Object-Centric Representation Learning with Generative Spatial-Temporal Factorization

Li Nanbo
School of Informatics
University of Edinburgh
nanbo.li@ed.ac.uk

Muhammad Ahmed Raza
School of Informatics
University of Edinburgh
m.a.raza@ed.ac.uk

Hu Wenbin
School of Informatics
University of Edinburgh
wenbin.hu@ed.ac.uk

Zhaole Sun
School of Informatics
University of Edinburgh
zhaole.sun@ed.ac.uk

Robert B. Fisher
School of Informatics
University of Edinburgh
rbf@inf.ed.ac.uk

Abstract

Learning object-centric scene representations is essential for attaining structural understanding and abstraction of complex scenes. Yet, as current approaches for unsupervised object-centric representation learning are built upon either a stationary observer assumption or a static scene assumption, they often: i) suffer single-view spatial ambiguities, or ii) infer incorrectly or inaccurately object representations from dynamic scenes. To address this, we propose *Dynamics-aware Multi-Object Network* (DyMON), a method that broadens the scope of multi-view object-centric representation learning to dynamic scenes. We train DyMON on *multi-view-dynamic-scene* data and show that DyMON learns—without supervision—to factorize the entangled effects of observer motions and scene object dynamics from a sequence of observations, and constructs scene object spatial representations suitable for rendering at arbitrary times (*querying across time*) and from arbitrary viewpoints (*querying across space*). We also show that the factorized scene representations (w.r.t. objects) support querying about a single object by space and time independently.

1 Introduction

Object-centric representation learning promises improved interpretability, generalization, and data-efficient learning on various downstream tasks like reasoning (e.g. [18, 42]) and planning (e.g. [31, 4, 44]). It aims at discovering compositional structures around objects from the raw sensory input data, i.e. a *binding problem* [12], where the *segregation* (i.e. factorization) is the major challenge, especially in cases of no supervision. In the context of visual data, most existing focus has been on single-view settings, i.e. decomposing and representing 3D scenes based on a single 2D image [3, 10, 27] or a fixed-view video [24]. These methods often suffer from single-view spatial ambiguities and thus show several failures or inaccuracies in representing 3D scene properties. It was demonstrated by Nanbo et al. [32] that such ambiguities could be effectively resolved by multi-view information aggregation. However, current multi-view models are built upon a foundational static-scene assumption. As a result, they: 1) require static-scene data for training and 2) cannot handle well dynamic scenes where the spatial structures evolve over time. This greatly harms a model's potentials in real-world applications.

In this work, we target an unexplored problem—unsupervised object-centric latent representation learning in *multi-view-dynamic-scene* scenarios. Despite the importance of the problem to spatial-temporal understanding of 3D scenes, solving it presents several technical challenges. Consider one particularly interesting scenario where both an observer (e.g. a camera) and the objects in the scene are moving at the same time. To aggregate 3D object information from two consecutive observations, an agent needs not only to handle the cross-view object correspondence problem [32] but also to reason about the independent effects of the scene dynamics and observer motions. One can consider the aggregation as a process of answering two questions: "how much has an object really changed in the 3D space" and "what previous spatial unclarity can be clarified by the current view". In this paper, we refer to the relationship between the scene spatial structures and the viewpoints as the *temporal entanglement* because the temporal dependence of them complicates the identification of the *independent generative mechanism* [36].

We introduce DyMON (***Dy****namics-aware* ***M****ulti-***O****bject* ***N****etwork*), a unified unsupervised framework for multi-view object-centric representation learning. Instead of making a strong assumption of static scenes as that in previous multi-view methods, we only make two weak assumptions about the training scenes: i) observation sequences are taken at a high frame rate, and ii) there exists a significant difference between the speed of the observer and the objects (see Sec. 3). Under these two assumptions, in a short period, we can transition a *multi-view-dynamic-scene* problem to a *multi-view-static-scene* problem if an observer moves faster than a scene evolves, or to a *single-view-dynamic-scene* problem if a scene evolves faster than an observer moves. These local approximations allow DyMON to learn independently the generative relationships between scenes and observations, and viewpoints and observations during training, which further enable DyMON to address the problem of scene spatial-temporal factorization, i.e. solving the observer-scene *temporal entanglement* and scene object decomposition, at test time.

Through the experiments we demonstrate that: **(i)** DyMON represents the first unsupervised multi-view object-centric representation learning work in the context of dynamic-scene settings that can train and perform object-oriented inference on *multi-view-dynamic-scene* data (see Sec. 5). **(ii)** DyMON recovers the *independent generative mechanism* of an observer and scene objects from observations and permits querying predictions of scene appearances and segmentations across both space and time (see Sec. 5.1). **(iii)** As DyMON learns scene representations that are factorized in terms of objects, DyMON allows single-object manipulation along both the space (i.e. viewpoint) and time axis—e.g. replays dynamics of a single object without interferring the others (see Sec. 5.1).

2 Background

Object-centric Representations Consider object-centric representation inference as the inverse problem of an observation generation problem (i.e. the *vision-as-inverse-graphics* [43] idea). In the forward process, i.e. observation generation, we have a scene well-defined by a set of parameter vectors $\mathbf{z} = \{z_k\} = \{z_1, z_2, ..., z_K\}$, where a $z_k \in \mathbb{R}^D$ specifies one and only one object in the scene. An observation of the scene $\mathbf{x}$, e.g. an image $x \in \mathbb{R}^M$ or an RGB image $x \in \mathbb{R}^{M\times 3}$, can be taken only by a specified observer (often defined as $v \in \mathbb{R}^d$) which is independent of the scene in the forward problem, using a specific mapping $g : \mathbb{R}^D \times \mathbb{R}^d \mapsto \mathbb{R}^{M\times 3}$. Assuming a deterministic process, an observation x is generated as $x = g(\mathbf{z}, v)$, where v is often omitted in single-view scenarios (e.g. [3, 10]). With the forward problem defined, we can describe the goal of learning an object-centric representation as inferring the intrinsic parameters of the objects $\{z_k\}$ that compose a scene $\mathbf{z}$ based on the scene observation $\mathbf{x}$. In other words, computing a factorized posterior $p(\mathbf{z}|\mathbf{x}) = p(z_1, z_2, ..., z_K|\mathbf{x})$, even though it is computationally intractable. As the number of objects is unknown in the inverse problem, it is worth noting that i) K is often set

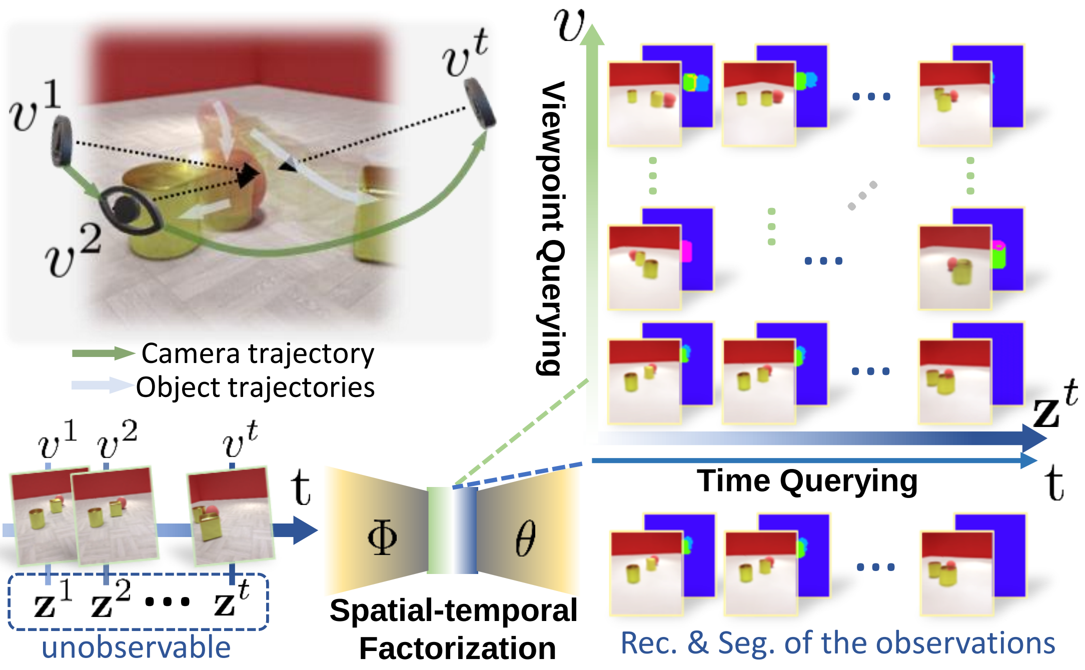

Figure 1: **Top Left:** *Multi-view-dynamic-scene* setup. v with a time index superscript denotes the spatial configuration (e.g. position, orientation, etc.) of an observer at a specific time. We highlight one particular interesting, yet unexplored, scenario where both an observer and scene objects are moving at the same time—which entangles the independent effects of the observer's and scene objects' motions on an scene observation, an image sequence (see **bottom left**). A latent variable $\mathbf{z}$ that is indexed by time describes the objects and their spatial configuration at a specific time (See Sec. 2 for detailed definition). **Right:** DyMON decouples the generative effects of observer motions and scene object motions and enables: 1) reconstruction and factorization of the observed views (see **bottom right**), and 2) novel-view appearance and decomposition prediction for arbitrary times—querying across both space and time (see **top right**).

globally to be a sufficiently large number (greater than the actual number of objects) to capture all scene objects, and ii) we allow empty "slots".

Temporal Entanglement The dynamic nature of the world suggests that the spatial configuration of a scene (denoted by $\mathbf{z}^t$) and an observer v^t are bound to the specific time t that an observation is taken (i.e. $x^t = g(\mathbf{z}^t, v^t)$). Let $\mathbf{X} = \{(x^t, v^t)\}_{1:T}$ [1] represent a data sample, e.g. a sequence or set of multi-view image observations, from dataset $\mathcal{D}$, where T is the number of the images in the sample. Assuming $\mathbf{z}^t$ is given in the data sample for now, i.e. focusing on the generative process only, we augment a scene data sample as $\mathbf{X}_a = \{(x^t, v^t, \mathbf{z}^t)\}_{1:T}$. In general, we assume an independent scene-observer relation: $\mathbf{z}^t \perp v^t | \emptyset$ but they nevertheless become dependent when the corresponding observation is given: $\mathbf{z}^t \not\perp v^t | x^t$. Under a static-scene assumption, we can treat an augmented data sample as $\mathbf{X}_a = \{(x^t, v^t), \mathbf{z}^t\}_{1:T}$ where $\mathbf{z}^t$ and v^t are separable (i.e. can be sampled independently). In this case, to recover the independent generative mechanism (i.e. train a $g(\cdot)$) w.r.t. scenes and observers from data, GQN [9] and MulMON [32] fix $\mathbf{z}^t$ to $\mathbf{z}$ and intervene on the viewpoints v^t. From a causal perspective, this can be seen as estimating $p(x^{t'} | \boldsymbol{do}(v^t = v^{t'}), \mathbf{z}^t = \mathbf{z})$, where $(x^{t'}, v^{t'}) \sim \{(x^t, v^t)\}_{1:T}$, implicitly under a causal model: $\mathbf{z}^t \rightarrow x^t \leftarrow v^t$. However, in dynamic settings, the same estimation, i.e. sampling $(x^{t'}, v^{t'}) \sim \{(x^t, v^t)\}_{1:T}$ independently of $\mathbf{z}^t$, is forbidden by the $(\cdot)$ indicator. Intuitively, an observer cannot take more than one observations from different viewpoints at the same time t. In this paper, we refer to this issue as *temporal entanglement* in view of the temporal implication of the $(\cdot)$ indicator.

[1] We define $(\cdot)$ as a joint sample indicator that forbids independent sampling of the random variables wherein.

3 DyMON

Our goal is to train a multi-view object-centric representation learning model that recovers the *independent generative mechanism* of scene objects and their motions and observer motions from dynamic-scene observations. In this section, we detail how DyMON addresses these two presented challenges: 1) temporal disentanglement (see Sec. 3.1), and 2) scene spatial factorization (see Sec. 3.2). We discuss the training of DyMON in Sec. 3.3.

3.1 Temporal Disentanglement

The key to resolving *temporal entanglement*, i.e. temporal disentanglement, is to enable sampling (x^t, v^t) independently of $\mathbf{z}^t$, or $(x^t, \mathbf{z}^t)$ independently of v^t. This is seemingly impossible in the *multi-view-dynamic-scene* setting as it requires to fix either $\mathbf{z}^t$ (static scene) or v^t (single-view) respectively. In this paper, we make two assumptions about the training scenes to ensure the satisfaction of the aforementioned two requirements without violating the global *multi-view-dynamic-scene* setting. Let us first describe the dynamics of scenes and observers with two independent dynamical systems:

$$\mathbf{z}^{t+\Delta t} - \mathbf{z}^t = \overline{f_{\mathbf{z}}}(\mathbf{z}^t, t)\Delta t \quad , \quad v^{t+\Delta t} - v^t = \overline{f_v}(v^t, t)\Delta t, \tag{1}$$

where t and $t + \Delta t$ are the times that two consecutive observations were taken, $\overline{f_{\mathbf{z}}}(\mathbf{z}^t, t)$ and $\overline{f_v}(v^t, t)$, or simply $\overline{f_{\mathbf{z}^t}}$ and $\overline{f_{v^t}}$, are the average velocities of scene objects and the observer within $[t, t + \Delta t]$. Note that we use a z^t to capture both the shape and pose information of an object. However, we do not consider shape changes in this work. With the dynamical systems defined, we introduce our assumptions (which defines a tractable subset of all possible situations) as:

- **(A1)** ***The high-frame-rate assumption*** $\Delta t \to 0$ s.t. $x^{t+\Delta t} \approx x^t$,
- **(A2)** ***The large-speed-difference assumption*** The data comes from one of two cases (SCFO: Slow Camera, Fast Objects or FCSO: Fast Camera Slow Objects), that satisfy: $|\frac{|\overline{f_{\mathbf{z}}}|}{|\overline{f_v}|}| \geq C_{SCFO}$ or $|\frac{|\overline{f_{\mathbf{z}}}|}{|\overline{f_v}|}| \leq C_{FCSO}$, where $|velocity|$ computes a speed, and C_{SCFO} and C_{FCSO} are positive constants.

A1 allows us to assume a nearly static scene $\mathbf{z}^t$ or a fixed viewpoint v^t for a short period. Consider an example where we assume a static scene, i.e. $\mathbf{z}^{\tau-\Delta t} \approx \mathbf{z}^{\tau} \approx \mathbf{z}^{\tau+\Delta t}$, in $[\tau - \Delta t, \tau + \Delta t]$, **A1** essentially allows us to extract $\mathbf{z}^t$ out of a joint sample as: $\mathbf{X}_a = \{(x^t, v^t), \mathbf{z}^t\}_{\tau-\Delta t:\tau+\Delta t}$. An intuitive way to define **A2** is: $|\overline{f_{\mathbf{z}}}| \gg |\overline{f_v}|$ or $|\overline{f_{\mathbf{z}}}| \ll |\overline{f_v}|$, which specify a large speed difference between scene speeds and observer speeds.

These two assumptions enable us to accumulate instant changes (velocities) on one variable (e.g. either $\mathbf{z}^t$ or v^t) over a finite number of Δt while ignoring the small changes of the other (assumed fixed). We then treat a *slow-camera-fast-objects* (i.e. SCFO) scenario, where $|\overline{f_{\mathbf{z}}}| \gg |\overline{f_v}|$, as an approximate *single-view-dynamic-scene* scenario, and a *fast-camera-slow-objects* (i.e. FCSO) scenario, where $|\overline{f_{\mathbf{z}}}| \ll |\overline{f_v}|$, an approximate *multi-view-static-scene* scenario. Either case allows us to resolve the *temporal entanglement* problem. Importantly, to answer the question: "is a given data sample an SCFO or FCSO sample", we need to quantitatively specify the two assignment criteria C_{SCFO} and C_{FCSO}. However, a direct calculation of these two constants is often difficult and does not generalize as: i) $|\overline{f_{\mathbf{z}}}|$ is not available in unsupervised scene representation learning data, and ii) the two constants vary across different datasets. In practice, we cluster the data samples into SCFO and FCSO clusters using only the viewpoint speed $|\overline{f_v}|$, i.e. assuming $|\overline{f_{\mathbf{z}}}| = 1$ for training (see Sec. 3.3). In testing, DyMON treats them equally.

3.2 Spatial Object Factorization

DyMON tackles scene spatial decomposition in a similar way to MulMON [32] using a generative model and an inference model. The generative likelihood of a single image observation is modelled with a spatial Gaussian mixture [41, 11]:

$$p_\theta(x^t|\mathbf{z}^t = \{z_k^t\}, v^t) = \prod_{i=1}^{M}\sum_{k=1}^{K} p_\theta(C_i^t = k|z_k^t) \cdot \mathcal{N}(x_{k,i}^t;\, g_\theta(z_k^t, v^t), \sigma^2\mathbf{I}), \tag{2}$$

where i indexes a pixel location (M in total) and RGB values (e.g. $x_{k,i}^t$) that pertain to an object k are sampled from a Gaussian distribution $\mathcal{N}(x_{k,i}^t;\, g_\theta(z_k^t, v^t), \sigma^2\mathbf{I})$ whose mean is determined by the decoder network $g_\theta(\cdot)$ (defined in Sec. 2) with trainable parameter θ and standard deviation σ is globally set to a fixed value 0.1 for all pixels. The mixing coefficients $p_\theta(C_i = k|z_k)$ capture the categorical probability of assigning a pixel i to an object k (i.e. $C_i = k$). This imposes a competition over the K objects as every pixel has to be explained by one and only one object in the scene.

DyMON adapts the *cross-view inference module* [32] of MulMON to handle: i) the cross-view object correspondence problem, ii) recursive approximation of a factorized posterior, and iii) temporal evolution of spatial structures (which indicates the major difference between the inference modules of DyMON and MulMON). The decomposition and recursive approximation of the posterior is:

$$p(\mathbf{z}^t = \{z_k^t\}|x^{\leqslant t}, v^{\leqslant t}) \approx q_\Phi(\mathbf{z}^t = \{z_k^t\}|x^{\leqslant t}, v^{\leqslant t}) = q(\mathbf{z}^0)\prod_t q_\Phi(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{<t}), \tag{3}$$

where $q_\Phi(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{<t})$ denotes the approximate posterior to a subproblem w.r.t. an observation x^t taken from viewpoint v^t at time t, and assumes a standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for the scene prior $q(\mathbf{z}^0)$. The intuition is to treat a posterior inferred from previous observations as the new prior to perform Bayesian inference for a new posterior based on a new observation. We use $\mathbf{z}^t$ to denote the inferred scene representations after observing x^t, i.e. a new posterior, and $\mathbf{z}^{<t}$ to denote the new prior before observing x^t. Note that we can advance t either regularly or irregularly. The single-view (or within-view) inference is handled by DyMON using *iterative amortized inference* [28] with amortization function Φ (modelled with neural networks). Refer to Appendix B. for full details about the generative and inference models of DyMON.

3.3 Training

To enable DyMON to learn independently the generative relationships between scenes and observations, and viewpoints and observations during training, built upon MulMON's architecture, we break a long moving-cam-dynamic-scene sequence into short sub-sequences (see Algo. 1) where sampling $(x^{t'}, v^{t'}) \sim (x^t, v^t)_{1:T}$ independently of $\mathbf{z}^t$ is possible. Similar to MulMON [32], we then train DyMON by maximizing the following objective function that linearly combines an evidence lower bound (abbr. ELBO) and the log likelihood (abbr. LL) of the querying views:

$$\begin{aligned}\mathcal{L} =& \boldsymbol{ELBO} \;+\; \beta \cdot \boldsymbol{LL}_{query}\\ =& \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbf{E}_{q_\Phi(\mathbf{z}^t|\cdot)}[\log p_\theta(x^t|\mathbf{z}^t, v^t)] - \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathcal{D}_{\mathrm{KL}}[q_\Phi(\mathbf{z}^t|x^{\leqslant t}, v^{\leqslant t})||q_\Phi(\mathbf{z}^{<t}|x^{<t}, v^{<t})]\\ &+ \beta \cdot \frac{1}{|\mathcal{T}|\cdot|\mathcal{Q}|}\sum_{t\in\mathcal{T}}\sum_{t_q\in\mathcal{Q}} \mathbf{E}_{q_\Phi(\mathbf{z}^t|\cdot)}[\log p_\theta(x^q|\mathbf{z}^t, v^q)],\end{aligned} \tag{4}$$

where $\mathcal{T}$ and $\mathcal{Q}$ record the times when DyMON performs inference and v^t interventions (i.e. viewpoint-queried generation) and β is the weighting coefficient. We construct $\mathcal{T}$ by sampling t (either regularly or irregularly) with a random walk through $[1, T]$, where a uniform distribution $\mathcal{U}\{\Delta t - 2, \Delta t + 2\}$ of an expected value Δt (> 2) is used as the step distribution. As shown in

Algo. 1, by varying the updating periods of $\mathbf{z}^t$ and v^t (denoted as $\Delta t_{\mathbf{z}}$ and Δt_v respectively), DyMON imitates the behaviours of a *multi-view-static-scene* model and a *single-view-dynamic-scene* model to handle the SCFO and FCSO samples respectively. In addition, using different β for the SCFO and FCSO samples allows alternating the training focus between spatial reasoning (w.r.t. objects and viewpoints) and temporal updating.

Algorithm 1: DyMON Training Algorithm

```
Input: training data $\mathcal{D}$
Hyperparameters $|\mathcal{Q}|$, $(\beta_{FCSO}, \beta_{SCFO})$, $(\Delta t, \Delta\tau)$ ; // $\Delta t > \Delta\tau > 2, |\mathcal{Q}| = \textbf{sizeof}(\mathcal{Q})$
Initialize trained parameters $\Phi$, $\theta$, and latent prior $\boldsymbol{\lambda}^0 = \{(\mu_k = \mathbf{0}, \sigma_k = \mathbf{I})\}$;
repeat
    Sample a sequence $\mathbf{X} = \{(x^t, v^t)\}_{1:T} \sim \mathcal{D}$ ;    // $T$ (RGB images, viewpoints)
    if assign($\mathbf{X}; \mathcal{D}$) == $FCSO$ then
        $\beta$, $\Delta t_v$, $\Delta t_{\mathbf{z}}$ = $\beta_{FCSO}$, $\Delta\tau$, $\Delta t$ ;    // $\Delta t_v < \Delta t_{\mathbf{z}}$, update $v^t$ more often
    else
        $\beta$, $\Delta t_v$, $\Delta t_{\mathbf{z}}$ = $\beta_{SCFO}$, $\Delta t$, $\Delta\tau$ ;    // $\Delta t_{\mathbf{z}} < \Delta t_v$, update $\mathbf{z}^t$ more often
    $\mathcal{T}$ = random_walk_t(s $= 1$, e $= T$, step_dist $= \mathcal{U}\{\Delta t_{\mathbf{z}} - 2, \Delta t_{\mathbf{z}} + 2\}$) ;
    $(x, v)$, $t$, $\boldsymbol{\lambda}^t$, $\boldsymbol{ELBO}$, $\boldsymbol{LL}_{query}$, = $\mathbf{X}[1]$, 1, $\boldsymbol{\lambda}^0$, 0, 0;
    while $t \leqslant T$ do
        $(x^t, v^t) = \mathbf{X}[t]$ ;
        if mod($t, \Delta t_v$) == 0 then
            $v = v^t$ ;    // update $v$
        if $t \in \mathcal{T}$ then
            $x = x^t$ ;    // update $x$
            $\boldsymbol{ELBO}^{(t)}$, $\boldsymbol{\lambda}^t$ = iterative_inference$_{\Phi,\theta}(x, v, \boldsymbol{\lambda}^t)$ ;
            $\mathbf{z^t} \sim \mathcal{N}(\mathbf{z}^t; \boldsymbol{\lambda}^t)$ ;    // sample updated $\mathbf{z}^t$
            $\mathcal{Q} = \{t_q\}$ = sample_query_t(dist $= \mathcal{U}\{t - \Delta t_{\mathbf{z}}/2,\ t + \Delta t_{\mathbf{z}}/2\}$, size $= |\mathcal{Q}|$);
            for $t_q \in \mathcal{Q}$ do
                $(x^q, v^q) = \mathbf{X}[t_q]$;
                $\boldsymbol{LL}_{query} += (1/(|\mathcal{Q}| \cdot |\mathcal{T}|)) \cdot \log p_\theta(x^q|\mathbf{z}^t, v^q)$ ;    // fix $\mathbf{z}^t$, do $v = v^q$
            $\boldsymbol{ELBO} += (1/|\mathcal{T}|) \cdot \boldsymbol{ELBO}^{(t)}$;
        $t += 1$;
    $\mathcal{L} = \boldsymbol{ELBO} + \beta \cdot \boldsymbol{LL}_{query}$ ;    // $\beta_{FCSO} > \beta_{SCFO}$
    $\theta$, $\Phi \leftarrow$ optimizer$_{\mathbf{max}}(\mathcal{L}, \theta, \Phi)$;
until $\theta, \Phi$ converge;
```

Assignment Function and Batching As the samplers of T and Q behave differently for SCFO and FCSO data (see Algo. 1), we need to determine if a $\mathbf{X} \sim \mathcal{D}$ is an SCFO sample or an FCSO sample. Under **A2**, we consider any dataset consisting of only a mix of SCFO and FCSO samples (where a sample is a sequence of images). For a given dataset, we cluster all training samples of a dataset into two clusters w.r.t. the SCFO and FCSO scenarios. This then gives us an assignment function, **assign**$(\mathbf{X}; \mathcal{D})$ (as shown in Algo. 1. In practice, to avoid breaking parallel training processes with loading SCFO and FCSO samples into the same batch, we assign the training data beforehand instead of assigning every data sample on the fly during training. This allows to batch FCSO or SCFO samples independently at every training step.

4 Related Work

Single-View-Static-Scene The breakthrough of unsupervised object discovery based on a primary scenario, i.e. a single-view-image setting, lays a solid foundation for the recent rise of unsupervised object-centric representation learning research. Built upon a VAE [22], early success was shown by AIR [8] that searches for one object at a time on image regions. Because AIR and

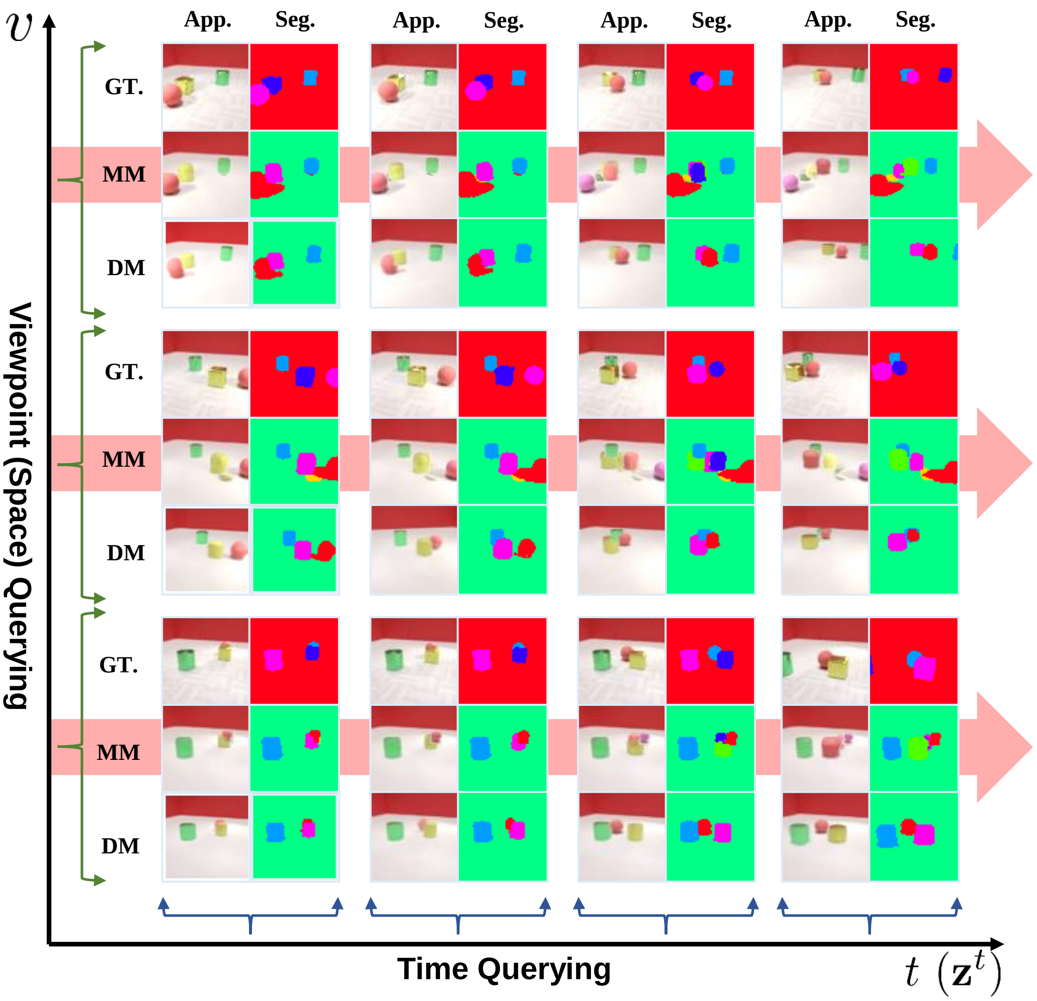

Figure 2: Qualitative results of spatial-temporal factorization. The GT rows show the true scene. The “MM” and “DM” entries are the scene re-rendered from the corresponding models, i.e. MulMON and DyMON respectively. The vertical row pairs show the results from viewpoint changes and the horizontal direction shows the results at different times. Note that we train MulMON and DyMON on different datasets as MulMON cannot train on multi-view-dynamic-scene datasets. We also visualize MulMON’s tendency of generating degenerated results along the temporal direction (marked with red arrows).

most of its successors (e.g. [23]) treat objects as flat pixel patches and the image generation process as “paste flat objects on canvas” using a spatial transformer [17], they often cannot summarize well scene spatial properties that are suitable for 3D manipulation: for example, they do not render smaller objects when the objects are “moved” further away from the camera. To overcome this, most recent advances [3, 10, 25, 7, 27, 6] model a single 2D image with a spatial Gaussian mixture model [41, 11] that allows explicit handling of background and occlusions. Although these models suffer from single-view ambiguities like occlusions or optical illusions, they have the potential for attaining factorized representations of 3D scenes. Our work has close relationship to IODINE [10]: we handle the object-wise inference from an image observation at each time point using the *iterative amortized inference* [28] design and capture the compositional generative process with a spatial Gaussian mixture model.

Multi-View-Static-Scene A natural way of resolving single-view ambiguities is to aggregate information from multi-view observations. Although multi-view scene explorations do not directly facilitate object-level 3D scene factorization, Eslami et al. [9] demonstrated that they do reduce the spatial uncertainty and enable explicit 3D knowledge evaluation—novel-view prediction. As combining GQN [9] and IODINE [10], Nanbo et al. [32] showed that MulMON effectively leverages multi-view exploration to extract accurate object representations of 3D scenes. However, like GQN, MulMON can only train on static-scene samples and thus does not generalize well to dynamic scenes ROOTS [5] combines GQN and AIR's merits to perform multi-view-static-scene object-centric representation learning whereas it requires camera intrinsic parameters to overcome AIR's deficiency of 3D scene learning — it is thus camera-dependent hence less general. In our work, we propose DyMON as an extension of MulMON to dynamic scenes and a unified model for unsupervised multi-view object-centric representation learning.

Single-View-Dynamic-Scene A line of unsupervised scene object-centric representation learning research was established on the *single-view-dynamic-scene* setting [14, 23, 19], where they explicitly model and represent object dynamics based on video observations. However, as most of these works employ a similar image composition design to AIR, they deal with only flat 2D objects that are similar to MNIST digits and thus cannot model 3D spatial properties. A closely-related work is that of Lin et al. [24], i.e. GSWM, where they modelled relative depth information and pair-wise interactions of 3D object patches. In our work, the spatial-temporal factorization allows us to show the dynamics and depths of the objects from different viewpoints at different times.

Other Related Work As a *multi-view-dynamic-scene* representation learning framework, T-GQN [37] represents the most closely-related work to ours. It models the spatial representation learning at each time step as a stochastic process (SP) and transitions between these time-stamped SPs with a state machine. However, a notable distinction between the problems that T-GQN and DyMON are targeting based on that: 1) T-GQN does not attain object-level scene factorization and 2) a typical T-GQN requires multi-view observations at each time step (as so-called "context") to perform spatial learning so as to get rid of the *temporal entanglement* problem (which has been the core focus of our work). Our work is essentially dealing with disentangled representation learning problems, which are often formulated under the frameworks of causal inference [34, 38, 36] and *independent component analysis* (abbr. ICA) [16, 15]. Unlike traditional disentanglement representation learning works (e.g. [13, 21, 26]) that aims at feature-level disentanglement, in this work, we handle not only the object-level disentanglement that resides in the object-centric representation learning research, but also the time-dependent scene-observer disentanglement problem.Recent trend of neural radiance field (e.g. [30, 29, 35]) are relevant to our work in the sense of 3D scene representations using multi-view images. However, from an *vision-as-inverse-graphics* [43] perspective, we do not consider them scene understanding models as they only aim to memorize the volumetric structure of a single scene during "training" thus cannot perform representation inference for unseen scenes.

5 Experiments

We used two simulated *multi-view-dynamic-scene* synthetic datasets, namely DRoom and MJC-Arm, and a real-world dataset, namely CubeLand (see Appendix C.3 for details), in this work. We conducted quantitative analysis on DRoom and show qualitative results on the other two datasets. The DRoom dataset consists of five subsets (including both training and testing sets): one subset (denoted as DR0-$|\overline{f_{\mathbf{z}}}|$) with zero object motion (*multi-view-static-scene* data), one subset (denoted as DR0-$|\overline{f_v}|$) with zero camera motion (*single-view-dynamic-scene* data), and three *multi-view-dynamic-scene* subsets of increasing speed difference levels from 1 to 3 (denoted as DR-Lvl.1 $\sim$ 3). Each of the five subsets consists of around 200 training sequences (40 frames of RGB images per sequence) and 20 testing sequences (40 frames from 12 different views, i.e. 57.6k images). Although DyMON's focus is on a more general problem, we nevertheless compare it against two recent and specialized unsupervised object-centric representation learning methods, i.e. GSWM [24], and MulMON [32], in two respective settings: *single-view-dynamic-scenes*, and *multi-view-static-scenes.* All models were trained with 3 different random seeds for quantitative comparisons. Refer to our supplementary material for full details on experimental setups, and ablation studies and more qualitative results.

5.1 Space-Time Querying

The recovery of *the independent generative mechanism* permits DyMON to make both viewpoint-queried and time-queried predictions, i.e. querying across space and times, of scene appearances and segmentations using the inferred scene representations, which enables the below two demonstrations:

Novel-view Prediction at Arbitrary Times Recall that a scene observation x is the generative product of a specific scene (composed by objects) and observer at a specific time t with a well-defined generative mapping, i.e. $x = g(\mathbf{z}, v)$ (see sec. 2). Like previous multi-view object-centric representation learning models (e.g. MulMON [32]), we query from an arbitrary viewpoint v w.r.t. a scene of interest $\mathbf{z}$ by fixing $\mathbf{z}$ and manually setting the viewpoint v to arbitrary configurations. Similarly, we can query about the spatial state of a dynamic scene at time t from a specific viewpoint by fixing the viewpoint and manually inputting z^t at arbitrary times t to the generative function. We trained a DyMON on the DR-Lvl.3 data and show qualitatively the prediction results that are queried by space-time tuples in Figure 2.

Dynamics Replay of Scenes & Objects From Arbitrary Viewpoints In this experiment, we give DyMON a sequence of image observations of a dynamic scene as input, and have it replay the dynamics from a novel viewpoint using the scene representations it infers from the observations. This is done by fixing the v to the desired values and querying about consecutive times. As the inferred scene representations are factorized in terms of objects, we show in Figure 3 (left) that, besides the complete scene dynamics, DyMON also allows to replay the dynamics of a single object independently of the others. We present the qualitative results on the MJC-Arm datasets in Figure 3 (right) where one can see that DyMON not only replays object dynamics as global position changes, it also captures object local motions.

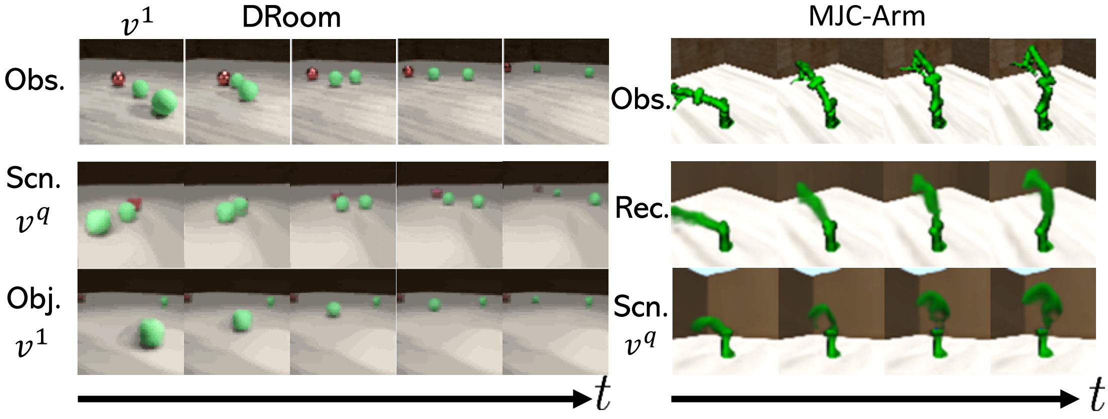

Figure 3: **Left:** DyMON performing dynamics replays on the DRoom dataset, where the first row is the observation sequence input to DyMON, second and third rows show replays of the scene dynamics (all objects' original motions) and object dynamics (just the foreground green ball moves) respectively from an arbitrary viewpoint v^q. **Right:** DyMON replays local motions of robot arm from an arbitrary viewpoint (top: observation, middle: reconstruction, bottom: replay from a higher viewpoint).

Dynamics On Real-World Data To demonstrate that our model has the potential for real-world applications, we conduct experiments and show qualitative results on real images (i.e. CubeLand data). We refer the readers to Appendix D.4 for the results.

5.2 Versatile Evaluation

DyMON is designed to handle object-centric representation learning in a general setting—*multi-view-dynamic-scenes*. In this section, we experiment to evaluate how well DyMON handles the specialized settings.

	MSE↓		mIoU↑	
Models	Obs.Rec.	Nv.Obs.	Obs.Seg.	Nv.Seg.
MulMON	0.011 ± 0.001	$\mathbf{0.019 \pm 0.002}$	0.511 ± 0.001	0.461 ± 0.062
DyMON	$\mathbf{0.004 \pm 0.001}$	0.021 ± 0.002	$\mathbf{0.717 \pm 0.000}$	$\mathbf{0.508 \pm 0.065}$

(a) DyMON vs. *Multi-View-Dynamic-Scenes*

	MSE↓		mIoU↑	
Models	Obs.Rec.	Nv.Obs.	Obs.Seg.	Nv.Seg.
MulMON	$\mathbf{0.006 \pm 0.001}$	$\mathbf{0.012 \pm 0.005}$	$\mathbf{0.583 \pm 0.080}$	$\mathbf{0.538 \pm 0.105}$
DyMON	0.014 ± 0.001	0.019 ± 0.007	0.529 ± 0.005	0.506 ± 0.105

(b) DyMON vs. *Multi-View-Static-Scenes*

	MSE↓	mIoU↑
Models	Obs.Rec.	Obs.Seg.
GSWM	0.039 ± 0.007	0.402 ± 0.082
DyMON	$\mathbf{0.014 \pm 0.011}$	$\mathbf{0.682 \pm 0.107}$

(c) DyMON vs. *Single-View-Dynamic-Scenes*

Table 1: Quantitative comparisons of DyMON and two baseline models, i.e. GSWM and MulMON, in handling scenarios that the baseline models are specialized at. The models in table (a) are trained and tested on the DR0-$|\overline{f_v}|$ data, and those in (b) and (c) are trained and tested on the DR0-$|\overline{f_\mathbf{z}}|$ data. "Obs." tags reconstructions and segmentations that are computed for the observations and "Nv." tags those from novel viewpoints. Mean ± stddev for 3 training seeds. ↑ indicates higher is better and ↓ indicates the opposite.

DyMON vs. Dynamic Scenes We first evaluate DyMON's performance in the *multi-view-dynamic-scene* setting in comparison to MulMON. MulMON also learns the *independent generative mechanism* of scene objects and observer, but with a strict static-scene constraint. Note that both DyMON and MulMON permit novel-view predictions of scene appearances and segmentations, this allows explicit quantification of the correctness and accuracy of the inferred scene representations. We use a mean-squared-error (MSE) measure and a mean-intersection-over-union score (mIoU) measure. We train DyMON on the DR-Lvl.3 subset and MulMON on the DR0-$|\overline{f_\mathbf{z}}|$ subset (because it is UNABLE to train on dynamic-scene data) and conduct comparison across the three DRoom dynamic-scene subsets (i.e. DR-Lvl.1 $\sim$ 3). Table 1a shows that, although we train MulMON on a more strict dataset, i.e. the DR0-$|\overline{f_\mathbf{z}}|$ dataset, DyMON still outperforms MulMON on almost all the various indicators. We show the qualitative comparison results in Figure 2 and observe that MulMON's performance declines along the temporal axis when large object motions appear. As neither DyMON nor MulMON impose any orders for object discovery, we used the Hungarian matching algorithm to find the best match that maximizes the mIoU score to handle the bipartite matching between the output and the Ground-truth masks.

DyMON vs. Static Scenes We evaluate how well it handles *multi-view-static-scene* scenarios in comparison with a specialized model, i.e. MulMON. We train and test both DyMON and MulMON on the DR0-$|\overline{f_\mathbf{z}}|$ subset w.r.t. reconstructions and segmentations of both the observed and unobserved views. Table 1b summarizes the results. They show that DyMON can handle this strict constraint setting, even though it exhibits a slight performance gap compared with the specialized model. Also, it is worth noting that DyMON and MulMON produce high variances in segmentations. One possible reason is that both MulMON and DyMON employ stochastic parallel inference mechanisms that can sometimes infer duplicate latent representations and harm segmentations [33]. This experiment along with the **DyMON-versus-dynamics-scenes** experiment provides useful guidance for model selection in multi-view applications—use a specialized model in a well-controlled environment and DyMON to handle complex scenarios.

DyMON vs. Fixed-View Observations of Dynamic Scenes We assessed DyMON's performance on handling *single-view-dynamic-scene* observations by comparing it with GSWM [24], which is a specialized object-centric representation model for this specific setting, although it is unable to achieve pixel-level segmentation. We train both DyMON and GSWM on the DR0-$|\overline{f_v}|$

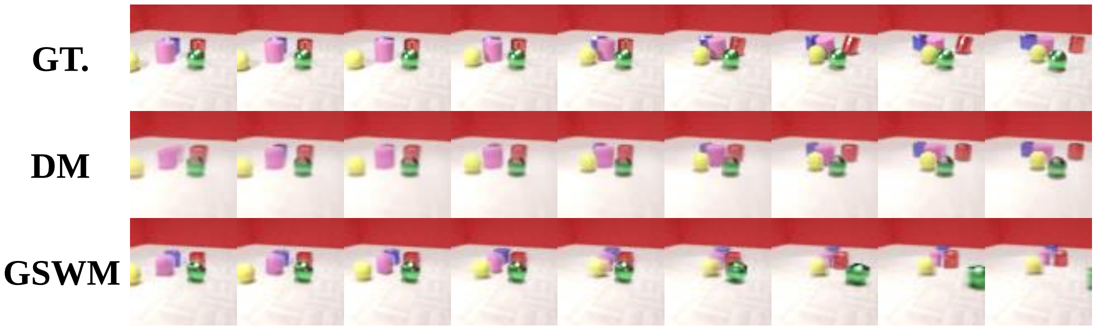

Figure 4: **Left:** Qualitative comparisons of DyMON and GSWM on reconstructing the DR0-$|\overline{f_v}|$ scenes. The GT rows show the actual observations of a dynamic scene, and the "DM" and "GSWM" rows show observation reconstruction results of DyMON and GSWM, respectively.

subset and measure the reconstruction quality of the observations. Table 1c shows that DyMON not only outperforms GSWM in observation reconstruction, but it also permits pixel-wise segmentation which the specialized model cannot. The qualitative results in Figure 4 show that GSWM learns better object appearances (especially for textures) than DyMON, whereas DyMON learns more accurate scene dynamics than GSWM. This is understandable as GSWM models object dynamics explicitly, which introduces risks of overfitting the observed motions. DyMON supports well temporal interpolations, i.e. dynamics replays (as shown in Figure 3 and 4), but it does not model the object dynamics nor interactions explicitly like GSWM. As a result, it does not provide readily extrapolatable features along the time (or dynamics) axis for predicting into the future.

DyMON vs. T-GQN T-GQN [37] is a closely related work as it targets unsupervised scene representation learning in the multi-view-dynamic-scene settings, even though it does not attain object-centric factorization in the latent space. Although T-GQN requires multi-view observations at each time step (as "context" information) to sidestep the temporal entanglement issue, we nevertheless train it on our DRoom data and show that it fails to represent the DRoom scenes (see Appendix D.3 for the results and discussions).

6 Conclusion

We have presented Dynamics-aware Multi-Object Network (DyMON), a method for learning object-centric representations in a *multi-view-dynamic-scene* setting. We have made two weak assumptions that allows DyMON to recover the *independent generative mechanism* of observers and scene objects from both training and testing *multi-view-dynamic-scene* data—achieving *spatial-temporal factorization.* This permits querying the predictions of scene appearances and segmentations across both space and time. As this work focuses on representing the spatial scene configurations at every specific time point, i.e. DyMON does not model dynamics explicitly so it cannot predict the future evolution of scenes, which leaves space for future exploration.

Acknowledgements

The first authors would like to acknowledge the School of Informatics, the University of Edinburgh for providing his PhD scholarship. This research is partly supported by the *Trimbot2020* project, which is funded by the European Union Horizon 2020 programme. The authors would like to thank Prof. C. K. I. Williams and Cian Eastwood for valuable discussions.

References

[1] CLEVR Blender Environment, BSD licencse. `https://github.com/facebookresearch/clevr-dataset-gen`. Accessed: 2021-06-02.

[2] DeepMind MultiObject Dataset, Apache-2.0 licencse. `https://github.com/deepmind/multi_object_datasets`. Accessed: 2021-06-02.

[3] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[4] D. Carlos, A. Cohen, and M. L. Littman. An object-oriented representation for efficient reinforcement learning. In *International Conference on Machine Learning*, pages 240–247, 2008.

[5] C. Chen, F. Deng, and S. Ahn. Object-centric representation and rendering of 3d scenes. *JMLR*, 2021.

[6] M. Engelcke, O. P. Jones, and I. Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *arXiv preprint arXiv:2104.09958*, 2021.

[7] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2019.

[8] S. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, pages 3225–3233, 2016.

[9] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[10] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2424–2433, 2019.

[11] K. Greff, S. Van Steenkiste, and J. Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6691–6701, 2017.

[12] K. Greff, S. van Steenkiste, and J. Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

[14] J.-T. Hsieh, B. Liu, D.-A. Huang, L. F. Fei-Fei, and J. C. Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018.

[15] A. Hyvarinen and H. Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. In *Advances in Neural Information Processing Systems*, 2016.

[16] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[18] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu. Reasoning about physical interactions with object-oriented prediction and planning. In *International Conference on Learning Representations*, 2019.

[19] M. Jaques, M. Burke, and T. Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations*, 2020.

[20] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[21] H. Kim and A. Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658, 2018.

[22] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[23] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, pages 8606–8616, 2018.

[24] Z. Lin, Y.-F. Wu, S. Peri, B. Fu, J. Jiang, and S. Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning*, 2020.

[25] Z. Lin, Y.-F. Wu, S. V. Peri, W. Sun, G. Singh, F. Deng, J. Jiang, and S. Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2019.

[26] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[27] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, 2020.

[28] J. Marino, Y. Yue, and S. Mandt. Iterative amortized inference. In *International Conference on Machine Learning*, pages 3403–3412, 2018.

[29] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021.

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[32] L. Nanbo, C. Eastwood, and R. B. Fisher. Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems*, 2020.

[33] L. Nanbo and R. B. Fisher. Duplicate latent representation suppression for multi-object variational autoencoders. In *BMVC*, 2021.

[34] J. Pearl et al. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[35] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. *arXiv preprint arXiv:2011.13961*, 2020.

[36] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[37] G. Singh, J. Yoon, Y. Son, and S. Ahn. Sequential neural processes. In *Advances in Neural Information Processing Systems*, 2019.

[38] R. Suter, D. Miladinovic, B. Schölkopf, and S. Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *International Conference on Machine Learning*, pages 6056–6065. PMLR, 2019.

[39] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[40] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.

[41] C. K. I. Williams and M. K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, 2004.

[42] J. Yang, J. Mao, J. Wu, D. Parikh, D. D. Cox, J. B. Tenenbaum, and C. Gan. Object-centric diagnosis of visual reasoning. *arXiv preprint arXiv:2012.11587*, 2020.

[43] A. Yuille and D. Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 2006.

[44] A. Zadaianchuk, M. Seitzer, and G. Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations*, 2021.

A. Algorithms

A.1 Iterative inference algorithm

Algorithm 2: Iterative Inference Algorithm

```
Input: observation $x$, viewpoint $v$, latent Gaussian parameters $\boldsymbol{\lambda}^t = \{(\mu_k^t, \sigma_k^t)\}$
ModelParameters $\Phi, \theta$, and the number of single-view iterations $L(default : 5)$
Initialize $\boldsymbol{\lambda}^{t(l)} = \boldsymbol{\lambda}^t$, $\boldsymbol{ELBO}^t = 0$
for $l = 1$ to $L$ do
    $\mathbf{z}^{\mathbf{t(l)}} \sim \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)})$ ;           // sample from a prior--make a guess
    $p_\theta(x^{t(l)}|\mathbf{z}^{t(l)}, v) = g_\theta(\mathbf{z}^{\mathbf{t(l)}}, v)$ ;                  // render and verify
    $\boldsymbol{ELBO}^{t(l)} = -\log p_\theta(x^{t(l)}|\mathbf{z}^{t(l)}, v) + \mathcal{D}_{\mathrm{KL}}(\mathcal{N}(z^t; \boldsymbol{\lambda}^{t(l)})||\mathcal{N}(z^t; \boldsymbol{\lambda}^t))$ ;
    $\boldsymbol{\lambda}^{t(l)} = \Phi(x, \boldsymbol{ELBO}^{t(l)}, \boldsymbol{\lambda}^{t(l)})$ ;       // refine and then repeat (until $l = L$)
    $\boldsymbol{ELBO}^t += (1/L) \cdot \boldsymbol{ELBO}^{t(l)}$
Output $\boldsymbol{ELBO}^t, \boldsymbol{\lambda}^{t(l)} = \{(\mu_k^{t(l)}, \sigma_k^{t(l)})\}$
```

A.2 Testing algorithm

Algorithm 3: DyMON Testing Algorithm

```
Input: Trained parameters $\Phi$, $\theta$, and latent Gaussian parameters
       $\boldsymbol{\lambda}^0 = \{(\mu_k = \mathbf{0}, \sigma_k = \mathbf{I})\}$
Initialize $\boldsymbol{\lambda}^t = \boldsymbol{\lambda}^0$ ;
while Access $(x^t, v^t)$ do
    $\boldsymbol{ELBO}^t$, $\boldsymbol{\lambda}^t = \mathbf{iterative\_inference}_{\Phi,\theta}(x^t, v^t, \boldsymbol{\lambda}^t)$ ;
    Output $\boldsymbol{\lambda}^t = \{(\mu_k^t, \sigma_k^t)\}$ ;
```

B. Implementation Details

B.1 Training configurations

We show the training configurations used in this work in Table 2.

Table 2: Training Configurations

Type	the trainings of DyMON, MulMON, GSWM
Optimizer	Adam
Initial learning rate η_0	$3e^{-3}$
Learning rate at step s	$\max\{0.1\eta_0 + 0.9\eta_0 \cdot (1.0 - s/1e^6), 0.1\eta_0\}$
Total gradient steps	$300k$ for DyMON vs. GSWM, $200k$ for DyMON vs. MulMON
Batch size	2 $(2\,seqs \times 40\,images = 80\,images)$
number of GPU/per training	1 $(Mem >= 11GB)$

* the same scheduler as the original GQN except for faster attenuation

B.2 Model implementation

We show the designs of the generative mapping function g_θ and the refinement function in Table 3 and 4 respectively. After obtaining a set of K RGBM outputs from this function, i.e. $\{(\mu_{xk}, \hat{m}_{xk})\}$ (see Table 3), we render (i.e. compose) an image as: $x = \sum_k \mathbf{softmax}(\hat{m}_{xk}) \cdot x_k$, where $x_k \sim \mathcal{N}(x_k;\, \mu_{xk}, 0.1^2\mathbf{I})$,

Table 3: Generator function g_θ

Parameters	Type	Channels (out)	Activations.	Descriptions
	Input	$D + d$		$\mathbf{z}^t \sim \mathcal{N}(\mathbf{z}^t;\, \boldsymbol{\lambda}^t)$, v^t
θ^1 (projection)	Linear	256	Relu	
	Linear	D	Linear	$\tilde{\mathbf{z}}^t = g_{\theta_1}(\mathbf{z}^t, v^t)$
	Input	D		$\tilde{z}_k^t = g_{\theta_2}(z_k^t, v^t)$
θ^2 (rendering)	Broadcast	D+2		* Broadcast to grid
	Conv 3×3	32	Relu	
	Conv 3×3	32	Relu	
	Conv 3×3	32	Relu	
	Conv 3×3	32	Relu	
	Conv 3×3	4	Linear	RGBM: rgb μ_{xk} + mask logits $\hat{m}_{xk}$

D: the dimension of a latent representation, set to 16 for all experiments

d: the dimension of a viewpoint vector, set to 3 for all experiments

*: see spatial broadcast decoder [40]

Stride= 1 set for all Convs.

Table 4: Refinement Network Φ

Parameters	Type	Channels (out)	Activations.	Descriptions
	Input	17		* Auxiliary inputs $\mathbf{a}(x^t)$
Φ	Conv 3×3	32	Relu	
	Conv 3×3	32	Relu	
	Conv 3×3	64	Relu	
	Conv 3×3	64	Relu	
	Flatten			
	Linear	256	Relu	
	Linear	128	Linear	
	Concat	128+4*D		
	LSTMCell/GRUCell	128		
	Linear	128	Linear	output $\Delta\lambda$

D: the dimension of a latent representation, set to 16 for all experiments

Stride= 1 set for all Convs.

* see IODINE[10] for details

LSTMCell/GRUCell channels: the dimensions of the hidden states

C. Datasets

C.1 DRoom (DynamicRoom)

Simulation Environment We created the DRoom simulation on the top of the CLEVR Blender environment [20, 1]. Like other multi-object datasets [2], we initialized every sequence by randomly selecting and placing 2-5 scene objects in a simulated room (with background and walls specified). These objects are randomized in terms of shapes (incl. deformations, sizes), colors, and textures. Under the Blender physics engine settings, we enabled foreground objects' movements by setting their dynamics status to "active" and disabled the background objects' (i.e. walls and ground's) movements by setting their dynamics status to "passive". We then created a centrifugal force field within a fixed center and range on the ground across all DRoom datasets. In this work, we sample the magnitude of the force using: **random.choice**(vals $= 8500 \times \{0, 0.1, 0.2, ..., 1\}$, probs $= \mathcal{C}at(...)$), which allows us to simulate scene object motions of different speeds by inputing different selection categorical probability $\mathcal{C}at(...)$. Moreover, we enabled object collisions to simulate scenes with rather complex object dynamics. The control of the observer (an RGB camera) motion is independent of the scene objects. We consider an observer or camera performing random walks on the surface of a dome (top half of a sphere) whose center aligns with the center of the ground—we randomly initialize the starting position of a camera and randomly sample its next move. Note that as the camera can only move on the dome (with a fixed radius r), we can use azi and ele, i.e. the azimuth and elevation of the camera, to represent a camera location. We sample the increment Δazi and Δele independently from: **random.choice**$_{azi}$(vals $= 5.0\,degs \times \{0, 0.1, 0.2, ..., 1\}$, probs $= \mathcal{C}at_{azi}(...)$) and **random.choice**$_{ele}$(vals $= 1.0\,degs \times \{0, 0.1, 0.2, ..., 1\}$, probs $= \mathcal{C}at_{ele}(...)$), which suggests that we can control the speed of the camera by inputting different $\mathcal{C}at_{azi}(...)$) and $\mathcal{C}at_{ele}(...)$).

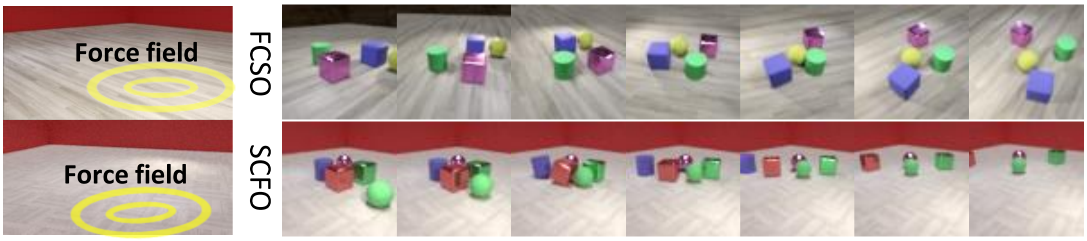

Figure 5: **Left:** DRoom simulation environment setup where yellow rings denote the force fields. **Right:** One *fast-camera-slow-object* (FCSO) sample (**top row**) and *slow-camera-fast-object* (SCFO) sample (**bottom row**). Both are randomly selected from the DR-Lvl.3 dataset.

Dataset We rendered all scenes using a resolution of 64×64 for 40 frames (4-second motions)—record 40 images with their corresponding viewpoints $\{(x^t, v^t)\}_{1:40}$, where we represent the viewpoints using their 3-D Cartesion coordinates. The sampler specifications, i.e. the categorical distributions $\mathcal{C}at(...)$), used to generate the five DRoom subsets are listed in Table 5. As discussed in Sec.3.3, we clustered all the data samples based on their average camera speeds across each sequence to assign them into the FCSO and SCFO partitions. We visualize the clustering results for DR-Lvl.1 $\sim$ 3 in Figure 6

Table 5: DRoom Generator Specs

Subsets		Force Magnitude (constant in its range)	Camera Random Walk Next Move (for both azi and ele)
DR0-$\lvert\overline{f_{\mathbf{z}}}\rvert$	—	$\{1, 0, 0, ..., 0\}$	$\{0, 0, 0, 0, 0, 0, 0.01, 0.11, 0.28, 0.3, 0.3\}$
DR0-$\lvert\overline{f_v}\rvert$	—	$\{0, 0, 0, 0, 0, 0.02, 0.08, 0.15, 0.35, 0.35, 0.05\}$	$\{1, 0, 0, ..., 0\}$
DR-Lvl.1	FCSO	$\{0.05, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095\}$	$\{0.05, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095\}$
	SCFO	$\{0.05, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095\}$	$\{0.05, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095, 0.095\}$
DR-Lvl.2	FCSO	$\{0.2, 0.2, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0, 0, 0.2, 0.2, 0.2, 0.2, 0.2\}$
	SCFO	$\{0, 0, 0, 0, 0, 0.2, 0.2, 0.2, 0.2, 0.2, 0\}$	$\{0.2, 0.2, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0, 0\}$
DR-Lvl.3	FCSO	$\{0.25, 0.38, 0.33, 0.02, 0.02, 0, 0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0, 0, 0.01, 0.11, 0.28, 0.3, 0.3\}$
	SCFO	$\{0, 0, 0, 0, 0, 0.02, 0.08, 0.15, 0.35, 0.35, 0.05\}$	$\{0.3, 0.3, 0.28, 0.11, 0.01, 0, 0, 0, 0, 0, 0\}$

C.2 MJC-Arm (Mujoco-Arm)

Simulation Environment The environment is built with MuJoCo physics simulator [39], and the Franka Emika robot arm with a Barret hand attached it the main scene object. The arm has 7 degrees of freedom and the joints of robotic hand are fixed during the data generation. 8 different collision-free robot arm motion trajectories are pre-defined, and each has unique initial and target joint configuration. Every joint is controlled in the position-derivative manner with a constant velocity, which is the product of the nominal velocity and the sampled weight. The nominal velocities for all 7 arm joints (from base to end-effector) are $[0.65, 0.65, 0.27, 0.27, 0.03, 0.03, 0.005]$, which are related to the link lengths of the robot arm. The joint velocity weights for FCSO and SCFO data trials are sampled from **random.choice**$_{FCSO}(\{0, 0.1, 0.2, ..., 1\}$, probs $= \{0.34, 0.34, 0.25, 0.07, 0.0, ..., 0.0\})$ and **random.choice**$_{SCFO}(\{0, 0.1, 0.2, ..., 1\}$, probs $= \{0.0, ..., 0.0, 0.07, 0.25, 0.34, 0.34\})$. We also introduced a moving ball with random fixed direction and constant weighted velocity in the simulation. The control of the RGB camera is the same as introduced in the former section, with a fixed point of view towards the base link of the robot arm.

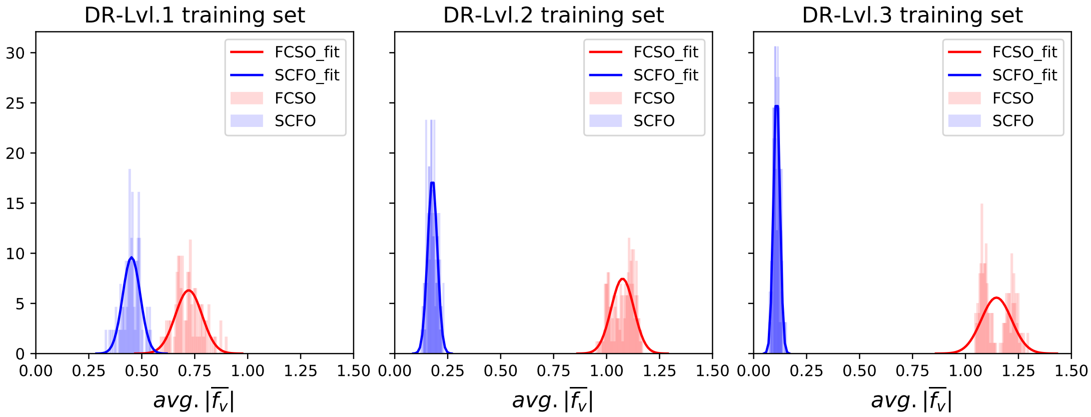

Figure 6: Visualization of the data assignment results on the DR-Lvl.1 $\sim$ 3 datasets.

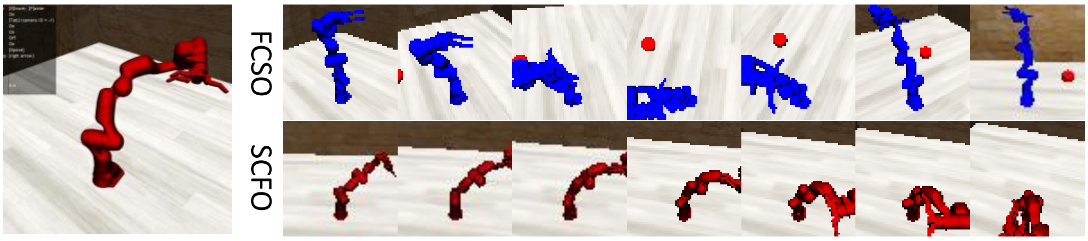

Figure 7: **Left:** Mujoco simulation environment. **Right:** One *fast-camera-slow-object* (FCSO) sample (**top row**) and *slow-camera-fast-object* (SCFO) sample. Both are randomly selected from the MJC-Arm dataset.

Dataset For each data sample, the scenes are rendered with resolution 64 × 64 at 10Hz for 4 seconds (40 frames per sample). At the beginning of every trail, the textures of the robot arm and the moving ball are randomly selected from a colour set. The robot arm is initialised with the starting pose of the randomly selected motion trajectory.

C.3 Real-world aataset (CubeLand)

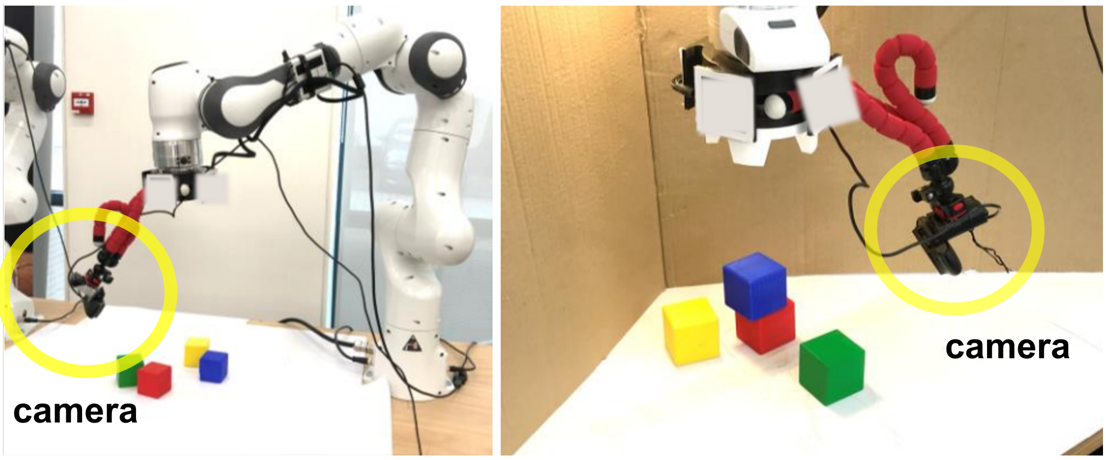

Figure 8: CubeLand data-collection platform.

Data-collection Environment We created CubeLand in a controlled real-world environment. Four cubes of different colours (i.e., red, blue, green and yellow) were placed on a table. To avoid glare, reflections and unnecessary background clutter, the surface of the table was made white by designing a bicolour data collection environment. A camera was mounted on the end effector of Franka (a robotic arm with 7 D.O.F.) as shown in Figure 8. The end effector has a fixed motion, i.e., it only rotates back and forth 120 degrees with no translation motion involved. The cubes were taped with threads at the bottom to move them freely and randomly. Moreover, the simulations had two configurations, i.e., slow camera, fast objects (SCFO) and fast camera, slow objects (FCSO) (see Figure 9). In the first configuration, the speed of the rotation of the end effector was 1.67 rpm (10 degrees per second) while the objects were manually pulled and thrown back into the scene at an arbitrary faster speed. In the latter configuration, the speed of the rotation of the end effector was set to be 4.17 rpm (25 degrees per second) whereas the objects were pulled and pushed by hand back into the scene at a slower rate. The height of the camera is 14.5 cm and the radius this assembly (centre of the end effector to the camera) spans is 19.5 cm.

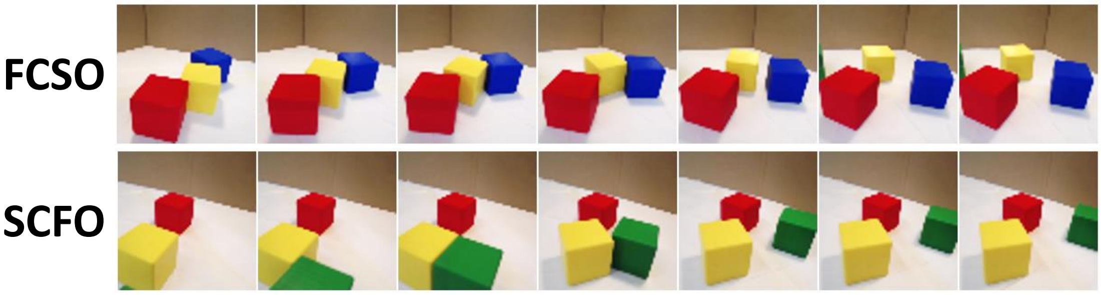

Figure 9: CubeLand data samples. **Top:** a fast camera, slow objects (FCSO) data sample. **Bottom:** a slow camera, fast objects (SCFO) data sample.

Dataset All the frames collected were initially 480x480. During the post-processing steps, these frames were resized to 64x64 after applying a median filter (the centre of the kernel is replaced by the median of all the neighbouring pixels) of kernel size 9. Overall, 100 sequences of 50 frames each were extracted. Furthermore, each of the viewpoints was converted into 3D cartesian coordinates. The classification between SCFO and FCSO is solely based on the rotations per minute of the end effector.

D. Additional Results

D.1 Assumption Validation

As discussed in Sec.3.1. of the main paper, the training of DyMON on *multi-view-dynamic-scene* is based upon two assumptions that favor high frame-rate image sequences and large difference between the speeds of an observer and scene objects. In this experiments, as we know that the average speed differences of DR-Lvl.1 $\sim$ 3 are in an ascending order, we can thus assess the robustness of DyMON against our assumptions. We trained DyMON on the DR-Lvl.1 $\sim$ 3 training sets respectively and then evaluated their performance on space-time-queried prediction of scene appearances on the DR-Lvl.1 $\sim$ 3 test sets. We visualize the MSE as a function of increased levels of speed differences in Figure 10. As shown, 1) there is no significant performance drops across different training and test sets, 2) the faster the observer speeds and the scene speeds, the better the models perform. This holds for both training (see the overall performance on the left Figure) and testing (see a model's testing performance against different test sets on the right Figure). These supports our claims about DyMON's robustness against complex and potentially dynamic environments.

D.2 Ablation Study

We highlight two hyperparameters that play significant roles in the training of DyMON: 1) the updating periods of v and $\mathbf{z}$, i.e. Δt_v and $\Delta t_{\mathbf{z}}$, 2) weighting coefficient of viewpoint-queried generative log likelihood β. We varied these two groups of parameters and visualized their influences on DyMON—similar to Sec.D.1, we measure DyMON's novel-view synthesis performance at every time point and visualize them as a function of these hyperparameters. We varied $\Delta t_{\mathbf{z}}$ and Δt_v with values that are selected from discrete sets $\{3, 5\}$ and $\{5, 6, 8\}$, this allows us to show the joint effects of these two updating periods in a 2×3 grid (see **top** half of Figure 11). To analyze the independent effects of $\Delta t_{\mathbf{z}}$ and Δt_v, we "squeezed"

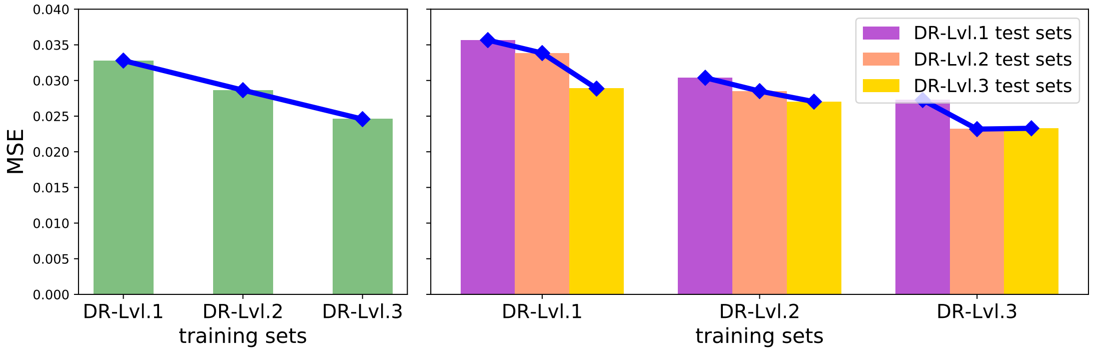

Figure 10: The space-time-queried scene appearance prediction performance comparison between three DyMONs that are trained on three levels of scene-observer speed differences, i.e. DR-Lvl.1 $\sim$ 3, respectively. **Left:** Averaged MSE achieved by the three models on three DRoom testing sets, i.e. the testing sets of DR-Lvl.1 $\sim$ 3. **Right:** The performance of the three models on each of the three testing sets.

the 2×3 grid by computing the MSE averaged over the $\Delta t_{\mathbf{z}}$ axes and Δt_v axes of the grid (see bottom right two plots of Figure 11 for the results). One can see that a short updating period for $\Delta t_{\mathbf{z}}$ is preferred as this allows to capture more detailed scene object motions, while the selection of Δt_v is relative subtler. One might run pre-analysis before training, e.g. visually look several sequences, to select a better Δt_v. Similarly, we varied β by setting its values to 0.5, 1.0, and 2.0 respectively and we show the results in the bottom left of Figure 11.

D.3 T-GQN Results

We used the official implementation of T-GQN (`https://github.com/singhgautam/snp`) and trained a T-GQN on the DR-Lvl.3 data. Although the training has converged (see Figure 12), we observe that it fails to represent the underlying 3D scenes (see Figure 13) and training T-GQN with a posterior dropout, i.e. T-GQN-PD, does not fix the issue. We speculate that this is because it lacks multiple views at each time steps to resolve the temporal entanglement issue. However, future investigations are required to validate our speculation.

D.4 Additional Qualitative Results

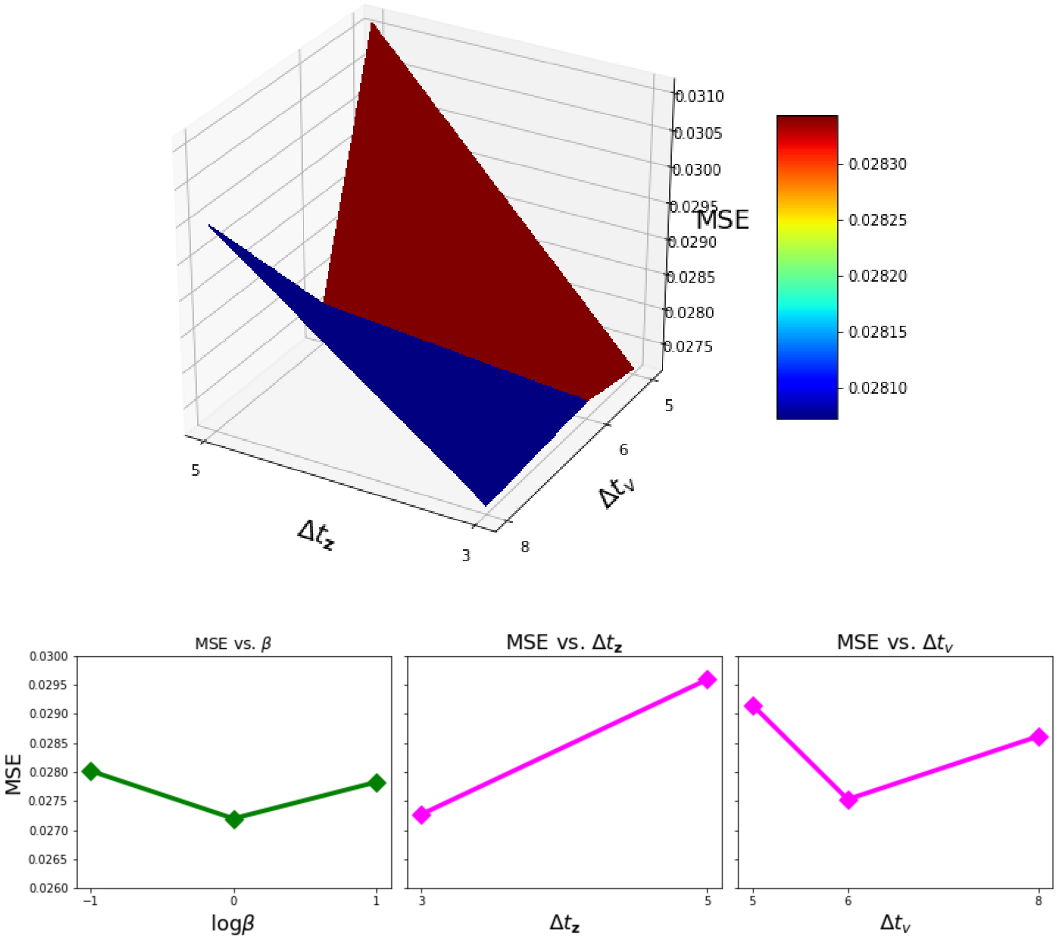

Figure 11: Ablation study results. **Top:** Space-time queried view synthesis MSE vs. nested $\Delta t_{\mathbf{z}}$ and Δt_v. **Bottom left:** MSE vs. different β (in $\log_2$ space). **Bottom middle:** MSE vs. different $\Delta t_{\mathbf{z}}$ (MSE computed by averaging across different Δt_v). **Bottom right:** MSE vs. different Δt_v (MSE computed by averaging across different $\Delta t_{\mathbf{z}}$).

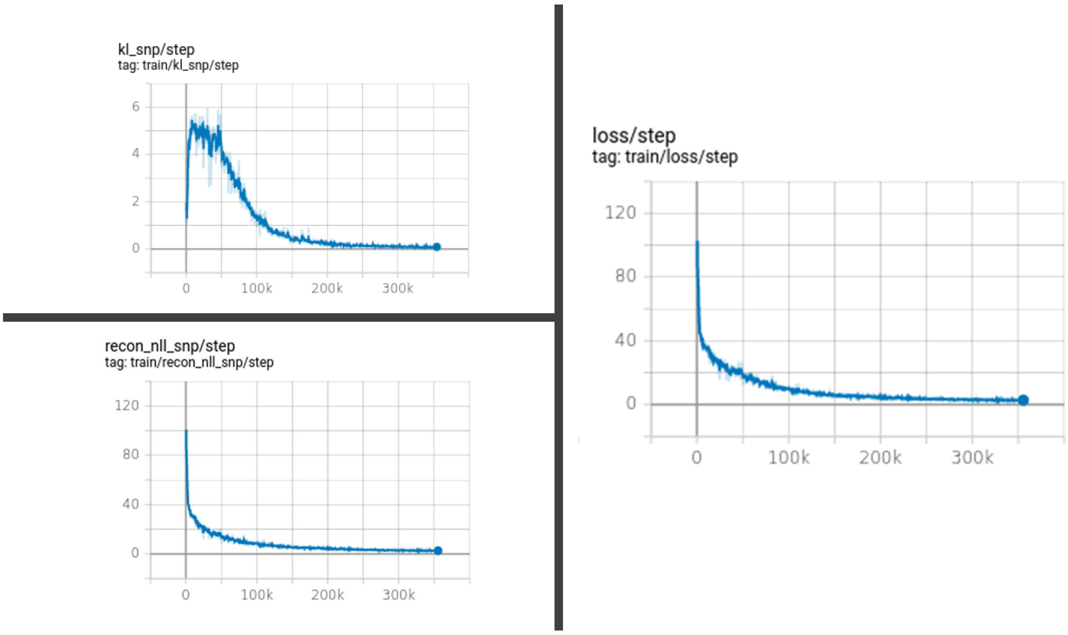

Figure 12: T-GQN training curves. We train t-GQN on our DRoom data until it converges.

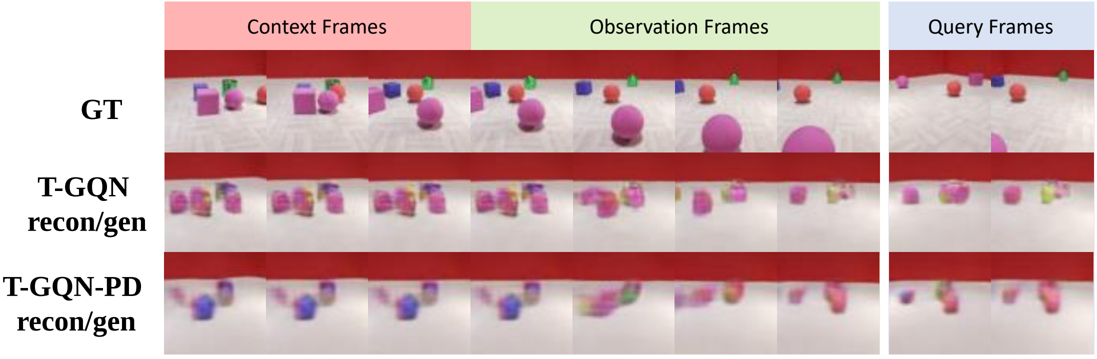

Figure 13: Qualitative results of T-GQN on DR-Lvl.3 test data.

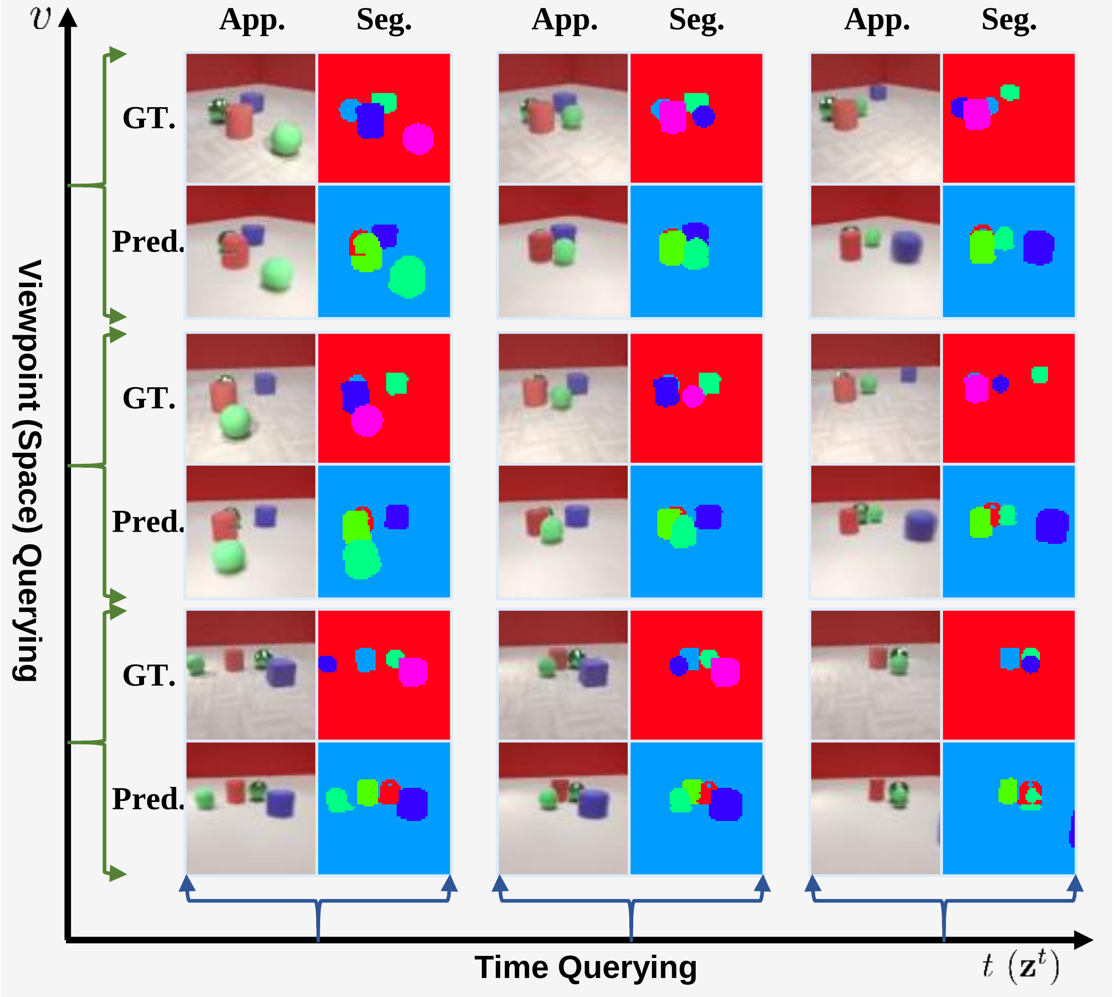

Figure 14: Spatial-temporal factorization results of a DRoom scene.

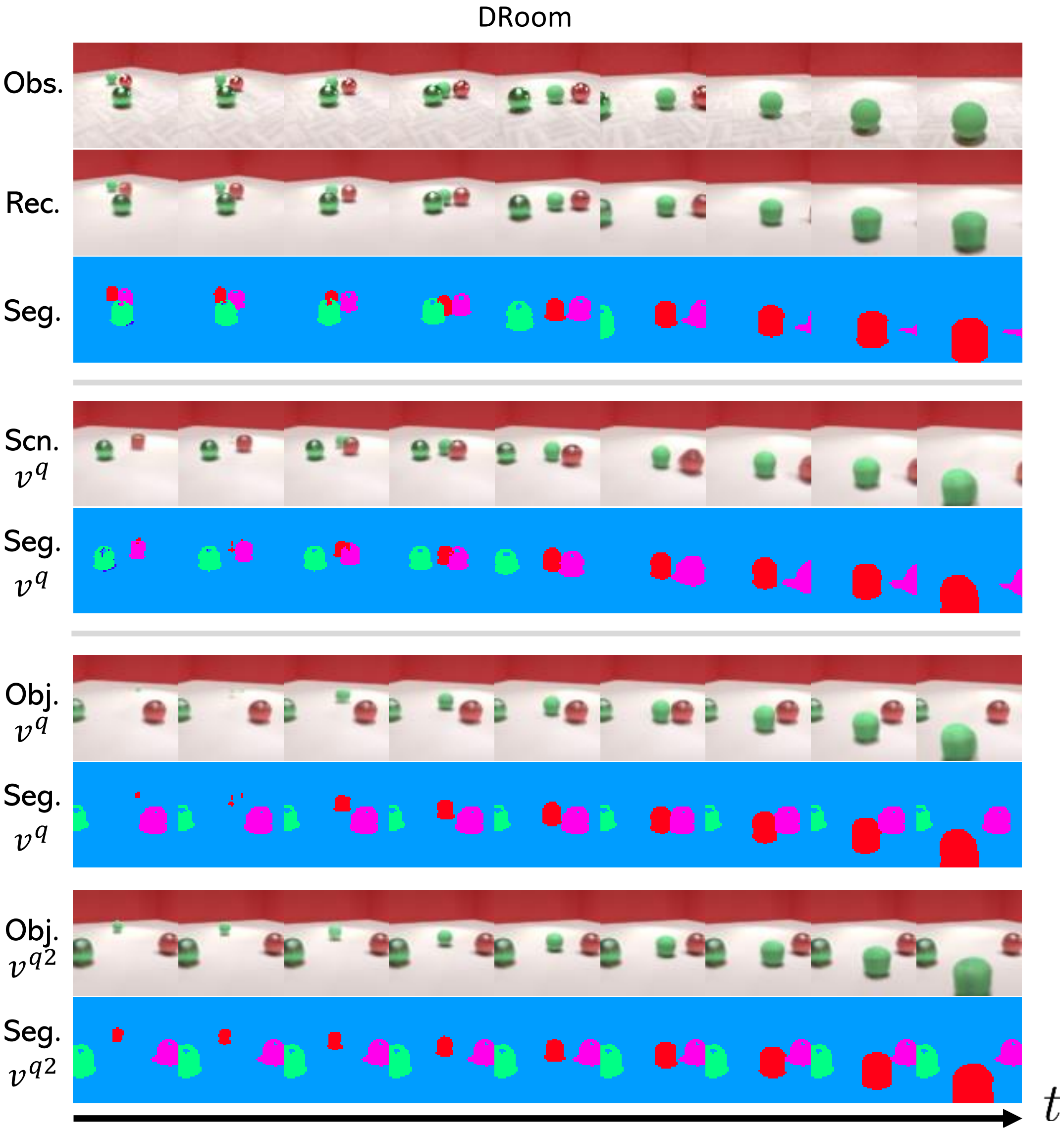

Figure 15: Dynamics replay of a DRoom scene.

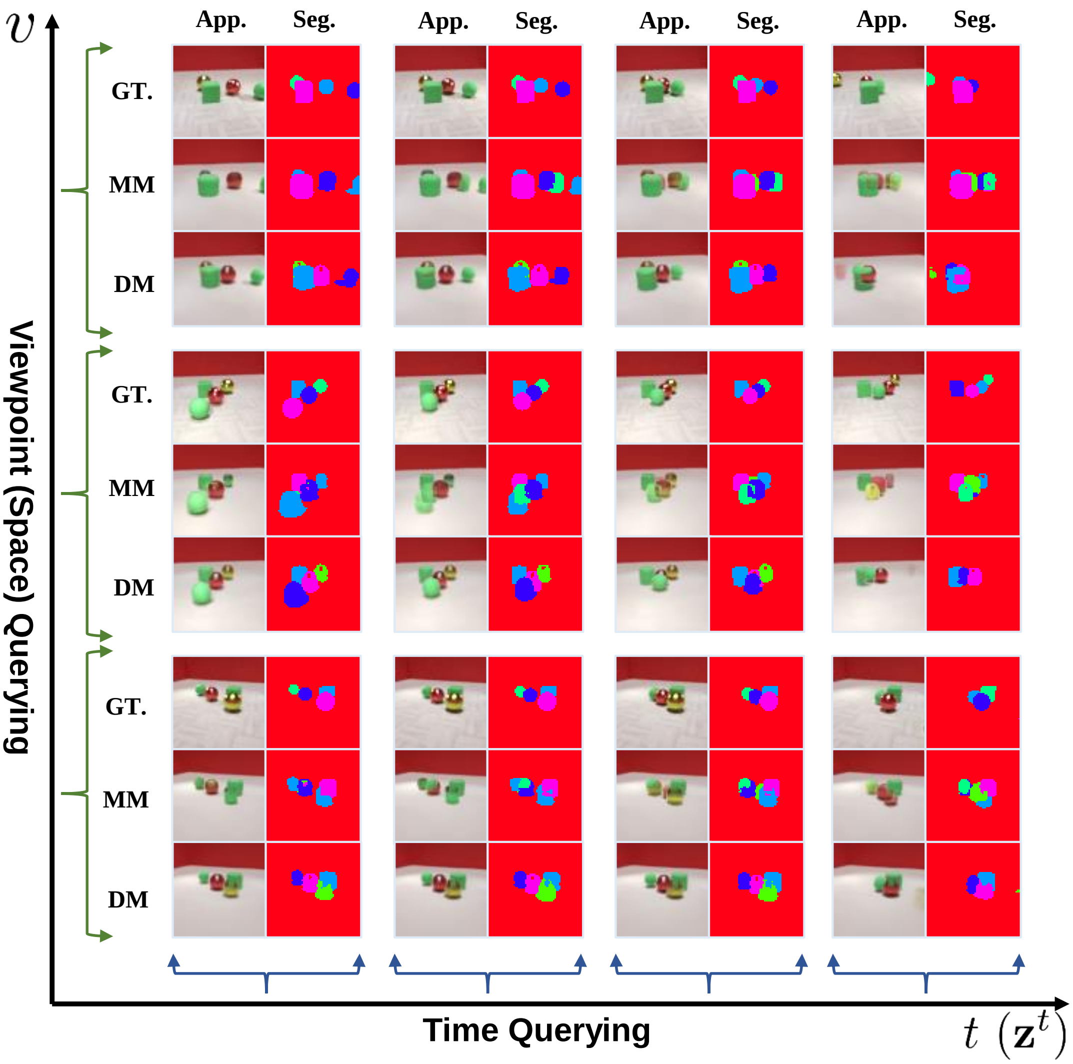

Figure 16: Qualitative comparisons of DyMON and MulMON on DRoom. **Left:** reconstruction performance. **Right:** spatial-temporal factorization performance. We train DyMON on DR-Lvl.3 and train MulMON on DR0-$|\overline{f_{\mathbf{z}}}|$.

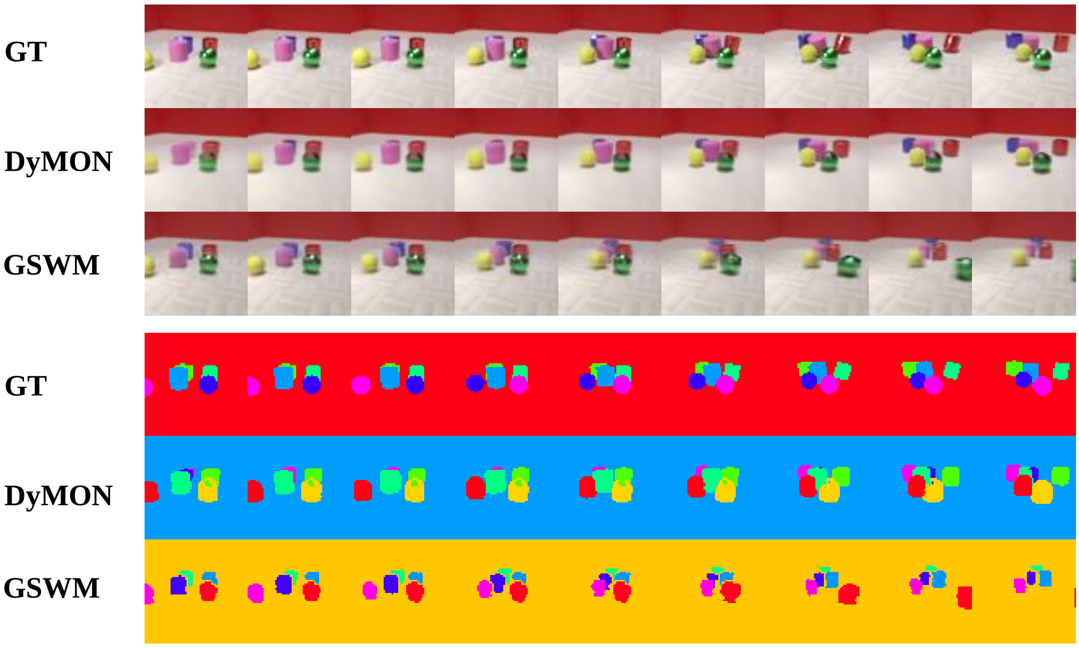

Figure 17: Qualitative comparisons of DyMON and GSWM on DR0-$|\overline{f_v}|$. **Top:** reconstruction performance. **Bottom:** segmentation performance (we observe that DyMON outperforms GSWM in segmenting scenes).

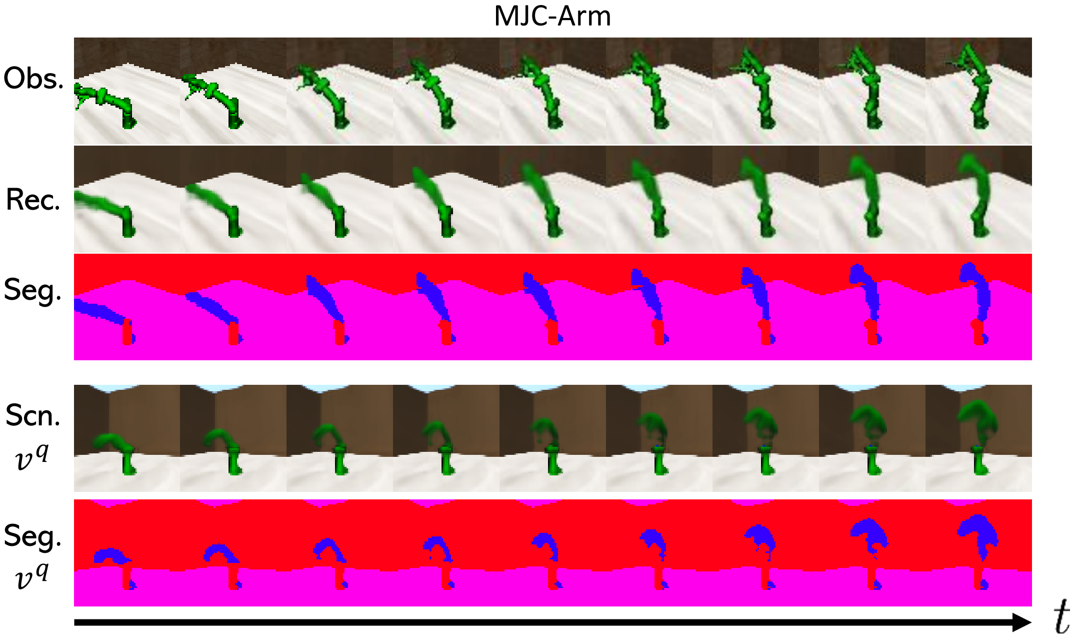

Figure 18: Dynamics replay of a MJC-Arm scene.

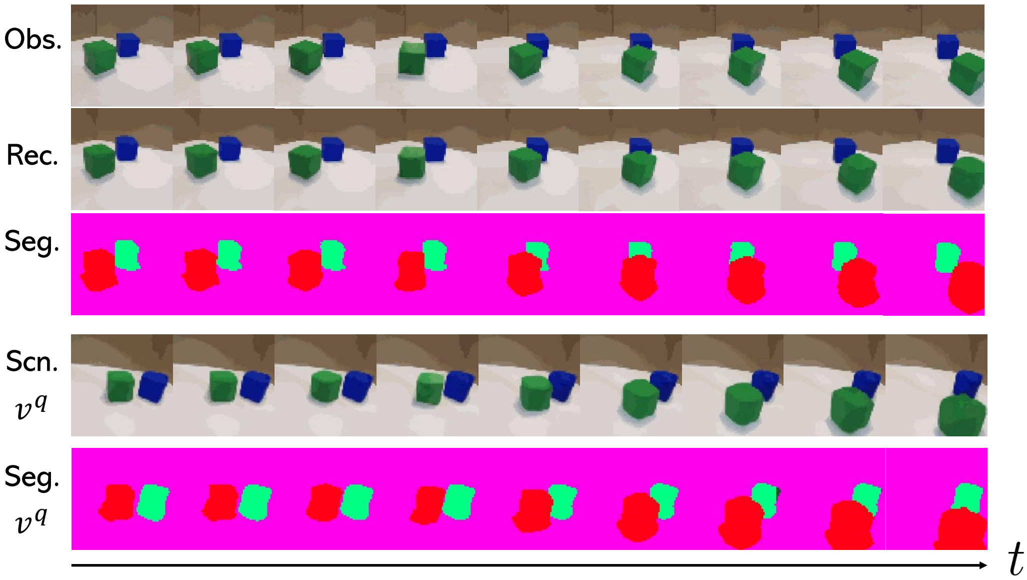

Figure 19: Dynamics replay of a real scene (i.e. CubeLand data). We conduct experiments on real-world data to show DyMON's potential for real-world applications.