SCALABLE OBJECT-ORIENTED SEQUENTIAL GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The main limitation of previous approaches to unsupervised sequential object-oriented representation learning is scalability. Most of the previous models have been shown to work only on scenes with a few objects. In this paper, we propose SCALOR, a probabilistic generative model for SCALable sequential Object-oriented Representation. With the proposed spatially-parallel attention and proposal-rejection mechanisms, SCALOR can deal with orders of magnitude more number of objects compared to the previous state-of-the-art model. Besides, we introduce the background model so that SCALOR can model complex background along with many foreground objects. We demonstrate that SCALOR can deal with crowded scenes containing nearly a hundred objects while modeling complex background jointly. Importantly, SCALOR is the first unsupervised object-representation model demonstrating its working in *natural scenes* containing several tens of moving objects.[1]

1 INTRODUCTION

Unsupervised structured representation learning for visual scenes is a key challenge in machine learning. When a scene is properly decomposed into meaningful entities such as foreground objects and background, we can benefit from numerous advantages of symbolic representation. These include interpretability, sample efficiency, the ability of reasoning and causal inference, as well as compositionality and transferability for better generalization. In addition to symbols, another essential dimension is time. Objects, agents, and spaces all operate under the governance of time. Without accounting for temporal developments, it is often much harder if not impossible to discover certain relationships in a scene.

Among a few methods that have been proposed for unsupervised object-oriented representation learning of temporal scenes, SQAIR (Kosiorek et al., 2018) is by far the most complete model. As a probabilistic temporal generative model, it can learn object-wise structured representation while modeling underlying stochastic temporal transitions in the observed data. Introducing the propagation-discovery model, SQAIR can also handle dynamic scenes where objects may disappear or be introduced in the middle of a sequence. Although SQAIR provides promising ideas and shows the potential of this important direction, a few key challenges remain, limiting its applicability merely to synthetic toy tasks that are far simpler than typical natural scenes.

The first and foremost limitation is scalability. Sequentially processing every object in an image, SQAIR has a fundamental limitation in scaling up to scenes with a large number of objects. As such, the state-of-the-art remains at the level of modeling videos containing only a few objects such as MNIST digits per image. Considering the complexity of typical natural scenes as well as the importance of scalable unsupervised object perception for applications such as self-driving systems, it is thus a challenge of the highest priority to scale robustly to scenes with a large number of objects. Scaling up object-attention capacity is an important problem because it allows us to maximize the modern parallel computation that can maximize search capacity, unlike humans who can attend only a few objects at a time, resulting in time-consuming sequential scanning. The AlphaGo system (Silver et al., 2017) is an example demonstrating the power of such parallel search (attention) beyond human capacity.

[1] https://sites.google.com/view/scalor/home

Second, SQAIR conspicuously lacks any form of background modeling and thus only copes with scenes without background, whereas natural scenes usually have a complex background. Thus, a temporal generative model that can deal with complex backgrounds along with many foreground objects is an important step toward natural video scene understanding.

In this paper, we propose a model called **SCAL**able Sequential **O**bject-Oriented **R**epresentation (SCALOR), which resolves the above key limitations and hence can model complex videos with several tens of moving objects along with complex backgrounds, *eventually making the model applicable to natural videos*. In SCALOR, we achieve scalability with respect to the object density by parallelizing both the propagation and discovery processes, reducing the parallel time complexity per scene image to $\mathcal{O}(1)$ from $\mathcal{O}(N)$ with N the number of objects in an image. We also observe that the serial object processing in SQAIR based on an RNN not only increases the computation time but also deteriorates discovery performance. To this end, we propose a parallel discovery model with much better discovery capacity and performance. Temporally predicting and detecting trajectories of objects, SCALOR can also be regarded as a generative tracking model. In our experiments, we show that SCALOR can model videos with nearly one hundred moving objects along with a complex background on synthetic datasets. Furthermore, we evaluate and demonstrate SCALOR on natural videos as well with tens of objects with a complex background.

The contribution of this work are: (i) We propose the SCALOR model that significantly improves (two orders of magnitude) the scalability with regard to the object density. It is applicable to nearly a hundred objects with comparable computation time to SQAIR, which scales only to a few objects. (ii) We propose parallelizing the propagation–discovery process by introducing the propose–reject model, reducing the time complexity to $\mathcal{O}(1)$. (iii) SCALOR can model scenes with complex background. (iv) SCALOR is the first probabilistic model demonstrating its working not only on natural images but also at a significant complexity level of tens of objects and background.

2 PRELIMINARIES: SEQUENTIAL ATTEND INFER REPEAT (SQAIR)

Before introducing our model, we first review the SQAIR model. SQAIR models an observed sequence of images $\mathbf{x} = \mathbf{x}_{1:T}$ by assuming that the observation $\mathbf{x}_t$ at time t is generated from a set of object latent variables $\mathbf{z}_t^{\mathcal{O}} = \{\mathbf{z}_{t,n}\}_{n\in\mathcal{O}_t}$ with $\mathcal{O}_t$ a set of objects present at time t. Each latent variable $\mathbf{z}_n$ for an object n consists of the factors $(z_n^{\text{pres}}, \mathbf{z}_n^{\text{where}}, \mathbf{z}_n^{\text{what}})$, which represent the existence, pose, and appearance of the object, respectively. SQAIR also assumes that an object can disappear or be introduced in the middle of a sequence. To model this, it introduces the propagation–discovery model. In propagation, a subset of currently existing object is propagated to the next time step and those not propagated (e.g., because an object disappears) are deleted. In discovery, after deciding how many objects D_t will be discovered, new D_t objects are introduced into the scene. Combining the propagated $\mathcal{P}_t$ and discovered $\mathcal{D}_t$, we obtain the set of currently existing objects $\mathcal{O}_t$. The complete process can be formalized as:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}^{\text{fg}}, D_{1:T}) = p(D_1, \mathbf{z}_1^{\mathcal{D}}) \prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{z}_t^{\text{fg}}) p(D_t, \mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) p(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{t-1}^{\text{fg}}). \tag{1}$$

Here, we use $\mathbf{z}_t^{\text{fg}}$ to denote $\mathbf{z}_t^{\mathcal{O}}$. Due to the intractable posterior, SQAIR is trained through variational inference with the following posterior approximation:

$$q(D_{1:T}, \mathbf{z}_{1:T}^{\text{fg}}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(D_t, \mathbf{z}_t^{\mathcal{D}}|\mathbf{x}_t, \mathbf{z}_t^{\mathcal{P}}) \prod_{n\in\mathcal{O}_{t-1}} q(\mathbf{z}_{t,n}|\mathbf{z}_{t-1,n}, \mathbf{x}_{\leq t}) \tag{2}$$

SQAIR is trained using an importance-weighted autoencoder (IWAE) objective (Burda et al., 2015). The VIMCO estimator (Mnih & Rezende, 2016) is used to backpropagate through the discrete random variables while using the reparameterization trick (Kingma & Welling, 2013; Williams, 1992) for continuous variables. SQAIR has two main limitations in scalability. First, for propagation, SQAIR relies on an RNN. Thus, the propagation is performed sequentially by conditioning on previously processed objects. Second, the discovery is also sequential because it uses RNN-based discovery based on AIR (Eslami et al., 2016). Consequently, SQAIR has $O(|\mathcal{O}_t|)$ time complexity per step t. In previous work (Crawford & Pineau, 2019), it is demonstrated that this sequential discovery can easily fail beyond the scale of a few objects. Moreover, SQAIR lacks any model for the background and its temporal transitions which is important in modeling natural scenes.

3 THE PROPOSED MODEL: SCALOR

3.1 GENERATIVE PROCESS

SCALOR assumes that an image $\mathbf{x}_t$ is generated by background latent $\mathbf{z}_t^{\text{bg}}$ and foreground latent $\mathbf{z}_t^{\text{fg}}$. The foreground is further factorized into a set of object representations $\mathbf{z}_t^{\text{fg}} = \{\mathbf{z}_{t,n}\}_{n\in\mathcal{O}_t}$. In SCALOR, we represent an object by $\mathbf{z}_{t,n} = (z_{t,n}^{\text{pres}}, \mathbf{z}_{t,n}^{\text{where}}, \mathbf{z}_{t,n}^{\text{what}}, z_{t,n}^{\text{depth}})$. The depth representation, which is missing in SQAIR, helps model object occlusion. The foreground mask $\mathbf{m}_{t,n}$ obtained from $\mathbf{z}_{t,n}^{\text{what}}$ is used to distinguish background and foreground. The appearance representation $\mathbf{z}_{t,n}^{\text{what}}$ is a continuous vector representation (e.g., as in VAE), and $\mathbf{z}_{t,n}^{\text{where}}$ is further decomposed into position $\mathbf{z}_{t,n}^{\text{pos}}$ and scale $\mathbf{z}_{t,n}^{\text{scale}}$. We adopt the propagation–discovery model from SQAIR, but improve it in such a way to resolve the scalability problem. The generative process of SCALOR can be written:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p(\mathbf{z}_1^{\mathcal{D}})(\mathbf{z}_1^{\text{bg}}) \prod_{t=2}^{T} \underbrace{p(\mathbf{x}_t|\mathbf{z}_t)}_{\text{rendering}} \underbrace{p(\mathbf{z}_t^{\text{bg}}|\mathbf{z}_{<t}^{\text{bg}}, \mathbf{z}_t^{\text{fg}})}_{\text{background transition}} \underbrace{p(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}})}_{\text{discovery}} \underbrace{p(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{<t})}_{\text{propagation}}. \tag{3}$$

where $\mathbf{z}_t = (\mathbf{z}_t^{\text{bg}}, \mathbf{z}_t^{\text{fg}})$. As shown, the generation process is decomposed into four modules: (i) propagation, (ii) discover, (iii) background transition, and (iv) rendering.

The **propagation** in SCALOR is modeled as follows:

$$p(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{<t}) = \prod_{n\in\mathcal{O}_t} p(z_{t,n}^{\text{pres}}|\mathbf{z}_{<t,n}) \left\{p(\mathbf{z}_{t,n}^{\text{where}}|\mathbf{z}_{<t,n})p(\mathbf{z}_{t,n}^{\text{what}}|\mathbf{z}_{<t,n})\right\}^{z_{t,n}^{\text{pres}}}, \tag{4}$$

where $p(z_{t,n}^{\text{pres}}|\mathbf{z}_{<t,n})$ is a Bernoulli distribution with parameter $\beta_{t,n}$. The distributions of what and where are defined only when the object is propagated. To implement this, for each object n we assign a *tracker* RNN denoted by its hidden state $h_{t,n}$. The RNN is updated by input $\mathbf{z}_{t,n}$ for all t where the object n is present in the scene. The parameter $\beta_{t,n}$ is obtained as $\beta_{t,n} = f_{\text{nn}}(h_{t,n})$. If $z_{t,n}^{\text{pres}} = 0$, the object n is not propagated and the tracker RNN is deleted. Importantly, unlike the RNN-based sequential propagation in SQAIR, the propagation in SCALOR is fully parallel.

Discovery by Proposal-Rejection. The main contribution in making our model scalable with respect to the number of objects is our new discovery model that consists of two phases: *proposal* and *rejection*. In the proposal phase, we assume that the target image can be divided into $H \times W$ grid cells, and we propose an object latent variable $\tilde{\mathbf{z}}_{t,h,w}$ per grid cell. This proposal phase can be written as:

$$p(\tilde{\mathbf{z}}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) = \prod_{h,w=1}^{HW} p(\tilde{\mathbf{z}}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) = \prod_{h,w=1}^{HW} p(\tilde{z}_{t,h,w}^{\text{pres}}|\mathbf{z}_t^{\mathcal{P}}) \left\{p(\tilde{\mathbf{z}}_{t,h,w}^{\text{where}}|\mathbf{z}_t^{\mathcal{P}})p(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}}|\mathbf{z}_t^{\mathcal{P}})\right\}^{\tilde{z}_{t,h,w}^{\text{pres}}}. \tag{5}$$

In the rejection phase, our goal is to reject some of the proposed objects if a proposed object largely overlaps with a propagated object. In our model, each object representation contains a mask variable $\mathbf{m}_{t,n}$, which is used to make the rejection decision. Specifically, if the overlap between the mask of a proposed object and that of a propagated one is over a threshold τ, we reject the proposed object. This procedure can be described as (i) $\tilde{\mathbf{z}}_t^{\mathcal{D}} \sim p(\tilde{\mathbf{z}}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}})$ and (ii) $\mathbf{z}_t^{\mathcal{D}} = f_{\text{accept}}(\tilde{\mathbf{z}}_t^{\mathcal{D}}, \mathbf{z}_t^{\mathcal{P}}, \tau)$ and thus $\mathbf{z}_t^{\mathcal{D}} \subseteq \tilde{\mathbf{z}}_t^{\mathcal{D}}$. Although we use a deterministic function for the rejection, it can be a design choice to implement this as a stochastic decision. While one rationale behind this design is to reflect an inductive bias from physics that two objects cannot coexist in the same position, we shall also see later while discussing the inference procedure further reasons as to why this design is effective. The final discovery model can be written as $p(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) = p(\tilde{\mathbf{z}}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) \prod_{h,w=1}^{HW} p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}, \tau)$, where the acceptance model is $p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}, \tau) = f(z_{t,h,w}^{\text{accept}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}, \tau)p(\tilde{\mathbf{z}}_{t,h,w}^{\text{where}})^{z_{t,h,w}^{\text{accept}}} p(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}})^{z_{t,h,w}^{\text{accept}}}$. Proposal independent to the propagation also worked well in our experiments.

Background Transition. Unlike SQAIR, SCALOR is endowed with a background model. The background image is encoded into a D-dimension continuous vector $\mathbf{z}_t^{\text{bg}}$ from the background transition $p(\mathbf{z}_t^{\text{bg}}|\mathbf{z}_{<t}^{\text{bg}}, \mathbf{z}_t^{\text{fg}})$. The background RNN encodes the temporal transition of background images.

Rendering. The implementation of the rendering process is the same as in SPAIR (Crawford & Pineau, 2019), except that we process the objects in parallel.

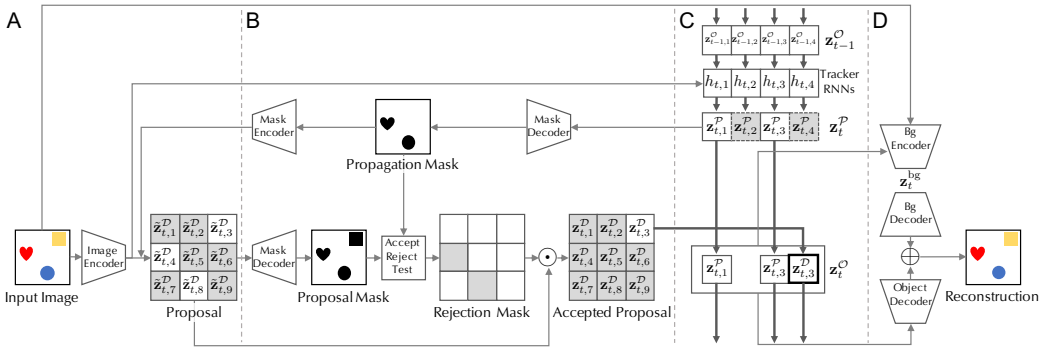

Figure 1: SCALOR inference procedure: (A) Proposal, (B) Accept-Reject, (C) Propagation, (D) Background Module and Rendering Process. (A) The proposal module takes input image and the propagation mask and combines them to make the proposal representation. (B) From the proposal representation, proposal mask is generated which is then compared to the propagation mask to make accept-reject decision. Only the accepted proposals are considered as discovered objects. (C) The tracker RNNs decide what and where to propagate after looking at the input image. The grey boxes are what is not propagated. (D) Given inferred foreground objects and the input image, the background module infers the background representation. The rendering process combines the foreground and background representations according to the foreground mask assignment.

3.2 Learning and Inference

Due to the intractability of the true posterior distribution $p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$, we train our model using variational inference with the following posterior approximation:

$$q(\mathbf{z}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t}) = \prod_{t=1}^{T} q(\mathbf{z}_t^{\text{bg}}|\mathbf{z}_t^{\text{fg}}, \mathbf{x}_t) q(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t}) q(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{<t}, \mathbf{x}_{\leq t}). \tag{6}$$

Posterior Propagation $q(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{<t}, \mathbf{x}_{\leq t})$ is similar to the propagation in generation, except that we now provide observation $\mathbf{x}_{\leq t}$ through an RNN encoding. Here, the propagation for each object n is done by $q(\mathbf{z}_{t,n}|\mathbf{z}_{<t,n}, \mathbf{x}_{\leq t})$ using attention $\mathbf{a}_{t,n} = f_{\text{att}}(\mathbf{x}_{\leq t})$ on the feature map for object n. To compute the attention, we use the previous position $\mathbf{z}_{t-1,n}^{\text{pos}}$ as the center position and extract half the width and height of the convolutional feature map using bilinear interpolation. This attention mechanism is motivated by the observation that only part of the image contains information for tracking an object, and an inductive bias that objects cannot move a large distance within a short time span (i.e., objects do not teleport).

Posterior Discovery. The posterior discovery also consists of proposal and rejection phases. The main difference is that we now compute the proposal in *spatially-parallel* manner by conditioning on the observations $\mathbf{x}_{\leq t}$, i.e., $q(\tilde{\mathbf{z}}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t}) = \prod_{h,w=1}^{HW} Q(\tilde{\mathbf{z}}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t})$. Here, the observation $\mathbf{x}_{\leq t}$ is encoded into the feature map of dimensionality $H \times W \times D$ using a Convolutional LSTM (Xingjian et al., 2015). Then, from each feature, we obtain $\tilde{\mathbf{z}}_{t,h,w}^{\mathcal{D}}$. Importantly, this is done in parallel over all the feature cells h, w. A similar approach is used in SPAIR (Crawford & Pineau, 2019), but it infers the object latent representations sequentially and thus is difficult to scale to a large number of objects (Anonymous, 2019). Even if this spatially-parallel proposal plays a key role in making our model scalable, we also observe a challenge due to this high-capacity of the discovery module. The problem is that the discovery module tends to dominate the propagation module and thus most of the objects in an image are explained by the discovery module, i.e., objects can be *re*discovered at every time step while nothing is propagated.

Why the model should tend to explain an image through discovery while suppressing propagation? First, the model does not care where—either from discovery or propagation—an object is sourced from as long as it can make accurate reconstruction. Second, the propagation performs a much harder task than the discovery. For the propagation to properly predict, the tracker n needs to learn to *find the matching object* from the image containing many objects. Even if we use attention for this, we observe that this is particularly difficult during the early training phase. On the contrary, the discovery module does not need to solve such difficult association problem because it only performs *local* image-to-latents encoding without associating latents of previous time step. Therefore, it is much easier for the discovery encoder to produce latents that are more accurate than those from

propagation. If we limit the capacity of the discovery module and sequentially process objects like in SQAIR, we may mitigate this problem because the propagation module is naturally enforced to explain what the discovery module cannot, but this approach is difficult to scale.

We employ two methods to resolve this problem. First, we simply bias the initial network parameter so that it has a high propagation probability at the beginning of the training. This helps the model try to explain the observation first through propagation and then the discovery module takes the remaining as the training proceeds. The second method is our proposal-rejection mechanism, which is implemented the same way as in the generation process. This prevents the discovery model from redundantly explaining what is already explained by the propagation module. The final posterior discovery model can be written as:

$$q(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t}) = q(\tilde{\mathbf{z}}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t}) \prod_{h,w=1}^{HW} p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}), \tag{7}$$

where the acceptance model is $p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}) = p(z_{t,h,w}^{\text{pres}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}})(p(\tilde{\mathbf{z}}_{t,h,w}^{\text{where}})p(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}}))^{z_{t,h,w}^{\text{pres}}}$.

Posterior Background. The posterior of the background $q(\mathbf{z}_t^{\text{bg}}|\mathbf{z}_t^{\text{fg}}, \mathbf{x}_t)$ is conditioned on the input image and currently existing objects. The existing objects describe what part of the image should already have been dealt with by the foreground object models, so that the remaining should be explained by the background module.

Training. We train our model by maximizing the following evidence lower bound $\mathcal{L}(\theta, \phi) =$

$$\sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{z}_{<t}|\mathbf{x}_{<t})} \Big[\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t})} \left[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t)\right] - \mathbb{KL}\left[q_\phi(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t}) \parallel p_\theta(\mathbf{z}_t|\mathbf{z}_{<t})\right] \Big]. \tag{8}$$

We use the reparameterization trick (Williams, 1992; Kingma & Welling, 2013) for continuous random variables such as $\mathbf{z}^{\text{what}}$, and the Gumbel-Softmax trick (Jang et al., 2016) for discrete variables such as z^{pres}. We found that our proposed model works well and stably with these simpler training methods than what is used in SQAIR, i.e., VIMCO and IWAE.

4 Related Work

AIR (Eslami et al., 2016) decomposes scenes into latent variables representing the appearance, position and size of the underlying object by making use of a recurrent neural network. SPAIR (Crawford & Pineau, 2019) takes a similar approach but applies spatially invariant attention to extract local feature maps. Although SPAIR provides better scalability than AIR, it is still limited as it performs sequential inference on objects. Scene-mixture models (Greff et al., 2017; Van Steenkiste et al., 2018; Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2019) decompose scenes into a collection of components, each being a full-image level representation. Although these models allow decomposition of the input image into components, multiple objects can be in a component and the obtained representation does not contain explicit features like position and scale, etc. SQAIR (Kosiorek et al., 2018), discussed in Section 2, extends the ideas proposed in AIR to temporal sequences.

DDPAE (Hsieh et al., 2018) is another sequential generative model for object-oriented learning, which models each object in the scene with an appearance and a position vector. The model assumes the appearance of an object to be fixed and thus shares the content vector across different time steps. NEM (Greff et al., 2017) and RNEM (Van Steenkiste et al., 2018) introduce a spatial mixture model to disentangle the scene into multiple components representing each entity. Since each component generates a full scene image, the latent representations are not interpretable. On another front, Tracking-By-Animation (He et al., 2019) introduces a deterministic model to tackle the task of object tracking in an unsupervised fashion. Overall, there has numerous amount of works on object tracking from the computer vision community using the same "bounding box"-representation approach proposed in SCALOR (Kosiorek et al., 2017; Ning et al., 2017; Nam & Han, 2016; Tao et al., 2016), but such methods use provided labels to tackles the problem of object tracking and thus are usually fully or semi-supervised and not probabilistic models. Therefore they do not model object-wise representation.

	Experimental Setting	MOTA ↑	MOTP ↓	Precision ↑	Recall ↑	MAE ↓
dSprites	SCALOR – VHD	87.9%	1.49	94.6%	94.5%	0.10
	SCALOR – HD	89.6%	1.65	99.4%	91.9%	0.08
	SCALOR – MD	91.5%	1.76	98.2%	96.1%	0.05
	SCALOR – LD	86.6%	1.78	96.4%	93.8%	0.10
	SCALOR – VLD	87.8%	1.59	97.2%	96.6%	0.07
	SQAIR – LD	21.8%	0.78	99.9%	54.2%	0.45
	SQAIR – VLD	65.9%	1.08	99.6%	68.8%	0.30
MNIST	SCALOR – MD	91.2%	1.85	95.4%	96.3%	0.09
	SCALOR – LD	84.2%	1.90	94.3%	90.8%	0.15
	SCALOR – VLD	95.8%	0.90	98.6%	99.2%	0.03
	SQAIR – VLD	78.5%	0.96	99.9%	78.8%	0.22
Extra	SCALOR – BG	90.1%	1.71	97.9%	95.2%	0.07
	SCALOR – LG	90.5%	1.79	97.2%	95.4%	0.07
	SCALOR – CG	90.0%	1.67	-	92.0%	0.08

Table 1: Quantitative results of SCALOR for different experimental settings. "SCALOR - BG" refers to the computed metrics for Experiment 3, which contains frames with complex background. "SCALOR – LG" and "SCALOR – CG" refer to length generalization and count generalization experiments in Experiment Set 4.

5 EXPERIMENTS

Since we cannot provide all the figures in the following section, some figures, denoted by ".A", are moved to the Appendix. Furthermore, We have also created a project webpage with additional qualitative examples and videos of SCALOR.

5.1 TASK 1: LARGE-SCALE MNIST AND DSPRITES SHAPES

We first evaluate our model on a dataset of moving dSprites shapes and a dataset consisting of moving MNIST digits. In all experiments, objects move in a constrained environment but the processed trajectory images cover only a 64×64 partial view of the center of the whole scene. Therefore, while there is a fixed number of objects in the environment at each time step, only a subset of them are visible in the observed image and others are outside the observed view. We experiment on five different scale settings. In each setting, the number of objects in each trajectory is sampled uniformly from an interval [*min*, *max*]. Each *scale-setting* is specified with a triplet (min, avg, max), where *min* and *max* are as we mentioned and *avg* represents the average number of visible objects in all the trajectories in that setting. The five settings investigated in this paper are referred to as Very Low Density (VLD) [(2, 2.9, 4)], Low Density (LD) [(10, 9.2, 12)], Medium Density (MD) [(24, 21.9, 27)], High Density (HD) [(50, 54.5, 64)] and Very High Density (VHD) [(100, 99, 120)]. For example, in the Medium Density setting, there are always between 24 to 27 objects in the overall environment, while only ~ 22 of them are visible on average in each viewpoint frame.

Experiment 1: Measuring Tracking Performance. The aim of this experiment is to evaluate the ability to track moving objects in a simple environment without background. Table 1 includes various performance measures, encompassing standard object tracking metrics such as Multi Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) as well as Precision and Recall of the inferred bounding boxes (Bernardin & Stiefelhagen, 2008). While MOTA and MOTP measure the accuracy and consistency of the inferred bounding boxes over time, precision and recall measure the number of false positive or negatives, regardless of the associated ids. Note that MOTP only considers the dissimilarity between the *true positives* and their ground truth targets, i.e. only the correctly matched trajectories are used for calculating the error. And MOTA considers all error terms including false negative, false positive and id swaps. For computing the MOT metrics, we chose the distance threshold to be the maximum possible Euclidean distance so that all objects are considered at the association phase. MAE is mean average error which measures the absolute difference between the number of predicted objects and the actual number of objects, normalized by the latter.

We compare the performance of the proposed model to SQAIR in the VLD setting for both the MNIST and dSprites datasets, and LD setting for dSprites. Note that we are not able to make SQAIR work on other settings. As shown, SCALOR can outperform SQAIR in all these settings, obtaining significantly higher accuracy and recall, as SQAIR either misses some objects or misidentifies distinct objects as one. SCALOR also has lower values of counting MAE in comparison to

		dSprites				MNIST		
Method		VLD	LD	MD	HD	VLD	LD	MD
NLL	SCALOR	22.4838	23.8230	23.8753	25.4659	8.0017	8.3304	8.0455
	VRNN	22.3320	22.8000	23.2784	23.5707	7.7527	7.7012	7.7101
	SQAIR	22.7332	36.7311	-	-	7.7778	-	-
MSE	SCALOR	0.0015	0.0032	0.0023	0.0090	0.0010	0.0026	0.0010
	VRNN	0.0018	0.0071	0.0105	0.0251	0.0027	0.0054	0.0035
	SQAIR	0.0125	0.0386	-	-	0.0084	-	-

Table 2: Negative Log Likelihood and Mean Squared Error measured for SCALOR vs. baselines.

SQAIR, showing that SCALOR can infer the correct number of objects in the scene more accurately. Furthermore, We observe that increasing the number of objects in the scene does not significantly impede SCALOR's tracking ability, which demonstrates the strength of SCALOR when applied to images with a high number of objects. SCALOR is able to achieve relatively high precision and recall even for scenes containing $\sim$100 objects. Table 2 provides a comparison of SCALOR to SQAIR and VRNN in terms of the reconstruction error (MSE) and negative log-likelihood (NLL). NLL is computed per pixel for the thole sequence.

Figures 3.A and 4.A demonstrate the qualitative performance of SCALOR on dSprites (HD) and MNIST (MD), respectively. As for the moving MNIST dataset, we did not explore the High Density and Very High Density scenarios, as the resolution of MNIST digits became too small, making digits unidentifiable for that setting. To better show the consistency of the inferred bounding boxes over different time-steps, bounding boxes are represented by distinct colors. For clearer visualization of the trackers in the HD setting, bounding boxes are superimposed on the object centroids represented by colored dots. White circles show the discovered objects at each time-step. SCALOR tracks large numbers of objects accurately and consistently even after severe occlusion due to overlapping objects.

Experiments on Frequent Dense Discovery, Generalization to Unseen Data, Very High Density, and Handling Occlusion: We also perform experiments to demonstrate and evaluate various other abilities of SCALOR. Specifically, we show that SCALOR can deal with (i) frequent discovery of many objects happening in the middle of a sequence, (ii) generalization to unseen data, (iii) very high density of about 100 objects, and (iv) occlusions. For the experiment settings and results on each of these experiments, refer to sections E.1 to E.4 in the Appendix respectively. Although we describe these results in the Appendix due to the space constraint, we note that these results are also our major contributions.

5.2 TASK 2: REAL-WORLD DATASET TRACKING

This section considers the effectiveness of SCALOR when provided with noisy real-world video-frames, which previous work has not been able to handle. The challenges faced in this setting can be significantly more difficult to overcome, due to the inherent difficulty compared to synthetic data. Specifically, we consider the Crowded Grand Central Station dataset (Zhou et al., 2012), which was collected from CCTV cameras at New York City's Grand Central Station. Due to the complexity of pedestrian behavior, the dataset can be considered a mix of both low density and high density objects settings.

Figure 6.A shows the qualitative results of the model at inference time. SCALOR performs reasonably well on this pedestrian tracking dataset, maintaining consistent temporal trajectories. As shown in Figure 6.A(c), the background module infers the background composition and reconstructs the extracted background correctly. As for object detection, SCALOR succeeds in accurate pedestrian detection and tracking. It could also output the foreground mask produced by $\mathbf{z}^{\text{what}}$, which can represent the segmentation of the shape of individual pedestrians, as shown in Figure 6.A(d). We draw extracted trajectories in Figure 6.A(e) for each pedestrian in the natural scene. Trajectories in different colors correspond to different pedestrian ids, which is given by the identity of the inferred latent variable of each object.

Figure 7.A shows qualitative results of conditional generation. The last 5 frames are generated images conditioned on the first 5 observations. Starting from the 6th frame, the latent transition of the propagation trackers is modeled by the sequential prior network. During generation, a different prior of z^{pres} is introduced in the discovery module for capturing new objects emerging in the scene (see

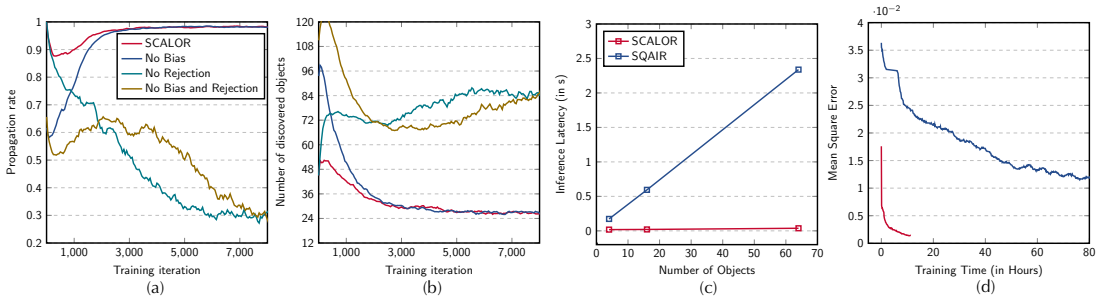

Figure 2: Quantitative result of ablation study and computational efficiency comparison. (a) Propagation rate, (b) Number of discovered objects, (c) Inference latency, (d) Convergence time.

Appendix G for details). As shown in the figure, for the trajectories, the model tends to generate the same directions as aligned in the previous frames. This also applies to newly generated objects from the discovery phase, the movement of which appears highly consistent as shown in Figure 7.A(f). Although the trajectory movements are generated consistently, the appearance of some objects in the generation phase sometimes varies across different time frames. Noticeably, in the generated sequence the segmentation mask for each object tends to deform into a different shape. This may stem from the imperfection during the inference that makes the learning of appearance transition to be difficult. Detail of the dataset and more qualitative results are given in Appendix E.5.

Since the ground truth trajectories of the Grand Central Station Dataset were no longer available at the time the experiment was conducted, we instead used negative log-likelihood (NLL) as the evaluation metric, and compared our model with two baselines to show the data modeling capacity, a sequential VAE and a Recurrent Latent Variable Model (VRNN). The sequential VAE has one latent variable $\mathbf{z}$ of dimensionality 64, and a sequential prior $p(\mathbf{z}_t|\mathbf{z}_{<t})$. It is similar to our background module with the same number of latent variables for encoder and decoder. As for the VRNN baseline, we implement a VRNN with 128 as the dimensionality of the LSTM hidden state and latent variable $\mathbf{z}$ and chose a convolutional network as the image encoder and decoder. The NLL value for our model is 28.30, and for the VAE and VRNN baseline, it is 27.59 and 27.79, respectively. As we can see, as a discrete representation learning model that learns a semantic structured representation, SCALOR can still attain a comparable level of generation quality to methods with continuous latent representation.

5.3 Ablation Study and Computational Efficiency Comparison

In this section, we present the quantitative result of the ablation study and computational efficiency comparison. The ablation study experiment is upon the rejection mechanism and initial bias of propagation. We measure the rate of propagation for objects discovered at the previous timestep, as well as the total number of discovery happens in the sequence. The result is shown in Fig. 2 (a) and (b). As we can see in the figure, the introduction of rejection mechanism significantly increases the propagation rate and reduce the number of discovery. The computational efficiency comparison use inference latency and time of convergence as metrics and compare with generative model SQAIR. Fig. 2 (c) and (d) shows the comparison result. As is shown, SCALOR outperforms SQAIR significantly on inference latency and convergence time. For detailed analysis for both experiments refer to section F in Appendix.

6 Conclusion

We introduce SCALOR (SCALable Sequential Object-Oriented Representation), a scalable sequential object-oriented representation learning model via generation. We make the model deal with two orders of magnitude more objects than previous models. To do this, we combine the spatially-parallel discovery with the proposal-rejection mechanism. The proposed model can also deal with complex background. Consequently, we significantly improve the state-of-the-art, which was at the level of dealing with a few simple objects without background, and demonstrate that SCALOR is the first probabilistic model that can model natural scenes with several tens of objects with complex background. An interesting future direction is to introduce more structure to the background and modeling relationships among objects and the background.

REFERENCES

Anonymous. A paper under review. 2019.

Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

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

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

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations, 2019.

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 *Advances in Neural Information Processing Systems*, pp. 3225–3233, 2016.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pp. 6691–6701, 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 preprint arXiv:1903.00450*, 2019.

Zhen He, Jian Li, Daxue Liu, Hangen He, and David Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1318–1327, 2019.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pp. 517–526, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Adam Kosiorek, Alex Bewley, and Ingmar Posner. Hierarchical attentive recurrent tracking. In *Advances in Neural Information Processing Systems*, pp. 3053–3061, 2017.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pp. 8606–8616, 2018.

Andriy Mnih and Danilo J Rezende. Variational inference for monte carlo objectives. *arXiv preprint arXiv:1602.06725*, 2016.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4293–4302, 2016.

Guanghan Ning, Zhi Zhang, Chen Huang, Xiaobo Ren, Haohong Wang, Canhui Cai, and Zhihai He. Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4. IEEE, 2017.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1420–1429, 2016.

Sjoerd Van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2871–2878. IEEE, 2012.

A ALGORITHMS

Algorithm 1: Proposal-Rejection Inference

Input: $\mathbf{x}_{1:t}$ - image sequence up to current time-step
$\mathbf{M}_t^{\mathcal{P}}$ - propagated mask from current time-step
τ - rejection hyper-parameter

```
/* Feature encoding                                                    */
```

$\mathbf{e}_t^{\text{img}} = \text{ConvLSTM}(\mathbf{x}_{1:t})$
$\mathbf{e}_t^{\text{mask}} = \text{MaskEncoder}(\mathbf{M}_t^{\mathcal{P}})$
$\mathbf{e}_t^{\text{agg}} = \text{Concat}[\mathbf{e}_t^{\text{img}}, \mathbf{e}_t^{\text{mask}}]$

```
/* Proposal-Rejection (done in parallel)                               */
```

$\text{AcceptedList}_t = []$
for $h \leftarrow 1$ **to** H **do**
  for $w \leftarrow 1$ **to** W **do**
    `/* Objects proposal */`
    $\tilde{z}_{t,h,w}^{\text{pres}} \sim \text{Bern}(\cdot | f_{nn}^{\text{pres}}(\mathbf{e}_{t,h,w}^{\text{agg}}))$
    if $\tilde{z}_{t,h,w}^{pres} == 0$ **then**
      continue
    end
    $\tilde{\mathbf{z}}_{t,h,w}^{\text{where}} \sim \mathcal{N}(\cdot | f_{nn}^{\text{where}}(\mathbf{e}_{t,h,w}^{\text{agg}}))$
    $\tilde{z}_{t,h,w}^{\text{depth}} \sim \mathcal{N}(\cdot | f_{nn}^{\text{depth}}(\mathbf{e}_{t,h,w}^{\text{agg}}))$
    $\mathbf{g}_{\mathbf{t,h,w}}^{\text{att}} = \text{STN}(\mathbf{x_t}, \tilde{\mathbf{z}}_{\mathbf{t,h,w}}^{\text{where}})$ `// Attented Glimpse`
    $\tilde{\mathbf{z}}_{t,h,w}^{\text{what}} \sim \mathcal{N}(\cdot | \text{GlimpseEnc}(\mathbf{g}_{\mathbf{t,h,w}}^{\text{att}}))$
    $\mathbf{o}_{t,h,w}, \mathbf{m}_{t,h,w} = \text{STN}^{-1}(\text{GlimpseDec}(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}})), \tilde{\mathbf{z}}_{t,h,w}^{\text{where}})$
    `/* Accept-Reject Test */`
    $\delta = \frac{\mathcal{A}(\mathbf{m}_{t,h,w}^{\mathcal{D}} \cap \mathbf{M}_t^{\mathcal{P}})}{\mathcal{A}(\mathbf{m}_{t,h,w}^{\mathcal{D}})}$ `// 𝒜 - pixel area`
    if $\delta < \tau$ **then**
      $\text{AcceptedList}_t.\text{Add}(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}}, \tilde{\mathbf{z}}_{t,h,w}^{\text{where}}, \tilde{z}_{t,h,w}^{\text{pres}}, \tilde{z}_{t,h,w}^{\text{depth}})$
    end
  end
end

Output: AcceptedList_t

Algorithm 2: Propagation Inference

Input: $\mathbf{x}_{1:t}$ - image sequence up to current time-step
PropList$_{t-1} = \{\mathbf{z}_{t-1,n}, h_{t-1,n}\}_{n \in \mathcal{O}_{t-1}}$ - latent variable from previous time-step

```
/* Feature encoding */
```

$\mathbf{e}_t^{\text{img}} = \text{ConvLSTM}(\mathbf{x}_{1:t})$

```
/* Object tracking (done in parallel) */
```

$\text{PropList}_t = \text{PropList}_{t-1}$

for $n \leftarrow 1$ **to** $\mathcal{O}_{t-1}$ **do**

- $\mathbf{e}_{t,n}^{\text{att}} = f_{nn}^{\text{att}}(\text{STN}(\mathbf{e}_{t,n}^{\text{img}}, \mathbf{z}_{t-1,n}^{\text{where}}))$ `// Feature map attention`
- $h_{t,n} = \text{GRU}(f_{nn}([\mathbf{e}_{t,n}^{\text{att}}, \mathbf{z}_{t-1,n}^{\text{what}}, \mathbf{z}_{t-1,n}^{\text{where}}, z_{t-1,n}^{\text{pres}}]), h_{t-1,n})$
- $\mathbf{e}_{t,n}^{\text{agg}} = f_{nn}^{\text{agg}}([\mathbf{e}_{t,n}^{\text{att}}, \mathbf{z}_{t-1,n}^{\text{what}}, \mathbf{z}_{t-1,n}^{\text{where}}, z_{t-1,n}^{\text{pres}}, h_{t,n}])$
- $\mathbf{z}_{t,n}^{\text{where}} \sim \mathcal{N}(.|f_{nn}^{\text{where}}(\mathbf{e}_{t,n}^{\text{agg}}))$
- $\mathbf{g}_{\mathbf{t,n}}^{\text{att}} = \text{STN}(\mathbf{x_t}, \mathbf{z}_{\mathbf{t,n}}^{\text{where}})$ `// Attented Glimpse`
- $\mathbf{z}_{t,n}^{\text{what}} \sim \mathcal{N}(\cdot|\text{GlimpseEnc}(\mathbf{g}_{\mathbf{t,n}}^{\text{att}}))$
- $z_{t,n}^{\text{depth}} \sim \mathcal{N}(\cdot|f_{nn}(\mathbf{e}_{t,n}^{\text{att}}, \mathbf{z}_{t,n}^{\text{what}}, h_{t,n}))$
- $z_{t,n}^{\text{pres}} \sim \text{Bern}(\cdot|f_{nn}^{\text{pres}}(\mathbf{z}_{t,n}^{\text{where}}, \mathbf{z}_{t,n}^{\text{what}}, h_{t,n}))$
- **if** $z_{t,n}^{pres} == 1$ **then**
  - $\text{PropList}_{t,n}.\text{Update}(\mathbf{z}_{t,n}^{\text{what}}, \mathbf{z}_{t,n}^{\text{where}}, \mathbf{z}_{t,n}^{\text{pres}}, \mathbf{z}_{t,n}^{\text{depth}})$
- **else**
  - $\text{PropList}_{t,n}.\text{Delete}()$
- **end**

end

Output: PropList_t

Algorithm 3: Background Module and Rendering

Input: $\mathbf{x}_t$ - image at current time-step,
$\mathbf{o}_t = \{\mathbf{o}_{t,n}\}_{n \in \mathcal{O}_t}$ - object RGB glimpses
$\mathbf{m}_t = \{\mathbf{m}_{t,n}\}_{n \in \mathcal{O}_t}$ - object masks glimpses,
$\{\mathbf{z}_{t,n}^{\text{where}}, \mathbf{z}_{t,n}^{\text{depth}}, \mathbf{z}_{t,n}^{\text{pres}}\}_{n \in \mathcal{O}_t}$ - object latents for position and existence

```
/* Foreground object rendering (done in parallel) */
```

for $n \leftarrow 1$ **to** $\mathcal{O}_t$ **do**

- $\mathbf{x}_{t,n}^{\text{fg}} = \text{STN}^{-1}(\mathbf{o}_{t,n}, \mathbf{z}_{t,n}^{\text{where}})$
- $\boldsymbol{\gamma}_{t,n} = \text{STN}^{-1}(\mathbf{m}_{t,n} \cdot z_{t,n}^{\text{pres}} \sigma(-\mathbf{z}_{t,n}^{\text{depth}}), \mathbf{z}_{t,n}^{\text{where}})$
- $\boldsymbol{\gamma}_{t,n} = \text{normalize}(\boldsymbol{\gamma}_{t,n}, \forall n)$

end

$\mathbf{x}_t^{\text{fg}} = \sum_{n \in \mathcal{O}_t} \mathbf{x}_{t,n}^{\text{fg}} \boldsymbol{\gamma}_{t,n}$

```
/* Foreground mask rendering */
```

for $n \leftarrow 1$ **to** $\mathcal{O}_t$ **do**

- $\mathbf{M}_{t,n} = \text{STN}^{-1}(\mathbf{m}_{t,n}, \mathbf{z}_{t,n}^{\text{where}})$

end

$\mathbf{M}_t = \min(\sum_{n \in \mathcal{O}_t} \mathbf{M}_{t,n}, 1)$

```
/* Background rendering */
```

$\mathbf{e}^{\text{bg}} = \text{BackgroundEncoder}(\text{Concat}[\mathbf{x}_t, (1 - \mathbf{M}_t)])$

$\mathbf{z}^{\text{bg}} \sim \mathcal{N}(.|f_{nn}^{\text{bg}}(\mathbf{e}^{\text{bg}}))$

$\mathbf{x}_t^{\text{bg}} = \text{BackgroundDecoder}(\mathbf{z}^{\text{bg}})$

```
/* Foreground background combination */
```

$\mathbf{x}_t = \mathbf{x}_t^{\text{fg}} + (1 - \mathbf{M}_t) \odot \mathbf{x}_t^{\text{bg}}$

Output: $\mathbf{x}_t$

B QUALITATIVE RESULTS

In this section, we show the qualitative results of the experiments.

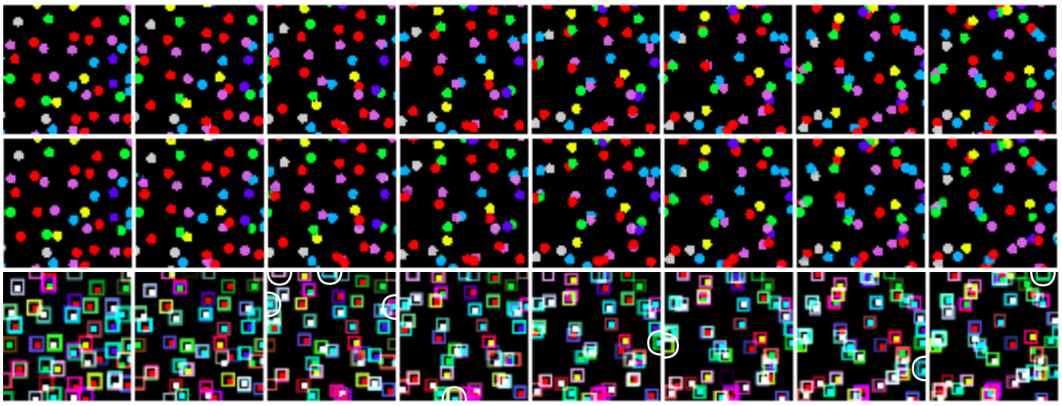

Figure 3: Qualitative samples from Moving dSprites (HD): a) Top row: Original image sequence, b) Middle row: Reconstructions of the images by SCALOR, c) Bottom row: Inferred bounding boxes superimposed on the object centroids represented by colored dots. Different object ids are shown by distinct bounding box colors. White circles show the discovered objects at each time-step. SCALOR identifies relatively large numbers of objects at each time-step while consistently keeping track of objects even after severe occlusion happens.

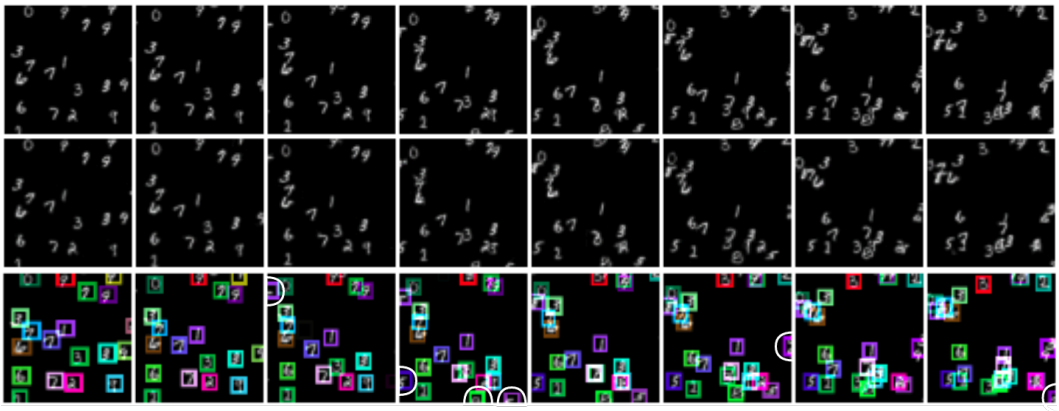

Figure 4: Qualitative samples for Moving MNIST (MD): a) Top row: Original image sequence; b) Middle row: Reconstructions of the images; c) Bottom row: Inferred bounding boxes.

C ADDITIONAL EXPERIMENT

Additional Experiment 1: Frequent Dense Discovery. This experiment evaluates the ability to discover many newly introduced objects across many time-steps. This is important because in many applications only key-frames of a video are available, i.e., frames at which significant changes happen. In the object tracking domain, an example of a key-frame is one in which many objects get introduced in the same frame due to a sudden change of the observer's view point. Our model can handle such challenging situations where there is a need to discover many objects in a single frame, as the powerful convolutional encoders of the discovery module enable us to discover many objects in each single frame. Figure 8 in the Appendix shows one such instance, where 10-15 objects are introduced at the first, fourth, and seventh time-steps, respectively, although it is still possible for objects to move in and out of the scene at other time-steps as well.

Additional Experiment 2: Complex Moving Background Separation. Another interesting yet challenging aspect of natural video processing is the presence of a complex moving background. SCALOR is able to separate such complex moving scenes from the foreground objects, as the background inference network is explicitly designed for that purpose. Figure 5.A provides one such

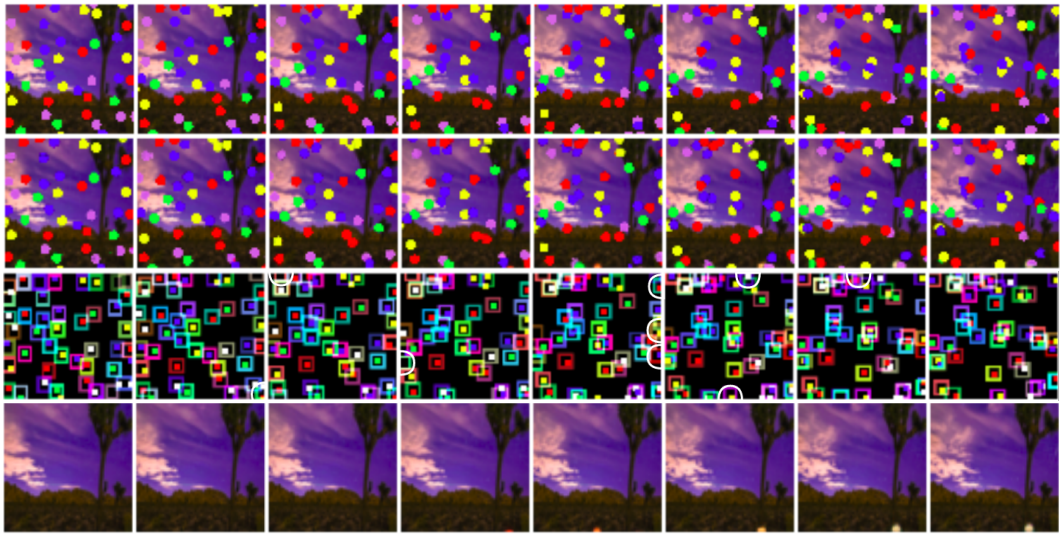

Figure 5: Complex Background Separation: a) First row: Original image; b) Second row: Reconstruction; c) Third row: Identified bounding boxes; d) Fourth row: Inferred background.

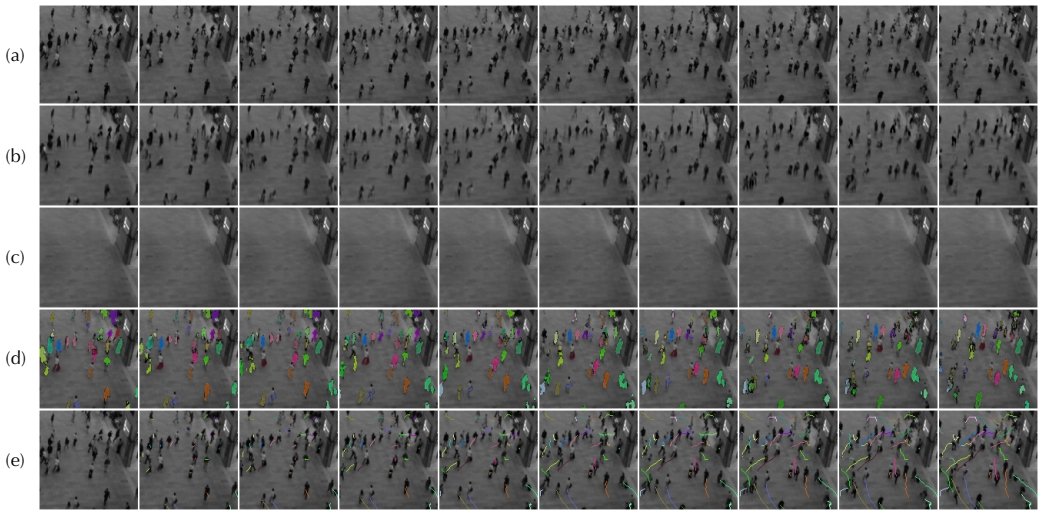

Figure 6: Qualitative result of SCALOR on Grand Central Station Dataset. (a) input sequence, (b) overall reconstruction of objects and background, (c) reconstruction of the extracted background, (d) segmentation for each object, colors indicate tracking ID, (e) extracted object trajectories.

instance, where the model is provided with a dynamic environment consisting of moving objects on top of a complex moving background. Figure 5.A(c) shows the inferred bounding boxes, while Figure 5.A(d) represents the inferred complex background. SCALOR is able to disentangle foreground objects from a complex moving background, while tracking them accurately and consistently. Even if some objects share similar colors to regions of the background, SCALOR is still able to separate them as the foreground correctly. Table 1 also provides tracking performance of the model for this setting as well to gain a better understanding of how more complex images affect the tracking quality. Interestingly, we observe that it attains a comparable degree of quality to the default no-background setting. These results are provided in Table 1 under "SCALOR – BG". MSE error for this case is 0.0099.

Additional Experiment 3: Generalization Ability. We conduct three sets of experiments to evaluate the ability to generalize to settings unseen at training time. In the first experiment, we investigate generalization ability to longer sequences, where the model is trained on trajectories of length 8 while being tested on trajectories of length 16. In the second experiment,we evaluate the generalizability to more crowded scenes, where the model is trained on scenes containing 10–15 objects

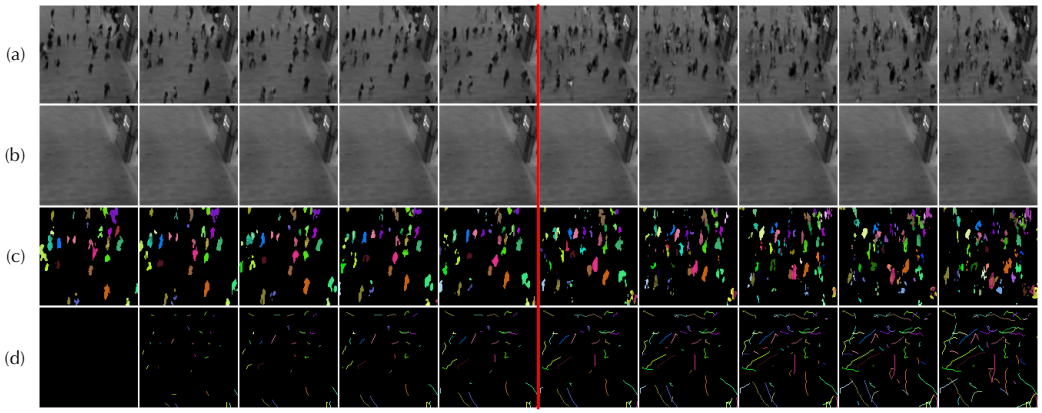

Figure 7: Conditional generation on Grand Central Station Dataset. The first 5 frames are observed, the last 5 frames are generated (after the red line). (a) overall reconstruction and generation, (b) conditional generation of background, (c) conditional generation of segmentation without background, (d) conditional generation of trajectories without background.

while it is tested on scenes containing 50–60 objects. This can be beneficial in real-world applications, as sometimes the training data is only available for less crowded instances of an environment, e.g. times of the day where streets are less crowded, while the model should be able to generalize to more crowded settings as well. The third experiment is intended to assess how well the model generalizes to unseen objects. The model is trained only on moving MNIST images containing digits 0 to 5, while being tested on images containing digits 6 to 9. Figures 9.A and 10.A in the Appendix show samples from these experiments. "SCALOR – LG" and "SCALOR – CG" in Table.1 provide tracking results for the "length generalization" and "count generalization" experiments discussed here. Furthermore, measured MSE is 0.0078 and 0.0081 respectively.

D ALGORITHM DETAILS

This section contains details of the algorithms used in discovery, propagation and image rendering processes.

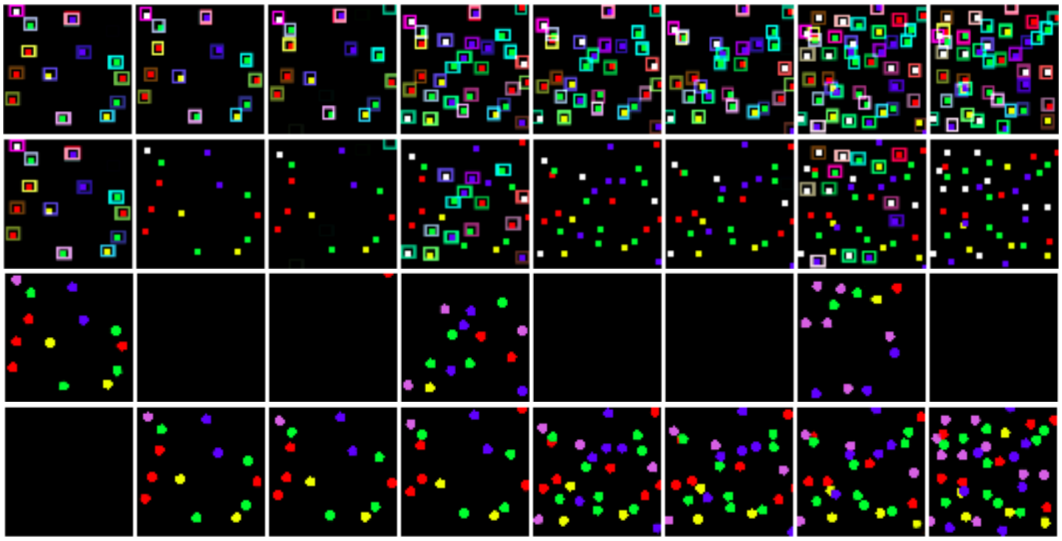

Figure 8: Frequent Dense Discovery experiment: a) First row shows all the bounding boxes. b) Second row shows bounding boxes inferred during discovery. c) Third row shows reconstructed images obtained from discovery step. d) Last row shows reconstructed images obtained from propagation step. Model is able to discover relatively large number of objects at each time-step while keeping the propagated objects consistent.

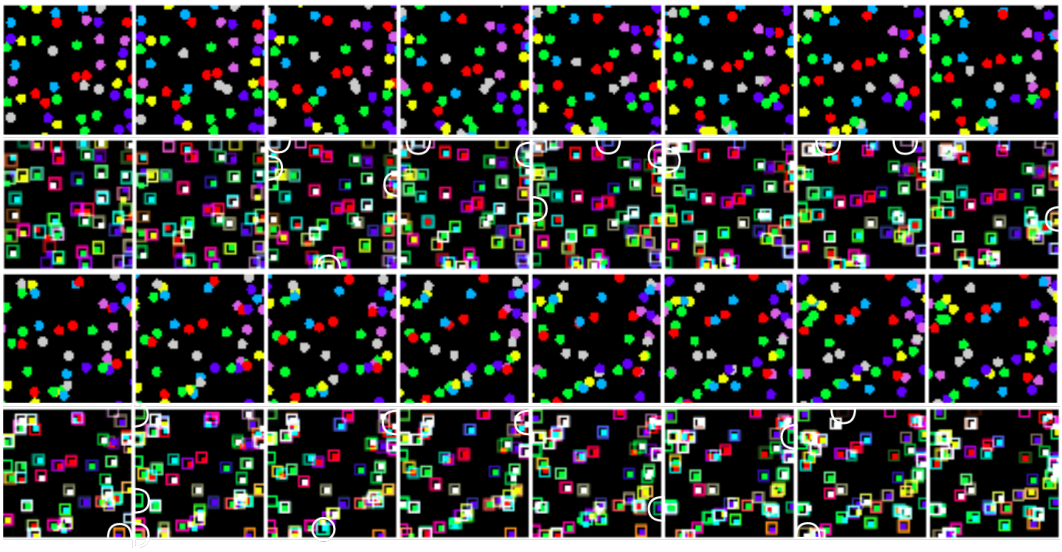

Figure 9: Generalization with respect to longer sequences. a) First row shows the original images of the first 8 time steps. b) Second row shows the inferred bounding boxes on the first 8 time steps. c) Third row shows the original images of the last 8 time-steps. d) Fourth row shows the inferred bounding boxes corresponding to the trajectories in the last 8 time-steps. The model is able to maintain ids of the inferred objects over the longer sequences and also discover newly introduced objects.

E APPENDIX: ADDITIONAL EXPERIMENTAL RESULTS

E.1 FREQUENT DENSE DISCOVERY

Figure 8.A shows sample images obtained in experiment 2 (Frequent Dense Discovery). SCALOR is able to accurately introduce new bounding boxes for the newly discovered objects. 10–15 objects are introduced at random places at certain time-steps. Although the performance is reasonable, we noticed some limitations when the newly introduced objects overlap significantly with the already present objects. In such cases, sometimes the propagation detects the newly introduced object as part of the previously overlapping object and infers a new $\mathbf{z}^{\text{what}}$ for the already present object rather than discovering a new object.

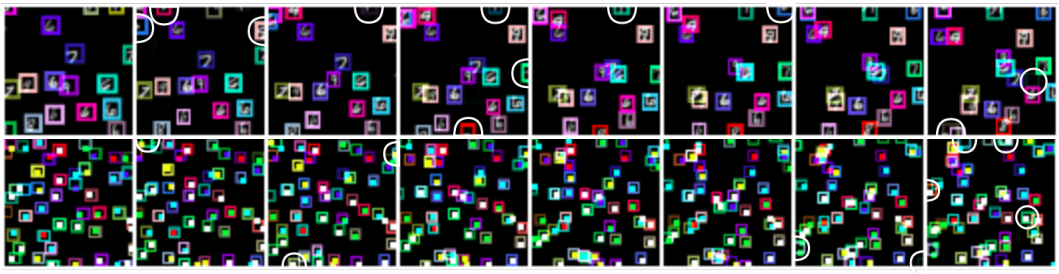

Figure 10: Generalization Experiment: a) Top row is the sample to demonstrate generalization with respect to other shapes. b) Bottom row image is the sample to demonstrate generalization with respect to more objects.

E.2 Generalization to unseen data

This section presents the results obtained during generalization experiments. Figure 9.A shows the ability of SCALOR to scale to longer time steps. While the model has been shown only 8 time steps during training, it is tested on sequences of length 16 at test time.

Figure 10.A (a) shows the generalizability of the model to unseen objects when it is trained on scenes containing only MNIST digits of shape 1–5 but tested on scenes containing digits 6–9. Figure 10.A (b) demonstrates bounding boxes of a sample sequence on a scene containing 50–60 objects while it was trained on scenes containing only 10–15.

E.3 Very High Density

Figure 11.A shows some samples from the Very High Density experiment. This experiment places 100–120 objects in the overall environment, 99 of which are visible at every time step on average. However, because of memory efficiency, the model only contains 64 cells for discovery at each time step. Interestingly, it is shown that the model tries its best to identify as many objects as possible in the first time step, however, due to the lack of sufficient number of cells, it will discover whatever object remains in the second time step. Furthermore, in the case that too many objects are densely packed in one region of the space, this delayed discovery might again happen due to a lack of a sufficient number of cells in that region of the space.

E.4 Ability to handle overlap and occlusion

Figures 12.A and 13.A show samples of the dSprites (HD) and MNIST(MD) settings in which multiple objects overlap each other and severe occlusion happens. As shown, the identities of the objects are preserved and identity swaps do not occur between them even if several objects overlap with each other.

E.5 Dataset Detail and Qualitative results of Grand Central Station Dataset

During the experiments, we spatially split the video into 8 parts and create a dataset of 400k frames in total. We choose the first 360k frames for training, and 40k frames for testing. Since the movement of pedestrians is too slow under 25 fps in the original video, we treat every other 7 frames as two consecutive frames in the dataset sequence. The length of the input sequence is 10, and each image is resized to 128×128.

We provide qualitative results in Figures 14.A to 19.A. Figures 14.A to 16.A show the examples for inference. Rows from top to bottom are input sequence, overall reconstruction, extracted background, foreground segmentation mask with IDs, center position from each $\mathbf{z}_{t,n}^{\text{pos}}$ latent, and extracted trajectories based on the transition of center position. Figures 17.A to 19.A show the examples from conditional generation, where images starting from column 6 are generation results (after red line). Rows from top to bottom are conditional generations of the overall images, conditional generations of background, extracted conditional generation of each segmentation mask, and conditional generation of trajectories.

F Detailed Analysis of Ablation Study and Computational Efficiency Comparison

Ablation Study We perform an ablation study on the proposed method to show the importance of the rejection mechanism and the initial bias of $\mathbf{z}^{\text{pres}}$ in propagation step. Here, we apply the model on the

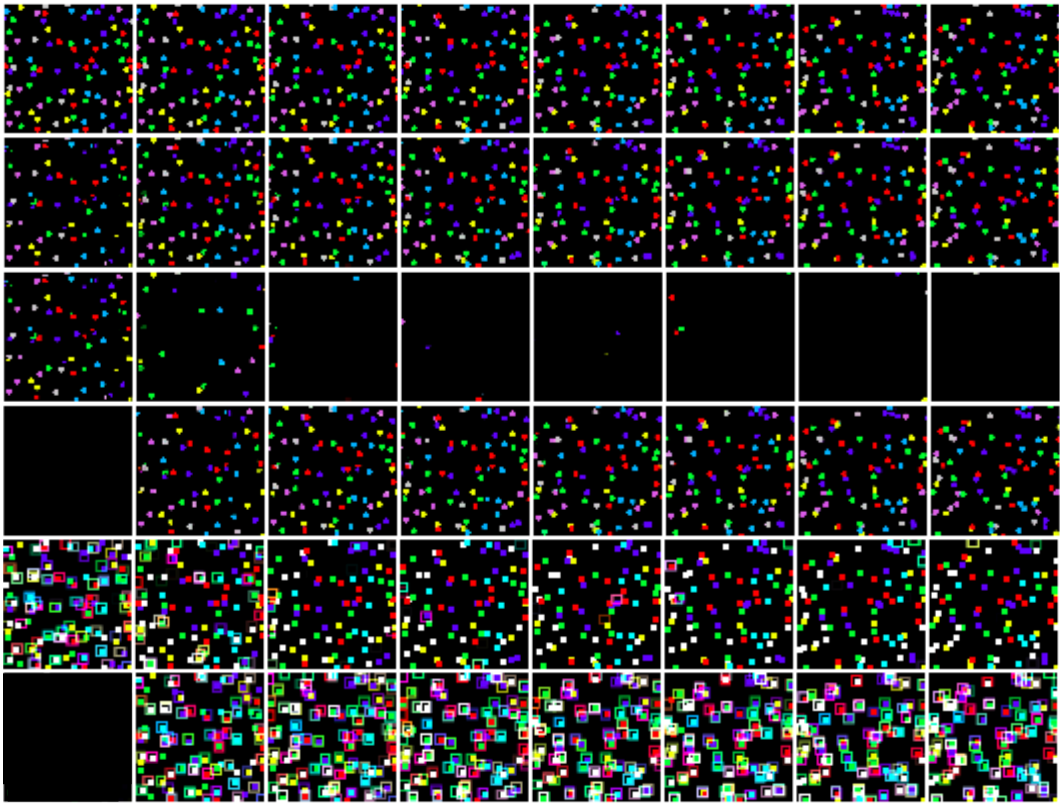

Figure 11: Sample from Very High Density Setting: a) First row: original Image, b) Second row: overall reconstruction, c) Third row: discovery Reconstruction, c) Fourth row: Propagation Reconstruction, d) Fifth row: bounding boxes obtained from Discovery step, e) Sixth row: bounding boxes of propagation step. SCALOR is trying its best to identify as many objects as possible in the first time-step however since number of cells is less than the number of objects, the rest of the objects get discovered in the next time-steps

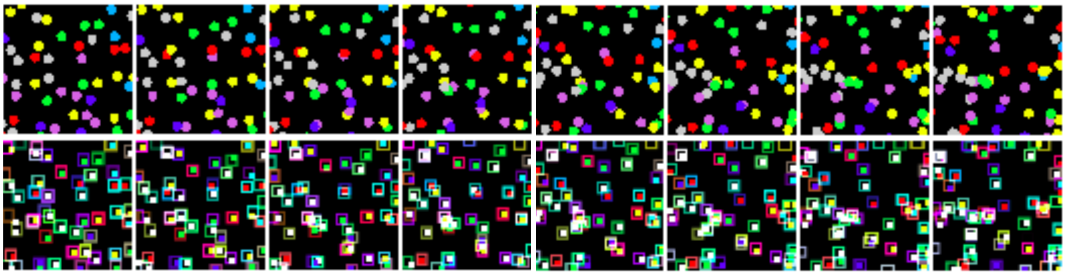

Figure 12: Highly occluded dSprites setting: a) The top row shows a sequence where object overlap happens several times between different objects. b) the bottom row demonstrates the inferred bounding boxes.

moving dSprites dataset and evaluate it using two metrics. First metric is the propagation rate, where we measure the probability of an object being propagated after being discovered at the previous time step. The second metric is the number of discovered objects in the whole sequence. Figure 2 (a) (b) shows the quantitative result for both metrics. With the use of rejection, the propagation rate converges to a value of 99%, meaning the model is able to learn to correctly propagate existing objects. While, in the model without rejection, the propagation rate decreases to 20%. The curve corresponding to the number of discovered objects in Figure 2 (b) further explains the behavior of the model with regards to the propagation rate. As is shown, the model without rejection attains a higher discovery rate. This is because when the propagation module is unable to work properly, the model has to explain the objects by re-discovering the same objects consistently at every time-step, in other words the discovery module dominates. In contrast, model having the rejection phase obtains a relatively low discovery rate since the discovery module only needs to detect new objects

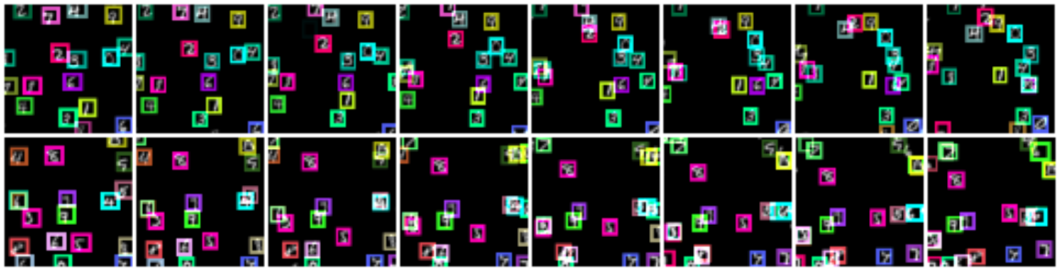

Figure 13: Highly occluded setting: Two sample sequences where object overlap happens several times between different MNIST digits.

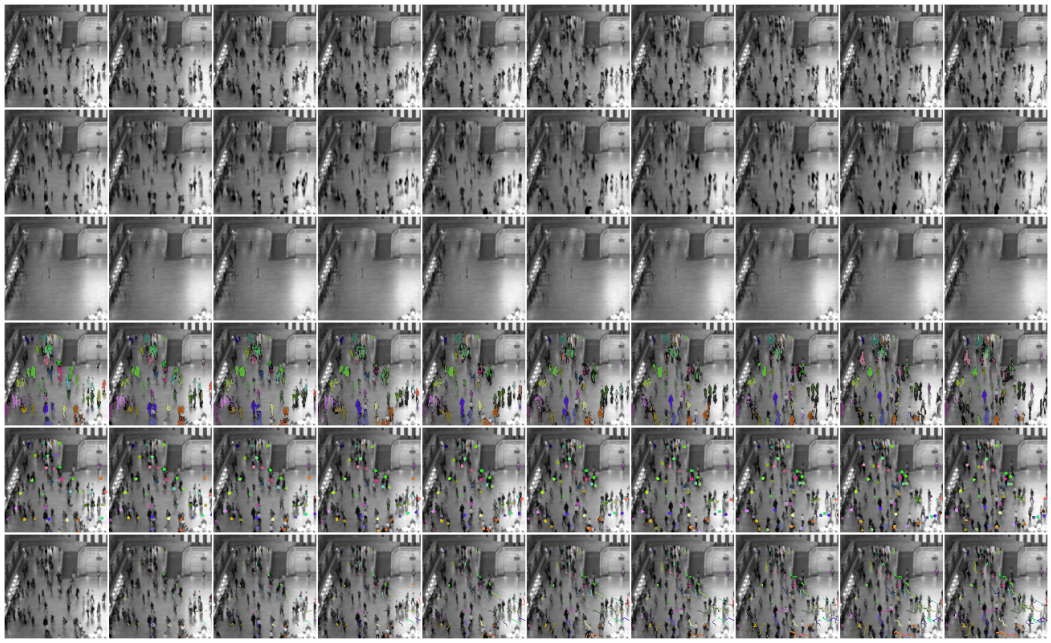

Figure 14: Detection example 1.

in the scene. We further compare models trained with and without some initial bias on $\mathbf{z}^{\text{pres}}$ in the propagation step. As shown in the figure, the initial bias encourages propagation in the early training iterations, which allows the model with rejection to converge faster. Note that, in all ablation experiments, the attention mechanism in propagation is replaced by a full-image attention, i.e. the propagation module takes the entire image as input. We found that with the introduction of a local attention mechanism, the propagation rate can reach up to 92% even without the rejection step, but this value shows high variance over different training examples and does not decrease during training.

Computational Efficiency We also empirically compare the computational efficiency of SCALOR to SQAIR, presented in Figure 2. More specifically we measure (a) inference time when model is provided with different number of objects in a scene and (b) how fast the model is trained. In (a) we provide a hypothetical situation where there are N objects in the first frame and the model should discover and propagate them accordingly. Number of cells in discovery module of SCALOR as well as discovery module capacity in SQAIR are set to match this number. Figure 2(c) demonstrates how much time one forward step takes as the number of objects/discovery capacity increases from 4 to 64. SQAIR's speed decreases linearly as both its discovery and propagation mechanisms process objects sequentially. In contrast, SCALOR does not suffer from such phenomena as discovery and propagation are done in parallel. Figure 2 (d) shows MSE Convergence error vs. training time when both models are trained on the MNIST VLD setting. SCALOR is not only able to converge to a better minimum, but it does so orders of magnitude faster.

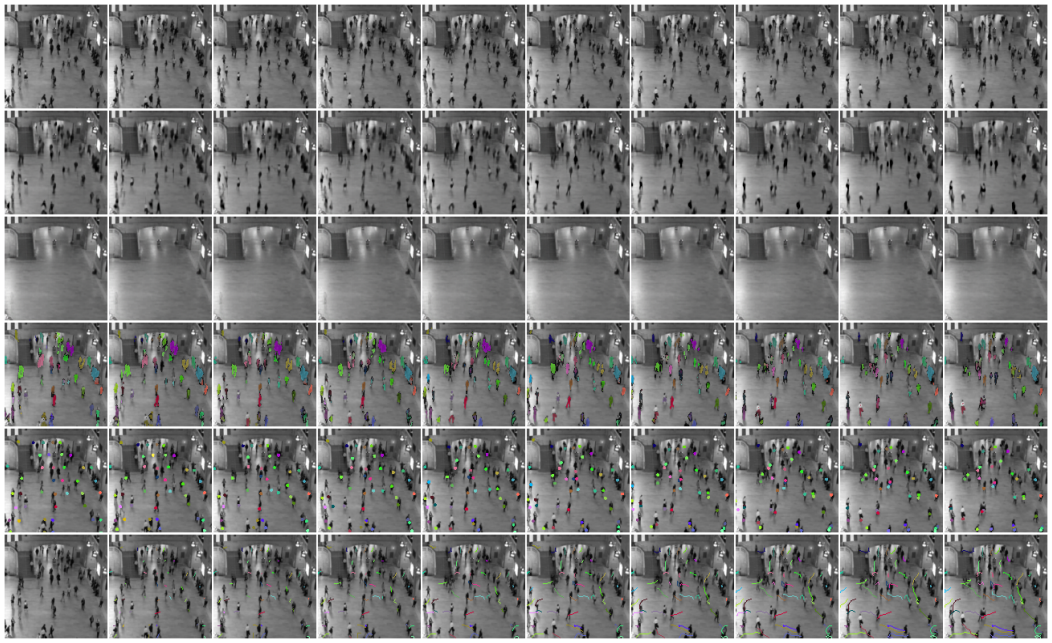

Figure 15: Detection example 2.

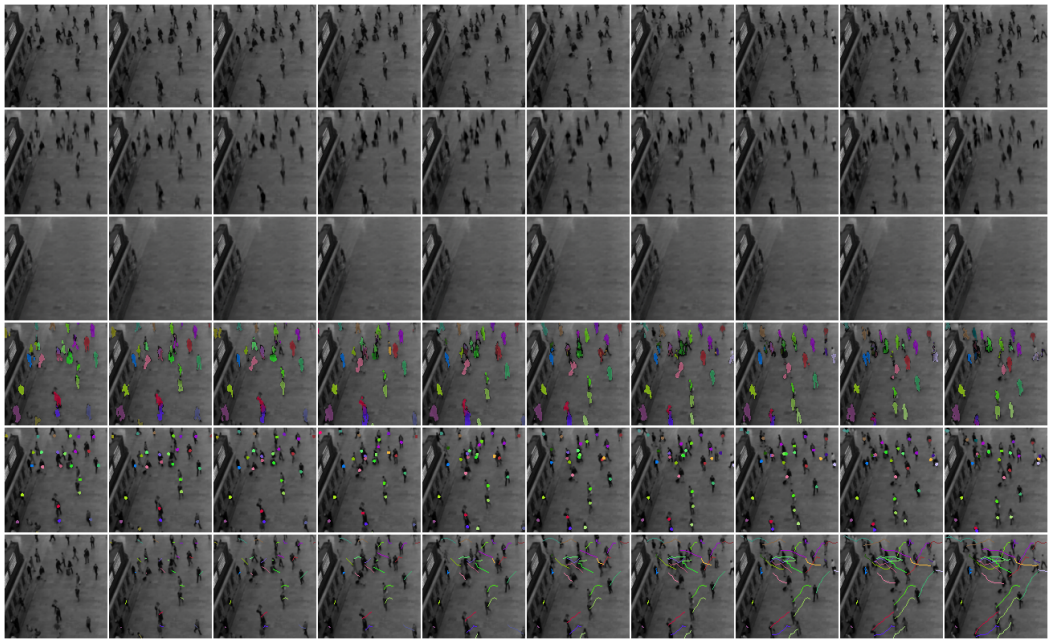

Figure 16: Detection example 3.

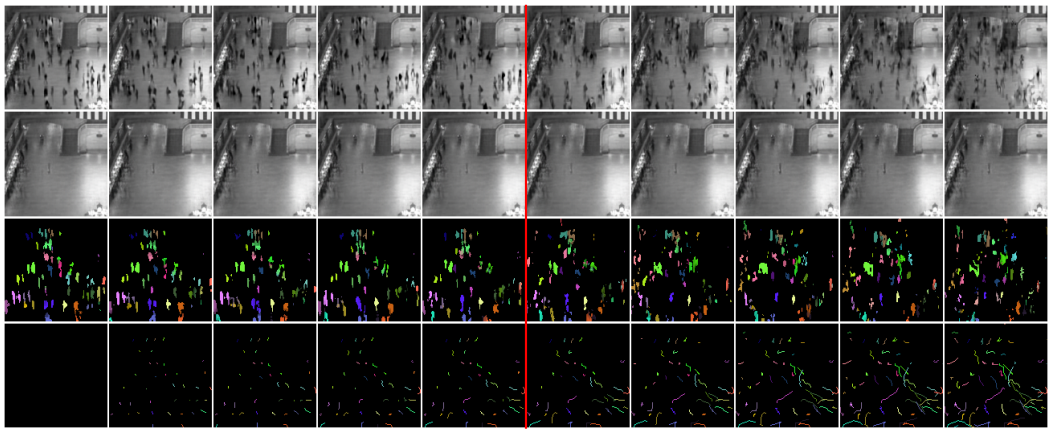

Figure 17: Generation example 1.

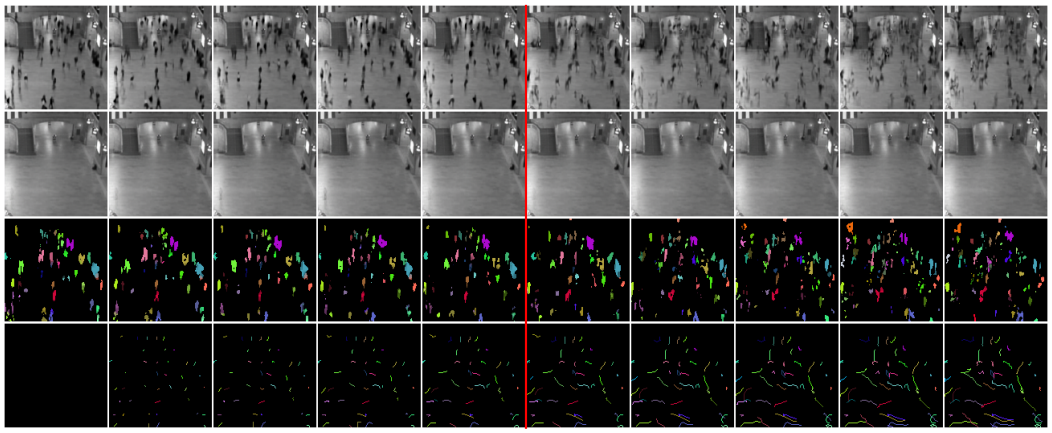

Figure 18: Generation example 2.

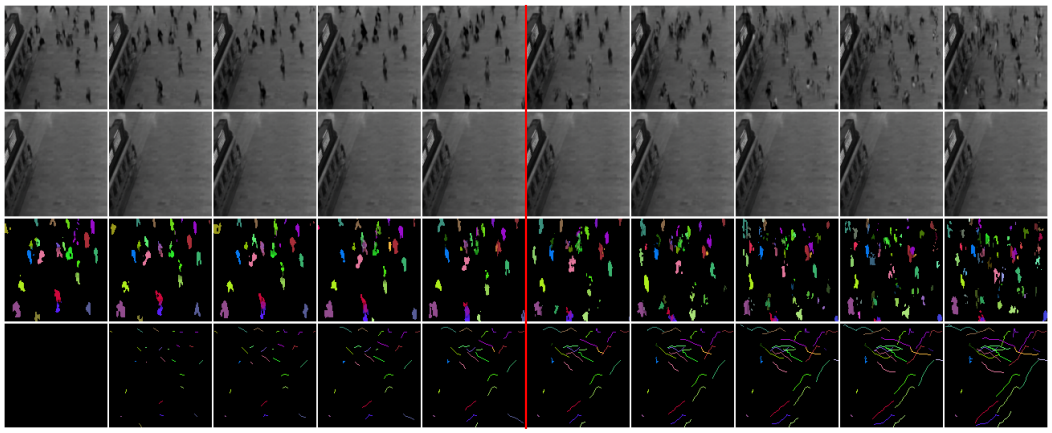

Figure 19: Generation example 3.

G MODEL ARCHITECTURE DETAILS

In this section, we provide additional details of the architecture and hyperparameters for the SCALOR model on pedestrian detection. For one input frame, the network uses a fully convolutional image encoder to obtain a $H \times W$ feature map. The feature map is input into a convolutional LSTM to model the sequential information along the sequence. The convolutional LSTM is shared across discovery module and propagation module for updating the tracker hidden state, and proposing new objects. The discovery module and propagation module share the same glimpse encoder and decoder. The encoder has a convolutional network followed by one fully connected layer, while the glimpse decoder uses a fully convolutional network with sub-pixel layer (Shi et al., 2016) for upsampling. The background module shares a similar structure with the glimpse encoder and decoder, while having a 4-dimensional input but producing a 3-dimensional output. We use GRUs for trackers in propagation and all other temporal prior transition networks.

We choose a batch size of 20 for the natural scene experiments and a batch size of 4 (due to GPU memory limitation) for MNIST/dSprites experiments. learning rate is fixes at 4e-5 for natural image experiments and 1e-4 for dSprites/MNIST experiments and we use RMSProp for optimization during training. The standard deviation of the image distribution is chosen to be 0.1. The prior for all Gaussian posteriors are set to standard normal. In pedestrian tracking dataset, we constrain the range of $\mathbf{z}^{\text{scale}}$ so that the inferred width can vary from 5.2 pixels to 11.7 pixels, and the height can vary from 12.0 to 28.8, and both with a prior of the middle value in discovery. Similarly, we constrain $\mathbf{z}^{\text{scale}}$ on synthetic dataset so that it can vary from 0 to twice/ or 1.5 $\times$ the actual object size depending on the experiment. In our implementation, $\mathbf{z}^{\text{pos}}$ is model as the deviation of position instead of the global position coordinate. The prior for z^{pres} in discovery is set to be 0.1 at the beginning of training and to quickly anneal to 1e-3 for natural image experiments and 1e-8 for dSprites/MNIST experiments. The temperature used for modelling z^{pres} is set to be 1.0 at the beginning and anneal to 0.5 after 20k iterations. Since it is also interesting to see new objects introduce into the scene during generation. We assume a distribution shift on z^{pres} to allow more objects to be generated in discovery phase, i.e. we assume the prior for new object appearing to be 0.1.

The rest of the architecture details are described in the following tables.

Name	Value	Comment
$\mathbf{z}^{\text{depth}}$ dim	1	
$\mathbf{z}^{\text{scale}}$ dim	2	for x and y axis
$\mathbf{z}^{\text{what}}$ dim	64	
$\mathbf{z}^{\text{pres}}$ dim	1	
$\mathbf{z}^{\text{pos}}$ dim	2	for x and y axis
glimpse shape	$(64, 64)$	for $\mathbf{o}, \boldsymbol{\alpha}$
image shape	$(128, 128)$	
H	8	
W	8	

Image Encoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	3		
Conv 4×4	16	2	GN(4)/CELU
Conv 4×4	32	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 1×1	32	1	GN(8)/CELU

Mask Encoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	1		
Conv 4×4	16	2	GN(4)/CELU
Conv 4×4	32	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 1×1	32	1	GN(8)/CELU

Glimpse Encoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	3		
Conv 4×4	16	2	GN(4)/CELU
Conv 4×4	32	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	128	2	GN(16)/CELU
Conv 4×4	128	1	GN(16)/CELU
Linear	128		

Glimpse Decoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	64		
Conv 1×1	128	1	GN(16)/CELU
Conv 1×1	1024	1	GN(16)/CELU
ConvSub(4)	64	1	GN(8)/CELU
Conv 3×3	64	1	GN(8)/CELU
Conv 1×1	1024	1	GN(8)/CELU
ConvSub(4)	64	1	GN(8)/CELU
Conv 3×3	64	1	GN(8)/CELU
Conv 1×1	128	1	GN(8)/CELU
ConvSub(2)	32	1	GN(8)/CELU
Conv 3×3	32	1	GN(8)/CELU
Conv 1×1	64	1	GN(8)/CELU
ConvSub(2)	16	1	GN(4)/CELU
Conv 3×3	8	1	GN(4)/CELU
Conv 3×3	3	1	GN(4)/CELU

Background Encoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	$4 \times 128 \times 128$		
Conv 4×4	16	2	GN(4)/CELU
Conv 4×4	32	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	64	2	GN(8)/CELU
Conv 4×4	20	1	

Background Decoder

Layer	Size/Ch.	Stride	Norm./Act.
Input	20		
Conv 1×1	256	1	GN(16)/CELU
Conv 1×1	4096	1	
ConvSub(4)	256	1	GN(16)/CELU
Conv 3×3	256	1	GN(16)/CELU
Conv 1×1	1024	1	
ConvSub(4)	128	1	GN(16)/CELU
Conv 3×3	128	1	GN(16)/CELU
Conv 1×1	256	1	
ConvSub(2)	64	1	GN(8)/CELU
Conv 3×3	64	1	GN(8)/CELU
Conv 1×1	256	1	
ConvSub(4)	16	1	GN(4)/CELU
Conv 3×3	16	1	GN(4)/CELU
Conv 3×3	16	1	GN(4)/CELU
Conv 3×3	3	1	