SCALABLE OBJECT-ORIENTED SEQUENTIAL GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The most significant limitation of previous approaches to unsupervised learning for object-oriented representation is its 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 generative model for Scalable Sequential Object-Oriented Representation. With our spatially parallel attention and proposal-rejection mechanism, SCALOR is a scalable model that can deal with orders of magnitude more objects than previous models. Besides, we introduce the background model so that it can model the foreground objects and complex background together. In experiments on large-scale MNIST and DSprite datasets, we demonstrate that SCALOR can deal with scenes with near 100 objects as well as modeling complex natural background. Importantly, using SCALOR, we demonstrate for the first time a result of modeling natural scenes with several tens of moving objects.

1 INTRODUCTION

Unsupervised learning of structured representations 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 of 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 (Kosiorek et al., 2018b; Hsieh et al., 2018) that have been proposed for unsupervised object-oriented representation learning of temporal scenes, SQAIR (Kosiorek et al., 2018b) 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 and 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. Processing every object in an image sequentially, SQAIR has a fundamental limitation in scaling up to scenes with a large number of objects. As such, in the paper, the method is demonstrated for videos with only a few objects such as MNIST digits. Considering the complexity of typical natural scenes, it is thus a challenge of the highest priority to scale robustly to scenes with a large number of objects. Second, SQAIR conspicuously lacks any form of background modeling and thus only copes with scenes without any background, whereas natural scenes have a particularly 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 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 dynamic background on synthetic datasets. We evaluate and demonstrate SCALOR on natural videos as well with tens of objects with complex background.

The contribution of this work are:

1. We propose the SCALOR model that significantly improves the scalability with regard to the the object density (two orders of magnitude). It is applicable to nearly a hundred objects with comparable computation time to SQAIR, which scales only to a few objects.
2. We parallelize the propagation–discovery process by introducing the propose–reject model, reducing the time complexity to $\mathcal{O}(1)$.
3. The proposed model can model scenes with complex background.
4. As a stochastic generative model, we demonstrate the working of SCALOR not only on natural images for the first time but also at a significantly high complexity with tens of objects and background both moving.

2 Preliminaries: Sequential Attend Infer Repeat (SQAIR)

Before introducing our novel SCALOR approach, we first review the existing SQAIR approach. 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 $\mathbf{z}_t^{\mathcal{O}} = \{\mathbf{z}_{t,n}\}_{n\in\mathcal{O}_t}$. Each latent variable $\mathbf{z}_n$ for an object n consists of the factors $(z_n^{\text{pres}}, z_n^{\text{where}}, z_n^{\text{what}})$, 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:

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

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

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

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 a relational 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., 2016a). Consequently, SQAIR has $O(|\mathcal{O}_t|)$ time complexity per step t. In previous work (Crawford & Pineau, 2019), it is demonstrated that this sequential approach fails beyond the scale of a few objects. Moreover, SQAIR lacks any model for the background and its temporal transitions.

3 THE PROPOSED MODEL: SCALOR

3.1 GENERATIVE PROCESS

SCALOR assumes that an image $\mathbf{x}_t$ is generated by two decomposed latent representations, the background $\mathbf{z}_t^{\mathrm{B}}$ and foreground $\mathbf{z}_t^{\mathcal{O}}$. The foreground representation is further factorized into a set of object representations $\mathbf{z}_t^{\mathcal{O}} = \{\mathbf{z}_{t,n}\}_{n\in\mathcal{O}_t}$. SCALOR represents an object by $\mathbf{z}_n = (z_n^{\text{pres}}, z_n^{\text{where}}, z_n^{\text{what}}, z_n^{\text{depth}})$. The depth representation, which is missing in SQAIR, helps modeling object occlusion, while the foreground mask computes from the z_n^{what} model the distinction between background and foreground. The appearance representation $\mathbf{z}^{\text{what}}$ is a typical continuous vector representation (e.g., as in VAE), and z_n^{where} is further decomposed into position z_n^{pos} and scale z_n^{scale}. Following SQAIR, adopts a propagation–discovery model, but the version proposed for SCALOR resolves the scalability problem of SQAIR. Based on this, the generative process of SCALOR can be written as:

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

As shown, the generation process is decomposed into the following four modules. In SCALOR, the **propagation** is achieved by the following model

$$p(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{t-1}) = \prod_{n\in\mathcal{O}_t} p(\mathbf{z}_{t,n}^{\text{pres}}|\mathbf{z}_{<t,n}) \left(p(\mathbf{z}_{t,n}^{\text{where}}|\mathbf{z}_{<t,n})p(\mathbf{z}_{t,n}^{\text{what}}|\mathbf{z}_{<t,n})\right)^{\mathbf{z}_{t,n}^{\text{pres}}}, \tag{4}$$

where $p(\mathbf{z}_{t,n}^{\text{pres}}|\mathbf{z}_{t-1,n})$ is a Bernoulli distribution with parameter $\beta_{t,n}$. That is, the distribution of what and where is only defined when it is propagated. To implement this, for each object n we assign a *tracker* RNN denoted by its hidden state $h_{t,n}$. The tracker 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 by $\beta_{t,n} = f_{\text{nn}}(h_{t,n})$. If $\mathbf{z}_{t,n}^{\text{pres}} = 0$, the object n is not propagated and the tracker RNN is deleted. Importantly, the propagation in SCALOR is fully parallel, unlike SQAIR where the propagation is sequential, due to the use of a relational RNN.

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

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

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 $m_{t,n}$, which is used to make the rejection decision. Specifically, if the overlap between the mask of a proposed object and the mask of a propagated object is over a threshold τ, we reject the proposal. This procedure can be described as (i) $\tilde{\mathbf{z}}_t^{\mathcal{D}} \sim p(\tilde{\mathbf{z}}_t^{\mathcal{D}})$ and (ii) $\mathbf{z}_t^{\mathcal{D}} = f_{\text{reject}}(\tilde{\mathbf{z}}_t^{\mathcal{D}}, \mathbf{z}_t^{P}, \tau)$ and $\mathbf{z}_t^{\mathcal{D}} \subseteq \tilde{\mathbf{z}}_t^{\mathcal{D}}$. Even if we use a deterministic function to implement the rejection, it can be a design choice to implement this as a stochastic decision. While the basic 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 $p(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}) = p(\tilde{\mathbf{z}}_t^{\mathcal{D}}) \prod_{h,w=1}^{HW} p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}})$, where $p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_t^{\mathcal{D}}) = p(\mathbf{z}_{t,h,w}^{\text{pres}}|\mathbf{z}_t^{P}, \tilde{\mathbf{z}}_t^{\mathcal{D}})p(\tilde{\mathbf{z}}_{t,h,w}^{\text{where}})^{\mathbf{z}_{t,h,w}^{\text{pres}}}p(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}})^{\mathbf{z}_{t,h,w}^{\text{pres}}}$.

Background Transition. Unlike SQAIR, SCALOR is endowed with a background model. The background transition $p(\mathbf{z}_t^{\mathrm{B}}|\mathbf{z}_{t-1}^{\mathrm{B}})$ is conditioned on the background latent variable from the previous time step.

Rendering. For rendering, SCALOR needs to combine the foreground and background. This is done by learning foreground and background masks. Give the sampled $\mathbf{z}_t^{what,i}$ and $\mathbf{z}_t^{where,i}$, the model first computes $\mathbf{o}_t^i, \boldsymbol{\alpha}_t^i = f(\mathbf{z}_t^{what,i})$, where $\mathbf{o}_t^i$ is the RGB color of the object and $\boldsymbol{\alpha}_t^i$ is the one

channel segmentation mask. Then, we invoke a spatial transformer network (Jaderberg et al., 2015) to transform the local mask from patch size to image size $\tilde{\boldsymbol{\alpha}}_t^i = STN^{-1}(\boldsymbol{\alpha}_t^i, \mathbf{z}_t^{where,i})$. The summation of the mask $M = \min(\sum_{i=1}^{O} \tilde{\boldsymbol{\alpha}}_t^i, 1)$ over all objects i is treated as the foreground mask for the detection network. The background X_{bg} is generated directly from background decoder, while the foreground is computed by aggregating the generation of all objects. Here, using the depth information of objects $\mathbf{z}_t^{depth,i}$, the model computes an importance weight on each pixel over the object that appears at that pixel. This can be computed in parallel by first obtaining $\gamma_t^i = \tilde{\boldsymbol{\alpha}}_t^i \mathbf{z}_t^{pres,i} \sigma(-\mathbf{z}_t^{depth,i})$ and the normalization $\tilde{\gamma}_t^i = \frac{\gamma_t^i}{\sum_{j=1}^{O} \gamma_t^j}$. To complete the rendering of the foreground image, the model transforms the RGB local patches into full resolution $\tilde{\mathbf{o}}_t^i = STN^{-1}(\mathbf{o}_t^i, \mathbf{z}_t^{where,i})$ and computes a pixel-wise sum over all objects $X_{\text{fg}} = \sum_{i=1}^{O} \tilde{\mathbf{o}}_t^i \tilde{\gamma}_t^i$. The final output is the combination of foreground and background $X = X_{\text{fg}} + (1 - M) X_{\text{bg}}$.

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:

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

Posterior Propagation $q(\mathbf{z}_t^{\mathcal{P}}|\mathbf{z}_{<t}, \mathbf{x}_{\leq t})$ is similar to the propagation in generation, except that now observations $\mathbf{x}_{\leq t}$ are provided through an RNN encoding. Here, $q(\mathbf{z}_t^i|\mathbf{z}_{<t}^i, \mathbf{x}_{\leq t}) = q(\mathbf{z}_t^i|\mathbf{z}_{<t}^i, \mathbf{a}_t^i)$, where $\mathbf{a}_t^i$ is the attended convolutional feature for object i. To compute the attention for object i, we use the previous position $\mathbf{z}_{t-1}^{pos,i}$ as the location and extract half the width and height of the convolutional feature map and resize it as the original size using bilinear interpolation. This attention mechanism is inspired by the observation that only part of the image contains information for tracking the object, and the inductive bias that objects will not shift across a large distance within a short time span.

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 $\mathbf{x}_t$, i.e., $q(\tilde{\mathbf{z}}_{<t}^{\mathcal{D}}|\mathbf{x}_t) = \prod_{h,w=1}^{HW} Q(\tilde{\mathbf{z}}_{t,h,w}^{\mathcal{D}}|\mathbf{x}_t)$. Here, the observation $\mathbf{x}_t$ is encoded into the feature map of dimensionality $H \times W \times D$ using a convolutional network. 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. 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 two challenges due to the power of the spatially-parallel discovery proposal: (i) the discovery module dominates the propagation and thus all objects are often rediscovered at every time step while propagating nothing, and (ii) the discovery module may detect anew an object that is already propagated. These behaviors can easily be observed because they are actually helpful in obtaining a good reconstruction error, despite not being helpful in obtaining the desired latent structure. That is, in the case of (i), the reconstruction does not care from where (either from propagation or from discovery) an object is sourced, as long as it can use it to reconstruct well. Similarly, for (ii), a duplicate detection of an object allows overdrawing on top of the same object and this overdrawing may improve the reconstruction.

We found that the first problem can rather easily be resolved by biasing the initial network parameter such that it has a high propagation probability at the beginning of the training. This makes the model first try to explain the observation through propagation and then the discovery module takes the remaining part as the training proceeds. The second problem is overcome by the rejection procedure, which is implemented the same way as in the generation. Thus, the final discovery model in posterior form can be written as: $q(\mathbf{z}_t^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \mathbf{x}_{\leq t}) = q(\tilde{\mathbf{z}}_{<t}^{\mathcal{D}}|\mathbf{x}_{\leq t}) \prod_{h,w=1}^{HW} p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_{<t}^{\mathcal{D}})$, where $p(\mathbf{z}_{t,h,w}^{\mathcal{D}}|\mathbf{z}_t^{\mathcal{P}}, \tilde{\mathbf{z}}_{<t}^{\mathcal{D}}) = p(\mathbf{z}_{t,h,w}^{\text{pres}}|\mathbf{z}_t^{P}, \tilde{\mathbf{z}}_{<t}^{\mathcal{D}})(p(\tilde{\mathbf{z}}_{t,h,w}^{\text{where}}) p(\tilde{\mathbf{z}}_{t,h,w}^{\text{what}}))^{\mathbf{z}_{t,h,w}^{\text{pres}}}$.

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

$$\sum_{t=1}^{T} \mathbb{E}_{q_\phi(\mathbf{z}_{<t}|\mathbf{x}_{<t})} \Big[\mathbb{E}_{q_\phi(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t})} \left[\log p_\theta(\mathbf{x}_t|\mathbf{z}_t) \right] - D_{\mathrm{KL}} \left[q_\phi(\mathbf{z}_t|\mathbf{z}_{<t}, \mathbf{x}_{\leq t}) \parallel p_\theta(\mathbf{z}_t|\mathbf{z}_{<t}) \right] \Big]. \quad (7)$$

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 $\mathbf{z}^{\text{pres}}$. SQAIR uses the IWAE objective (Burda et al., 2015) and VIMCO gradient estimator (Mnih & Rezende, 2016), whereas our model works well and is stable with a simpler training method using the VAE ELBO objective and Gumbel-Softmax gradient estimation.

Posterior Background. The posterior of the background $q(\mathbf{z}_t^{\mathrm{B}}|\mathbf{z}_{t-1}^{\mathrm{B}}, \mathbf{x}_t)$ 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.

4 Related Work

Although conventional approaches to object tracking use supervised models Kosiorek et al. (2017), more recent endeavors into the field have focused on exploiting certain inductive biases in the model as a form of self-supervision to guide the learning procedure. AIR Eslami et al. (2016b), one of the seminal works in the field of object detection, imposes a structure on the latent space of a deep probabilistic model, making it more interpretable for object detection. In order to achieve this, the model uses a VAE-like architecture combined with an RNN, which optimizes the Evidence Lower Bound (ELBO). However AIR's power is limited by its sequential inference procedure using RNNs. Inspired by AIR, SPAIR Crawford & Pineau (2019) is more robust by using convolutional feature maps to parallelize inference. Other interesting directions such as iterative inference for scnece segmentataion has also been explored by Greff et al. (2019). However, their proposed method is more computataionally demanding due to the iterative nature of their inference procedure. Another interesting related work is that of Greff et al. (2017) where scenes are decomposed into objects using a neual EM algorithm by inferring inferring parameters of spatial mixture models.

In case of multi-object tracking, most simialr work to ours is that of Kosiorek et al. (2018a) which learns structured lateents for objects through time and is also a generative model. But their model is inherently limited and less robust compared to this work, as the RNNs have less power compared to convolutional features introduced in this work. Also their model is unable to detect complex backgrounds or scale to highly dense scenes. van Steenkiste et al. (2018) builds upon the concepts introduced in Neural Expectataion Maximization to track the objects in addition to modelling their interaction. However they learn an unstructured latent variable which cannot be interpreted easily. He et al. (2019) also discuss similar models, but their model is tracker-based instead of object based. They also use a memory module to help learn the transition of the trackers.

5 Experiments

We next describe a series of experiments that showcase several different aspects of our model. Furthermore, we also provide quantitative results of how well the model performs in terms of tracking as well as its ability to generalize to additional settings beyond what it has been trained for.

5.1 Task 1: Large-Scale MNIST and DSprite Shapes

The first task demonstrates our model's performance when it is exposed to a relatively simple environment. Specifically, as in previous work Crawford & Pineau (2019), we consider a toy dataset of moving DSprite shapes as well as a dataset consisting of moving MNIST digits. In all of our experiments, the objects can move in and out of the scene. More specifically, the video is a specific viewpoint of a larger environment, where the objects can bounce off the walls and re-enter the scene. Therefore, while there is a fixed number of objects in the scene, only a subset of them are visible in each time frame. The color of the bounding boxes in the qualitative image samples represent the consistency of the bounding boxes over different time-steps.

Experiment 1: Measuring Tracking performance. The aim of this experiment is to evaluate our

Experimental Setting	NLL	MSE	Rcll	Prcn	MOTA	MOTP	Count MAE
DSPRITES (UHD)	117381	464.93	90%	99%	86%	1.16	0.11
DSPRITES (HD)	112331	83.49	98.7%	96%	91.6%	1.34	0.05
DSPRITES (MD)	99107	57.03	96%	97%	88.8%	1.22	0.05
DSPRITES (LD)	102076	20.46	99%	96.1%	91.7%	1.34	0.05
MNIST (MD)	97124	4.24	90.7%	90.5%	86.1%	1.21	0.09
MNIST (LD)	95415	-	-	-	-	-	0.22

Table 1: Quantitative results of our model tested on different experimental settings.

model's tracking ability for simple moving objects in a simple environment where no background is present. We experiment on 4 different settings, specified by the number of objects contained in that experiment. Each setting is represented in a triplet format of (Min, Avg, Max) where Min corresponds to the minimum, Avg to the average and Max to the maximum number of possible objects in that experiment. Min and Max are in fact a lower and higher bound, respectively, on the number of objects from the observer's viewpoint. The 4 mentioned settings are called Low Density (LD) [(10,9.2,11)], Medium Density (MD) [(24,21.9,27)], High Density (HD) [(50,64,54.5)] and Ultra High Density (UHD) [(100,120,99)].

Different performance measures including Negative Log Likelihood (NLL), Evidence Lower Bound (ELBO), test MSE as well as Normalized count Mean Absolute Error are evaluated for these different settings. Furthermore, standard object tracking metrics such as Multi Object Tracking Accuracy (MOTA), Multi Object Tracking Precision (MOTP) as well as Precision and Recall of the inferred bounding boxes are also computed Bernardin & Stiefelhagen (2008).

As can be seen in Table 1, by increasing the number of objects, our model's tracking accuracy only drops moderately, which shows the superior power of our model in images with a high number of objects.

Figures 1 and 2 qualitatively demonstrate the performance of our proposed model on the DSprites (HD) and MNIST (MD) settings, respectively. As for the moving MNIST dataset, we did not explore the high density and ultra high density scenarios, as the resolution of MNIST digits became too small, thus making digits unidentifiable for that setting. As can be seen, a large number of bounding boxes can be inferred in each time-step due to the power of the discovery module. Furthermore it is interesting to note that SCALOR is much faster compared to RNN-based models such as SQAIR, since it does not need to iterate over all the objects. In crowded videos, this can greatly speed up inference and training, as it reduces the time from $\mathcal{O}(N)$ to $\mathcal{O}(1)$, due to the parallel computation across different objects.

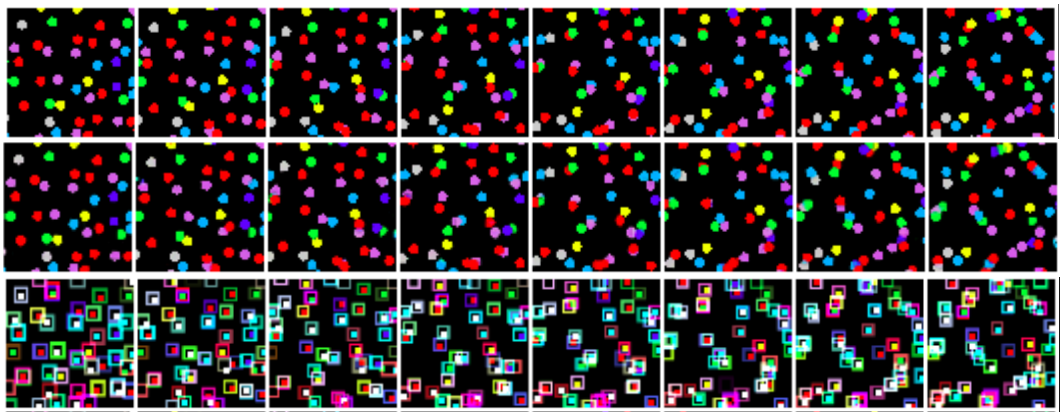

Figure 1: Qualitative samples from Moving DSprites (HD) setting: a) Original image sequence b) Reconstruction of the image by SCALOR c) Inferred bounding boxes

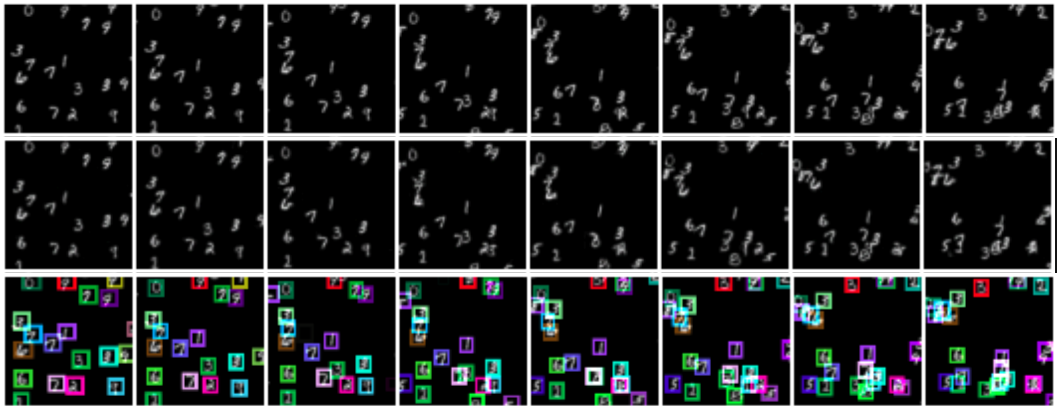

Figure 2: Qualitative samples from Moving MNIST (MD) setting: a) Original image sequence b) Reconstruction of the image by SCALOR c) Inferred bounding boxes

Experiment 2: Frequent Dense Discovery. One key challenge in modern video processing is that videos require substantial memory, inefficiently necessitating the use of advanced video compression. One such method is to represent videos with only the key-frames, which are the frames at which significant changes happen in a scene. In the object tracking domain, one such example of a key-frame is where many objects get introduced in the same frame due to a sudden change of the observer's view point or because of the lack of availability of the previous frames as a result of compression.

Our model can be an ideal choice even for such challenging situations where there is a need to discover many objects in a single frame. The powerful convolutional encoders of the discovery module enable us to discover many objects at each single frame. This directs us towards another application of SCALOR, since with the availability of such robust models, compression algorithms can work even more aggressively on videos.

Fig. 6 in the Appendix represents one such instance where 10-15 objects are introduced at the first, fourth, and seventh time-step, respectively, although it is still possible to have objects move in and out of the scene at other time-steps as well.

Experiment 3: Complex Background Separation. Another interesting yet challenging aspect of natural video processing is the presence of complex moving backgrounds. SCALOR is able to successfully separate such complex moving natural scenes, since the background inference network is explicitly designed to handle complex backgrounds. Fig. 3 provides one such instance, where the model is provided with a dynamic environment consisting of moving objects on top of a complex background. Fig. 3(c) shows the qualitative accuracy of the inferred bounding boxes, while Fig. 3(d) represents the inferred complex background. As can be seen, SCALOR is successfully able to disentangle foreground objects from a complex non-constant background and track them accurately. Interestingly, the fact that the color of some objects are similar to regions of the background does not limit its ability to detect and track them.

Experiment 4: Generalization ability. We conduct 3 sets of experiments to evaluate our model's ability to generalize to settings unseen at training time. In the first experiment, generalization to longer sequences is shown, where the model is trained on a dataset having 8 time-steps while being tested on 16 time-steps. In the second experiment, the model's ability to generalize to more crowded scenes is evaluated, where we train on scenes containing 10–15 objects while testing on 50–60 objects. This can be very beneficial in a real-world setting, as sometimes the dataset is only available for less crowded instances of an environment, e.g., less crowded times of the day in a subway station, while it should be able to be invoked in other more dense environments as well. The third experiment is intended to assess the model's ability to generalize to unseen objects. The model is trained on MNIST digits from 0 to 5, while tested on images containing only 6 to 9. Samples from these experiments are provided in the appendix.

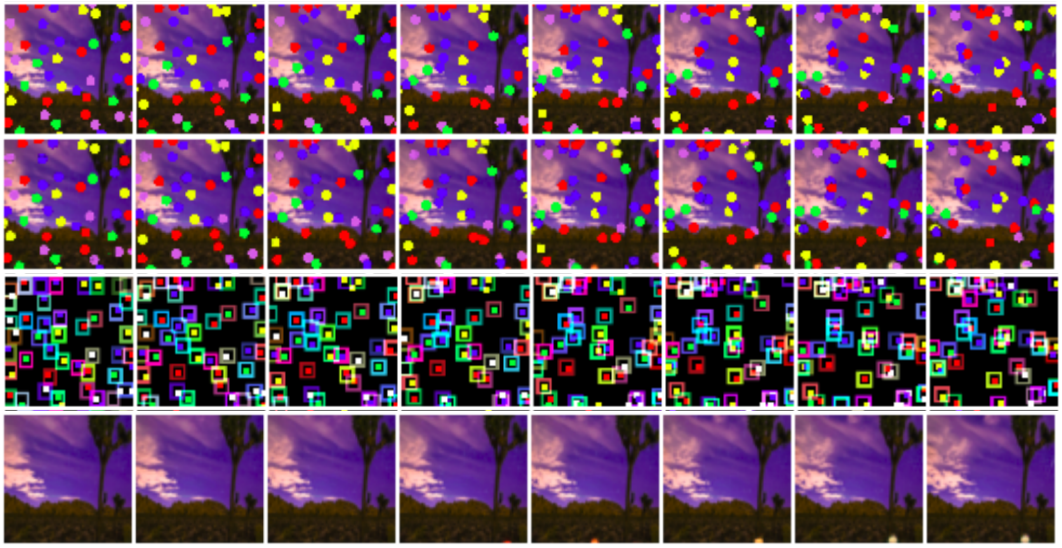

Figure 3: Complex Background Separation: a) The first row represents the original image. b) The second row shows the model's reconstruction. c) The third row provides the identified bounding boxes. d) The fourth row provides the inferred background.

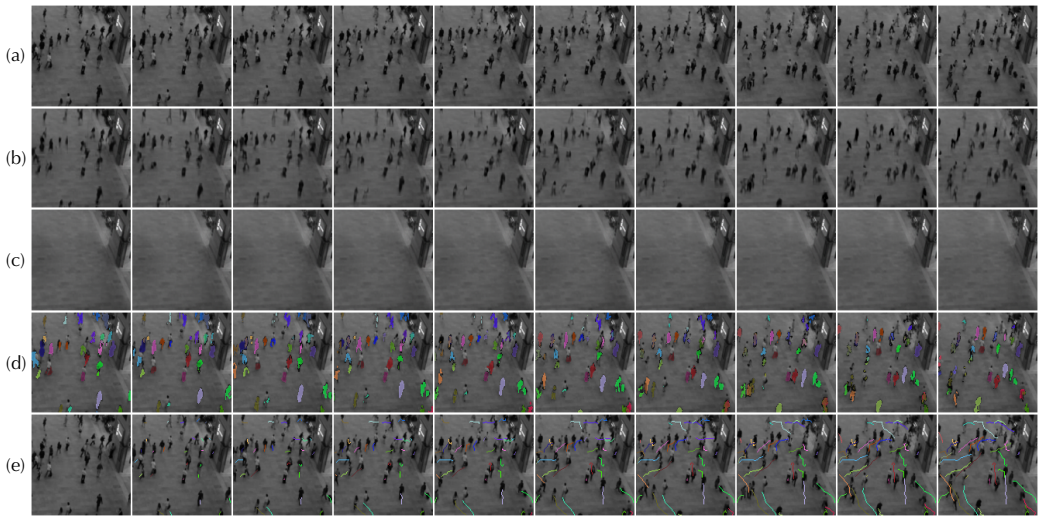

Figure 4: Qualitative result of SCALOR on Grand Central Station Dataset. (a) input sequence, (b) overall reconstruction of objects and background, (c) reconstruction of the extracted background, (d) segmentation for each object, colors indicate tracking ID, (e) extracted object trajectories.

5.2 Task 2: Real-World Dataset Tracking

This section showcases SCALOR's capabilities when it is provided 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 present in such scenarios. SCALOR is evaluated on the Crowded Grand Central Station Dataset (Zhou et al., 2012). The dataset is collected from the CCTV data of New York City's Grand Central Station. Due to the complexity of the pedestrian behavior, the dataset can be considered a mixture of both low density and high density objects settings. 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. The length of the input sequence is 10 and each image is resized to 128×128. We will make all datasets available upon publication.

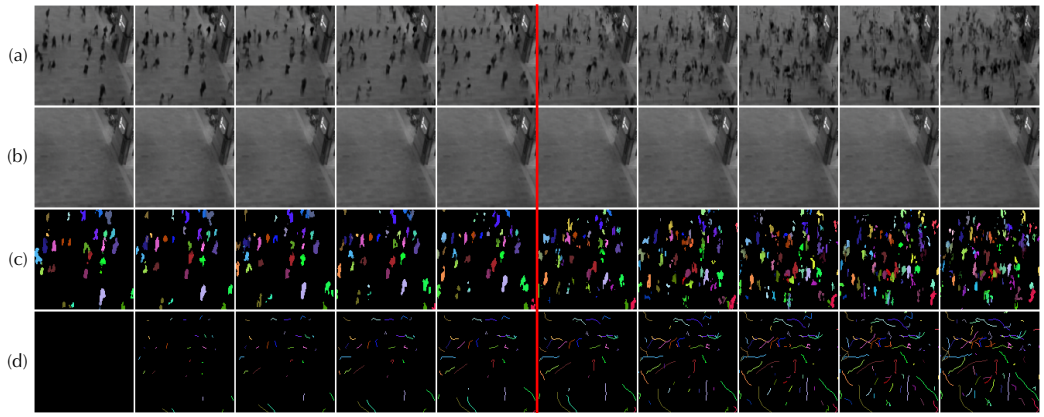

Figure 5: Conditional generation on Grand Central Station Dataset. The first 5 frames are observed, the last 5 frames are generated (after the red line). (a) overall reconstruction and generation, (b) conditional generation of background, (c) conditional generation of segmentation without background,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

Fig. 4 shows the qualitative results of the model. SCALOR performs reasonably well on this pedestrian tracking dataset, maintaining consistent temporal trajectories. As shown in Fig. 4(c), the background module infers the background composition and reconstructs the extracted background correctly. As for the object detection, SCALOR succeeds in accurate pedestrian detection and tracking. It could also output the foreground mask produced by z^{what}, which can represent the segmentation of the shape of individual pedestrians, as shown in Fig. 4(d). We draw extracted trajectories in Fig. 4(e) for each pedestrian in the natural scene. Trajectories in different colors correspond to different pedestrians. The color is given by the identity of the inferred latent variable of each object.

Fig. 5 shows qualitative results on conditional generation. The last 5 frames are generations conditioned on the first 5 observations. Starting from the 6th frame, the latent transition of the propagation trackers are modeled by the sequential prior network. The prior is also used in the discovery module for introducing new objects emerging in the scene. As we can see in the figure, the model tends to generate the same direction for the trajectories that are 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 Fig. 5(f), objects generated in the discovery phase usually tend to choose a single direction, and then follow that direction for the rest of the time frames. Although the trajectories' movements are generated consistently, the appearance of some objects in the generation phase sometimes varies across different time frames. Although the model's generation is reasonable, in some cases it is not ideal as the for some of the objects, the segmentation mask sometimes splits into multiple parts without any connection between them. This may stem from errors during the inference phase.

Since the ground truth trajectories of the Grand Central Station Dataset are no longer available at the time the experiment is conducted, we instead compare the negative log-likelihood (NLL) of our model with a baseline VAE as the quantitative result. Here we choose a VAE baseline that has one latent variable z of dimensionality 64, and with a sequential prior $p(z_t \mid z_{t-1})$. It is similar to our background module with the same number of latent variables for encoder and decoder. The NLL value for our model is 28.30, and for the VAE baseline it is 27.59. The computation is per pixel for the whole sequence. As we can see, our model has a very similar negative log-likelihood with the baseline VAE on natural image scenes, but is capable of doing object-wise representation learning.

6 Conclusion

This paper introduces SCALOR (SCALable Sequential Object-Oriented Representation), an object-centric method for tracking trajectories in videos. Our method requires no supervision from data and is entirely self-supervised, which can be immensely beneficial, as many computer vision applications lack sufficient training data. SCALOR not only infers object-wise trajectories and segmentation masks, but also models highly complex backgrounds present in the scene. The model's

performance and robustness is empirically assessed on real-world data settings as well as controlled environments, achieving accurate results on scenes containing 50–60 objects and also reasonable results for scenes containing up to a 100 objects. Furthermore, the model's ability to generate consistent trajectories as well for future timesteps is also assessed. The model is not only more robust and scalable compared to the state-of-art iterative inference methods but also much faster in terms of computation, since it infers object states in parallel. Although the model is reasonably robust, it still has some limitations, e.g., the segmentation mask may sometimes partially overlap with the actual object, occasionally missing certain body parts such as hands. In the future, additional ablation studies could be performed to identify techniques to further improve this aspect. Moreover, although we have shown complex backgrounds, additional improvements could be made to fully scale to arbitrary natural backgrounds, especially for noisy datasets. Finally, the learned latent variables can be further examined to see their application in other downstream tasks. The authors hope this opens a new avenue of research into unsupervised object-oriented tracking and representataion learning.

REFERENCES

Anonymous. Our other 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.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of AAAI*, 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, 2016a.

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

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.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pp. 2017–2025, 2015.

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, 2018a.

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, 2018b.

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

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.

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.

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.

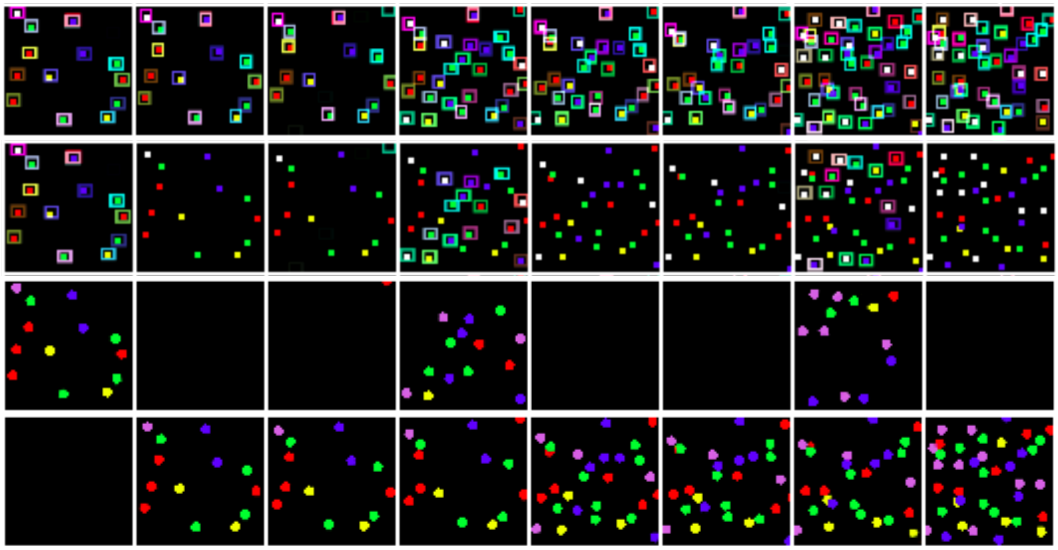

Figure 6: Frequent Dense Discovery: a)) All the Bounding boxes obtained from SCALOR superimposed on the original image. b) Bounding boxes only for the discovered objects. c) Reconstruction of the discovery step. d) Reconstruction of the Propagation step.

A Appendix: Additional Experimental Results

A.1 Frequent Dense Discovery

Images for Experiment 2: Frequent Dense Discovery are provided in Fig. 6. 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 frames. 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 z_what instead of discovering a new object.

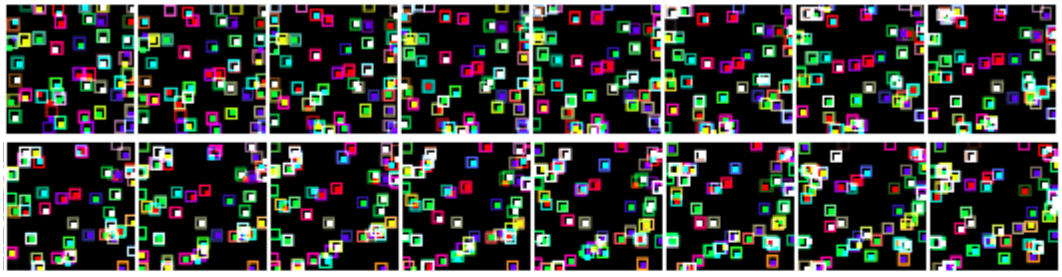

Figure 7: Generalization with respect to longer sequences. Top row represents the first 8 time steps, the bottom row the next 8 time steps.

A.2 Generalization to unseen data

Results for the generalization experiment are provided in this section. Fig. 7 shows the ability of SCALOR to scale to longer time steps. While it has been shown only 8 time steps during training, it is tested on sequences of length 16 at test time. The first row shows the bounding boxes on the first 8 frames, while the second row shows the next 8 frames. The model is able to still maintain the ids of the objects over the longer sequences as well and also discover the newly introduced objects. Fig. 7 assesses the model's ability to generalize well to unseen objects or more crowded scenes. The top row showcases the model's performance trained on digits of shape 1–5 but tested on 6–9. The

bottom row represents the case of it being trained on scenes containing 10-15 objects, but testing on scenes containing 40–60 objects.

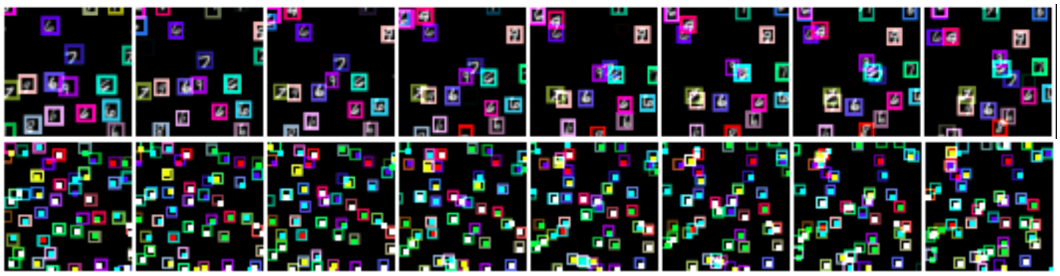

Figure 8: Generalization Experiment: a) Top image is the sample to demonstrate generalization with respect to other shapes. b) The middle image is the sample to demonstrate generalization with respect to more objects.

A.3 Ultra High Density

Fig. 9 represents some samples from the Ultra high density experiment. This experiment places 100–120 objects in the overall scene, on average 99 of which are visible at every time step. However, due to memory efficiency, the model only contains 64 cells for discovery at each time step. Interestingly, it can be seen that the model tries its best to identify as many objects as possible in the first time step, however, due to the lack of enough cells, it will discover whatever object remains in the second time step. Furthermore, in the case of too many objects being 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.

A.4 Some Interesting cases

In Fig. 10, multiple instances can be seen in which, although some objects might be severely occluded by one or multiple other objects, their identity is still preserved and identity swaps do not occur. Fig. 11 showcases other such cases on MNIST, highlighting our model's robustness.

A.5 Quantitative results from Grand Central Station Dataset

We provide quantitative results in Figures 12 to 16 and Figures 17 to 21.

B Model Architecture Details

Finally, 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 extracted convolutional features are used for both the propagation module to update the tracker hidden state, and for the discovery model to propose 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 fixed learning rate of 4e-5 and use RMSProp optimization during training. The variance of the image distribution is chosen to be 0.1. The prior for all Gaussian posteriors are set to standard normal. We constrain the range of z^{scale} to be 5.2 to 11.7 for the width, and 12.0 to 28.8 for the height with a prior at the middle value in discovery. The prior for the z^{pres} in discovery is set to be 0.1 at the beginning of training and to quickly anneal to 1e-3. The temperature used for modelling z_{pres} is set to be 1.0 at the beginning and anneal to 0.5 after 20k iterations.

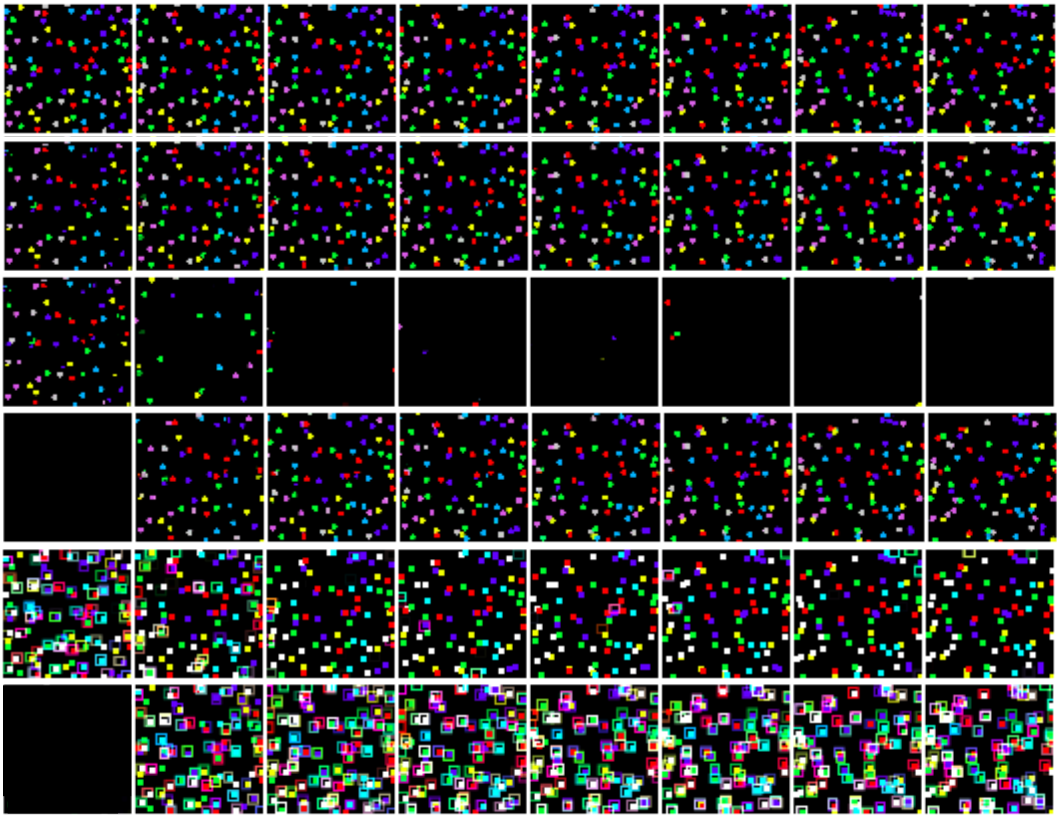

Figure 9: Ultra High Density setting: a) Original Image, b) Overall reconstruction, c) Discovery Reconstruction, c) Propagation Reconstruction, d) Bounding boxes obtained from Discovery phase, e) Bounding boxes of propagation phase.

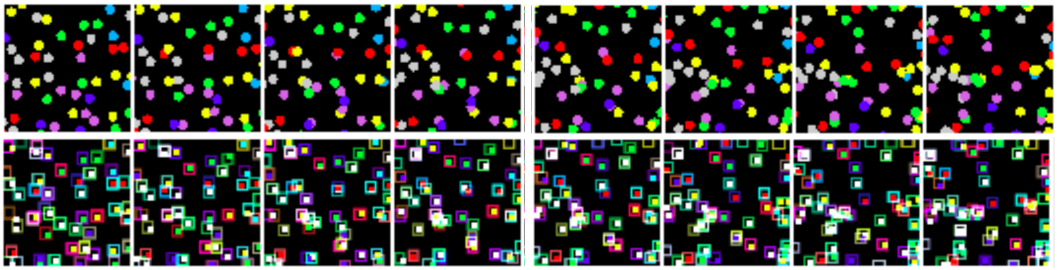

Figure 10: The top row shows a sequence where several objects overlap at a certain point; the bottom row demonstrates the inferred bounding boxes.

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

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

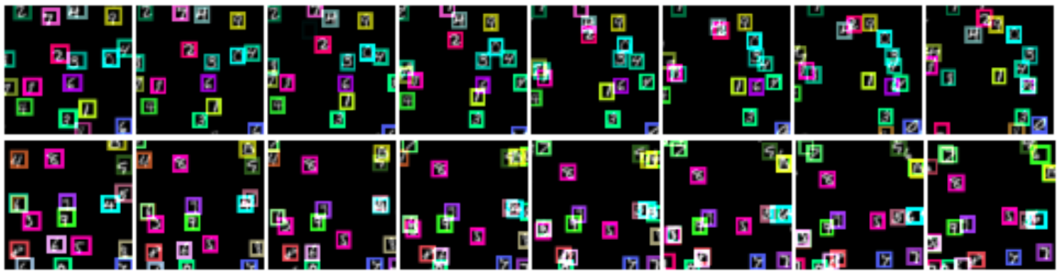

Figure 11: Highly occlusion cases on Crowded Moving MNIST dataset.

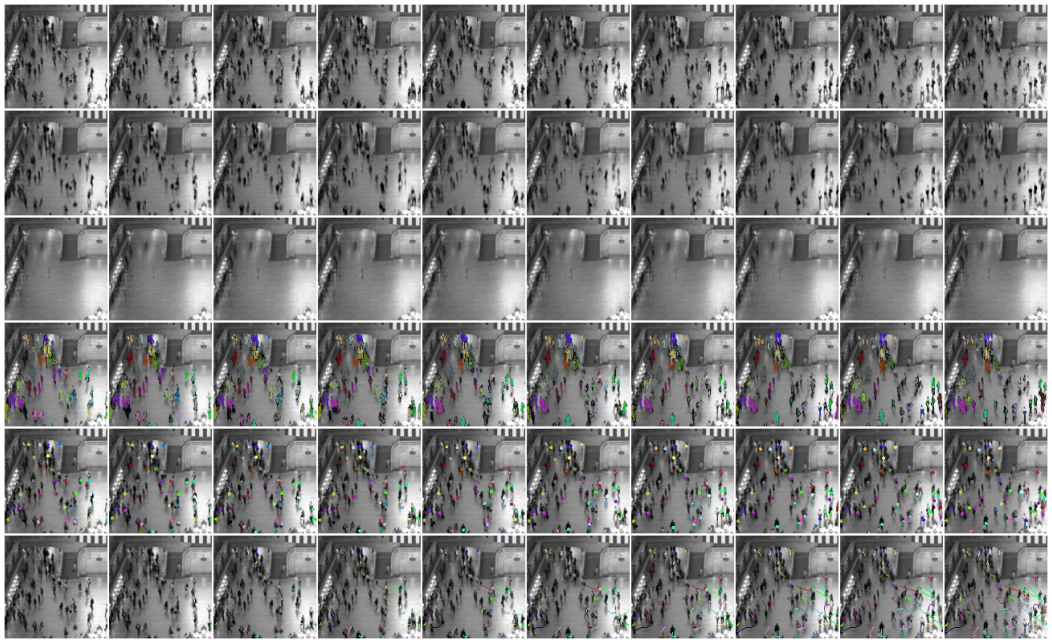

Figure 12: Detection example 1.

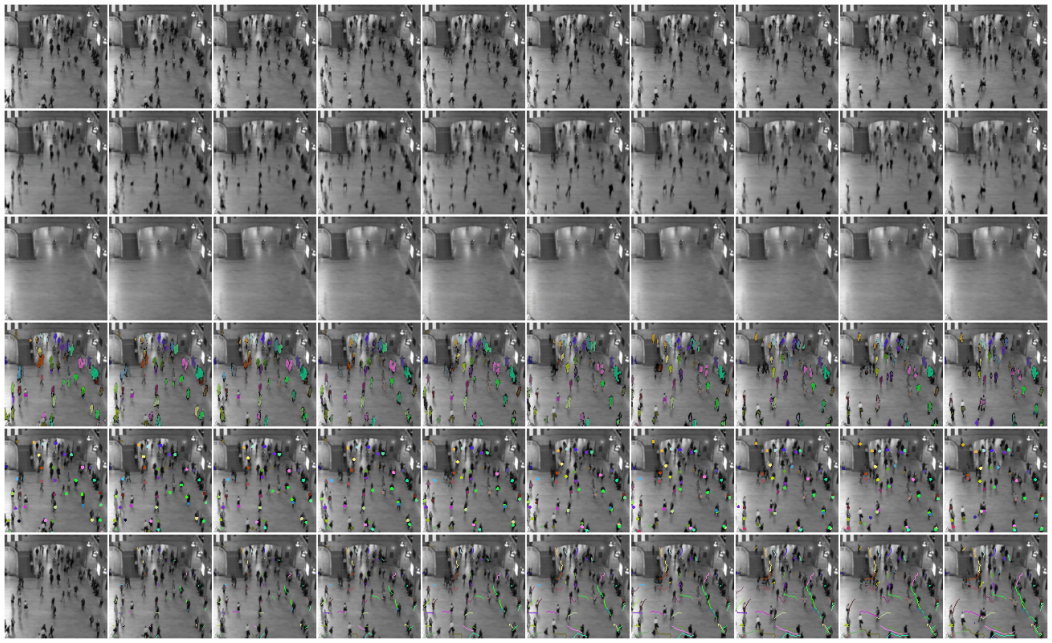

Figure 13: Detection example 2.

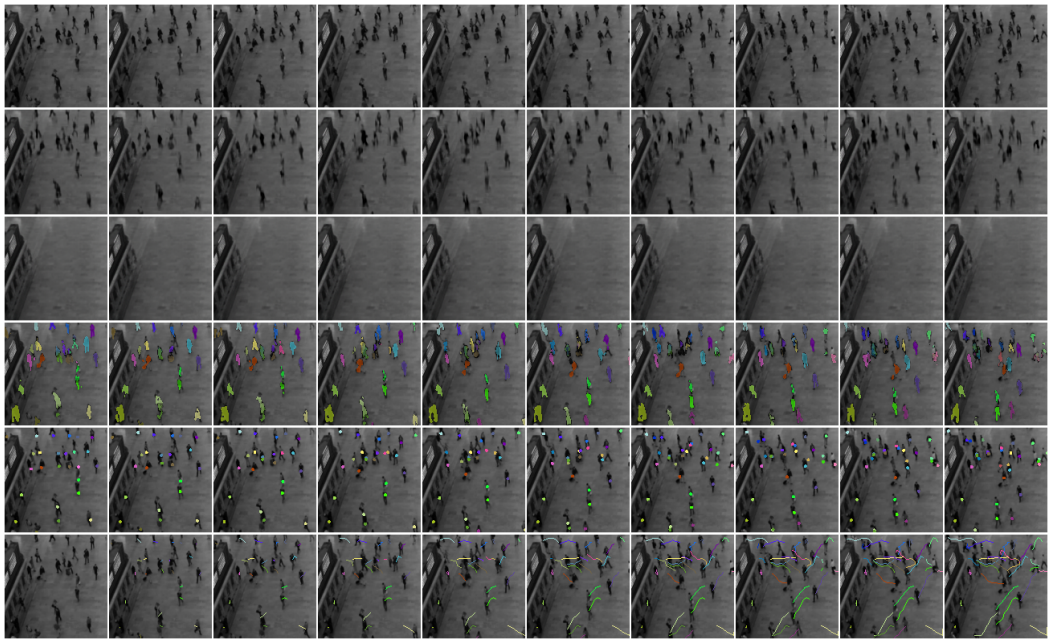

Figure 14: Detection example 3.

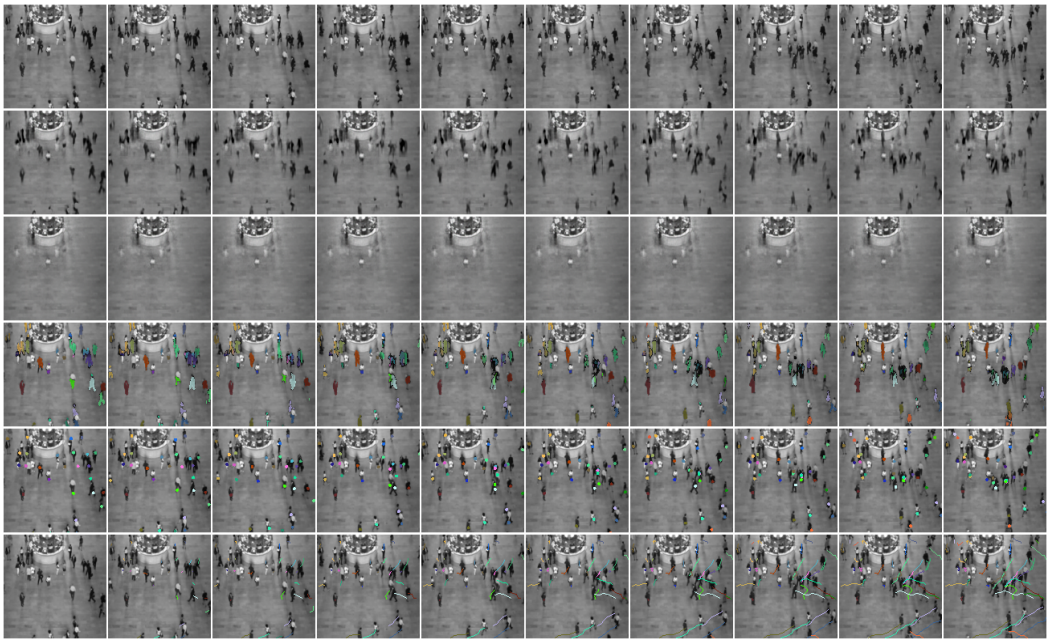

Figure 15: Detection example 4.

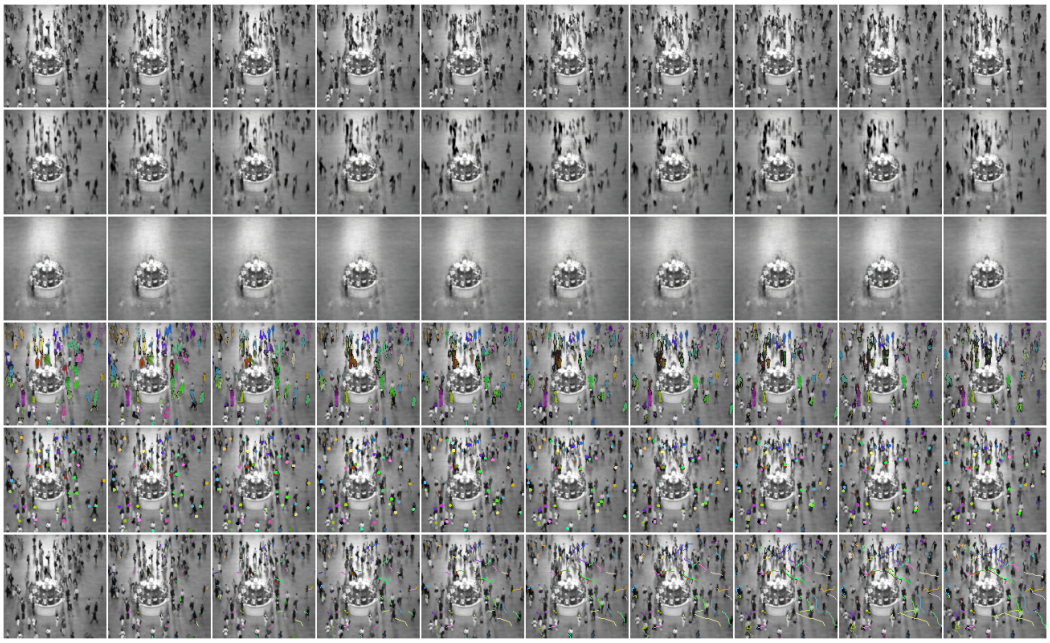

Figure 16: Detection example 5.

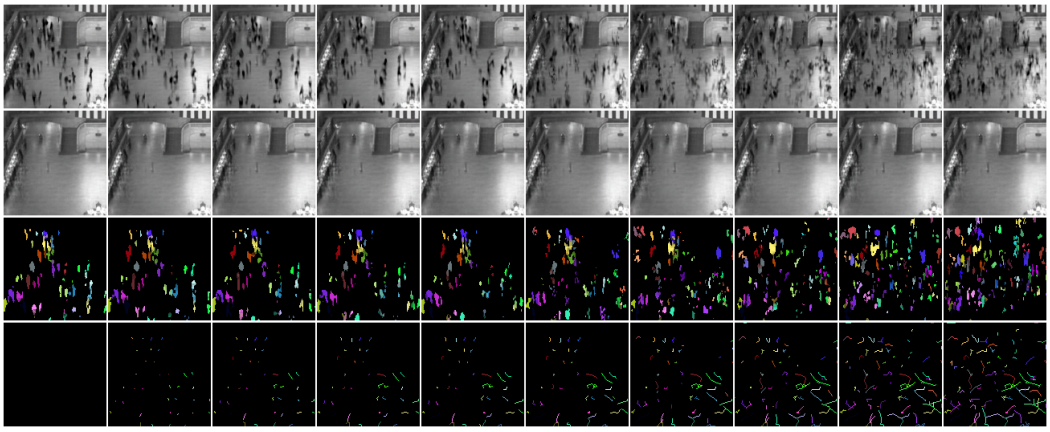

Figure 17: Generation example 1.

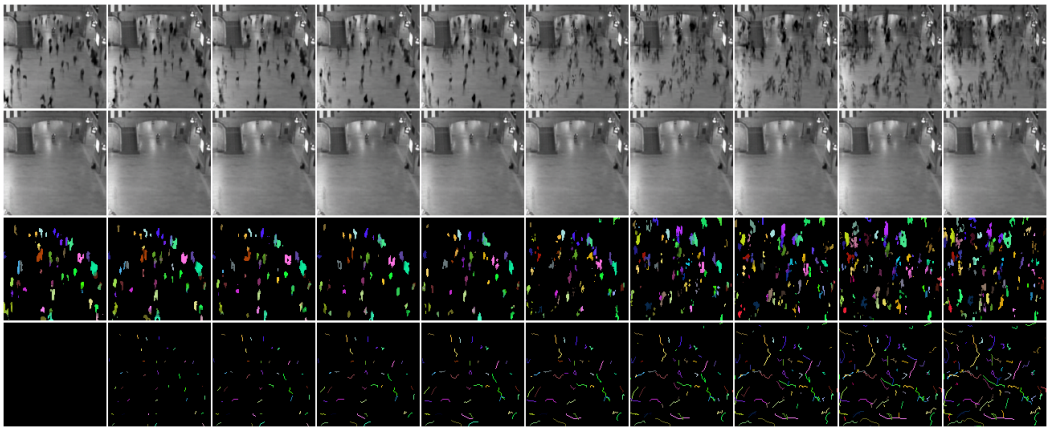

Figure 18: Generation example 2.

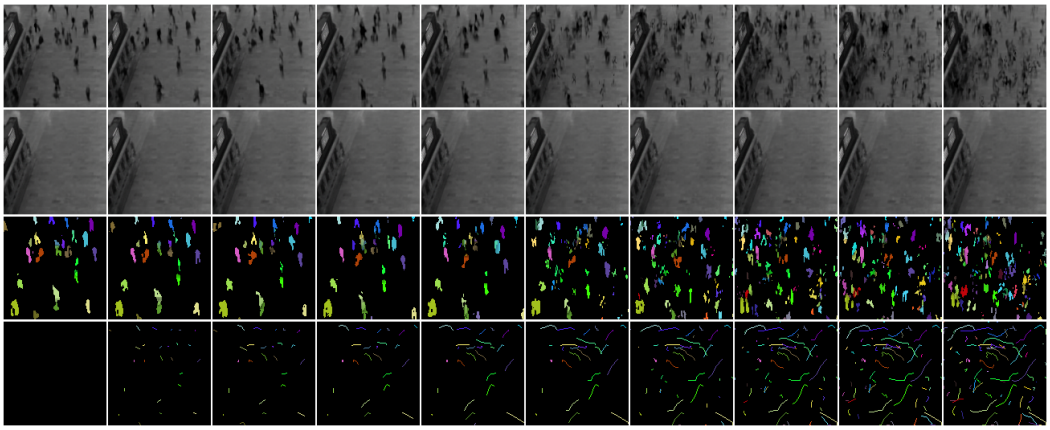

Figure 19: Generation example 3.

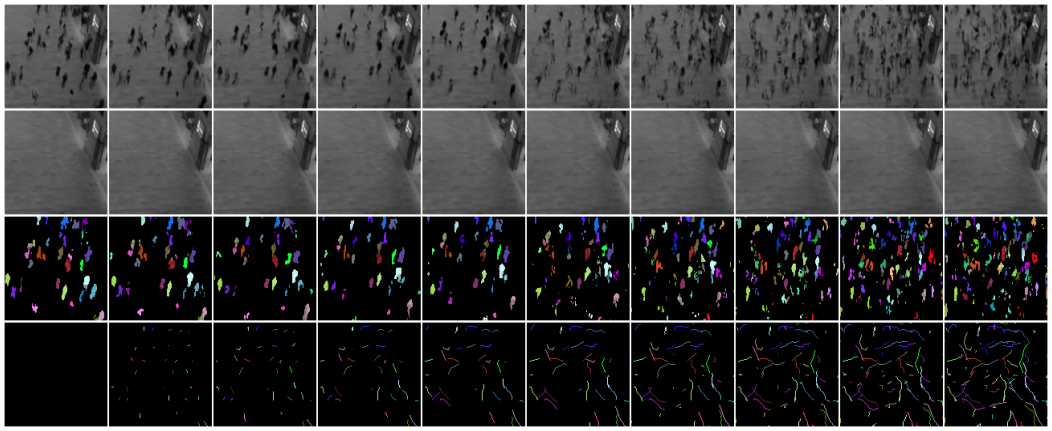

Figure 20: Generation example 4.

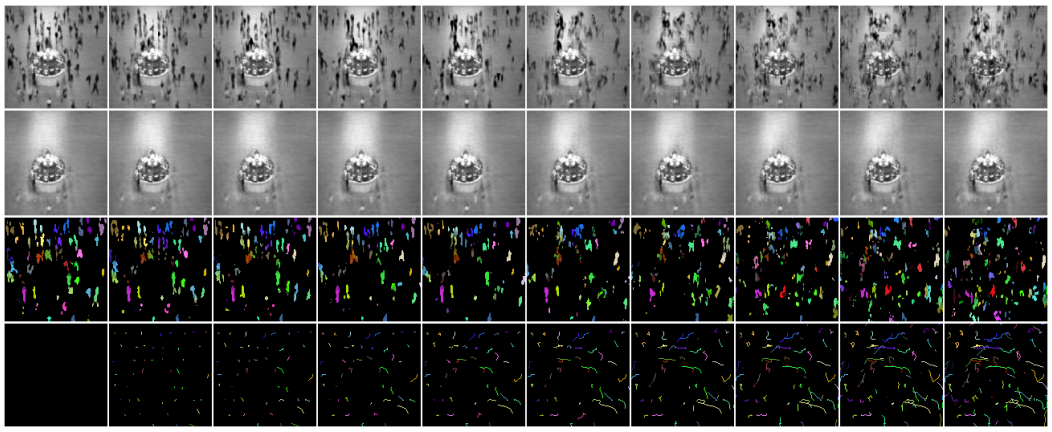

Figure 21: Generation example 5.

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

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	