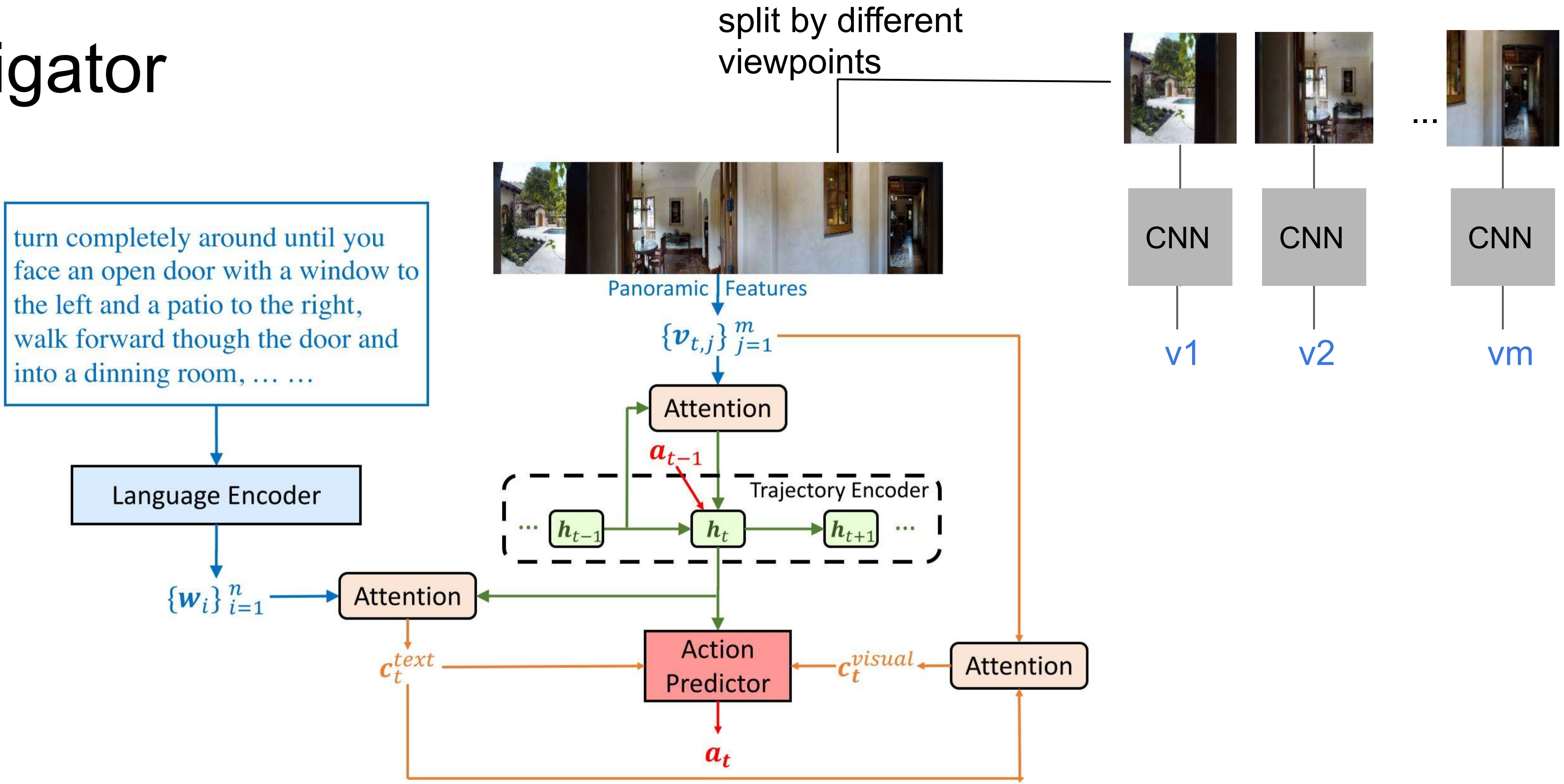
Follows the instruction?

Navigator

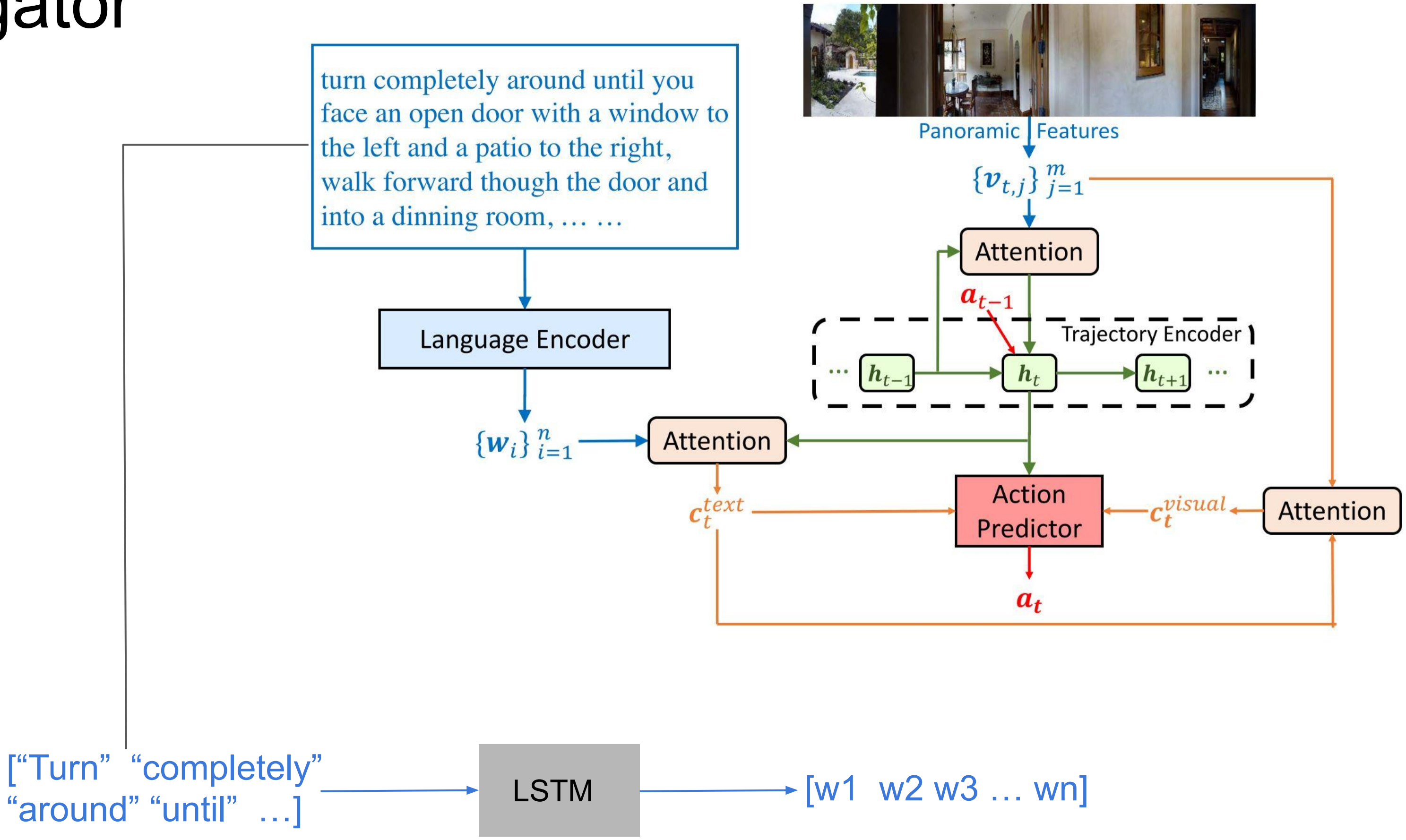


turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Language Encoder
$\{w_i\}_{i=1}^{n}$
Attention
Panoramic Features
$\{v_{t,j}\}_{j=1}^{m}$
Attention
a_{t-1}
Trajectory Encoder
… h_{t-1} h_t h_{t+1} …
Action Predictor
a_t
c_t^{text}
c_t^{visual}
Attention

Navigator


split by different viewpoints
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Language Encoder
$\{w_i\}_{i=1}^{n}$
Attention
Panoramic Features
$\{v_{t,j}\}_{j=1}^{m}$
Attention
a_{t-1}
Trajectory Encoder
h_{t-1}
h_t
h_{t+1}
Action Predictor
c_t^{text}
c_t^{visual}
Attention
a_t
CNN
CNN
CNN
v1
v2
vm

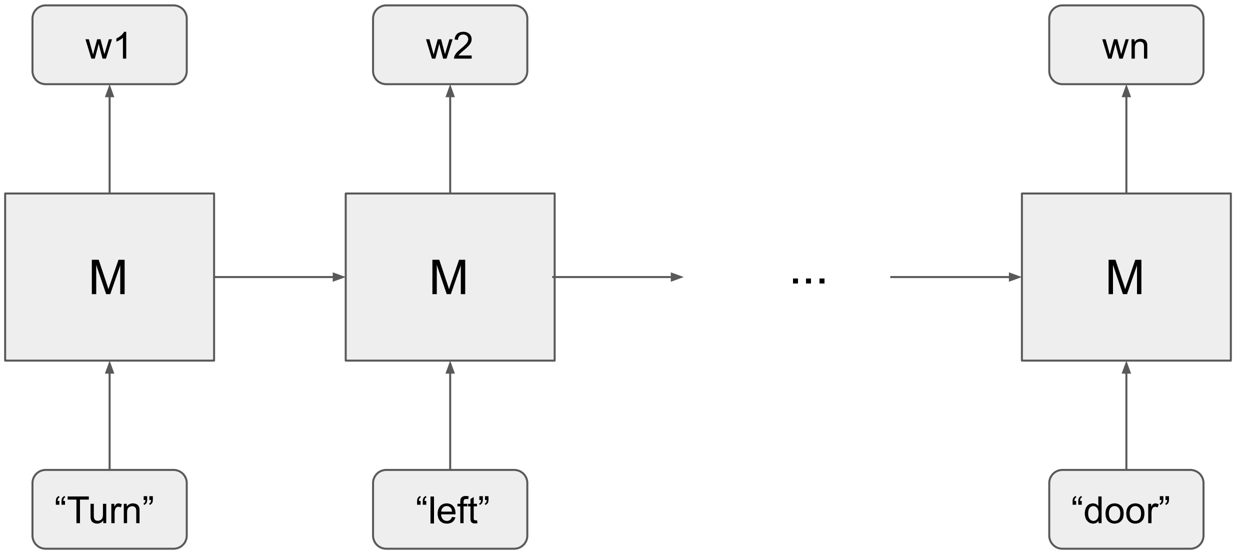
Navigator



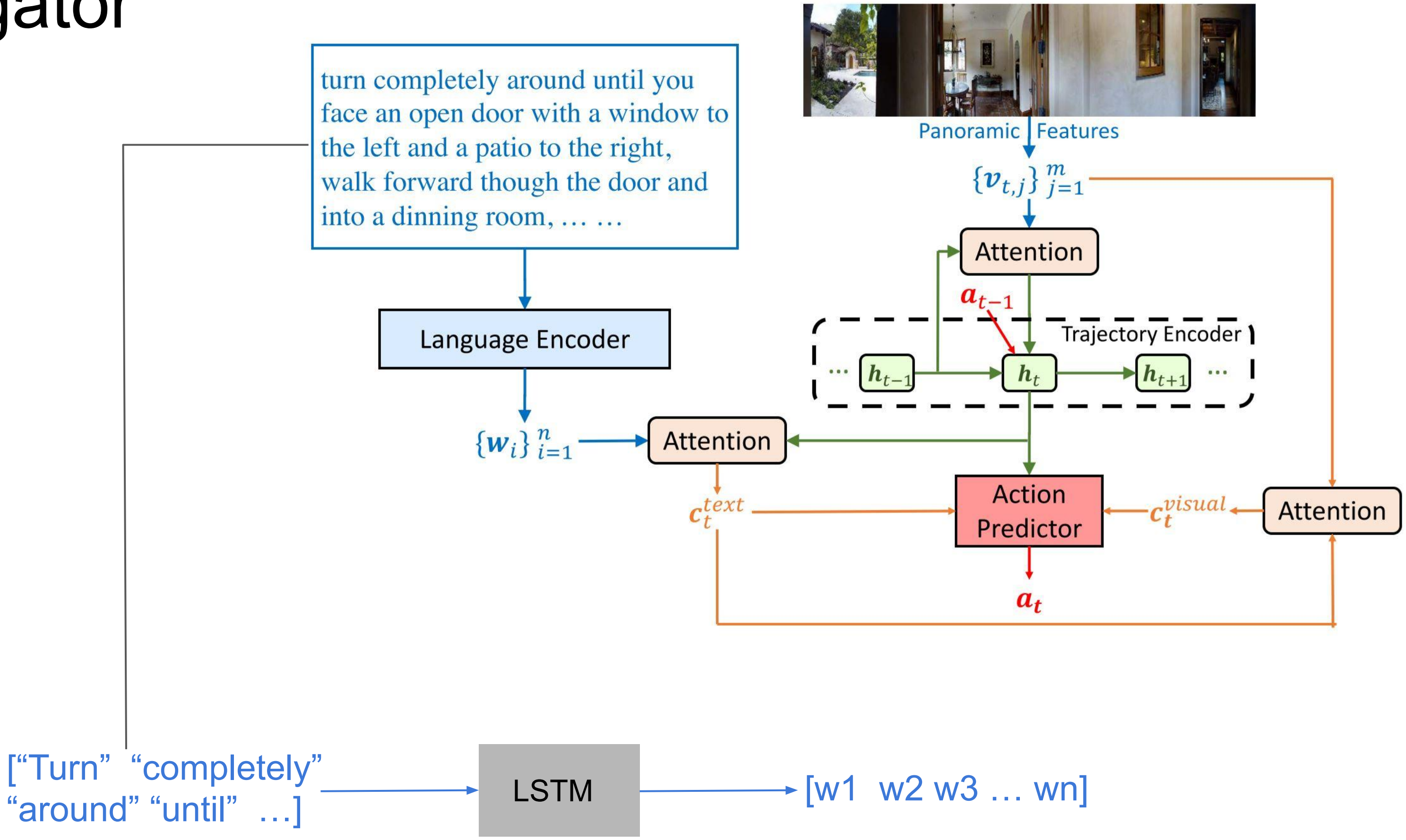
turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Language Encoder
$\{w_i\}_{i=1}^{n}$
Attention
Panoramic Features
$\{v_{t,j}\}_{j=1}^{m}$
Attention
a_{t-1}
Trajectory Encoder
… h_{t-1} h_t h_{t+1} …
c_t^{text}
Action Predictor
c_t^{visual}
Attention
a_t
["Turn" "completely" "around" "until" …]
LSTM
[w1 w2 w3 … wn]

LSTM

- Good for representation learning of sequential data

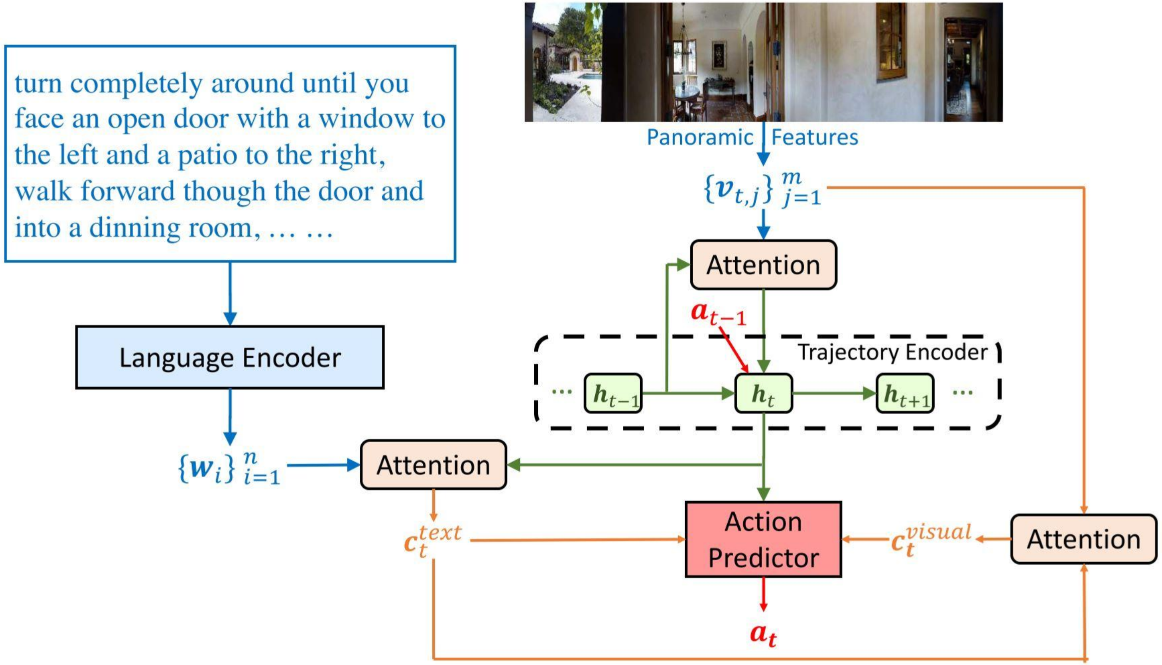


Navigator



turn completely around until you face an open door with a window to the left and a patio to the right, walk forward though the door and into a dinning room, … …
Language Encoder
$\{w_i\}_{i=1}^{n}$
Panoramic Features
$\{v_{t,j}\}_{j=1}^{m}$
Attention
a_{t-1}
Trajectory Encoder
… h_{t-1} h_t h_{t+1} …
Attention
c_t^{text}
Action Predictor
c_t^{visual}
Attention
a_t
["Turn" "completely" "around" "until" …]
LSTM
[w1 w2 w3 … wn]

Cross modal reasoning



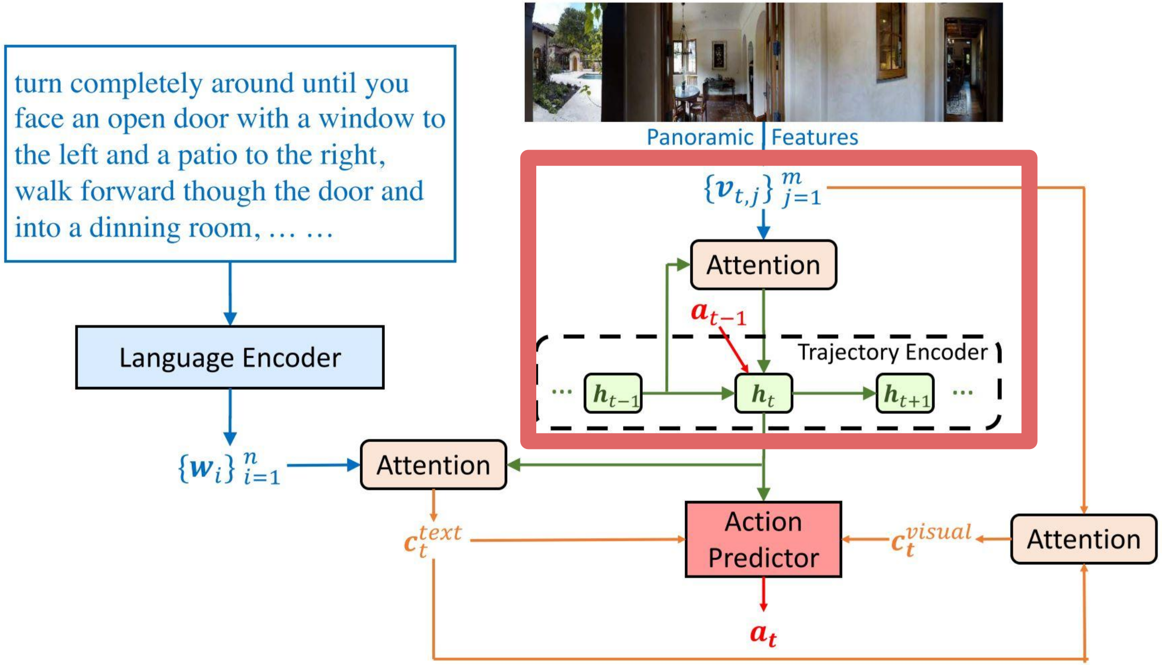
Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

.

Cross modal reasoning



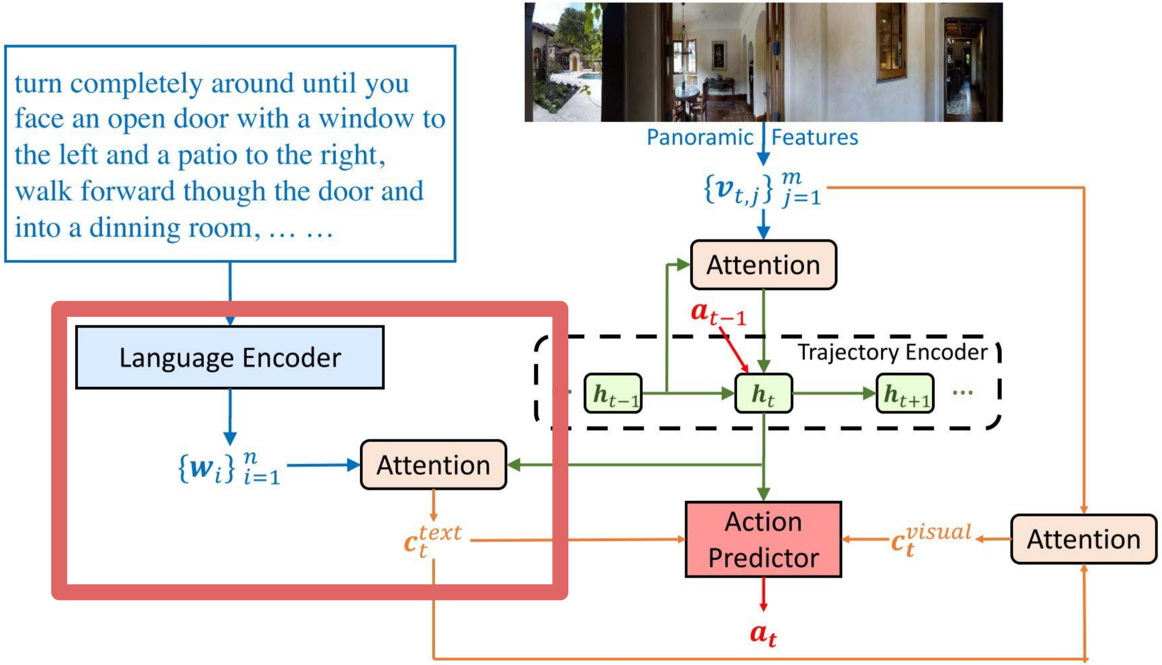
Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

.

Cross modal reasoning



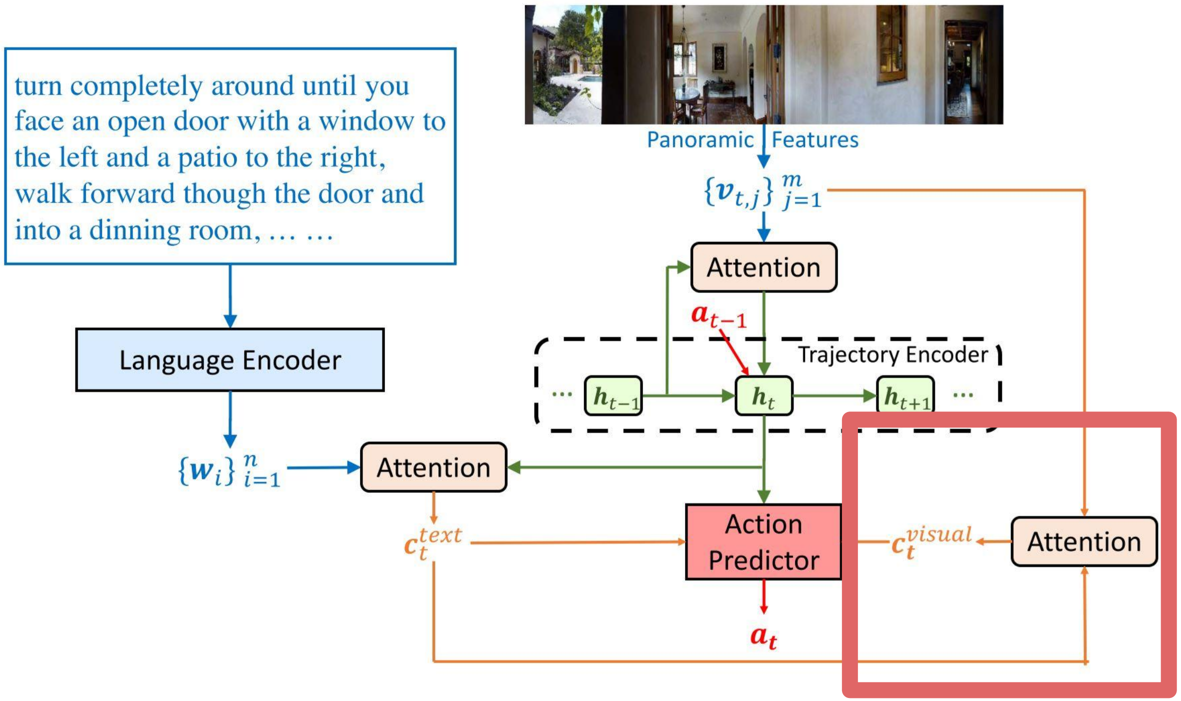
Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

.

Cross modal reasoning



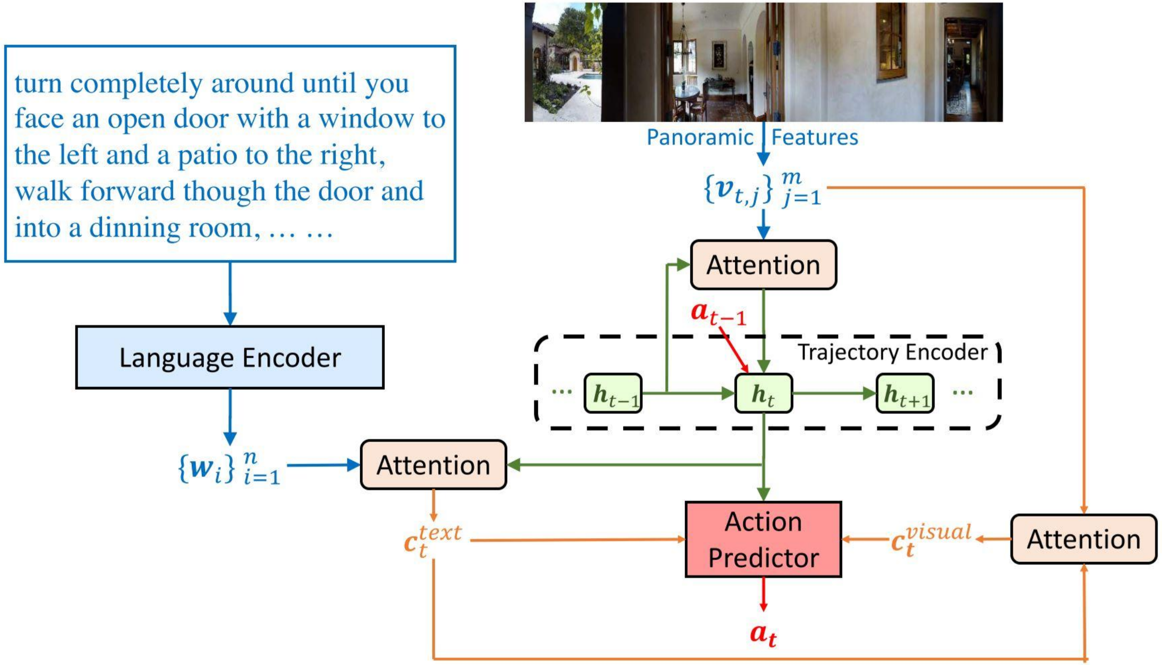
Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

.

Cross modal reasoning



Learns history based on previous action and visual input

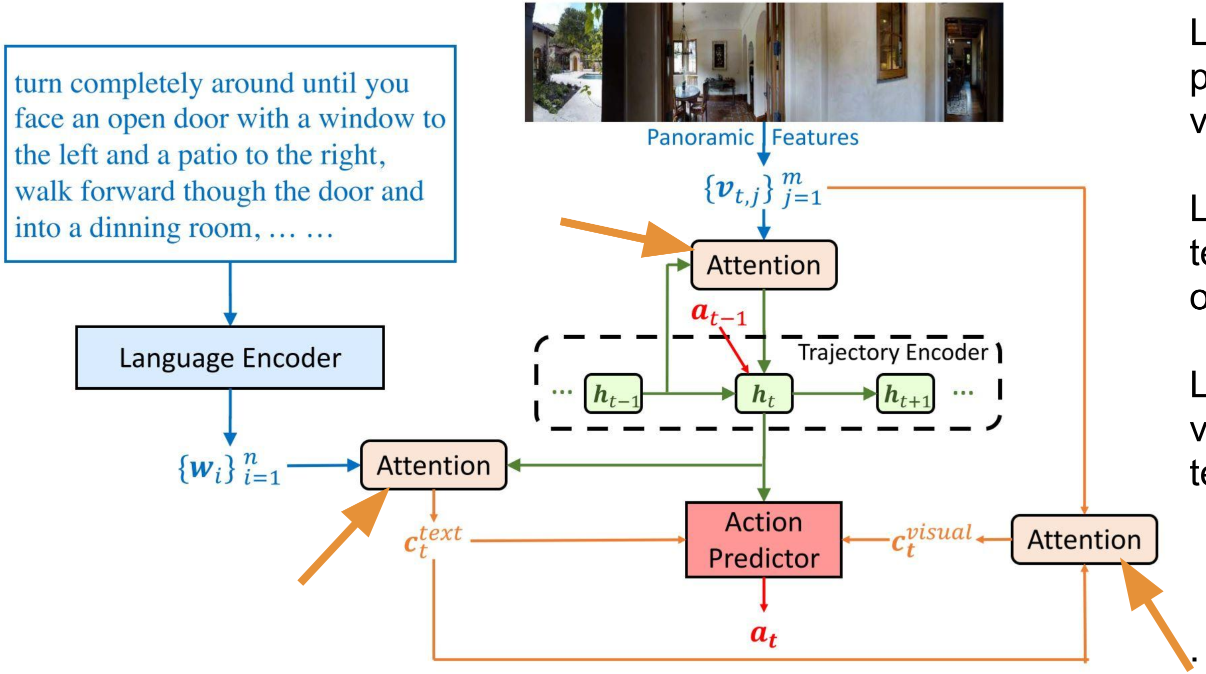
Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

.

Text ⇄ Image

Cross modal reasoning



Learns history based on previous action and visual input

Learns how to focus on textual instruction based on history

Learns how to focus on visual input based on textual instruction

Attention (generic)

Depending on the state, which input should be focused on (weighted more) ?

State at time t

Set of inputs

$$\hat{x} = attention(s_t, \{x_i\}_{i=1}^{n}) = \sum_{i=1}^{n} softmax(s_t W_s (x_i W_x)^T) x_i$$

Attention (generic)

Depending on the state, which input should be focused on (weighted more) ?

State at time t

Set of inputs

$$\hat{x} = attention(s_t, \{x_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \underbrace{softmax(s_t W_s (x_i W_x)^T)}_{\text{Weight of xi}} x_i$$

Attention (generic)

Depending on the state, which input should be focused on (weighted more) ?

State at time t

Set of inputs

$$\hat{x} = attention(s_t, \{x_i\}_{i=1}^{n}) =$$

$$\sum_{i=1}^{n} \underbrace{softmax(s_t W_s (x_i W_x)^T)}_{\text{Weight of xi}} x_i$$

How state contributes

Attention (generic)

Depending on the state, which input should be focused on (weighted more) ?

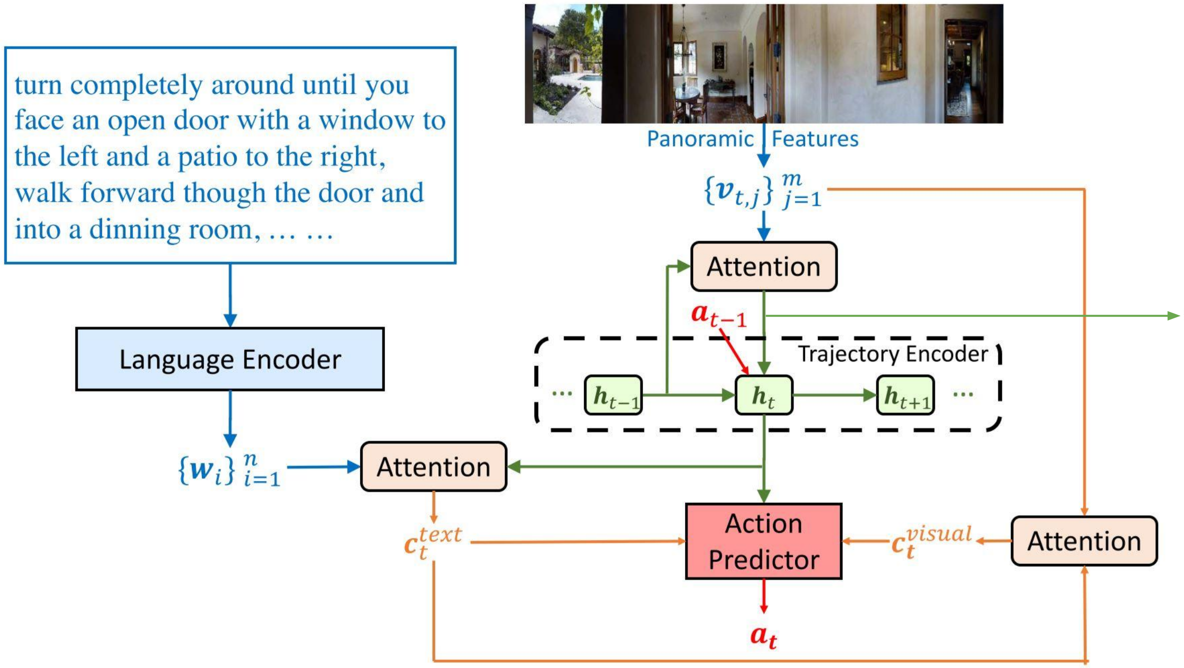
State at time t

Set of inputs

$$\hat{x} = attention(s_t, \{x_i\}_{i=1}^n) = \sum_{i=1}^{n} \underbrace{softmax(s_t W_s (x_i W_x)^T)}_{\text{Weight of xi}} x_i$$

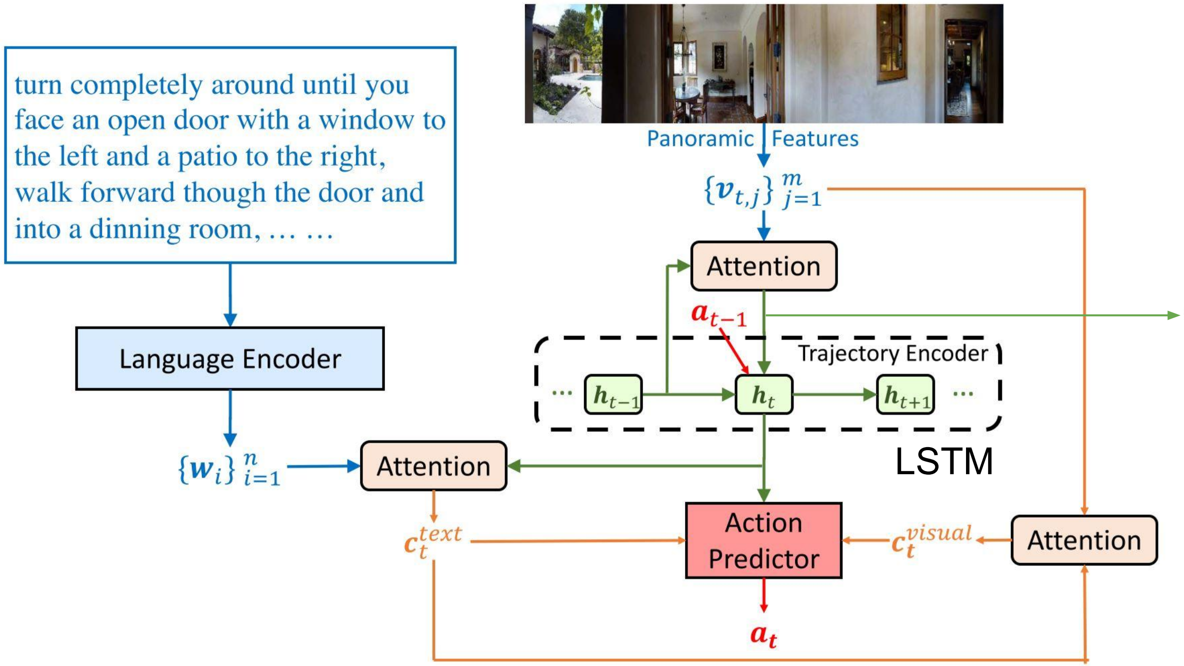
How input contributes

History Context



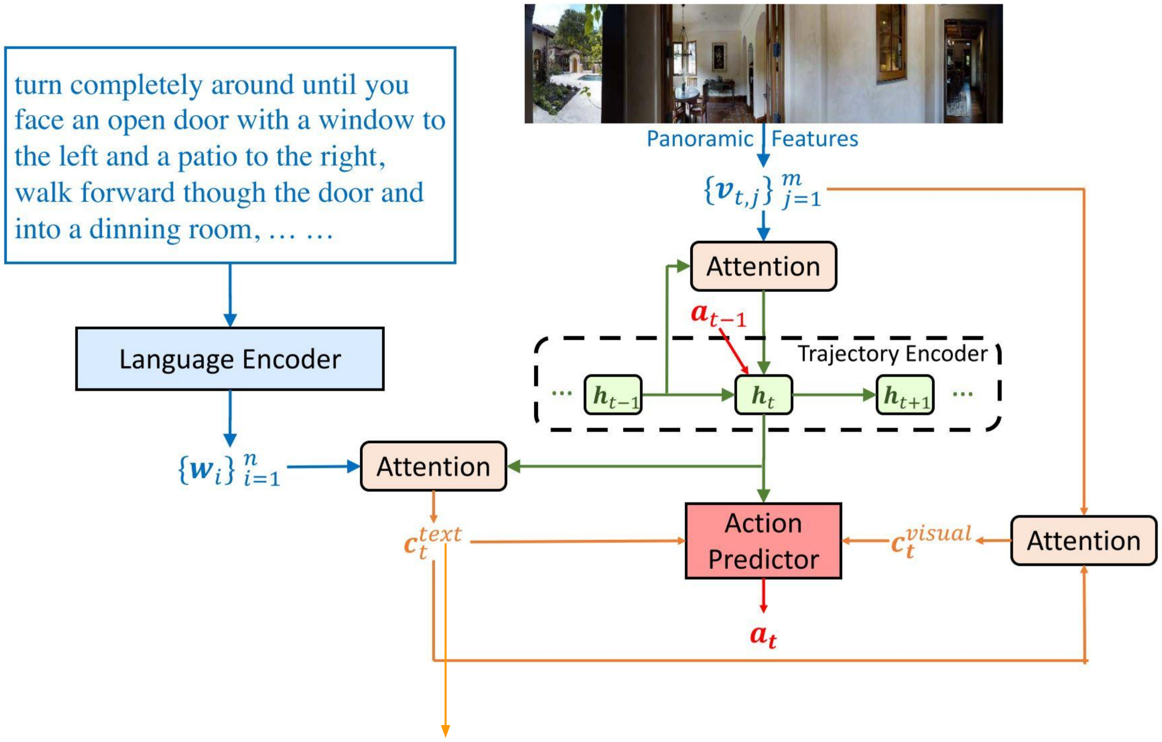
$$v_t = attention(h_{t-1}, \{v_{t,j}\}_{j=1}^{m})$$
$$= \sum_j softmax(h_{t-1}W_h(v_{t,j}W_v)^T)v_{t,j}$$

History Context



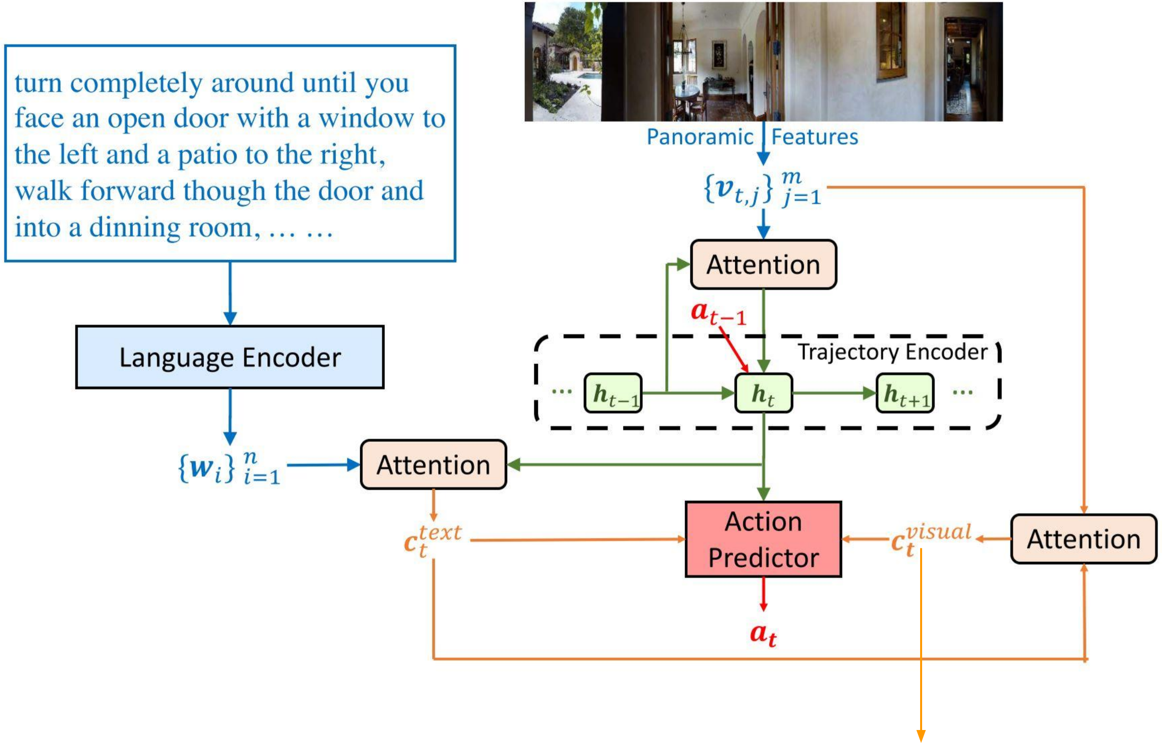
$$v_t = attention(h_{t-1}, \{v_{t,j}\}_{j=1}^{m})$$
$$= \sum_j softmax(h_{t-1} W_h (v_{t,j} W_v)^T) v_{t,j}$$

Visually conditioned textual context



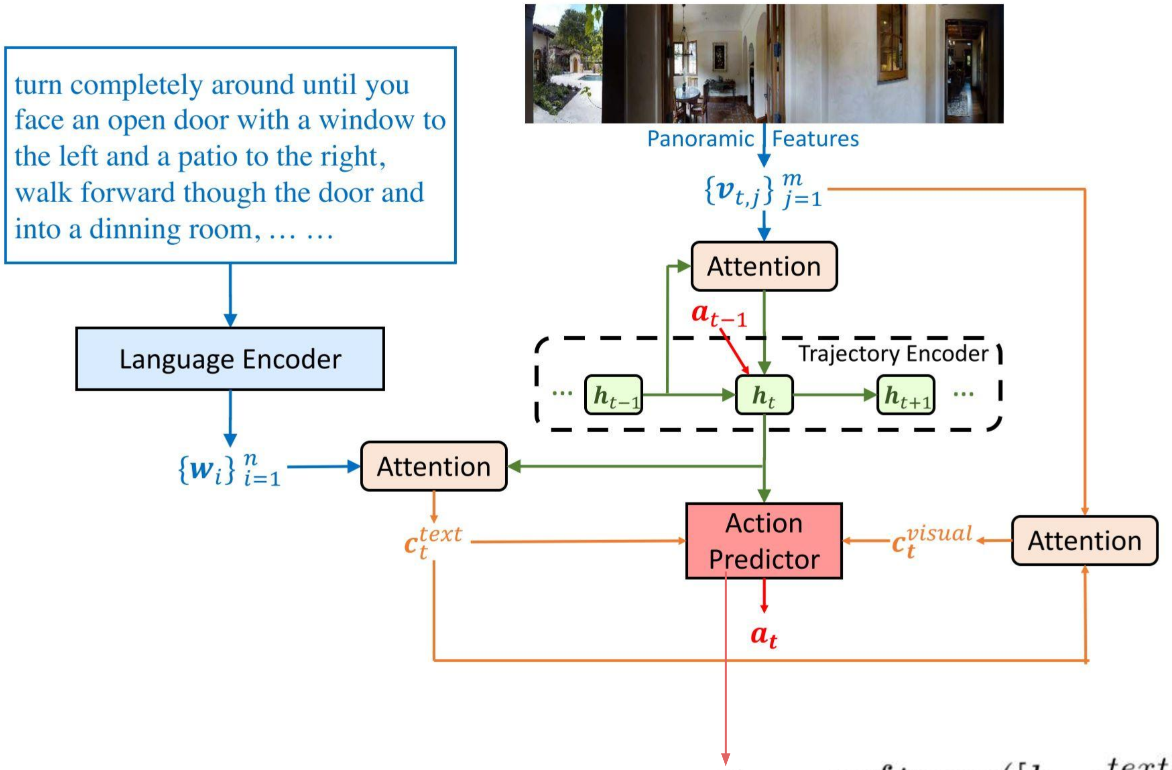
$$c_t^{text} = attention(h_t, \{w_i\}_{i=1}^{n})$$

Textually conditioned visual context



$$c_t^{visual} = attention(c_t^{text}, \{v_j\}_{j=1}^{m})$$

Action prediction



$$p_k = softmax([h_t, c_t^{text}, c_t^{visual}]W_c(u_k W_u)^T$$

Action prediction

image patch from
one viewpoint

heading

elevation

$$[v_i; sin\psi; cos\psi; sin\omega; cos\omega]$$

Each viewpoint corresponds to a
next-step reachable location

action vector

$$p_k = softmax([h_t, c_t^{text}, c_t^{visual}]W_c(u_k W_u)^T$$

Action prediction

image patch from
one viewpoint

heading

elevation

$$[v_i; sin\psi; cos\psi; sin\omega; cos\omega]$$

action vector

$$p_k = softmax(\underbrace{[h_t, c_t^{text}, c_t^{visual}]}_{\text{Context vector}} W_c(u_k W_u)^T$$

Action prediction

image patch from
one viewpoint

heading

elevation

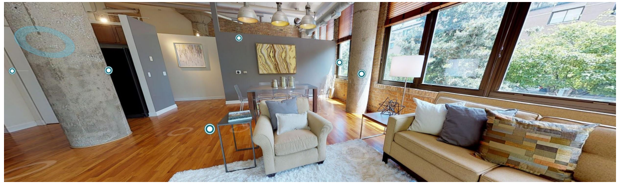
$$[v_i; sin\psi; cos\psi; sin\omega; cos\omega]$$

action vector

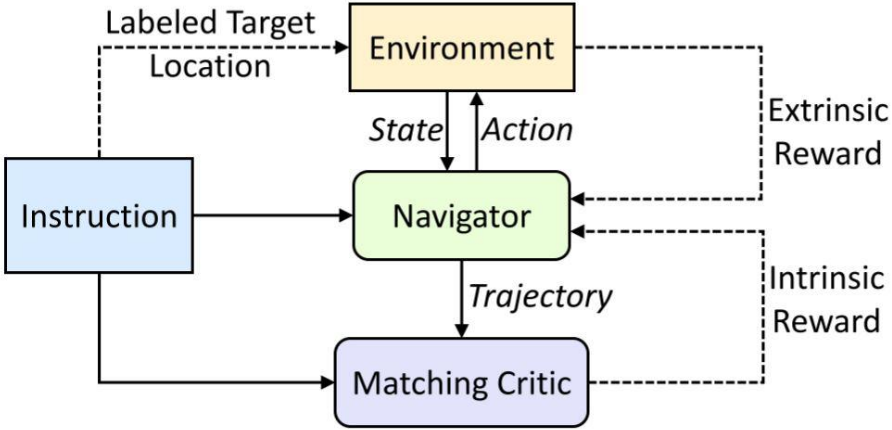
$$p_k = softmax([h_t, c_t^{text}, c_t^{visual}]W_c(u_kW_u)^T$$

Context vector

Trainable weights

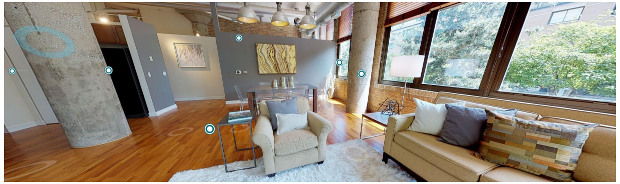


Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front
of the toilet.

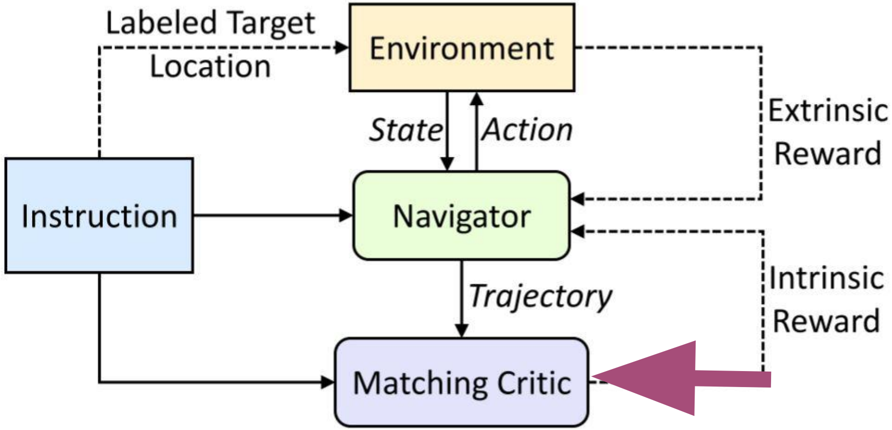


Close to the target?

Follows the instruction?



Turn right and head towards the kitchen. Then turn left, pass a table and enter the hallway. ...Stop in front
of the toilet.



Close to the target?

Follows the instruction?

Matching Critic (intrinsic reward)

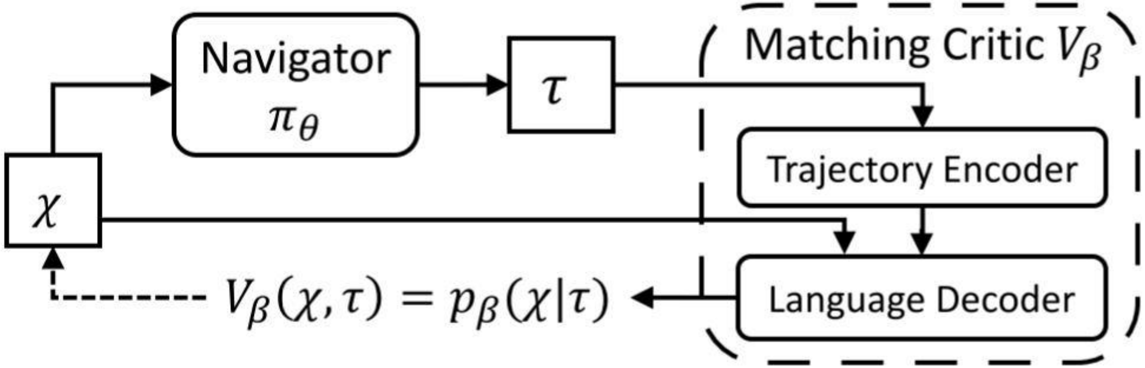
Enforces the agent to follow the instruction by rewarding trajectories that match

the instruction and to determine matching

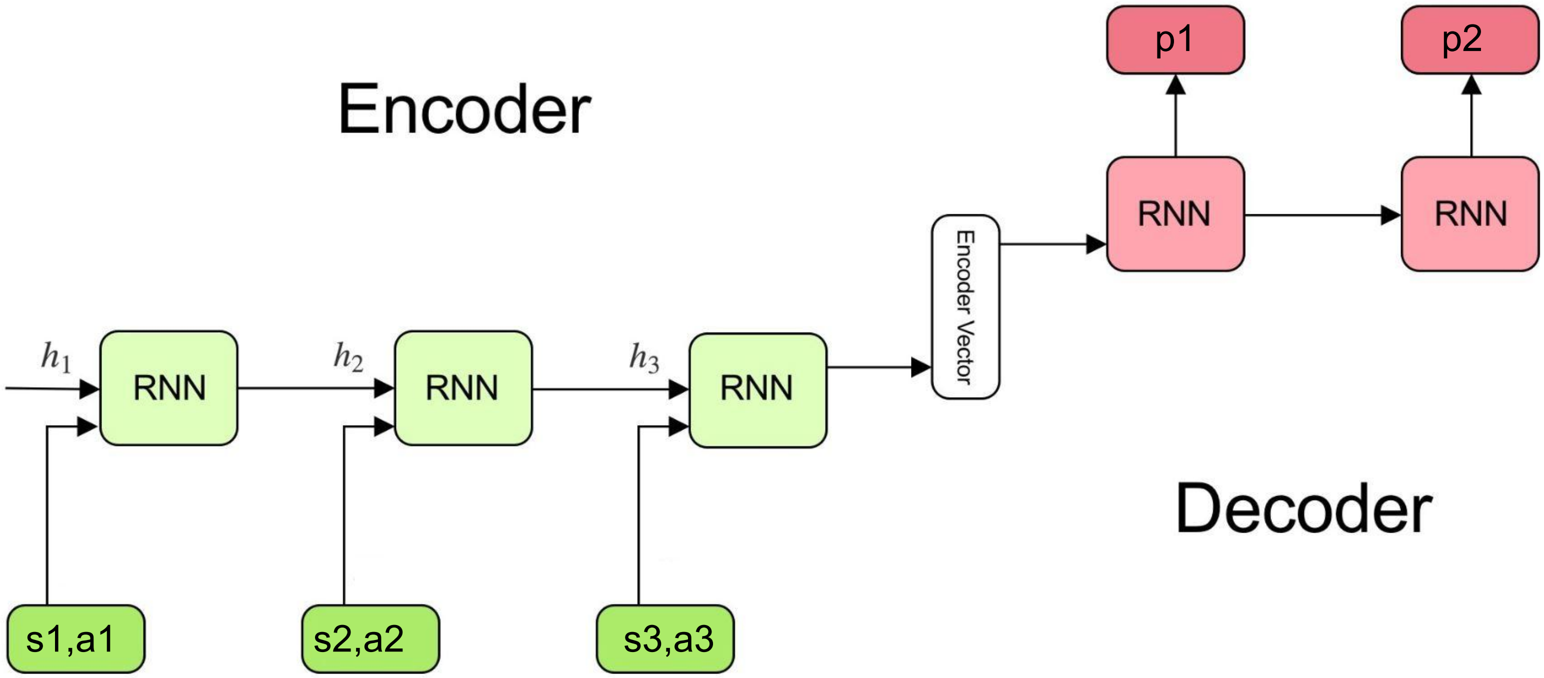
Matching Critic (intrinsic reward)

Try to reconstruct the instruction from the trajectory

and see how it matches the original one

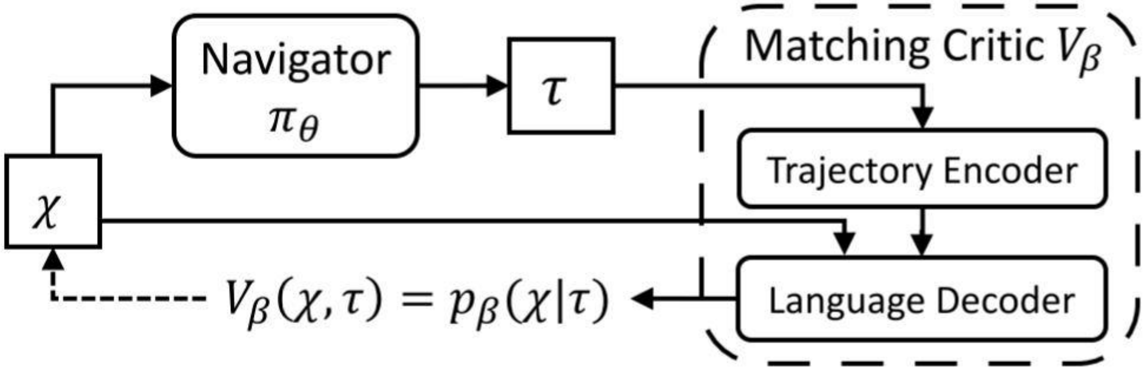


Sequence to sequence



Matching Critic (intrinsic reward)

- Try to reconstruct the instruction from the trajectory and see how it matches the original one



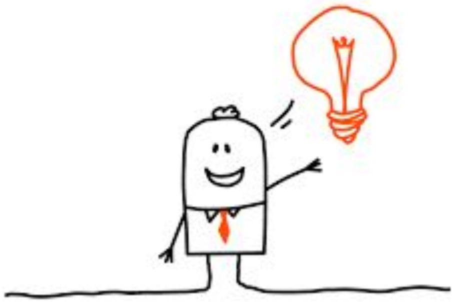
Learning

- Goal: learn the policy **π: S -> A**
- Approximate the policy first!
    - Behavior cloning
    - Supervised Learning
    - Use state-action data from seen environments
    - $\boldsymbol{\alpha^*_t}$ - demonstration action

$$L_{sl} = -\mathbb{E}[\log(\pi_\theta(a_t^*|s_t))]$$

Learning

- **Problem:** limited generalizability
  - Unpredictable behavior in unseen states
- **Solution:** switch over to Reinforcement Learning!



Learning - Extrinsic Reward

- Two metrics:
  - **Immediate reward:**
    - **Distance** from the current state $\mathbf{s_t}$ to the target state $\mathbf{s_{target}}$: $\mathcal{D}_{\mathbf{target}}(\mathbf{s_t})$

$$r(s_t, a_t) = \mathcal{D}_{target}(s_t) - \mathcal{D}_{target}(s_{t+1})$$

  - **Delayed reward:**
    - When the agent reaches its destination:

$$r(s_T, a_T) = \mathbb{1}(\mathcal{D}_{target}(s_T) \leq d)$$

Learning - Extrinsic Reward

$$R_{extr}(s_t, a_t) = \underbrace{r(s_t, a_t)}_{\text{immediate reward}} + \underbrace{\sum_{t'=t+1}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})}_{\text{discounted future reward}}$$

Learning - Extrinsic & Intrinsic Rewards

- Matching critic calculates intrinsic reward $\boldsymbol{R}_{intr}$
- Based on alignment between the trajectory and the original instruction
- Final Reward function:
  - $L_{rl} = -\mathbb{E}_{a_t \sim \pi_\theta}[A_t]$ where $A_t = R_{extr} + \delta R_{intr}$
  - δ is a hyperparameter defining importance of $\boldsymbol{R}_{intr}$
  - Gradient of the loss function: $\nabla_\theta L_{rl} = -A_t \nabla_\theta \log \pi_\theta(a_t|s_t)$

Self-Supervised Imitation Learning

- **Original setting**: train on *seen* environments and test on *unseen* ones
- **New setting:** allow exploration on *unseen* environments as well
  - Lifelong learning & adaptation
  - No ground-truth demonstrations
  - Don’t know where the target location is => **no extrinsic reward**

Self-Supervised Imitation Learning

1. Get an instruction $\mathcal{X}$
2. Navigator produces possible trajectories
3. Matching Critic **V$_\beta$** finds the best trajectory: $\hat{\tau} = \arg\max_{\tau} V_\beta(\mathcal{X}, \tau)$
4. Navigator stores it into a replay buffer
5. The agent uses the saved trajectory $\hat{\tau}$ as a ground-truth
6. Self-supervised learning loss: $L_{sil} = -\mathbb{E}[\log(\pi_\theta(\hat{a}_t | s_t))]$

Experimental setup:

•**R2R dataset**: This dataset covers most of the visual diversity and instructions. It consists of training, seen validation, unseen validation and test sets.

•**Testing scenarios**:

Standard: Train in seen environments and test in unseen environments without prior exploration.

Lifelong learning: The agent is encouraged to explore the environment and learn from the feedback.

Evaluation metrics:

- **PL**: Path length
- **NE**: Navigation error
- **OSR**: Oracle success rate
- **SR**: Success rate
- **SPL**: Success rate weighted by inverse Path Length

Test Set (VLN Challenge Leaderboard)					
Model	PL ↓	NE ↓	OSR ↑	SR ↑	**SPL** ↑
Random	9.89	9.79	18.3	13.2	12
seq2seq [2]	**8.13**	7.85	26.6	20.4	18
RPA [48]	9.15	7.53	32.5	25.3	23
Speaker-Follower [11]	14.82	6.62	44.0	35.0	28
+ beam search	*1257.38*	4.87	96.0	53.5	*1*
Ours					
RCM	15.22	**6.01**	**50.8**	**43.1**	35
RCM + SIL (train)	**11.97**	6.12	49.5	43.0	**38**
RCM + SIL (unseen)[3]	9.48	4.21	66.8	60.5	59

Table 1: Comparison between RCM, SIL and the state-of-the-art

#	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**

Table 2: Ablation study on seen and unseen environments

(a) A successful case

(b) A failure case

Figure 6: Qualitative examples from the unseen validation set.

Strength and weakness:

Strengths:

- Reinforcement learning leads to better generalizability
- Self-Imitation learning facilitates exploring on unseen environments
- Cross modal grounding effectively enhances the model’s ability to capture context information

Strength and weakness:

Weakness:

- Limited in dataset diversity (Only on R2R)
- Model implementation heavily depends on simulator specifics
- Not using depth data

Conclusion:

This paper presents two novel approaches, RCM and SIL, which combine reinforcement learning and self-supervised imitation learning.

The experiment results show that these methods are effective and efficient in both the standard testing scenario and lifelong learning scenario.

SIL help achieve strong generalizability in unseen environments.

Questions