A. Variational Lower Bound Derivation

The original variational lower bound was derived in [4].

$$
\begin{aligned}
&\log p_\theta(\mathbf{x}) = \log \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \\
&= \log \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right) \\
&= \sum_t \left[\mathbb{E}_{q_\phi(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} \log p_\theta\left(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t}\right)\right. \\
&\qquad \left. - D_{KL}\left(q_\phi\left(\mathbf{z}_t \mid \mathbf{x}_{1:t}\right) \| p\left(\mathbf{z}_t\right)\right)\right]
\end{aligned}
$$

The final step [1] is obtained through the factorization of reconstruction and KL-Divergence term into individual time steps due to the independence across time.

B. Invertible Architecture and Coupling Layer

The additive coupling layer was first introduced in [2]. Following [5], we use it as the building block to construct the invertible autoencoder. More specifically, the reshaped input x is divided into two groups, denoted as x^1 and x^2, channel-wisely. In its forward pass, one group, e.g. x^1, passes through several convolutional layers and updates the other group, x^2, through addition.

$$\hat{x}^2 = x^2 + \mathcal{F}_1(x^1) \tag{1}$$

$$\hat{x}^1 = x^1 + \mathcal{F}_2(\hat{x}^2) \tag{2}$$

where $\mathcal{F}$ is a composite non-linear transformation consisting of convolutions and activations, and $\hat{x}^1$ and $\hat{x}^2$ are the updated x^1 and x^2. In its backward pass, we can retrieve x^1 and x^2 from $\hat{x}^2$ and $\hat{x}^1$ by the following inverse computation:

$$x^1 = \hat{x}^1 - \mathcal{F}_2(\hat{x}^2) \tag{3}$$

$$x^2 = \hat{x}^2 - \mathcal{F}_1(x^1) \tag{4}$$

Pixel shuffle layer, a bijective downsampling, is also employed to change the shape of feature from (w, h, c) to $(w/n, h/n, c \times n^2)$ to enable the invertibility of the entire network. Stacking these building blocks and downsampling in an alternating fashion between two groups, we will obtain a two-way autoencoder. The property of invertibility ensures no information loss during feature extraction, which is better at preserving the attributes of moving objects. The same network can serve as both the encoder and the decoder by using its forward and backward pass respectively.

C. Training Setup

In the deterministic setting, MAC adopts DCGAN and a mirrored network as encoder and decoder and 2 layers of residual ConvLSTM as predictor. In the stochastic setting, sMAC replaces its encoder with 24-layer invertible autoencoder and use its backward pass as decoder. Additionally, it also deploys two inference networks composed of 2 layers of ConvLSTM, named *prior* and *posterior*, to model conditionally Gaussian distribution of trajectories .

We use the Adam optimizer [3] with a starting learning rate of 2×10^{-4} to optimize the MAC and sMAC. The training process is stopped after $200,000$ iterations with the batch size of 4. 20,000 video clips of CLEVR-Building-Blocks and 30,000 of Sapien-Kitchen are generated for model training and additionally 5,000 videos are generated for each dataset for evaluation. Considering the size of Tower-Creation dataset, various traditional data augmentation methods are used and we also implement a new trick in which the neighbouring frames of key frames are sampled from Gaussian distributions to serve as small temporal variations. This trick can significantly improve the visual quality and diversity of stochastic generation for both sMAC and SVG-LP.

D. Object Detection

The quantitative results and visualization of object detection is provided in the Table 1 and Fig 1. SSD head was optimized following its protocol while the MAC encoder was frozen to demonstrate that features learnt through self-supervision can be directly transferred for detection because our video prediction task is highly location-dependent.

References

[1] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018. 1

[2] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 1

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[5] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2019. 1

Method	Oven	Fridge	Dishwasher	Bottle	Kettle	Kitchen pot	mAP
MAC + SSD	92.75	94.56	90.89	83.25	77.18	81.32	86.66

Table 1. Quantitative measures of object detection on Sapien-Kitchen in terms of average precision.

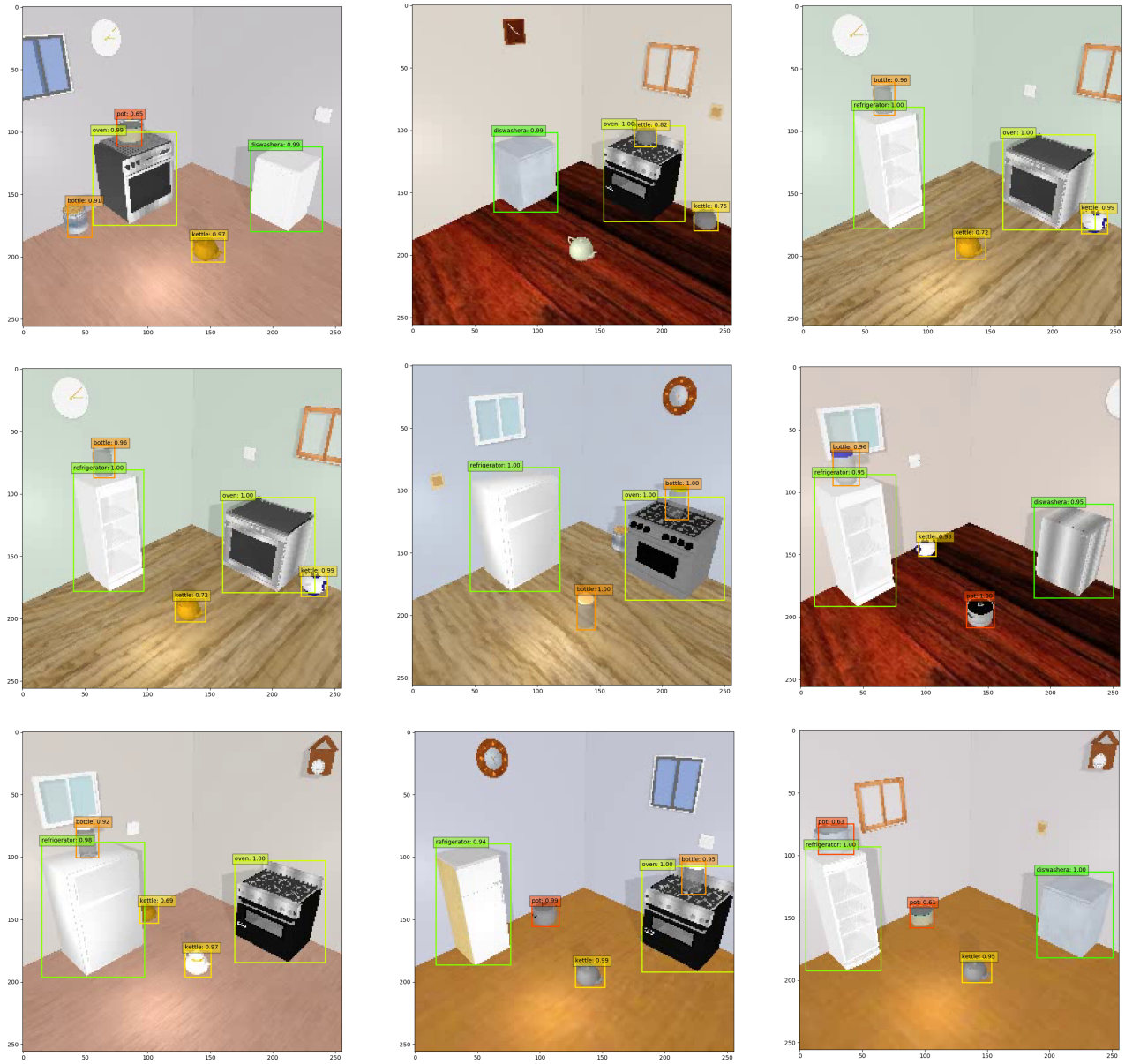

Figure 1. Visualization of 2D Object Detection on Sapien-Kitchen.