GATSBI: Generative Agent-centric Spatio-temporal Object Interaction

Cheol-Hui Min Jinseok Bae Junho Lee Young Min Kim

Dept. of Electrical and Computer Engineering, Seoul National University, Korea
{mch5048, capoo95, twjhlee, youngmin.kim}@snu.ac.kr

Abstract

We present GATSBI, a generative model that can transform a sequence of raw observations into a structured latent representation that fully captures the spatio-temporal context of the agent's actions. In vision-based decision making scenarios, an agent faces complex high-dimensional observations where multiple entities interact with each other. The agent requires a good scene representation of the visual observation that discerns essential components and consistently propagates along the time horizon. Our method, GATSBI, utilizes unsupervised object-centric scene representation learning to separate an active agent, static background, and passive objects. GATSBI then models the interactions reflecting the causal relationships among decomposed entities and predicts physically plausible future states. Our model generalizes to a variety of environments where different types of robots and objects dynamically interact with each other. We show GATSBI achieves superior performance on scene decomposition and video prediction compared to its state-of-the-art counterparts.

1. Introduction

An ideal intelligent agent should be able to learn various tasks in diverse environments without relying on specific sensor configurations or control parameters. Recent approaches employ visual observation as the sole input to infer the physical context of the agent and its surroundings, thus aim to adapt to a general setup. One may interpret the visual input via conventional computer vision techniques employing deep neural networks [20, 33]. While they exhibit performance comparable to human perception, such approaches require a large volume of annotated database. Not only are the groundtruth labels costly to obtain, but also such supervised approaches are limited to specific tasks that they are trained on.

In contrast, unsupervised generative models extract the latent variables that encode the compositional relationship between different entities without prior knowledge [25]. The quality of representation can be verified by the ability of reconstructing the input video sequence from the disentangled latent variables [5]. In an ideal case, the latent variables contain the time-varying composition between the agent and the set of objects, and the structural knowledge must propagate temporally with a consistent inference of the latent dynamics. The latent dynamics of the learned representation reflect the underlying physics between the extracted entities, thus the agent can leverage the latent dynamics in predicting the various physical contexts conditioned on its own action.

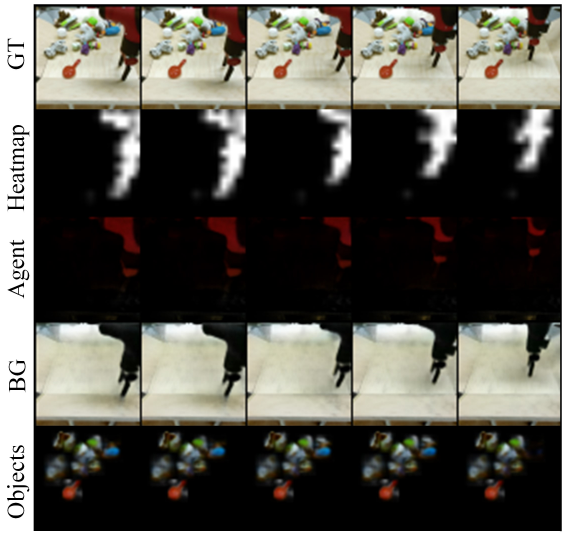


Figure 1. Our method, GATSBI, can explicitly identify the agent by utilizing the keypoint-based heatmap. Thus, the observation is decomposed into the agent, background, and objects. In addition, GATSBI infers the dynamic properties of the agent and temporally models the agent-centric interaction with the objects.

We propose a fully-unsupervised action-conditioned video prediction model, named Generative Agent-cenTric Spatio-temporal oBject Interaction (GATSBI). Our method is explicitly designed for vision-based learning of robot agents and is able to distinguish the active, passive and static com-

ponents from the robot-object interaction sequence, Fig. 1. Conditioning only on actions and a few frames, the learned latent dynamics can predict the long-term future observations without any prior labels of individual components or physics model.

Our generative model sequentially factorizes each video frame into individual components and extracts the latent dynamics. Specifically, our unsupervised network first models relatively large scene components as 2D-Gaussian mixture model (GMM). In addition, a group of 2D-Gaussian keypoints captures actively moving pixels in response to the given action. One of the GMM modes that matches best with the keypoint-based representation is selected and refined to learn the latent dynamics of the active agent. In the meantime, small passive objects are extracted by attention-based object discovery models [30]. Finally, graph neural networks (GNN) encodes the interactions between the active agent, passive objects, and static background that are disentangled in the extracted latent variable. The three different categories of the scene entities are reflected as inherent physical properties within the graphical model, which correctly updates the state of each object in response to the diverse interactions.

In summary, GATSBI is an unsupervised representation learning framework that infers a decomposed latent representation of the observation sequence and predicts associated latent dynamics in an agent-centric manner. GATSBI can distinguish various components and correctly understand the causal relationship between them from a sequence of visual observations without specific labels or prior. Being able to locate the active agent, the acquired latent representation is aware of the dynamics in response to the control action, and can readily be applied to an agent in making physically-plausible decisions. We provide extensive investigation on both qualitative and quantitative performance of GATSBI for video prediction on various robot-object interaction scenarios. We also compare our model with previous methods on spatio-temporal representation learning and show their promise and limitation.

2. Related Work

2.1. Object-centric Representation Learning

Deep generative models project the high-dimensional visual observation into low-dimensional latent representations [5, 25, 35]. Especially, object-centric representation learning extracts a structured representation that can be mapped into semantic entities.

The representation can be grouped into three categories depending on how the entities are distinguished: attention-based, spatial mixture-based, and keypoint-based methods. Attention-based methods [6,7,12,26,36,37] use spatial attention for object discovery and capture the locally consistent spatial features. They are good at detecting a large number of scene entities that are confined to small segments of the scene [7]. Spatial mixture methods [3,11,15,16], on the other hand, represent relatively large scene entities with Gaussian-mixture model (GMM). In contrast to the attention-based method, they struggle with scaling to a larger number of scene entities [31]. Keypoint-based methods [28, 32] extract keypoints from feature maps in an unsupervised way and are recently getting attention for their flexibility in representation. All of the three approaches have different capabilities of representation, and we carefully coordinate them to correctly disentangle various scene components in an unsupervised setting. As a concurrent work, [2] augments typical convolutional neural networks (CNN) with a graph architecture to find the scene structure during the pixel encoding process.

2.2. Latent Dynamics Model from Visual Sequence

The latent representation with discovered objects can be extended to model the temporal transition and interaction of the detected objects [6, 7, 23, 26, 27, 36, 37, 39]. Specifically, state-space model (SSM) [5, 8, 14, 24] utilizes recurrent neural networks (RNN) [4, 21] to pass latent information over a long time sequence, then a graph neural network (GNN) is used to model the interaction between entities [27]. Concurrent works temporally extend spatial mixture model to achieve the same objective [43, 44]. The aforementioned works use object-centric representation to model passive dynamics within the scene, but do not model the intelligent agents.

On the other hand, several recent works incorporate the action (i.e., control command) of the agents in the latent representation [1, 13, 17–19, 29, 40, 41, 45]. While one can use the convolutional recurrent neural network to embed the entire past observations and actions to yield rich temporal information [1], most works first extract the low-dimensional latent dynamics model from the observation with action-conditioned SSM, and integrate the learned latent model into the agent's policy [29] or vision-based planning [19]. However, these approaches use a simple variational autoencoder to extract the latent state and thus cannot represent entity-wise interaction. Previous approaches using structured representation in control tasks either do not detect active agents [40] or are not tested on the scenes with agents [30, 42]. Compared to these approaches, GATSBI learns representation that explicitly locates active agents and is suitable for learning the different physical properties of agent-object interactions or that of object-object interactions.

3. GATSBI: Generative Agent-centric Spatio-temporal Object Interaction

Given a sequence of observation $o_{0:T}$ and action $a_{0:T}$, GATSBI is designed to embed individual frames into a set of

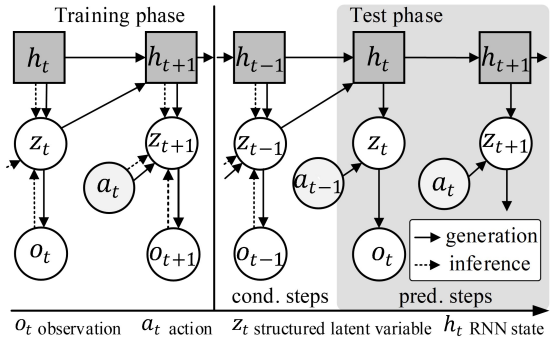


Figure 2. A probabilistic graphical model of GATSBI. Left: in the training phase, a set of structured latent variables z_t is inferred (dashed lines) by leveraging recurrent states h_t and observation o_t. Right: after updating h_t from z_t conditioned on observations for a few steps, GATSBI consecutively generates (solid lines) the future observations by leveraging recurrent states.

decomposed latent variables z_t which allows us to explicitly represent the dynamics of the agent and resulting entity-wise interactions within the latent space.

Our representation of the observation bases on the variational autoencoder (VAE) [25] that encodes the high-dimensional visual observation o into a low-dimensional latent variable z sampled from a probabilistic distribution. We can approximate the probability distribution of the observation $p_\theta(o)$ by maximizing the following empirical lower bound,

$$\log p_\theta(o) \geq \mathbb{E}\left[\log p_\theta(o|z)\right] - \mathrm{D}_{\mathrm{KL}}\left(q_\phi(z|o) \parallel p_\theta(z)\right). \quad (1)$$

The lower bound on the right side of the inequality is the evidence lower bound (ELBO) [25] and optimized with neural networks parameterized by θ and ϕ. $p_\theta(o|z)$, $q_\phi(z|o)$, and $p_\theta(z)$ represent the observation likelihood, posterior distribution, and the prior distribution, respectively.

Adopting the state-space model (SSM) [24], we can temporally expand the basic VAE as in Fig. 2. Given a sequence of observation $o_{0:T}$ and the action $a_{0:T}$, the set of structured latent embedding $z_{0:T}$ is also defined as a temporal variable. In order to maintain the consistent structured representation of the complex observation sequence, RNN memorizes the information to the hidden state $h_{0:T}$,

$$h_t = \mathrm{LSTM}(z_{t-1}, \mathrm{CNN}(o_{t-1}), h_{t-1}). \quad (2)$$

The hidden state h_t is leveraged with the action a_{t-1} for both *posterior inference* and *prior generation*. The posterior distribution $q_\phi(z_t|o_t, a_{t-1}, h_t)$ projects the high-dimensional observation into the latent space (inference, dashed line in Fig. 2). Sampling q_ϕ provides the compact semantic of the scene from which the agent can make a decision. Further, $p_\theta(z_t|a_{t-1}, h_t)$ models the prior knowledge of such semantic given the action [29]. We can model a function that predicts the future semantics by incorporating p_θ with proper latent dynamics model, and generate new observations (generation, solid line in Fig. 2).

GATSBI further encodes the spatio-temporal context by factorizing the latent embedding z_t into the background, agent, and objects. The history h_t is factorized accordingly to represent the entity-wise states, and we train dedicated LSTMs in Eq. (2) for each posterior-prior sampling of the individual entities. This way GATSBI maintains the spatio-temporal consistency of different entities.

In addition, we guarantee a comparable contribution of the action to the latent dynamics by enhancing its dimension. Since the action as a raw vector is relatively low-dimensional compared to the observation, we increase the dimension of the action with a multi-layer perceptron $\hat{a}_t = \mathrm{MLP}(a_t)$. In contrast to the entity-wise history h_t, $\hat{a}_t$ is shared across different modules of GATSBI. During the sampling process, the action a_t plays a key role in identifying the scene entities and modeling the interaction among them.

In summary, GATSBI is a recurrent-SSM that samples a disentangled z_t conditioned on a_{t-1} and entity-wise h_t. In the following, we further explain the action-conditioned entity-wise decomposition (Sec. 3.1) and the interaction dynamics between them (Sec. 3.2).

3.1. Entity-wise Decomposition

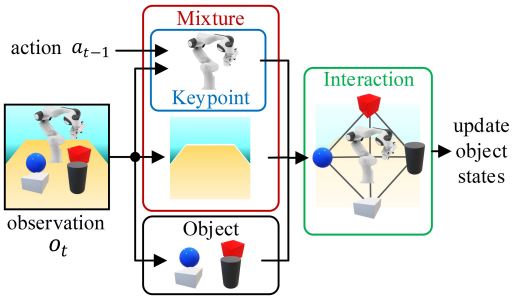


Figure 3. An overall scheme of GATSBI. *mixture module* extracts large components leaving the objects. *keypoint module* specifies the agent from the mixture and the remaining entities are assigned as background. *Object module* passes the objects into *interaction module* where a GNN updates the state of objects.

GATSBI disentangles different entities from the observation sequence and models the interaction between them. A similar goal has been achieved using attention-based object discovery [6, 7, 12, 26, 36, 37], but they can only represent passive interactions among small objects. Specifically, they divide the frame into a coarse grid and individual objects are assigned into one of the cells. However, when the agents are actively interacting with and manipulating objects within the scene, the motions of agents cannot be constrained within the size of a cell. GATSBI is explicitly designed to locate an active agent in an unsupervised fashion, which appears in diverse motion and shape.

At a high-level, GATSBI decomposes the entities within the observation in three steps as shown in Fig. 3. First, the *mixture module* acquires latent variables that embed the

Gaussian mixture model (GMM) of the static background and the active agent. Next, one of the mixture modes is specified as an agent by the *keypoint module*, whereas the remaining modes are specified as the background. The keypoint module detects dynamic features in observation, where the movement is highly correlated with the action of the agent. In the meantime, the *object module* discovers passive scene entities adapting attention-based object discovery [30]. The resulting entities are the active agent, static backgrounds, and the passive objects. Finally, the *interaction module* constructs the agent-centric interaction graph with the decomposed entities, and updates the hidden state of the object properties. This makes GATSBI accurately reflect the complex interactions caused by the agent.

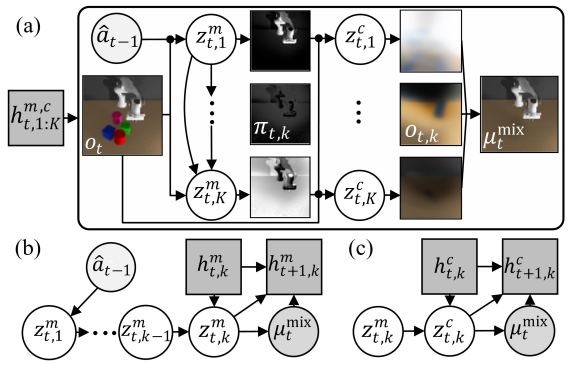


Figure 4. Spatio-temporal GMM. (a): Conditioning on the recurrent states $h^{m,c}_{t,k}$ and the action of the agent, the mixture module spatially decomposes observation o_t into K individual latent variables $z^{m,c}_{t,k}$ that comprise the mixture μ^{mix}_t. (b): The recurrent states of each mask variable is temporally updated $h^m_{t,k} \rightarrow h^m_{t+1,k}$ from autoregressive $z^m_{t,k}$ and μ^{mix}_t, (c): while temporal update of each component variable $h^c_{t,k} \rightarrow h^c_{t+1,k}$ is done by $z^c_{t,k}$ and μ^{mix}_t.

Mixture Module. The GMM-based representation learning [3, 11, 15] is one way to extract separate entities in the latent representation given the observation o. In contrast to the standard latent representation z of VAE, it assumes that there exist K entities in the scene, and each entity is embedded into separate latent variables $z_k,\ k = 1, \dots, K$ that follow a Gaussian distribution. Therefore the overall distribution is represented as the mixture of K Gaussians.

We handle the structure and the appearance of individual components separately, and this information should be consistently propagated over a time sequence. As shown in Fig. 4, omitting the time index t, GATSBI factorizes the latent variable for each entity z_k into a mask z^m_k and the corresponding component z^c_k. The observation likelihood $p_\theta(\mu^{\text{mix}}|z^m_{1:K}, z^c_{1:K})$ conditioned on these is formulated as

$$p_\theta(\mu^{\text{mix}}|z^m_{1:K}, z^c_{1:K}) = \sum_{k=1}^{K} \pi_\theta(z^m_k) p_\theta(o_k|z^c_k). \quad (3)$$

For k-th entity, the latent variables for mask z^m_k generate the observation mask of M pixels in the image $\pi_\theta(z^m_k) \in [0,1]^M$ whereas z^c_k encodes the component appearance and generates the observation $p_\theta(o_k|z^c_k)$. The mask variable $z^m_{1:K}$ is formulated such that the occupancy of individual scene entities are decided sequentially, i.e., $\pi_\theta(z^m_{1:K}) = \prod_{k=1}^{K} \pi_\theta(z^m_k|z^m_{1:k-1})$. Then $z^c_{1:K}$ is conditioned on the mask $z^m_{1:K}$. This makes z^m_k first determine how much portion each entity k contributes to o then z^c_k determine how each component looks like.

As mentioned, the spatial decomposition is temporally extended where the entity-wise history $h^m_{t,k}$ and $h^c_{t,k}$ follow the update rule defined as Eq. (2). At each time step t, we condition the sampling of latent variables of the first mask on the enhanced action from the previous time step $\hat{a}_{t-1}$ as well as its own history $h^m_{t,k=1}$,

$$z^m_{t,1} \sim q_\phi(z^m_{t,k=1}|o_t, \hat{a}_{t-1}, h^m_{t,k=1}). \quad (4)$$

We optimize q_ϕ and p_θ with the ELBO objective in Eq. (1). In this way, the posterior network q_ϕ learns the latent transition from $z^m_{t-1,k}$ to $z^m_{t,k}$ that is induced by a_{t-1}. In addition, with the sequential inference of the mask latent variables, conditioning on $z^m_{t,1}$ transfers the effect of enhanced action for all modes of the mixture model. Therefore, action-conditioning effectively increases the correlation between the action and the masks, and eventually coordinates the motion of the agent with the temporal change of the masks. The equations for the full sampling process and the objective of the mixture module are included in Sec.A.1 of the supplementary material.

With the limited number of modes for the Gaussian mixture, objects in the observation are less prone to be captured by the mixture module. The weighted sum of components constitutes a reconstruction of the scene where only the agent and background entities exist $\mu^{\text{mix}}_t = \sum_{k=1}^{K} \pi_{t,k} o_{t,k}$. As only the agent and the backgrounds forms μ^{mix}_t, we can find the salient feature which solely consists the objects $o_t - \mu^{\text{mix}}_t$. We use this for better object discovery.

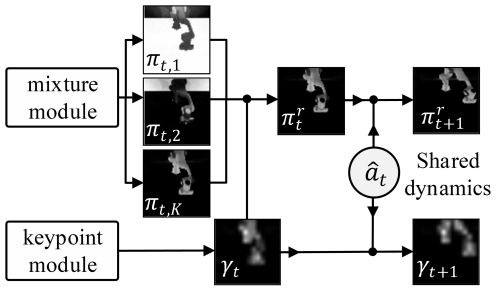


Figure 5. Keypoint module. By comparing against the keypoint map γ_t, we find the index of the agent mask, and fine-tune it to segment out the exact morphology of the agent. The dynamics of keypoints and the mask of the agent are shared through enhanced action output.

Keypoint Module. Even though the mixture module extracts the spatial layout of different entities, it is not trivial

to assign a specific index of modes k for the agent under general visual configuration. In the keypoint module, we utilize a swarm of N object keypoints [32] to describe the morphology of the agent and also represent the implication of their motions.

Fig. 5 describes how the keypoint module can extract the agent information from the mixture module. Given observation, the keypoint module detects salient features that actively move in response to the enhanced action $\hat{a}_t$ as a set of keypoints. The detected keypoints are aggregated to construct a keypoint map γ_t, from which we can compare and select the matching index k of the mixture mode. The details for finding the index are described in Sec. A.2 of the supplementary material.

More importantly, we modify the training objective in [32] as

$$\mathrm{D}_{\mathrm{KL}}\left(q_\phi(z_t^r|o_t, h_t^r, \hat{a}_{t-1}) \parallel p_\theta(z_t^r|h_t^r, \hat{a}_{t-1})\right) + \|\gamma_t - \pi_t^r\|. \tag{5}$$

The former term is the KL-divergence from ELBO in Eq. (1) conditioned on the history of the keypoints h_t^r and the enhanced action $\hat{a}_{t-1}$. The latter term represents the pixel-wise l_2 distance between the aggregated keypoint map γ_t and the mask of the robot agent $\pi_t^r = \pi_\theta(z_{t,k=r}^m)$ with the index $k = r$ specified for the agent.

Object Module. The object module adapts the attention-based object discovery by G-SWM [30] to find small objects that could not be captured by the mixture module. In addition, the object module can discover the rich attributes of individual components as well as their relational context. For the completeness of the discussion, we briefly introduce the formulation.

The input scene is first divided into coarse grid cells [12]. For each (u, v)-th cell of the 2D grid, a list of latent attributes are specified as $z_{(u,v)} = (z_{(u,v)}^{\text{pres}}, z_{(u,v)}^{\text{where}}, z_{(u,v)}^{\text{what}}, \dots)$. Each of the latent variables represents: the likelihood for its existence; position in the image space; its appearance; and optional other features [7, 27]. The dynamic history h_t^o of the latent vectors $z_t^o = \{z_{(u,v)}\}_t$ is condensed with a recurrent-SSM as other modules such that the module maintains the temporal consistency. The explicit representation of the latent vectors z_t^o enables probabilistic encoding of the various interactions in the state h_t^o. The information is accumulated in h_t^o using a fully-connected graph neural network [27, 30], whose nodes represent the discovered entities, and the edges encode the dynamic interaction between them. We further extend the approach and posit our agent-centric object interaction.

3.2. Interaction

The interaction module models the agent-centric interaction and can generate physically plausible future frames. After the entity-wise decomposition, GATSBI can extract information of the active agent z_t^r, h_t^r and I passive objects $z_{t,i}^o, h_{t,i}^o, i \in I$. The graph-based interaction in [27, 30] encodes the interaction dynamics of object i using the object feature $u_{t,i}$,

$$\tilde{\mathcal{I}}_{t,i} = \sum\nolimits_{j \neq i} f^o(u_{t,i}, u_{t,j}). \tag{6}$$

The interaction module of GATSBI extends the above formulation with two modifications. First, we confine the physical interaction only among k nearest neighbors, instead of the fully-connected graph in Eq. (6). By focusing on the entities in close proximity, we greatly reduce the number of edges in the graph. The reduced formulation not only allows the network to handle a larger number of objects, but also enhances the prediction accuracy as shown in the experimental results.

Second, GATSBI can model the interactions considering the spatio-temporal context, and separately handle the active, passive, and static components of other entities. This is the immediate benefit from the entity-wise decomposition in Sec. 3.1 and successfully modeling the acting agent within the scene. The spatial component uses the latent embedding of the object $z_{t,i}^o$ and the surrounding observation, which is obtained by cropping the non-object observation μ_t^{mix} near the object. Recall that μ_t^{mix} is reconstructed from the mixture module and corresponds to the scene without objects. The temporal aspect of an interaction is calculated along object feature $u_{t,i}^o$ and agent feature u_t^r. Similar to the object feature in [27, 30], the agent dynamics u_t^r is modeled from the latent variable of the agent z_t^r and its history h_t^r.

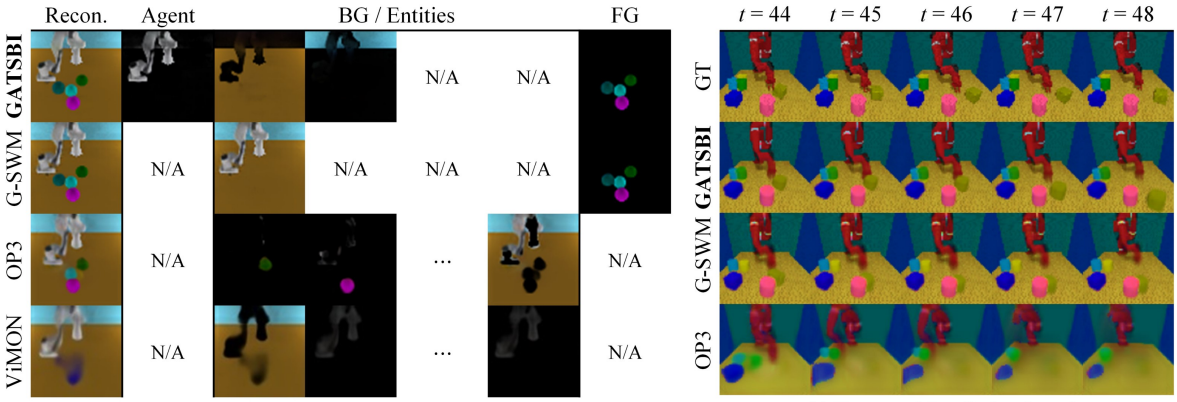
The total interaction $\mathcal{I}_{t,i}$ upon the object i is

$$\sum\nolimits_{j \in \mathcal{N}(i)} f^o(u_{t,i}^o, u_{t,j}^o) + f^s(\mu_t^{\text{mix}}, u_{t,i}^o) + f^t(u_t^r, u_{t,i}^o). \tag{7}$$

Here f^o, f^s, and f^t are neural networks that encode different interactions: f^o extracts passive interaction among objects included in $\mathcal{N}(i)$, the k-nearest-neighbor objects, while f^t encodes the response to the movement of the agent. Lastly, f^s takes only positional information into account. The state of each object i is updated with the aggregated dynamics as $h_{t+1,i}^o = \mathrm{LSTM}(\mathcal{I}_{t,i}, h_{t,i}^o)$. As demonstrated in Sec. 4, our unsupervised formulation accurately predicts the physical contact between the agent and multiple objects, and learns reasonable consequences to interactions.

4. Experiments

We evaluate the performance of extracted representation on four synthetic datasets, namely ROLL, PUSH1, PUSH2, and BALLS using physics-based robot simulators [22, 34]. The first three synthetic datasets involve a variety of interactions of agents under different appearances of background,



(a) Spatial Decomposition Results (b) Temporal Prediction Results

Figure 6. Spatial decomposition and temporal prediction results of GATSBI. (a): For ROLL dataset, GATSBI decomposes a scene into the agent, background, and objects. G-SWM disentangles scene into the background and the objects. Both OP3 and ViMON do not capture explicit scene entities. (b): Long-term prediction results of scenario with a difference of Cartesian pose defined as action. GATSBI predicts the long-term trajectory of agent and its interaction. Prediction of G-SWM is relatively inaccurate, and OP3 loses track of the agent.

agent, and object, whereas BALLS dataset contains the interaction sequence of multiple balls. Additionally, we use BAIR [10] to test our algorithm in a real-world dataset. The input observation is a video sequence that contains a robot agent interacting with objects, and the action space of the agent is defined as the 7 degree-of-freedom (DoF) joint velocities (6 DoF pose of the end-effector + gripper). The code and the dataset are available.[1] GATSBI is compared against the state-of-the-arts in the structured scene prediction: G-SWM [30], OP3 [40], and concurrent work ViMON [44]. We first show the results of spatial decomposition in Sec. 4.1 then examine the spatio-temporal prediction in Sec. 4.2. Finally, the design choices are verified with the ablation study in Sec. 4.3. Additional experimental results and settings are in the supplementary material.

4.1. Qualitative Results on Spatial Decomposition

The spatial decomposition is verified by a precise segmentation of the agent, background, and objects. Fig. 6 (a) shows the spatial decomposition result with the ROLL dataset. OP3 and ViMON decompose the scene into different mixture modes without knowledge about different entities. OP3 assigned the robot agent into several slots, and ViMON failed to separate the object entities, which shows an inherent problem of GMM-based approaches [31]. GATSBI overcomes this limitation of the GMM-based approaches by combining with attention-based object discovery, and successfully represents both amorphous shape and small entities. G-SWM utilizes attention-based object discovery to detect multiple entities of foreground objects but fails to represent a complex environment with the background because they use a unimodal Gaussian representation. While all previous works do not model the agent layer, GATSBI is designed to explicitly decompose the scene into background, agent, and objects.

4.2. Agent-centric Spatio-temporal Interaction

The precise spatial decomposition of GATSBI plays an essential role to make a physically plausible prediction in response to the agent. We test the performance of the video-prediction task. Given the initial five frames of a video sequence, the task is to predict the subsequent frames. For a fair comparison, previous works are modified to observe action sequence in the latent dynamics model. For G-SWM, which adopts a recurrent-SSM as GATSBI, we additionally augment its background latent dynamics with the input action, as the background slot is assumed to contain information related to agent movements. We use the configuration of OP3 that uses the action sequence to train for BAIR cloth manipulation dataset [9] and ViMON is also modified to adopt the action in the latent dynamics.

Fig. 7 shows a subset of frames predicted after observing the first five frames of PUSH1 dataset. As expected, GATSBI generates the agent-object interaction sequence that is nearly identical to the ground truth. G-SWM predicts a similar configuration of the agent, but the resulting movement of the object is not correctly predicted. OP3 generates slightly degraded robot agent configurations. The agent in ViMON is approximately similar, but the shape is blurry and not exact. The results imply that the segmentation of the agent and the agent-centric interaction contribute to accurate prediction of both the trajectory of the agent and the consequences of physical interaction.

The contribution of agent-centric representation of GATSBI is more prominent when tested with the real dataset, Fig. 8. Even though the motion of the agent in BAIR dataset is much more stochastic than the synthetic datasets, GATSBI robustly predicts the noisy movement of the agent. Since GATSBI adopts the object discovery module from G-SWM,

[1] https://github.com/mch5048/gatsbi

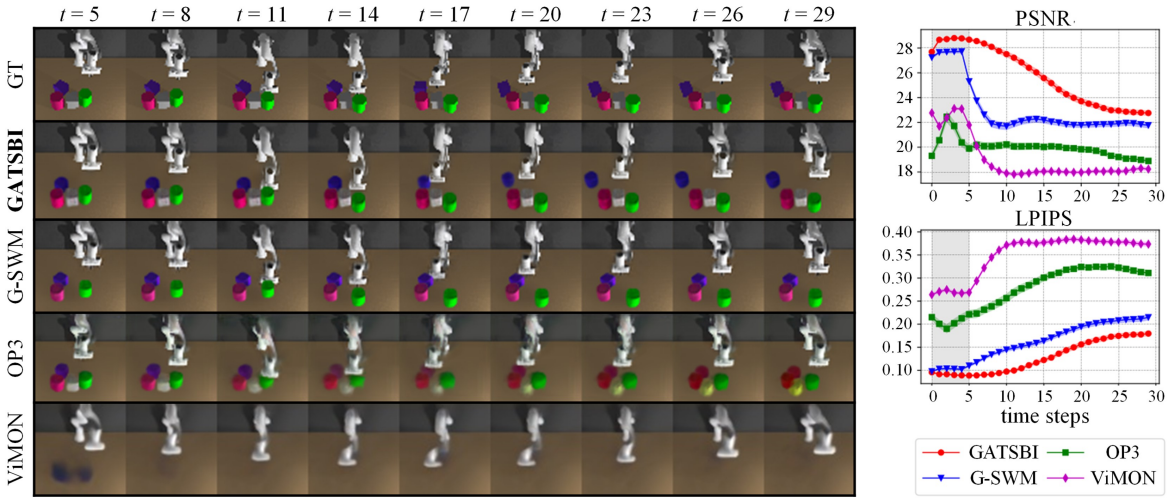


Figure 7. Spatio-temporal prediction results on PUSH1 dataset. Left: the prior generation process results over 25 prediction steps. The figure compares the reconstruction of predicted futures for each method. Right: Quantitative evaluation of predicted video frames. PSNR (*higher is better*) and LPIPS (*lower is better*) are plotted in 95%-confidence interval.

the reconstructions of foreground objects of the two models are nearly identical. However, G-SWM fails to predict the trajectory of the agent as the agent and action information is mixed in the background slot whereas GATSBI dedicates a separate layer for the agent. OP3 makes a relatively accurate prediction on the trajectory of the agent, but fails to capture the meaningful context of the scene, and ViMON totally fails to generate meaningful temporal context.

The graphs on the right side of Fig. 7 and 8 present the quantitative evaluation of the video prediction in terms of peak signal-to-noise ratios (PSNR) and learned perceptual image patch similarity (LPIPS) [46]. PSNR (higher is better) is a widely-used metric for video prediction that aggregates the pixel-wise differences of the predicted frames compared to the ground truth, and LPIPS (lower is better) measures how realistic the predicted frames are. GATSBI achieves superior performance in terms of both metrics. We observe that the mixture models of OP3 and ViMON have limited capacity to express detailed visual features and cannot faithfully recover the observation even for the first five frames (shaded in gray) where the ground truth is given. After the five frames, the system starts to make pure predictions and the performance rapidly deteriorates for all other approaches. In contrast, the curves for GATSBI are relatively smooth in both PSNR and LPIPS. It demonstrates that GATSBI leverages the information from observation much more effectively than other methods. Additional results with all of the datasets are available in the Sec. E of supplementary material.

Table 1 summarizes the performance with all four datasets measured with Fréchet video distance (FVD) [38]. FVD (lower is better) measures the distance in the feature space to reflect the similarity of human perception. GATSBI correctly models the latent dynamics of the agent and objects,

Table 1. FVD (*lower is better*) comparison for all methods on the four robotics dataset. The lower value implies the generated frames are closer to that of ground truth in the feature space. Bold values indicate the best performing method for each dataset. Values inside the parenthesis denote the 95%- confidence interval for each setup.

Models	ROLL	PUSH1	PUSH2	BAIR
GATSBI	**484.0** (27.57)	**630.4** (37.68)	**859.0** (35.43)	**1620** (55.63)
G-SWM	627.3 (30.00)	910.6 (76.89)	1072 (32.92)	2603 (121.4)
OP3	1025 (39.33)	1118 (35.19)	2568 (90.01)	2904 (128.0)
ViMON	1217 (28.98)	1620 (58.9)	2823 (93.54)	3983 (204.9)

and consistently exhibits superior results in all datasets. G-SWM can make a relatively precise trajectory prediction on synthetic datasets with the entity-wise decomposition and outperforms OP3 and ViMON by a large margin. However, it fails to model the agent-object interaction.

We further evaluate the GATSBI with PUSH2 dataset which we created with a different agent that moves significantly, interacting with more objects. In addition, we create a challenging setting by providing the change of translation and rotation of end-effector. The correct action configuration needs to be inferred from the relative action information and the history of agent motions. Fig. 6 (b) shows the five consecutive predicted frames with the noticeable agent-object interaction. The robot agent moves its end-effector and hits the yellow cube. GATSBI, with the explicit embedding of the agent dynamics incorporated in the interaction model, predicts the passive movement of the yellow cube. In contrast, G-SWM only predicts the motion of the agent and fails to capture the interaction. Lastly, OP3 and ViMON show poor prediction of the agent, and could not propagate the objects through time.

4.3. Ablation Study on Interaction

This section provides the ablation study on the interaction module. Further studies on the mixture and keypoint module

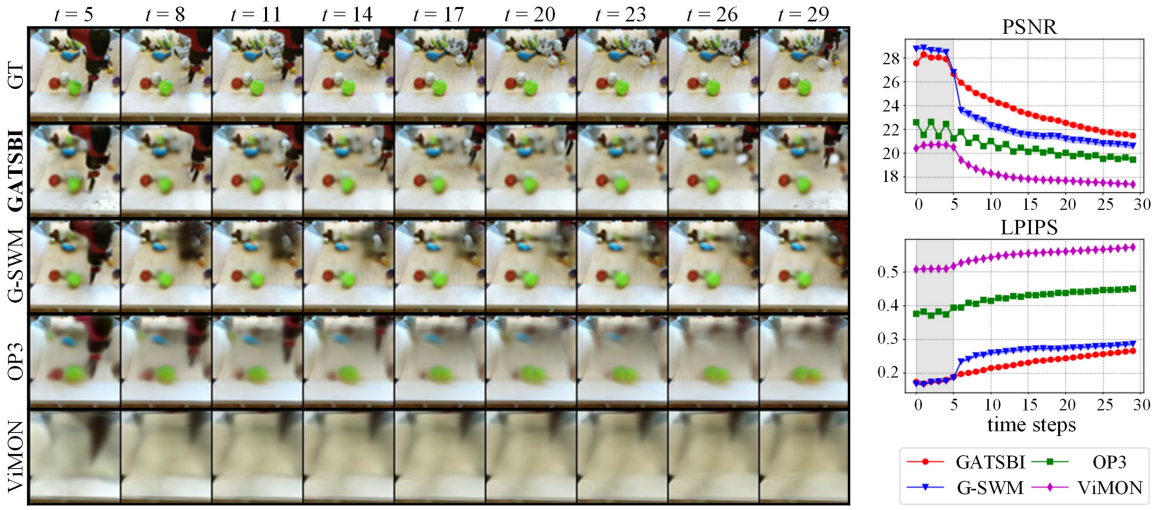


Figure 8. Spatio-temporal prediction results of BAIR dataset. Left: action-conditioned video prediction result on real-world robot dataset. Right: PSNR and LPIPS on BAIR dataset. Solid colored mean values are shaded by 95%-confidence interval.

Table 2. PSNR (*higher is better*), LPIPS (*lower is better*), FVD comparison among the three interaction modes.

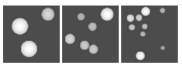
Mode	PSNR	LPIPS	FVD
INTER1	22.80 (0.2202)	0.2089 (6.227e-3)	841.6 (51.17)
INTER2	24.78 (0.1767)	0.1570 (2.672e-3)	484.0 (27.575)
INTER3	**25.44** (0.1765)	**0.1463** (2.622e-3)	**482.5** (22.320)

are included in the Sec. E.6 of supplementary material.

Comparison of Interaction Modes. Here we compare different methods of processing interactions, and demonstrate that the latent information of the agent enhances the performance of video prediction. First mode considers the interaction of individual objects as G-SWM, but the remaining components are regarded as a static background (INTER1). The other two methods extract the variables of the agent and The latent dynamics is incorporated into the interaction graph. INTER2 encodes the variable of the agent as a localized feature for each object, whereas INTER3 (*ours*) uses it as a global feature of the interaction network. We provide the detailed implementation of each mode in the Sec. A.3 of supplementary material. Table 2 shows the comparison among the three interaction models in terms of PSNR, LPIPS, and FVD in 95%-confidence interval. INTER3 performs the best, implying that the agent information provides sufficient constraints on all the objects within the scene.

Agent-free Object Interactions. Finally, we evaluate the k nearest neighbors search method in the object-object interaction model of GATSBI. We generate synthetic scenes where multiple objects interact with each other, and test the accuracy of video prediction with scenes. The synthetic scenes contain interactions of different numbers of balls as shown in the inset of Table 3. Table 3 presents the numerical pixel error before and after the interaction among objects. The result exhibits that the precision of interaction increases as the number of objects increases and outperforms the original fully-connected graphical model. With the sparse graph, the network better captures the physical context between multiple objects.

Table 3. Average pixel error for different connectivity of interaction graph. FC denotes the fully-connected graph model of G-SWM and KNN (k) denotes the k nearest neighbor graph model of GATSBI.

	Method	3 Balls	6 Balls	9 Balls
	FC	**3.039**	3.58	5.477
	KNN (3)	3.483	**3.374**	5.975
	KNN (5)	-	-	**3.775**

5. Conclusion

In this work, we proposed GATSBI, a spatio-temporal scene representation model that decomposes a video observation into an agent, background, and objects. With an appropriate representation of the action of the agent, our model reliably predicts the long-term trajectory of the agent as well as the physical interaction between the agent and other objects in the scene. The experimental results prove our agent-centric video prediction model can generate physically plausible future frames in various synthetic and real environments. Our method excels concurrent state-of-the-art methods both in the qualitative and quantitative results. In the future, we will apply GATSBI to solving vision-based robotics tasks, since our prediction model can be applied to model-based reinforcement learning.

6. Acknowledgment

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1C1C1008195) and the National Convergence Research of Scientific Challenges through the National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (NRF-2020M3F7A1094300).

References

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 2

[2] Daniel M Bear, Chaofei Fan, Damian Mrowca, Yunzhu Li, Seth Alter, Aran Nayebi, Jeremy Schwartz, Li Fei-Fei, Jiajun Wu, Joshua B Tenenbaum, et al. Learning physical graph representations from visual scenes. *arXiv preprint arXiv:2006.12373*, 2020. 2

[3] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. 2, 4

[4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[5] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015. 1, 2

[6] Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. *arXiv preprint arXiv:1911.09033*, 2019. 2, 3

[7] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3412–3420, 2019. 2, 3, 5

[8] Andreas Doerr, Christian Daniel, Martin Schiegg, Duy Nguyen-Tuong, Stefan Schaal, Marc Toussaint, and Sebastian Trimpe. Probabilistic recurrent state-space models. *arXiv preprint arXiv:1801.10395*, 2018. 2

[9] Frederik Ebert, Sudeep Dasari, Alex X Lee, Sergey Levine, and Chelsea Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. *arXiv preprint arXiv:1810.03043*, 2018. 6

[10] Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *arXiv preprint arXiv:1710.05268*, 2017. 6

[11] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019. 2, 4

[12] 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*, pages 3225–3233, 2016. 2, 3, 5

[13] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016. 2

[14] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in neural information processing systems*, pages 2199–2207, 2016. 2

[15] Klaus Greff, Raphaël Lopez Kaufman, Rishabh 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. 2, 4

[16] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6691–6701, 2017. 2

[17] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018. 2

[18] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019. 2

[19] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018. 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[22] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 5

[23] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *ICLR*, 2020. 2

[24] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick Van der Smagt. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016. 2, 3

[25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2, 3

[26] 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*, pages 8606–8616, 2018. 2, 3

[27] Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. *arXiv preprint arXiv:1910.02425*, 2019. 2, 5

[28] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. In *Advances in neural information processing systems*, pages 10724–10734, 2019. 2

[29] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *arXiv preprint arXiv:1907.00953*, 2019. 2, 3

[30] Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. *arXiv preprint arXiv:2010.02054*, 2020. 2, 4, 5, 6

[31] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020. 2, 6

[32] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. In *Advances in Neural Information Processing Systems*, pages 92–102, 2019. 2, 5

[33] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[34] Eric Rohmer, Surya PN Singh, and Marc Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. 5

[35] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015. 2

[36] Aleksandar Stanić and Jürgen Schmidhuber. R-sqair: relational sequential attend, infer, repeat. *arXiv preprint arXiv:1910.05231*, 2019. 2, 3

[37] Karl Stelzner, Robert Peharz, and Kristian Kersting. Faster attend-infer-repeat with tractable probabilistic models. In *International Conference on Machine Learning*, pages 5966–5975, 2019. 2, 3

[38] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 7

[39] 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. 2

[40] Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua B Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. *arXiv preprint arXiv:1910.12827*, 2019. 2, 6

[41] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pages 2746–2754, 2015. 2

[42] Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019. 2

[43] Marissa A Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S Ecker. Unmasking the inductive biases of unsupervised object representations for video sequences. *arXiv preprint arXiv:2006.07034*, 2020. 2

[44] Polina Zablotskaia, Edoardo A Dominici, Leonid Sigal, and Andreas M Lehrmann. Unsupervised video decomposition using spatio-temporal iterative inference. *arXiv preprint arXiv:2006.14727*, 2020. 2, 6

[45] Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. Solar: Deep structured representations for model-based reinforcement learning. In *International Conference on Machine Learning*, pages 7444–7453. PMLR, 2019. 2

[46] 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*, pages 586–595, 2018. 7