Probing Emergent Semantics in Predictive Agents via Question Answering

Abhishek Das* [1] **Federico Carnevale*** [2] **Hamza Merzic** [2] **Laura Rimell** [2] **Rosalia Schneider** [2] **Josh Abramson** [2]
Alden Hung [2] **Arun Ahuja** [2] **Stephen Clark** [2] **Gregory Wayne** [2] **Felix Hill** [2]

Abstract

Recent work has shown how predictive modeling can endow agents with rich knowledge of their surroundings, improving their ability to act in complex environments. We propose question-answering as a general paradigm to decode and understand the representations that such agents develop, applying our method to two recent approaches to predictive modeling – action-conditional CPC (Guo et al., 2018) and SimCore (Gregor et al., 2019). After training agents with these predictive objectives in a visually-rich, 3D environment with an assortment of objects, colors, shapes, and spatial configurations, we probe their internal state representations with synthetic (English) questions, without backpropagating gradients from the question-answering decoder into the agent. The performance of different agents when probed this way reveals that they learn to encode factual, and seemingly compositional, information about objects, properties and spatial relations from their physical environment. Our approach is intuitive, *i.e.* humans can easily interpret responses of the model as opposed to inspecting continuous vectors, and model-agnostic, *i.e.* applicable to any modeling approach. By revealing the implicit knowledge of objects, quantities, properties and relations acquired by agents as they learn, *question-conditional agent probing* can stimulate the design and development of stronger predictive learning objectives.

1. Introduction

Since the time of Plato, philosophers have considered the apparent distinction between "knowing how" (procedural knowledge or skills) and "knowing what" (propositional knowledge or facts). It is uncontroversial that deep reinforcement learning (RL) agents can effectively acquire procedural knowledge as they learn to play games or solve tasks. Such knowledge might manifest in an ability to find all of the green apples in a room, or to climb all of the ladders while avoiding snakes. However, the capacity of such agents to acquire factual knowledge about their surroundings – of the sort that can be readily hard-coded in symbolic form in classical AI – is far from established. Thus, even if an agent successfully climbs ladders and avoids snakes, we have no certainty that it 'knows' that ladders are brown, that there are five snakes nearby, or that the agent is currently in the middle of a three-level tower with one ladder left to climb.

The acquisition of knowledge about objects, properties, relations and quantities by learning-based agents is desirable for several reasons. First, such knowledge should ultimately complement procedural knowledge when forming plans that enable execution of complex, multi-stage cognitive tasks. Second, there seems (to philosophers at least) to be something fundamentally human about having knowledge of facts or propositions (Stich, 1979). If one of the goals of AI is to build machines that can engage with, and exhibit convincing intelligence to, human users (*e.g.* justifying their behaviour so humans understand/trust them), then a need for uncovering and measuring such knowledge in learning-based agents will inevitably arise.

Here, we propose the question-conditional probing of agent internal states as a means to study and quantify the knowledge about objects, properties, relations and quantities encoded in the internal representations of neural-network-based agents. Couching an analysis of such knowledge in terms of question-answering has several pragmatic advantages. First, question-answering provides a general purpose method for agent-analysis and an intuitive investigative tool for humans – one can simply *ask* an agent what it knows about its environment and get an answer back, without having to inspect internal activations. Second, the space of questions is essentially open-ended – we can pose arbitrarily complex questions to an agent, enabling a comprehensive analysis of the current state of its propositional knowledge. Question-answering has previously been studied in textual (Rajpurkar et al., 2016; 2018; Devlin et al., 2018; Yang et al., 2019), visual (Malinowski & Fritz, 2014; Antol et al., 2015; Das et al., 2017) and embodied (Gordon et al.,

*Equal contribution [1]Georgia Institute of Technology [2]DeepMind. Correspondence to: <felixhill@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

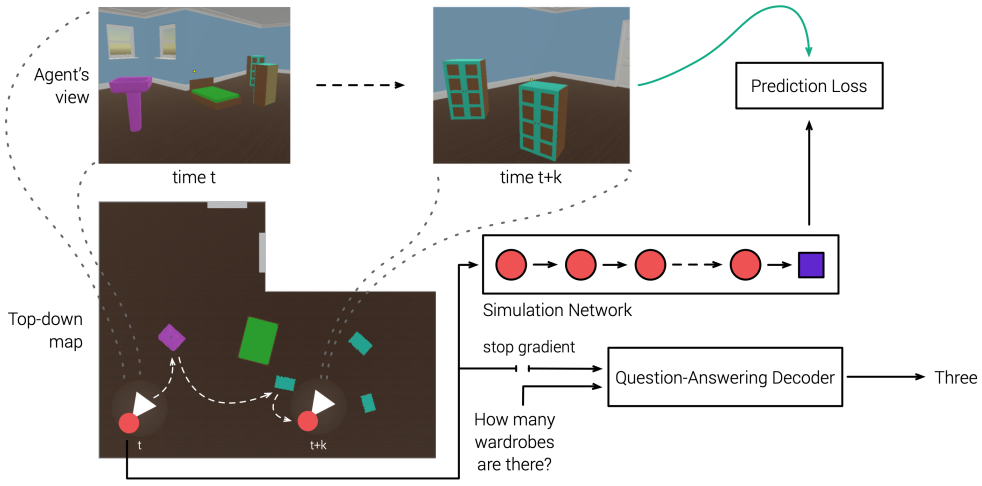


Figure 1. We train predictive agents to explore a visually-rich 3D environment with an assortment of objects of different shapes, colors and sizes. As the agent navigates (trajectory shown in white on the top-down map), an auxiliary network learns to simulate representations of future observations (labeled 'Simulation Network') k steps into the future, self-supervised by a loss against the ground-truth egocentric observation at $t+k$. Simultaneously, another decoder network is trained to extract answers to a variety of questions about the environment, conditioned on the agent's internal state but without affecting it (notice 'stop gradient' – gradients from the QA decoder are not backpropagated into the agent). We use this question-answering paradigm to decode and understand the internal representations that such agents develop. Note that the top-down map is only shown for illustration and not available to the agent.

2018; Das et al., 2018a) settings. Crucially, however, these systems are trained end-to-end for the goal of answering questions. Here, we utilize question-answering simply to probe an agent's internal representation, without backpropagating gradients from the question-answering decoder into the agent. That is, we view question-answering as a general purpose (conditional) decoder of environmental information designed to assist the development of agents by revealing the extent (and limits) of their knowledge.

Many techniques have been proposed for endowing agents with general (*i.e.* task-agnostic) knowledge, based on both hard-coding and learning. Here, we specifically focus on the effect of self-supervised predictive modeling – a learning-based approach – on the acquisition of propositional knowledge. Inspired by learning in humans (Elman, 1990; Rao & Ballard, 1999; Clark, 2016; Hohwy, 2013), predictive modeling, *i.e.* predicting future sensory observations, has emerged as a powerful method to learn general-purpose neural network representations (Elias, 1955; Atal & Schroeder, 1970; Schmidhuber, 1991; Schaul & Ring, 2013; Schaul et al., 2015; Silver et al., 2017; Wayne et al., 2018; Guo et al., 2018; Gregor et al., 2019; Recanatesi et al., 2019). These representations can be learned while exploring in and interacting with an environment in a task-agnostic manner, and later exploited for goal-directed behavior.

We evaluate predictive *vs.* non-predictive agents (both trained for exploration) on our question-answering testbed to investigate how much knowldge of object shapes, quantities, and spatial relations they acquire *solely by egocentric prediction*. The set includes a mix of questions that can plausibly be answered from a single observation or a few consecutive observations, and those that require the agent to integrate global knowledge of its entire surroundings.

Concretely, we make the following contributions:

- In a visually-rich 3D room environment developed in the Unity engine, we develop a set of questions designed to probe a diverse body of factual knowledge about the environment – from identifying shapes and colors ('What shape is the red object?') to counting ('How many blue objects are there?') to spatial relations ('What is the color of the chair near the table?'), exhaustive search ('Is there a cushion?'), and comparisons ('Are there the same number of tables as chairs?').
- We train RL agents augmented with predictive loss functions – 1) action-conditional CPC (Guo et al., 2018) and 2) SimCore (Gregor et al., 2019) – for an exploration task and analyze the internal representations they develop by decoding answers to our suite of questions. Crucially, the QA decoder is trained independent of the predictive agent and we find that QA performance is indicative of the agent's ability to capture global environment structure and semantics *solely through egocentric prediction*. We compare these predictive agents to strong non-predictive LSTM baselines as well as to an agent that is explicitly optimized for the question-answering task.
- We establish generality of the encoded knowledge by testing zero-shot generalization of a trained QA decoder to compositionally novel questions (unseen combinations of seen attributes), suggesting a degree of

compositionality in the internal representations captured by predictive agents.

2. Background and Related Work

Our work builds on studies of predictive modeling and auxiliary objectives in reinforcement learning as well as grounded language learning and embodied question answering.

Propositional knowledge is knowledge that a statement, expressed in natural or formal language, is true (Truncellito, 2007). Since at least Plato, epistemologist philosophers have contrasted propositional knowledge with *procedural knowledge* (knowledge of how to do something), and some (but not all) distinguish this from *perceptual knowledge* (knowledge obtained by the senses that cannot be translated into a proposition) (Dretske, 1995). An ability to exhibit this sort of knowledge in a convincing way is likely to be crucial for the long-term goal of having agents achieve satisfying interactions with humans, since an agent that cannot express its knowledge and beliefs in human-interpretable form may struggle to earn the trust of users.

Predictive modeling and auxiliary loss functions in RL. The power of predictive modeling for representation learning has been known since at least the seminal work of (Elman, 1990) on emergent language structures. More recent examples include Word2Vec (Mikolov et al., 2013), Skip-Thought vectors (Kiros et al., 2015), and BERT (Devlin et al., 2019) in language, while in vision similar principles have been applied to context prediction (Doersch et al., 2015; Noroozi & Favaro, 2016), unsupervised tracking (Wang & Gupta, 2015), inpainting (Pathak et al., 2016) and colorization (Zhang et al., 2016). More related to us is the use of such techniques in designing auxiliary loss functions for training model-free RL agents, such as successor representations (Dayan, 1993; Zhu et al., 2017a), value and reward prediction (Jaderberg et al., 2016; Hermann et al., 2017; Wayne et al., 2018), contrastive predictive coding (CPC) (Oord et al., 2018; Guo et al., 2018), and SimCore (Gregor et al., 2019).

Grounded language learning. Inspired by the work of (Winograd, 1972) on SHRDLU, several recent works have explored linguistic representation learning by grounding language into actions and pixels in physical environments – in 2D gridworlds (Andreas et al., 2017; Yu et al., 2018; Misra et al., 2017), 3D (Chaplot et al., 2018; Das et al., 2018a; Gordon et al., 2018; Cangea et al., 2019; Puig et al., 2018; Zhu et al., 2017a; Anderson et al., 2018; Gupta et al., 2017; Zhu et al., 2017b; Oh et al., 2017; Shu et al., 2018; Vogel & Jurafsky, 2010; Hill et al., 2020) and textual (Matuszek et al., 2013; Narasimhan et al., 2015) environments. Closest to our work is the task of Embodied Question Answering (Gordon et al., 2018; Das et al., 2018a;b; Yu et al., 2019; Wijmans et al., 2019) – where an embodied agent in an environment (*e.g.* a house) is asked to answer a question (*e.g.* "What color is the piano?"). Typical approaches to EmbodiedQA involve training agents to move for the goal of answering questions. In contrast, our focus is on learning a predictive model in a *goal-agnostic* exploration phase and using question-answering as a post-hoc testbed for evaluating the semantic knowledge that emerges in the agent's representations from predicting the future.

Neural population decoding. Probing an agent with a QA decoder can be viewed as a variant of neural population decoding, used as an analysis tool in neuroscience (Georgopoulos et al., 1986; Bialek et al., 1991; Salinas & Abbott, 1994) and more recently in deep learning (Guo et al., 2018; Gregor et al., 2019; Azar et al., 2019; Alain & Bengio, 2016; Conneau et al., 2018; Tenney et al., 2019). The idea is to test whether specific information is encoded in a learned representation, by feeding the representation as input to a probe network, generally a classifier trained to extract the desired information. In RL, this is done by training a probe to predict parts of the ground-truth state of the environment, such as an agent's position or orientation, without backpropagating through the agent's internal state.

Prior work has required a separate network to be trained for each probe, even for closely related properties such as position vs. orientation (Guo et al., 2018) or grammatical features of different words in the same sentence (Conneau et al., 2018). Moreover, each probe is designed with property-specific inductive biases, such as convnets for top-down views vs. MLPs for position (Gregor et al., 2019). In contrast, we train a single, general-purpose probe network that covers a variety of question types, with an inductive bias for language processing. This generality is possible because of the external conditioning, in the form of the question, supplied to the probe. External conditioning moreover enables agent analysis using novel perturbations of the probe's training questions.

Cognitive Science. Predictive modeling is thought to be a fundamental component of human cognition (Elman, 1990; Hohwy, 2013; Seth, 2015). In particular, it has been proposed that perception, learning and decision-making rely on the minimization of prediction error (Rao & Ballard, 1999; Clark, 2016). A well-established strand of work has focused on decoding predictive representations in brain states (Nortmann et al., 2013; Huth et al., 2016). The question of how prediction of sensory experience relates to higher-order conceptual knowledge is complex and subject to debate (Williams, 2018; Roskies & Wood, 2017), though some have proposed that conceptual knowledge, planning, reasoning, and other higher-order functions emerge in deeper layers of a predictive network. We focus on the emergence of propositional knowledge in a predictive agent's internal representations.

Table 1. QA task templates. In every episode, objects and their configurations are randomly generated, and these templates get translated to QA pairs for all unambiguous <shape, color> combinations. There are 50 shapes and 10 colors in total. See A.4 for details.

Question type	Template	Level codename	# QA pairs
Attribute	What is the color of the <shape>?	color	500
	What shape is the <color> object?	shape	500
Count	How many <shape> are there?	count_shape	200
	How many <color> objects are there?	count_color	40
Exist	Is there a <shape>?	existence_shape	100
Compare + Count	Are there the same number of <color1> objects as <color2> objects?	compare_n_color	180
	Are there the same number of <shape1> as <shape2>?	compare_n_shape	4900
Relation + Attribute	What is the color of the <shape1> near the <shape2>?	near_color	24500
	What is the <color> object near the <shape>?	near_shape	25000

3. Environment & Tasks

Environment. We use a Unity-based visually-rich 3D environment (see Figure 1). It is a single L-shaped room that can be programmatically populated with an assortment of objects of different colors at different spatial locations and orientations. In total, we use a library of 50 different objects, referred to as 'shapes' henceforth (*e.g.* chair, teddy, glass, *etc.*), in 10 different colors (*e.g.* red, blue, green, *etc.*). For a complete list of environment details, see Sec. A.4.

At every step, the agent gets a 96×72 first-person RGB image as its observation, and the action space consists of movements (move-{forward,back,left,right}), turns (turn-{up,down,left,right}), and object pick-up and manipulation (4 DoF: yaw, pitch, roll, and movement along the axis between the agent and object). See Table 5 in the Appendix for the full set of actions.

Question-Answering Tasks. We develop a range of question-answering tasks of varying complexity that test the agent's local and global scene understanding, visual reasoning, and memory skills. Inspired by (Johnson et al., 2017; Das et al., 2018a; Gordon et al., 2018), we programmatically generate a dataset of questions (see Table 1). These questions ask about the presence or absence of objects (existence_shape), their attributes (color, shape), counts (count_color, count_shape), quantitative comparisons (compare_count_color, compare_count_shape), and elementary spatial relations (near_color, near_shape). Unlike the fully-observable setting in CLEVR (Johnson et al., 2017), the agent does not get a global view of the environment, and must answer these questions from a sequence of partial egocentric observations. Moreover, unlike prior work on EmbodiedQA (Gordon et al., 2018; Das et al., 2018a), the agent is *not* being trained end-to-end to move to answer questions. It is being trained to explore, and answers are being decoded (without backpropagating gradients) from its internal representation. Thus, in order to answer these questions, the agent *must* learn to encode relevant aspects of the environment in a representation amenable to easy decoding into symbols (*e.g.* what does the word "chair" mean? or what representations does computing "how many" require?).

4. Approach

Learning an exploration policy. Predictive modeling has proven to be effective for an agent to develop general knowledge of its environment as it explores and behaves towards its goal, typically maximising environment returns (Gregor et al., 2019; Guo et al., 2018). Since we wish to evaluate the effectiveness of predictive modeling independent of the agent's specific goal, we define a simple task that stimulates the agent to visit all of the 'important' places in the environment (*i.e.* to acquire an exploratory but otherwise task-neutral policy). This is achieved by giving the agent a reward of $+1.0$ every time it visits an object in the room for the first time. After visiting all objects, rewards are refreshed and available to be consumed by the agent again (*i.e.* re-visiting an object the agent has already been to will now again lead to a $+1.0$ reward), and this process continues for the duration of each episode (30 seconds or 900 steps).

During training on this exploration task, the agent receives a first-person RGB observation x_t at every timestep t, and processes it using a convolutional neural network to produce z_t. This is input to an LSTM policy whose hidden state is h_t and output a discrete action a_t. The agent optimizes the discounted sum of future rewards using an importance-weighted actor-critic algorithm (Espeholt et al., 2018).

Training the QA-decoder. The question-answering decoder is operationalized as an LSTM that is initialized with the agent's internal representation h_t and receives the question as input at every timestep (see Fig. 2). The question is a string that we tokenise into words and then map to learned embeddings. The question decoder LSTM is then unrolled for a fixed number of computation steps after which it predicts a softmax distribution over the vocabulary of one-word answers to questions in Table 1, and is trained via a cross-entropy loss. Crucially, this QA decoder is trained

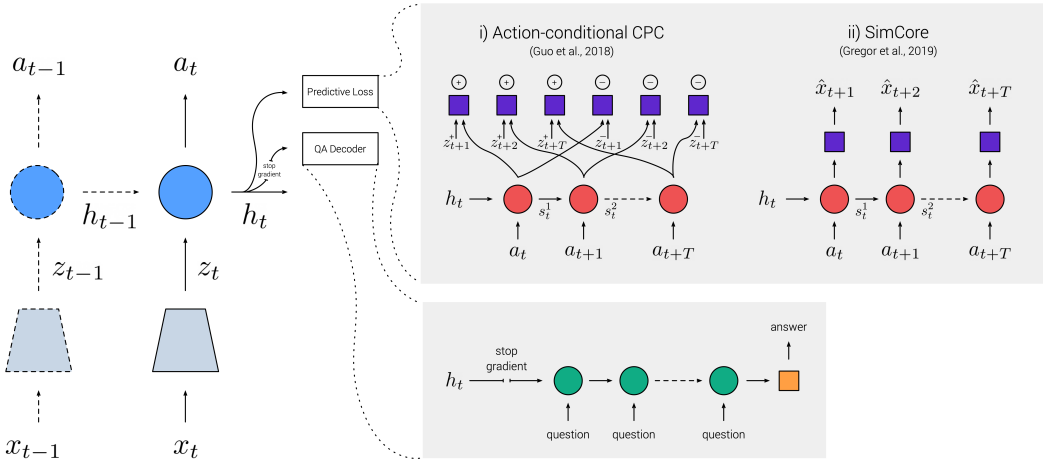


Figure 2. Approach: at every timestep t, the agent receives an RGB observation x_t as input, processes it using a convolutional neural network to produce z_t, which is then processed by an LSTM to select action a_t. The agent learns to explore – it receives a reward of 1.0 for navigating to each new object. As it explores the environment, it builds up an internal representation h_t, which receives pressure from an auxiliary predictive module to capture environment semantics so as to accurately predict consequences of its actions multiple steps into the future. We experiment with a vanilla LSTM agent and two recent predictive approaches – CPC|A (Guo et al., 2018) and SimCore (Gregor et al., 2019). The internal representations are then probed via a question-answering decoder whose gradients are not backpropagated into the agent. The QA decoder is an LSTM initialized with h_t and receiving the question at every timestep.

independent of the agent policy; *i.e.* gradients from this decoder are not allowed to flow back into the agent. We evaluate question-answering performance by measuring top-1 accuracy at the end of the episode – we consider the agent's top predicted answer at the last time step of the episode and compare that with the ground-truth answer.

The QA decoder can be seen as a general purpose decoder trained to extract object-specific knowledge from the agent's internal state without affecting the agent itself. If this knowledge is not retained in the agent's internal state, then this decoder will not be able to extract it. This is an important difference with respect to prior work (Gordon et al., 2018; Das et al., 2018a) – wherein agents were trained to move to answer questions, *i.e.* all parameters had access to linguistic information. Recall that the agent's navigation policy has been trained for exploration, and so the visual information required to answer a question need not be present in the observation at the end of the episode. Thus, through question-answering, we are evaluating the degree to which agents encode relevant aspects of the environment (object colors, shapes, counts, spatial relations) in their internal representations *and* maintain this information in memory beyond the point at which it was initially received. See A.1.3 for more details about the QA decoder.

4.1. Auxiliary Predictive Losses

We augment the baseline architecture described above with an auxiliary predictive head consisting of a simulation network (operationalized as an LSTM) that is initialized with the agent's internal state h_t and deterministically simulates future latent states $s_t^1, \ldots, s_t^k, \ldots$ in an open-loop manner, receiving the agent's action sequence as input. We evaluate two predictive losses – action-conditional CPC (Guo et al., 2018) and SimCore (Gregor et al., 2019). See Fig. 2 for overview, A.1.2 for details.

Action-conditional CPC (CPC|A, (Guo et al., 2018)) makes use of a noise contrastive estimation model to discriminate between true observations processed by the convolutional neural network z^+_{t+k} (k steps into the future) and negatives randomly sampled from the dataset z^-_{t+k}, in our case from other episodes in the minibatch. Specifically, at each timestep $t+k$ (up to a maximum), the output of the simulation core s_t^k and z^+_{t+k} are fed to an MLP to predict 1, and s_t^k and z^-_{t+k} are used to predict 0.

SimCore (Gregor et al., 2019) uses the simulated state s_t^k to condition a generative model based on ConvDRAW (Gregor et al., 2016) and GECO (Rezende & Viola, 2018) that predicts the distribution of true observations $p(x_{t+k}|h_t, a_{t,\ldots,(t+k)})$ in pixel space.

Baselines. We evaluate and compare the above approaches with 1) a vanilla RL agent without any auxiliary predictive losses (referred to as 'LSTM'), and 2) a question-only agent that receives zero-masked observations as input and is useful to measure biases in our question-answering testbed. Such a baseline is critical, particularly when working with simulated environments, as it can uncover biases in the environment's generation of tasks that can result in strong but uninteresting performance from agents capable of powerful function approximation (Thomason et al., 2019).

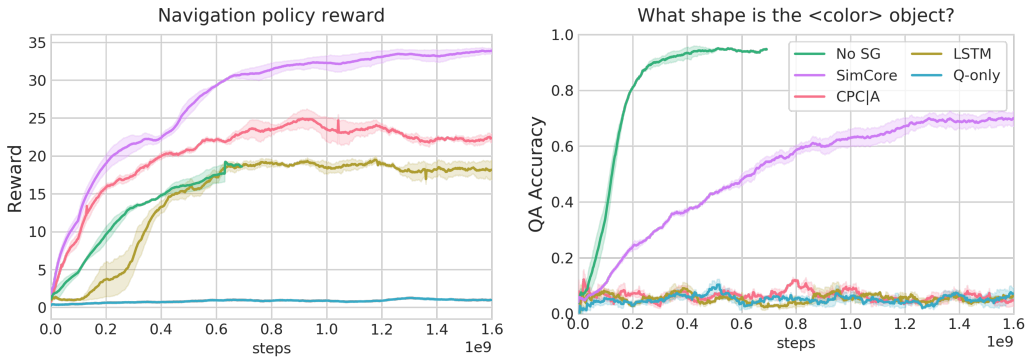


Figure 3. L – Reward in an episode. R – Top-1 QA accuracy. Averaged over 3 seeds. Shaded region is 1 SD.

No stop gradient. We also compare against an agent without blocking the QA decoder gradients (labeled 'No SG'). This model differs from the above in that it is trained end-to-end – with supervision – to answer the set of questions in addition to the exploration task. Hence, it represents an agent receiving privileged information about how to answer and its performance provides an upper bound for how challenging these question-answering tasks are in this context.

5. Experiments & Results

5.1. Question-Answering Performance

We begin by analyzing performance on a single question – `shape` – which are of the form "what shape is the <color> object?". Figure 3 shows the average reward accumulated by the agent in one episode (left) and the QA accuracy at the last timestep of the episode (right) for all approaches over the course of training. We make the following observations:

- **All agents learn to explore**. With the exception 'question-only', all agents achieve high reward on the exploration task. This means that they visited all objects in the room more than once each and therefore, in principle, have been exposed to sufficient information to answer all questions.
- **Predictive models aid navigation**. Agents equipped with auxiliary predictive losses – CPC|A and SimCore – collect the most rewards, suggesting that predictive modeling helps navigate the environment efficiently. This is consistent with findings in (Gregor et al., 2019).
- **QA decoding from LSTM and CPC|A representations is no better than chance**.
- **SimCore's representations lead to best QA accuracy**. SimCore gets to a QA accuracy of $\sim 72\%$ indicating that its representations best capture propositional knowledge and are best suited for decoding answers to questions. Figure 4 (Left) shows example predictions.
- **Wide gap between SimCore and No SG**. There is a $\sim 24\%$ gap between SimCore and the No SG oracle, suggesting scope for better auxiliary predictive losses.

It is worth emphasizing that answering this `shape` question from observations is not a challenging task in and of itself. The No SG agent, which is trained end-to-end to optimize both for exploration and QA, achieves almost-perfect accuracy ($\sim 96\%$). The challenge arises from the fact that we are not training the agent end-to-end – from pixels to navigation to QA – but decoding the answer from the agent's internal state, which is learned agnostic to the question. The answer can only be decoded if the agent's internal state contains relevant information represented in an easily-decodable way.

Decoder complexity. To explore the possibility that answer-relevant information is present in the agent's internal state but requires a more powerful decoder, we experiment with QA decoders of a range of depths. As detailed in Figure 7 in the appendix, we find that using a deeper QA decoder with SimCore does lead to higher QA accuracy (from $1 \rightarrow 12$ layers), although greater decoder depths become detrimental after 12 layers. Crucially, however, in the non-predictive LSTM agent, the correct answer cannot be decoded irrespective of QA decoder capacity. This highlights an important aspect of our question-answering evaluation paradigm – that while the absolute accuracy at answering questions may also depend on decoder capacity, relative differences provide an informative comparison between internal representations developed by different agents.

Table 2 shows QA accuracy for all QA tasks (see Figure 8 in appendix for training curves). The results reveal large variability in difficulty across question types. Questions about attributes (`color` and `shape`), which can be answered from a single well-chosen frame of visual experience, are the easiest, followed by spatial relationship questions (`near_color` and `near_shape`), and the hardest are counting questions (`count_color` and `count_shape`). We further note that:

- **All agents perform better than the question-only baseline**, which captures any biases in the environment or question distributions (enabling strategies such as constant prediction of the most-common answer).
- **CPC|A representations are not better than LSTM**

Table 2. Top-1 accuracy on question-answering tasks.

	Overall	shape	color	exist	count_shape	count_color	compare_n_color	compare_n_shape	near_shape	near_color
Baseline: Question-only	29 ± 3	04 ± 2	10 ± 2	63 ± 4	24 ± 3	24 ± 3	49 ± 3	70 ± 3	04 ± 2	09 ± 3
LSTM	31 ± 4	04 ± 1	10 ± 2	54 ± 6	34 ± 3	38 ± 3	53 ± 3	70 ± 3	04 ± 2	09 ± 3
CPC|A	32 ± 3	06 ± 2	08 ± 2	64 ± 3	**39** ± 3	39 ± 3	50 ± 4	70 ± 3	06 ± 2	10 ± 3
SimCore	**60** ± 3	**72** ± 3	**81** ± 3	**72** ± 3	**39** ± 3	**57** ± 3	**56** ± 3	**73** ± 3	**30** ± 3	**59** ± 3
Oracle: No SG	63 ± 3	96 ± 2	81 ± 2	60 ± 3	45 ± 3	57 ± 3	51 ± 3	76 ± 3	41 ± 3	72 ± 3

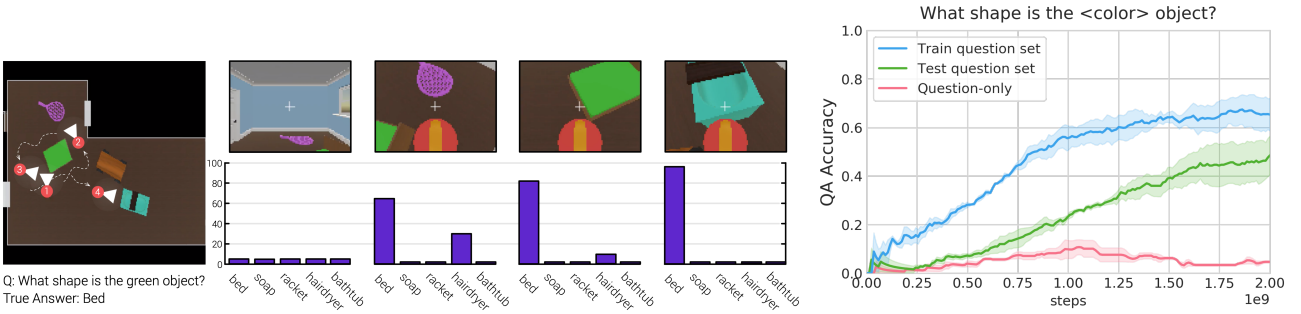


Figure 4. (Left): Sample trajectory ($1 \rightarrow 4$) and QA decoding predictions (for top 5 most probable answers) for the 'What shape is the green object?' from SimCore. Note that top-down map is not available to the agent. (Right): QA accuracy on disjoint train and test splits.

on most question types.

- **SimCore representations achieve higher QA accuracy than other approaches**, substantially above the question-only baseline on `count_color` (57% *vs.* 24%), `near_shape` (30% *vs.* 4%) and `near_color` (59% *vs.* 9%), demonstrating a strong tendency for encoding and retaining information about object identities, properties, and both spatial and temporal relations.

Finally, as before, the No SG agent trained to answer questions without stopped gradients achieves highest accuracy for most questions, although not all – perhaps due to trade-offs between simultaneously optimizing performance for different QA losses and the exploration task.

5.2. Compositional Generalization

While there is a high degree of procedural randomization in our environment and QA tasks, overparameterized neural-network-based models in limited environments are always prone to overfitting or rote memorization. We therefore constructed a test of the generality of the information encoded in the internal state of an agent. The test involves a variant of the `shape` question type (*i.e.* questions like "what shape is the <color> object?"), but in which the possible question-answer pairs are partitioned into mutually exclusive training and test splits. Specifically, the test questions are constrained such that they are compositionally novel – the <color, shape> combination involved in the question-answer pair is never observed during training, but both attributes are observed in other contexts. For instance, a test question-answer pair "Q: what shape is the **blue** object?, A: **table**" is excluded from the training set of the QA decoder, but "Q: what shape is the **blue** object?, A: **car**" and "Q: What shape is the **green** object?, A: **table**" are part of the training set (but not the test set).

We evaluate the SimCore agent on this test of generalization (since other agents perform poorly on the original task). Figure 4 (right) shows that the QA decoder applied to SimCore's internal states performs at substantially above-chance (and all baselines) on the held-out test questions (although somewhat lower than training performance). This indicates that the QA decoder extracts and applies information in a comparatively factorized (or compositional) manner, and suggests (circumstantially) that the knowledge acquired by the SimCore agent may also be represented in this way.

5.3. Robustness of the results

To check if our results are robust to the choice of environment, we developed a similar setup using the DeepMind Lab environment (Beattie et al., 2016) and ran the same experiments *without* any change in hyperparameters.

The environment consists of a rectangular room that is populated with a random selection of objects of different shapes and colors in each episode. There are 6 distinct objects in each room, selected from a pool of 20 objects and 9 different colors. We use a similar exploration reward structure as in our earlier environment to train the agents to navigate and observe all objects. Finally, in each episode, we introduce a question of the form 'What is the color of the <shape>?' where <shape> is replaced by the name of an object present in the room.

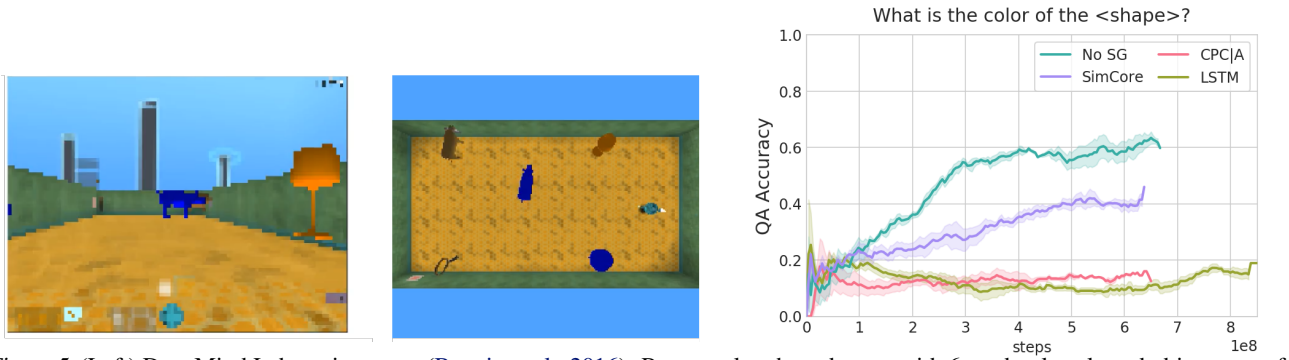


Figure 5. (Left) DeepMind Lab environment (Beattie et al., 2016): Rectangular-shaped room with 6 randomly selected objects out of a pool of 20 different objects of different colors. (Right) QA accuracy for `color` questions (What is the color of the `<shape>`?) in DeepMind Lab. Consistent with results in the main paper, internal representations of the SimCore agent lead to the highest accuracy while CPC|A and LSTM perform worse and similar to each other.

Figure 5 shows question-answering accuracies in the DeepMind Lab environment. Consistent with the results presented above, internal representations of the SimCore agent lead to the highest answering accuracy while CPC|A and the vanilla LSTM agent perform worse and similar to each other. Crucially, for running experiments in DeepMind Lab, we *did not* change any hyperparameters from the experimental setup described before. This demonstrates that our approach is not specific to a single environment and that it can be readily applied in a variety of settings.

6. Discussion

Developing agents with world models of their environments is an important problem in AI. To do so, we need tools to evaluate and diagnose the internal representations forming these world models in addition to studying task performance. Here, we marry together population or glass-box decoding techniques with a question-answering paradigm to discover how much propositional (or declarative) knowledge agents acquire as they explore their environment.

We started by developing a range of question-answering tasks in a visually-rich 3D environment, serving as a diagnostic test of an agent's scene understanding, visual reasoning, and memory skills. Next, we trained agents to optimize an exploration objective with and without auxiliary self-supervised predictive losses, and evaluated the representations they form as they explore an environment, via this question-answering testbed. We compared model-free RL agents alongside agents that make egocentric visual predictions and found that the latter (in particular SimCore (Gregor et al., 2019)) are able to reliably capture detailed propositional knowledge in their internal states, which can be decoded as answers to questions, while non-predictive agents do not, even if they optimize the exploration objective well.

Interestingly, not all predictive agents are equally good at acquiring knowledge of objects, relations and quantities. We compared a model learning the probability distribution of future frames in pixel space via a generative model (SimCore (Gregor et al., 2019)) with a model based on discriminating frames through contrastive estimation (CPC|A (Guo et al., 2018)). We found that while both learned to navigate well, only the former developed representations that could be used for answering questions about the environment. (Gregor et al., 2019) previously showed that the choice of predictive model has a significant impact on the ability to decode an agent's position and top-down map reconstructions of the environment from its internal representations. Our experiments extend this result to decoding factual knowledge, and demonstrate that the question-answering approach has utility for comparing agents.

Finally, the fact that we can even decode answers to questions from an agent's internal representations learned solely from egocentric future predictions, without exposing the agent itself directly to knowledge in propositional form, is encouraging. It indicates that the agent is learning to form and maintain invariant object identities and properties (modulo limitations in decoder capacity) in its internal state *without explicit supervision*.

It is ∼30 years since (Elman, 1990) showed how syntactic structures and semantic organization can emerge in the units of a neural network as a consequence of the simple objective of predicting the next word in a sequence. This work corroborates Elman's findings, showing that language-relevant general knowledge can emerge in a situated neural-network agent that predicts future low-level visual observations via sufficiently powerful generative mechanism. The result also aligns with perspectives that emphasize the importance of between sensory modalities in supporting the development of conceptual or linguistic knowledge (McClelland et al., 2019). Our study is a small example of how language can be used as a channel to probe and understand what exactly agents can learn from their environments. We hope it motivates future research in evaluating predictive agents using natural linguistic interactions.

7. Acknowledgments

We are grateful to Neil Rabinowitz, Matt Botvinick, Tim Lillicrap, Chris Dyer, Tim Harley, Aida Nematzadeh, Lucas Smaira, and Devi Parikh for fruitful discussions and valuable feedback.

References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and van den Hengel, A. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

Andreas, J., Klein, D., and Levine, S. Modular multitask reinforcement learning with policy sketches. In *ICML*, 2017.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., and Parikh, D. VQA: Visual Question Answering. In *ICCV*, 2015.

Atal, B. S. and Schroeder, M. R. Adaptive predictive coding of speech signals. *Bell System Technical Journal*, 1970.

Azar, M. G., Piot, B., Pires, B. A., Grill, J.-B., Altché, F., and Munos, R. World discovery models. *arXiv preprint arXiv:1902.07685*, 2019.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. Deepmind lab. *CoRR*, abs/1612.03801, 2016. URL http://arxiv.org/abs/1612.03801.

Bialek, W., Rieke, F., Van Steveninck, R. D. R., and Warland, D. Reading a neural code. *Science*, 252(5014): 1854–1857, 1991.

Cangea, C., Belilovsky, E., Liò, P., and Courville, A. Videonavqa: Bridging the gap between visual and embodied question answering. *arXiv preprint arXiv:1908.04950*, 2019.

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. Gated-attention architectures for task-oriented language grounding. In *AAAI*, 2018.

Clark, A. *Surfing uncertainty*. Oxford University Press, Oxford, 2016.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. What you can cram into a single $&# vector: Probing sentence embeddings for linguistic properties. In *Proceedings of ACL*, 2018.

Das, A., Kottur, S., Gupta, K., Singh, A., Yadav, D., Moura, J. M., Parikh, D., and Batra, D. Visual Dialog. In *CVPR*, 2017.

Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. Embodied Question Answering. In *CVPR*, 2018a.

Das, A., Gkioxari, G., Lee, S., Parikh, D., and Batra, D. Neural Modular Control for Embodied Question Answering. In *CORL*, 2018b.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 1993.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Dretske, F. Meaningful perception. *An Invitation to Cognitive Science: Visual Cognition,*, pp. 331–352, 1995.

Elias, P. Predictive coding – I. *IRE Transactions on Information Theory*, 1955.

Elman, J. L. Finding structure in time. *Cognitive science*, 1990.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.

Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., and Farhadi, A. IQA: Visual Question Answering in Interactive Environments. In *CVPR*, 2018.

Gregor, K., Besse, F., Rezende, D. J., Danihelka, I., and Wierstra, D. Towards conceptual compression. In *NeurIPS*, 2016.

Gregor, K., Rezende, D. J., Besse, F., Wu, Y., Merzic, H., and Oord, A. v. d. Shaping Belief States with Generative Environment Models for RL. In *NeurIPS*, 2019.

Guo, Z. D., Azar, M. G., Piot, B., Pires, B. A., Pohlen, T., and Munos, R. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018.

Gupta, S., Davidson, J., Levine, S., Sukthankar, R., and Malik, J. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W., Jaderberg, M., Teplyashin, D., et al. Grounded language learning in a simulated 3D world. *arXiv preprint arXiv:1706.06551*, 2017.

Hill, F., Mokra, S., Wong, N., and Harley, T. Human instruction-following with deep reinforcement learning via transfer-learning from text, 2020.

Hohwy, J. *The predictive mind*. Oxford University Press, Oxford, 2013.

Huth, A. G., Lee, T., Nishimoto, S., Bilenko, N. Y., Vu, A. T., and Gallant, J. L. Decoding the semantic content of natural movies from human brain activity. *Frontiers in systems neuroscience*, 2016.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. Skip-thought vectors. In *NIPS*, 2015.

Malinowski, M. and Fritz, M. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In *NIPS*, 2014.

Matuszek, C., Herbst, E., Zettlemoyer, L., and Fox, D. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, 2013.

McClelland, J. L., Hill, F., Rudolph, M., Baldridge, J., and Schtze, H. Extending machine language models toward human-level language understanding, 2019.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Misra, D., Langford, J., and Artzi, Y. Mapping instructions and visual observations to actions with reinforcement learning. In *ACL*, 2017.

Narasimhan, K., Kulkarni, T., and Barzilay, R. Language understanding for text-based games using deep reinforcement learning. In *EMNLP*, 2015.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Nortmann, N., Rekauzke, S., Onat, S., König, P., and Jancke, D. Primary visual cortex represents the difference between past and present. *Cerebral Cortex*, 2013.

Oh, J., Singh, S., Lee, H., and Kohli, P. Zero-shot task generalization with multi-task deep reinforcement learning. In *ICML*, 2017.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., and Torralba, A. Virtualhome: Simulating household activities via programs. In *CVPR*, 2018.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*, 2016.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018.

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 1999.

Recanatesi, S., Farrell, M., Lajoie, G., Deneve, S., Rigotti, M., and Shea-Brown, E. Predictive learning extracts latent space representations from sensory observations. *bioRxiv*, 2019.

Rezende, D. J. and Viola, F. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

Roskies, A. and Wood, C. Catching the prediction wave in brain science. *Analysis*, 77:848–857, 2017.

Salinas, E. and Abbott, L. Vector reconstruction from firing rates. *Journal of computational neuroscience*, 1(1-2): 89–107, 1994.

Schaul, T. and Ring, M. Better generalization with forecasts. In *IJCAI*, 2013.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *ICML*, 2015.

Schmidhuber, J. Curious model-building control systems. In *IJCNN*, 1991.

Seth, A. K. The cybernetic bayesian brain: From interoceptive inference to sensorimotor contingencies. In Windt, T. M. . J. M. (ed.), *Open MIND: 35(T)*. MIND Group, Frankfurt am Main, 2015.

Shu, T., Xiong, C., and Socher, R. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. In *ICLR*, 2018.

Silver, D., van Hasselt, H., Hessel, M., Schaul, T., Guez, A., Harley, T., Dulac-Arnold, G., Reichert, D., Rabinowitz, N., Barreto, A., et al. The predictron: End-to-end learning and planning. In *ICML*, 2017.

Stich, S. P. Do animals have beliefs? *Australasian Journal of Philosophy*, 57(1):15–28, 1979.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S., Das, D., and Pavlick, E. What do you learn from context? probing for sentence structure in contextualized word representations. In *ICLR*, 2019.

Thomason, J., Gordan, D., and Bisk, Y. Shifting the baseline: Single modality performance on visual navigation & QA. In *NAACL*, 2019.

Truncellito, D. Epistemology. internet encyclopedia of philosophy, 2007.

Vogel, A. and Jurafsky, D. Learning to follow navigational directions. In *ACL*, 2010.

Wang, X. and Gupta, A. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

Wayne, G., Hung, C.-C., Amos, D., Mirza, M., Ahuja, A., Grabska-Barwinska, A., Rae, J., Mirowski, P., Leibo, J. Z., Santoro, A., et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018.

Wijmans, E., Datta, S., Maksymets, O., Das, A., Gkioxari, G., Lee, S., Essa, I., Parikh, D., and Batra, D. Embodied Question Answering in Photorealistic Environments with Point Cloud Perception. In *CVPR*, 2019.

Williams, D. Predictive coding and thought. *Synthese*, pp. 1–27, 2018.

Winograd, T. Understanding natural language. *Cognitive Psychology*, 1972.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.

Yu, H., Zhang, H., and Xu, W. Interactive Grounded Language Acquisition and Generalization in a 2D World. In *ICLR*, 2018.

Yu, L., Chen, X., Gkioxari, G., Bansal, M., Berg, T. L., and Batra, D. Multi-target embodied question answering. In *CVPR*, 2019.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *ECCV*, 2016.

Zhu, Y., Gordon, D., Kolve, E., Fox, D., Fei-Fei, L., Gupta, A., Mottaghi, R., and Farhadi, A. Visual Semantic Planning using Deep Successor Representations. In *ICCV*, 2017a.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017b.

A. Appendix

A.1. Network architectures and Training setup

A.1.1. Importance Weighted Actor-Learner Architecture

Agents were trained using the IMPALA framework (Espeholt et al., 2018). Briefly, there are N parallel 'actors' collecting experience from the environment in a replay buffer and one learner taking batches of trajectories and performing the learning updates. During one learning update the agent network is unrolled, all the losses (RL and auxiliary ones) are evaluated and the gradients computed.

A.1.2. Agents

Input encoder To process the frame input, all models in this work use a residual network (He et al., 2016) of 6 64-channel ResNet blocks with rectified linear activation functions and bottleneck channel of size 32. We use strides of (2, 1, 2, 1, 2, 1) and don't use batch-norm. Following the convnet we flatten the ouput and use a linear layer to reduce the size to 500 dimensions. Finally, we concatenate this encoding of the frame together with a one hot encoding of the previous action and the previous reward.

Core architecture The recurrent core of all agents is a 2-layer LSTM with 256 hidden units per layer. At each time step this core consumes the input embedding described above and updates its state. We then use a 200 units single layer MLP to compute a value baseline and an equivalent network to compute action logits, from where one discrete action is sampled.

Simulation Network Both predictive agents have a simulation network with the same architecture as the agent's core. This network is initialized with the agent state at some random time t from the trajectory and unrolled forward for a random number of steps up to 16, receiving only the actions of the agent as inputs. We then use the resulting LSTM hidden state as conditional input for the prediction loss (SimCore or CPC|A).

SimCore We use the same architecture and hyperparameters described in (Gregor et al., 2019). The output of the simulation network is used to condition a Convolutional DRAW (Gregor et al., 2016). This is a conditional deep variational auto-encoder with recurrent encoder and decoder using convolutional operations and a canvas that accumulates the results at each step to compute the distribution over inputs. It features a recurrent prior network that receives the conditioning vector and computes a prior over the latent variables. See more details in (Gregor et al., 2019).

Action-conditional CPC We replicate the architecture used in (Guo et al., 2018). CPC|A uses the output of the simulation network as input to an MLP that is trained to discriminate true versus false future frame embedding. Specifically, the simulation network outputs a conditioning vector after k simulation steps which is concatenated with the frame embedding z_{t+k} produced by the image encoder on the frame x_{t+k} and sent through the MLP discriminator. The discriminator has one hidden layer of 512 units, ReLU activations and a linear output of size 1 which is trained to binary classify true embeddings into one class and false embeddings into another. We take the negative examples from random time points in the same batch of trajectories.

A.1.3. QA network architecture

Question encoding The question string is first tokenized to words and then mapped to integers corresponding to vocabulary indices. These are then used to lookup 32-dimensional embeddings for each word. We then unroll a 64-units single-layer LSTM for a fixed number of 15 steps. The language representation is then computed by summing the hidden states for all time steps.

QA decoder. To decode answers from the internal state of the agents we use a second LSTM initialized with the internal state of the agent's LSTM and unroll it for a fixed number of steps, consuming the question embedding at each step. The results reported in the main section were computed using 12 decoding steps. The terminal state is sent through a two-layer MLP (sizes 256, 256) to compute a vector of answer logits with the size of the vocabulary and output the top-1 answer.

A.1.4. Hyper-parameters

The hyper-parameter values used in all the experiments are in Table 3.

A.1.5. Negative sampling strategies for CPC|A

We experimented with multiple sampling strategies for the CPC|A agent (whether or not negative examples are sampled from the same trajectory, the number of contrastive prediction steps, the number of negative examples). We report the best results in the main text. The CPC|A agent did provide better representations of the environment than the LSTM-based agent, as shown by the top-down view reconstruction loss (Figure 6a). However, none of the CPC|A agent variations that we tried led to better-than-chance question-answering accuracy. As an example, in Figure 6b we compare sampling negatives from the same trajectory or from any trajectory in the training batch.

A.2. Effect of QA network depth

To study the effect of the QA network capacity on the answer accuracy, we tested decoders of different depths applied to both the SimCore and the LSTM agent's internal represen-

Agent	
Learning rate	1e-4
Unroll length	50
Adam β_1	0.90
Adam β_2	0.95
Policy entropy regularization	0.0003
Discount factor	0.99
No. of ResNet blocks	6
No. of channel in ResNet block	64
Frame embedding size	500
No. of LSTM layers	2
No. of units per LSTM layer	256
No. of units in value MLP	200
No. of units in policy MLP	200
Simulation Network	
Overshoot length	16
No. of LSTM layers	2
No. of units per LSTM layer	256
No. of simulations per trajectory	6
No. of evaluations per overshoot	2
SimCore	
No. of ConvDRAW Steps	8
GECO kappa	0.0015
CPC|A	
MLP discriminator size	64
QA network	
Vocabulary size	1000
Maximum question length	15
No. of units in Text LSTM encoder	64
Question embedding size	32
No. of LSTM layers in question decoder	2
No. of units per LSTM layer	256
No. of units in question decoder MLP	200
No. of decoding steps	12

Table 3. Hyperparameters.

tations (7). The QA network is an LSTM initialized with the agent's internal state that we unroll for a fixed number of steps feeding the question as input at each step. We found that, indeed, the answering accuracy increased with the number of unroll steps from 1 to 12, while greater number of steps became detrimental. We performed the same analysis on the LSTM agent and found that regardless of the capacity of the QA network, we could not decode the correct answer from its internal state, suggesting that the limiting factor is not the capacity of the decoder but the lack of useful representations in the LSTM agent state.

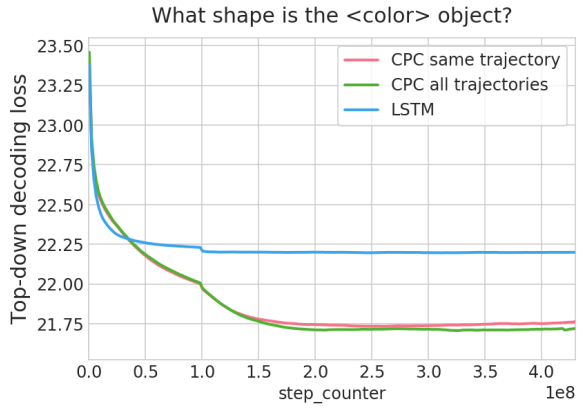
A.3. Answering accuracy during training for all questions

The QA accuracy over training for all questions is shown in Figure 8.

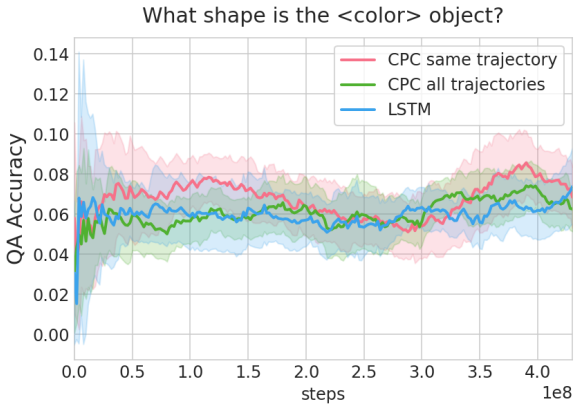
A.4. Environment

Our environment is a single L-shaped 3D room, procedurally populated with an assortment of objects.

Actions and Observations. The environment is episodic, and runs at 30 frames per second. Each episode takes 30 seconds (or 900 steps). At each step, the environment provides the agent with two observations: a 96x72 RGB image with the first-person view of the agent and the text containing the



(a) To test whether the CPC|A loss provided improved representations we reconstructed the environment top-down view, similar to (Gregor et al., 2019). Indeed the reconstruction loss is lower for CPC|A than for the LSTM agent.



(b) QA accuracy for the CPC|A agent is not better than the LSTM agent, for both sampling strategies of negatives.

Figure 6.

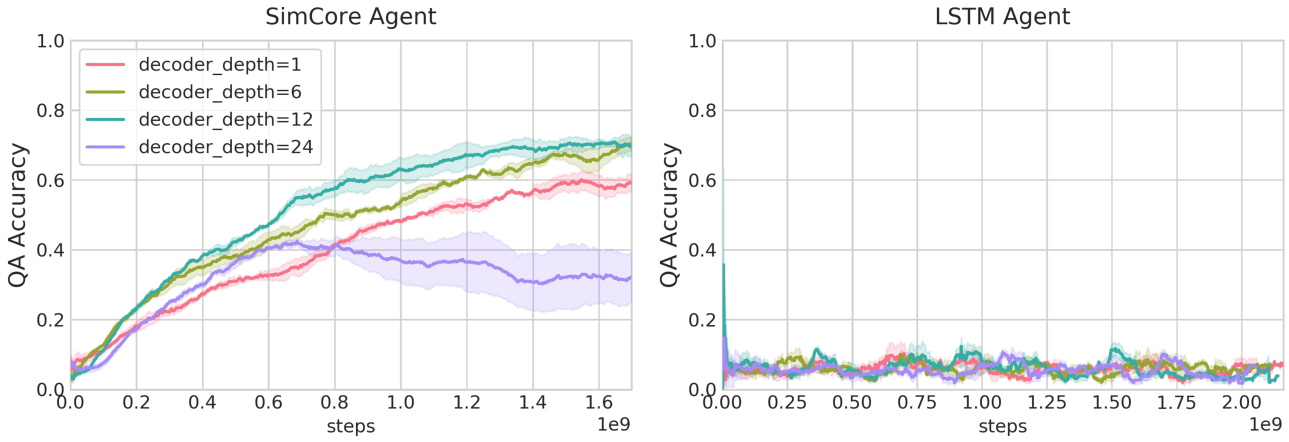


Figure 7. Answer accuracy over training for increasing QA decoder's depths. Left subplot shows the results for the SimCore agent and right subplot for the LSTM baseline. For SimCore, the QA accuracy increases with the decoder depth, up to 12 layers. For the LSTM agent, QA accuracy is not better than chance regardless of the capacity of the QA network.

question.

The agent can interact with the environment by providing multiple simultaneous actions to control movement (forward/back, left/right), looking (up/down, left/right), picking up and manipulating objects (4 degrees of freedom: yaw, pitch, roll + movement along the axis between agent and object).

Rewards. To allow training using cross-entropy, as described in Section 4, the environment provides the ground-truth answer instead of the reward to the agent.

Object creation and placement. We generate between 2 and 20 objects, depending on the task, with the type of the object, its color and size being uniformly sampled from the set described in Table 4.

Objects will be placed in a random location and random orientation. For some tasks, we required some additional constraints - for example, if the question is "What is the color of the cushion near the bed?", we need to ensure only one cushion is close to the bed. This was done by checking the constraints and regenerating the placement in case they were not satisfied.

Attribute	Options
Object	basketball, cushion, carriage, train, grinder, candle, teddy, chair, scissors, stool, book, football, rubber duck, glass, toothpaste, arm chair, robot, hairdryer, cube block, bathtub, TV, plane, cuboid block, car, tv cabinet, plate, soap, rocket, dining table, pillar block, potted plant, boat, tennisball, tape dispenser, pencil, wash basin, vase, picture frame, bottle, bed, helicopter, napkin, table lamp, wardrobe, racket, keyboard, chest, bus, roof block, toilet
Color	aquamarine, blue, green, magenta, orange, purple, pink, red, white, yellow
Size	small, medium, large

Table 4. Randomization of objects in the Unity room. 50 different types, 10 different colors and 3 different scales.

Body movement actions	**Movement and grip actions**	**Object manipulation**
NOOP	GRAB	GRAB + SPIN_OBJECT_RIGHT
MOVE_FORWARD	GRAB + MOVE_FORWARD	GRAB + SPIN_OBJECT_LEFT
MOVE_BACKWARD	GRAB + MOVE_BACKWARD	GRAB + SPIN_OBJECT_UP
MOVE_RIGHT	GRAB + MOVE_RIGHT	GRAB + SPIN_OBJECT_DOWN
MOVE_LEFT	GRAB + MOVE_BACKWARD	GRAB + SPIN_OBJECT_FORWARD
LOOK_RIGHT	GRAB + LOOK_RIGHT	GRAB + SPIN_OBJECT_BACKWARD
LOOK_LEFT	GRAB + LOOK_LEFT	GRAB + PUSH_OBJECT_AWAY
LOOK_UP	GRAB + LOOK_UP	GRAB + PULL_OBJECT_CLOSE
LOOK_DOWN	GRAB + LOOK_DOWN	

Table 5. Environment action set.

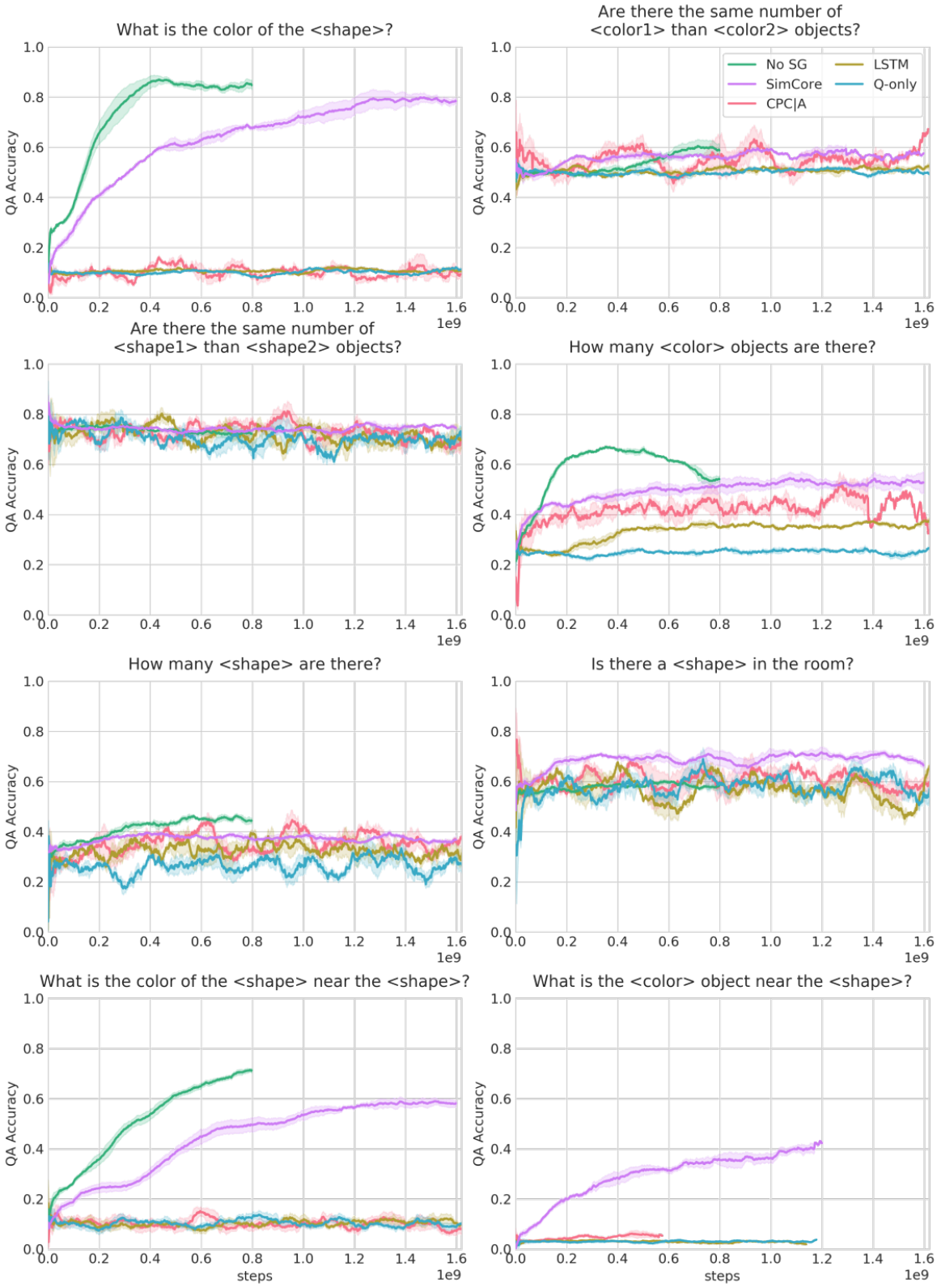


Figure 8. QA accuracy over training for all questions and all models.