RELATIONAL NEURAL EXPECTATION MAXIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Common-sense physical reasoning is an essential ingredient for any intelligent agent operating in the real-world. For example, it can be used to simulate the environment, or to infer the state of parts of the world that are currently unobserved. In order to match real-world conditions this causal knowledge must be learned without access to supervised data. To solve this problem, we present a novel method that incorporates prior knowledge about the compositional nature of human perception to factor interactions between object-pairs and to learn them efficiently. It learns to discover objects and to model physical interactions between them from raw visual images in a purely *unsupervised* fashion. On videos of bouncing balls we show the superior modelling capabilities of our method compared to other unsupervised neural approaches, that do not incorporate such prior knowledge. We show its ability to handle occlusion and that it can extrapolate learned knowledge to environments with different numbers of objects.

1 INTRODUCTION

Humans rely on common-sense physical reasoning to solve many everyday physics-related tasks (Lake et al., 2016). For example, it enables them to foresee the consequences of their actions (simulation), or to infer the state of parts of the world that are currently unobserved. This *causal* understanding is an essential ingredient for any intelligent agent that is to operate within the world.

Common-sense physical reasoning is facilitated by the discovery and representation of objects (a *core* domain of human cognition (Spelke & Kinzler, 2007)) that serve as primitives of a compositional system. They allow humans to decompose a complex visual scene into distinct parts, describe relations between them and reason about their dynamics as well as the consequences of their interactions (Battaglia et al., 2016) .The most successful machine learning approaches to common-sense physical reasoning incorporate such prior knowledge in their design. They maintain explicit object representations, which allow for general physical dynamics to be learned between object pairs in a compositional manner (Battaglia et al., 2016; Chang et al., 2016; Watters et al., 2017). Learning is *supervised*, however, since it relies on object-representations from external sources (e.g. a physics simulator), which are unavailable in most real-world scenarios.

Neural approaches that learn to directly model motion or physical interactions in pixel space offer an alternative solution (Srivastava et al., 2015; Sutskever et al., 2009). While unsupervised, these methods suffer from a lack compositionality at the representational level of objects. This prevents such end-to-end neural approaches from efficiently learning functions that operate on the entity level and generalize in a human way (c.f. Battaglia et al. (2013); Lake et al. (2016); Santoro et al. (2017), but see Perez et al. (2017)).

In this work we propose *Relational N-EM* (R-NEM), a novel approach to common-sense physical reasoning that learns physical interactions between objects from raw visual images in a purely *unsupervised* fashion. At its core is *Neural Expectation Maximization* (N-EM; Greff et al., 2017), a method that allows for the discovery of compositional object-representations, yet is unable to model interactions between objects. Therefore, we endow N-EM with a relational mechanism inspired by previous work (Battaglia et al., 2016; Chang et al., 2016; Santoro et al., 2017), enabling it to factor interactions between object-pairs, learn efficiently, and generalize to environments with a different number of objects without re-training.

2 METHOD

Our goal is to learn common-sense physical reasoning in a purely unsupervised fashion directly from visual observations. We have argued, that in order to solve this problem, we need to exploit its compositional structure. Conventional unsupervised representation learning approaches (eg. VAEs, Kingma & Welling (2013); GANs Goodfellow et al. (2014)) learn a single distributed representation that *superimposes* information about the input, without imposing any structure regarding objects or other low-level primitives. These monolithic representations can not factorize physical interactions between pairs of objects and therefore lack an essential inductive bias to learn these efficiently. Hence, we require an alternative approach that can discover objects representations as primitives of a visual scene in an unsupervised fashion.

One such approach is *Neural Expectation Maximization* (N-EM; Greff et al. (2017)), which learns a separate distributed representation for each object described in terms of the same features through an iterative process of perceptual grouping and representation learning. The compositional nature of these representations enable us to formulate *Relational N-EM* (R-NEM): a novel *unsupervised* approach to common-sense physical reasoning that combines N-EM (Section 2.1) with an interaction function that models relations between objects efficiently (Section 2.2).

2.1 NEURAL EXPECTATION MAXIMIZATION

Neural Expectation Maximization (N-EM; Greff et al. (2017)) is a differentiable clustering method that learns a representation of a visual scene composed of primitive object representations. These representations adhere to many useful properties of a symbolic representation of objects and can therefore be used as primitives of a compositional system (Hummel et al., 2004). They are described in the same format and each contain only information about the object in the visual scene that they correspond to. Together, they form a representation of a visual scene composed of objects that is learned in an unsupervised way, which therefore serves as a starting point for our approach.

The goal of N-EM is to group pixels in the input that belong to the same object (perceptual grouping) and capture this information efficiently in a distributed representation $\boldsymbol{\theta}_k$ for each object. At a high-level, the idea is that if we were to have access to the family of distributions $P(\boldsymbol{x}|\boldsymbol{\theta}_k)$ (a statistical model of images given object representations $\boldsymbol{\theta}_k$) then we can formalize our objective as inference in a mixture of these distributions. By using *Expectation Maximization* (EM; Dempster et al., 1977) to compute a Maximum Likelihood Estimate (MLE) of the parameters of this mixture $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$, we obtain a grouping (clustering) of the pixels to each object (component) and their corresponding representation. In reality, we do not have access to $P(\boldsymbol{x}|\boldsymbol{\theta}_k)$, which N-EM learns instead by parametrizing the mixture with a neural network, to then back-propagate through the iterations of the unrolled generalized EM procedure.

Following Greff et al. (2017), we model each image $\boldsymbol{x} \in \mathbb{R}^D$ as a spatial mixture of K components parametrized by vectors $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \in \mathbb{R}^M$. A neural network f_ϕ is used to transform these representations $\boldsymbol{\theta}_k$ into parameters $\psi_{i,k} = f_\phi(\boldsymbol{\theta}_k)_i$ for separate pixel-wise distributions. A set of binary latent variables $\boldsymbol{\mathcal{Z}} \in [0,1]^{D \times K}$ encodes the unknown true pixel assignments, such that $z_{i,k} = 1$ iff pixel i was generated by component k. The full likelihood for $\boldsymbol{x}$ given $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ is given by:

$$P(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_{i=1}^{D} \sum_{\boldsymbol{z}_i} P(x_i, \boldsymbol{z}_i|\boldsymbol{\psi}_i) = \prod_{i=1}^{D} \sum_{k=1}^{K} P(z_{i,k} = 1) P(x_i|\psi_{i,k}, z_{i,k} = 1). \tag{1}$$

If f_ϕ has learned a statistical model of images given object representations $\boldsymbol{\theta}_k$, then we can compute the object representations for a given image $\boldsymbol{x}$ by maximizing $P(\boldsymbol{x}|\boldsymbol{\theta})$. Marginalization over $\boldsymbol{z}$ complicates this process and we use generalized EM to maximize the following lowerbound instead:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{z}} P(\mathbf{z}|\boldsymbol{x}, \boldsymbol{\psi}^{\text{old}}) \log P(\boldsymbol{x}, \mathbf{z}|\boldsymbol{\psi}). \tag{2}$$

Each iteration of generalized EM consists of two steps: the *E-step* computes a new estimate of the posterior probability distribution over the latent variables $\gamma_{i,k} := P(z_{i,k} = 1|x_i, \psi_i^{\text{old}})$ given $\boldsymbol{\theta}^{\text{old}}$ from the previous iteration. It yields a new soft-assignment of the pixels to the components (clusters),

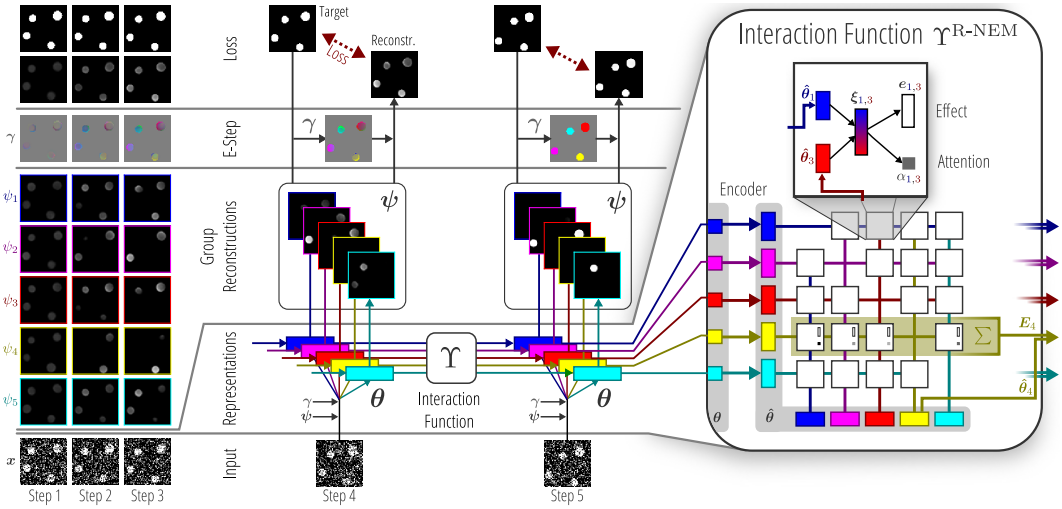

Figure 1: Illustration of the different computational aspects of R-NEM when applied to a sequence of bouncing balls. Note that $\boldsymbol{\gamma}, \boldsymbol{\psi}$ at the *representations* level correspond to the $\boldsymbol{\gamma}$ (E-step), $\boldsymbol{\psi}$ (Group Reconstructions) from the previous time-step. On the right side, a computational overview of $\Upsilon^{\text{R-NEM}}$ is shown.

based on how accurate they model $\boldsymbol{x}$. The generalized *M-step* updates $\boldsymbol{\theta}^{\text{old}}$ by taking a gradient ascent step on (2), using the previously computed soft-assignments: $\boldsymbol{\theta}_k^{\text{new}} = \boldsymbol{\theta}_k^{\text{old}} + \eta \cdot \partial \mathcal{Q} / \partial \boldsymbol{\theta}_k$.[1]

The unrolled computational graph of the generalized EM steps is differentiable, which provides a means to train f_ϕ to implement a statistical model of images given object representations. Using back-propagation through time (eg. Werbos (1988); Williams (1989)) we optimize ϕ in the direction that minimizes the following loss:

$$L(\boldsymbol{x}) = -\sum_{i=1}^{D}\sum_{k=1}^{K} \underbrace{\gamma_{i,k} \log P(x_i, z_{i,k}|\psi_{i,k})}_{\text{intra-cluster loss}} - \underbrace{(1-\gamma_{i,k}) D_{KL}[P(x_i)||P(x_i|\psi_{i,k}, z_{i,k})]}_{\text{inter-cluster loss}}. \quad (3)$$

The intra-cluster term is identical to (2), which credits each component for accurately representing pixels that have been assigned to it. The inter-cluster term ensures that each representation only captures the information about the pixels that have been assigned to it. A more powerful variant of N-EM can be obtained (RNN-EM) by substituting the generalized M-step with a recurrent neural network having hidden state $\boldsymbol{\theta}_k$. In this case the entirety of f_ϕ consists of a recurrent encoder-decoder architecture that receives $\boldsymbol{\gamma}_k(\boldsymbol{x} - \boldsymbol{\psi}_k)$ as input at each step.

The learning objective in (3) is prone to trivial solutions in case of overcapacity, which prevents the network from modelling the statistical regularities in the data that correspond to objects. By adding noise to the input image, or reducing $\boldsymbol{\theta}$ in dimension we can guide the network to consider statistical structure that corresponds to objects. Additionally, RNN-EM can be applied to sequential data and guided to learn object-representations that capture dynamics by evaluating (3) at the next time-step (*predictive coding*). Another interpretation of the denoising and next-step prediction objective is to guide the network to learn about essential properties of objects, in this case those that correspond to the Gestalt Principles of *prägnanz* and *common fate* (Hatfield & Epstein, 1985).

2.2 Relational Neural Expectation Maximization

RNN-EM (unlike N-EM) is able to capture the dynamics of objects through a parametrized recurrent connection that operates on the object representation $\boldsymbol{\theta_k}$ across consecutive time-steps. However, the relations and interactions that take place *between* objects can not be captured in this way. In order to overcome this short-coming we propose *Relational N-EM* (R-NEM), which adds relational

[1] We can not compute $\text{argmax}_\theta \, \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$ analytically, due to non-linearity of f_ϕ.

structure to the recurrence to model interactions between objects without violating key properties of the learned object representations.

Consider a generalized form of the standard RNN-EM dynamics equation, which computes the object representation $\boldsymbol{\theta}_k$ at time t as a function of all object representations $\boldsymbol{\theta} := [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K]$ at the previous time-step through an *interaction function* Υ:

$$\boldsymbol{\theta}_k^{(t)} = \text{RNN}(\tilde{\boldsymbol{x}}^{(t)}, \Upsilon_k(\boldsymbol{\theta}^{(t-1)})) := \sigma(\boldsymbol{W} \cdot \tilde{\boldsymbol{x}}^{(t)} + \boldsymbol{R} \cdot \Upsilon_k(\boldsymbol{\theta}^{(t-1)})). \tag{4}$$

Here, $\boldsymbol{W}, \boldsymbol{R}$ are weight matrices, σ is the sigmoid activation function, and $\tilde{\boldsymbol{x}}^{(t)}$ is the input to the recurrent model at time t (possibly transformed by an encoder). This dynamics model coincides with a standard RNN update rule when $\Upsilon_k^{\text{RNN-EM}}(\boldsymbol{\theta}) := \boldsymbol{\theta}_k$, such that the original RNN-EM is recovered.

The inductive bias incorporated in Υ reflects the modeling assumptions about the interactions between objects in the environment, and therefore the nature of $\boldsymbol{\theta}_k$'s interdependence. If Υ incorporates the assumption that no interaction takes place between objects, then the $\boldsymbol{\theta}_k$'s are fully independent and we recover $\Upsilon^{\text{RNN-EM}}$. On the other hand, if we do assume that interactions among objects take place, but assume very little about the structure of the interdependence between the $\boldsymbol{\theta}_k$'s, then we forfeit useful properties of $\boldsymbol{\theta}_k$, such as compositionality. For example if $\Upsilon := \text{MLP}(\boldsymbol{\theta})$ we can no longer extrapolate learned knowledge to environments with more or fewer than K objects and lose overall data efficiency (Santoro et al., 2017). Instead, we can make efficient use of compositionality among the learned object-representations $\boldsymbol{\theta}_k$ to incorporate general but guiding constraints on how these may influence one another (Battaglia et al., 2016; Chang et al., 2016). In doing so we constrain Υ to capture interdependence between $\boldsymbol{\theta}_k$'s in a compositional manner that enables physical dynamics to be learned efficiently, and allow for learned dynamics to be extrapolated to a variable number of objects.

We propose a parametrized interaction function $\Upsilon^{\text{R-NEM}}$ that incorporates these modeling assumptions and updates $\boldsymbol{\theta}_k$ based on the pairwise effects of the objects $i \neq k$ on k:

$$\begin{aligned} \Upsilon_k^{\text{R-NEM}}(\boldsymbol{\theta}) &= [\hat{\boldsymbol{\theta}}_k; \boldsymbol{E}_k] \;\text{ with }\; \hat{\boldsymbol{\theta}}_k = \text{MLP}^{enc}(\boldsymbol{\theta}_k) \;\;,\;\; \boldsymbol{E}_k = \sum_{i \neq k} \alpha_{k,i} \cdot \boldsymbol{e}_{k,i} \\ \alpha_{k,i} &= \text{MLP}^{att}(\boldsymbol{\xi}_{k,i}) \;\;,\;\; \boldsymbol{e}_{k,i} = \text{MLP}^{eff}(\boldsymbol{\xi}_{k,i}) \;\;,\;\; \boldsymbol{\xi}_{k,i} = \text{MLP}^{emb}([\hat{\boldsymbol{\theta}}_k; \hat{\boldsymbol{\theta}}_i]) \end{aligned} \tag{5}$$

where $[\cdot;\cdot]$ is the concatenation operator and $\text{MLP}^{(\cdot)}$ corresponds to a multi-layer perceptron. First, each $\boldsymbol{\theta}_i$ is transformed using MLP^{enc} to obtain $\hat{\boldsymbol{\theta}}_i$, which enables information that is relevant for the object dynamics to be made more explicit in the representation. Next, each pair $(\hat{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\theta}}_i)$ is concatenated and processed by MLP^{emb}, which computes a shared embedding $\boldsymbol{\xi}_{k,i}$ that encodes the interaction between object k and object i. Unlike other work (Battaglia et al., 2016) this function is not commutative and we opt for a clear separation between the *focus* object k and the *context* object i as in previous work (Chang et al., 2016). From $\boldsymbol{\xi}_{k,i}$ we compute $\boldsymbol{e}_{k,i}$: the effect of object i on object k; and an attention coefficient $\alpha_{k,i}$ that encodes whether interaction between object i and object k takes place. These attention coefficients (Bahdanau et al., 2014; Xu et al., 2015) help to select relevant context objects, and can be seen as a more flexible unsupervised replacement of the distance based heuristic that was used in previous work (Chang et al., 2016). Finally we compute the total effect of $\boldsymbol{\theta}_{i \neq k}$ on $\boldsymbol{\theta}_k$ as a weighted sum of the effects multiplied by their attention coefficient. A visual overview of $\Upsilon^{\text{R-NEM}}$ can be seen in Figure 1.

3 Related Work

Machine learning approaches to common-sense physical reasoning can roughly be divided in two groups: symbolic approaches and approaches that perform state-to-state prediction. The former group performs inference over the parameters of a symbolic physics engine (Battaglia et al., 2013; Ullman et al., 2017; Wu et al., 2015), which limits the to synthetic environments. The latter group employs machine learning methods to make state-to-state predictions, often describing the state of a system as a set of compact object-descriptions that are either used as an input to the system (Battaglia et al., 2016; Chang et al., 2016; Fragkiadaki et al., 2015; Grzeszczuk et al., 1998) or for training purposes (Watters et al., 2017). By incorporating information (eg. position, velocity) about objects they have achieved excellent generalization and simulation capabilities. Purely unsupervised approaches (Agrawal et al.,

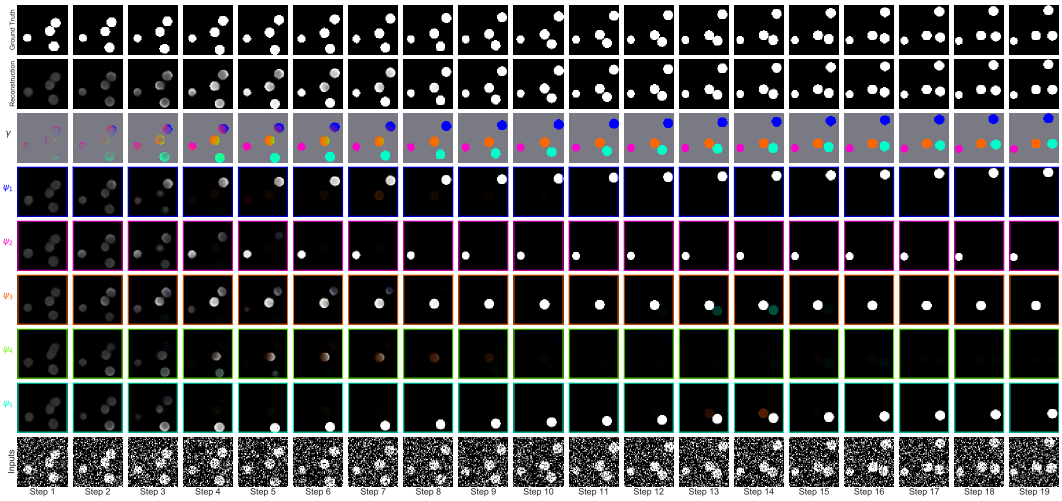

Figure 2: R-NEM applied to a sequence of 4 bouncing balls. Each column corresponds to a time-step, which coincides with an EM step. At each time-step, R-NEM computes $K = 5$ new representations $\boldsymbol{\theta}_k$, according to (4) (see also *Representations* in Figure 1) from the input $\boldsymbol{x}$ with added noise (bottom row). From each new $\boldsymbol{\theta}_k$ a group reconstruction $\boldsymbol{\psi}_k$ is produced (rows 2-6) that predicts the environment at the *next* time-step. Based on the prediction accuracy of $\boldsymbol{\psi}$, the *E-step* (see Figure 1) computes a new soft-assignment $\boldsymbol{\gamma}$ (row 7), visualized by coloring each pixel i according to their distribution over components $\boldsymbol{\gamma}_i$. Row 8 visualizes the total prediction by the network ($\sum_k \boldsymbol{\psi}_k \cdot \boldsymbol{\gamma}_k$) and row 9 the ground-truth sequence at the next time-step.

2016; Lerer et al., 2016; Sutskever et al., 2009) that use raw visual inputs as state-descriptions have yet to rival these capabilities. Our method is a purely unsupervised state-to-state prediction method that operates in pixel space and takes a first step in that direction.

R-NEM relies on N-EM (Greff et al., 2017) to discover a compositional object representation from raw visual inputs. A closely related approach to N-EM is the TAG framework (Greff et al., 2016), which utilizes a similar mechanism to perform inference over group representations and in addition performs inference over the group assignments. In recent work TAG was combined with a recurrent ladder network (Ilin et al., 2017) to obtain a powerful model (RTagger) that can be applied to sequential data. However, the lack of a single compact representation that captures all information about a group (object) makes a compositional treatment of physical interactions more difficult. Other unsupervised approaches rely on attention to group together parts of the visual scene corresponding to objects (Eslami et al., 2016; Gregor et al., 2015). They suffer from a similar problem in that the sequential nature of these models prevents a coherent representation to take shape.

Other work, has also taken steps toward combining the learnability of neural networks with the compositionality of symbolic programs in modeling physics (Battaglia et al., 2016; Chang et al., 2016), playing games (Denil et al., 2017; Kansky et al., 2017), learning algorithms (Bošnjak et al., 2017; Cai et al., 2017; Li et al., 2016; Reed & De Freitas, 2015), visual understanding (Ellis et al., 2017; Johnson et al., 2017), and natural language processing (Andreas et al., 2016; Hu et al., 2017).

4 Experiments

In this section we evaluate R-NEM on three different physical reasoning tasks that each vary in their dynamical and visual complexity: bouncing balls with variable mass, bouncing balls with an invisible curtain and the Arcade Learning Environment (Bellemare et al., 2013). We compare R-NEM to other *unsupervised* neural methods that do not incorporate any inductive biases reflecting real-world dynamics and show that these are indeed beneficial.

All experiments use ADAM (Kingma & Ba, 2014) with default parameters, on $50K$ train + $10K$ validation + $10K$ test sequences and early stopping with a patience of 10 epochs. For MLP^{enc,emb,eff} we used a single layer neural network with 250 *rectified linear* units. For MLP^{att} we used a two-layer

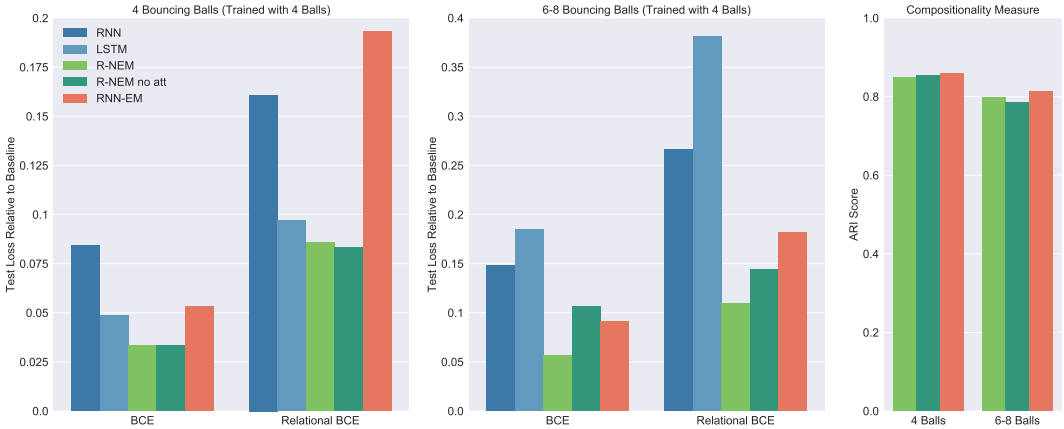

Figure 3: Performance of each method on the *balls mass* task. Each method was trained on a dataset with 4 balls and evaluated on a test set with 4 balls (left) and on a test-set with 6-8 balls (middle). The losses are reported relative to the loss of a baseline for each dataset that always predicts the current frame. The ARI score (right) is used to evaluate the degree of compositionality that is achieved.

neural network: 100 *tanh* units followed by a single *sigmoid* unit. A detailed overview of the experimental setup can be found in Appendix A.

Bouncing Balls We study the physical reasoning capabilities of R-NEM on the *bouncing balls* task, a standard environment to evaluate physical reasoning capabilities that exhibits low visual complexity and complex non-linear physical dynamics. We train R-NEM on sequences of 64×64 binary images over 30 time-steps that contain four bouncing balls with different masses corresponding to their radii. The balls are initialized with random initial positions, masses and velocities. Balls bounce elastically against each other and the image window.

Qualitative Evaluation Figure 1 presents a qualitative evaluation of R-NEM on the bouncing balls task. After 10 time-steps it can be observed that the pixels that belong to each of the balls are grouped together and assigned to a unique component (with a saturated color), and that the background (colored grey) has been divided among all components (resulting in a grey coloring). This indicates that the representation $\boldsymbol{\theta}_k$ from which each component produces the group reconstruction $\boldsymbol{\psi}_k$ does indeed only contain information about a unique object, such that together the $\boldsymbol{\theta}_k$'s yield a compositional object-representation of the scene. The total reconstruction (that combines the group reconstructions and the soft-assignments) displays an accurate reconstruction of the input sequence at the next time-step, indicating that R-NEM has learn to model the dynamics of bouncing balls.

Comparison We compare the modelling capabilities of R-NEM to an RNN, LSTM (Hochreiter & Schmidhuber, 1997) and RNN-EM in terms of the Binomial Cross-Entropy (BCE) loss between the predicted image and the ground-truth image of the last frame, as well as the *relational* BCE that only takes into objects that currently take part in collision.[2] On a test-set with sequences containing four balls we observe that R-NEM produces markedly lower losses when compared to all other methods (left plot in Figure 3).

Moreover, in order to validate that each component captures only a single ball (and thus compositionality is achieved), we report the Adjusted Rand Index (ARI; Hubert & Arabie (1985)) score between the soft-assignments $\boldsymbol{\gamma}$ and the ground-truth assignment of pixels to objects. In the left column of the ARI plot (right side in Figure 3) we find that R-NEM achieves an ARI score of 0.8, meaning that in roughly 80% of the cases each ball is modeled by a single component. This suggests that a compositional object-representation is achieved for most of the sequences. Together these observations are in line with our qualitative evaluation and validate that incorporating real world

[2]Since the E-step in R-NEM and RNN-EM utilizes the ground-truth for reconstruction, we substitute it with a simple $\max$ operator. The resulting loss serves as an *upperbound* to the true BCE loss.

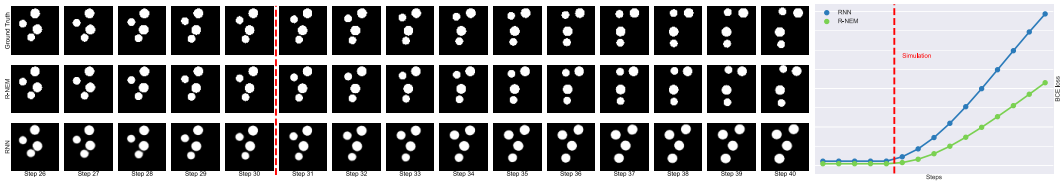

Figure 4: **Left**: Three sequences of 15 time-steps ground-truth (top), R-NEM (middle), RNN (bottom). The last ten time-steps of the sequences produced by R-NEM and RNN are simulated. **Right**: The BCE loss on the entire test-set for these same time-steps.

priors is greatly beneficial (comparing to RNN, LSTM) and that (when comparing to RNN-EM in terms of the relational BCE) $\Upsilon^{\text{R-NEM}}$ enables interactions to be modelled accurately.

Extrapolating learned knowledge We use a test-set with sequences containing 6-8 balls summarize the ability of each method to *extrapolate* their learned knowledge about physical interactions between four balls to environments with more balls. As can be seen from the middle plot in Figure 3 that R-NEM again greatly outperforms all other methods. Notice, that since we report performance relative to a baseline, we roughly factor out the increased complexity of the task. Perfect extrapolation of the learned knowledge would therefore amount to *no change* in relative performance. For the LSTM we observe far worse performance (relative to the baseline) when evaluated on this dataset with extra balls. It suggests that the gating mechanism of the LSTM has allowed it to learn a very sophisticated solution for sequences with four balls that does not generalize to dataset with 6-8 balls.

R-NEM and RNN-EM scale far better to this dataset than LSTM. Notice that although the RNN also suffers to a lesser extend from this type of "overfitting", this is most likely due its inability to produce a reasonable solution in the first place. Therefore we conclude that the compositional representations that RNN-EM and R-NEM frequently produce (see right side of the right plot in Figure 3) yields superior extrapolation capabilities.

Attention Further insight in the role of the attention mechanism can be gained by visualizing the attention coefficients, as is done in Figure 2. For each component k we draw $\alpha_{k,i} * \boldsymbol{\psi}_i$ on top of the reconstruction $\boldsymbol{\psi}_k$, colored according to the color of component i. These correspond to the colored balls (that are for example seen in time-steps 13, 14), which indicate whether component k took information about component i into account when computing the new state (recall (5)). It can be observed that the attention coefficient $\alpha_{k,i}$ becomes non-zero whenever collision takes place, such that a colored ball lights up in the *following* time-steps. The attention mechanism learned by R-NEM thus assumes the role of the distance-based heuristic in previous work (Chang et al., 2016), matching our own intuitions of how it should be utilized.

A quantitative evaluation of the attention mechanism is obtained by comparing R-NEM to a variant of itself that does not incorporate attention (*R-NEM no att*). Figure 3 shows that both methods perform equally well on the regular test set (4 balls), but that *R-NEM no att* fails to extrapolate the learned knowledge (6-8 balls). A likely reason for this behavior is that the range of the sum changes with K. When extrapolating to an environment with more balls, the total sum may exceed previous boundaries and impede learned dynamics.

Simulation Once a scene has been accurately modelled, R-NEM can approximately simulate its dynamics through recursive application of (4) for each $\boldsymbol{\theta}_k$.[3] In Figure 4 we compare the simulation capabilities of R-NEM to an RNN on the bouncing balls environment. On the left it shows for each method a sequence with five normal steps followed by 10 simulation steps, as well as the ground-truth sequence. From the last frame in the sequence it can clearly be observed that R-NEM has managed to accurately simulate the environment. Each ball is approximately in the correct place, and the shape of each ball is preserved. The balls simulated by the RNN, on the other hand, deviate substantially from their ground-truth position and their size has increased. In general we find that R-NEM produces mostly very accurate simulations, whereas the RNN consistently fails.Interestingly we found that the

[3]Note that in this case the input to the neural network encoder in component k corresponds to $\boldsymbol{\gamma}_k(\boldsymbol{x}^{(t)} - \boldsymbol{\psi}^{(t-1)})$, such that the output of the encoder $\tilde{\boldsymbol{x}}^{(t)} \approx \mathbf{0}$ when $\boldsymbol{\psi}_k^{(t-1)} = \boldsymbol{x}^{(t)}$.

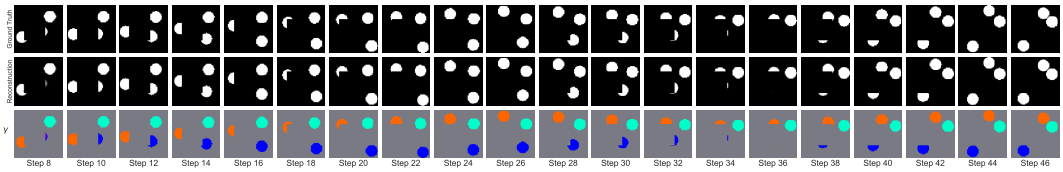

Figure 5: R-NEM applied to a sequence of bouncing balls with an invisible curtain. The ground truth sequence is displayed in the top row, followed by the prediction of R-NEM (middle) and the soft-assignments of pixels to components (bottom). R-NEM models objects, as well as its interactions, even when the object is completely occluded (step 36). Only a subset of the steps is shown.

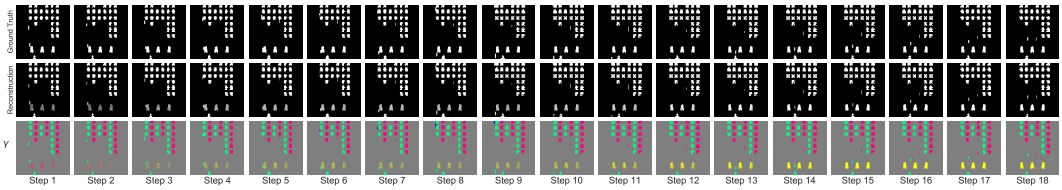

Figure 6: R-NEM accurately models a sequence of frames obtained by an agent playing Space Invaders. An group no longer corresponds to an object, but instead assumes the role of high-level entities that engage in similar movement patterns.

cases in which R-NEM fails are those for which a single component models more than one ball. The right side of Figure 4 summarizes the BCE loss for these same time-steps across the *entire* test-set. Although this is a crude measure of simulation performance (since it does not take into account the identity of the balls), we still observe that R-NEM consistently outperforms an RNN.

Hidden Factors Occlusion is abundant in the real world, and the ability to handle hidden factors is crucial for any physical reasoning system. We therefore evaluate the capability of R-NEM to handle occlusion using a variant of bouncing balls that contain an invisible "curtain". Figure 5 shows that R-NEM accurately models the sequence and can maintain object states, even when confronted with occlusion. For example, note that in step 36 the "blue ball", is completely occluded and is about to collide with the "orange ball". In step 38 the ball is accurately predicted to re-appear at the bottom of the curtain (since collision took place) as opposed to the left side of the curtain. This demonstrates that R-NEM has a notion of *object permanence* and implies that it understands a scene on a level beyond pixels: it assigns persistence and identity to the objects.

Space Invaders To test the performance of R-NEM in a more challenging environment, we train it on sequences of 84×84 binarized images over 25 time-steps of game-play on Space Invaders from the Arcade Learning Environment (Bellemare et al., 2013)[4] We use $K = 4$ and also feed the action of the agent to the interaction function. Figure 6 confirms that R-NEM is able to accurately model the environment, even though the visual complexity has increased. Notice that these visual scenes comprise a large numbers of (small) primitive objects that behave similarly. Since we trained R-NEM with four components it is unable to group pixels according to individual objects and is forced to consider a different grouping. We find that R-NEM groups together different columns of aliens together with the spaceship and three large "shields". These seem to be grouped based on their movement, which to some degree coincides with their semantic roles of the environment. In other examples (not shown) we found that R-NEM frequently groups every other column of the aliens and the three large "shields" separately. Individual bullets and the space ship are less frequently grouped separately, which may have to do with the action-noise of the environment (that controls the movement of the space-ship) and the small size of the bullets at the current resolution that makes sense less predictable.

[4]Binarization ensures that the color group of the entities on the screen does not give away their grouping.

5 DISCUSSION AND CONCLUSION

We have argued that the ability to discover and describe a scene in terms of objects provides an essential ingredient for common-sense physical reasoning. This is supported by converging evidence from cognitive science (Ullman et al., 2017) and developmental psychology that intuitive physics and reasoning capabilities are built upon the ability to perceive objects and their interactions Spelke (1988). The fact that young infants already exhibit this ability, may even suggest an innate bias towards compositionality (Lake et al., 2016; Munakata et al., 1997; Spelke & Kinzler, 2007). Inspired by these observations we have proposed R-NEM, a method that incorporates inductive biases about the existence of objects and interactions, implemented by its clustering objective and interaction function respectively. The specific nature of the objects, and their dynamics and interactions can then be learned efficiently just from visual observation.

In our experiments find that R-NEM indeed captures the (physical) dynamics of various environments more accurately than other methods, and that it exhibits improved generalization to environments with different numbers of objects. It can be used as an approximate simulator of the environment, and to predict movement and collisions of objects, even when they are completely occluded. This demonstrates a notion of object permanence and aligns with evidence that young infants seem to infer that occluded objects move in connected paths and continue to maintain object-specific properties (Spelke, 1990). Moreover, young infants also appear to expect that objects only interact when they come into contact (Spelke, 1990), which is analogous to the behaviour of R-NEM to only attend to other objects when a collision is imminent. In summary, we believe that our method presents an important step towards learning a more human-like model of the world in a completely unsupervised fashion.

These capabilities could benefit any intelligent agent that needs to operate in a physical environment, with provides exciting future research opportunities. For example, a reinforcement learner could benefit from the compositional representations produced by R-NEM to learn a modular policy that easily generalizes to novel combinations of known objects. Another example is to utilize the *causal* knowledge of the environment captured by R-NEM to foresee the consequences of ones actions. R-NEM itself could also benefit from top-down feedback to produce groupings, that are more targeted to the task at hand and may deviate from the built-in inductive biases.

REFERENCES

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems*, pp. 5074–5082, 2016.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 39–48, 2016.

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

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and others. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pp. 4502–4510, 2016.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.(JAIR)*, 47:253–279, 2013.

Matko Bošnjak, Tim Rocktäschel, Jason Naradowsky, and Sebastian Riedel. Programming with a differentiable forth interpreter. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 547–556, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/bosnjak17a.html.

Jonathon Cai, Richard Shin, and Dawn Song. Making neural programming architectures generalize via recursion. *arXiv preprint arXiv:1704.06611*, 2017.

Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society.*, pp. 1–38, 1977.

Misha Denil, Sergio Gómez Colmenarejo, Serkan Cabi, David Saxton, and Nando de Freitas. Programmable agents. *arXiv preprint arXiv:1706.06383*, 2017.

Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Joshua B. Tenenbaum. Learning to infer graphics programs from hand-drawn images. *CoRR*, abs/1707.09627, 2017. URL http://arxiv.org/abs/1707.09627.

SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Juergen Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems*, pp. 4484–4492, 2016.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural Expectation Maximization. *arXiv:1708.03498 [cs, stat]*, August 2017.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 9–20. ACM, 1998.

Gary Hatfield and William Epstein. The status of the minimum principle in the theoretical analysis of visual perception. *Psychological Bulletin*, 97(2):155, 1985.

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

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. *arXiv preprint arXiv:1704.05526*, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

John E. Hummel, Keith J. Holyoak, Collin Green, Leonidas AA Doumas, Derek Devnich, Aniket Kittur, and Donald J. Kalar. A solution to the binding problem for compositional connectionism. In *Compositional Connectionism in Cognitive Science: Papers from the AAAI Fall Symposium, Ed. SD Levy & R. Gayler*, pp. 31–34, 2004.

Alexander Ilin, Isabeau Prémont-Schwarz, Tele Hotloo Hao, Antti Rasmus, Rinu Boney, and Harri Valpola. Recurrent Ladder Networks. *arXiv:1707.09219 [cs, stat]*, July 2017.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. *arXiv preprint arXiv:1705.03633*, 2017.

Ken Kansky, Tom Silver, David A Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, Scott Phoenix, and Dileep George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *arXiv preprint arXiv:1706.04317*, 2017.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pp. 1–101, 2016.

Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*, 2016.

Chengtao Li, Daniel Tarlow, Alexander L Gaunt, Marc Brockschmidt, and Nate Kushman. Neural program lattices. 2016.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Yuko Munakata, James L McClelland, Mark H Johnson, and Robert S Siegler. Rethinking infant knowledge: toward an adaptive process account of successes and failures in object permanence tasks. *Psychological review*, 104(4):686, 1997.

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and Checkerboard Artifacts. *Distill*, 2016. doi: 10.23915/distill.00003.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *arXiv:1709.07871 [cs, stat]*, September 2017.

Scott Reed and Nando De Freitas. Neural programmer-interpreters. *arXiv preprint arXiv:1511.06279*, 2015.

Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.

Elizabeth S Spelke. Where perceiving ends and thinking begins: The apprehension of objects in infancy. 1988.

Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, pp. 843–852, 2015.

Ilya Sutskever, Geoffrey E. Hinton, and Graham W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2009.

Tomer D Ullman, Elizabeth Spelke, Peter Battaglia, and Joshua B Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in Cognitive Sciences*, 21(9):649–665, 2017.

Nicholas Watters, Andrea Tacchetti, Theophane Weber, Razvan Pascanu, Peter Battaglia, and Daniel Zoran. Visual Interaction Networks. *arXiv:1706.01433 [cs]*, June 2017.

Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4):339–356, 1988.

Ronald J Williams. Complexity of exact gradient computation algorithms for recurrent neural networks. Technical report, Technical Report Technical Report NU-CCS-89-27, Boston: Northeastern University, College of Computer Science, 1989.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in neural information processing systems*, pp. 127–135, 2015.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pp. 2048–2057, 2015.

A EXPERIMENT DETAILS

In all experiments we train the networks using ADAM Kingma & Ba (2014) with default parameters, a batch size of 64 and 50 000 train + 10 000 validation + 10 000 test inputs. The quality of the learned groupings is evaluated by computing the Adjusted Rand Index (ARI; Hubert & Arabie (1985)) with respect to the ground truth, while ignoring the background and overlap regions (as is consistent with earlier work Greff et al. (2017)). We use early stopping when the validation loss has not improved for 10 epochs.

A.1 BOUNCING BALLS

The bouncing balls data is similar to previous work Sutskever et al. (2009) with a few modifications. The data consists of sequences of 64×64 binary images over 30 time-steps and balls are randomly sampled from two types: one ball is six times heavier and 1.25 times larger in radius than the other. The balls are initialized with random initial positions and velocities. Balls bounce elastically against each other and the image window.

As in previous work Greff et al. (2017) we use a convolutional encoder-decoder architecture with a recurrent neural network as bottleneck, that is updated according to (4):

1. 4×4 conv. 16 ELU. stride 2. layer norm
2. 4×4 conv. 32 ELU. stride 2. layer norm
3. 4×4 conv. 64 ELU. stride 2. layer norm
4. fully connected. 512 ELU. layer norm
5. recurrent. 250 Sigmoid. layer norm on the output
6. fully connected. 512 RELU. layer norm
7. fully connected. $8 \times 8 \times 64$ RELU. layer norm
8. 4×4 reshape 2 nearest-neighbour, conv. 32 RELU. layer norm
9. 4×4 reshape 2 nearest-neighbour, conv. 16 RELU. layer norm
10. 4×4 reshape 2 nearest-neighbour, conv. 1 Sigmoid

Instead of using transposed convolutions (to implement the "de-convolution") we first reshape the image using the default nearest-neighbour interpolation followed by a normal convolution in order to avoid frequency artifacts Odena et al. (2016). Note that we do not add layer norm on the recurrent connection.

At each timestep t we feed $\gamma_{:,k}(\boldsymbol{\psi}_{:,k}^{(t-1)} - \hat{\boldsymbol{x}}^{(t)})$ as input to the network, where $\tilde{\boldsymbol{x}}$ is the input with added bitflip noise ($p = 0.2$). Consistent with earlier work Greff et al. (2017) R-NEM is trained with a next-step prediction objective, the prior for each pixel in the data is set to a Bernoulli distribution with $p = 0$, and we prevent conflicting gradient updates by not back-propagating any gradients through γ.

The Interaction Function $\boldsymbol{\Upsilon}^{\text{R-NEM}}$network is structured as follows:

- MLP^{enc}: fully connected. 250 RELU. layer norm
- MLP^{emb}: fully connected. 250 RELU. layer norm
- MLP^{eff}: fully connected. 250 RELU. layer norm
- MLP^{att}: fully connected. 100 Tanh. layer norm - fully connected. 1 Sigmoid.

We experimented with deeper architectures, but were unable to observe significant improvement.

Comparison and Extrapolation In the comparison experiment both R-NEM and RNN-EM are trained with $K = 5$, following insights from Greff et al. (2017). On the extrapolation task we adjust the number of components at test time to $K = 8$

When comparing to RNN-EM we use $\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^{\text{RNN-EM}}$. For comparing to RNN we take $K = 1$, and use $\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}^{\text{RNN-EM}}$, which yields a standard recurrent autoencoder that receives at each time-step

the difference between the prediction and the noisy ground-truth as input. In case of LSTM, we additionally replace the recurrent layer with an LSTM update. The R-NEM *no att* model is the same as R-NEM, without MLP^{att}, such that $\alpha_{:,:} = 1$

Simulation Since the E-step relies on the ground-truth, which was not available for simulation, we used a thresholded version of $\max_k \boldsymbol{\psi}$ at 0.1 (such that everything below becomes 0 and everything above becomes 1) as a replacement in stead.

Occlusion On the occlusion dataset we used three balls with a single mass. The curtain was spawned at a random location for each sequence.

A.2 SPACE INVADERS

We used a pre-trained DQN to produce a dataset with sequences of 25 time-steps. The DQN receives a stack of four frames as input and we recorded every first frame of this stack. These frames are first pre-processed according to Mnih et al. (2013) and then we threshold the frames at 0.0001 to obtain binary images.

Since the images are 84×84 we used a different encoder and decoder, given by:

1. 4×4 conv. 16 ELU. stride 2. layer norm
2. 4×4 conv. 32 ELU. stride 2. layer norm
3. 4×4 conv. 32 ELU. stride 2. layer norm
4. 4×4 conv. 32 ELU. stride 2. layer norm
5. fully connected. 512 ELU. layer norm
6. recurrent. 250 Sigmoid. layer norm on the output
7. fully connected. 512 RELU. layer norm
8. fully connected. $8 \times 8 \times 64$ RELU. layer norm
9. 4×4 reshape 2 nearest-neighbour, conv. 32 RELU. layer norm
10. 4×4 reshape 2 nearest-neighbour, conv. 32 RELU. layer norm
11. 4×4 reshape 2 nearest-neighbour, conv. 16 RELU. layer norm
12. 4×4 reshape 2 nearest-neighbour, conv. 1 Sigmoid

We used the same architecture for $\Upsilon^{\text{R-NEM}}$, with the only difference that at each time-step we concatenated an embedding of the action produced by the agent to the hidden state. We used a single layer MLP with 10 units and a *ReLU* activation function.

In the Atari experiment we trained with $K = 4$ and reduced the input noise to 0.02, in order to preserve tiny elements such as bullets (that only occupy 1-2 pixels).