CSCI 566: Deep Learning and Its Applications

Jesse Thomason

Lecture 9: Deep Learning Agents

CSCI 566 Roadmap

Module 1:
Neural Network Basics

Module 2:
Deep Learning Applications

Module 3:
Advanced Topics in Deep Learning

January 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

February 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

March 2023

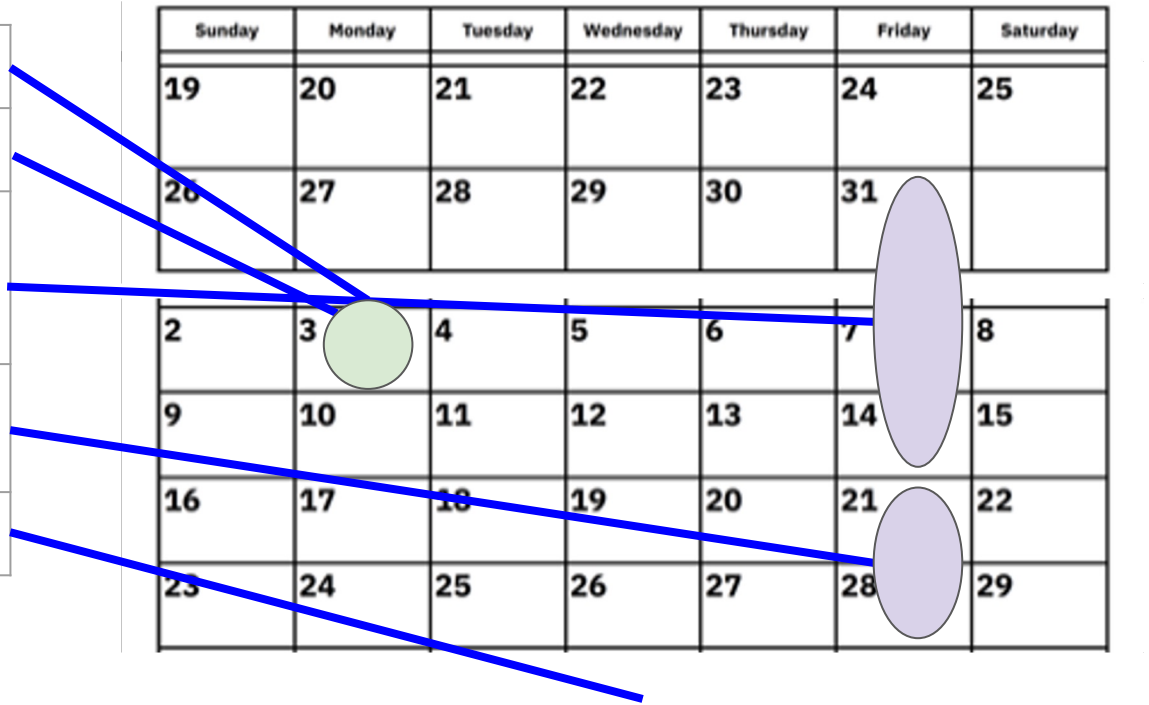
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

April 2023

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

[Mar 31] Endgame Deliverables

Assignment 2 Due	*April 3*
Project Midterm Report	*April 3*
Paper Roleplaying Breakout Sessions	Mar 31, April 7, April 14
Final Project Presentations	April 21, April 28
Project Final Report	May 3



Project Midterm Reports Due *Apr 3, 11:59pm*

- Details about the set of initial experiments you have done for the project
  - Can involve a short abstract/introduction
- Should involve reproducing the results of a state-of-the-art baseline model for the task of interest with code that you have implemented, or a pilot version of your proposed approach
- Talk about the specific dataset(s) you choose to use, the evaluation protocol and metric you decide to have, and the experiment settings.
- Perform an analysis of what remaining errors your mid-report model makes and describe how you will approach these by the final report
- Format: a 3-page report + contributions page
- Syllabus provides suggested format outline and grading breakdown

Project Final Presentations: April 21, April 28

- Schedule is available now:
- Your group will submit to us a multiple choice question about your final presentation, which we'll use to build quizzes that summarize each day's presentations.
    - Project *questions* are due one week before your scheduled prez date.

April 21	April 28
The Gradient Gangsters	Venture
Ex Machina	DeepTrojans
Info Knights	x-people
Data Bee	Average and Savage
Logical Node	One Eye DingZhen
KISS MY AI	LEAP
Team Discount GPT	Data Wizards
Neural Nexus - Deqing TA	DeepFreaks
adam	The Parameters
Shallow Minds	Bellman
QL - Quartet	Neural Navigators
Team GXYZ	Team Sigmoid
Don't be k-Mean	4D Tensor
Ambiguous Dilemma	Alchemists
Outliers	Rural Notice aka J.A.R.V.I.S.
Visual Vanguards	BrainStorm Squad
Hidden Layer Cake	metateam
CV Pathfinder	Generators
The Humpback Anglerfish	VisionFormers
DODO Bird	Neural Nexus - Tejas TA
too lazy to name	Convoluted Thinkers
Unstable Diffusion	SAIK
CatGPT	phi-neurons
The Team	Trojan Transformers
IUYJNTBHGRVFECD	Deep House
Pixlr	GPUs

Overview of Today's Plan

- ~~Course organization and deliverables~~
  - Any questions before we move on?
- Lecture 8 Recap
- Deep Learning for Agents
- March 31st Project Roleplaying Breakouts

Images and Text for Retrieval

Text Query

"A tropical bird perches in the jungle."

Candidate Images






Images and Text for Retrieval

Image Query

Candidate Captions

“A rabbit sits in the palm of a hand.”

“Men are talking on a basketball court.”

“A tropical bird perches in the jungle.”

“Children play soccer in a field.”

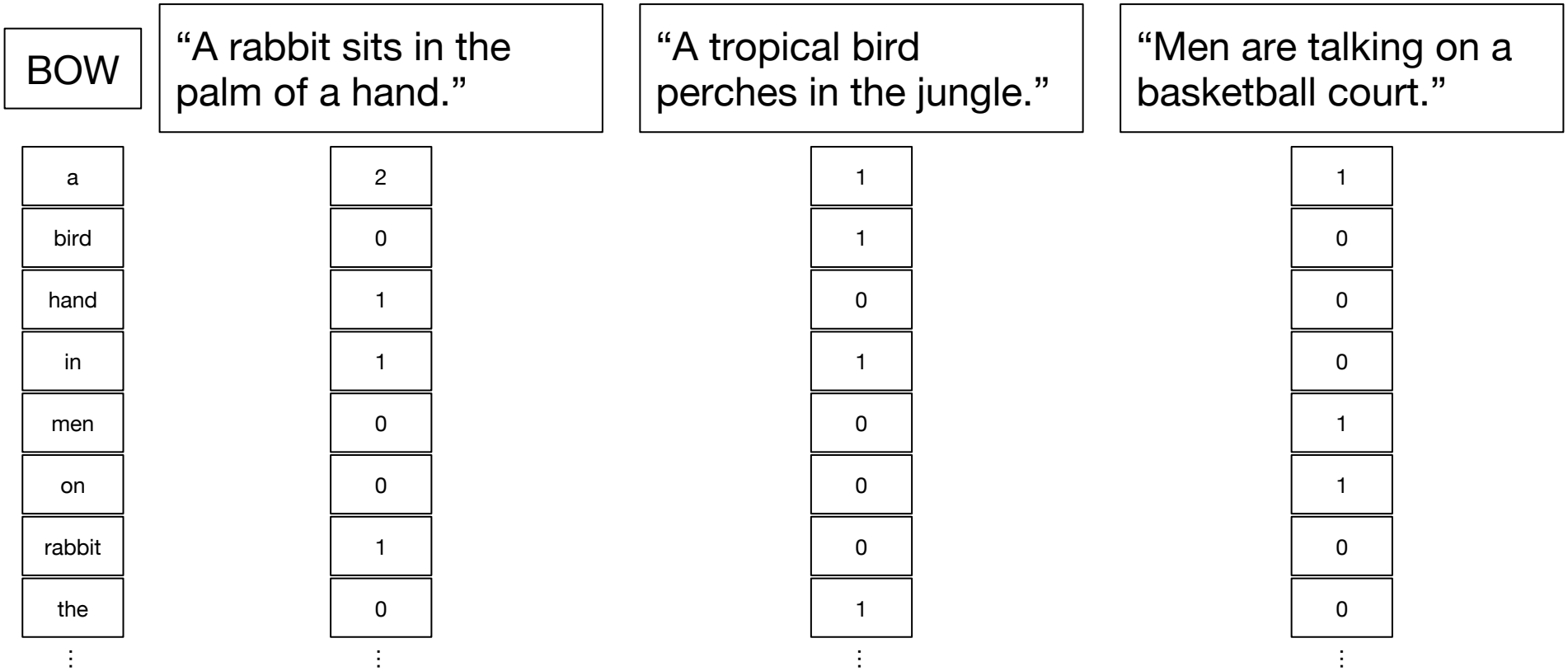
“A white fox is looking at the camera.”

“Sports equipment is staged for a photo.”

Formulating the Retrieval Problem

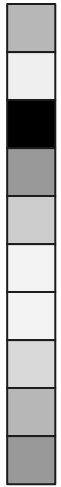
Images	$\mathcal{I}$
Captions	$\mathcal{L}$
Scoring Function	$F : \mathcal{I} \times \mathcal{L} \to \mathbb{R}$

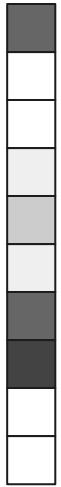
Feature Extraction for Language



Feature Extraction for Vision

RGB
kNN bins





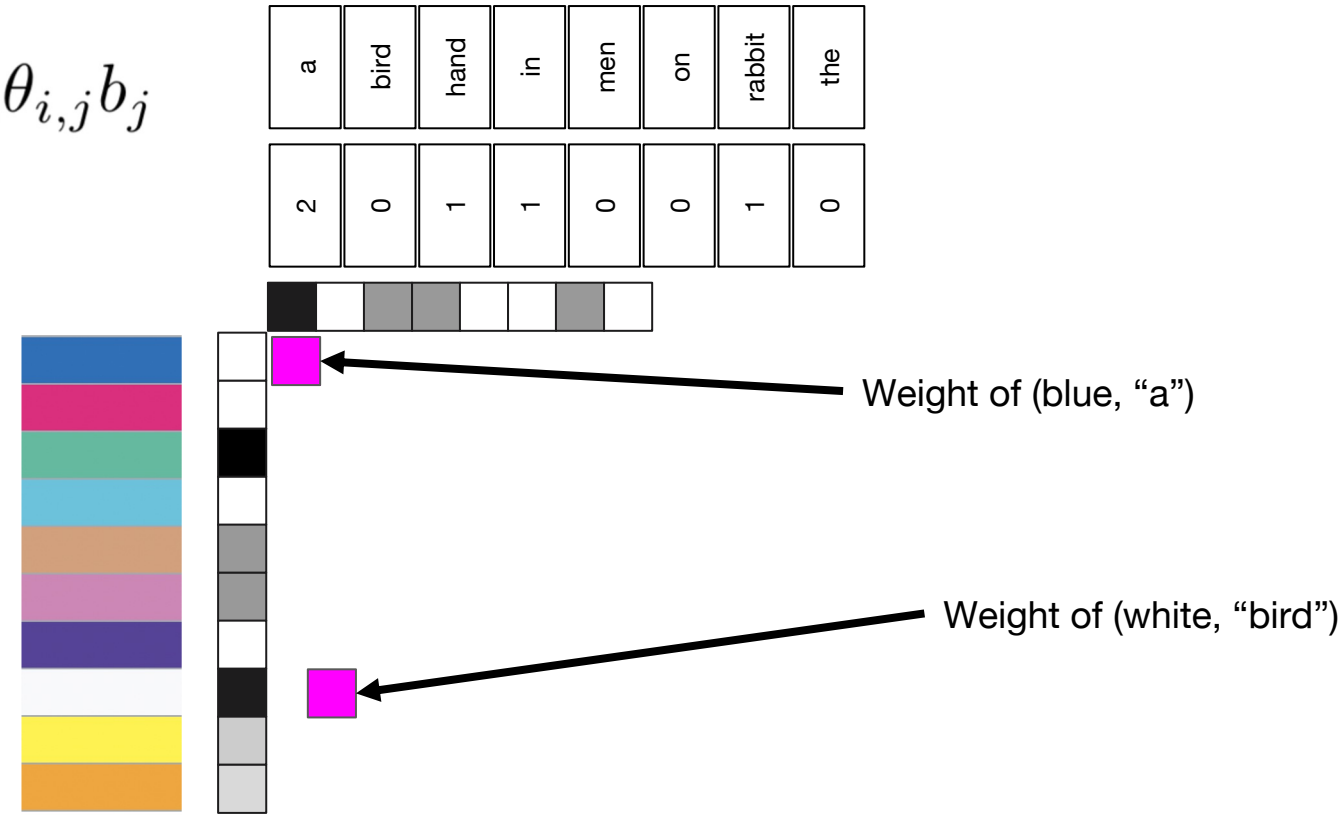
Contextual Information

You shall know a word by the company it keeps (Firth, J. R. 1957:11)

You shall know the features of modality A by the company they keep in the features of modality B. And vice versa.

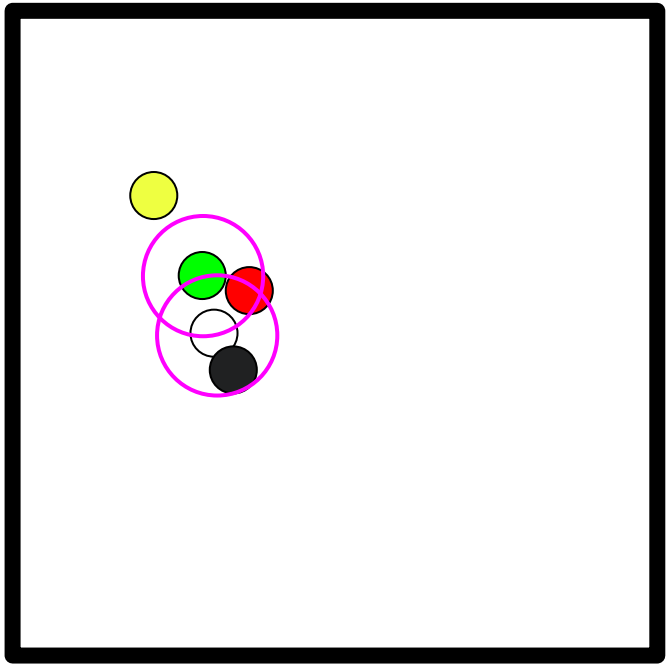
Formulating the Retrieval Problem as a Linear Model

$$\sum_{i=1,j=1}^{n,m} a_i \theta_{i,j} b_j$$



Pretrained Language Token Embeddings

$\omega(\mathcal{L})$



Training

Inference

“A white bunny rabbit held in a green field.”

“A black rabbit watches a red sunset.”

Estimating Joint Distributions

Scoring Function $F_{(\psi,\omega)} : \psi(\mathcal{I}) \times \omega(\mathcal{L}) \rightarrow \mathbb{R}$

- Define F as cosine similarity between image and caption embeddings
- Then we can learn:
  - (minimally) a projection network that maps image embeddings and caption embeddings to the same space
  - (bonus) an image feature extractor (psi)
  - (bonus) a language feature extractor (omega)

"A rabbit sits in the palm of a hand."

"Children play soccer in a field."

"A white fox is looking at the camera."

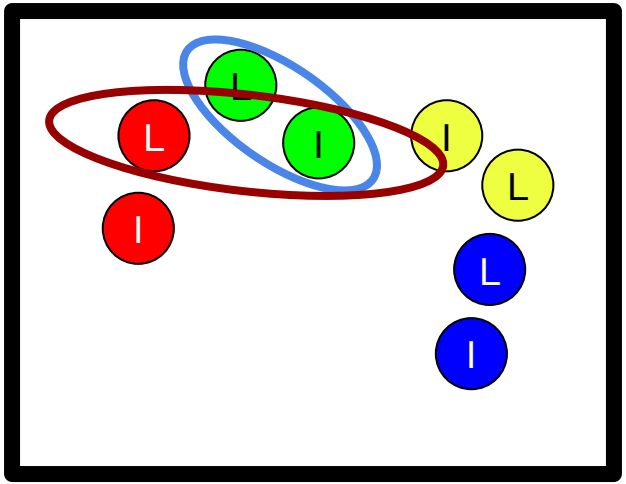
"Men are talking on a basketball court."

Image Embedder $\psi(\mathcal{I})$

Caption Embedder $\omega(\mathcal{L})$

Pull matching imag embeddings together.

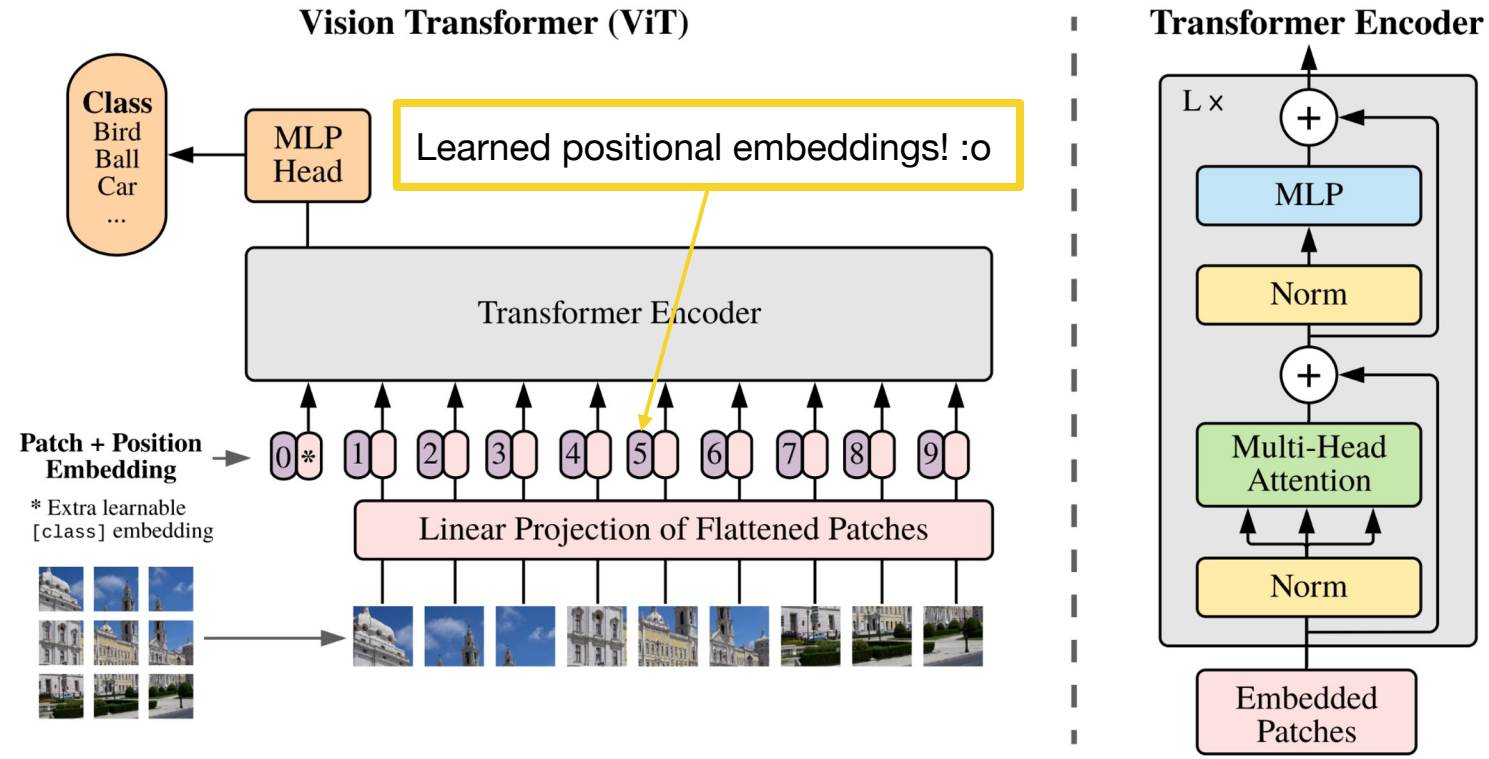
Learned Projection



Push distractor on embeddings apart.

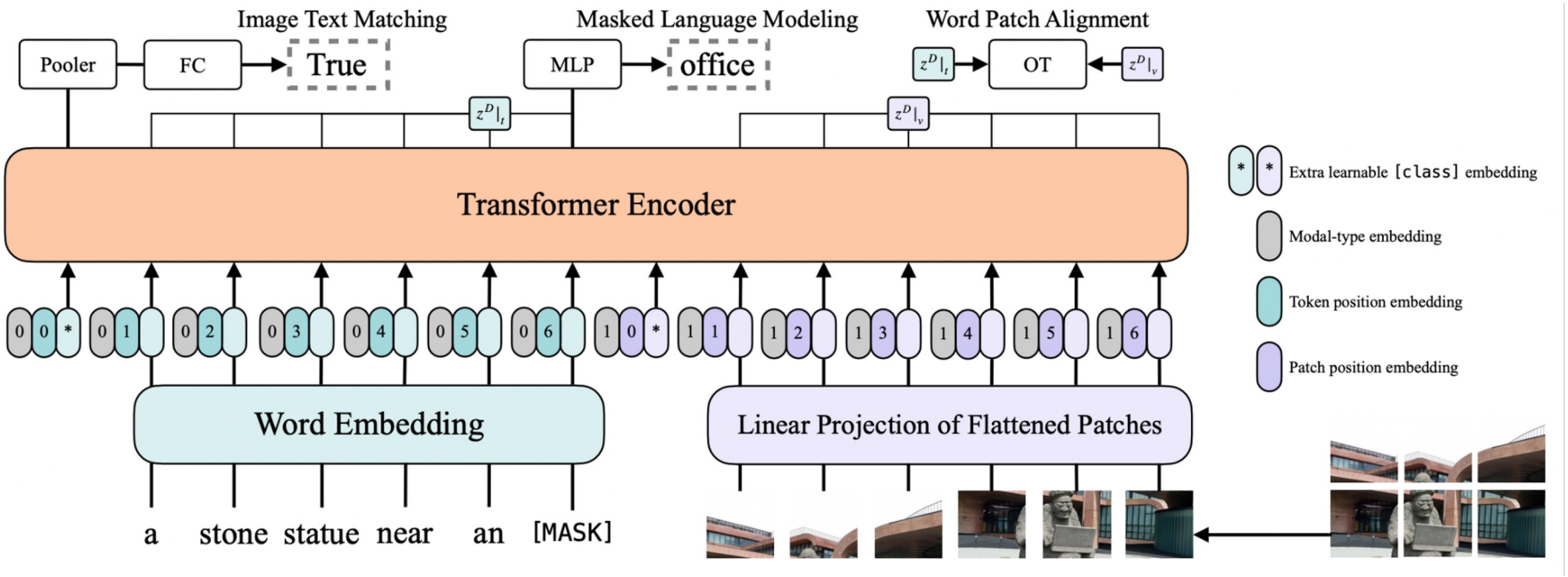
Learned Projection

Transformers for Vision: Patches



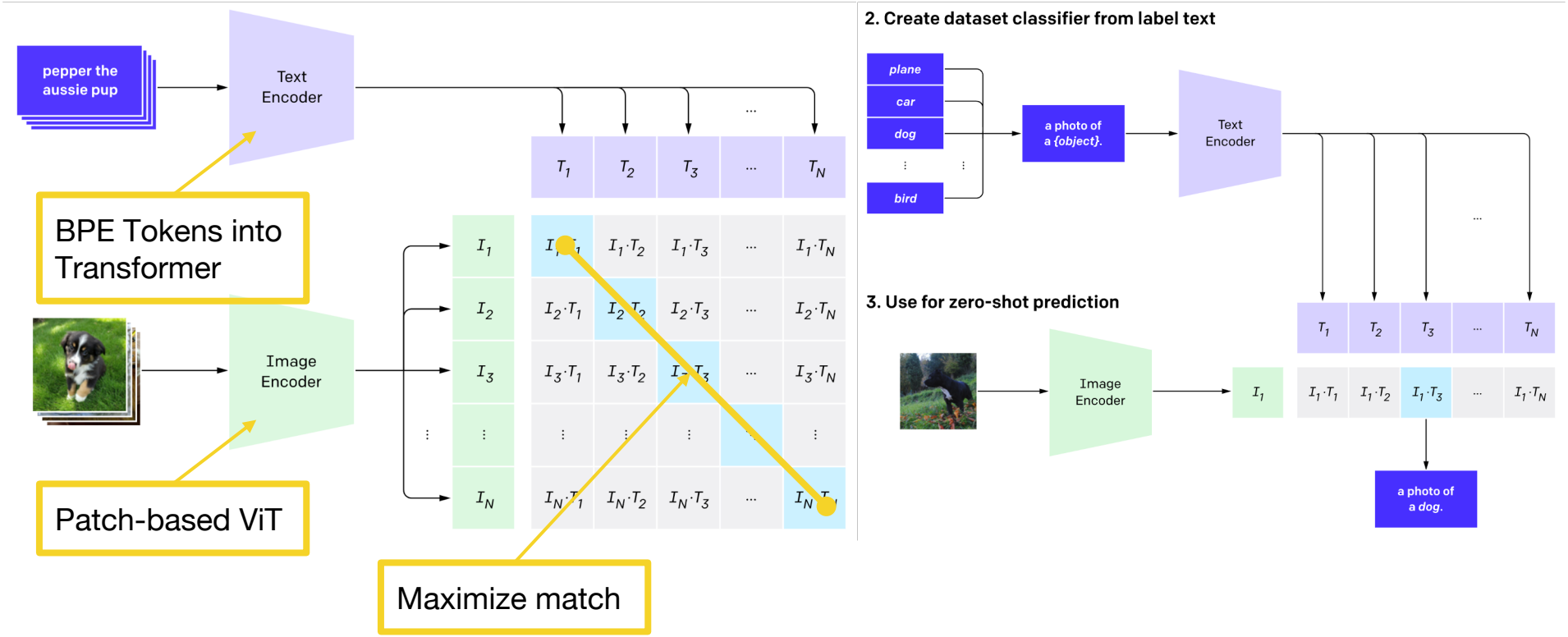
An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

Language and Vision: Patches



ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision

Contrastive Language–Image Pre-training (CLIP)



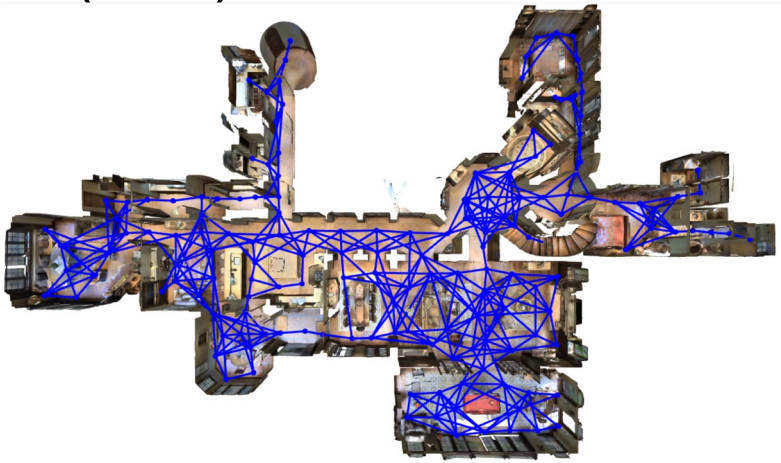
https://openai.com/blog/clip/

Overview of Today's Plan

- ~~Course organization and deliverables~~
- ~~Lecture 8 Recap~~
  - Any questions before we move on?
- Deep Learning for Agents
- March 31st Project Roleplaying Breakouts

Vision-and-Language Navigation (VLN)

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



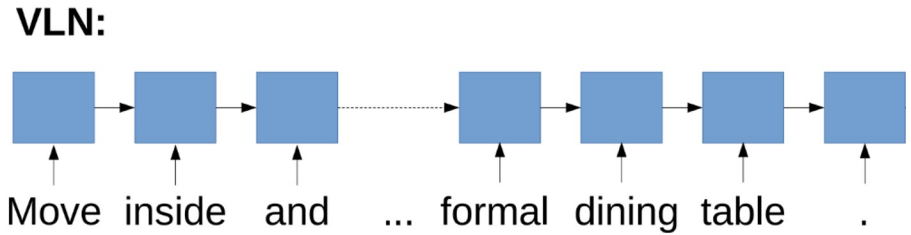
Visual Observations	$\mathcal{I}$
Language Instructions	$\mathcal{L}$
Action Space	$\mathcal{A}$
History	$\mathcal{H} = \{(v_i, l_i), i = 1 \dots T; v_i \in \mathcal{I}, l_i \in \mathcal{L}\}$
Policy	$\pi : \mathcal{I} \times \mathcal{L} \times \mathcal{H} \rightarrow \mathcal{A}$

https://bringmeaspoon.org/

World Actions and Consequences

- Agents perform sequence prediction
- We are no longer taking in input and producing a single output
  - E.g., retrieval of an image from language
- Want to learn a *policy* $\pi(a|s)$ from *states* to *actions*
- We are taking in observations (e.g., language and vision) and producing *one action at a time* that causes *new observations we can consider;* our state is estimated by observations
  - Open-loop: Produce whole sequence at once
  - Closed-loop: Consider observations after each action

Vision-and-Language Navigation with LSTM



Visual Observations	$\mathcal{I}$
Language Instructions	$\mathcal{L}$
Action Space	$\mathcal{A}$
History	$\mathcal{H} = \{(v_i, l_i), i = 1 \dots T; v_i \in \mathcal{I}, l_i \in \mathcal{L}\}$
Policy	$\pi : \mathcal{I} \times \mathcal{L} \times \mathcal{H} \rightarrow \mathcal{A}$

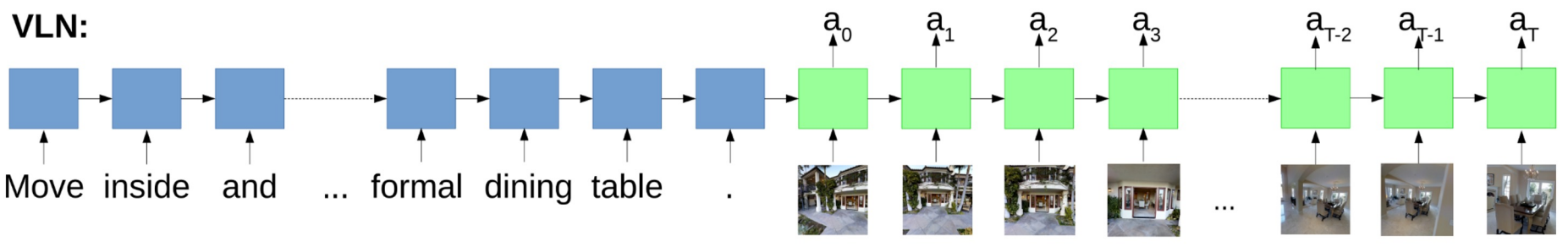
Vision-and-Language Navigation with LSTM

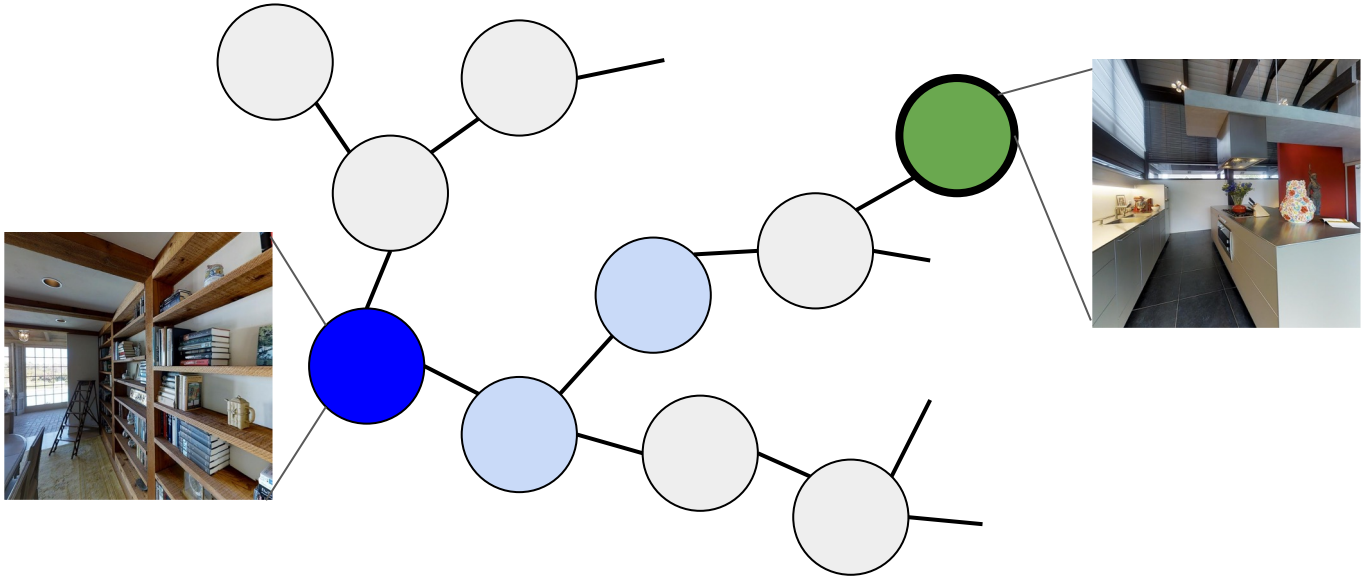
- The resulting LSTM is a *policy* from *states* represented as history so far, image observations, and language instructions to *actions* in the environment
- With training data, this is just *supervised learning* still; in this community you'll also see this called *imitation learning*

Visual Observations	$\mathcal{I}$
Language Instructions	$\mathcal{L}$
Action Space	$\mathcal{A}$

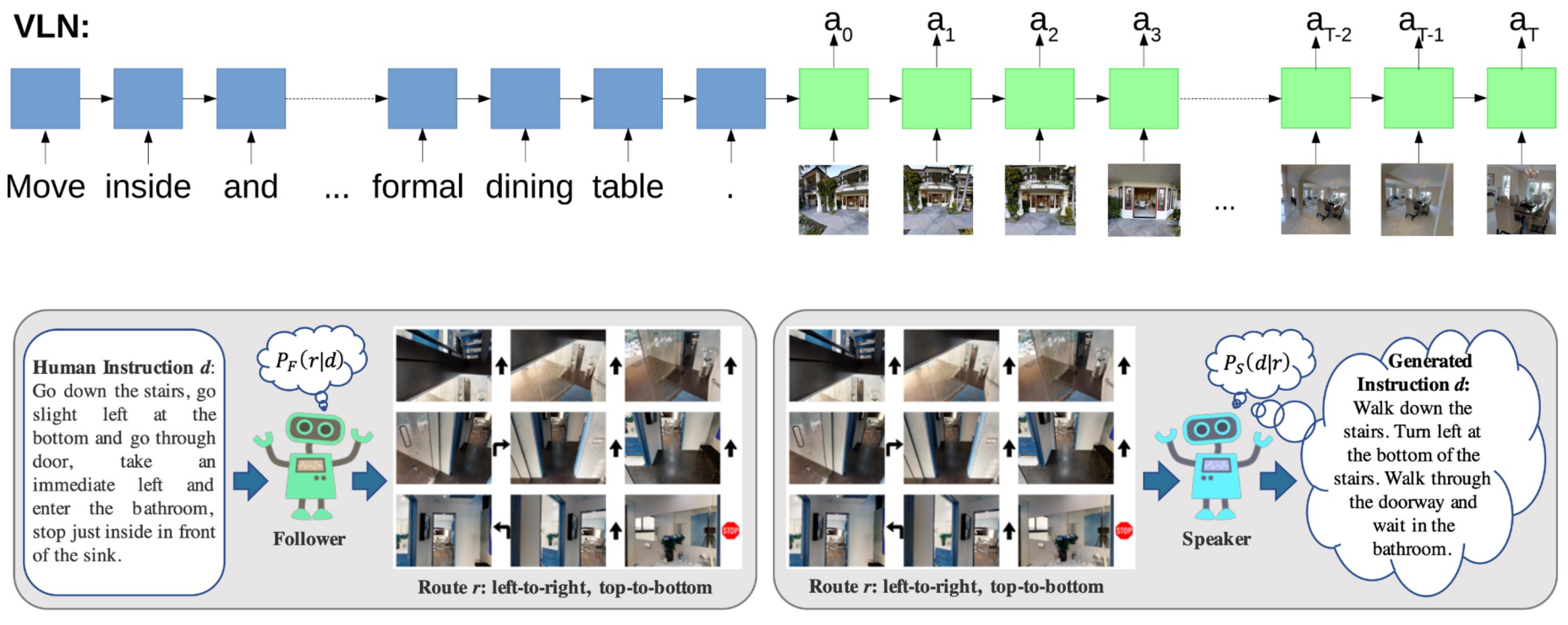
History	$\mathcal{H} = \{(v_i, l_i), i = 1 \dots T; v_i \in \mathcal{I}, l_i \in \mathcal{L}\}$
Policy	$\pi : \mathcal{I} \times \mathcal{L} \times \mathcal{H} \rightarrow \mathcal{A}$

Vision-and-Language Navigation with LSTM





Vision-and-Language Navigation Speaker-Follower



Speaker-Follower Models for Vision-and-Language Navigation

Vision-and-Language Navigation (VLN)

- What is our transition function complexity?
  - Other agents?
  - Do our actions affect the world?
  - Do our actions affect our future observations? (e.g., ourself)
- What is our *state* best represented as?
- Do we know what state we're in at each timestep?
  - No!
  - We have to *estimate* our position with respect to the goal based only on image *observations*

Partially Observable Markov Decision Processes

- A discrete-time POMDP models the relationship between an agent and its environment. Formally, a POMDP is a 7-tuple:

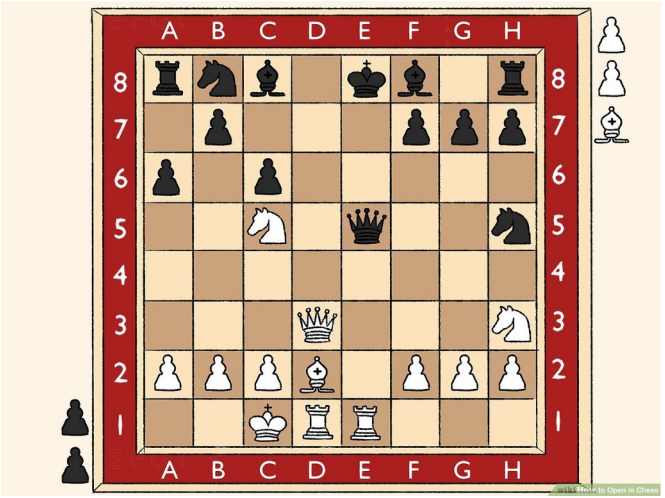
- S is a set of states,
- A is a set of actions,
- T is a set of conditional transition probabilities between states,
- $R : S \times A \to \mathbb{R}$ is the reward function.
- Ω is a set of observations,
- O is a set of conditional observation probabilities, and
- $\gamma \in [0, 1]$ is the discount factor.

- **S**: underlying environment
- **A**: what agent can do
- **T**: how what the agent does changes **S**
- **Om**: what can be observed by the agent

POMDP or MPD?

- MDP is a special case of the more general POMDP formulation.
- **Om == S**, observation is full state.

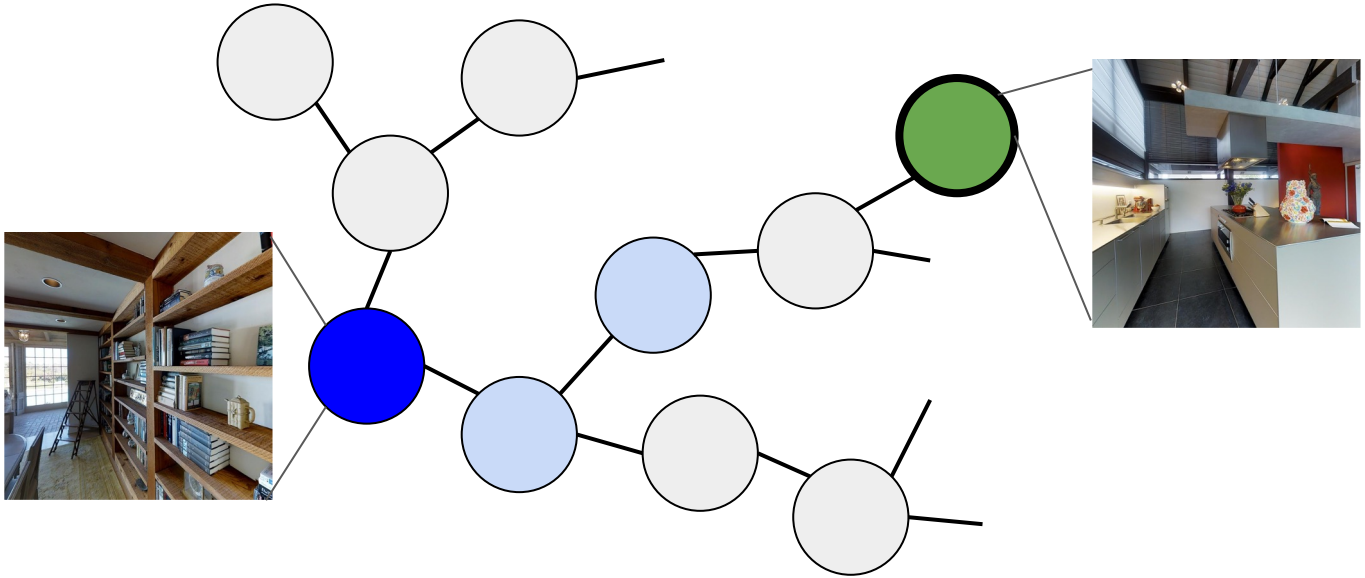
- S is a set of states,
- A is a set of actions,
- T is a set of conditional transition probabilities between states,
- $R : S \times A \to \mathbb{R}$ is the reward function.
- Ω is a set of observations,
- O is a set of conditional observation probabilities, and
- $\gamma \in [0, 1]$ is the discount factor.



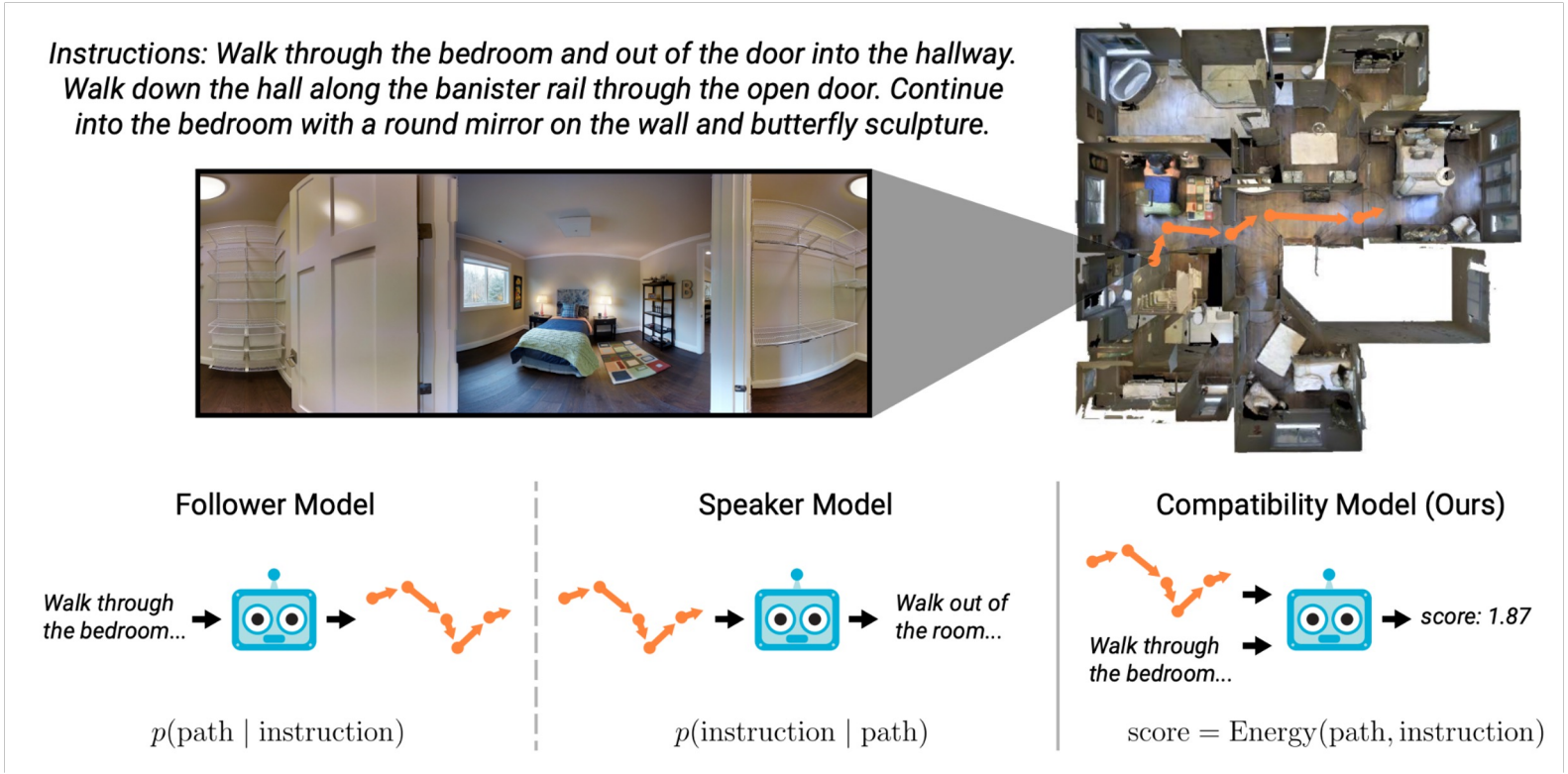
- **S**: underlying environment
- **A**: what agent can do
- **T**: how what the agent does changes **S**
- **Om**: what can be observed by the agent

Tractable, Discrete State Space; Discrete Action Space

- Enumerate nodes + viewpoints; search over graph paths.
- Could create *open-loop* plan since nothing “new” can happen if we’ve mapped everything out.

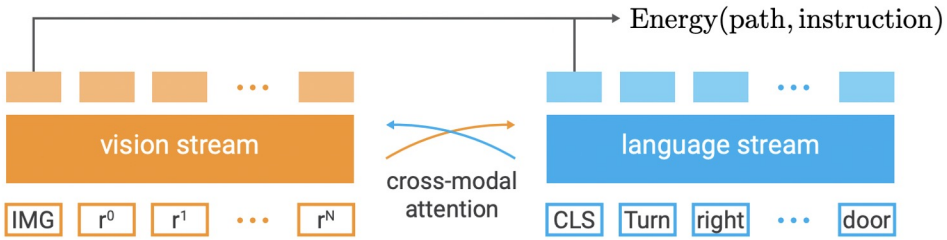


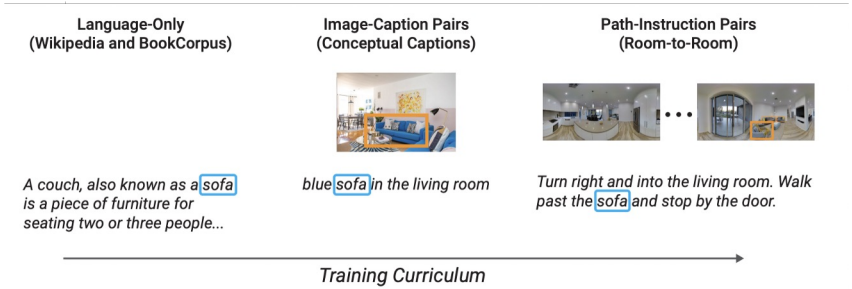
Language and Vision and “Time”

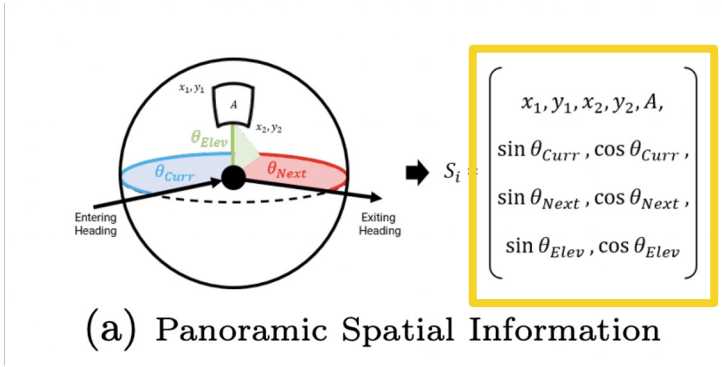


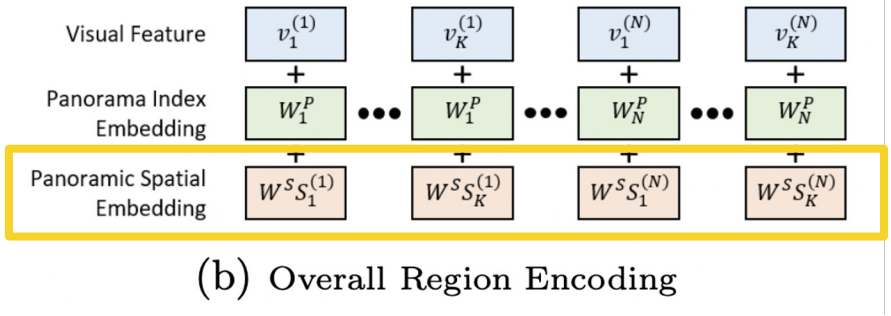
[Improving Vision-and-Language Navigation with Image-Text Pairs from the Web]

Language and Vision and “Time”





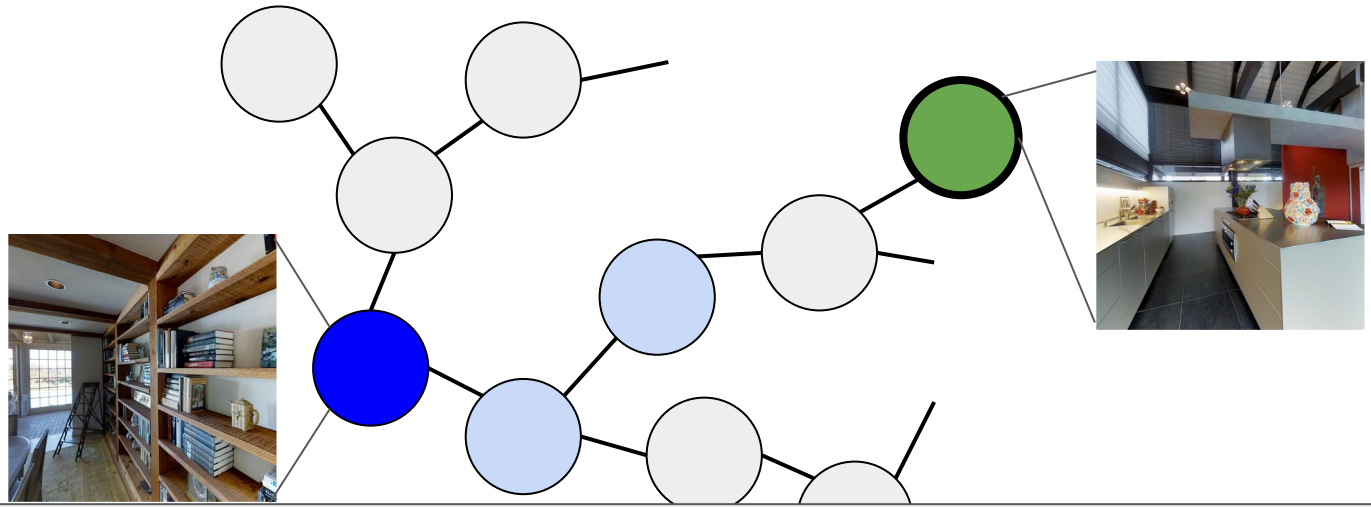




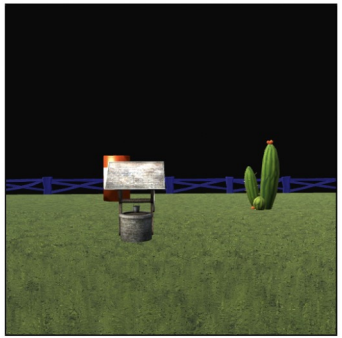
[Improving Vision-and-Language Navigation with Image-Text Pairs from the Web]

Tractable, Discrete State Space; Discrete Action Space

- Use the *follower* model to generate *k* possible paths
- Use the learned *scoring* model to calculate the path with the highest “energy” against the instruction and walk it



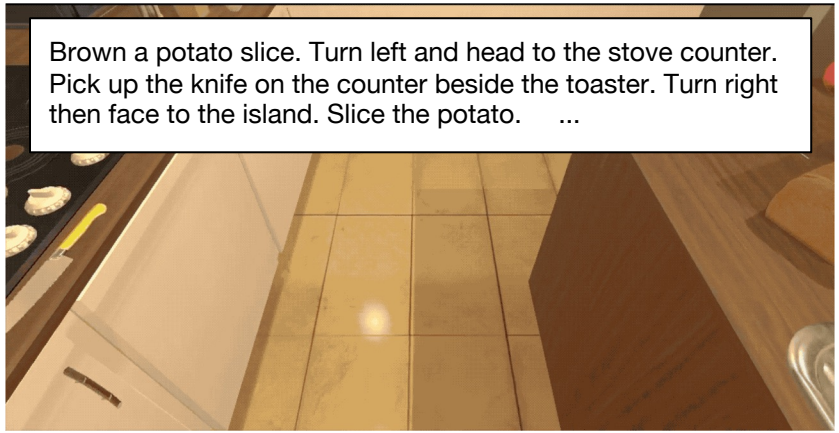
Language-guided Task Completion



After reaching the hydrant head towards the blue fence and pass towards the right side of the well.

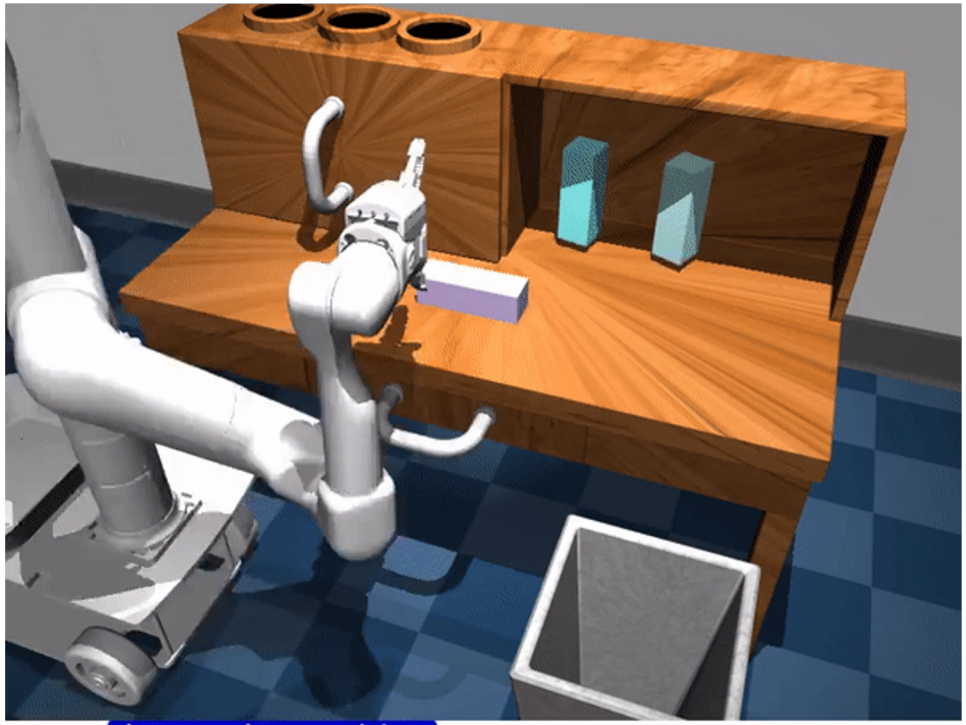
Put the cereal, the sponge, and the dishwashing soap into the cupboard above the sink.

CHALET
[Misr et al., EMNLP 2018]



ALFRED
[Shridhar et al., CVPR 2020]

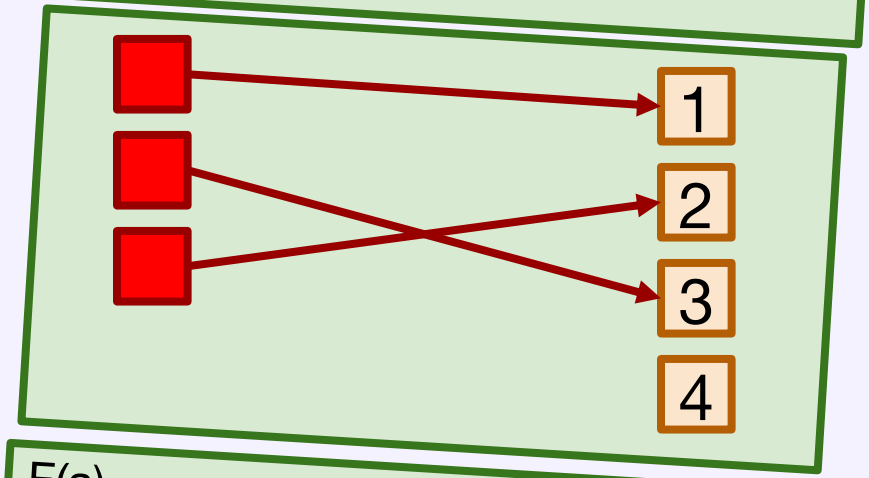
Language-guided Task Completion



https://language-play.github.io/

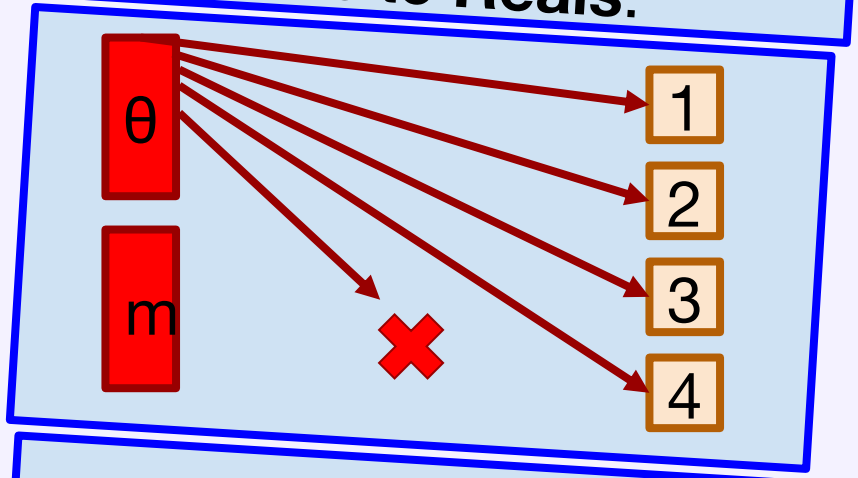
Al te

Discrete *A*: there exists an **injective** map from A to **Natural Numbers**



F(a) =
{1 if a=forward; 2 if a=left;
3 if a=right}

Continuous *A*: there does not exist an **injective** map to **Natural Numbers**, but exists one to **Reals**.

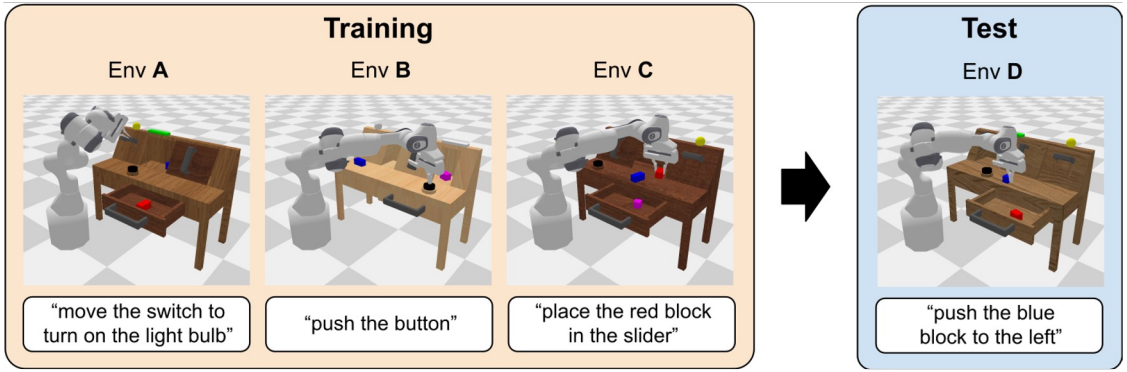


F(a, θ, m):
{θ if a=turn; m+2π if a=forward}

Discrete Actions, Continuous State Spaces

https://github.com/jacobkrantz/VLN-CE

Continuous Action and State Spaces



In CALVIN, the agent must perform closed-loop continuous control to follow unconstrained language instructions characterizing complex robot manipulation tasks, sending continuous actions to the robot at 30hz. In order to give researchers and practitioners the freedom to experiment with different action spaces, CALVIN supports the following actions spaces:

1. **Absolute cartesian pose** - EE position (3), EE orientation in euler angles (3), gripper action (1).
2. **Relative cartesian displacement** - EE position (3), EE orientation in euler angles (3), gripper action (1).
3. **Joint action** - Joint positions (7), gripper action (1).

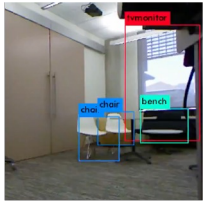
https://github.com/mees/calvin

A Loose Taxonomy for Embodied AI

Sensory Observation

Simulation Render

Simulation Photo



Mounted Camera

Task Specification Observation

"What color is the car?"

"Find a teddy bear on an office chair."

"Boil a potato."

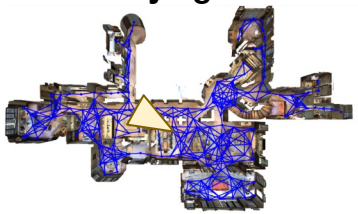
Goal-based / Implicit / High-level

"Turn around and exit the bathroom, walk down the hall and stop at the …"

"First, go to the counter and pick up the shallow pot, then turn around and turn on the sink…"

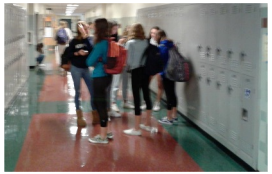
Instruction-based / Explicit / Low-level

Underlying State



Agent position and orientation (state)

Agent state and object states



Multi-agent states, object states, …

Action Space

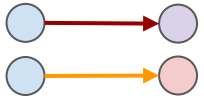
Actions that adjust agent state only.

Actions that affect object states.

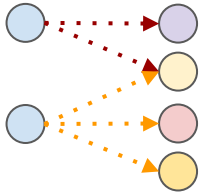
Actions that affect other agent states.

Continuous versus discrete actions.

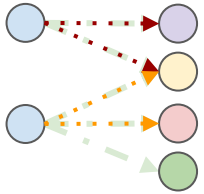
Transition Dynamics



Deterministic

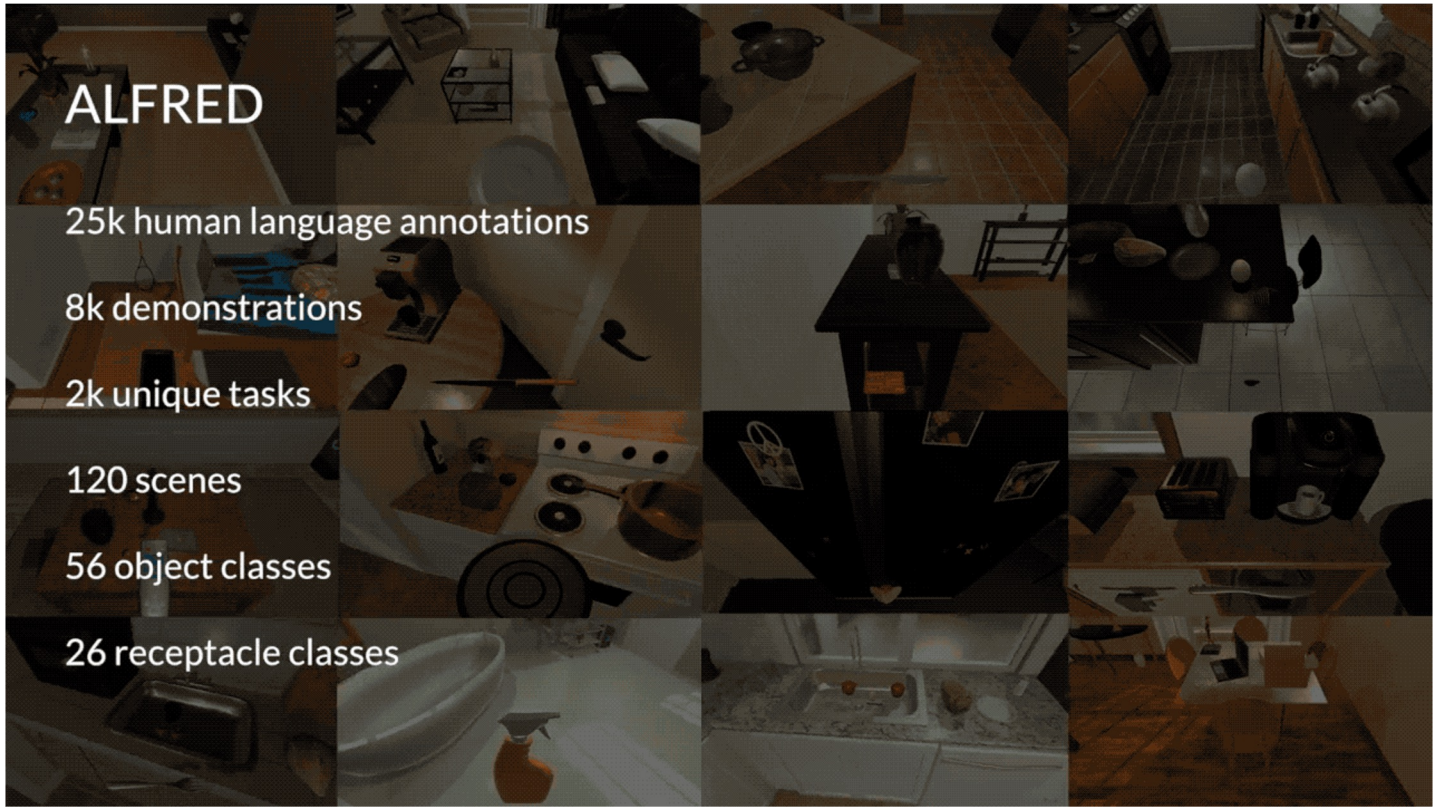


Probabilistic



Environment Dynamics

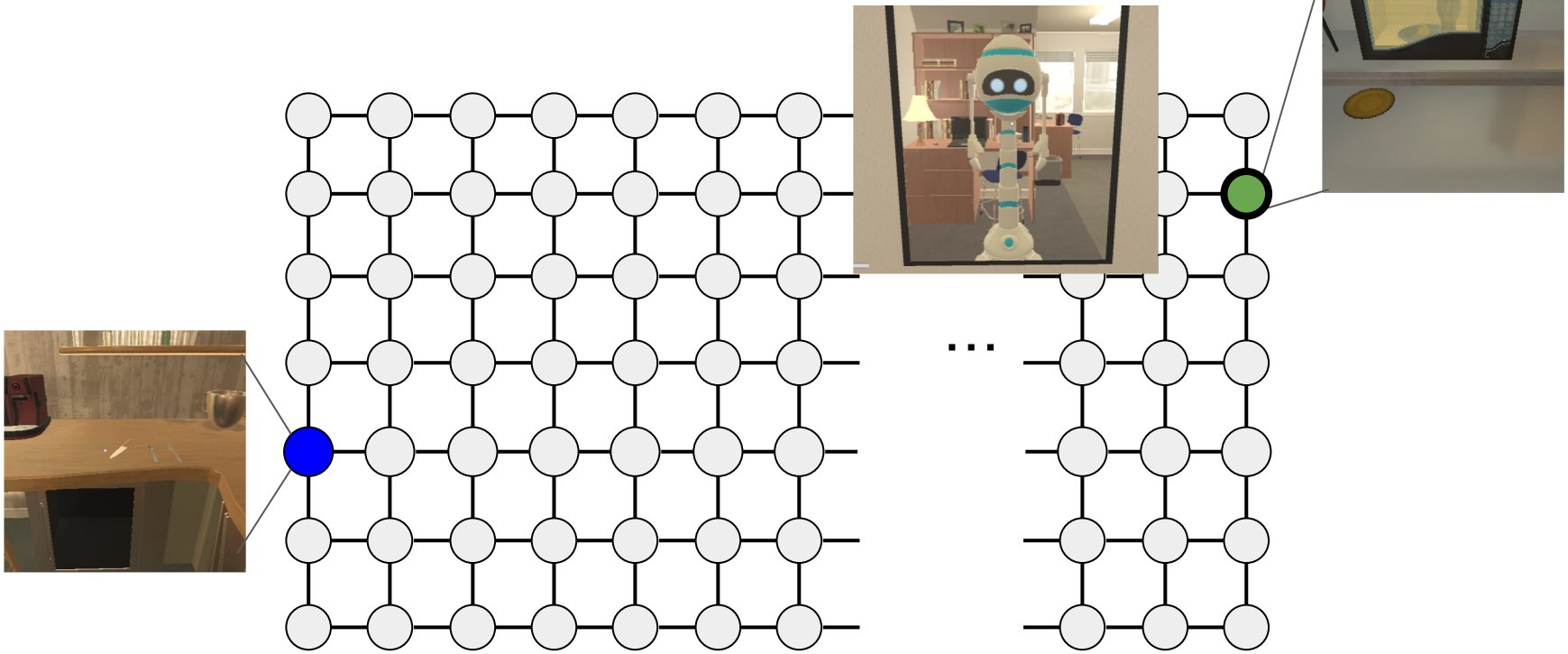
ALFRED Tasks



https://askforalfred.com/

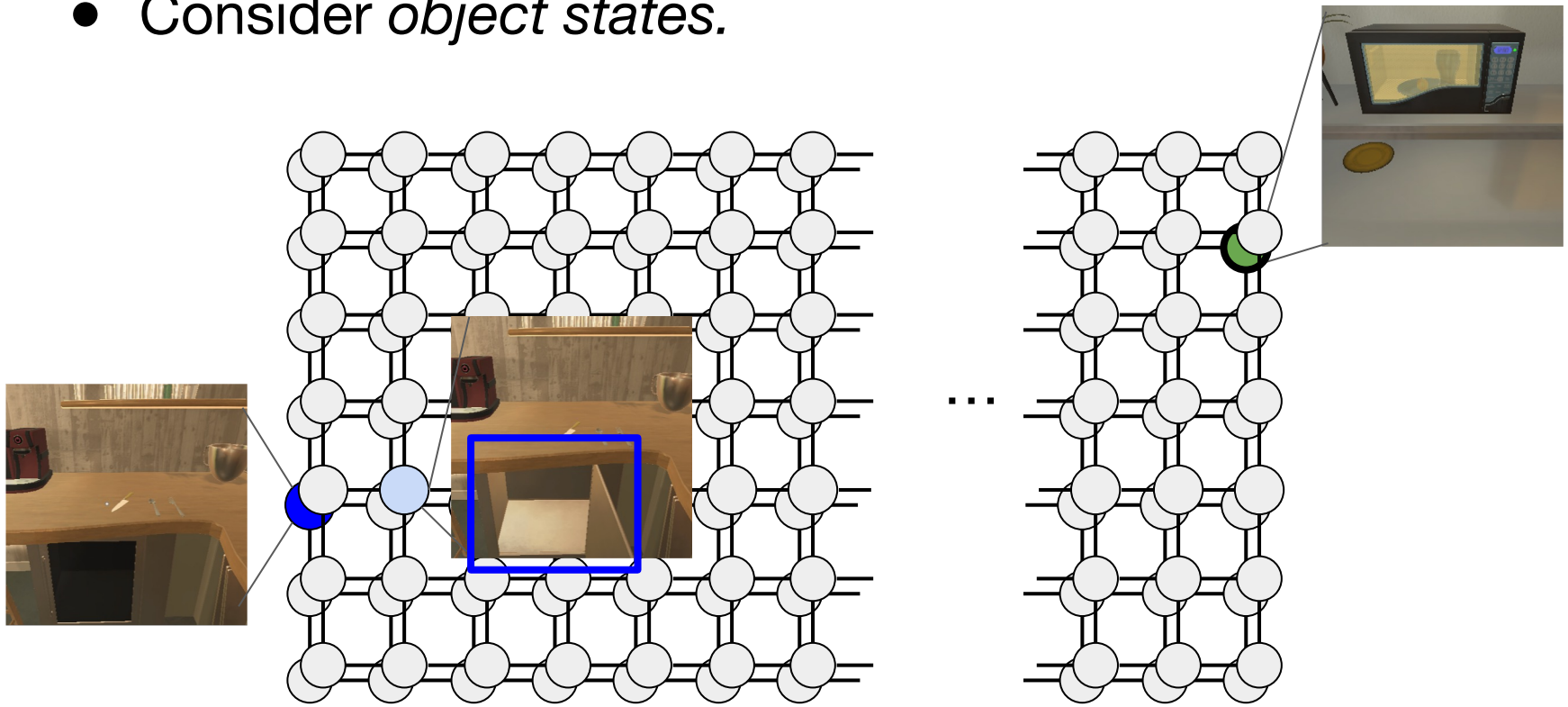
Can We Do Open Loop Planning + Scoring In this Space?

- Navigate on a grid of occupiable space.



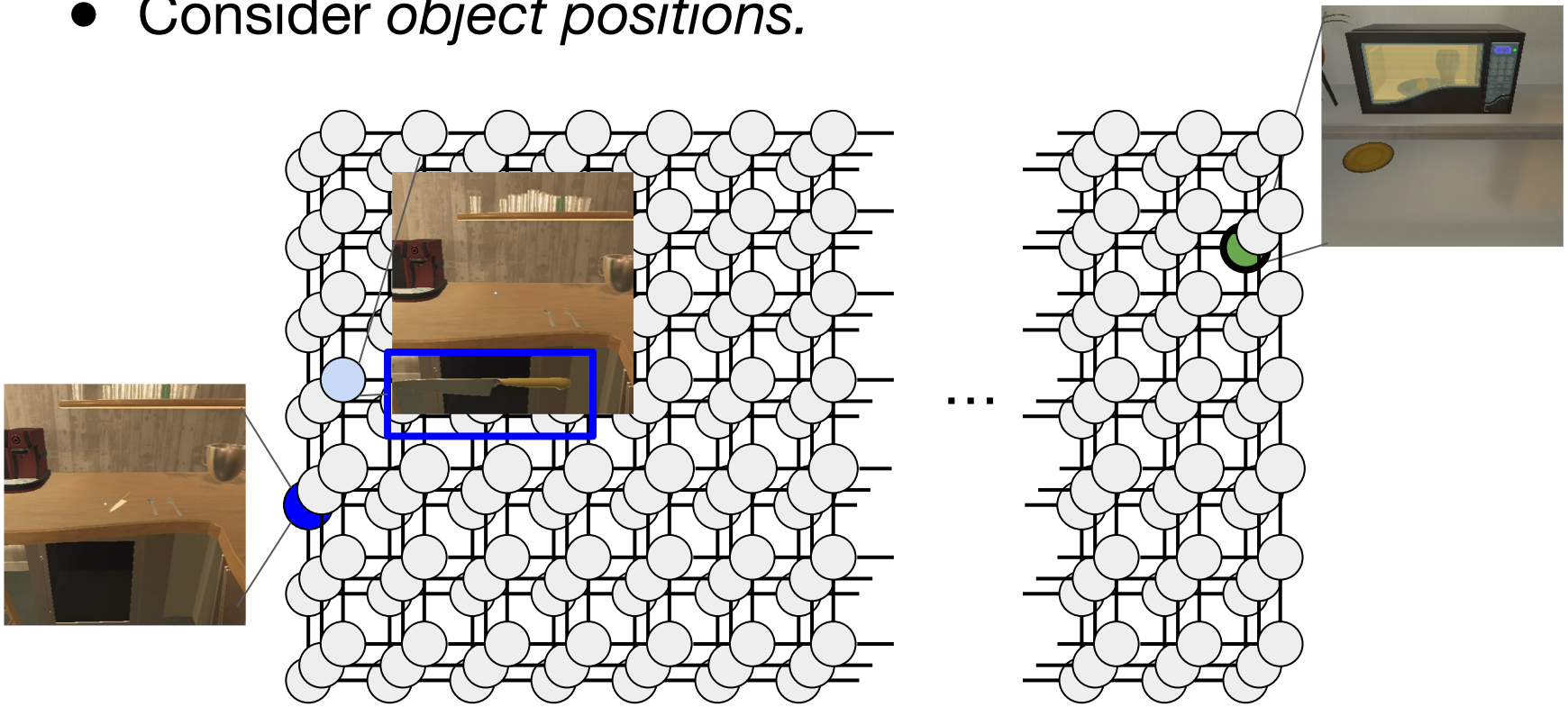
Can We Do Open Loop Planning + Scoring In this Space?

- Consider *object states.*



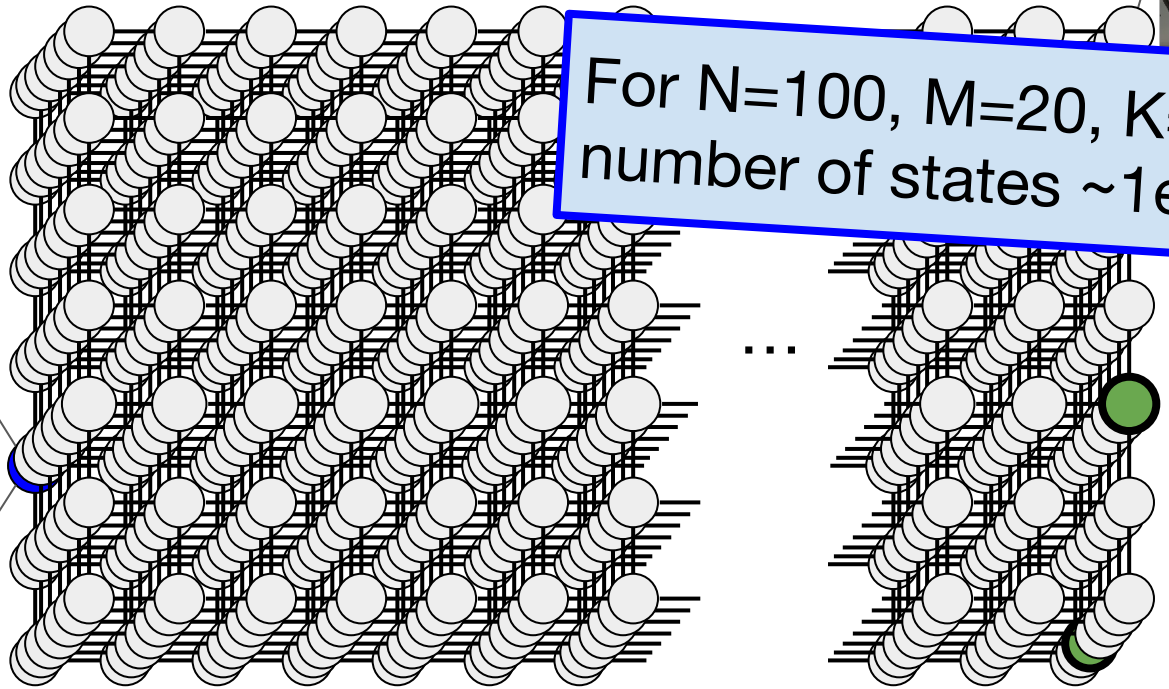
Can We Do Open Loop Planning + Scoring In this Space?

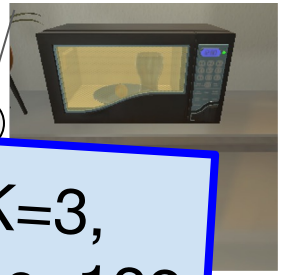
- Consider *object positions.*



Intractable, ~Discrete State Space with Discrete Actions

- Positions are continuous in AI2THOR
- Descritized, still ~N^(1+MK) states

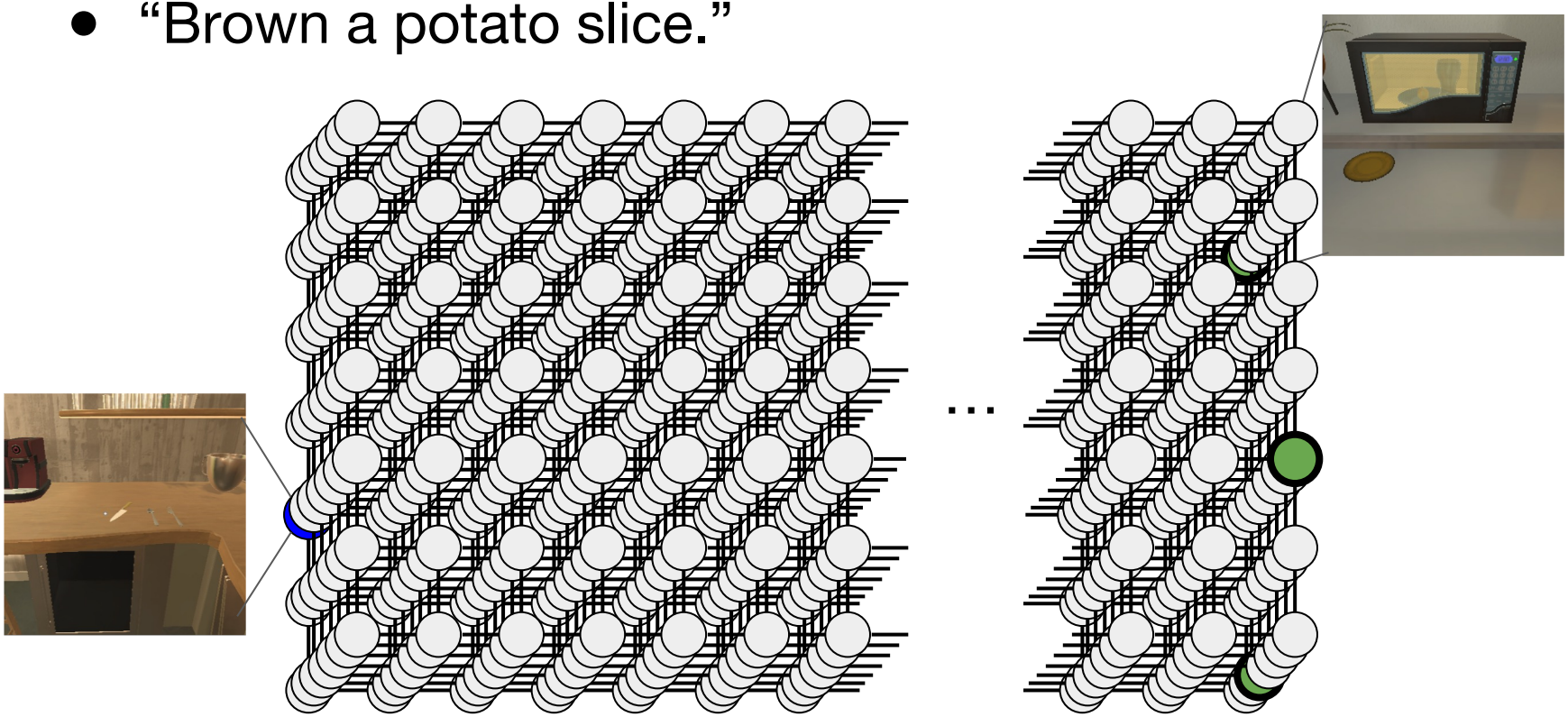




For N=100, M=20, K=3,
number of states ~1e+122

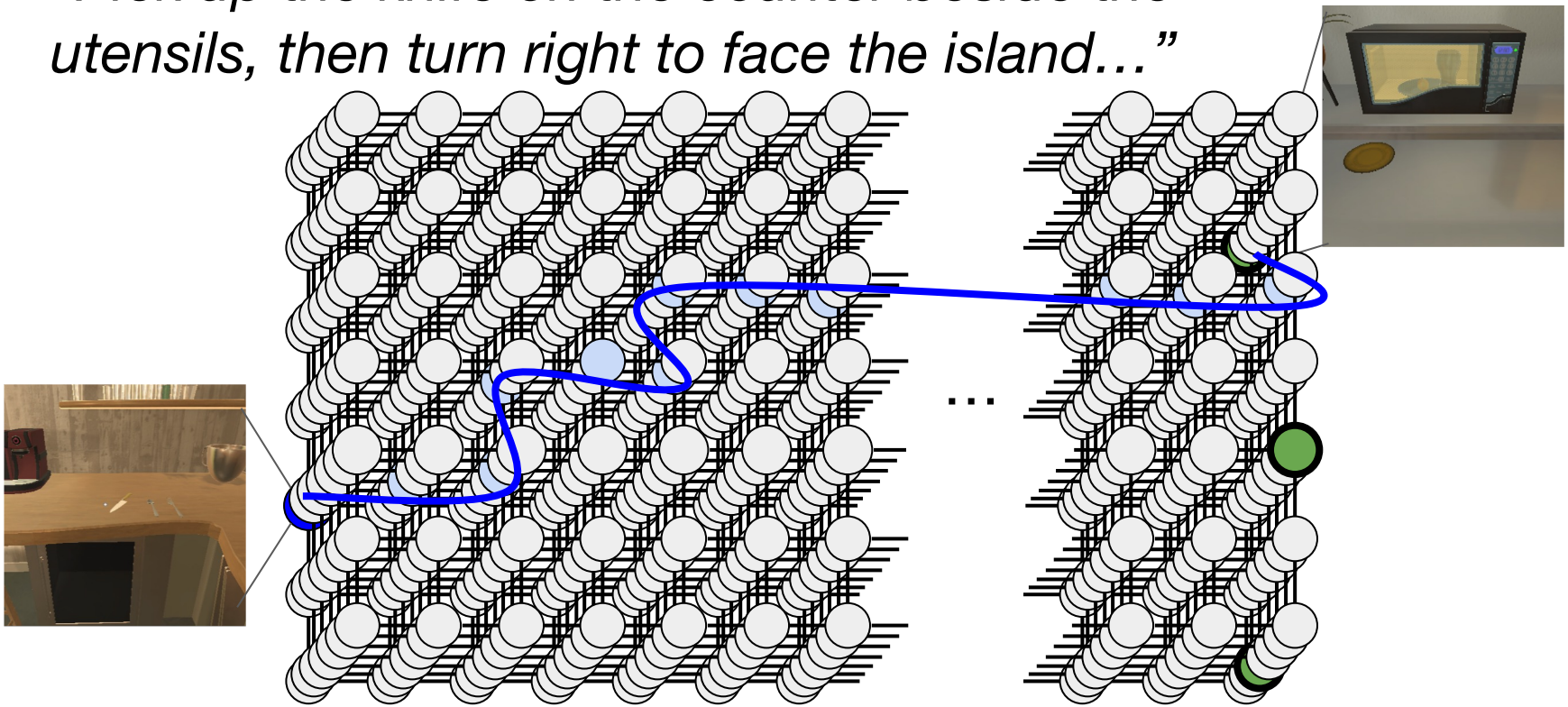
High-level instructions provide hints about a goal

- “Brown a potato slice.”



Low-level instructions provide hints to a “path”

“Pick up the knife on the counter beside the utensils, then turn right to face the island…”



Action and State Space

- Chain together low-level actions to accomplish goals.
- Navigation + Manipulation actions:
  - Predict an interaction mask (*wrapper* for AI2THOR).

ALFRED Seq2Seq

Put In

$\mathbf{a}_t$; $\mathbf{m}_t$

Action Decoder

x_t

Attention over x

h_t

LSTM

a_{t-1}

Language Instructions Encoding (t=0)

Put a chilled cup of water in the cupboard. Take a step to your right then walk to the counter in front of you then turn right and walk to the fridge. Open the freezer and grab the glass from behind the egg then close the door. Turn around and walk to the counter then turn right. Put the cup in the sink then full it with water and pick it back up remembering to shut off the tap. ...

BiLSTM Encoder

x_1 x_2 x_3 ... x_L

x

Visual Encoding (t)

Resnet Conv5

Encoding

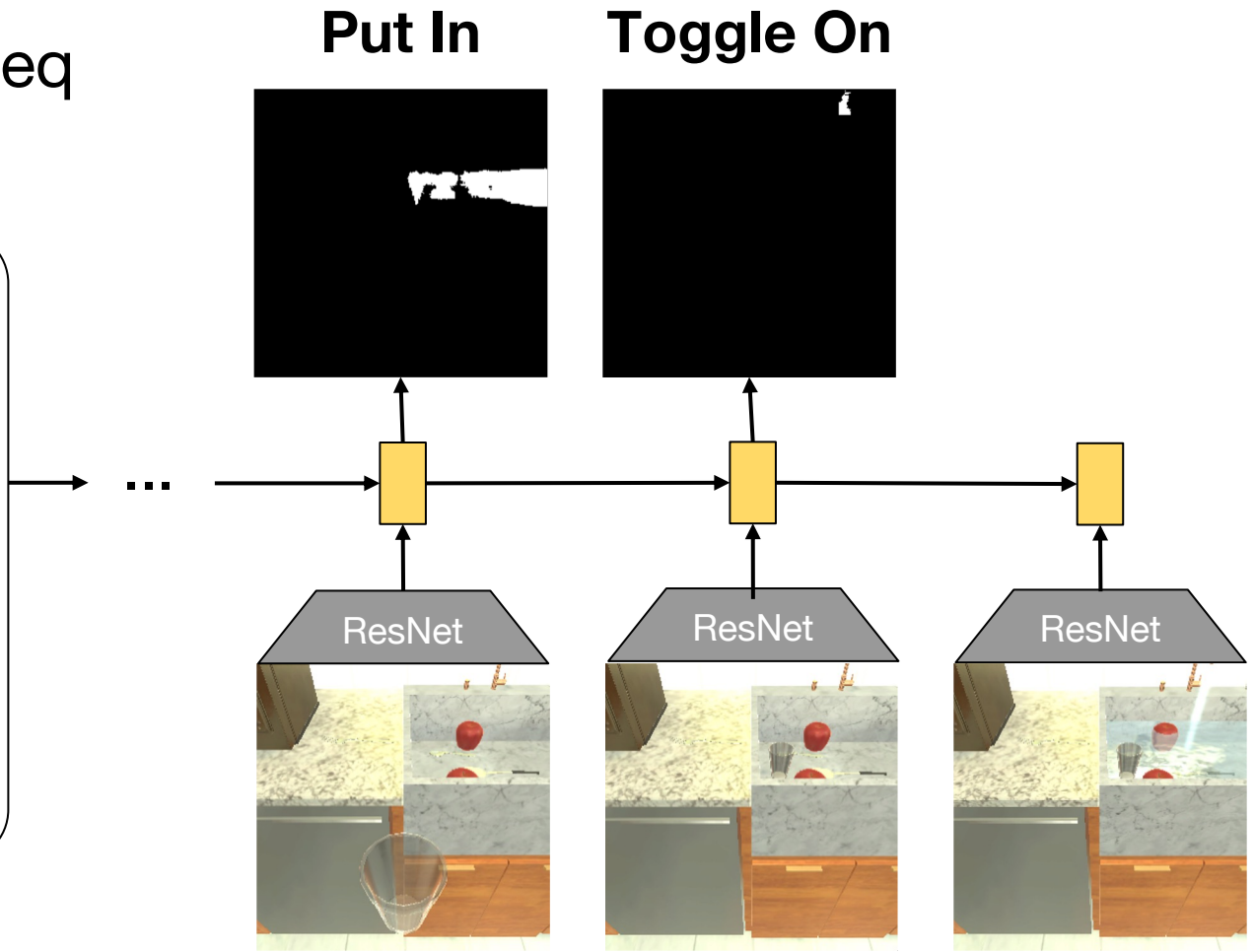
v_t

ALFRED Seq2Seq

Language Directive Encoding

Put a chilled cup of water in the cupboard. Take a step to your right then walk to the counter in front of you then turn right and walk to the fridge. Open the freezer and grab the glass from behind the egg then close the door. Turn around and walk to the counter then turn right. Put the cup in the sink then full it with water and pick it back up remembering to shut off the tap. ...

BiLSTM Encoder



ALFRED Seq2Seq

$\mathbf{a}_t$; $\mathbf{m}_t$

Action Decoder

x_t

Attention over x

h_t

LSTM

a_{t-1}

Language Instructions Encoding (t=0)

Put a chilled cup of water in the cupboard. Take a step to your right then walk to the counter in front of you then turn right and walk to the fridge. Open the freezer and grab the glass from behind the egg then close the door. Turn around and walk to the counter then turn right. Put the cup in the sink then full it with water and pick it back up remembering to shut off the tap. ...

BiLSTM Encoder

x_1 x_2 x_3 … x_L

x

Visual Encoding (t)

Resnet Conv5

Encoding

v_t

- **S**: underlying environment
- **A**: what agent can do
- **T**: how what the agent does changes **S**
- **Om**: what can be observed by the agent

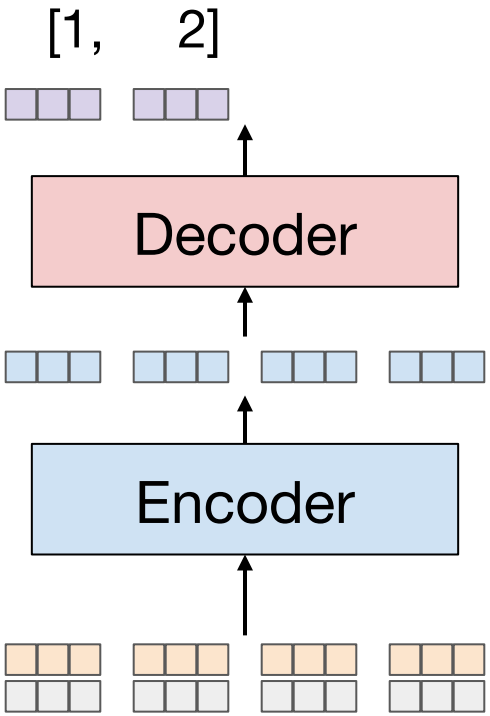
History $\mathcal{H} = \{(v_i, l_i), i = 1 \ldots T; v_i \in \mathcal{I}, l_i \in \mathcal{L}\}$

Policy $\pi : \mathcal{I} \times \mathcal{L} \times \mathcal{H} \rightarrow \mathcal{A}$

Why Would We Use an LSTM Anymore?

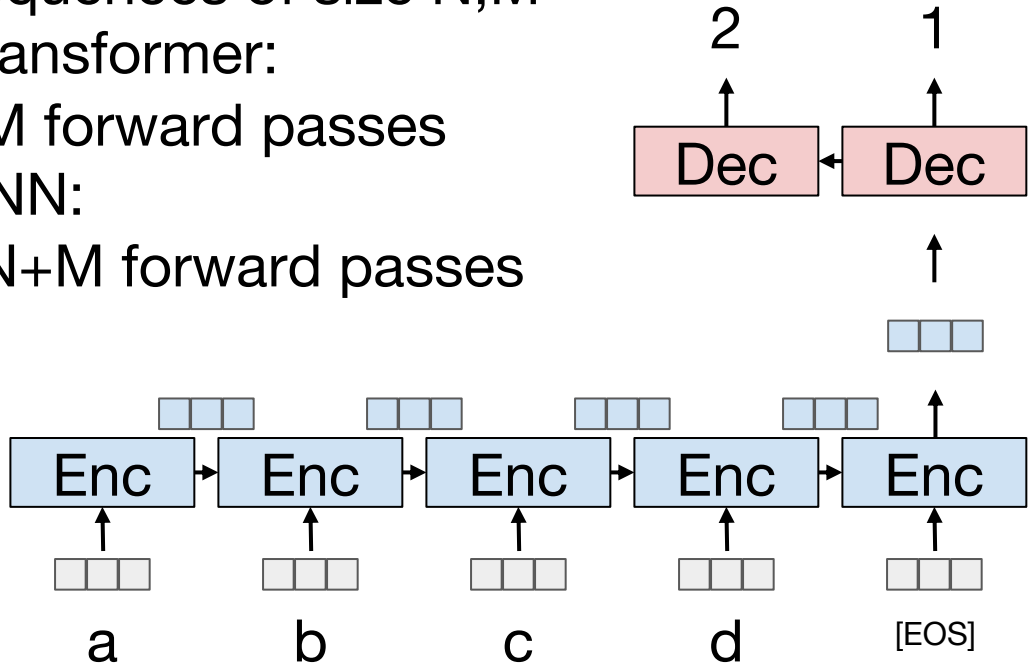
- Can we formulate language in, action policy out problems using Transformers?
- Let's think through what Transformers *gain* us and what they *lose* us compared to recurrent architectures

The Key Ingredient of the Transformer over the RNN

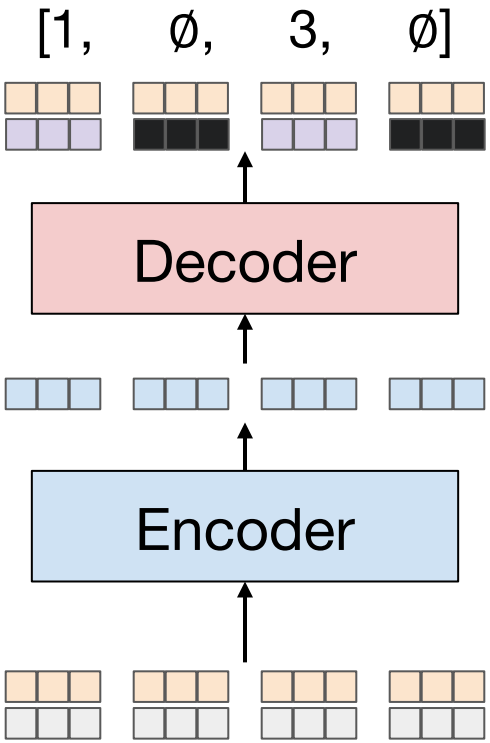


{(a, *i1*), (b, *i2*), (c, *i3*), (d, *i4*)}

- Encode/decode sequences of size N,M
- Transformer:
  M forward passes
- RNN:
  N+M forward passes

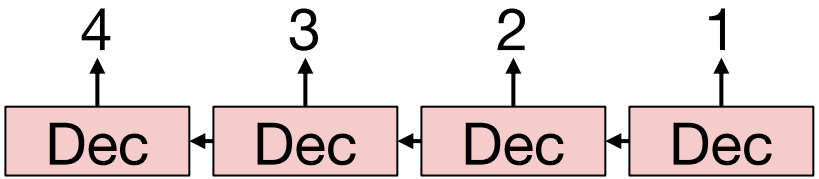


The Key Ingredient of the Transformer over the RNN

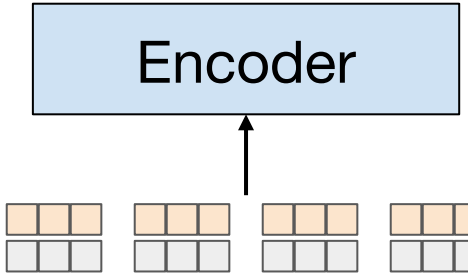


{(a, *i1*), (b, *i2*), (c, *i3*), (d, *i4*)}

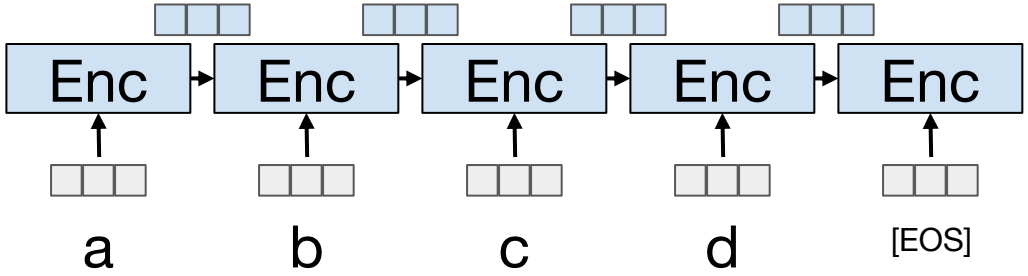
- Encode size N, decode multiple tokens
- Transformer: **one forward pass**
- RNN: stuck at N+M
  - Need to decode whole sequence, token by token, to update state for next prediction.
  - Can’t “take advantage” of decoder ‘given’ tokens to speed training.



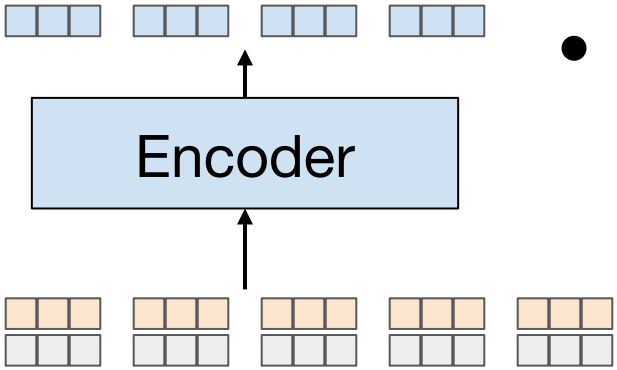
The Key Ingredient of the RNN over the Transformer



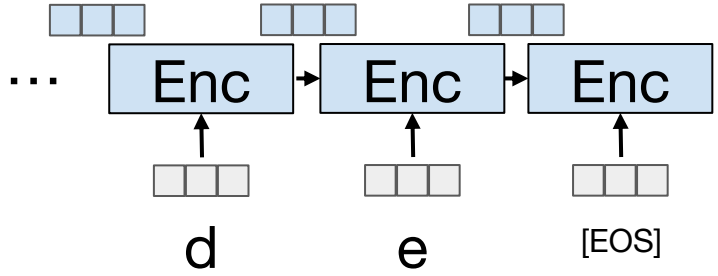
{(a, *i1*), (b, *i2*), (c, *i3*), (d, *i4*)}



- RNN creates one *state* per input
- Transformer inputs are interdependent



{(a, *i1*), (b, *i2*), (c, *i3*), (d, *i4*), (e, *i5*)}



Who cares if there's a state?

- RNN operates on a sequence and compresses each input into a continually refined *state* from which to decode
  - Word sequences [e.g., Machine Translation]
  - Image sequences [e.g., Video Understanding; VLN]
  - *Assumption*: state sufficiently represents the entire past
- Transformer operates on a set, compresses the whole set interdependently into an embedding from which to decode
  - Spatially and temporally embedded inputs (words, regions)
  - *Assumption*: all output depends on all input

Who cares if there's a state?

- Let's look at an ALFRED example

Goal: Put a cold egg in the sink.

Inst:
Cross the kitchen, go between the counter and the stove, then turn to the right and face the sink.
Pick up the egg from in the sink.
Bring the egg with you to go face the white fridge.
Open the fridge and put the egg inside, close the door and let it get cool, then take the egg back out of the fridge.
Bring the egg with you and go back over to face the sink.
Put the egg back in the sink.

https://askforalfred.com/?vid=42478

Who cares if there's a state?

- Probably your compute constraints

Goal: Put a cold egg in the sink.

Inst:
Cross the kitchen, go between the counter and the stove, then turn to the right and face the sink.

... t-1

Pick up the egg from in the sink.

t

Transformer

RNN

https://askforalfred.com/?vid=42478

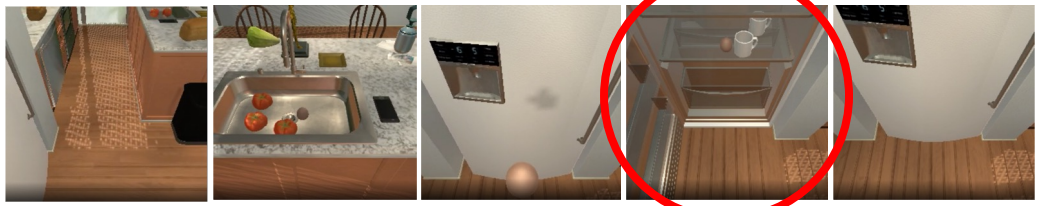
The *quality* of your learned state can be crucial

Goal: Put a cold egg in the sink.

... t-1

Inst:

Cross the kitchen, go between the counter and the stove, then turn to the right and face the sink.
Pick up the egg from in the sink.
Bring the egg with you to go face the white fridge.
Open the fridge and put the egg inside, close the door and let it get cool,



then take the egg back out of the fridge.

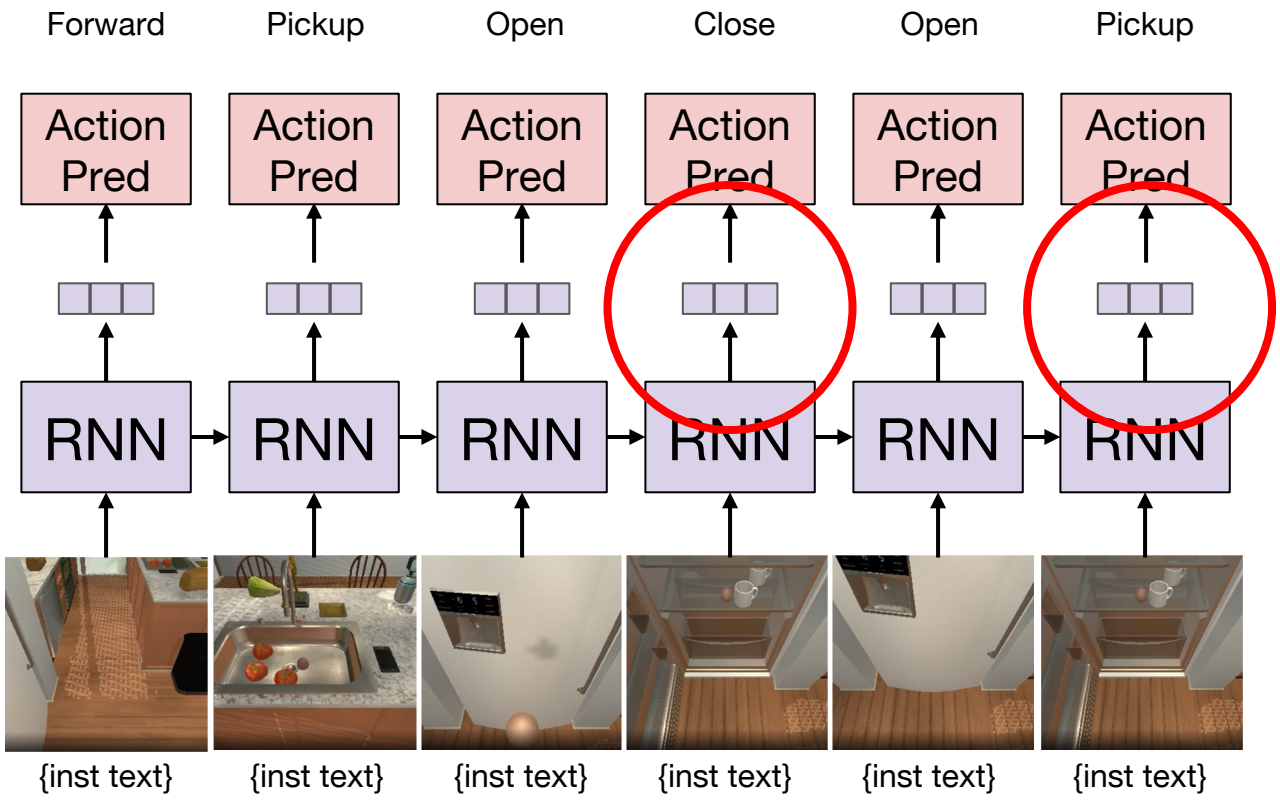
t

Transformer

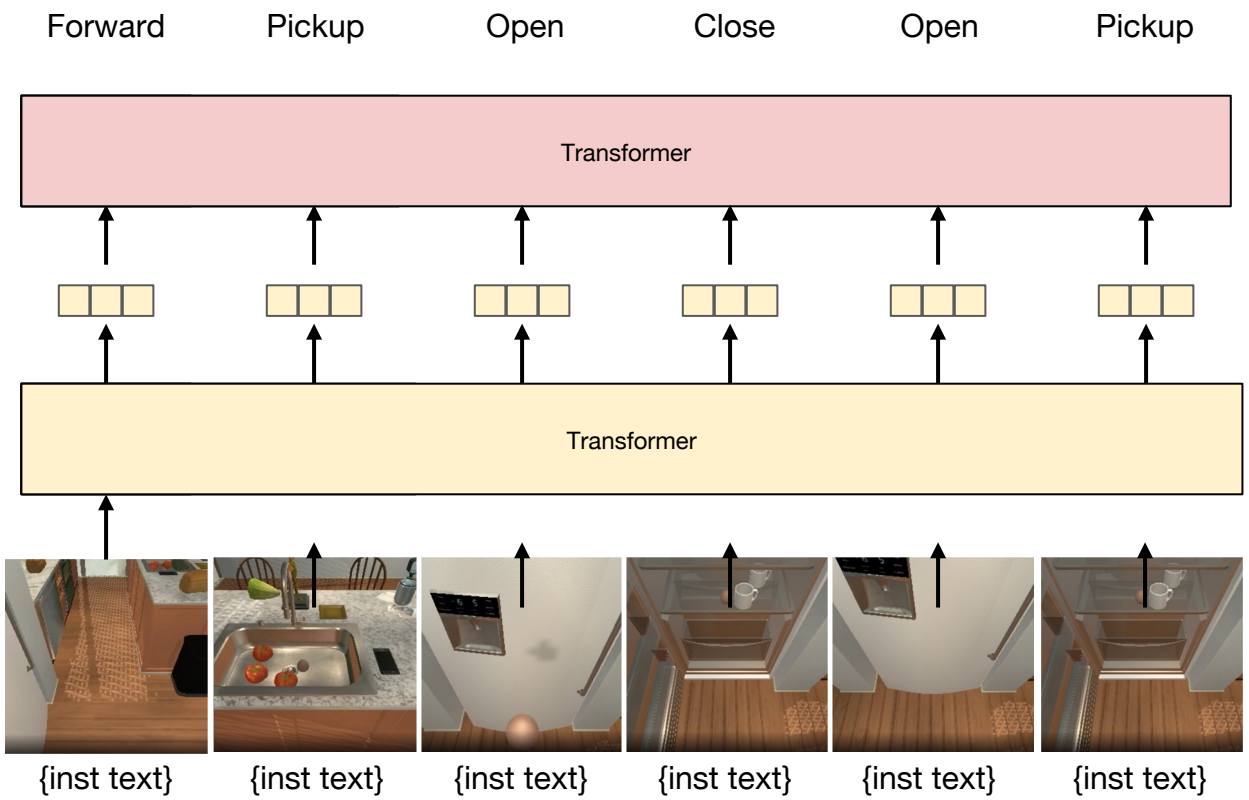
RNN

https://askforalfred.com/?vid=42478

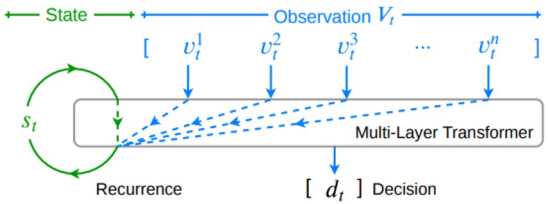
The *quality* of your learned state can be crucial

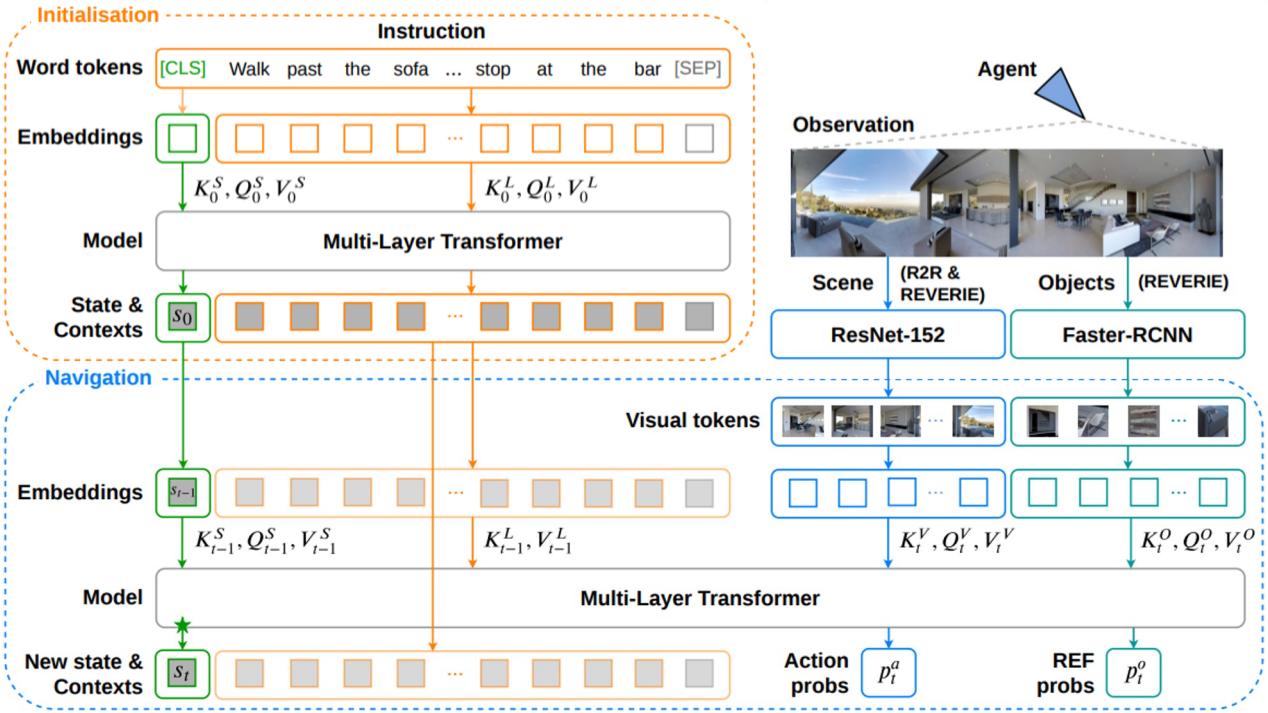


Without States, Our Transformer has to Learn So Much!



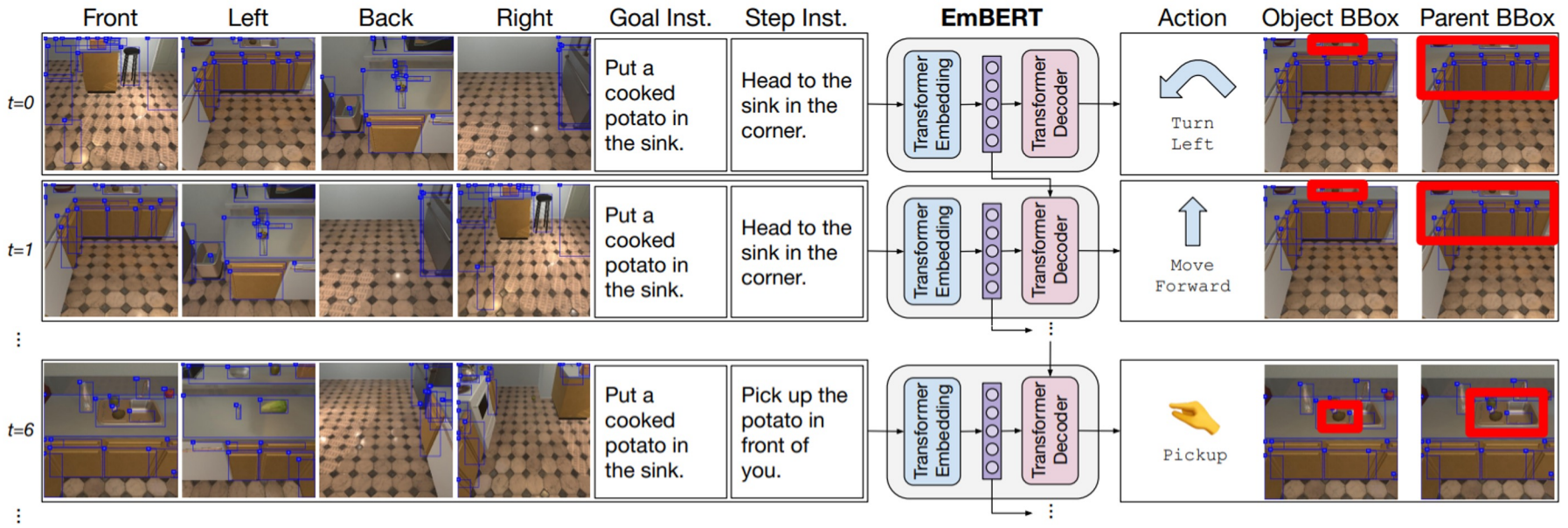
Stateful Transformers: VLN





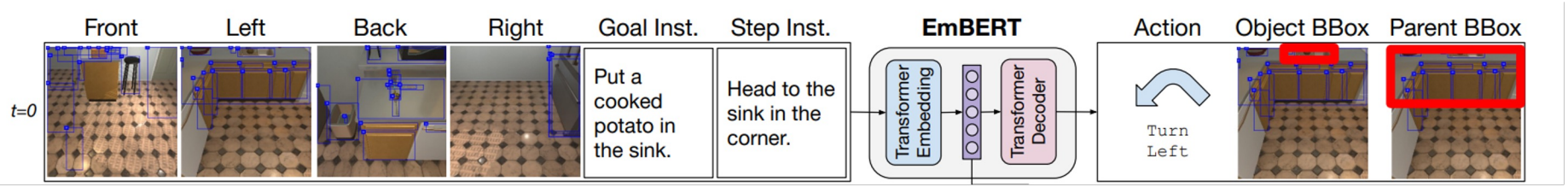
https://arxiv.org/abs/2011.13922

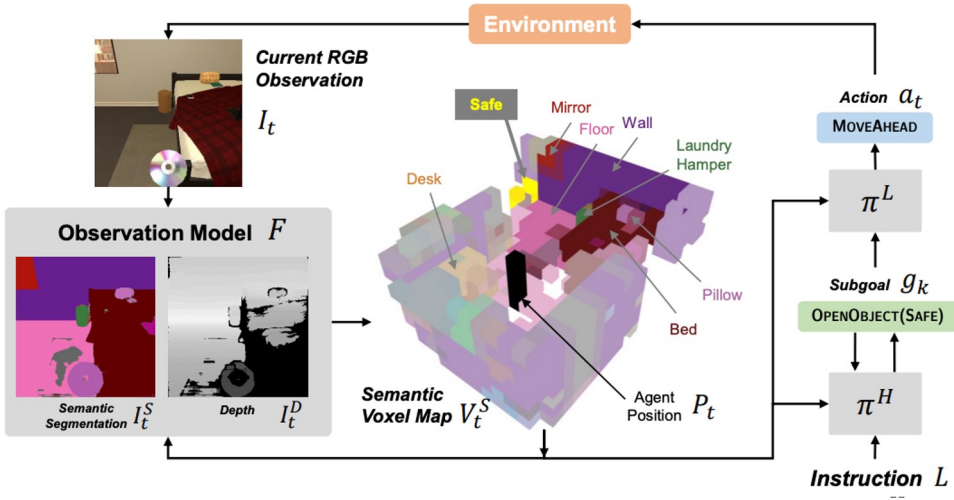
Stateful Transformers: ALFRED

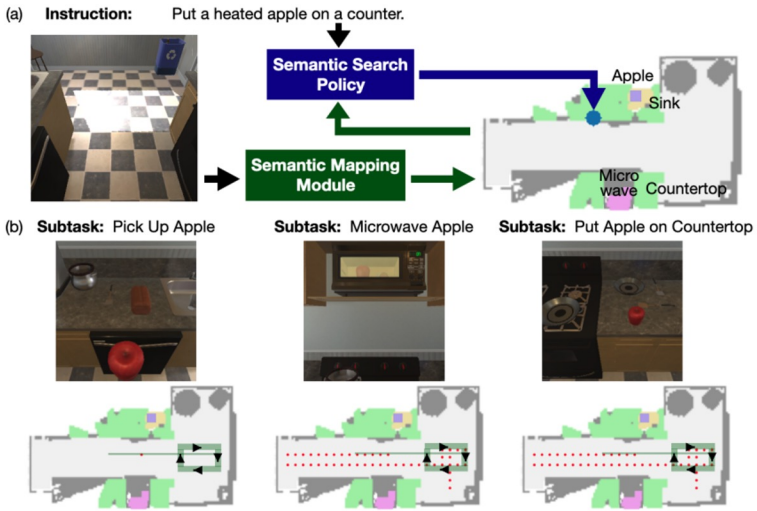


https://arxiv.org/abs/2108.04927

Implicit and Explicit States







https://arxiv.org/abs/2108.04927 ; https://arxiv.org/abs/2107.05612; https://arxiv.org/abs/2110.07342

ALFRED Semantic Mapping

https://hlsm-alfred.github.io/

Deep Learning for Agents

- So far, we've talked about situations where we have *demonstrations* available as training data
  - Enables us to perform learning from demonstrations / imitation learning / supervised learning
- What if we have no demonstrations, just a binary signal that tells us something like "yep, you completed the goal" or "nope, it's not done yet"?
- Then we move into the realm of *reinforcement learning*

Overview of Today's Plan

- ~~Course organization and deliverables~~
- ~~Lecture 8 Recap~~
- ~~Deep Learning for Agents~~
    - Any questions before we move on?
- March 31st Project Roleplaying Breakouts

Action Items for You

- Your project midterm reports are due **Apr 3** 11:59pm
- Assignment 2 is *also due* **Apr 3** 11:59pm
- For those doing a Paper Roleplaying Breakout session today, turn in your 2-page paper role report by 11:59pm tonight through the gDrive submission form

Paper Roleplaying Breakouts: March 31 [Schedule]

- On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜 **(Lounge outside of THH 201; find Jesse)**
- Neural Word Embedding as Implicit Matrix Factorization **(Back of THH 201; find Deqing)**
- CLIPort: What and Where Pathways for Robotic Manipulation **(Front of THH 201; find Gautam)**
- Learning Transferable Visual Models From Natural Language Supervision **(Virtual; join Shihan)**
- Auto-Encoding Variational Bayes **(Virtual; join Ayush)**

CSCI 566: Deep Learning and Its Applications

Jesse Thomason

Lecture 9: Deep Learning Agents