Modular Action Concept Grounding in Semantic Video Prediction

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent works in video prediction have mainly focused on passive forecasting and low-level action-conditional prediction, which sidesteps the learning of interaction between agents and objects. We introduce the task of semantic action-conditional video prediction, which uses semantic action labels to describe those interactions and can be regarded as an inverse problem of action recognition. The challenge of this new task primarily lies in how to effectively inform the model of semantic action information. Inspired by the idea of Mixture of Experts, we embody each abstract label by a structured combination of various visual concept learners and propose a novel video prediction model, **M**odular **A**ction **C**oncept Network (MAC). Our method is evaluated on two newly designed synthetic datasets, CLEVR-Building-Blocks and Sapien-Kitchen, and one real-world dataset called Tower-Creation. Extensive experiments demonstrate that MAC can correctly condition on given instructions and generate corresponding future frames without need of bounding boxes. We further show that the trained model can make out-of-distribution generalization, be quickly adapted to new object categories and exploit its learnt features for object detection, showing the progression towards higher-level cognitive abilities. More visualizations can be found at `https://iclr-mac.github.io/MAC/`.

1 INTRODUCTION

Recently, video prediction has drawn a lot of attention due to its ability to capture meaningful representations through self-supervision (Wang et al. (2018b); Yu et al. (2019)). Although modern video prediction methods have made significant progress in improving predictive accuracy, most of their applications are limited in the scenarios of passive forecasting (Villegas et al. (2017); Wang et al. (2018a); Byeon et al. (2018); Jin et al. (2020)), meaning models can only passively observe a short period of dynamics and accordingly make a short-term extrapolation. Such settings neglect the fact that the observer can also become an active participant in the environment.

To model the movements of active manipulators, several low-level action-conditional video prediction models have been proposed in the community (Oh et al. (2015); Mathieu et al. (2015); Babaeizadeh et al. (2017); Ebert et al. (2017)). In this work, we go one step further by introducing the task of *semantic action-conditional video prediction* which emphasizes the modeling of interactions between agents and environment. Instead of using low-level single-entity actions such as action vectors of robot arms as done in prior works (Finn et al. (2016); Kurutach et al. (2018)), our new task provides semantic descriptions of interactive actions, e.g. *"Open the door"*, and asks the model to imagine *"What if I open the door"* in the form of future frames. This task requires the model to recognize the object identity, assign correct affordances to objects and envision the long-term expectation by planning a reasonable trajectory toward the goal, which resembles how humans might imagine conditional futures. The ability to predict correct and semantically consistent future perceptual information is indicative of conceptual grounding of actions, in a manner similar to object grounding in image-based detection and generation tasks.

The challenge of action-conditional video prediction primarily lies in how to correctly inform the model of more abstract semantic action information. Existing low-level counterparts usually achieve this by employing a naive concatenation (Finn et al. (2016); Babaeizadeh et al. (2017)) with action vector of each timestep. While this implementation might enable model to move the desired objects, it fails to produce consistent long-term predictions toward target locations in the multi-entity settings
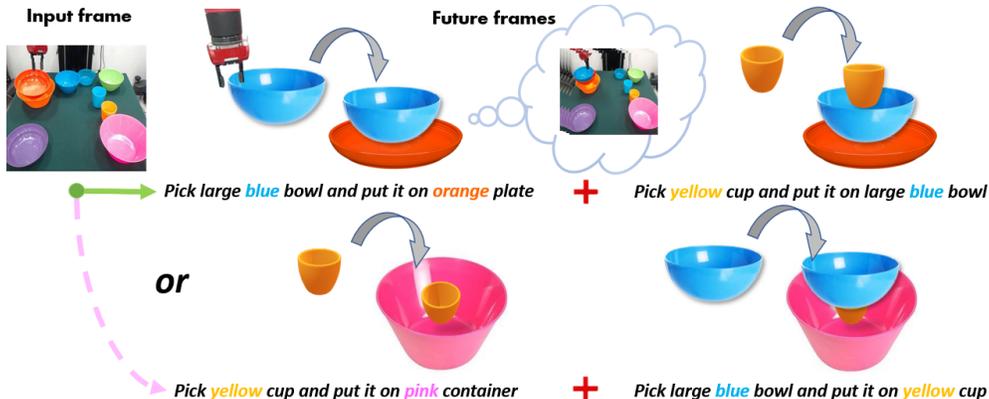
Figure 1: **Concept Grounding in Semantic Video Prediction**. After observing the scene, an agent predicts future frames conditioned on a series of semantic actions describing agent-object interactions. Neither bounding boxes nor key points are provided. Conditioning on different action labels leads to *Counterfactual* generations.

because it was originally designed to only encode the motion information of a single entity. If we take "*put A on B*" as an example, it turns out to be difficult to make the model learn what and where B is, because the main self-supervisory signals in the framework of video prediction are pixel changes and B is not moving in this case. In order to distinguish and locate instances in the scene, other related works heavily rely on pre-trained object detectors or ground-truth bounding boxes (Bar et al. (2020); Ji et al. (2020); Huang et al. (2018); Wu et al. (2020)). However, we argue that utilizing a pre-trained detector actually simplifies the task since such a detector already solves the major difficulty by mapping high-dimension inputs to low-dimension groundings. Furthermore, bounding boxes cannot effectively describe complex visual changes including rotations and occlusions. Thus, a more flexible way of representing objects and actions is required.

We present a new video prediction model, MAC, short for Modular Action Concept Network. Inspired by the idea of Mixture of Experts, MAC embodies each semantic label by a structured combination of various concept slots, each of which encodes the spatial representation of a specific concept. Such design allows MAC to reuse and integrate the knowledge learnt from different scenarios so that it can perceive the locations of motionless objects and extrapolate to unseen cases, showing the progression towards higher-level cognitive abilities. The contributions of this work are summarized as follows:

1. We introduce a new task, semantic action-conditional video prediction as illustrated in Fig 1, which can be viewed as an inverse problem of action recognition.

2. We create two new synthetic video datasets, CLEVR-Building-blocks and Sapien-Kitchen, and label one real-world dataset called Tower-Creation for evaluation.

3. We propose a novel video prediction model, Modular Action Concept Network, in which routing of visual concept slots is directly controlled by action labels. We show that MAC can successfully depict the long-term counterfactual evolution without need of bounding boxes.

4. We demonstrate that the trained MAC can make out-of-distribution generalization, be adapted for new object categories with a small number of samples and exploit its learnt features for detection.

2 APPROACH

We begin with defining the task of semantic action-conditional video prediction. Given an initial frame x_0 and a sequence of action labels $a_{1:T}$, the model is required to predict the corresponding future frames $x_{1:T}$. Each action label is a pre-defined semantic description of a spatiotemporal movement that involves multiple objects in a scene and spans over multiple frames such as "*take the yellow cup on the table*" from $t = 0$ to $t = 10$. So technically, one can regard this task as an inverse problem of action recognition. It should also be pointed out that our semantic task is different from common *dense video prediction and generation tasks* in the sense that it focuses on predicting **time-agnostic** events. Hence, we design the corresponding datasets as videos capturing sufficient key frames of entire actions. In future practices, we can further apply video interpolation methods in CV or motion planner algorithms in RL to make up the intermediate process if needed.
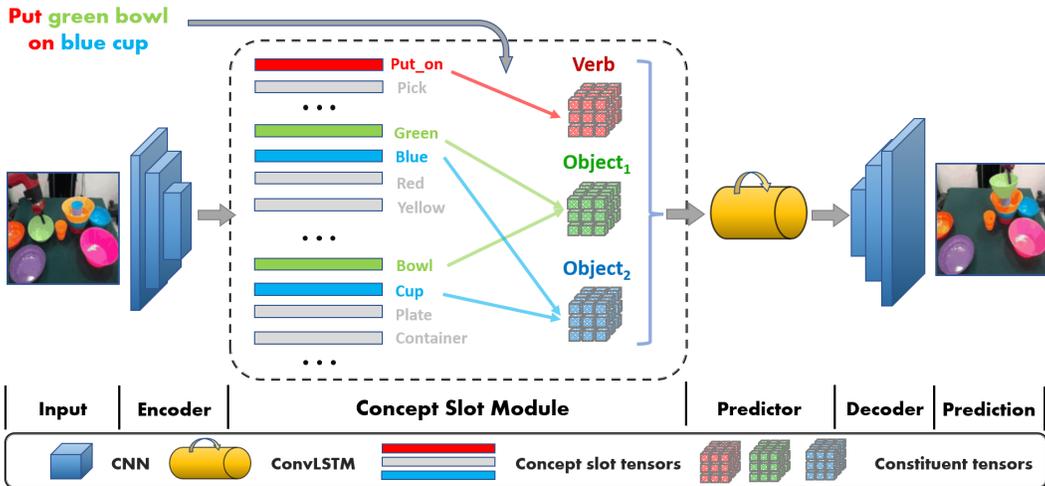
Figure 2: The pipeline of MAC in which the computation of concept slot module is elaborated (Better viewed in color). Feature maps extracted by encoder are mapped into the concept slot tensors. Concept slot module receives an action label that controls the collection of concept slot tensors and outputs representations encapsulating this action. A recurrent predictor updates representations before sending them to decoder to predict the next frame.

2.1 MOTIVATION

The design of our new task is necessary for studying compositional generalization as it detaches the definition of object from its specific location. However, it also requires a successful model to figure out where the desired object is through leveraging abstract labels. Our main idea is that we create a large number of small specialized learners called concept slots for each word in the dictionary of action labels to capture their corresponding spatial representations from observations. During training, action labels will be translated as constituency trees to control the activations of all related concept slots and to assemble the representations of given actions for next-frame prediction. As a result, this language-guided gating mechanism embeds the syntactic structures into the learning system and enables the proposed model to dynamically recombine its learnt concepts so that it can understand the combinatorial complexity of the world. In this paper, we demonstrate that our method possesses many key characteristics of *system-2* learning (Goyal et al. (2019); Goyal & Bengio (2020)), including concept grounding, sample efficiency, counterfactual generations, out-of-distribution generalization and fast transfer.

2.2 MODULAR ACTION CONCEPT NETWORK

The MAC model is composed of 4 modules including encoder $\mathcal{E}$, decoder $\mathcal{D}$, concept slot module $\mathcal{C}$ and recurrent predictor $\mathcal{P}$. The goal of our model is to learn the following mapping:

$$\hat{x}_t = \mathcal{D}(\mathcal{P}(\mathcal{C}(\mathcal{E}(x_{t-1})|a_t)|h_{t-1})) \tag{1}$$

where x_t, a_t and h_t are video frame, action labels and hidden states at time t. The overall architecture of our method is illustrated in Fig 2. In the case of stochastic video generation, another two modules, prior $p(z)$ and posterior $q(z)$, will be added to help estimate the latent distribution of trajectories.

Encoder and Decoder: At each timestep $t-1$, the encoder $\mathcal{E}$ receives visual input x_{t-1} and extracts a set of multi-scale feature maps. In the deterministic setting, we employ a convolutional neural network with an architecture similar to VGG16 (Simonyan & Zisserman (2014)). The matching decoder $\mathcal{D}$ is a mirrored version of the encoder with down-sampling operations replaced with spatial up-sampling and additional sigmoid output layer. It aggregates the updated latent representations produced by predictor and multi-scale feature maps from encoder to predict the next frame $\hat{x}_t$.

In the stochastic setting, we use invertible autoencoder introduced in CrevNet (Yu et al. (2019)) instead as we find this information-preserving architecture can better preserve the attributes of randomly moving objects. The corresponding decoder is the backward pass, i.e. inverse computation, of the same network of the encoder. Readers can find more details about invertible autoencoder and coupling layer in Appendix B.

Concept Slot Module: The concept slot module $\mathcal{C}$ is the core module of MAC. It resembles the mixture of experts as each slot focuses on only one concept in the space of action labels and will be activated and assembled to represent the given actions through the language-guided gating functions.

Each action label will first be decomposed into several constituents of sentence. A constituent is a verb or object phrase, like "*pick*" or "*large red bowl*". Since we are mostly dealing with manipulation videos, actions are usually divided into 3 constituents, verb, $object_1$, $object_2$. Each constituent will have its own dictionary recording all pre-defined words or concepts and gating functions can be derived based on these dictionaries to establish bottom-up connections from concept slots. The computation of concept slot module is given as follows:

$$\mathbf{w}^i = \Psi^i(\mathbf{f}), \qquad\qquad \mathbf{c}^j = \Phi^j(\text{Concat}(\{\mathbf{w}^i \,|\, \forall i, \; \delta^j(i) = 1\}))$$

where $\mathbf{w}$ and $\mathbf{c}$ are concept and constituent representations and δ^j is the indicator function for gating function of j_{th} constituent. More specifically, after the feature maps $\mathbf{f}$ are extracted from the input image, they are fed into $\mathcal{K}$ convolutional units Ψ^i, i.e. the concept slot layer, to create $\mathcal{K}$ concept slot tensors of dimension N_d. Here, $\mathcal{K}$ is the total number of possible concepts we pre-defined in the dictionary of action labels. Since verbs can be interpreted as spatiotemporal changes of relationships between objects, not only slots for objects but also slots for verbs, like '*take*' or '*put on*', are computed from the extracted feature maps.

Next, a gating function will collect all involved concept slot tensors and create an ensemble as input for each constituent. This assembly process simulates the formation of simplified constituency parse trees. Constituent slot layer Φ^j can either be resolution-preserving or upsampling operators as spatial information is important for our new task. Finally, outputs of all constituent slots are concatenated pixel-wisely to obtain the representation of actions before sending them to predictor. It is worth noticing that MAC is allowed to have multiple concurrent actions in a scene at inference time. In this case, we copy additional groups of trained constituent slots to represent other actions.

Learned Prior: We leverage a technique called *learned prior* from SVG (Denton & Fergus (2018)) to model the stochastic movements in videos. In particular, we build two additional recurrent inference networks, prior and posterior respectively, to capture the randomness of motions. During training, the posterior inference network $q(z)$ can access to the representations of target frames to estimate a true distribution of trajectory that we expect its prior counterpart $p(z)$ to mimic at test time. Codes of motions z_t estimated by posterior during training (or by prior during testing) will then be concatenated with latent representations before sent to predictor.

Predictor: The recurrent predictor $\mathcal{P}$, implemented as a stack of residual ConvLSTM layers (Shi et al. (2015)), calculates the spatiotemporal evolution for each action label respectively. The memory mechanism of ConvLSTM is essential for MAC to remember its previous actions and to recover the occluded objects. To prevent interference between concurrent actions, hidden states are not shared between actions. The outputs of predictor for all action labels are added point-wisely.

Training: In the deterministic setting, we train our model by minimizing the mean squared error the between the target frames and the predictions. In the stochastic setting, we optimize the following variational lower bound (ELBO) using re-parameterization trick (Kingma & Welling (2013)):

$$\mathcal{L}_{\theta,\phi,\psi}(x_{1:T}) = \sum_{t=1}^{T}[\mathbb{E}_{q_\phi(z_{1:t}|x_{1:t})}\log p_\theta(x_t|z_{1:t}, x_{1:t-1}) - \beta D_{KL}(q_\phi(z_t|x_{1:t})||p_\psi(z_t|x_{1:t-1})]$$

where p_θ is the future frame generator, z_t represents the latent codes of motion, $p_\psi(z_t|x_{1:t-1})$ is the prior distribution, $q_\phi(z_t|x_{1:t})$ is the posterior distribution and D_{KL} denotes the Kullback–Leibler (KL) divergence which forces the posterior to approximate the prior distribution. Since p_θ is modeled by conditional Gaussian, the likelihood term reduces to MSE measure between the ground truth frames and the predictions. The full derivation of ELBO is provided in the Appendix A.

At the inference phase, the model will use its previous predictions as visual inputs instead except for the first pass. Hence, a training strategy called scheduled sampling (Bengio et al. (2015)) is adopted to alleviate the discrepancy between training and inference.

3 DATASETS

In this study, we create two new synthetic datesets, CLEVR-Building-blocks and Sapien-Kitchen, and label one real-world dataset called Tower-Creation from Roboturk (Mandlekar et al. (2018))

for evaluation. This is because most existing video datasets either don't come with semantic action labels (Babaeizadeh et al. (2017)) or fail to provide necessary visual information in their first frames due to egomotions and occlusions (Hundt et al. (2018)). Although there are several candidate datasets like Penn Action (Zhang et al. (2013)), BAIR (Finn et al. (2016)) and KTH (Schuldt et al. (2004)) for multi-modal learning, they all adopt the same single-entity setting which actually indicates they can be solved by a much simpler model. To tackle the above issues, we design each video in our datatsets as a depiction of certain atomic action performed by an agent with objects which are observable in the starting frame. Furthermore, we add functions to generate bounding boxes of all objects for both synthetic datasets in order to train AG2Vid. It is worth noting that all three of these domains exhibit a key property named **combinatorial explosion**, resulting in factorial complexity growth in both spatial and temporal dimensions even with a small object set. For instance, a sequence with 6 (out of 32) objects and 6 actions can have 333,396,000 possibilities without considering any continuous factor. Hence, our model only sees a small fraction of these potential scenarios during training.

3.1 CLEVR-BUILDING-BLOCKS DATASET

CLEVR-Building-blocks dataset is built upon CLEVR environment (Johnson et al. (2017)). For each video, the data generator initializes the scene with 4 - 6 randomly positioned and visually different objects. There are totally 32 combinations of shapes, colors and materials of objects and at most one instance of each combination is allowed to appear in a video sequence. The agent can perform one of the following 8 actions on objects $\mathcal{O}_A$ and $\mathcal{O}_B$: *Pick $\mathcal{O}_A$*, *Pick and Rotate $\mathcal{O}_A$ transversely / longitudinally*, *Put $\mathcal{O}_A$ on $\mathcal{O}_B$*, *Put $\mathcal{O}_A$ on the left / right side of $\mathcal{O}_B$*, *Put $\mathcal{O}_A$ in the front of / behind $\mathcal{O}_B$*. Each training sample contains a video of three consecutive *Pick-* and *Put-* action pairs and a sequence of semantic action labels of every frame.

3.2 SAPIEN-KITCHEN DATASET

Sapien-Kitchen Dataset describes a more complicated environment in the sense that: (a). It contains deformable actions like *"open"* and *"close"*; (b). The structures of different objects in the same category are highly diverse; (c). Objects can be initialized with randomly assigned relative positions like *"along the wall"* and *"on the dishwasher"*. We collect totally 21 types of small movable objects in 3 categories, *bottle, kettle* and *kitchen pot*, and 19 types of large openable appliances in another 3 categories, *oven, refrigerator* and *dishwasher*, from Sapien engine (Xiang et al. (2020)). The agent can perform one of the following 6 atomic actions on small object $\mathcal{O}_s$ and large appliance $\mathcal{O}_l$: *Take $\mathcal{O}_s$ on $\mathcal{O}_l$, Take $\mathcal{O}_s$ in $\mathcal{O}_l$, Put $\mathcal{O}_s$ on $\mathcal{O}_l$, Put $\mathcal{O}_s$ in $\mathcal{O}_l$, Open $\mathcal{O}_l$* and *Close $\mathcal{O}_l$*. Composite action sequences are defined as follows: *"Take_on–Put_on"*, *"Take_on–Open–Put_in–Close"*, *"Open–Take_in–Close"*.

3.3 TOWER-CREATION DATASET

Each video in Tower-Creation Dataset depicts a robotic arm building a tower with flatware present on the table. We have labeled 524 videos in total since semantic descriptions are not provided and prodce 1867 samples consists of two actions: *Pick $\mathcal{O}_A$* and *Put $\mathcal{O}_A$ on $\mathcal{O}_B$*. We use 1536 video clips for training and 331 for evaluation. It should be pointed out that the size of Tower-Creation dataset is small compared with commonly used datasets such as BAIR (Finn et al. (2016)) which has 59k videos in total. Thus, our experiments can also tell whether evaluated methods are data efficient.

4 EXPERIMENTAL EVALUATION

4.1 ACTION-CONDITONAL VIDEO PREDICTION

Baselines and setup: We evaluate the proposed model on CLEVR-Building-blocks and Sapien-Kitchen Datasets. AG2Vid (Bar et al. (2020)) is re-implemented as the baseline model because it is the most related work. Unlike our method which only needs visual input and action sequence, AG2Vid also requires bounding boxes of all objects and progress meters of actions, i.e. clock edge, for training and testing. Furthermore, we conduct an ablation study by replacing concept slot module with the concatenation of features and tiled action vector, which is commonly used in low-level action-conditional video prediction (Finn et al. (2016)), to show the effectiveness of our module.

Metrics: To estimate the fidelity of action-conditional video prediction, MSE, SSIM (Wang et al. (2004)), PSNR and LPIPS (Zhang et al. (2018)) are calculated between the predictions and groundtruths. However, these metrics may not effectively tell if actions are successfully completed due to the small sizes of the moving objects. Hence, we also perform a human study to assess the accuracy of performing the correct action in generated videos for each model. The human judges annotate whether the model can identify the desired objects, perform actions specified by action
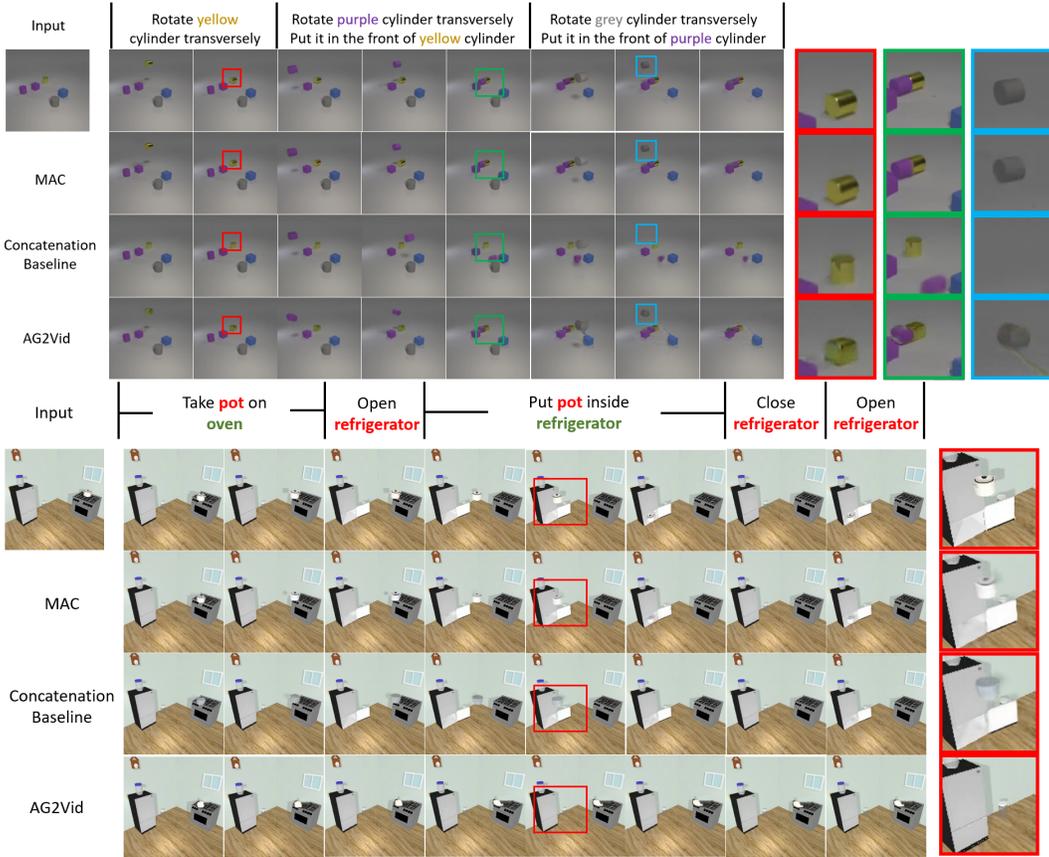
Figure 3: The qualitative comparison on CLEVR-Building-blocks and Sapien-Kitchen. The first row of each figure is the groundtruth sequence. The red, blue and green boxes highlight the quality of predictions by each method. In contrast to the success of MAC, concatenation-based method fails to find the correct destinations or to preserve attributes of moving objects. Also, bounding boxes used in AG2Vid cannot portray visual changes like rotations correctly.

| Model | CLEVR-Building-blocks | | | | Sapien-Kitchen | | | |
	SSIM↑	MSE↓	LPIPS↓	Accuracy↑	SSIM↑	MSE↓	LPIPS↓	Accuracy↑
Copy-First-Frame	0.962	251.38	0.1320	-	0.951	152.87	0.0393	-
Concatenation Baseline	0.961	226.53	0.1301	50.8%	0.962	23.13	0.0232	52.4%
AG2Vid	0.956	58.67	0.0399	78.8%	0.947	270.87	0.0684	5.2%
MAC	**0.983**	**43.52**	**0.0303**	**95.2%**	**0.971**	**11.16**	**0.0178**	**86.4%**

Table 1: Quantitative evaluation on CLEVR-Building-blocks and Sapien-Kitchen. All metrics are averaged frame-wisely except for accuracy.

labels and maintain the consistent visual appearances of all objects in its generations and only videos meeting all three criterions are scored as correct.

Results: The quantitative comparisons of all methods are summarized in Table 1. The MAC achieves the best scores on all metrics without access to additional information like bounding boxes, showing the superior performance of our concept slot module. The qualitative analysis in Fig 3 further reveals the drawbacks of other baselines. For CLEVR-Building-blocks, the concatenation-based variant fails to recognize the right objects due to its limited inductive bias. Although AG2Vid has no difficulty in identifying the desired objects, assumptions made by flow warping are too strong to handle rotation and occlusion. Consequently, the adversarial loss enforces AG2Vid to fix these errors by converting them to wrong poses or colors. These limitations of AG2Vid will be further amplified in a more complicated environment, i.e. Sapien-Kitchen. The same architecture used for CLEVR can only learn to remove the moving objects from their starting positions in Sapien-Kitchen because rotation and occlusion occur more often. The concatenation baseline performs better by showing correct
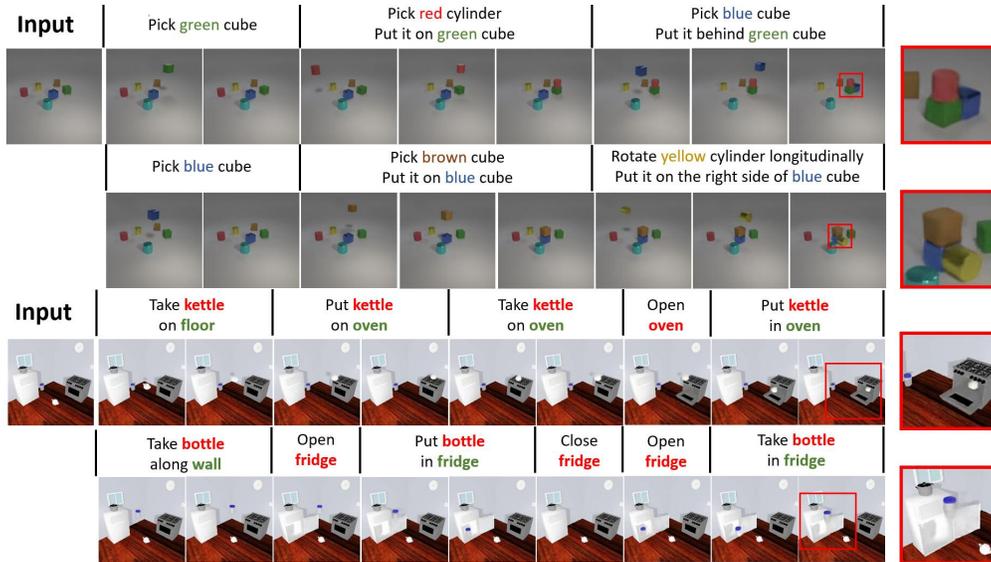
Figure 4: **Counterfactual video generation**: Conditioning on the same initial frame and different action labels, MAC can produce high-quality imaginations of counterfactual futures. Various visual outcomes present in the final frames are highlighted with red boxes and enlarged in the final column. **Top**: Generative results on CLEVR-Building-blocks. 34 frames are generated. **Bottom**: Generative results on Sapien-Kitchen dataset. 35 frames are generated.
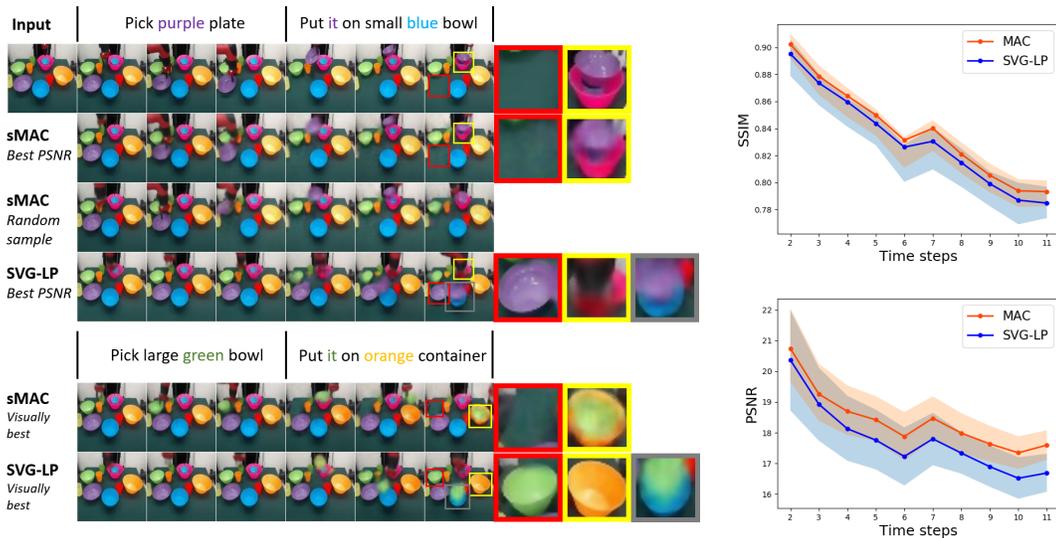

Figure 5: **Left**: Visual comparison between sMAC and SVG-LP on Tower-Creation. The supposed completions of *Pick* and *Put* in the final frames are highlighted by red and yellow boxes while incorrect completions in SVG-LP generations are labelled by grey boxes. The last two rows are counterfactual generations in which models are given different action labels. **Right**: Quantitative comparison per-frame. Higher SSIM and PSNR indicate better performance.

generation of open and close actions on large appliance. Yet, it still fails to produce long-term consistent predictions as the visual appearances of moving objects are altered. On the contrary, MAC can authentically depict the correct actions specified by action labels on both datasets.

4.2 COUNTERFACTUAL GENERATION

Counterfactual generation: The most intriguing application of MAC is counterfactual generation. More specifically, counterfactual generation means that our model will observe the same starting frame but receive different valid action labels to produce the corresponding future frames.
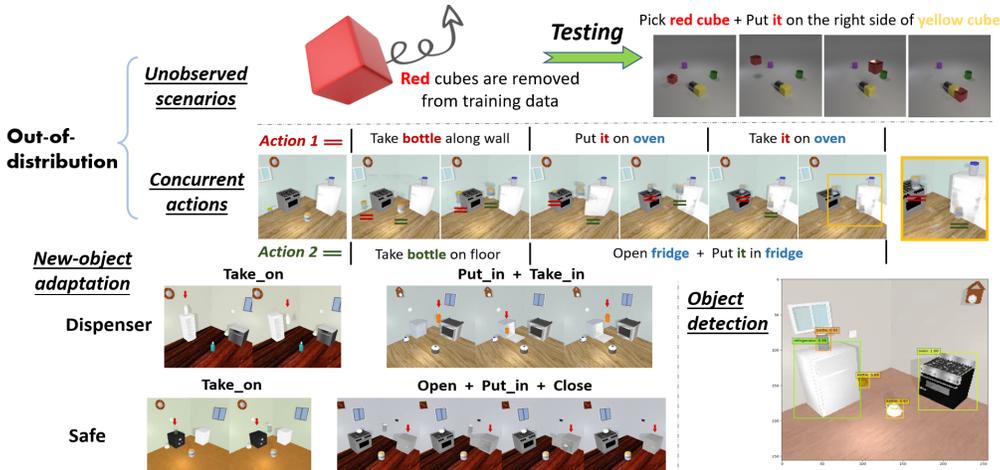
Figure 6: Compositional generalization and feature reuse. **Top**: Unobserved scenarios. All red cubes are removed from the tranining data, but the trained model can still manipulate red cube at test time. **Middle**: Concurrent actions. Inputting two action sequences at the same time. Both actions are depicted correctly. **Bottom Left**: New-object adaptation. Even with a few training samples, MAC can be fast adapted for generation of new objects. Red arrows point to new objects present in images. **Bottom Right**: Object detection.

Results: The visual results of counterfactual generations on each dataset are displayed in Fig 4. As we can see, our model successfully identifies the desired objects, plans correct trajectories toward the target places and generates high-quality imaginations of counterfactual futures. It is also worth noticing that all displayed generations are long-term generations , i.e. more than 30 frames are predicted for each sequence. Our recurrent predictor plays an very important role in sustaining the spatiotemporal consistency and in reconstructing the fully-occluded objects.

4.3 STOCHASTIC VIDEO GENERATION

Baselines and setup: We continue to evaluate the stochastic version of MAC (sMAC) on Tower-Creation dataset. SVG-LP was extended to action-conditional version in (Villegas et al. (2019)) so that we can adopt it as the baseline model to demonstrate the effectiveness of concept slot module.

Results: The qualitative and quantitative comparison between sMAC and action-conditional SVG-LP is provided in Fig 5. Although SVG-LP can partially understand the given action labels, it often fails to locate and manipulate the desired objects. Consequently, it will generate the moving object out of nowhere and often place it on a wrong target object. In contrast, sMAC can successfully simulate the trajectory of robotic arms and correctly animate the "*Pick*" and "*Put*" actions thanks to the concept slot module. Row 3 and 5 in Fig 5 show that sMAC is also capable of producing diverse future frames and predicting counterfactual results following different action instructions.

4.4 COMPOSITIONAL GENERALIZATION

We further explore other interesting features of our MAC. We first demonstrate that MAC is capable of making out-of-distribution generalization by designing two experiments. We evaluate how quickly our model can be adapted to new objects. It turns out for each new object, the trained MAC only requires a few training video examples to generate decent results. Finally, to verify that our model encodes the spatial information, we add SSD (Liu et al. (2016)) head after the frozen encoder and concept slot layer to conduct object detection.

Unobserved scenarios: We design an interesting experiment where only a subset of CLEVR-Building-blocks data are used for training and check what will happen if we input the unobserved action labels to the trained model. More precisely, we exclude all videos manipulating red cubes in the training sets and send the instructions involving red cubes at test time. The visualization of this experiment can be found in Fig 6. As we can see, MAC can still identify and manipulate red cubes correctly, showing its ability to recombine the learnt concept to comprehend new objects.

Concurrent actions: Concurrent actions means multiple action inputs at the same time. It can be considered as out-of-distribution generalization because our model only observes single-action videos during training. Generating concurrent-action videos needs to employ copied constituent slots and

parallel hidden states. As illustrated in Fig 6, MAC can linearly integrate the action information in the latent space and correctly portray 2 concurrent actions in the same scene.

Adaptation: We add a new openable category "*safe*" and a new movable category "*dispenser*" into Sapien-Kitchen and generate 100 video sequences for each new object showing its interaction with other objects. Approximately, there are about 5 new sequences created for each new action pair between 2 objects. Blank concept slots for new categories are attached to trained MAC and we finetune it on this small new training set. Visualization in Fig 6 shows that even with a few training samples, MAC is accurately adapted for video generation of new objects. This is because, with the help of concept slots, MAC can disentangle actions into relatively independent grounded concepts. When it learns new concepts, MAC reuses and integrates prior knowledge learnt from different cases.

Object detection: The quantitative results of object detection and more visualizations can be found in Appendix D. We observe that the features learnt by MAC can be easily transferred for detection as our video prediction task is highly location-dependent. This result indicates that utilizing bounding boxes might be a little redundant for some video tasks because videos already provide rich motion information that can be used for salient object detection.

5 RELATED WORK

Video prediction: ConvLSTM (Shi et al. (2015)) was the first deep learning model that employed a hybrid of convolutional and recurrent units for passive video prediction. This architectural design was soon followed by studies looking at a similar problem (Kalchbrenner et al. (2017); Mathieu et al. (2015); Wang et al. (2017); Yu et al. (2019); Wang et al. (2018b)). However, the capability of passive video prediction framework is very limited as models usually don't have sufficient information to predict the long-term future due to partial observation, egomotion and randomness. More importantly, this setting prevents models from interacting with environment. On the other hand, the low-level action-conditional video prediction task provides an action vector at each timestep as additional input to guide the prediction (Oh et al. (2015); Chiappa et al. (2017); Babaeizadeh et al. (2017); Wu et al. (2021)). CDNA (Finn et al. (2016)) is a representative of such models. In CDNA, the states and action vectors of the robotic manipulator are first spatially tiled and integrated into the model through concatenation. SVG (Denton & Fergus (2018)) was initially proposed for stochastic video generation but later was extended to action-conditional version in (Villegas et al. (2019)). It is worth noticing that SVG also used concatenation to incorporate action information. Such implementations are prevalent in low-level action-conditional video prediction because the action vector only encodes the spatial information of a single entity, usually a robotic manipulator (Finn et al. (2016)) or a human hand. A common failure case for such models is the presence of multiple affordable entities (Kim et al. (2019)), a scenario that our task definition and datasets focus on.

Modularity: Mixture of Experts refers to a classical machine learning technique where various learners are employed, each of which specializes in one particular function, and their output are aggregated through a gating function. This modular design makes each submodule relatively independent and thus leads to better generalization and robustness to compositional changes, which has been studied in several works (Goyal et al. (2019); Afshar et al. (2021); Sabour et al. (2017); Henaff et al. (2016)). In this work, we hypothesis that the underlying syntactic structures of semantic labels can tell how to aggregate the representations of individual concept learners. By translating labels into constituency trees, action graphs are embedded into the learning system to get the entire perspective of ongoing activities while each concept learner can focus on its specific subtask.

6 CONCLUSION

In this work, we propose the new task of semantic action-conditional video prediction and introduce 3 new datasets that are meant to bridge the gap towards a robust solution to this task in complex interactive scenarios. MAC, a novel video prediction model, was also designed by utilizing the idea of Mixture of Experts to ground action concept for video generation. Our proposed model can generate alternative futures without requiring additional auxiliary data such as bounding boxes, and is shown to be both quickly extendible and adaptable to novel scenarios and entities. It is our hope that our contributions will advance progress and understanding within this new task space, and that a model robust enough for real-world applications (i.e. in robotic systems) in perception and control will be eventually proposed as a descendant of this work.

REFERENCES

Parnian Afshar, Farnoosh Naderkhani, Anastasia Oikonomou, Moezedin Javad Rafiee, Arash Mohammadi, and Konstantinos N Plataniotis. Mixcaps: A capsule network-based mixture of experts for lung nodule malignancy prediction. *Pattern Recognition*, 116:107942, 2021.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

Amir Bar, Roei Herzig, Xiaolong Wang, Gal Chechik, Trevor Darrell, and Amir Globerson. Compositional video synthesis with action graphs. *arXiv preprint arXiv:2006.15327*, 2020.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2015.

Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 753–769, 2018.

Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*, 2017.

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

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pp. 64–72, 2016.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*, 2016.

De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles. Finding" it": Weakly-supervised reference-aware visual grounding in instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5948–5957, 2018.

Andrew Hundt, Varun Jain, Chia-Hung Lin, Chris Paxton, and Gregory D Hager. The costar block stacking dataset: Learning with workspace constraints. *arXiv preprint arXiv:1810.11714*, 2018.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10236–10247, 2020.

Beibei Jin, Yu Hu, Qiankun Tang, Jingyu Niu, Zhiping Shi, Yinhe Han, and Xiaowei Li. Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4554–4563, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

Nal Kalchbrenner, Aäron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. Proceedings of Machine Learning Research, 2017. URL http://proceedings.mlr.press/v70/kalchbrenner17a.html.

Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *Advances in Neural Information Processing Systems*, pp. 3814–3824, 2019.

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

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

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pp. 8733–8744, 2018.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pp. 879–893. PMLR, 2018.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pp. 2863–2871, 2015.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pp. 3856–3866, 2017.

Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pp. 32–36. IEEE, 2004.

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.

Xingjian Shi, 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.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

Ruben Villegas, Arkanath Pathak, Harini Kannan, Dumitru Erhan, Quoc V Le, and Honglak Lee. High fidelity video prediction with large stochastic recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 81–91, 2019.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and S Yu Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems*, pp. 879–888, 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. *arXiv preprint arXiv:1804.06300*, 2018a.

Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations*, 2018b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Bohan Wu, Suraj Nair, Roberto Martin-Martin, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. *arXiv preprint arXiv:2103.04174*, 2021.

Yue Wu, Rongrong Gao, Jaesik Park, and Qifeng Chen. Future video synthesis with object motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5539–5548, 2020.

Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11097–11107, 2020.

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.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2248–2255, 2013.

A Variational Lower Bound Derivation

The original variational lower bound was derived in (Kingma & Welling (2013)).

$$\log p_\theta(\mathbf{x}) = \log \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})$$

$$= \log \int_{\mathbf{z}} p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{q_\phi(\mathbf{z} \mid \mathbf{x})}$$

$$= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}$$

$$\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{p(\mathbf{z})}$$

$$= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x} \mid \mathbf{z}) - D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z}) \right)$$

$$= \sum_t \big[\mathbb{E}_{q_\phi(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})} \log p_\theta \left(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}, \mathbf{z}_{1:t} \right)$$

$$- D_{KL} \left(q_\phi \left(\mathbf{z}_t \mid \mathbf{x}_{1:t} \right) \| p \left(\mathbf{z}_t \right) \right) \big]$$

The final step (Denton & Fergus (2018)) 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 (Dinh et al. (2014)). Following (Yu et al. (2019)), 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.

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

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

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:

$$x^1 = \hat{x}^1 - \mathcal{F}_2(\hat{x}^2) \tag{4}$$

$$x^2 = \hat{x}^2 - \mathcal{F}_1(x^1) \tag{5}$$

Pixel shuffle layer (Shi et al. (2016)), 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 VGG16 (Simonyan & Zisserman (2014)) 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 (Kingma & Ba (2014)) 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

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

Table 2: Quantitative measures of object detection on Sapien-Kitchen in terms of average precision.
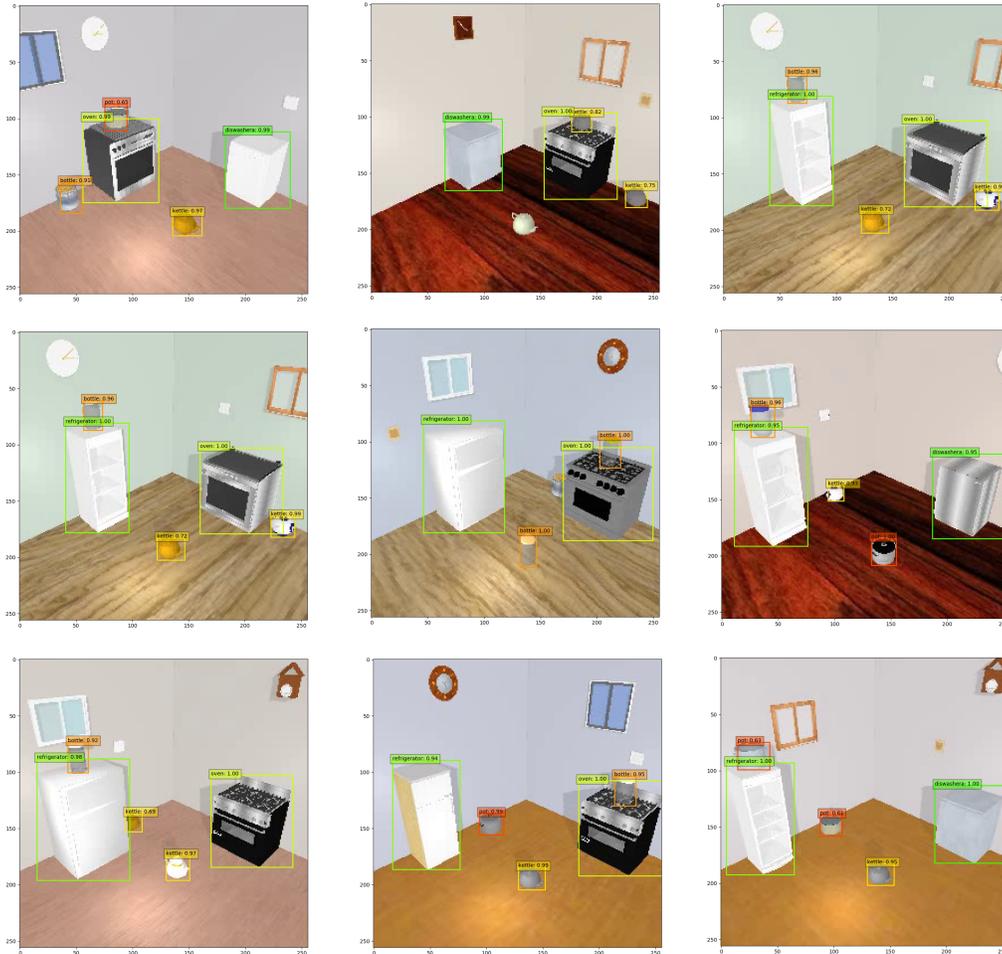

Figure 7: Visualization of 2D Object Detection on Sapien-Kitchen.

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 2 and Fig 7. 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.