PROVIDE: A Probabilistic Framework for Unsupervised Video Decomposition

Polina Zablotskaia[1,2,3*] **Edoardo A. Dominici**[2] **Leonid Sigal**[1,2,3,4] **Andreas M. Lehrmann**[1]

[1] Borealis AI, Vancouver, Canada
[2] University of British Columbia, Vancouver, Canada
[3] Vector Institute for Artificial Intelligence, Toronto, Canada
[4] CIFAR AI Chair

Abstract

Unsupervised multi-object scene decomposition is a fast-emerging problem in representation learning. Despite significant progress in static scenes, such models are unable to leverage important dynamic cues present in videos. We propose PROVIDE, a novel unsupervised framework for PRObabilistic VIdeo DEcomposition based on a temporal extension of iterative inference. PROVIDE is powerful enough to jointly model complex individual multi-object representations and explicit temporal dependencies between latent variables across frames. This is achieved by leveraging 2D-LSTM, temporally conditioned inference and generation within the iterative amortized inference for posterior refinement. Our method improves the overall quality of decompositions, encodes information about the objects' dynamics, and can be used to predict trajectories of each object separately. Additionally, we show that our model has a high accuracy even without color information. We demonstrate the decomposition capabilities of our model and show that it outperforms the state-of-the-art on several benchmark datasets, one of which was curated for this work and will be made publicly available.

1 INTRODUCTION

Decomposition describes the task of separating a scene into a collection of objects with individual representations, similar to how humans break down scenes into a set of abstract building blocks with their own properties. Learning to perceive the world as a collection of individual components (objects) with their own latent representations brings us closer to human perception and constitutes a natural ability of an intelligent vision system [Johnson et al., 2017]. *Unsupervised* learning of visual object representations is invaluable for extending the generality and interpretability of such models, enabling compositional reasoning [Garnelo et al., 2016, Lake et al., 2017] and transferability [Erhan et al., 2010]. However, learning rich *video* representations that, agnostic to occlusion and object quantities, can decouple object appearance and shape in complex visual scenes containing multiple moving objects has remained elusive.

Recent works that attempt to address this challenge can be characterized as: (i) attention-based methods [Eslami et al., 2016, Crawford and Pineau, 2019b], which infer latent representations for each object in a scene using attention, and (ii) iterative refinement models [Greff et al., 2017, 2019], which decompose a scene into a collection of components by grouping pixels. Importantly, the former have been limited to latent representations at object- or image patch-levels, while the latter class of models have illustrated the ability for more granular latent representations at the pixel (segmentation)-level. Specifically, most refinement models learn pixel-level generative models driven by spatial mixtures [Greff et al., 2017] and utilize amortized iterative refinements [Marino et al., 2018] for inference of disentangled[1] latent representations within the VAE framework [Kingma and Welling, 2014]; a prime example is IODINE [Greff et al., 2019]. However, while providing a powerful model and abstraction which is able to segment and disentangle complex scenes, IODINE [Greff et al., 2019] and other similar architectures are fundamentally limited by the fact that they only consider images. Even when applied for inference in video, they process one frame at a time. This makes it excessively challenging to discover and represent individual instances of objects that may share properties such as appearance and shape but differ in dynamics.

[*] Now at Google Research, Berlin, Germany

[1] *Disentanglement* refers to a model's ability to learn interpretable latent factors; this is in contrast to *decomposition* which only requires separation but does not necessitate interperability. Ideally, however, one desires disentangled decomposition where each dimension of an object's latent representation encodes a semantically meaningful factor, such as color, position, size, etc.

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

In computer vision, it has been a long-held belief that motion carries important information for decomposing a scene with many objects [Weiss and Adelson, 1996, Jepson et al., 2002]. Armed with this intuition, we propose a spatio-temporal amortized inference model capable of not only unsupervised multi-object scene decomposition, but also of learning and leveraging the implicit probabilistic dynamics of each object from raw video alone. This is achieved by introducing temporal dependencies between the latent variables across time. As such, IODINE [Greff et al., 2019] could be considered a special (spatial) case of our spatio-temporal formulation. Modeling temporal dependencies among video frames also allows us to make use of conditional priors [Chung et al., 2015] for variational inference, leading to more accurate and efficient inference.

The proposed framework for `PRO`babilistic `VI`deo `DE`composition (PROVIDE)[2], illustrated in Fig. 1, achieves superior performance on complex multi-object benchmark datasets (Bouncing Balls and CLEVRER) with respect to state-of-the-art models, including R-NEM [Van Steenkiste et al., 2018] and IODINE [Greff et al., 2019], in terms of decomposition, prediction, and generalization. PROVIDE has a number of appealing properties, including temporal extrapolation, computational efficiency, and the ability to work with complex data exhibiting non-linear dynamics, colors, and changing number of objects within the same video sequence. In addition, we introduce an entropy prior to improve our model's performance in scenarios where object appearance alone is not sufficiently distinctive (*e.g.*, greyscale data).

2 RELATED WORK

Unsupervised Scene Representation Learning. Unsupervised scene representation learning can generally be divided into two groups: attention-based methods, which infer latent representations for each object in a scene using attention mechanisms, and more complex and powerful iterative refinement models, which often make use of spatial mixtures and can decompose a scene into a collection of estimated components by grouping pixels together. *Attention-based* methods, such as AIR [Eslami et al., 2016, Xu et al., 2019] and SPAIR [Crawford and Pineau, 2019b], decompose scenes into latent variables representing the appearance, position, and size of the underlying objects. However, both methods can only infer the objects' bounding boxes and have not been shown to work on non-trivial 3D scenes with perspective distortions and occlusions. MoNet [Burgess et al., 2019] is the first model in this family tackling more complex data and inferring representations that can be used for precise and granular decomposition of objects. On the other hand, it is not a probabilistic generative model and thus not suitable for density estimation. GENESIS [Engelcke et al., 2020] extends it and alleviates some of its limitations by introducing a probabilistic framework and allowing for spatial relations between the objects. DDPAE [Hsieh et al., 2019] is a framework that uses structured probabilistic models to decompose a video into low-dimensional temporal dynamics with the sole purpose of prediction. It is shown to operate on binary scenes with no perspective distortion and is not capable of separating objects from each other well enough. *Iterative refinement* models started with Tagger [Greff et al., 2016], which reasons about the perceptual grouping of its inputs. However, it does not allow explicit latent representations and cannot be scaled to more complex images. NEM [Greff et al., 2017], as an extension of Tagger, uses a spatial mixture model inside an expectation maximization framework but is limited to binary data. Finally, IODINE [Greff et al., 2019] is a notable example of a model employing iterative amortized inference w.r.t. a spatial mixture formulation and achieves state-of-the-art performance in scene decomposition.

Unsupervised Video Tracking and Object Detection. SQAIR [Kosiorek et al., 2018], SILOT [Crawford and Pineau, 2019a] and SCALOR [Jiang et al., 2020] are temporal extensions of the static attention-based models that are tailored to tracking and object detection tasks. SQAIR is restricted to binary data and operates at the level of bounding boxes. SILOT and SCALOR are more expressive and can cope with cluttered scenes, a larger numbers of objects, and dynamic backgrounds, but they do not work on colored perspective[3] data; accurate segmentation remains a challenge.

Unsupervised Video Decomposition. Models employing spatial mixtures and iterative inference in a temporal setting are, from a technical perspective, closest to the proposed PROVIDE. Notably, there are only few models falling into this line of work: RTagger [Prémont-Schwarz et al., 2017] is a recurrent extension of Tagger and has the same limitations as its predecessor. R-NEM [Van Steenkiste et al., 2018] effectively learns the objects' dynamics and interactions through a relational module and can produce segmentations but is limited to orthographic binary data.

Methods without Latent Modeling. GAN-based ReDO [Chen et al., 2019] uses a model built around the assumption that object regions are independent, guiding the generator by drawing the objects' pixel regions separately and composing them after segmentation. Another model [Arandjelović and Zisserman, 2019] employs the same principles but guide the generator by copying a region of an image into another one. Both architectures are shown to operate on static images only and do not have a clearly interpretable latent space or prediction capabilities. Unsupervised segmentation of videos, which often amounts to clustering, is an important area of research which diverges from decomposition as it does not

[2]Code: `https://github.com/BorealisAI/PROVIDE`

[3]Perspective videos are more complex as objects can occlude one another and change in size over time.

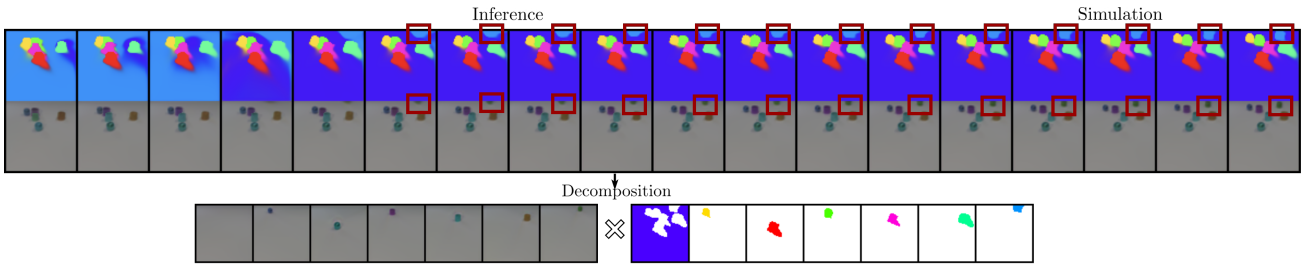

Figure 1: **Unsupervised Video Decomposition.** PROVIDE allows to infer precise decompositions of the objects via interpretable latent representations that can be used to decompose each frame and simulate future dynamics, all in an unsupervised fashion. Whenever a new object enters a frame, the model adapts and assigns one of the slots to the new object.

explicitly model the latent space for each object. As such, unsupervised segmentation models are unable to model or decouple object factors (e.g., instances, shape, appearance) or leverage priors over them, making them fundamentally incomparable to decomposition methods.

3 DYNAMIC VIDEO DECOMPOSITION

We now introduce PROVIDE, our dynamic model for unsupervised video decomposition. PROVIDE builds upon a generative model of multi-object representations and leverages elements of iterative amortized inference. We briefly review both concepts (§3.1) and then introduce our model (§3.2).

3.1 BACKGROUND

Multi-Object Representations. The multi-object framework introduced in [Greff et al., 2019] decomposes a static image $\mathbf{x} = (x_i)_i \in \mathbb{R}^D$ into K objects (including background). Each object is represented by a latent vector $\mathbf{z}^{(k)} \in \mathbb{R}^M$ capturing the object's unique appearance and can be thought of as an encoding of common visual properties, such as color, shape, position, and size. For each $\mathbf{z}^{(k)}$ independently, a broadcast decoder [Watters et al., 2019] generates pixelwise pairs $(m_i^{(k)}, \mu_i^{(k)})$ describing the assignment probability and appearance of pixel i for object k. Together, they induce the generative image formation model

$$p(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^{D} \sum_{k=1}^{K} m_i^{(k)} \mathcal{N}(x_i;\, \mu_i^{(k)}, \sigma^2), \qquad (1)$$

where $\mathbf{z} = (\mathbf{z}^{(k)})_k$, $\sum_{k=1}^{K} m_i^{(k)} = 1$ and σ is the same and fixed for all i and k. The original image pixels can be reconstructed from this probabilistic representation as $\widetilde{x}_i = \sum_{k=1}^{K} m_i^{(k)} \mu_i^{(k)}$.

Iterative Amortized Inference. Our approach leverages the iterative amortized inference framework [Marino et al., 2018], which uses the learning to learn principle [Andrychowicz et al., 2016] to close the amortization gap [Cremer et al., 2017] typically observed in traditional variational inference. The need for such an iterative process arises due to the multi-modality of Eq.(1), which results in an order invariance and assignment ambiguity in the approximate posterior that standard variational inference cannot overcome [Greff et al., 2019].

The idea of amortized iterative inference is to start with randomly guessed parameters $\boldsymbol{\lambda}_1^{(k)}$ for the approximate posterior $q_{\boldsymbol{\lambda}}(\mathbf{z}_1^{(k)}|\mathbf{x})$ and update this initial estimate through a series of R refinement steps. Each refinement step $r \in \{1, \dots, R\}$ first samples a latent representation from $q_{\boldsymbol{\lambda}}$ to evaluate the ELBO $\mathcal{L}$ and then uses the approximate posterior gradients $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$ to compute an additive update f_ϕ, producing a new parameter estimate $\boldsymbol{\lambda}_{r+1}^{(k)}$:

$$\begin{aligned} \mathbf{z}_r^{(k)} &\overset{k}{\sim} q_{\boldsymbol{\lambda}}(\mathbf{z}_r^{(k)}|\mathbf{x}), \\ \boldsymbol{\lambda}_{r+1}^{(k)} &\overset{k}{\leftarrow} \boldsymbol{\lambda}_r^{(k)} + f_\phi(\mathbf{a}^{(k)}, \mathbf{h}_{r-1}^{(k)}), \end{aligned} \qquad (2)$$

where $\mathbf{a}^{(k)}$ is a function of $\mathbf{z}_r^{(k)}$, $\mathbf{x}$, $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$, and additional inputs (mirrors definition in [Greff et al., 2019]). The function f_ϕ consists of a sequence of convolutional layers and an LSTM. The memory unit takes as input a hidden state $\mathbf{h}_{r-1}^{(k)}$ from the previous refinement step.

3.2 SPATIO-TEMPORAL ITERATIVE INFERENCE

Our proposed model builds upon the concepts introduced in the previous section and enables robust learning of dynamic scenes through spatio-temporal iterative inference. Specifically, we consider the task of decomposing a video sequence $\mathbf{x} = (\mathbf{x}_t)_{t=1}^T = (x_{t,i})_{t,i=1}^{T,D}$ into K slot sequences $(\mathbf{m}_t^{(k)})_t$ and K appearance sequences $(\boldsymbol{\mu}_t^{(k)})_t$. To this end, we introduce explicit temporal dependencies into the sequence of posterior refinements and show how to leverage this contextual information during decoding with a generative model. The resulting computation graph can be thought of as a 2D grid with time dimension t and refinement dimension r (Fig. 2a). Propagation of information along these two axes is achieved with a 2D-LSTM [Graves et al., 2007] (Fig. 2b), which allows us to model the joint probability over the entire video sequence inside the iterative amortized inference

framework. The proposed method is expressive enough to model the multimodality of our image formation process and posterior, yet its runtime complexity is smaller than that of its static counterpart.

3.2.1 Variational Objective

Since exact likelihood training is intractable, we formulate our task in terms of a variational objective. In contrast to traditional optimization of the evidence lower bound (ELBO) through static encodings of the approximate posterior, we incorporate information from two dynamic axes: (1) variational estimates from previous refinement steps; (2) temporal information from previous frames. Together, they form the basis for spatio-temporal variational inference via iterative refinements. Specifically, we train PROVIDE by maximizing the following ELBO objective[4]:

$$\mathcal{L}_{\text{ELBO}}(\mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\mathbf{z}_{\leq T,R}|\mathbf{x}_{\leq T})} \sum_{t=1}^{T} \sum_{r=1}^{\widehat{R}} \Big[\beta \log \left(p\left(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leq t,r}\right)\right) - \text{KL}(q_{\boldsymbol{\lambda}}(\mathbf{z}_{t,r}|\mathbf{x}_{\leq t}, \mathbf{z}_{<t,r}) \,||\, p(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})) \Big], \tag{3}$$

where the first term expresses the reconstruction error of a single frame and the second term measures the divergence between the variational posterior and the prior. The relative weight between terms is controlled with a hyperparameter β [Higgins et al., 2017]. Furthermore, to reduce the overall complexity of the model and to make it easier to train, we set $\widehat{R} := \max(R - t, 1)$ (see Fig. 2 for an illustration). Compared to a static model, which infers each frame independently, reusing information from previous refinement steps also makes our model more computationally efficient. In the next sections, we discuss the form of the conditional distributions in Eq.(3) in more detail.

3.2.2 Inference and Generation

Posterior Refinement. Optimizing Eq.(3) inside the iterative amortized inference framework (Section 3.1) requires careful thought about the nature and processing of the hidden states. While there is vast literature on the propagation of a single signal, including different types of RNNs [Hochreiter and Schmidhuber, 1997, Graves et al., 2005, Cho et al., 2014, Chung et al., 2017] and transformers [Vaswani et al., 2017], the optimal solution for multiple axes with different semantic meaning (*i.e.*, time and refinements) is less obvious.

Here, we propose to use a 2D version of the uni-directional MD-LSTM [Graves et al., 2007] to compute our variational objective (Eq.(3)) in an iterative manner. In order to do so, we replace the traditional LSTM in the refinement network (Eq.(2)) with a 2D extension. This extension allows the posterior gradients to flow through both the grid of the previous refinements and the previous time steps (see Fig. 2a). Writing $\mathbf{z}_{t,r}$ for the latent encoding at time t and refinement r, we can formalize this new update scheme as follows:

$$\begin{aligned} \mathbf{z}_{t,r} &\sim q_{\boldsymbol{\lambda}}(\mathbf{z}_{t,r}|\mathbf{x}_{\leq t}, \mathbf{z}_{<t,r}), \\ \boldsymbol{\lambda}_{t,r+1} &\leftarrow \boldsymbol{\lambda}_{t,r} + f_\phi(\mathbf{a}, \mathbf{h}_{t,r-1}, \mathbf{h}_{t-1,\widehat{R}}). \end{aligned} \tag{4}$$

Note that the hidden state from the previous time step is always $\mathbf{h}_{t-1,\widehat{R}}$, *i.e.*, the one computed during the final refinement $\widehat{R}$ at time $t-1$. Our reasoning for this is that the approximation of the posterior only improves with the number of refinements [Marino et al., 2018].

Temporal Conditioning. Inside the learning objective we set the prior and the likelihood to be conditioned on the previous frames and the refinement steps. This naturally comes from an idea that each frame is dependent on the predecessor's dynamics and therefore latent representations should follow the same property. Conditioning on the refinement steps is essential to the iterative amortized inference procedure. To model the prior and the likelihood distributions accordingly we adopt the approach proposed in [Chung et al., 2015] but tailor it to our iterative amortized inference setting. Specifically, the parameters of our Gaussian prior are now computed from the temporal hidden state $\mathbf{h}_{t-1,\widehat{R}}$:

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}) &= \mathcal{N}(\mathbf{z}_t; \widetilde{\boldsymbol{\mu}}_t, \text{diag}(\widetilde{\boldsymbol{\sigma}}_t^2)), \\ [\widetilde{\boldsymbol{\mu}}_t, \widetilde{\boldsymbol{\sigma}}_t] &= \xi_\theta(\mathbf{h}_{t-1,\widehat{R}}), \end{aligned} \tag{5}$$

where ξ_θ is a simple neural network with a few layers.[5] Please refer to the supplemental material for details. Note that the prior only changes along the time dimension and is independent of the refinement iterations, because we refine the posterior to be as close as possible to the dynamic prior for the current time step. Finally, to complete the conditional generation, we modify the likelihood distribution as follows[6]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t,r}) &= \prod_{i=1}^{D} \sum_{k=1}^{K} m_{t,r,i}^{(k)} \mathcal{N}(x_{t,i}; \mu_{t,r,i}^{(k)}, \sigma^2), \\ [m_{t,r,i}^{(k)}, \mu_{t,r,i}^{(k)}] &= g_\theta(\mathbf{z}_{t,r}^{(k)}, \mathbf{h}_{t-1,\widehat{R}}^{(k)}), \end{aligned} \tag{6}$$

where $\mu_{t,r,i}^{(k)}, m_{t,r,i}^{(k)}$ are mask and appearance of pixel i in slot k at time step t and refinement step r. g_θ is a spatial mixture broadcast decoder [Greff et al., 2019] with preceding MLP to transform the pair $\left(\mathbf{z}_{t,r}^{(k)}, \mathbf{h}_{t-1,\widehat{R}}^{(k)}\right)$ into a single vector representation.

[4] For simplicity, we drop references to the object slot $\bullet^{(k)}$ from now on and formulate all equations on a per-slot basis.

[5] In practice, ξ_θ predicts $\log \boldsymbol{\sigma}_t$ for stability reasons.

[6] Since our likelihood is a Gaussian mixture model, we are now referencing the object slot $\bullet^{(k)}$ again.

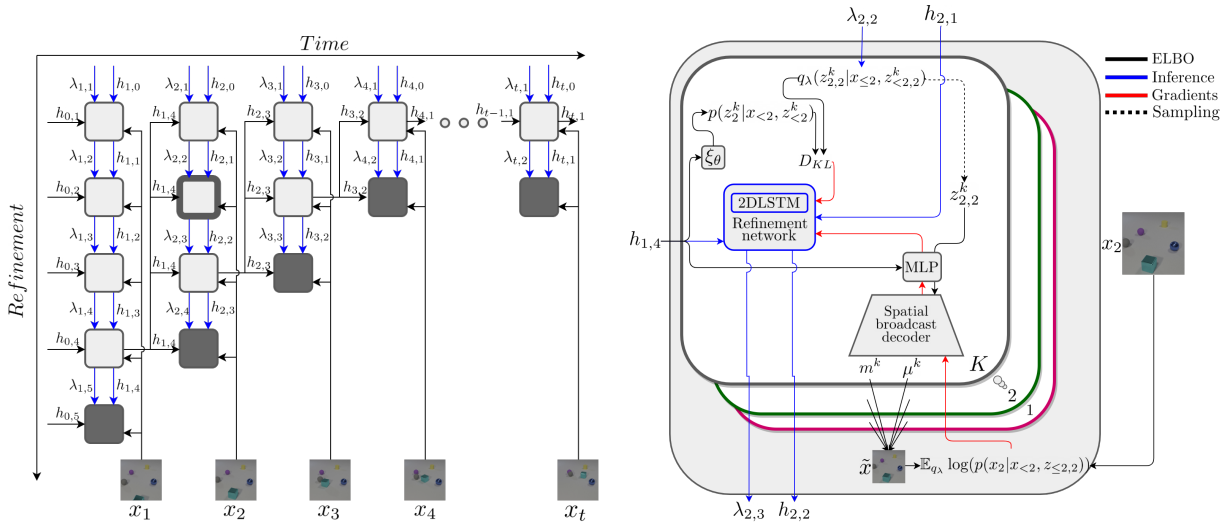

Figure 2: **Model Overview.** (**a**) Inference in PROVIDE passes through a 2D grid in which light gray cells (r, t) represent the r-th refinement at time t, dark gray cells are where the final reconstruction is computed and no refinement is needed. Each light gray cell receives three inputs: a refinement hidden state $\mathbf{h}_{t,r-1}$, a temporal hidden state $\mathbf{h}_{t-1,\widehat{R}}$, and posterior parameters $\boldsymbol{\lambda}_{t,r}$. The outputs are a new hidden state $\mathbf{h}_{t,r}$ and new posterior parameters $\boldsymbol{\lambda}_{t,r+1}$. (**b**) An example of the internal structure of the highlighted cell from Fig.(a). We process the inputs with the help of a spatial broadcast decoder and a 2D LSTM. The rest of the light gray cells have the same structure.

3.2.3 Learning and Prediction

Architecture. From a graphical point of view, we can think of the refinement steps and time steps as being organized on a 2D grid (Fig. 2a), with light gray cells (r, t) representing the r-th refinement at time t. According to Eq.(4), each such cell takes as input the hidden state from a previous refinement $\mathbf{h}_{t,r-1}$, the temporal hidden state $\mathbf{h}_{t-1,\widehat{R}}$, and the posterior parameters $\boldsymbol{\lambda}_{t,r}$. The outputs of each cell are new posterior parameters $\boldsymbol{\lambda}_{t,r+1}$ and a new hidden state $\mathbf{h}_{t,r}$. At the last refinement $\widehat{R}$ at time t, the value of the refinement hidden state $\mathbf{h}_{t,r}$ is assigned to a new temporal hidden state $\mathbf{h}_{t,\widehat{R}}$.

Training Objective. Instead of a direct optimization of Eq.(3), we propose two modifications that we found to improve PROVIDE's practical performance: (1) similar to observations made by [Greff et al., 2019], we found that color is an important factor for high-quality decompositions. In absence of such information, we mitigate the arising ambiguity by maximizing the entropy of the masks $m^{(k)}_{t,r,i}$ along the slot dimension k, *i.e.*, we train by maximizing the objective

$$\mathcal{L}_{\text{ELBO}} + \gamma \sum_{i=1}^{D} \sum_{k=1}^{K} m^{(k)}_{t,r,i} \log(m^{(k)}_{t,r,i}), \tag{7}$$

where γ defines the weight of the entropy loss. (2) In addition to the entropy loss, we also prioritize later refinement steps by weighting terms in the inner sum of Eq.(3) with $\frac{r}{\widehat{R}}$.

Prediction. On top of pure video decomposition, PROVIDE is also able to simulate future frames $\mathbf{x}_{T+1}, \ldots, \mathbf{x}_{T+T'}$. Because our model requires image data $\mathbf{x}_t$ as input, which is not available during simulation of new frames, we use the reconstructed image $\widetilde{\mathbf{x}}_t$ in place of $\mathbf{x}_t$ to compute the likelihood $p(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t,r})$ in these cases. We also set the gradients $\nabla_{\boldsymbol{\lambda}}\mathcal{L}$, $\nabla_{\boldsymbol{\mu}}\mathcal{L}$, and $\nabla_{\mathbf{m}}\mathcal{L}$ to zero.

Complexity. Our model's ability to reuse information from previous refinements leads to a runtime complexity of $\mathcal{O}(R^2 + T)$, which is much more efficient than the $\mathcal{O}(RT)$ complexity of the traditional IODINE model (where each frame is inferred independently) in a typical case of $T \gg R$.

4 EXPERIMENTS

We validate PROVIDE on Bouncing Balls [Van Steenkiste et al., 2018], an augmented version of CLEVRER [Yi et al., 2020], and Grand Central Station [Zhou et al., 2012]. Our experiments comprise quantitative studies of decomposition quality during generation and prediction, as well as an ablation study.

The quality of a decomposition can be measured by the accuracy of the individual objects' reconstructions. The more refined and precise they are, the better the model can distinguish one object from another and from the background. In practice, this accuracy can be computed by comparing the reconstructed objects with their ground truth counterparts, e.g., by extracting them from the ground truth image with the help of a segmentation mask and computing the MSE.

One complication of this approach is that there is no natural or prespecified ordering among the objects. On the other hand, our model is designed to explicitly produce individual object masks first and then combine them into a scene. Given this property, we focus on the masks and compute metrics that are agnostic to slot permutations. For the sake of clarity we refer to these mask predictions as a "segmentation task" but note that this is just a proxy to measure the quality of the scene decomposition. This evaluation process is consistent with the literature and our baselines.

We demonstrate our model's *disentanglement* properties in Appendix F.5. In contrast to decomposition, which is *explicitly* encoded into our model structure, disentanglement describes the model's *implicit* ability to learn interpretable generative factors for each object. Because disentanglement is difficult to quantify in general, we resort to qualitative experiments. While not our primary focus, disentanglement is important and we illustrate that we can maintain this property, induced by the spatial broadcast decoder [Watters et al., 2019], despite the added complexity of the temporal domain.

4.1 SETUP

Datasets. Bouncing Balls consists of 50 frame, binary, 64×64 resolution video sequences. Each video shows simulated balls with different masses bouncing elastically off each other and the image border. We train PROVIDE on the first 40 frames of 50K videos containing 4 balls in each frame. We use two different test sets consisting of 10K videos with 4 balls and 10K videos with 6-8 balls. We also validate our model on a color version of this dataset that we generate using the segmentation masks.

CLEVRER contains synthetic videos of moving and colliding objects. Each video is 5 seconds long (128 frames) and has a resolution of 480×320, which we trim and rescale to 64×64 pixels.[7] For training, we use the same 10K videos as in the original source. For testing, we compute ground truth masks for the validation set using the provided annotations and test on 2.5K instances containing 3-5 objects and on 1.1K instances containing 6 objects. In training, we set the number of slots K to 6 for CLEVRER and to one more than the maximum number of objects in all other cases.

Grand Central Station is a video feed from the main hall of a busy train station, containing a high number of people moving at various paces in different directions. It has a total of 50010 frames in a resolution of 720×480. In order to make the dataset more manageable, we have extracted a portion of the feed of resolution 128×128 and segmented it into sequences of 20 frames each. Each sequence contains approximately 10 people. We set K to 8 during training and to 10 for testing. Since the dataset does not contain ground truth segmentation masks, a quantitative evaluation was not possible.

Please refer to Appendix A for more information about the Bouncing Balls and CLEVRER datsets and Appendix B for dataset-specific hyperparameters. Qualitative results on the Grand Central Station dataset are discussed in Appendix F.4.

Baselines. We compare PROVIDE to recent baselines: R-NEM [Van Steenkiste et al., 2018], IODINE [Greff et al., 2019] and DDPAE [Hsieh et al., 2019]. R-NEM is a state-of-the-art model for unsupervised video decomposition and physics learning. While showing impressive results on simulation tasks, it is limited to binary data and has difficulties with perspective scenes. IODINE is more expressive but static in nature and cannot capture temporal dynamics within its probabilistic framework. However, as noted in [Greff et al., 2019], it can be readily applied to temporal sequences by feeding a new video frame to each iteration of the LSTM in the refinement network. We call this variant SEQ-IODINE. Since our model can also perform simulation of short sequences, we include a comparison of its predictive power against DDPAE [Hsieh et al., 2019]. Please refer to Appendix C for additional information about these baseline models.

4.2 EVALUATION METRICS

ARI. The Adjusted Rand Index [Rand, 1971, Hubert and Arabie, 1985] is a measure of clustering similarity. It is computed by counting all pairs of samples that are assigned to the same or different clusters in the predicted and true clusterings. It ranges from -1 to 1, with a score of 0 indicating a random clustering and 1 indicating a perfect match. We treat each pixel as one sample and its segmentation as the cluster assignment.

F-ARI. The Foreground Adjusted Rand Index is a modification of the ARI score ignoring background pixels, which often occupy the majority of the image. We argue that both metrics are necessary to assess the decomposition quality of a video decomposition method; this metric is also used in [Van Steenkiste et al., 2018, Greff et al., 2019].

MSE. The mean squared error between pixels of the reconstructed frames $\widehat{\mathbf{x}}$ and the ground truth frames $\mathbf{x}$.

4.3 VIDEO DECOMPOSITION

We optimize our model using ADAM [Kingma and Ba, 2014] (see Appendix D for training details) and evaluate it on a video decomposition task with different sequence lengths. As shown in Table 1, PROVIDE outperforms the baselines regardless of the presence of color information, which further reduces the error. It performs at least 7% better than R-NEM on all metrics and at least 20% better than

[7] Our method is robust enough to handle 128x128 resolution as it is built on top of IODINE.

Table 1: **Quantitative Evaluation (Scene Decomposition).** We show our model's ability to produce high-quality decomposition for sequences with varying length. We test on sequences with 4 balls and two different types of data (binary, colored) for Bouncing Balls and on sequences with 3-5 objects for CLEVRER. Note that R-NEM cannot handle color data, hence we only run it on binary data.

Bouncing Balls

		ARI (↑)				F-ARI (↑)				MSE (↓)			
	Length	10	20	30	40	10	20	30	40	10	20	30	40
binary	R-NEM	0.5031	0.6199	0.6632	0.6833	0.6259	0.7325	0.7708	0.7899	0.0252	0.0138	0.0096	0.0076
	IODINE			0.0318				0.9986				0.0018	
	SEQ-IODINE	0.0230	0.0223	0.0021	-0.0201	0.8645	0.6028	0.5444	0.4063	0.0385	0.0782	0.0846	0.0968
	PROVIDE	**0.7169**	**0.7263**	**0.7286**	**0.7294**	**0.9999**	**0.9999**	**0.9999**	**0.9999**	**0.0004**	**0.0004**	**0.0004**	**0.0004**
color	IODINE			0.5841				0.9752				0.0014	
	SEQ-IODINE	0.3789	0.3743	0.3225	0.2654	0.7517	0.8159	0.7537	0.6734	0.0160	0.0164	0.0217	0.0270
	PROVIDE	**0.7275**	**0.7291**	**0.7298**	**0.7301**	**1.0000**	**1.0000**	**0.9999**	**0.9999**	**0.0002**	**0.0002**	**0.0002**	**0.0002**

CLEVRER

		ARI (↑)				F-ARI (↑)				MSE (↓)			
	Length	10	20	30	40	10	20	30	40	10	20	30	40
color	IODINE			0.1791				**0.9316**				0.0004	
	SEQ-IODINE	0.1171	0.1378	0.1558	0.1684	0.8520	0.8774	0.8780	0.8759	0.0009	0.0009	0.0010	0.0010
	PROVIDE	**0.2220**	**0.2403**	**0.2555**	**0.2681**	0.9182	0.9258	0.9309	0.9312	**0.0003**	**0.0003**	**0.0003**	**0.0003**

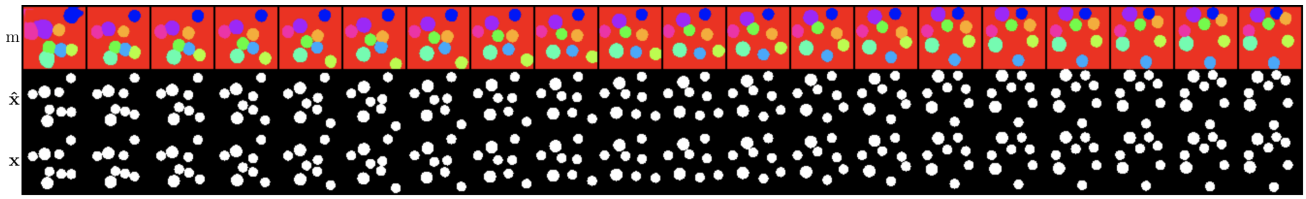

Figure 3: **Qualitative Evaluation (Bouncing Balls).** PROVIDE can generalize to sequences with 8 balls when trained on 4 balls. Top-to-bottom: output masks, reconstructions, and ground truth video.

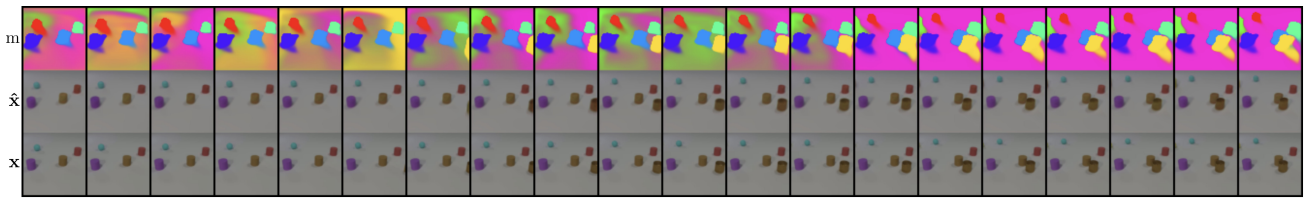

Figure 4: **Qualitative Evaluation (CLEVRER).** PROVIDE can generalize to sequences with 6 objects. We also demonstrate the ability to handle a dynamically changing number of objects, ranging from 4 in the beginning to 6 at the end.

IODINE on ARI and MSE. Since R-NEM cannot cope well with colored data or the perspective of scenes, it is only evaluated on the Bouncing Balls dataset (binary), producing high-error results in the first frames, a phenomenon not observed with PROVIDE. IODINE is not designed to utilize temporal information. On both datasets, IODINE's results are therefore computed independently on each frame of the longest sequence. By processing frames separately, IODINE does not keep the same object-slot assignment, which we ignore when computing the scores. SEQ-IODINE tends to perform even worse than IODINE in many experiments, which we attributed to exploding gradients caused by limited refinement steps and a lack of dynamics modeling. Qualitative results for IODINE and SEQ-IODINE can be found in Figure 6 in the supplementary material.

4.4 GENERALIZATION

We investigated how well PROVIDE adapts to a higher number of objects, evaluating its performance on the Bouncing Balls dataset (6 to 8 objects) and on the CLEVRER dataset (6 objects). Table 2 shows that PROVIDE's F-ARI and MSE scores are at least 50% better than those for R-NEM, while its ARI scores are only marginally worse and only on the binary data. In comparison to IODINE, our model is at least 4% better across all metrics. For the Bouncing Balls dataset, we also investigated the impact of changing the total number of possible colors to 4 and 8 (the former resulting in duplicate colors for different objects and the latter in unique colors for each object). The higher MSE scores for the 8 balls variant is due to the model not being able to reconstruct

Table 2: **Generalization**. At test-time, we change the number of slots in the models from 5 to 9 for the Bouncing Balls test dataset (6-8 balls), and from 6 to 7 for the CLEVRER test dataset (6 objects).

	Bouncing Balls			
		ARI (↑)	F-ARI (↑)	MSE (↓)
binary	R-NEM	**0.4484**	0.6377	0.0328
	IODINE	0.0271	0.9969	0.0040
	SEQ-IODINE	0.0263	0.8874	0.0521
	PROVIDE	0.4453	**0.9999**	**0.0008**
color	IODINE (4)	0.4136	0.8211	0.0138
	IODINE (8)	0.2823	0.7197	0.0281
	SEQ-IODINE (4)	0.2068	0.5854	0.0338
	SEQ-IODINE (8)	0.1571	0.5231	0.0433
	PROVIDE (4)	0.4275	**0.9998**	**0.0004**
	PROVIDE (8)	**0.4317**	0.9900	0.0114

	CLEVRER			
		ARI (↑)	F-ARI (↑)	MSE (↓)
color	IODINE	0.2205	0.9305	0.0006
	SEQ-IODINE	0.1482	0.8645	0.0012
	PROVIDE	**0.2839**	**0.9355**	**0.0004**

the unseen colors. Sample qualitative results are shown in Fig. 3 and 4, while more can be found in Appendix F.

4.5 PREDICTION

We compare the predictions of our model (Section 3.2.3) to those of R-NEM after 20 steps of inference on 10 predicted steps on the Bouncing Balls dataset (Fig. 5 left). As we can see from the results, PROVIDE is superior to R-NEM on shorter sequences, however, for longer sequences we are outperforming R-NEM only on colored data. Our model is capable of more accurate frame prediction than R-NEM on the Bouncing Balls dataset during the first few predicted frames (5-7), with predictions slowly diverging over time due to the temporal consistency. This behavior is also observable on the CLEVRER dataset (Fig. 5 right), albeit to a lesser extent, likely because the scene dynamics are simpler due to fewer moving objects, even if the motion itself is non-linear. We refer to Appendix E for an extended discussion. In Figure 6 we compute velocity vectors between bounding box centroids and compare the cosine similarity to the predictions of DDPAE on the Bouncing Balls dataset. While PROVIDE outperforms DDPAE on the first three frames, its quality falls below DDPAE performance for longer simulations. This behavior is not surprising and in line with the results in Fig. 5. We note that DDPAE uses a dedicated RNN to capture the temporal dependencies and interactions between the components of a scene (see Hsieh et al. [2019]; Figure 2). Similar to R-NEM, this allows DDPAE to learn

Table 3: **Ablation Study**. A 2D-LSTM extension of IODINE trained on sequences of 20 frames is unstable and its output segmentation lacks precision and consistency. Our efficient version of a 2D-LSTM grid (Fig. 2a) and the conditional prior and generation increase both segmentation and reconstruction quality. By training these models on longer sequences of 40 frames we observe further improvements.

	Base	Grid	CP+G	Entropy	Length	ARI (↑)	F-ARI (↑)	MSE (↓)
BB	✓				20	0.0126	0.7765	0.0340
	✓	✓	✓		20	0.2994	0.9999	0.0010
	✓	✓	✓		40	0.3528	0.9998	0.0010
	✓	✓	✓	✓	40	**0.7263**	**0.9999**	**0.0004**
CLEVRER	✓				20	0.1900	0.8200	0.0011
	✓	✓			20	0.1100	0.9000	0.0005
	✓	✓	✓		20	0.2403	0.9258	**0.0003**
	✓	✓			40	0.1700	0.9100	0.0005
	✓	✓	✓		40	**0.2681**	**0.9312**	**0.0003**

[Base: base model using 2D-LSTM; Grid: efficient triangular grid structure (Fig. 2a); CP+G: conditional prior and generation; Length: sequence length; Entropy: entropy term (Eq.(7))]

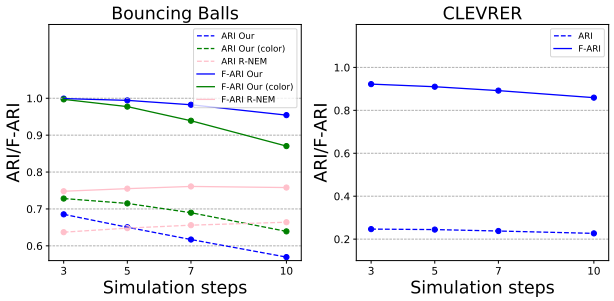

Figure 5: **Prediction.** We show the (F-)ARI for 3, 5, 7, and 10 simulated frames after 20 inference steps.

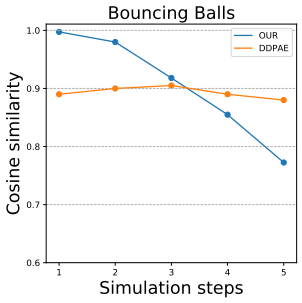

Figure 6: **Velocity Prediction.** Cosine similarity for 5 simulated frames after 10 inference steps.

an explicit model of object-object interactions for the purpose of prediction. PROVIDE, on the other hand, learns disentangled latent representations which also encode dy-

namics. It is thus not primarily designed to be a simulator but allows the use of its latent representations for a variety of downstream tasks, including prediction.

4.6 ABLATION

The quantitative results of an ablation study on the binary Bouncing Balls dataset and CLEVRER are shown in Table 3. We investigate the effects of the efficient grid, conditional prior and generation, length of training sequences and entropy term on the performance of PROVIDE; all contributions are necessary and important. Note that the base models are too large to be trained on 40 frames, which confirms the superiority of our model in terms of both runtime and memory. The CLEVRER dataset is not binary, which is why we do not include the entropy term (see Section 3.2.3). We validate our choice of $\widehat{R}$ and compare it to alternative options in a supplemental study discussed in Appendix F.1.

5 CONCLUSION AND DISCUSSION

We presented a novel unsupervised learning framework capable of precise scene decomposition in multi-object videos with complex appearance and motion. Our temporal component enables modeling of dynamics inside the amortized iterative inference framework but also improves the quality of the results overall. From the quantitative and qualitative comparisons with IODINE and SEQ-IODINE, we see that PROVIDE shows more accurate results on the decomposition task. PROVIDE can also detect new objects faster and is less sensitive to color, because it can leverage the objects' motion cues. For our experiments, we have chosen a setup consistent with other SOTA methods and a focus on the objects' dynamics. PROVIDE is currently not targeting complex textured datasets, as they are not designed for unsupervised learning and impose additional challenges, such as limited coverage of the input space as well as a superposition of the scene's intrinsic components (object location, articulation, motion, albedo, shading, etc.). We refer the reader to Fig. 7 in the Appendix for a decomposition of a real-world video stream and Appendix E for an extended discussion and future work.

References

M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016.

Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting GAN. *arXiv:1905.11369*, 2019.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: unsupervised scene decomposition and representation. *arXiv:1901.11390*, 2019.

Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *NeurIPS*, 2019.

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. In *EMNLP*, 2014.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. In *ICLR*, 2017.

Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. In *AAAI*, 2019a.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *AAAI*, 2019b.

C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2017.

Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why does unsupervised pre-training help deep learning? *JMLR*, 2010.

SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, and Geoffrey E Hinton. Attend, infer, repeat: fast scene understanding with generative models. In *NIPS*, 2016.

Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv:1609.05518*, 2016.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional LSTM networks for improved phoneme classification and recognition. In *ICANN*, 2005.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. In *International Conference on Artificial Neural Networks*, 2007.

Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber. Tagger: deep unsupervised perceptual grouping. In *NIPS*, 2016.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *NIPS*, 2017.

Klaus Greff, Raphaël Lopez Kaufmann, Rishab Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *arXiv:1903.00450*, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. *CoRR*, 2019.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 1985.

A. Jepson, D. Fleet, and M. Black. A layered motion representation with occlusion and compact spatial support. In *ECCV*, 2002.

Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. SCALOR: generative world models with scalable object representations. In *ICLR*, 2020.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: generative modelling of moving objects. In *NeurIPS*, 2018.

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*, 2017.

Joseph Marino, Yisong Yue, and Stephan Mandt. Iterative amortized inference. In *ICML*, 2018.

Isabeau Prémont-Schwarz, Alexander Ilin, Tele Hao, Antti Rasmus, Rinu Boney, and Harri Valpola. Recurrent ladder networks. In *NIPS*, 2017.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: unsupervised discovery of objects and their interactions. In *ICLR*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: a simple architecture for learning disentangled representations in VAEs. *arXiv:1901.07017*, 2019.

Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *CVPR*, 1996.

Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang. Multi-objects generation with amortized structural regularization. *CoRR*, 2019.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: collision events for video representation and reasoning. In *ICLR*, 2020.

B. Zhou, X. Wang, and X. Tang. Understanding collective crowd behaviors: learning a mixture model of dynamic pedestrian-agents. In *CVPR*, 2012.