VideoNavQA: Bridging the Gap between Visual and Embodied Question Answering

Cătălina Cangea*
Department of Computer Science and Technology
University of Cambridge
Catalina.Cangea@cst.cam.ac.uk

Eugene Belilovsky
Mila, Université de Montréal
eugene.belilovsky@umontreal.ca

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
Pietro.Lio@cst.cam.ac.uk

Aaron Courville
Mila, Université de Montréal
CIFAR Fellow
aaron.courville@umontreal.ca

Abstract

Embodied Question Answering (EQA) is a recently proposed task, where an agent is placed in a rich 3D environment and must act based solely on its egocentric input to answer a given question. The desired outcome is that the agent learns to combine capabilities such as scene understanding, navigation and language understanding in order to perform complex reasoning in the visual world. However, initial advancements combining standard vision and language methods with imitation and reinforcement learning algorithms have shown EQA might be too complex and challenging for these techniques. In order to investigate the feasibility of EQA-type tasks, we build the VideoNavQA dataset that contains pairs of questions and videos generated in the House3D environment. The goal of this dataset is to assess question-answering performance from nearly-ideal navigation paths, while considering a much more complete variety of questions than current instantiations of the EQA task. We investigate several models, adapted from popular VQA methods, on our benchmark. This establishes an initial understanding of how well VQA-style methods can perform within the novel EQA paradigm.

1 Introduction

The Embodied Question Answering [7] (EQA) and Interactive Question Answering [11] (IQA) tasks were introduced as a means to study the capabilities of agents in rich, realistic environments, requiring both *navigation and reasoning* to achieve success. Each of these skills typically needs a different approach [22, 13, 14] that should nevertheless be smoothly integrated with the rest of the system leveraged by the agent. These abilities are assessed via placing the agent at a random location in a house environment and asking it a question—successful completion of the task involves the agent knowledgeably exploring the environment and reasoning about visual stimuli.

Initial attempts at solving the EQA task [7, 8] have thus combined typically used vision (convolutional neural networks for object detection) and language (question encoding or program generation) techniques with imitation learning and reinforcement learning. However, these approaches either suffer from potentially weaker performance than when using a language-only model [1] or are preceded by additional hand-engineered steps (manually defined sub-goals, imitation learning on pre-computed expert trajectories). Indeed, even for simple single object questions, the agent is often unable to advance meaningfully towards the target [8], producing visual streams that cannot be used

*Work partially carried out while the author was a research intern at Mila.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Figure 1: High-level description of the VideoNavQA task and our approach: the VQA system receives an example containing a video of a trajectory inside the house environment and a question, processes the input and produces an answer.

to answer the query. This suggests that EQA is a highly challenging task which might not be easily approached from this angle—further investigation is required to realistically assess the gap between current state-of-the-art and desired human level performance on the EQA performance.

Single-image Visual Question Answering is only now starting to tackle complex reasoning questions, even in limited settings [14, 17]—it is unclear if existing methods can handle the rich video streams produced in our task. On the other hand, EQA introduces a navigation component, which can often lead to video inputs which are uninformative with respect to the question. A natural issue arises: can the desired tasks be solved with current methods, if we assume the agent is given correct visual streams (ones that can be used to provide an answer)? In our attempt to answer this question, we propose a dataset that decouples the visual reasoning from the navigation aspects in the EQA problem. We introduce the VideoNavQA task, illustrated in Figure 1. While removing the navigation and action selection requirements from EQA, we increase the difficulty of the visual reasoning component via a much larger question space, tackling the sort of complex reasoning questions that make QA tasks challenging [17]. By designing and evaluating several VQA-style models on the dataset[2], we *establish a novel way of evaluating EQA feasibility* given existing methods, while highlighting the difficulty of the problem even in the most ideal setting.

2 Related work

To the best of our knowledge, there are no previous works that position themselves at the intersection of these two paradigms. Instead, VQA and EQA have been tackled from separate angles, with multiple interesting developments in each case.

Das [7] proposed the **EQA**-v1 dataset containing 4 types of questions (`location`, `color`, `color_room`, `preposition`) that always refer to a single object, the navigation goal of the agent. They train a model using imitation and reinforcement learning, while revealing that RL finetuning often results in overshooting the goal. A subsequent improvement is achieved via hierarchical policy learning with neural modular control [8]—however, this approach uses hand-crafted sub-policies. Anand [1] study the performance of question-only baselines on EQA-v1, which give better results when the agent is spawned more than 10 steps away, concluding that existing EQA methods struggle to exploit (and are often impeded by) the environment. A more recent extension of EQA, multi-target EQA [36], requires reaching multiple goals to answer the question (for example, comparing the sizes of two objects in different rooms). The authors build the MT-EQA dataset containing 6 types of questions and tackle the task by first decomposing each question into small sub-goals, which are more easily achievable by a model similar to the one used in previous works. Another EQA variant based on photorealistic environments [33] uses point clouds instead of RGB input; the authors "port" three of the EQA-v1 questions to Matterport3D. Although in this work we focus on the House3D environment and EQA, we also note that the IQA task [11] for the AI2Thor [19] environment was introduced with similar goals. However, this task is defined in single room settings, hence the negative effect of poor navigation is less severe.

Visual question answering has been extensively tackled over the past few years, with numerous datasets being released, various algorithms designed and general studies carried out [21, 4]. The VQA dataset [2] is one such example, containing free-form and open-ended questions about real-life

[2]The code and dataset are available at `https://github.com/catalina17/VideoNavQA/`.

images and abstract scenes that require natural language answers. CLEVR [17] uses a functional program-based question representation to generate a vast range of questions from synthetic scene graphs that contain 3–10 objects with restricted variability (3 types, 2 sizes, 2 materials, 8 colors). More recently, the GQA dataset [15] has been proposed to address some of the issues of current datasets, including biases in the answer distribution. Widely used VQA models include *stacked attention networks* [35], which use the question embedding as a query for attending over the visual input, *multimodal compact bilinear pooling* [9], which fuses the text and visual embeddings via multiplying them in the Fourier space, *feature-wise linear modulation* [27], which selects informative visual feature maps via question-based conditioning, its *multi-hop* extension [30], which conditions feature maps by iteratively attending over the language input, and *compositional attention networks* (MAC) [14], which can reason about the visual input in a multi-step fashion using the memory-attention-control cell. Graph neural network-based approaches have also been used for VQA tasks lately [26, 24, 32], operating on the relational representation of objects in the image. The model that most closely resembles the ones we propose is explored in a multi-turn QA setting [25], where the system is provided with a dialogue (set of question-answer pairs) and a video—the question encoding is used to perform both per-frame conditioning and attention over the hidden states of the LSTM which encodes all the video frames.

Several datasets have been proposed that also consider **video question answering** in settings such as movies [31, 23, 20]. They are often provided with rich per-frame annotations (for example, subtitles in movies). However, these tasks largely focus on identifying actions or other dynamic behaviors and are distant from the aims of VideoNavQA. Our task considers indoor navigation trajectories that exhibit rich visual data at each time step. This poses an additional challenge for video QA systems—these are now required to *isolate the relevant information from a large pool of visual concepts* present in each frame and *perform more advanced reasoning to answer the extensive variety of questions* designed.

	Houses	Samples
Train	622	84990
Validation	65	8755
Test	56	7587

Table 1: Dataset split statistics. The three sets of environments are disjoint.

EQAv1 (Q types: 4)	What room is the <OBJ> located in? What color is the <OBJ> in the <ROOM>?
VideoNavQA (Q types: 28)	Are both <attr1> <OBJ1> and <attr2> <OBJ2> <color>? How many <attr> <OBJ> are in the <ROOM>? Is there <art> <attr> <OBJ>?

Table 2: Some examples of the type of question templates found in EQAv1 versus the more complex existence, counting and comparison questions found in VideoNavQA.

3 The VideoNavQA dataset

Our new benchmark aims to study the capabilities of VQA approaches in a variant of the EQA task, where the agent is required to answer a question while having access to a near-optimal trajectory. This task can be seen as complementary to the Habitat Challenge [29], where the focus is on navigation instead of question answering. We use the House3D virtual environment [34] to generate approximately 101,000 pairs of videos and questions; the dataset contains 28 types of questions belonging to 8 categories (see Figure 2), with 70 possible answers. The complexity of our questions is far beyond that of similar tasks using the same generation method (such as CLEVR)[3].

Video generation and ground truth extraction We use the environment grid to compute shortest paths between arbitrary locations in two different rooms, then render the trajectories in House3D. The ground truth information is obtained by indexing the objects and rooms that are seen in each frame.

Question generation Similarly to other synthetic benchmark datasets (EQA-v1 [7], CLEVR [17]), we generate questions according to functional, template-style representations (e.g. *"How many <attr> <obj_type-pl> are in the <room_type>?"*). First, we build sets of possible values for each tag. For example, if the current template contains a *<room_type>* tag and we have only seen a kitchen and a living room on our trajectory, then the set of possible instantiations is {*kitchen*, *living room*}. To generate a question, we then randomly assign each tag a value from its set. Finally, we run the functional program to compute whether the question is valid and can be answered from ground truth.

[3]The full list of question types and dataset counts can be found in the appendix.

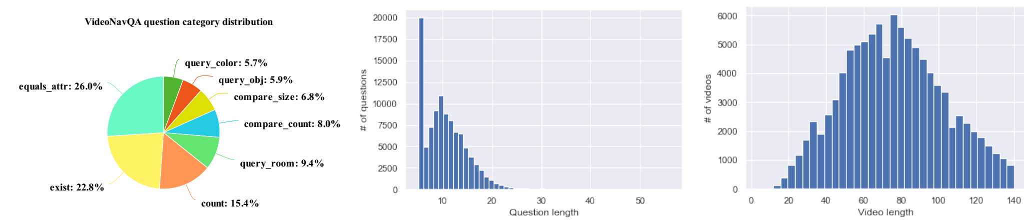

Figure 2: VideoNavQA dataset distribution. (Left:) Proportions for each question category. (Middle:) Question lengths (maximum is 56). (Right:) Video lengths (maximum is 140).

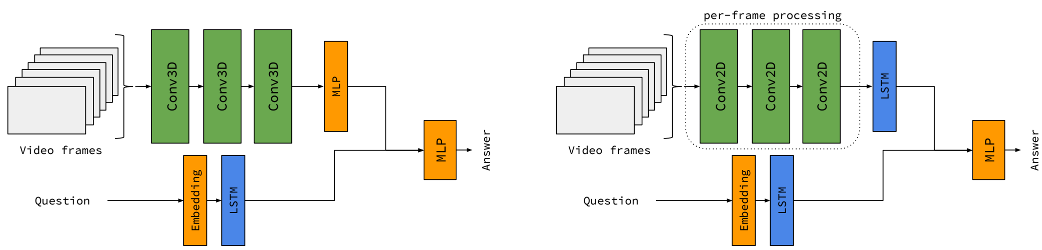

Figure 3: (Left:) Concat-CNN3D processes the entire video. (Right:) Concat-CNN2D aggregates frame features via an LSTM. Both merge the result with the question embedding.

4 Models

We have designed the VideoNavQA task to address the EQA challenge from an alternative perspective, requiring a smaller degree of fusion among different classes of methods. In this section, we describe the architectures used to establish more realistic expectations on EQA performance: several essential baselines and new models inspired by previous successes in VQA and computer vision.

Language models We evaluate two simple yet powerful baselines: a *1-layer LSTM* [12] and a *bag-of-words (BoW)* [28]. These reveal the inherent biases in the environment distribution and place a lower bound on the desired performance of models that can usefully exploit visual information.

Concatenation models We integrate the LSTM model with two vision models (VGG-like and C3D-like architectures), in order to derive a joint representation of the obtained features. We achieve this by concatenating the final question representation with their respective video representations (Concat-CNN2D/3D; see Figure 3). An MLP then obtains the most likely answer via softmax.

FiLM-based reasoning Feature-wise linear modulation (FiLM) [27] directly uses the question embedding to scale and shift the feature maps of a CNN pipeline taking the image as input. Here, we extend FiLM as per Figure 4: each frame is conditioned independently by a fixed number of ResBlocks [27]. We obtain a series of visual features and aggregate them using either (a) *attention* (FiLM-AT), where we apply a linear transformation to the features from each frame, then use recurrent attention [3, 6] to obtain a final encoding, or (b) *global max-pooling* (FiLM-GP), where we apply a 1×1 convolution, followed by feature-wise max-pooling over all frames.

Temporal multi-hop FiLM The multi-hop extension of FiLM [30] modulates CNN feature maps by attending over the hidden states of the question encoder; the attention mechanism is initialized with the context vector from the previous CNN depth level. We propose a *temporal multi-hop* approach that computes FiLM parameters for all ResBlocks at frame t by initializing the attention context with the one from $t-1$, attending over question encoder hidden states, obtaining the new context vector and passing it through a linear layer. The FiLM parameters at a certain time step are thus based on what has already been computed for previous frames, modelling the temporal structure of the input.

MAC Compositional Attention Networks (MAC) [14] have achieved excellent performance in several VQA tasks [14, 15, 4]. Here we adapt this by applying a MAC model after a 2D-CNN and integrating the resulting frame embeddings over time using an LSTM.

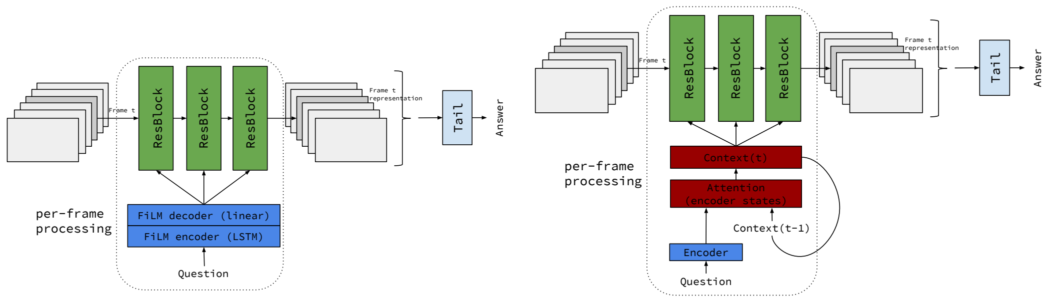

Figure 4: (Left:) FiLM across time. Frames are processed by the ResBlocks, then resulting features are aggregated to answer the question. (Right:) Temporal Multi-hop. The FiLM parameters are computed from the current attention context, which is initialized with the one from the previous frame. Temporal summarization is achieved via global max-pooling.

Model	Accuracy All	Yes/No	Other	Num
BoW	49.02	57.67	30.57	40.21
LSTM	56.49	68.36	35.27	38.90
Concat-CNN3D	64.00	72.99	49.12	49.10
Concat-CNN2D	64.47	73.50	49.20	49.59
FiLM-GP	63.79	72.91	47.71	50.00
FiLM-AT	64.08	72.93	49.54	49.26
Temporal multi-hop	63.53	71.81	49.54	50.16
MAC	62.32	69.02	51.37	50.99

Table 3: Results for all models reported in terms of accuracy on the VideoNavQA test set. We also use standard VQA reporting of Yes/No, other, and number categories [2].

5 Experiments

We evaluate the models described in the previous section on the VideoNavQA task and produce an initial expectation of the feasibility of EQA.[4] According to the results in Table 3, the LSTM proves to be the most powerful language model, achieving 7.5% more than the bag-of-words. Surprisingly, Concat-CNN2D manages to outperform the other models, obtaining an accuracy of 64.47%. Figure 5 provides a more detailed analysis of the model capabilities across the eight VideoNavQA sub-tasks. Overall, the results suggest that the proposed models manage to exploit the visual context of the environment, improving by a significant margin over the language-only baselines. This finding represents an initial validation of the feasibility of the VideoNavQA task, ensuring that the manner in which we constructed the dataset allows the visual reasoning process to be generalizable across new environments. A more focused investigation of novel VQA techniques is further required to accommodate the rich dimensionality and contextual information encountered in the videos.

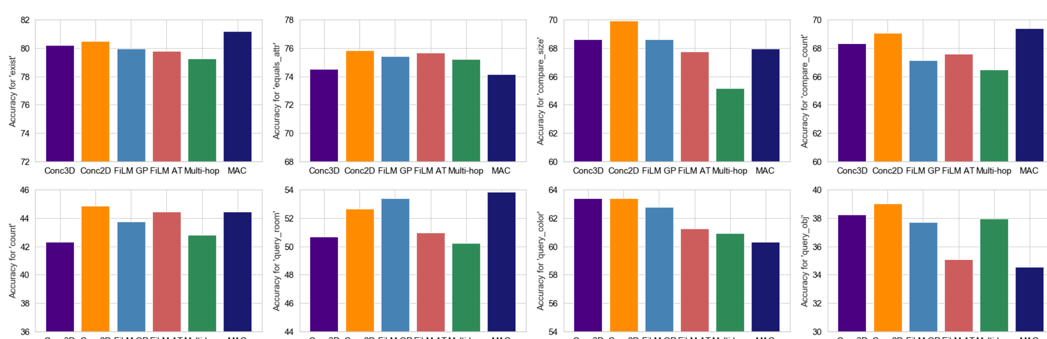

Figure 5: Comparative performance of the models on each question category. Questions from the top row categories have binary answers, whereas the ones from the bottom row are answered by integer counts, room types, colors and object types, respectively.

[4]The detailed model configurations can be found in the appendix.

References

[1] Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. Blindfold Baselines for Embodied QA. *arXiv preprint arXiv:1811.05013*, 2018.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic Generalization: What Is Required and Can It Be Learned? *arXiv preprint arXiv:1811.12889*, 2018.

[5] Remi Cadene, Hedi Ben-Younes, Nicolas Thome, and Matthieu Cord. MUREL: Multimodal Relational Reasoning for Visual Question Answering. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2019.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural Modular Control for Embodied Question Answering. *arXiv preprint arXiv:1810.11181*, 2018.

[9] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[11] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4089–4098, 2018.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813, 2017.

[14] Drew A Hudson and Christopher D Manning. Compositional Attention Networks for Machine Reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

[15] Drew A Hudson and Christopher D Manning. GQA: a new dataset for compositional question answering over real-world images. *arXiv preprint arXiv:1902.09506*, 2019.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[20] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*, 2018.

[21] Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. Learning visual question answering by bootstrapping hard attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–20, 2018.

[22] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking Classic and Learned Navigation in Complex 3D Environments. *arXiv preprint arXiv:1901.10915*, 2019.

[23] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. MarioQA: Answering Questions by Watching Gameplay Videos. In *ICCV*, 2017.

[24] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *Advances in Neural Information Processing Systems*, pages 2654–2665, 2018.

[25] Dat Tien Nguyen, Shikhar Sharma, Hannes Schulz, and Layla El Asri. From FiLM to Video: Multi-turn Question Answering with Multi-modal Context. *arXiv preprint arXiv:1812.07023*, 2018.

[26] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. In *Advances in Neural Information Processing Systems*, pages 8334–8343, 2018.

[27] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[28] Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961, 2015.

[29] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. *arXiv preprint arXiv:1904.01201*, 2019.

[30] Florian Strub, Mathieu Seurin, Ethan Perez, Harm De Vries, Jérémie Mary, Philippe Preux, Aaron Courville, and Olivier Pietquin. Visual Reasoning with Multi-hop Feature Modulation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–800, 2018.

[31] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] Damien Teney, Lingqiao Liu, and Anton van den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2017.

[33] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied Question Answering in Photorealistic Environments with Point Cloud Perception. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3D environment. *arXiv preprint arXiv:1801.02209*, 2018.

[35] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[36] Licheng Yu, Xinlei Chen, Georgia Gkioxari, Mohit Bansal, Tamara L. Berg, and Dhruv Batra. Multi-Target Embodied Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Appendix

Detailed breakdown of question templates and respective counts

- **Equals<attr>**
  'Are all <attr> <obj_type-pl> <color>?': 4014
  'Are all <attr> <obj_type-pl> in the <room_type>?': 3811
  'Are all <attr> things <obj_type-pl>?': 3539
  'Are both the <attr1> <obj_type1> and the <attr2> <obj_type2> <color>?': 3968
  'Are both the <attr1> <obj_type1> and the <attr2> <obj_type2> in the <room_type>?': 3804
  'Are the <attr1> <obj_type1> and the <attr2> <obj_type2> the same color?': 4018
  'Is the <attr1> thing <rel> the <attr2> <obj_type2> <art> <obj_type1>?': 3315
- **Count**
  'How many <attr1> <obj_type1-pl> are in the room containing the <attr2> <obj_type2>?': 3999
  'How many <attr> <obj_type-pl> are in the <room_type>?': 3763
  'How many <obj_type-pl> are <attr>?': 4120
  'How many rooms have <attr> <obj_type-pl>?': 3834
- **Compare<count>**
  'Are there <comp> <attr1> <obj_type1-pl> than <attr2> <obj_type2-pl>?': 4058
  'Are there as many <attr1> <obj_type1-pl> as there are <attr2> <obj_type2-pl>?': 4100
- **Compare<size>**
  'Is the <attr1> <obj_type> <comp_rel> than the <attr2> one?': 3272
  'Is the <room_type1> <comp_rel> than the <room_type2>?': 3148
- **Exist**
  'Is there <art> <attr> <obj_type>?': 4122
  'Is there <art> <room_type>?': 3335
  'Is there a room that has set(<art> <attr{}> <obj_type{}>)?': 3877
  'Is there set(<art> <attr{}> <obj_type{}>) in the <room_type>?': 4025
  'Is there set(<art> <attr{}> <obj_type{}>)?': 4107
  'Is there set(<art> <room_type{}>)?': 3750
- **Query<color>**
  'What color is the <attr1> <obj_type1> <rel> the <attr2> <obj_type2>?': 2178
  'What color is the <attr> <obj_type>?': 3592
- **Query<obj_type>**
  'What is the <attr1> thing <rel> the <attr2> <obj_type2>?': 3119
  'What is the <attr> thing?': 2883
- **Query<room_location>**
  'Where are the set(<attr{}> <obj_type{}>)?': 3816
  'Where is the <attr1> <obj_type1> <rel> the <attr2> <obj_type2>?': 2284
  'Where is the <attr> <obj_type>?': 3481

Evaluation setup

All models are trained with the Adam optimizer [18], by monitoring the validation accuracy.

Language-only models The recurrent model has an embedding layer of 512 units and an LSTM with 128 hidden units, whereas the BoW model uses an embedding size of 128. We use a batch size of 1024 and learning rates of $5e^{-5}$ and $1e^{-5}$, respectively.

Concatenation models The C3D-like sub-network has 3 {convolutional, max-pool, batch normalization (BN)} blocks with 64, 128, and 128 output feature maps, respectively, kernel size $(1, 2, 2)$ for the first block and $(4, 4, 4)$ otherwise. The classifier has 2 linear layers with 2048 and 128 hidden units respectively. BN [16] and ReLU [10] activations are used for each layer. For the VGG-like sub-network, we use a VGG configuration with 5 {convolutional, max-pool} blocks with 16, 32, 64, 128 and 128 channels, respectively, and kernel size 2.

FiLM-extended models The models that process each frame in a FiLM fashion have 4 (GP) or 5 (AT) ResBlocks with 1024 channels, preceded by a 1×1 convolution on the input. The attention mechanism for FiLM AT has 128 hidden units, whereas the global max-pooling classifier obtains 32 channels via the 1×1 convolution. The temporal multi-hop model has 3 ResBlocks and 64 tail channels. It is trained with a learning rate of $5e^{-5}$ and a batch size of 16, whereas the other two use $1e^{-4}$ and 32.

Model hyperparameters

In order to find the best performance on VideoNavQA, we have ran several combinations of hyperparameters for each of the described models. We detail settings evaluated on the validation set below. The average running time per epoch for the visual reasoning models is approximately 5 hours on a 16GB Tesla P100 GPU.

LSTM Embedding size: {128, 256, 512, 1024}. Learning rate: {$5e^{-5}$, $1e^{-4}$}.

BoW Embedding size: {128, 256, 512}. Learning rate: {$1e^{-5}$, $5e^{-5}$}.

FiLM GP Number of ResBlocks: {3, 4, 5}. Learning rate: {$1e^{-4}$, $1e^{-3}$}. Number of classifier channels: {32, 64}.

FiLM AT Number of ResBlocks: {3, 4, 5}. Attention hidden size: {128, 256}. Learning rate: {$1e^{-5}$, $1e^{-4}$}.

Multi-hop Number of ResBlocks: {3, 4, 5}. Learning rate: {$1e^{-5}$, $1e^{-4}$}. Number of classifier channels: {32, 64}.

MAC Number of CNN Layers: {2,3}. Width: {512, 1024}. MAC time steps: {5,6}. We adapt the ramp-up/down Adam learning schedule popularly used in VQA [5], ramping up the learning rate to $1e^{-4}$ in the first 2 epochs and then decaying it to $1e^{-5}$ after epoch 10 (training is done for a total of 15 epochs).