RESEARCH ARTICLE

MACHINE LEARNING

# Neural scene representation and rendering

**S. M. Ali Eslami*†, Danilo Jimenez Rezende†, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, Demis Hassabis**

**Scene representation—the process of converting visual sensory data into concise descriptions—is a requirement for intelligent behavior. Recent work has shown that neural networks excel at this task when provided with large, labeled datasets. However, removing the reliance on human labeling remains an important open problem. To this end, we introduce the Generative Query Network (GQN), a framework within which machines learn to represent scenes using only their own sensors. The GQN takes as input images of a scene taken from different viewpoints, constructs an internal representation, and uses this representation to predict the appearance of that scene from previously unobserved viewpoints. The GQN demonstrates representation learning without human labels or domain knowledge, paving the way toward machines that autonomously learn to understand the world around them.**

Modern artificial vision systems are based on deep neural networks that consume large, labeled datasets to learn functions that map images to human-generated scene descriptions. They do so by, for example, categorizing the dominant object in the image (*1*), classifying the scene type (*2*), detecting object-bounding boxes (*3*), or labeling individual pixels into predetermined categories (*4*, *5*). In contrast, intelligent agents in the natural world appear to require little to no explicit supervision for perception (*6*). Higher mammals, including human infants, learn to form representations that support motor control, memory, planning, imagination, and rapid skill acquisition without any social communication, and generative processes have been hypothesized to be instrumental for this ability (*7*–*10*). It is thus desirable to create artificial systems that learn to represent scenes by modeling data [e.g., two-dimensional (2D) images and the agent's position in space] that agents can directly obtain while processing the scenes themselves, without recourse to semantic labels (e.g., object classes, object locations, scene types, or part labels) provided by a human (*11*).

To that end, we present the Generative Query Network (GQN). In this framework, as an agent navigates a 3D scene $i$, it collects $K$ images $\mathbf{x}_i^k$ from 2D viewpoints $\mathbf{v}_i^k$, which we collectively refer to as its observations $o_i = \{(\mathbf{x}_i^k, \mathbf{v}_i^k)\}_{k=1,\ldots,K}$. The agent passes these observations to a GQN composed of two main parts: a representation network $f$ and a generation network $g$ (Fig. 1). The representation network takes as input the agent's observations and produces a neural scene representation $\mathbf{r}$, which encodes information about the underlying scene (we omit scene subscript $i$ where possible, for clarity). Each additional observation accumulates further evidence about the contents of the scene in the same representation. The generation network then predicts the scene from an arbitrary query viewpoint $\mathbf{v}^q$, using stochastic latent variables $\mathbf{z}$ to create variability in its outputs where necessary. The two networks are trained jointly, in an end-to-end fashion, to maximize the likelihood of generating the ground-truth image that would be observed from the query viewpoint. More formally, (i) $\mathbf{r} = f_\theta(o_i)$, (ii) the deep generation network $g$ defines a probability density $g_\theta(\mathbf{x}|\mathbf{v}^q, \mathbf{r}) = \int g_\theta(\mathbf{x}, \mathbf{z}|\mathbf{v}^q, \mathbf{r}) d\mathbf{z}$ of an image $\mathbf{x}$ being observed at query viewpoint $\mathbf{v}^q$ for a scene representation $\mathbf{r}$ using latent variables $\mathbf{z}$, and (iii) the learnable parameters are denoted by $\theta$. Although the GQN training objective is intractable, owing to the presence of latent variables, we can employ variational approximations and optimize with stochastic gradient descent.

The representation network is unaware of the viewpoints that the generation network will be queried to predict. As a result, it will produce scene representations that contain all information (e.g., object identities, positions, colors, counts, and room layout) necessary for the generator to make accurate image predictions. In other words, the GQN will learn by itself what these factors are, as well as how to extract them from pixels. Moreover, the generator internalizes any statistical regularities (e.g., typical colors of the sky, as well as object shape regularities and symmetries, patterns, and textures) that are common across different scenes. This allows

DeepMind, 5 New Street Square, London EC4A 3TW, UK.
*Corresponding author. Email: aeslami@google.com
†These authors contributed equally to this work.



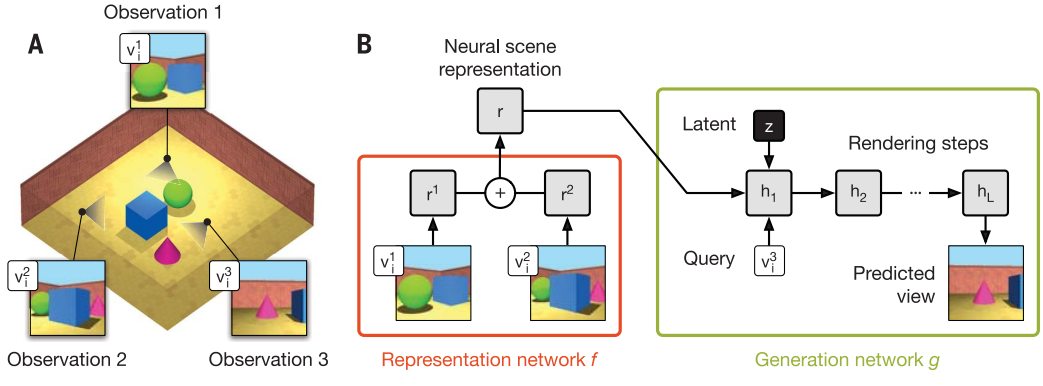


**Fig. 1. Schematic illustration of the Generative Query Network.** (**A**) The agent observes training scene $i$ from different viewpoints (in this example, from $v_i^1$, $v_i^2$, and $v_i^3$). (**B**) The inputs to the representation network $f$ are observations made from viewpoints $v_i^1$ and $v_i^2$, and the output is the scene representation $r$, which is obtained by element-wise summing of the observations' representations. The generation network, a recurrent latent variable model, uses the representation to predict what the scene would look like from a different viewpoint $v_i^3$. The generator can succeed only if $r$ contains accurate and complete information about the contents of the scene (e.g., the identities, positions, colors, and counts of the objects, as well as the room's colors). Training via back-propagation across many scenes, randomizing the number of observations, leads to learned scene representations that capture this information in a concise manner. Only a handful of observations need to be recorded from any single scene to train the GQN. $h_1$, $h_2$,...$h_L$ are the $L$ layers of the generation network.

the GQN to reserve its representation capacity for a concise, abstract description of the scene, with the generator filling in the details where necessary. For instance, instead of specifying the precise shape of a robot arm, the representation network can succinctly communicate the configuration of its joints, and the generator knows how this high-level representation manifests itself as a fully rendered arm with its precise shapes and colors. In contrast, voxel (*12*–*15*) or point-cloud (*16*) methods (as typically obtained by classical structure-from-motion) employ literal representations and therefore typically scale poorly with scene complexity and size and are also difficult to apply to nonrigid objects (e.g., animals, vegetation, or cloth).

### Rooms with multiple objects

To evaluate the feasibility of the framework, we experimented with a collection of environments in a simulated 3D environment. In the first set of experiments, we considered scenes in a square room containing a variety of objects. Wall textures—as well as the shapes, positions, and colors of the objects and lights—are randomized, allowing for an effectively infinite number of total scene configurations; however, we used finite datasets to train and test the model [see section 4 of (*17*) for details]. After training, the GQN computes its scene representation by observing one or more images of a previously unencountered, held-out test scene. With this representation, which can be as small as 256 dimensions, the generator's predictions at query viewpoints are highly accurate and mostly indistinguishable from ground truth (Fig. 2A). The only way in which the model can succeed at this task is by perceiving and compactly encoding in the scene representation vector $\mathbf{r}$ the number of objects present in each scene, their positions in the room, the colors in which they appear, the colors of the walls, and the indirectly observed position of the light source. Unlike in traditional supervised learning, GQNs learn to make these inferences from images without any explicit human labeling of the contents of scenes. Moreover, the GQN's generator learns an approximate 3D renderer (in other words, a program that can generate an image when given a scene representation and camera viewpoint) without any prior specification of the laws of perspective, occlusion, or lighting (Fig. 2B). When the contents of the scene are not explicitly specified by the observation (e.g., because of heavy occlusion), the model's uncertainty is reflected in the variability of the generator's samples (Fig. 2C). These properties are best observed in real-time, interactive querying of the generator (movie S1).

Notably, the model observes only a small number of images (in this experiment, fewer than five) from each scene during training, yet it is capable of rendering unseen training or test scenes from arbitrary viewpoints. We also monitored

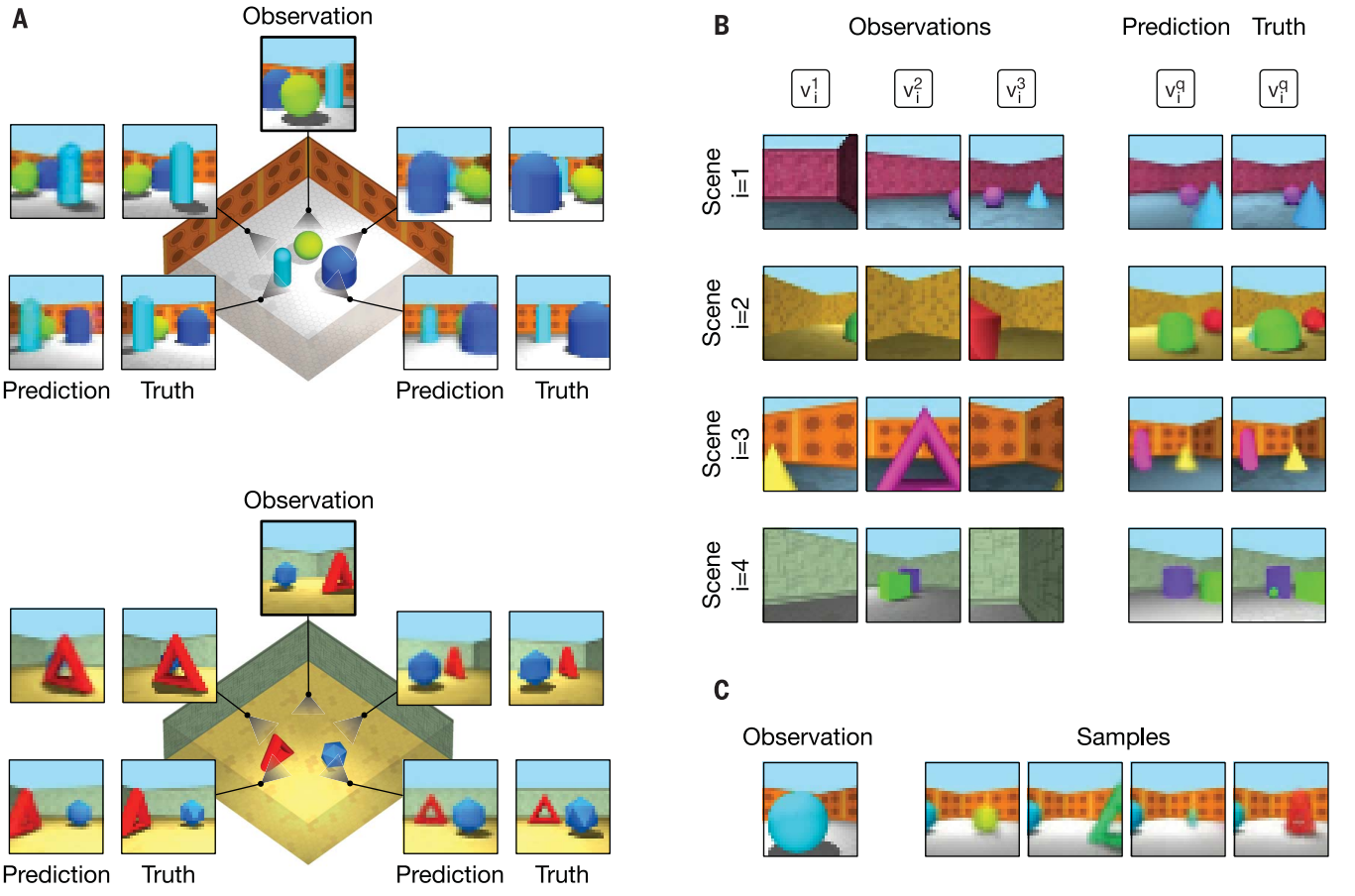


**Fig. 2. Neural scene representation and rendering.** (**A**) After having made a single observation of a previously unencountered test scene, the representation network produces a neural description of that scene. Given this neural description, the generator is capable of predicting accurate images from arbitrary query viewpoints. This implies that the scene description captures the identities, positions, colors, and counts of the objects, as well as the position of the light and the colors of the room. (**B**) The generator's predictions are consistent with laws of perspective, occlusion, and lighting (e.g., casting object shadows consistently). When observations provide views of different parts of the scene, the GQN correctly aggregates this information (scenes two and three). (**C**) Sample variability indicates uncertainty over scene contents (in this instance, owing to heavy occlusion). Samples depict plausible scenes, with complete objects rendered in varying positions and colors (see fig. S7 for further examples). The model's behavior is best visualized in movie format; see movie S1 for real-time, interactive querying of GQN's representation of test scenes.

the likelihood of predicted observations of training and test scenes (fig. S3) and found no noticeable difference between values of the two. Taken together, these points rule out the possibility of model overfitting.

Analysis of the trained GQN highlights several desirable properties of its scene representation network. Two-dimensional t-distributed stochastic neighbor embedding (t-SNE) (*18*) visualization of GQN scene representation vectors shows clear clustering of images of the same scene, despite marked changes in viewpoint (Fig. 3A). In contrast, representations produced by autoencoding density models such as variational autoencoders (VAEs) (*19*) apparently fail to capture the contents of the underlying scenes [section 5 of (*17*)]; they appear to be representations of the observed images instead. Furthermore, when prompted to reconstruct a target image, GQN exhibits compositional behavior, as it is capable of both representing and rendering combinations of scene elements it has never encountered during training (Fig. 3B), despite learning that these compositions are unlikely. To test whether the GQN learns a factorized representation, we investigated whether changing a single scene property (e.g., object color) while keeping others (e.g., object shape and position) fixed leads to similar changes in the scene representation (as defined by mean cosine similarity across scenes). We found that object color, shape, and size; light position; and, to a lesser extent, object positions are indeed factorized [Fig. 3C and sections 5.3 and 5.4 of (*17*)]. We also found that the GQN is able to carry out "scene algebra" [akin to word embedding algebra (*20*)]. By adding and subtracting representations of related scenes, we found that object and scene properties can be controlled, even across object positions [Fig. 4A and section 5.5 of (*17*)]. Finally, because it is a probabilistic model, GQN also learns to integrate information from different viewpoints in an efficient and consistent manner, as demonstrated by a reduction in its Bayesian "surprise" at observing a held-out image of a scene as the number of views increases [Fig. 4B and section 3 of (*17*)]. We include analysis on the GQN's ability to generalize to out-of-distribution scenes, as well as further results on modeling of Shepard-Metzler objects, in sections 5.6 and 4.2 of (*17*).

## Control of a robotic arm

Representations that succinctly reflect the true state of the environment should also allow agents to learn to act in those environments more robustly and with fewer interactions. Therefore, we considered the canonical task of moving a robotic arm to reach a colored object, to test the GQN representation's suitability for control. The end-goal of deep reinforcement learning is to learn the control policy directly from pixels; however, such methods require a large amount of experience to learn from sparse rewards. Instead, we first trained a GQN and used it to succinctly represent the observations. A policy was then trained

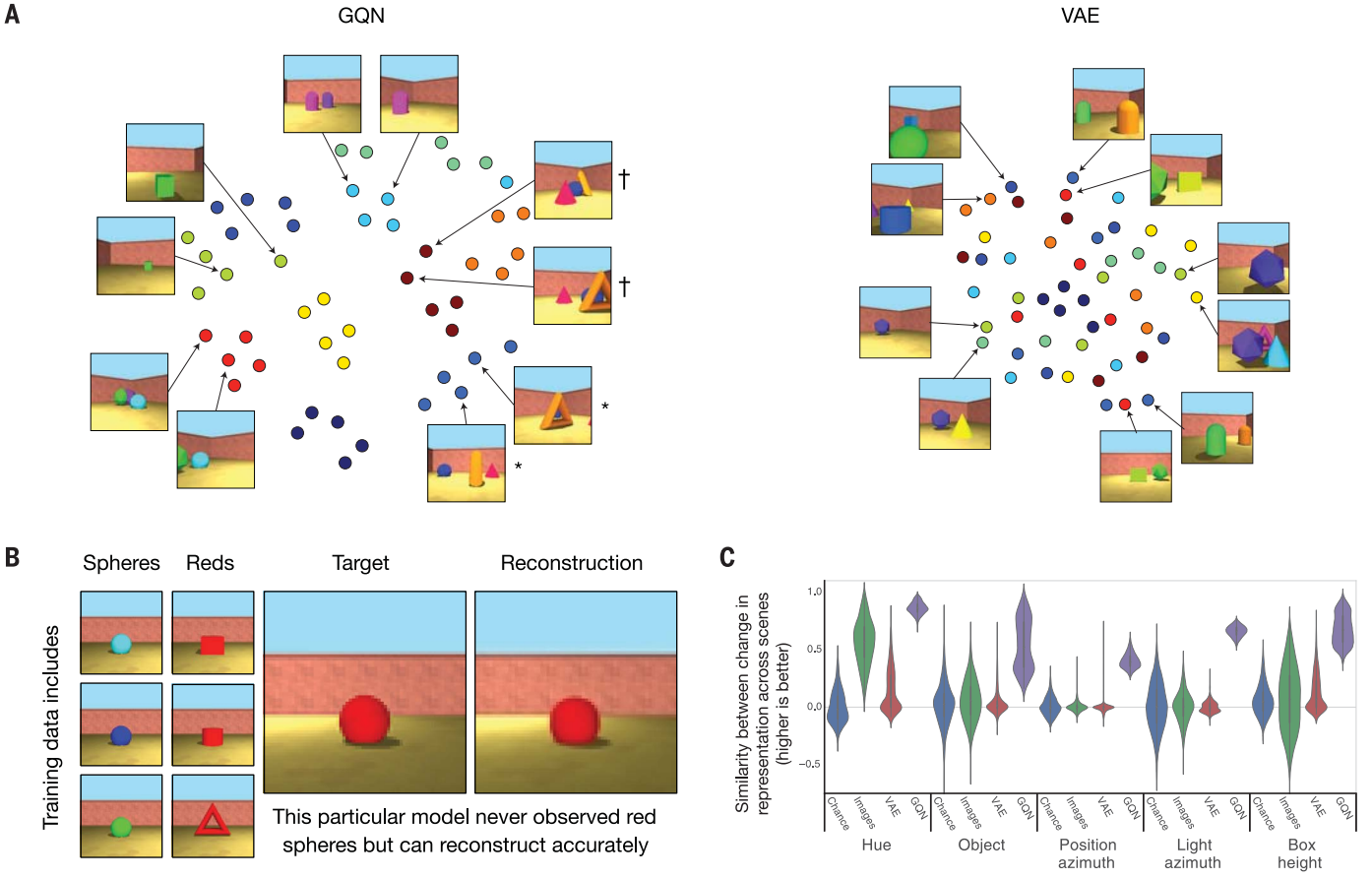


**Fig. 3. Viewpoint invariance, compositionality, and factorization of the learned scene representations.** (**A**) t-SNE embeddings. t-SNE is a method for nonlinear dimensionality reduction that approximately preserves the metric properties of the original high-dimensional data. Each dot represents a different view of a different scene, with color indicating scene identity. Whereas the VAE clusters images mostly on the basis of wall angles, GQN clusters images of the same scene, independent of view (scene representations computed from each image individually). Two scenes with the same objects (represented by asterisk and dagger symbols) but in different positions are clearly separated. (**B**) Compositionality demonstrated by reconstruction of holdout shape-color combinations. (**C**) GQN factorizes object and scene properties because the effect of changing a specific property is similar across diverse scenes (as defined by mean cosine similarity of the changes in the representation across scenes). For comparison, we plot chance factorization, as well as the factorization of the image-space and VAE representations. See section 5.3 of (*17*) for details.

to control the arm directly from these representations. In this setting, the representation network must learn to communicate only the arm's joint angles, the position and color of the object, and the colors of the walls for the generator to be able to predict new views. Because this vector has much lower dimensionality than the raw input images, we observed substantially more robust and data-efficient policy learning, obtaining convergence-level control performance with approximately one-fourth as many interactions with the environment as a standard method using raw pixels [Fig. 5 and section 4.4 of (*17*)]. The 3D nature of the GQN representation allows us to train a policy from any viewpoint around the arm and is sufficiently stable to allow for arm-joint velocity control from a freely moving camera.

### Partially observed maze environments

Finally, we considered more complex, procedural maze-like environments to test GQN's scaling properties. The mazes consist of multiple rooms connected via corridors, and the layout of each maze and the colors of the walls are randomized in each scene. In this setting, any single observation provides a small amount of information about the current maze. As before, the training objective for GQN is to predict mazes from new viewpoints, which is possible only if GQN successfully aggregates multiple observations to determine the maze layout (i.e., the wall and floor colors, the number of rooms, their positions in space, and how they connect to one another via corridors). We observed that GQN is able to make correct predictions from new first-person viewpoints (Fig. 6A). We queried the GQN's

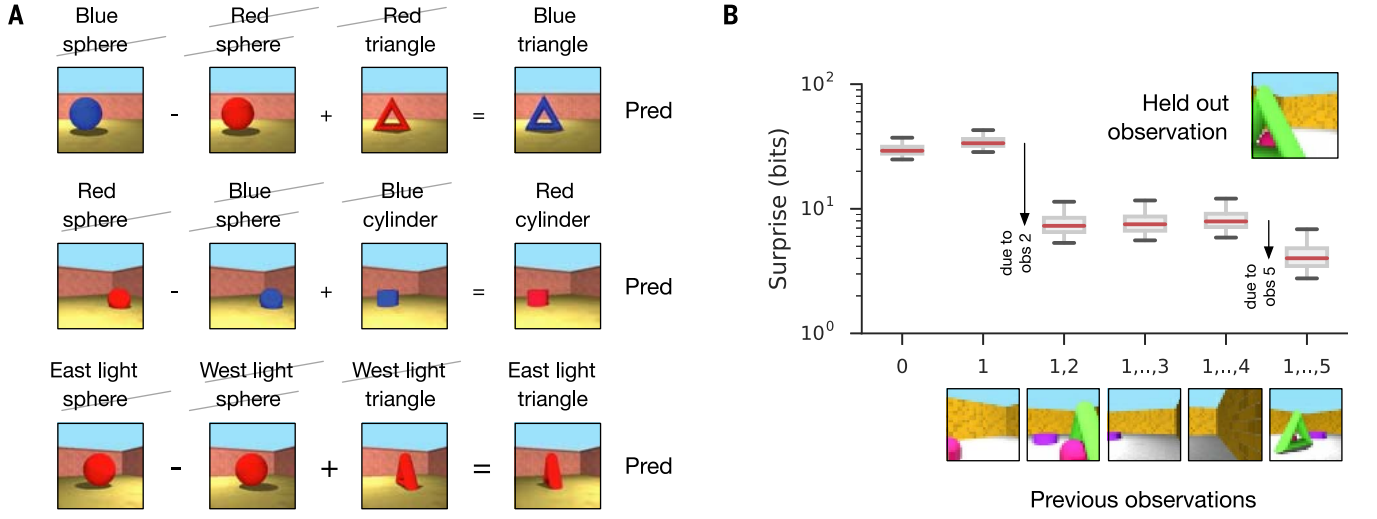


**Fig. 4. Scene algebra and Bayesian surprise.** (**A**) Adding and subtracting representations of related scenes enables control of object and scene properties via "scene algebra" and indicates factorization of shapes, colors, and positions. Pred, prediction. (**B**) Bayesian surprise at a new observation after having made observations 1 to $k$ for $k$ = 1 to 5. When the model observes images that contain information about the layout of the scene, its surprise (defined as the Kullback-Leibler divergence between conditional prior and posterior) at observing the held-out image decreases.

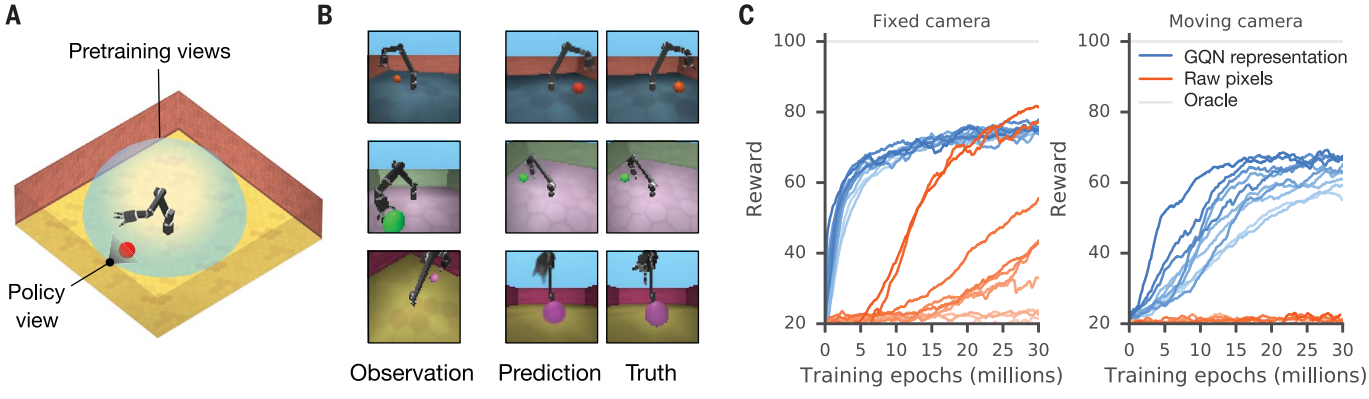


**Fig. 5. GQN representation enables more robust and data-efficient control.** (**A**) The goal is to learn to control a robotic arm to reach a randomly positioned colored object. The controlling policy observes the scene from a fixed or moving camera (gray). We pretrain a GQN representation network by observing random configurations from random viewpoints inside a dome around the arm (light blue). (**B**) The GQN infers a scene representation that can accurately reconstruct the scene. (**C**) (Left) For a fixed camera, an asynchronous advantage actor-critic reinforcement learning (RL) agent (*44*) learns to control the arm using roughly one-fourth as many experiences when using the GQN representation, as opposed to a standard method using raw pixels (lines correspond to different hyperparameters; same hyperparameters explored for both standard and GQN agents; both agents also receive viewpoint coordinates as inputs). The final performance achieved by learning from raw pixels can be slightly higher for some hyperparameters, because some task-specific information might be lost when learning a compressed representation independently from the RL task as GQN does. (Right) The benefit of GQN is most pronounced when the policy network's view on the scene moves from frame to frame, suggesting viewpoint invariance in its representation. We normalize scores such that a random agent achieves 0 and an agent trained on "oracle" ground-truth state information achieves 100.

representation more directly by training a separate generator to predict a top-down view of the maze and found that it yields highly accurate predictions (Fig. 6B). The model's uncertainty, as measured by the entropy of its first-person and top-down samples, decreases as more observations are made [Fig. 6B and section 3 of (*17*)]. After about only five observations, the GQN's uncertainty disappears almost entirely.

### Related work

GQN offers key advantages over prior work. Traditional structure-from-motion, structure-from-depth, and multiview geometry techniques (*12*–*16*, *21*) prescribe the way in which the 3D structure of the environment is represented (for instance, as point clouds, mesh clouds, or a collection of predefined primitives). GQN, by contrast, learns this representational space, allowing it to express the presence of textures, parts, objects, lights, and scenes concisely and at a suitably high level of abstraction. Furthermore, its neural formulation enables task-specific fine-tuning of the representation via backpropagation (e.g., via further supervised or reinforced deep learning).

Classical neural approaches to this learning problem—e.g., autoencoding and density models (*22*–*27*)—are required to capture only the distribution of observed images, and there is no explicit mechanism to encourage learning of how different views of the same 3D scene relate to one another. The expectation is that statistical compression principles will be sufficient to enable networks to discover the 3D structure of the environment; however, in practice, they fall short of achieving this kind of meaningful representation and instead focus on regularities of colors and patches in the image space.

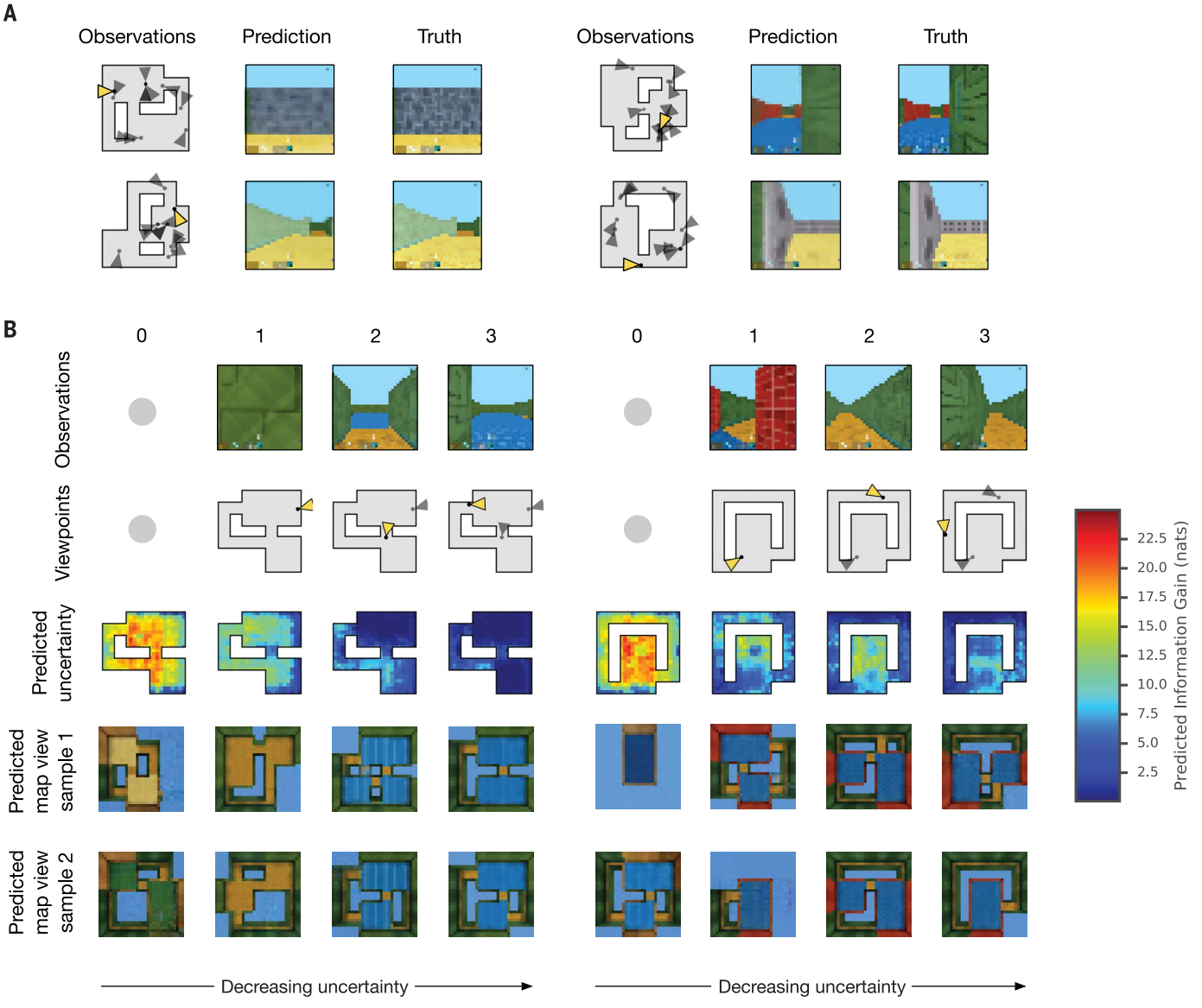


**Fig. 6. Partial observability and uncertainty.** (**A**) The agent (GQN) records several observations of a previously unencountered test maze (indicated by gray triangles). It is then capable of accurately predicting the image that would be observed at a query viewpoint (yellow triangle). It can accomplish this task only by aggregating information across multiple observations. (**B**) In the *k*th column, we condition GQN on observations 1 to *k* and show GQN's predicted uncertainty, as well as two of GQN's sampled predictions of the top-down view of the maze. Predicted uncertainty is measured by computing the model's Bayesian surprise at each location, averaged over three different heading directions. The model's uncertainty decreases as more observations are made. As the number of observations increases, the model predicts the top-down view with increasing accuracy. See section 3 of (*17*), fig. S8, and movie S1 for further details and results. nats, natural units of information.

Viewpoint transformation networks do explicitly learn this relationship; however, they have thus far been nonprobabilistic and limited in scale—e.g., restricted to rotation around individual objects for which a single view is sufficient for prediction (*15*, *28*–*33*) or to small camera displacements between stereo cameras (*34*–*36*).

By employing state-of-the-art deep, iterative, latent variable density models (*25*), GQN is capable of handling free agent movement around scenes containing multiple objects. In addition, owing to its probabilistic formulation, GQN can account for uncertainty in its understanding about a scene's contents in the face of severe occlusion and partial observability. Notably, the GQN framework is not specific to the particular choice of architecture of the generation network, and alternatives such as generative adversarial networks (*37*) or autoregressive models (*38*) could be employed.

A closely related body of work is that of discriminative pose estimation (*39*–*41*), in which networks are trained to predict camera motion between consecutive frames. The GQN formulation is advantageous, as it allows for aggregation of information from multiple images of a scene (see maze experiments); it is explicitly probabilistic, allowing for applications such as exploration through Bayesian information gain; and, unlike the aforementioned methods where scene representation and pose prediction are intertwined, the GQN architecture admits a clear architectural separation between the representation and generation networks. The idea of pose estimation is complementary, however—the GQN can be augmented with a second "generator" that, given an image of a scene, predicts the viewpoint from which it was taken, providing a new source of gradients with which to train the representation network.

## Outlook

In this work, we have shown that a single neural architecture can learn to perceive, interpret, and represent synthetic scenes without any human labeling of the contents of these scenes. It can also learn a powerful neural renderer that is capable of producing accurate and consistent images of scenes from new query viewpoints. The GQN learns representations that adapt to and compactly capture the important details of its environment (e.g., the positions, identities, and colors of multiple objects; the configuration of the joint angles of a robot arm; and the layout of a maze), without any of these semantics being built into the architecture of the networks. GQN uses analysis-by-synthesis to perform "inverse graphics," but unlike existing methods (*42*), which require problem-specific engineering in the design of their generators, GQN learns this behavior by itself and in a generally applicable manner. However, the resulting representations are no longer directly interpretable by humans.

Our experiments have thus far been restricted to synthetic environments for three reasons: (i) a need for controlled analysis, (ii) limited availability of suitable real datasets, and (iii) limitations of generative modeling with current hardware. Although the environments are relatively constrained in terms of their visual fidelity, they capture many of the fundamental difficulties of vision—namely, severe partial observability and occlusion—as well as the combinatorial, multi-object nature of scenes. As new sources of data become available (*41*) and advances are made in generative modeling capabilities (*37*, *43*), we expect to be able to investigate application of the GQN framework to images of naturalistic scenes.

Total scene understanding involves more than just representation of the scene's 3D structure. In the future, it will be important to consider broader aspects of scene understanding—e.g., by querying across both space and time for modeling of dynamic and interactive scenes—as well as applications in virtual and augmented reality and exploration of simultaneous scene representation and localization of observations, which relates to the notion of simultaneous localization and mapping in computer vision.

Our work illustrates a powerful approach to machine learning of grounded representations of physical scenes, as well as of the associated perception systems that holistically extract these representations from images, paving the way toward fully unsupervised scene understanding, imagination, planning, and behavior.

## REFERENCES AND NOTES


1. A. Krizhevsky, I. Sutskever, G. E. Hinton, in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, 2012), pp. 1097–1105.
2. B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, 2014), pp. 487–495.
3. S. Ren, K. He, R. Girshick, J. Sun, in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett, Eds. (Curran Associates, 2015), pp. 91–99.
4. R. Girshick, J. Donahue, T. Darrell, J. Malik, in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2014), pp. 580–587.
5. M. C. Mozer, R. S. Zemel, M. Behrmann, in *Advances in Neural Information Processing Systems 4 (NIPS 1991)*, J. E. Moody, S. J. Hanson, R. P. Lippmann, Eds. (Morgan-Kaufmann, 1992), pp. 436–443.
6. J. Konorski, *Science* **160**, 652–653 (1968).
7. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information* (Henry Holt and Co., 1982).
8. D. Hassabis, E. A. Maguire, *Trends Cogn. Sci.* **11**, 299–306 (2007).
9. D. Kumaran, D. Hassabis, J. L. McClelland, *Trends Cogn. Sci.* **20**, 512–534 (2016).
10. B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, *Science* **350**, 1332–1338 (2015).
11. S. Becker, G. E. Hinton, *Nature* **355**, 161–163 (1992).
12. Z. Wu *et al*., in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2015), pp. 1912–1920.
13. J. Wu, C. Zhang, T. Xue, W. Freeman, J. Tenenbaum, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett, Eds. (Curran Associates, 2016), pp. 82–90.
14. D. J. Rezende *et al*., in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett, Eds. (Curran Associates, 2016), pp. 4996–5004.
15. X. Yan, J. Yang, E. Yumer, Y. Guo, H. Lee, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett, Eds. (Curran Associates, 2016), pp. 1696–1704.
16. M. Pollefeys *et al*., *Int. J. Comput. Vision* **59**, 207–232 (2004).
17. See supplementary materials.
18. L. van der Maaten, *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
19. I. Higgins *et al*., at International Conference on Learning Representations (ICLR) (2017).
20. T. Mikolov *et al*., in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger, Eds. (Curran Associates, 2013), pp. 3111–3119.
21. Y. Zhang, W. Xu, Y. Tong, K. Zhou, *ACM Trans. Graph.* **34**, 159 (2015).
22. D. P. Kingma, M. Welling, arXiv:1312.6114 [stat.ML] (20 December 2013).
23. D. J. Rezende, S. Mohamed, D. Wierstra, in *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)* (JMLR, 2014), vol. 32, pp. 1278–1286.
24. I. Goodfellow *et al*., in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, 2014), pp. 2672–2680.
25. K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, D. Wierstra, in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett, Eds. (Curran Associates, 2016), pp. 3549–3557
26. P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, in *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)* (ACM, 2008), pp. 1096–1103.
27. P. Dayan, G. E. Hinton, R. M. Neal, R. S. Zemel, *Neural Comput.* **7**, 889–904 (1995).
28. G. E. Hinton, A. Krizhevsky, S. D. Wang, in *Proceedings of the 21st International Conference on Artificial Neural Networks and Machine Learning (ICANN 2011)*, T. Honkela, W. Duch, M. Girolami, S. Kaski, Eds. (Lecture Notes in Computer Science Series, Springer, 2011), vol. 6791, pp. 44–51.
29. C. B. Choy, D. Xu, J. Gwak, K. Chen, S. Savarese, in *Proceedings of the 2016 European Conference on Computer Vision (ECCV)* (Lecture Notes in Computer Science Series, Springer, 2016), vol. 1, pp. 628–644.
30. M. Tatarchenko, A. Dosovitskiy, T. Brox, in *Proceedings of the 2016 European Conference on Computer Vision (ECCV)* (Lecture Notes in Computer Science Series, Springer, 2016), vol. 9911, pp. 322–337.
31. F. Anselmi *et al*., *Theor. Comput. Sci.* **633**, 112–121 (2016).
32. D. F. Fouhey, A. Gupta, A. Zisserman, in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 1516–1524.
33. A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, T. Brox, *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 692–705 (2017).
34. C. Godard, O. Mac Aodha, G. J. Brostow, in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2017), pp. 6602–6611.
35. T. Zhou, S. Tulsiani, W. Sun, J. Malik, A. A. Efros, in *Proceedings of the 2016 European Conference on Computer Vision (ECCV)* (Lecture Notes in Computer Science Series, Springer, 2016), pp. 286–301.
36. J. Flynn, I. Neulander, J. Philbin, N. Snavely, in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 5515–5524.
37. T. Karras, T. Aila, S. Laine, J. Lehtinen, arXiv:1710.10196 [cs.NE] (27 October 2017).
38. A. van den Oord *et al*., in *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett, Eds. (Curran Associates, 2016), pp. 4790–4798.
39. D. Jayaraman, K. Grauman, in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2015), pp. 1413–1421.
40. P. Agrawal, J. Carreira, J. Malik, arXiv:1505.01596 [cs.CV] (7 May 2015).
41. A. R. Zamir *et al*., in *Proceedings of the 2016 European Conference on Computer Vision (ECCV)* (Lecture Notes in Computer Science Series, Springer, 2016), pp. 535–553.
42. T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, V. Mansinghka, in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2015), pp. 4390–4399.
43. Q. Chen, V. Koltun, in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)* (IEEE, 2017), pp. 1511–1520.
44. A. A. Rusu *et al*., arXiv:1610.04286 [cs.RO] (13 October 2016).

**ACKNOWLEDGMENTS**

We thank M. Shanahan, A. Zisserman, P. Dayan, J. Leibo, P. Battaglia, and G. Wayne for helpful discussions and advice; G. Ostrovski, N. Heess, D. Zoran, V. Nair, and D. Silver for reviewing the paper; K. Anderson for help creating environments; and the rest of the DeepMind team for support and ideas. **Funding:** This research was funded by DeepMind. **Author contributions:** S.M.A.E. and D.J.R. conceived of the model. S.M.A.E., D.J.R., F.B., and F.V. designed and implemented the model, datasets, visualizations, figures, and videos. A.S.M. and A.R. designed and performed analysis experiments. M.G. and A.A.R. performed robot arm experiments. I.D., D.P.R., O.V., and D.R. assisted with maze navigation experiments. L.B. and T.W. assisted with Shepard-Metzler experiments. H.K., C.H., K.G., M.B., D.W., N.R., K.K., and D.H. managed, advised, and contributed ideas to the project. S.M.A.E. and D.J.R. wrote the paper. **Competing interests:** The authors declare no competing financial interests. DeepMind has filed a U.K. patent application (GP-201495-00-PCT) related to this work. **Data and materials availability:** Datasets used in the experiments have been made available to download at https://github.com/deepmind/gqn-datasets.

**SUPPLEMENTARY MATERIALS**

www.sciencemag.org/content/360/6394/1204/suppl/DC1
Supplementary Text
Figs. S1 to S16
Algorithms S1 to S3
Table S1
References (*45*–*52*)
Movie S1

29 November 2017; accepted 10 April 2018
10.1126/science.aar6170

Science

# Neural scene representation and rendering

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu and Demis Hassabis

*Science* **360** (6394), 1204-1210.
DOI: 10.1126/science.aar6170

**A scene-internalizing computer program**

To train a computer to "recognize" elements of a scene supplied by its visual sensors, computer scientists typically use millions of images painstakingly labeled by humans. Eslami *et al.* developed an artificial vision system, dubbed the Generative Query Network (GQN), that has no need for such labeled data. Instead, the GQN first uses images taken from different viewpoints and creates an abstract description of the scene, learning its essentials. Next, on the basis of this representation, the network predicts what the scene would look like from a new, arbitrary viewpoint.

*Science*, this issue p. 1204