Generative Factorization For Object-Centric Representation Learning

Nanbo Li

Doctor of Philosophy

Institute of Perception, Action and Behaviour

School of Informatics

University of Edinburgh

2022

Abstract

Empowering machines to understand compositionality is considered by many (Lake et al., 2017; Lake and Baroni, 2018; Schölkopf et al., 2021) a promising path towards improved *representational interpretability* and *out-of-distribution generalization*. Yet, discovering the compositional structures of raw sensory data requires solving a *factorization* problem, i.e. decomposing the unstructured observations into modular components. Handling the *factorization* problem presents numerous technical challenges, especially in unsupervised settings which we explore to avoid the heavy burden of human annotation. In this thesis, we approach the factorization problem from a generative perspective. Specifically, we develop unsupervised machine learning models to recover the *compositional data-generation mechanisms around objects* from visual scene observations.

First, we present MulMON as the first feasible unsupervised solution to the multi-view *object-centric representation learning* problem. MulMON resolves the spatial ambiguities arising from single-image observations of *static scenes*, e.g. optical illusions and occlusion, with a multi-view inference design. We demonstrate that not only can MulMON perform better scene object factorization with less uncertainty than single-view methods, but it can also predict a scene's appearances and object segmentations for novel viewpoints. Next, we present a technique, namely for *latent duplicate suppression* (abbr. LDS), and demonstrate its effectiveness in fixing a common scene object factorization issue that exists in various unsupervised object-centric learning models—i.e. inferring duplicate representations for the same objects. Finally, we present DyMON as the first unsupervised learner that can recover object-centric compositional generative mechanism from *moving-view-dynamic-scene* observational data. We demonstrate that not only can DyMON factorize dynamic scenes in terms of objects, but it can also factorize the entangled effects of observer motions and object dynamics that function independently. Furthermore, we demonstrate that DyMON can predict a scene's appearances and segmentations at arbitrary times (querying across time) and from arbitrary viewpoints (querying across space)—i.e. answer counterfactual questions.

The scene modeling explored in this thesis is a proof of concept, which we hope will inspire: **1)** a broader range of downstream applications (e.g. "world modelling" and environment interactions) and **2)** generative factorization research that targets more complex compositional structures (e.g. complex textures, multi-granularity compositions).

Lay Summary

This thesis explores the idea of empowering machines to understand compositionality, i.e. how to form complex expressions with a set of simpler expressions. The understanding of compositionality plays a significant role in human cognition systems—it allows humans to discover, from direct experiences, a set of reusable modules and the rules used to efficiently combine or even re-combine them to explain or generate new experiences. However, for artificial intelligence systems, finding the compositional structures from raw sensory input is challenging as it requires overcoming a technical obstacle—*factorization*, i.e. decomposing unstructured observations, often complex data like images or videos, into smaller meaningful entities.

We focus on the scenarios of *object-centric* visual compositionality understanding, i.e. treating scenes as compositions of objects, and tackle the problem of factorizing scenes into a set of explanatory objects. Instead of treating the factorization problem like the traditional *object detection* and *image segmentation* methods, we approach it from a generative perspective: we build machine learning models that recover the observation-generating process from data *without human supervision.*

In this thesis, we investigate three common issues regarding *object-centric factorization.* First, as the factorization of 3D scenes based on single-view 2D images often gives high spatial uncertainty (due to, e.g. occlusions or optical illusions), we present a method (MulMON) that uses multiple viewpoints of a scene to reduce factorization uncertainty and improve accuracies. Next, we present a technique (LDS) that reduces the chance of factorizing the same objects repeatedly by reasoning about the relations between the inferred object representations. Finally, we present a method (DyMON) that alleviates the influence of moving observers in scenarios where both the observer and the objects are moving simultaneously—by disentangling the effects of independent object motions and observer motions.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. The work published within this thesis has been published in the following peer-reviewed articles with attribution and contribution as follows:

> **Li Nanbo**, Cian Eastwood, and Robert Fisher. "Learning Object-Centric Representations of Multi-Object Scenes From Multiple Views" *Advances in Neural Information Processing Systems*, 2020.
>
> **Li Nanbo** and Robert Fisher. "Duplicate Latent Representation Suppression For Multi-Object Variational Autoencoders." *The British Machine Vision Conference*, 2021.
>
> **Li Nanbo**, Muhammad Ahmed Raza, Hu Wenbin, Zhaole Sun, and Robert Fisher. "Object-Centric Representation Learning with Generative Spatial-Temporal Factorization." *Advances in Neural Information Processing Systems*, 2021.

Contribution:

- **Nanbo**: proposed and implemented ideas and theories, conducted experiments, and wrote papers.
- Eastwood: helped with disentanglement evaluations.
- Raza, Wenbin, Sun: helped with dataset creation.
- Fisher: helped with verification of the ideas and theories, experiments, and writing.

(*Nanbo Li*)

Acknowledgements

I would like to thank my supervisor, Prof. Robert Fisher (Bob), for his meticulous guidance and support throughout my PhD study. Most importantly, I always know that my personality can sometimes make me a difficult student to deal with, but Bob has shown me his exceptional patience and leniency—which helped me survive the "suffering" times I had in the first two years of my PhD. I am genuinely lucky to have Bob as my supervisor and friend.

I would also like to thank Prof. Chris Williams (my second supervisor), who introduced me to generative models and representation learning. It has been rather fun working on generative models regardless of the ongoing debates in the community on how far generative models could lead us on the path of pursuing artificial general intelligence.

I am grateful to Cian Eastwood, Wenbin Hu, Can Pu, Muhammad Ahmed Raza, Zhaole Sun, and Chuanyu Yang (listed in alphabetical order) for their help and support as my friends and collaborators. I wish all these fantastic people and all my other friends the brightest future.

I would like to thank the School of Informatics at The University of Edinburgh for providing a PhD scholarship and for research support by the Trimbot2020 project, which was funded by the European Union Horizon 2020 programme.

Last but not least, I want to thank my family and my partner for their unending love, and great mental accompaniment. Things have been difficult for us since the historical breakout of the COVID-19 pandemic. I hereby dedicate this thesis to them, even though it is truly *nothing* compared to what they have provided.

Table of Contents

Nomenclature

X,Y,Z,V	random variables (abbr. RVs); Z denotes latent variables
x,y,z,v	values of the corresponding random variables
$\mathbf{Z}=\{Z_1,...,Z_K\}$	a composition of K joint random variables
$\mathbf{z}=\{z_1,...,z_K\}$	a sample (i.e. K joint values) of $\mathbf{Z}$
$\mathbf{x}=\{x^{(1)},...,x^{(N)}\}$	a set of N *I.I.D.* samples of X; indices bracketed to emphasize *I.I.D.*
$\mathbf{D}$	data (set), observational
$P_X, P(X)$	probability distribution of X
$p(x)$	probability density/mass function of $P(X)$
$P(X\|\mathbf{Z})$	conditional probability distribution of X given $\mathbf{Z}$
$p(x\|\mathbf{z})$	conditional probability density/mass function of $P(X\|\mathbf{Z})$
$P(X\|\mathrm{do}(Z_k = z_k))$	interventional probability distribution of X under $\mathbf{Z}=\mathbf{z}$
$p_\theta(\cdot)$	a family of density/mass functions parameterized by θ
$q_\phi(\cdot)$	a family of approximating probability density/mass functions parameterized by ϕ
$g_\theta(\cdot)$	a family of deterministic functions parameterized by θ
$\mathbf{PA}_X$	a set of all parent random variables of X
$\mathbf{pa}_x$	values of X's parents
$X \perp\!\!\!\perp Y$	independence between random variables X and Y
$X \perp\!\!\!\perp Y\|\mathbf{Z}$	conditional independence between X and Y given $\mathbf{Z}$
$x \sim P_X$ or $p(x)$	draw samples from the distribution of X
$\mathbf{E}_{p(x)}[f(x)]$	expectation of a function w.r.t. a distribution
$\mathcal{D}_\star[q(\mathbf{z})\|\|p(\mathbf{z})]$	a divergence measure between two distributions, e.g. $\mathcal{D}_{\mathrm{KL}}[\cdot]$ is a Kullback-Leibler divergence
$W,\Psi,\Sigma,\mathbf{I}$	matrices; specified in the text to distinguish from RVs

M, D	dimensions; specified in the text to distinguish from RVs
$\mathbf{supp}(X)$	the support domain of variable X
$\mathbb{R}^D$	D-dimensional real number set
$\mathbb{N}^{[1,K]}$	natural numbers within range $[1, K]$

Chapter 1

Introduction

The beauty of nature lies in its diversity. Millions years of evolution have allowed humans to develop the cognitive capabilities necessary to appreciate, understand, and even create the world's diversity. It was suggested by Rosch and Mervis (1975); Smith and Osherson (1984); Biederman (1987); Kamp and Partee (1995) that these cognitive abilities of humans are centered around the understanding of **compositionality**[1]—i.e. being able to discover, from the direct experience, a set of reusable components and the rules used to efficiently combine or even re-combine them to explain or generate new experience. Many in the artificial intelligence community (Lake et al., 2017; Lake and Baroni, 2018; Schölkopf et al., 2021) believe that these findings could have profound implications for the development of artificial systems.

Artificial intelligence has achieved remarkable success in recent years while much of the success can be attributed to the progress of machine learning. Yet, most of the existing machine learning models, particularly artificial neural networks, are built upon the *independent and identically distributed* (abbr. *I.I.D.*) data assumptions. I.e. instead of uncovering the underlying compositional structures

[1] See *Principle of compositionality*: the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them (Grandy, 1990).

that generate the data, these models focus on constructing superficial statistical associations between the input and output variables. For example, if “grass” and “cows” always appear together in the training images, a “cow” detection model trained on those images could mistakenly detect “cow(s)” in a “grass” image even though there is no cow. Therefore, despite the remarkable expressiveness demonstrated by the existing deep models, they fall short of **i)** human-level **generalization** (e.g. handling out-of-distribution data) and **ii)** representational **interpretability**.

According to Greff et al. (2020), the key to systematically solving or improving these issues is enabling artificial systems to solve the *binding problem*, where the *segregation* (i.e. *factorization*) problem is the major challenge especially in an unsupervised setting. In the context of visual perception, established upon a principle of interpreting scenes as compositions of objects (i.e. the explanatory factors), ***object-centric scene representation learning*** **(abbr. OCRL)** has recently emerged as a promising approach towards improved visual scene interpretation, sample efficiency, and generalization for many down-stream applications like relational reasoning and control (Janner et al., 2019; Carlos et al., 2008; Bapst et al., 2019). In this line of research, the *factorization* goal of a model is to spatially decompose a scene into a set of interchangeable objects based on the scene observations.

The focus of this thesis is to develop machine learning models that handle the *factorization* problem from a generative perspective, i.e. learning, without supervision, the generative compositional structures from unstructured raw observational data. Specifically, we study the case of *object-centric scene compositions*. The following two sections (§1.1 & §1.2) provide an introduction of the *object-centric scene representation* setup and specifications of the *factorization* problem within such context, respectively.

1.1 Object-Centric Scene Representation

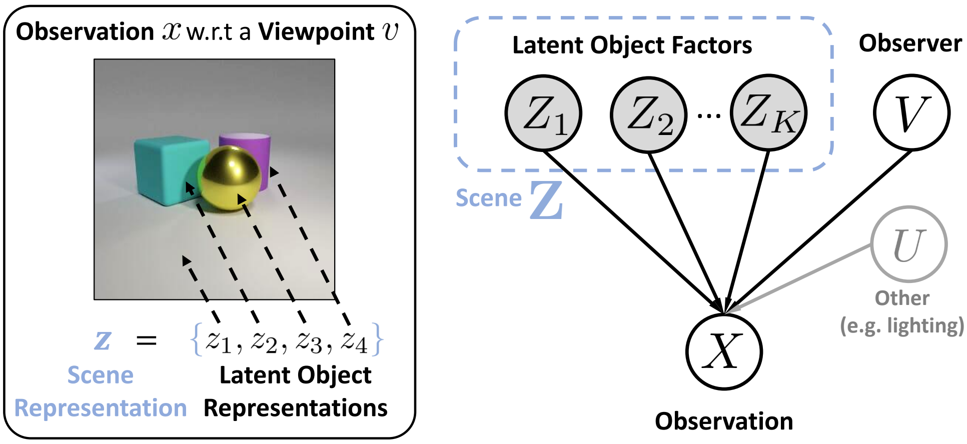


Figure 1.1: **Left:** An example ***object-centric*** *vision-as-inverse-generation* diagram—explaining a scene observation using an object-centric scene representation. **Right:** The *disentangled (causal) generative model*[2] constructed around independent generative factors—"objects" or "scene" (greyed: hidden from our view), "observer", and "other".

The idea of representing scenes as collections of underlying generative components (e.g. objects, see Figure 1.1) originates from the "*vision-as-Bayesian-inference*" paradigm that has been vastly studied in psychology, cognitive science, and artificial intelligence (Von Helmholtz, 1867; Minsky, 1988; Yuille and Kersten, 2006; Kulkarni et al., 2015b). Inspired by these studies, we consider object-centric scene representation inference as the inverse problem of a scene observation generation problem.

The Forward Problem Let $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$ (unobservable) denote scenes and each of its components, i.e. a multidimensional random variable Z_k, denote a scene object. Let X and V denote the sensory input observations and observer configurations respectively. Studying the forward (observation generation) problem is about discovering the causal relationships between these $\mathbf{Z}$, V, and X. We

[2] See Suter et al. (2019) for its general definition, here we consider a special case with confounders $\mathbf{C} = \emptyset$.

use the causal graph shown in Figure 1.1 (right) to describe these causal relationships, where, for simplicity, we consider the "other" factor (i.e. the U factor in the figure) uncorrelated noise and often ignore it in our discussions. In a forward pass, as shown by the example in Figure 1.1 (left), with a scene sample $\mathbf{z}$ well-defined by a set of objects, i.e. $\{z_k\}_{1:K} = \{z_1, z_2, ..., z_K\}$ (where each $z_k \in \mathbb{R}^D$ represents one and only one object in the scene including the background as a generalized object), and an independent observer specified as $v \sim P_V$, an observation (e.g. an image) $x \in \mathbb{R}^M$ (often $M \gg D$) can be taken using a specific compositional generative mapping g as $x = g(\mathbf{z}, v) = g(z_1, z_2, ..., z_K, v)$ or $x \sim p(x|\mathbf{z}, v)$ (which also captures the uncertainty introduced by the uncorrelated noise).

The Inverse Problem With the forward problem defined, we can describe the goal of learning an object-centric scene representation as inferring the intrinsic parameters of the objects of a scene, i.e. inferring $\mathbf{z} = \{z_k\}$ based on a given scene observation sample $x \sim P_X$. In the simplest case, where X represents single-view image observations and therefore we ignore the V variable (Burgess et al., 2019; Greff et al., 2019), the inverse problem can be defined as factorizing a posterior $p(\mathbf{z}|x) = p(z_1, z_2, ..., z_K|x)$ for an image x—we expect each z_k represents a different object. Although computing posteriors is generally intractable, approximate inference (Hoffman et al., 2013; Marino et al., 2018) is often feasible in practice. As the number of objects is unknown in the inverse problem, it is worth noting that **i)** K is often set globally to be a sufficiently large number (greater than the actual number of objects) to capture all scene objects, and **ii)** we allow empty or collapsed "slots".

1.2 Generative Factorization

Though we have previously introduced the general factorization goal of object-centric representation models as "spatially decompose a scene into a set of interchangeable objects based on the scene observations", it remains unclear how to train such models, especially in unsupervised settings.

In this thesis, we approach the *factorization* problem from a generative perspective (see §1.1) and have made two assumptions: **i)** the generative structure shown in Figure 1.1 (right) describes the underlying data generation process and **ii)** the generative factors $(Z_1, Z_2, ..., Z_K, V)$ within this model are jointly independent such that their joint distribution factorizes:

$$p(z_1, z_2, ..., z_K, v) = p(v) \prod_{k=1}^{K} p(z_k). \tag{1.1}$$

Note that, besides the statistical assumption, equation 1.1 also reflects the general *independent causal mechanism* (abbr. ICM) (Schölkopf et al., 2012; Peters et al., 2017) principle. ICM describes a more general property that permits independent interventions—changing one factor without affecting the others while allowing statistical dependence between the generative factors[3]. The property depicted by Eqn. 1.1 thus underpins the feasibility of learning and evaluating independent and modularized object-centric representations.

We can now specify the technical goals of *generative factorization* as:

i. scene object factorization—separating modularized object information in an unstructured scene observation, i.e. factorizing the posterior distribution $p(z_1, z_2, ..., z_K|x)$[4];

ii. learning the invariant compositional generative mechanism (i.e. the mapping $X = g(\mathbf{Z}, V)$ or the conditional $P(X|\mathbf{Z}, V)$) that captures independent cause-effect relations between the factors $(Z_1, Z_2, ..., Z_K, V)$ and the observations X in an interventional sense.

[3] The ICM principle implies: $p(z_1, z_2, ..., z_K, v) = p(v|\,\mathbf{pa}_v) \prod_{k=1}^{K} p(z_k|\,\mathbf{pa}_{z_k})$, where $\mathbf{PA}_{[\cdot]}$ collects all of node's parents and equation 1.1 depicts a special case where $\forall C \in \{Z_1, Z_2, ..., Z_K, V\}$: $\mathbf{PA}_C = \emptyset$. See more details in §2.2.2.

[4] We assume that the viewpoint/observer configurations V are given/observable in our studies so we do not infer V here.

These specifications coincide with that of *disentangled representation learning*, as described in Suter et al. (2019), only we focus on object-centric cases.

However, as *disentangled representation learning* in an unsupervised setting is provably unidentifiable (leads to non-unique solutions, see Locatello et al. 2019a), in this thesis, we **do *not* aim to identify the "ground-truth" disentangled causal generative model but an equivalence that can answer counterfactual questions successfully**. I.e. we consider a model that can answer counterfactual questions successfully a valid recovery of the underlying generative process. In this case, we evaluate our factorization methods by assessing how well the discovered models answer counterfactual questions about the generative factors, e.g. "what would the observation be if it was taken from a different viewpoint"?

1.3 Thesis Structure

This thesis is structured around three main chapters (Chapter 3, 4, 5), where each extends a published conference paper. In these three main chapters, we will discuss three *generative factorization* (see §1.2) methods, each with a different focus:

- **Chapter 3 presents a model (viz. MulMON) that can leverage multi-view observations to reduce its uncertainty in representing a *static* scene's spatial structure and improve its accuracy in *scene object factorization.*** Learning 3D spatial structure from a single 2D image observation naturally leads to several inaccuracies or even incorrectness, with single-view OCRL methods falling victim to single-view spatial ambiguities arising from, for instance, optical illusions and occlusions (see Figure 1.2). To address this, we present MulMON as the first feasible unsupervised solution to the multi-view OCRL problem. Through the experiments, we demonstrate that not only can MulMON better resolve single-view spatial ambiguities and thus learn more accurate disentangled object representations than single-view methods, but it can also predict a static scene's appearances and object segmentations for novel viewpoints—i.e. it can recover the compositional generative mapping $X = g(\mathbf{Z}, V)$ from observational data. We show one example result of MulMON in Figure 1.2 and refer the readers to chapter 3 for more. This work was adapted from the following published paper with improved discussions:

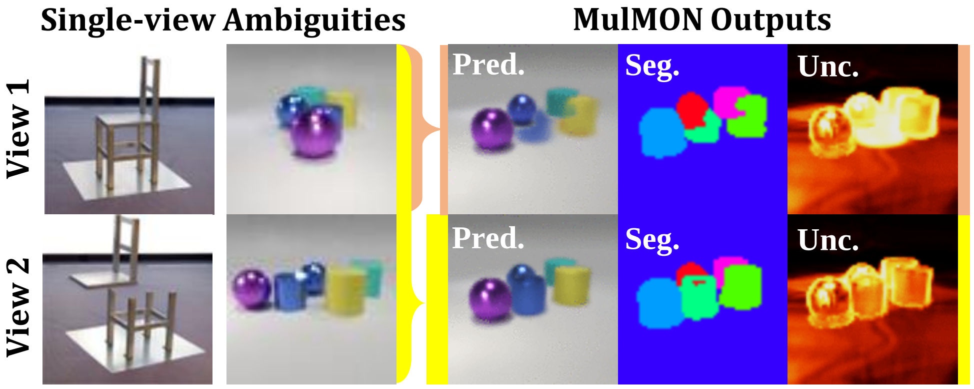


Figure 1.2: **Left**: Two example single-view visual cognitive ambiguities that can be resolved by introducing additional-view information: (first column) *optical illusions*—a perceived "chair" is an illusive effect of observing two separate parts from an accidental viewpoint (i.e. the famous "Beuchet chair", images adapted from Peters et al. (2017)) and (second column) *occlusions*. **Right**: An example of our multi-view model, i.e. MulMON, resolving single-view ambiguities (first row, second column) and producing improved novel-view rendering & segmentation results with lower spatial uncertainty (second row, column 3-5, highlighted in yellow) than the single-view baseline (first row, column 3-5, highlighted in orange).

> **Li Nanbo**, Cian Eastwood, and Robert Fisher. "Learning object-centric representations of multi-object scenes from multiple views." *Advances in Neural Information Processing Systems*, 2020.

- **Chapter 4 presents a technique (viz. LDS) that improves the *scene object factorization* by suppressing the duplicate object representations produced by under-constrained OCRL inference models.** For an observation x, the inference model of an OCRL method is responsible for inferring a set of latent object vectors $\mathbf{z} = \{z_1, z_2, ..., z_K\}$, where each z_k should capture a different object—i.e. *no* two latent vectors, e.g. z_i and z_j, $(i, j \in \mathbb{N}^{[1,K]}$ and $i \neq j)$ should capture the same object unless the slots are collapsed. This implies a *uniqueness assumption* among the inferred representations $z_1, z_2, ..., z_K$. However, most existing OCRL methods neglect this relational assumption such that they sometimes infer duplicate object representations which directly harm their performance in *scene object factorization.* In this work, we address this issue by introducing a *uniqueness constraint* (namely LDS) to regularize the original OCRL training processes. Our experiments show that OCRL models trained with the proposed method outperform the original models in *scene object factorization* and have fewer duplicate representations. This work was adapted from the following published paper with improved discussions:

  > **Li Nanbo** and Robert Fisher."Duplicate latent representation suppression for multi-object variational autoencoders" *The British Machine Vision Conference*, 2021.

- **Chapter 5 presents a model (viz. DyMON) that can recover the compositional generative mechanism from *moving-view-dynamic-scene* data so that it can factorize the entangled effects of observer motions and scene object dynamics that function independently.** Multi-view OCRL methods (e.g. MulMON, see Chapter 3) show advantages in handling *generative factorization* as compared to single-view methods. The key to their success is to recover a generative mechanism $P(X|\mathbf{Z}, V)$

that captures independent generative effects of the observers V and scene objects $\mathbf{Z}$ (see Figure 1.3, left) from data. However, to "destroy" accident correlations between Z and V in the data and ease training, the existing multi-view methods all assumed *static scenes*. As a result, they can not learn from *moving-view-dynamic-scene* data where both the observer and the scene objects are moving at the same—i.e. Z and V do correlate with each other along the time axis. To address this, we propose DyMON as the first feasible unsupervised framework that can recover a generative mechanism from *moving-view-dynamic-scene* data and infer time-indexed object representations (e.g. Z_k^t). As a result, DyMON can answer counterfactual questions about both "space" and "time"—i.e. it can predict a scene's appearances and segmentations at arbitrary times (querying across time) and from arbitrary viewpoints (querying across space). This work was adapted from the following published paper with improved discussions:

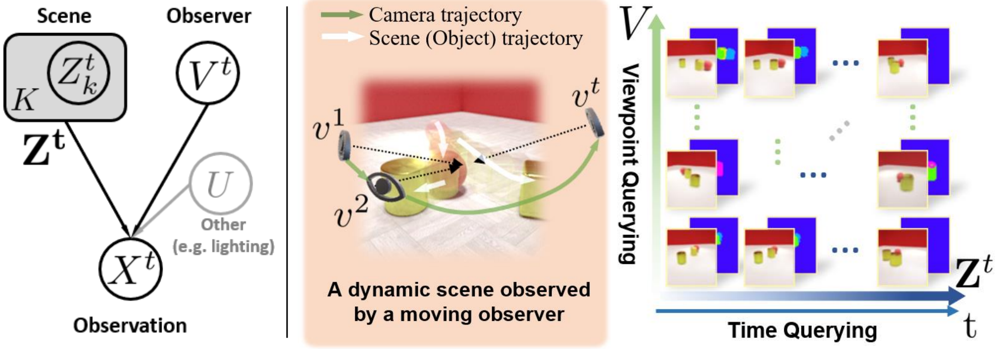


Figure 1.3: **Left**: A re-sketch of the causal generative model in Figure 1.1 with a special focus on the independent generative effects of the time-varying observers V^t and scene objects $\mathbf{Z^t}$. **Middle & Right**: DyMON can train on moving-camera-dynamic-scene data (middle) and learn a generative mechanism that allows it to perform space-time-queried rendering (right).

Li Nanbo, Muhammad Ahmed Raza, Hu Wenbin, Zhaole Sun, and Robert Fisher. "Object-Centric Representation Learning with Generative Spatial-Temporal Factorization." *Advances in Neural Information*

Processing Systems, 2021.

To help the readers better understand this thesis and its contributions, we will first provide the background knowledge of OCRL in **Chapter 2**. We will set up the context of OCRL from various perspectives, e.g. generative representation learning, scene understanding, and causality, and also review the existing OCRL methods. Lastly, in **Chapter 6** we will summarize the results presented in this work, outline the potential future directions, and discuss the broader impact of OCRL to society.

Chapter 2

Background

Although the trend of unsupervised *object-centric representation learning* (OCRL) emerged only recently and is still in an early stage, it encompasses the principle ideas of many well-established subjects. In this chapter, we will provide an overview of OCRL research from the perspectives of **generative representation learning** (§2.1), **factored latent space and compositionality** (§2.2), and **structural scene understanding and representation** (§2.3).

2.1 Generative Representation Learning

Unsupervised OCRL describes a subset of *representation learning* problems. Raw sensory data usually contain noise that can not only introduce excessive computation, but it can also interfere with the decision-making process of a model. Therefore, when doing *representation learning*, we expect to learn transformations of the raw data that summarize only the information needed for solving the downstream tasks so as to provide better efficiency and effectiveness. For notation convenience and showing the connections between OCRL and representation learning, in this section, we let X and $\mathbf{Z}$ (defined in §1.1) denote not only scenes but more general sensory input observations (X) and latent representations ($\mathbf{Z}$). A common assumption in representation learning is that a model's information processing process forms a Markov chain: $X \rightarrow \mathbf{Z} \rightarrow Y$, such that $X \perp\!\!\!\perp Y|\mathbf{Z}$ stands, i.e. X and Y are independent given $\mathbf{Z}$ (Tishby et al., 2000). Here Y denotes the target variables which are observables defined task-wise in supervised learning.

In unsupervised settings, representation learning needs to be handled in a generic way because the subsequent tasks are often unknown (i.e. Y is undefined). With the absence of Y, we consider that it is more meaningful to associate the observations (X) and the representations ($\mathbf{Z}$) along the direction of $\mathbf{Z} \rightarrow X$ rather than $X \rightarrow \mathbf{Z}$ as the former can be easily linked to natural generative narratives such as "$\mathbf{Z}$ causes X" and "$\mathbf{Z}$ controls X". In this thesis, as discussed in §1.1, we use $\mathbf{Z} \rightarrow X$ to formulate an unsupervised OCRL problem into the *generative representation learning* framework.

2.1.1 Latent-Variable Generative Models in General

Generative representation learning requires solving an inference problem within a generative framework. A convenient tool for this is *latent-variable generative models* because not only can they express the underlying generative structures (i.e. $\mathbf{Z} \rightarrow X$) but they also admit probabilistic interpretations and thus capture uncertainty. Figure 2.1 shows a general latent-variable generative model, where the inference process (highlighted in blue) demonstrates a natural connection to representation learning.

Most generative models assume that the observations (i.e. the values of X) are *I.I.D.* samples generated from the same distribution as $p(x)$. To generate new data samples that look "like" the observed ones, these models need to learn $p(x)$ from data, whether explicitly (Kingma and Welling, 2013; Rezende et al., 2014; Dinh et al., 2016) or implicitly (Goodfellow et al., 2014; Mohamed and

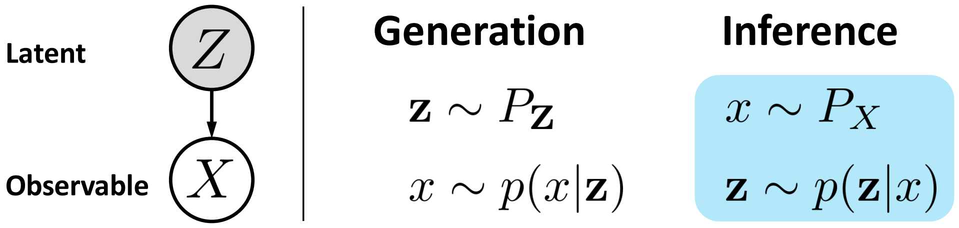


Figure 2.1: **Left:** A latent-variable generative model in the simplest and most general form. **Right:** Probabilistic views of the generative and inference processes of the left model.

Lakshminarayanan, 2017). Let us take the model in Figure 2.1 (left) for an example. For a latent variable model with a continuous latent variable $\mathbf{Z}$, the target density $p(x)$ of an observation x is computed explicitly as a marginal likelihood:

$$p(x;\theta) = \int_{\mathbf{z}} p(x,\mathbf{z};\theta)d\mathbf{z} = \int_{\mathbf{z}} p(x|\mathbf{z};\theta)p(\mathbf{z};\theta)d\mathbf{z}, \tag{2.1}$$

where we assume that the model is parameterized by some θ such that learning the model can be treated as estimating the model parameter θ. In theory, given a set of N *I.I.D.* training data $\mathbf{x} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$, one could obtain an estimate of θ with maximum (log-)likelihood estimation (MLE):

$$\hat{\theta}_{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log \int_{\mathbf{z}} p(x^{(n)}, \mathbf{z}; \theta)d\mathbf{z}, \tag{2.2}$$

However, this estimation is infeasible in general as it requires evaluating the marginal likelihood $p(x;\theta)$ using Eqn. 2.1, where computing the the integral over $\mathbf{z}$ as shown in Eqn. 2.1 is intractable. From a representation learning perspective, computing the Bayesian posterior over the unobservable variable $\mathbf{Z}$ for every sample of X:

$$p(\mathbf{z}|x;\theta) = \frac{p(x,\mathbf{z};\theta)}{p(x;\theta)} \tag{2.3}$$

requires knowing the "*Bayesian evidence*" ($p(x;\theta)$), so making exact inferences about the latent representations is also intractable in general. This, in fact, poses one of the central problems in Bayesian inference.

The problems depicted by the above three equations (i.e. Eqn. 2.1, 2.2, & 2.3) are referred to as *marginal inference*, *learning*, and *posterior inference* respectively in Kingma and Welling (2013). Although exact computation of these three equations are generally intractable, which make applying latent-variable generative models for unsupervised representation learning seemingly problematic, feasible solutions do exist—either with **model simplifying assumptions** or **efficient approximations**.

2.1.2 Linear Gaussian Models

To overcome the aforementioned computational issues and enable generative representation learning using latent-variable models, we can consider the linear Gaussian cases. I.e. we assume the model in Figure 2.1 (left) is a linear Gaussian model, i.e. $X \in \mathbb{R}^M$ and $\mathbf{Z} \in \mathbb{R}^D$ are both normally distributed and the relationship between them is linear:

$$p(\mathbf{z};\theta) = \mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{I}) \tag{2.4}$$

$$p(x|\mathbf{z};\theta) = \mathcal{N}(x|W\mathbf{z}+\mu,\Psi), \tag{2.5}$$

where $W \in \mathbb{R}^{M\times\ D}$ denotes a matrix, $\mu \in \mathbb{R}^M$ denotes a offset, $\Psi \in \mathbb{R}^{M\times\ M}$ denotes a diagonal covariance matrix, and $\theta = (W,\mu,\Psi)$ denotes the model parameters. We then show that, by taking Eqn. 2.4 &. 2.5 into Eqn. 2.1:

$$p(x;\theta) = \mathcal{N}(x|\mu, WW^T + \Psi), \tag{2.6}$$

the *marginal inference* problem becomes solvable. In this case, if we invert the generative model using Bayes' rule, exact *posterior inference* is also computable (Murphy, 2012):

$$p(\mathbf{z}|x;\theta) = \mathcal{N}(\mathbf{z}|\mu_{\mathbf{z}|x}, \Sigma_{\mathbf{z}|x}) \tag{2.7}$$

where the parameters $\Sigma_{\mathbf{z}|x} = (\mathbf{I}+W^T\Psi^{-1}W)^{-1}$ and $\mu_{\mathbf{z}|x} = \Sigma_{\mathbf{z}|x}W^T\Psi^{-1}(x-\mu)$ are computed per X value. Given a set of *I.I.D.* training data $\mathbf{x} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$, one can learn $\theta = (W, \mu, \Psi)$ by maximizing the data log-likelihood $\sum_{n=1}^{N} \log p(x^{(n)}; \theta)$ or its surrogates (cf. Barber 2012).

A latent-variable model with the above construction is essentially a *factor analysis* (FA) model (Fruchter, 1954; Cattell, 1965) which aims to explain the observations as linear combinations of a set of independent generative "factors". Note that as it is common to assume that the variation of the observed data can be explained by a relatively smaller set of common causes/factors in FA models, the number of "factors" (i.e. the dimension of the latent variable $\mathbf{Z}$) is often smaller than the dimensions of the observable variables ($D < M$). For this reason, *factor analysis* is often considered a technique for *dimensionality reduction* and closely related to *principle component analysis* (PCA). Such a relationship becomes more apparent if we further restrict the model by making its diagonal covariance matrix Ψ isotropic, i.e. $\Psi = \sigma^2 \mathbf{I}$. The linear Gaussian latent-variable models described by Eqn. 2.4 and Eqn. 2.5 will become *probabilistic principal component analysis* (PPCA) models (Tipping and Bishop, 1999).

2.1.3 Variational Auto-Encoders (VAEs)

Instead of imposing restrictive linear Gaussian assumptions and computing exact solutions, a more general approach to deal with the intractable marginalization in Eqn. 2.1 is performing approximations. The two most common approximation methods in Bayesian inference are *Monte Carlo sampling* (MC) and *Variational Inference* (VI) (Minka, 2001). Compared with MC methods, though VI methods can sometimes show inferior approximation accuracy (due to the *approximation gap*, see Bishop 2006; Cremer et al. 2018), they allow more efficient inference in high-dimensional cases (Andrieu et al., 2003) and thus have more general usage in a wide range of applications like image analysis. In this thesis, we work with VI methods to handle latent-variable generative models with constructions that

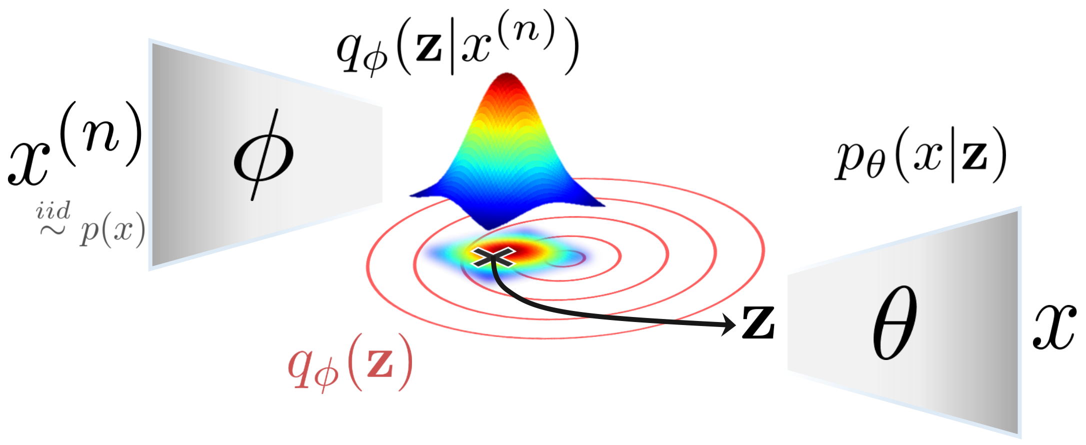


Figure 2.2: The construction of an example VAE model with 2D Bell-shape posteriors. The contours outlined by light-red curves represent the latent space recovered from the *I.I.D.* data as $q_\phi(\mathbf{z}) = \frac{1}{N}\sum_{n=1}^{N} q_\phi(\mathbf{z}\,|x^{(n)})$ (inspired by Kim and Mnih (2018)). All notations are defined in the text.

are far more general and expressive than the linear Gaussian cases. To be specific, we focus on latent-variable generative models with non-linear auto-encoder constructions, aka the *variational auto-encoders* (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014).

In VAEs we parameterize the model using neural networks with parameters θ and denote the resulting parametric families defined over the variables $\mathbf{Z}$ and X with $p_\theta(\mathbf{z})$ and $p_\theta(x|\mathbf{z})$, respectively. The neural network with parameter θ is often referred to as the *decoder* or the *generative model*, which describes the non-linear generative mapping from $\mathbf{Z}$ to X. Importantly, instead of handling the per-observation posterior inference with iterative EM coordinate ascent (which can be rather inefficient), Kingma and Welling (2013) introduce a probabilistic encoder $q_\phi(\mathbf{z}\,|x)$ (aka. the *inference model*), e.g. a neural network ϕ, to perform *amortized inference* (Kingma and Welling, 2013; Rezende et al., 2014; Marino et al., 2018). I.e. the encoder is used as a function that takes in x and outputs the parameters of the approximating posterior $q_\phi(\mathbf{z}\,|x)$ (see Figure 2.2). Recall that $q_\phi(\mathbf{z}\,|x)$ is an approximation of $p_\theta(\mathbf{z}|x^{(n)})$, we need to train both ϕ and θ (i.e. the encoder and decoder's parameters) to ensure the inference accuracy. This is formulated as a minimization of a Kullback–Leibler divergence $\mathcal{D}_{\text{KL}}[q_\phi(\mathbf{z}|x^{(n)})||p_\theta(\mathbf{z}|x^{(n)})]$ w.r.t. the global neural network parameters ϕ and θ. Despite the fact that computing such a KL divergence is impossible as $p_\theta(\mathbf{z}|x^{(n)})$ is unknown, we show that the target KL divergence can be decomposed (see Kingma and Welling 2013 for the derivation):

$$\mathcal{D}_{\text{KL}}[q_\phi(\mathbf{z}|x^{(n)})||p_\theta(\mathbf{z}|x^{(n)})] = \log p_\theta(x^{(n)}) - \underbrace{\mathbf{E}_{q_\phi(\mathbf{z}|x^{(n)})}[\log \frac{p_\theta(x^{(n)},\mathbf{z})}{q_\phi(\mathbf{z}|x^{(n)})}]}_{\mathbf{ELBO}^{(n)}} \tag{2.8}$$

$$\geq 0 \tag{2.9}$$

such that minimizing the KL divergence is equivalent to maximizing the evidence lower bound (i.e. $ELBO^{(n)}$). Note that $ELBO^{(n)}$ (in Eqn. 2.8) is defined for each data sample $x^{(n)}$. To train a VAE, we need to maximize an ELBO that is defined over the all training data ($\mathbf{x} = \{x^{(1)}, x^{(2)}, ..., x^{(N)}\}$), i.e.:

$$\begin{aligned}\mathcal{L}(\phi,\theta;\mathbf{x}) &= \sum_{n=1}^{N} \mathbf{E}_{q_\phi(\mathbf{z}|x^{(n)})}[\log \frac{p_\theta(x^{(n)},\mathbf{z})}{q_\phi(\mathbf{z}|x^{(n)})}] \\ &= \sum_{n=1}^{N} \mathbf{E}_{q_\phi(\mathbf{z}|x^{(n)})}[\log p_\theta(x^{(n)}|\mathbf{z})] - \sum_{n=1}^{N} \mathcal{D}_{\text{KL}}[q_\phi(\mathbf{z}|x^{(n)})||p_\theta(\mathbf{z})], \end{aligned} \tag{2.10}$$

w.r.t. both the encoder and decoder parameters (i.e. ϕ and θ, respectively). Although the gradient computations in this training process are less straightforward compared with most other neural networks, with certain treatments (e.g. a reparameterization trick, see Kingma and Welling 2013) applied, we can still train the model with gradient descent as is done for most differentiable neural networks.

One might have noticed that, because a KL divergence is always non-negative, Eqn. 2.8 also leads to an inequality expression (see Eqn. 2.9). This inequality implies that, unless the optimal parameters $\phi^\star$ and $\theta^\star$ are found, there will be a gap between the approximating posterior and the "true" posterior, i.e. $q_\phi(\mathbf{z}|x^{(n)})$ and $p_\theta(\mathbf{z}|x^{(n)})$. Cremer et al. (2018) referred to such gap as the *inference gap* and further factorized it into the *approximation gap* and the *amortization gap*. The latter is expressly ascribed to the introduction of amortized inference, while the former is often ascribed to the choice of variational priors. In chapter 4, we will show that an improved prior can effectively reduce the *approximation gap* and achieve better suboptimality. However, some recent discoveries (Gresele et al., 2021; Reizinger et al., 2022) also show that the *approximation gap* can provide some "unexpected" benefits in identifying the data generating mechanism.

2.2 Factored Latent Structure and Compositionality

So far, we have introduced latent-variable generative models for *generative representation learning*. As complex data are commonly considered generated from the rich interaction of higher-order explanatory factors (Fruchter, 1954; Schmidhuber, 1992; Desjardins et al., 2012; Bengio et al., 2013; Schölkopf et al., 2021), e.g. objects (see §2.2.3), it is appealing to show that latent-variable generative models are inherently powerful for modeling the compositional structure of data. Concretely, we can represent the explanatory factors and their interactions with a set of latent components (e.g. $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$) and some generative mechanism (e.g. $P(X|Z_1, Z_2, ..., Z_K)$), respectively. In unsupervised representation learning, as we do not have access to the "true" explanatory factors, a model needs to separate distinct factors of variation $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$ from the unstructured sensory input X—i.e. extracting ***disentangled representations***.

2.2.1 Disentangling Independent Factors of Variation

Though it seems ill-posed to disentangle "distinct factors of variation" without knowing what the generative factors are (it is indeed ill-posed, discussed in §2.2.2), we often aid disentangling by injecting prior knowledge about the factors in practice. Intuitively, we expect a change in a single latent component **1)** to be invariant to the other components (*modularity*, discussed in §2.2.2) and **2)** corresponds to a semantically meaningful variation in the input distribution (related to interpretability) (Bengio et al., 2013). For example, one can consider an inverse graphics model (Kulkarni et al., 2015b; Moreno et al., 2016; Yildirim et al., 2017; Yao et al., 2018) where we can independently change an object' properties (e.g. shape and color) by manipulating a single latent component. However, inverse graphics models often impose strong assumptions on the causal generative mechanism (assume a known and fixed renderer) so that they cannot learn and adapt when the domain-specific expert assumptions are violated (Schölkopf et al., 2021). In this thesis, we focus on learning the underlying (causal) generative mechanisms from data with weak and generic assumptions. In the framework of latent-variable generative models, disentangling factors of data variation is handled in the inference process (Desjardins et al., 2012), which can be viewed mathematically as factorization of a joint posterior: $p(z_1, z_2, ..., z_K|x)$. However, as our goal is to find common explanatory factors across all data points, we are more interested in factorizing globally a aggregated posterior $p(z_1, z_2, ..., z_K) = \mathbf{E}_{p_{data}(x)}[p(z_1, z_2, ..., z_K|x)]$.

Independent Latent Structure To ease the factorization of such a posterior, most of the existing works have resorted to imposing *linear independence* or *statistical independence* assumptions on the latent variables. Built upon (P)PCA and FA models, early studies of statistical shape/appearance representations (Turk and Pentland, 1991; Cootes et al., 1995; 2001; Prince et al., 2008) have shown that, by projecting the observation onto (linear) independent latent subspaces, these models can separate lower-level features of variation up to some interpretable level. For example, Cootes et al. (1995) showed that their resistor point distribution model can capture a resistor's position and deformation factors with two linearly-independent subspaces. Another family of models that explicitly express the same principle are the *independent component analysis* (ICA) (Comon, 1992; 1994). The original ICA models that an observation is a linear combination of multiple *statistically independent* non-Gaussian signals. Unlike (P)PCA and FA, which admit multiple equivalent solutions (unidentifiable), ICA models explicitly aim to separate the true signals that compose a multivariate observation. However, to guarantee its identifiability, such a model generally cannot handle non-linear mixings of signals (Hyvärinen and Pajunen, 1999) nor a dimensionality reduction setting (like (P)PCA and FA) (Hyvärinen and Oja, 2000).

Going beyond linear models, the idea of encouraging disentanglement with independence priors can also be generalized to deep latent-variable generative models (Chen et al., 2016; Voynov and Babenko, 2020), especially VAEs (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018a; Kumar et al., 2018; Esmaeili et al., 2019; Mathieu et al., 2019). As the recovery of VAEs' aggregated posteriors is handled by their probabilistic encoders as $q_\phi(z_1, z_2, ..., z_K) = \mathbf{E}_{p_{data}(x)}[q_\phi(z_1, z_2, ..., z_K|x)]$, these VAE-based approaches commonly encourage disentanglement by matching the recovered aggregated posterior to a factorized prior (e.g. an isotropic Gaussian) such that $q_\phi(z_1, z_2, ..., z_K)$ also factorizes: $q_\phi(z_1, z_2, ..., z_K) = \prod_{k=1}^{K} q_\phi(z_k)$. Therefore, the training of these disentanglement VAEs involves, either explicitly or implicitly, minimizing $\mathcal{D}_{\mathrm{KL}}[q_\phi(\mathbf{z})||p_\theta(\mathbf{z})]$ or other choices of divergence measures $\mathcal{D}_\star(q_\phi(\mathbf{z}), p_\theta(\mathbf{z}))$ (Mathieu et al., 2019). For examples, Higgins et al. (2017) proposed a model (beta-VAE) that does disentangled representation learning by explicitly emphasizing the minimization of the KL divergence item of Eqn. 2.10 during training. Kim and Mnih (2018) handled disentanglement by minimizing a cross-dimension correlation measure (a.k.a. the total correlation) extracted from the $\mathcal{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}), p_\theta(\mathbf{z}))$.

Interpretable Latent Structure So far, the focus of most existing unsupervised disentanglement research has been on enforcing latent factorization. Though these works generally produce impressive results in learning disentangled and interpretable representations, little understanding has been gained on why the factored latent structures of these models are interpretable. A heuristic explanation provided by Burgess et al. (2018) and Mathieu et al. (2019) suggests that the success of learning interpretable latent structures attributes to achieving a "sufficient" degree of overlap between the posterior distributions across the dataset. The intuition is easy to understand if we consider the data are composed of reusable modules that do not change simultaneously (i.e. the sparse mechanism shift principle, discussed in §2.2.2)—that is, encouraging overlap encourages the discovery of the shared and re-usable generative modules. These reusable modules are associated with specific concepts in human understanding of the world, e.g. colors, shapes, and positions. By looking at the extreme cases of the overlap degree, i.e. too high or too low, Mathieu et al. (2019) stated that **1)** insufficient overlap will lead to a look-up-table latent structure, which implies zero compositional structure (zero re-usable information) in the data, whereas **2)** too much overlap will diminish the value of the latent encodings as representations—the encodings contain little information about the "corresponding" observations. Yet, as a sound study of how significant overlap is in improving disentangled representation learning is still missing, it remains unclear how one should define "sufficiency" quantitatively.

2.2.2 Modularity and Causal Disentanglement

We discussed in §2.2.1 that most existing works in disentangled representation learning are established on several intuitive principles, e.g. *modularity* and *one-to-one correspondence* (see §2.2.1). As a result, various metric proposals for quantifying the quality of "disentangled representations" (Higgins et al., 2017; Ridgeway and Mozer, 2018; Eastwood and Williams, 2018; Kumar et al., 2018; Kim and Mnih, 2018) tend to somewhat disagree with each other (Locatello et al.,

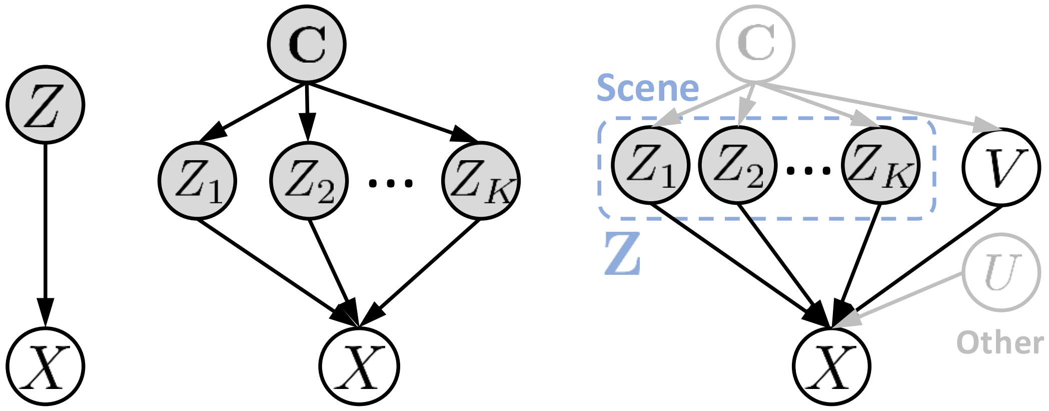


Figure 2.3: **Left:** A causal generative model with a single *element ingredient*/generative factor. **Middle:** A general form of disentangled causal generative model with a set of generative factors $\{Z_1, Z_2, ..., Z_K\}$ that are confounded by some confounder(s) $\mathbf{C}$. **Right:** An object-centric causal generative model, where V is the observer variable, each $Z_k \in \{Z_1, Z_2, ..., Z_K\}$ represents one and only one scene object and, together, they form a physical scene $\mathbf{Z}$. The variables $\mathbf{C}$ and U (in grey) are useful for sophisticated modeling, e.g. use $\mathbf{C}$ to summarize global scene layout (Reddy et al., 2022) and U to represent lightings, although they are often ignored for simplicity.

2019a). To address these issues and formalize the research of disentanglement, a generally-accepted definition of "disentanglement" is needed. A rising trend is to cast the problem of *disentanglement* in the *causality* theoretical framework (Locatello et al., 2019a; Suter et al., 2019; Schölkopf et al., 2021).

Interventional vs. Statistical Independence In the *causality* framework, disentanglement should be focusing on recovering the "true" disentangled causal mechanism (see Figure 2.3, middle), where inferring disentangled representations is viewed as reconstructing the "true" causal factors of an observation (Peters et al., 2017; Suter et al., 2019; Schölkopf et al., 2021). Importantly, Suter et al. (2019) formalized the previously intuitive *modularity* description from the perspective of intervention using the principle of *independent causal mechanisms* (abbr. ICM) (Schölkopf et al., 2012; Peters et al., 2017). They argue that, instead of *statistical independence*, "independent factors of variation" describes the invariance of the generative mechanisms. In this thesis, we emphasize the invariance of these modules under direct interventions—namely, *interventional independence*. Take $\forall_{i \neq j} Z_i, Z_j \in \{Z_1, Z_2, ..., Z_K\}$, we show the difference between *statistical independence* and *interventional independence* as follows:

$$Statistical: \qquad P(Z_i|Z_j) = P(Z_i) \tag{2.11}$$

$$Interventional: \qquad P(Z_i|\,\mathrm{do}(Z_j = z_j), \mathbf{PA}_{Z_i}) = P(Z_i|\mathbf{PA}_{Z_i}), \tag{2.12}$$

where do$(\cdot)$ denotes a mathematical operator (Pearl, 2012) that performs interventions (i.e. assigning manually a value to a variable, ignoring all its causes). It is easy to see that Eqn. 2.11 implies $P(Z_1, Z_2, ..., Z_K) = \prod_{k=1}^{K} P(Z_k)$ while Eqn. 2.12 does *not*. In other words, these generative factors can be statistically dependent. This can also be seen from Figure 2.3 (middle) that $\{Z_1, Z_2, ..., Z_K\}$ are independent only if their confounder(s) $\mathbf{C}$ is observed. In a special case where $\mathbf{C} = \emptyset$, *interventional independence* and *statistical independence* coincide with each other. Importantly, such conceptual generalization not only opens the avenue of disentangling correlated causal factors (Träuble et al., 2021; Moran et al., 2021; Reddy et al., 2022) but it also allows us to infer *interventional distibutions*—which is essential for counterfactual reasoning (Buesing et al., 2019; Besserve et al., 2020; Nanbo et al., 2021) and compositional *O.O.D.* generalization[1] (Higgins et al., 2018; Shen et al., 2021).

Identifiability From the *causality* point of view, recent advances in *disentangled representation learning* show particular interests in *identifiability*. Although the early results have exposed the unidenfiability issues of FA (Shapiro, 1985) and non-linear ICA models (Hyvärinen and Oja, 2000), the study of disentanglement

[1] We refer the readers to Fig. 1 in Schölkopf et al. (2021) for more details about how causal factorization aids *O.O.D.* generalization.

models' identifiability has just become popular recently. Notably, Locatello et al. (2019a) proved the existence of *entangled* latent structure that can yield the same observation distribution (i.e. rotational equivalence, under an standard choice of isotropic Gaussian prior)—which left assumption-free unsupervised disentangled representation learning an ill-posed problem. Considering the seemingly contradiction between the unidentifiability conclusion and the practical successes of many unsupervised disentangled representation learning works (Higgins et al., 2017; Kim and Mnih, 2018; Chen et al., 2018a; Kumar et al., 2018; Esmaeili et al., 2019; Mathieu et al., 2019), Locatello et al. (2019a) ascribed the practical successes of these works to the inductive biases implicitly introduced by either the models or the data. This motivates an extensive study of elucidating and understanding the effects of inductive biases on model identification (Rolinek et al., 2019; Locatello et al., 2019b; Duan et al., 2019; Locatello et al., 2020a; Besserve et al., 2020; Moran et al., 2021; Gresele et al., 2021; Reizinger et al., 2022).

2.2.3 Object-Centric Disentanglement and Composition

We introduce the problem of *object-centric representation learning* (OCRL) as a special case of disentanglement representation learning. In OCRL, we explore the compositional structures of scenes and aim to learn to capture scene objects with a set of latent components (e.g. $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$). As shown in Figure 2.3, the assumption about the underlying causal mechanisms of an OCRL model is rather similar to that of a generic disentangled representation learning model, only with more available domain-specific knowledge to ease the disentanglement around objects.

Compositional Mixing and Spatial Disentanglement The problem of OCRL is inherently tied with spatial reasoning. To associate the latent components $\mathbf{Z} = \{Z_1, Z_2, ..., Z_K\}$ with scene objects, a model needs to reason about compositional structures of scenes for object discovery. As it is generally accepted by the representation learning community that "object discovery should be treated as a crucial part of OCRL rather than a separate problem" (Greff et al., 2019), instead of using pre-segmented images (Yao et al., 2018) or introducing separate segmenters (Ren et al., 2015; He et al., 2017; Zheng et al., 2021; Strudel et al., 2021), recent advances (Eslami et al., 2016; Greff et al., 2016; Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2019; Nanbo et al., 2020) often exploit assumptions about the image composition function (i.e. the generative mechanisms $p(x|z_1, z_2, ..., z_K)$) to handle object discovery. Most of the existing image composition designs encompass the idea of *alpha blending* (Porter and Duff, 1984)—i.e. generating separate objects in separate layers and composing them by reasoning about occlusions (the orderings of the layers). It is important to note that the *alpha blending* design play an essential role in spatial disentanglement because they impose competition among the K latent components in explaining M RGB pixels. One can consider the competitions as clustering M pixels into the K latent components with a spatial attention module (Locatello et al., 2020b). Also, the "objects" here are defined in a statistical sense, i.e. we make no assumptions about the "objects" and only treat them as statistical modules that change independently. However, we admit that, like general disentanglement representation learning, object disentanglement without any assumption leads to unidenfiability issues (see the *identifiability* discussion in §2.2.2).

On Multi-Granularity Disentanglement Like many generic disentangled representation learning models (Higgins et al., 2017; Kim and Mnih, 2018; Mathieu et al., 2019), it is common to make statistical independence assumptions in OCRL (i.e. ignoring the confounders $\mathbf{C}$) for simplicity. However, compared to a generic model, it was shown by Burgess et al. (2019); Greff et al. (2019); Nanbo et al. (2020) that an OCRL allows to disentangle across multiple granularities: object-level and feature-level (like most generic disentanglement models). In ad-

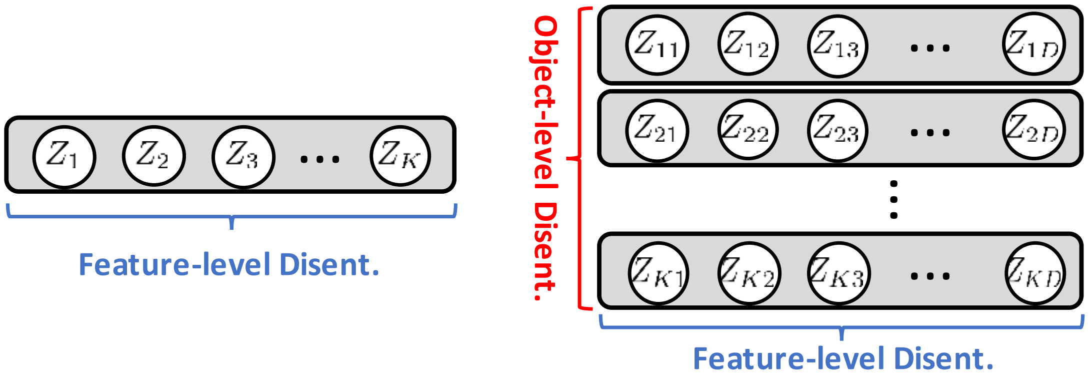


Figure 2.4: **Left:** The latent structure produced by a generic disentangled representation learning model. **Right:** The latent structure produced by an OCRL model, where factorization is achieved across multiple granularities: object-level (disentangled across rows) and feature-level (disentangled across columns).

dition, more sophisticated metrics (Dang-Nhu, 2021) are proposed for quantitatively evaluating the multi-level disentangled representations. Figure 2.4 shows the topological comparison between the latent structures learned by a generic disentangled representation learning model and an OCRL model. In the context of visual understanding, we can also consider a generic model a "single-object" model and an OCRL model a multi-object model—in analogy to VAEs and multi-object VAEs.

2.3 Structural Scene Understanding and Representation

The ultimate goal of *object-centric representation learning* (OCRL) is to enable AI systems to understand the physical world. Therefore, OCRL is widely considered a subsidiary subject of *visual scene understanding* (Kutulakos and Seitz, 2000; Wu et al., 2015; Jimenez Rezende et al., 2016; He et al., 2017; Eslami et al., 2018; Mildenhall et al., 2020) and *world model learning* (Schmidhuber, 2015; Ha and

Schmidhuber, 2018; Kipf et al., 2019; Lin et al., 2020; Subramanian et al., 2022). In this section, around spatial and temporal visual understanding of scenes, we review some of the important literature and discuss how they motivated the three papers published within this thesis (discussed in Chapter 3-5).

2.3.1 Spatial Structures of Scenes

Traditional spatial scene understanding has been around constructing explicit scene representations from observations (Curless and Levoy, 1996; Kutulakos and Seitz, 2000; Kolmogorov and Zabih, 2002; Pollefeys et al., 2004; Newcombe et al., 2011). Although the scene representations captured by these models are fully explainable, the spatial resolution of these representations is often limited by the discretization performance of the sensing and computing devices. As a result, they do not scale well when we increase the spatial scales or resolution of a scene—even with the support of deep neural networks (Liao et al., 2018; Gkioxari et al., 2019; Potamias et al., 2022). Unlike *explicit scene representations*, *implicit scene representations* often encode scenes into implicit parameters defined in continuous space, which theoretically allows to represent scenes with “infinite” resolution and scales. With the booming deep representation learning, an increasing amount of attention has been paid to *learning* ***implicit scene representations*** *from data.*

Global 3D Structure and Flat Representation As the family of latent-variable generative models equip *dimensionality reduction* and *unsupervised representation learning*, they stand out as one of the most important tools for implicit scene representation learning. Many recent breakthroughs (Wu et al., 2016; Eslami et al., 2018; Tobin et al., 2019; Yang et al., 2019) have shown impressive performance in inference efficiency and scalability by representing a scene globally as a latent (random) vector. Importantly, Eslami et al. (2018); Tobin et al. (2019) target the problem of learning 3D structure from multi-view RGB images. To reduce the spatial uncertainty and enable explicit 3D knowledge evaluation, they built models that can reason about viewpoint effects and perform with *multi-view explorations* (Hartley and Zisserman, 2003; Aulinas et al., 2008) and *novel-view synthesis* (Kulkarni et al., 2015a; Penner and Zhang, 2017; Mildenhall et al., 2019). It is also worth mentioning the recent rise of *neural radiance field* (NeRF) (Mildenhall et al., 2020; Martin-Brualla et al., 2021; Pumarola et al., 2020). Despite the fact that these NeRF models have shown remarkable success in representing scenes (as the parameters of the deep networks) at high resolutions, we do not consider them scene understanding models as they only aim to memorize (hard encoding) the scene structure of a single scene during "training". Regardless of the high rendering cost (w.r.t. both memory and runtime) and the assumption of known camera parameters, combing NeRFs' rendering power with latent-variable generative models can be a promising direction in scene understanding (Yu et al., 2021a; DeVries et al., 2021; Kosiorek et al., 2021; Sharma et al., 2022).

Structured Scene Representation around Objects Many recent advances in implicit unsupervised scene representation learning fixate on "global understanding" (i.e. representing a scene as a single "flat" random vector). I.e. they fail to interpret the rich compositional structures of natural scenes around objects (see the *granularity* discussion in §2.2.3). To form more structured and interpretable latent representations of scenes, as discussed in §2.2.3, a series of unsupervised OCRL methods (Eslami et al., 2016; Greff et al., 2016; 2017; Kosiorek et al., 2018; Burgess et al., 2019; Greff et al., 2019; Lin et al., 2019; Engelcke et al., 2019; Locatello et al., 2020b; Goyal et al., 2020; Didolkar et al., 2021; Engelcke et al., 2021; Emami et al., 2021; Yu et al., 2021b) have been proposed. Yet, as most existing models have been targeting a primary scenario, i.e. a single-view image observation setting, they fall victim to single-view spatial ambiguities (e.g. optical illusions and occlusions, see Figure 1.2 in §1.3). As a result, they fail to accurately capture the scenes' 3D spatial structures, and decompose scenes into objects (i.e. perform factorization). To overcome the single-view spatial ambiguity issue, **Nanbo et al. (2020) proposed MulMON as *the first unsupervised framework for learning accurate OCRL by leveraging multiple views*** (discussed in Chapter 3). MulMON essentially learns a generative mechanism that responds to the generative factors $(Z_1, Z_2, ..., Z_K, V)$ hence can answer counterfactual questions about spatial scene structures around objects (e.g. predicting scene appearances and object segmentations for novel views). A series of later works (Niemeyer and Geiger, 2021; Chen et al., 2021; Stelzner et al., 2021) exploited a similar idea, but in more specialized settings or applications. As an unconstrained factorization of $(Z_1, Z_2, ..., Z_K, V)$ generally leads to some unidentifiable latent structures, many existing unsupervised OCRL methods (Burgess et al., 2019; Greff et al., 2019; Nanbo et al., 2020) commonly produce duplicate scene object representations which directly harms the scene factorization performance. Inspired by the lines of *non-maximum suppression* works (Lowe, 2004; Bodla et al., 2017) and *contrastive learning* works (Chen et al., 2020; He et al., 2020), **Nanbo and Fisher (2021) proposed *a decorrelation prior to suppress the duplicate object latent representations*** (discussed in Chapter 4).

2.3.2 Temporal Structures of Scenes

Representing scene dynamics, i.e. the temporal evolution of spatial scene structures, is another fundamental aspect of scene understanding. In §2.3.1, we introduced multi-view information aggregation a natural and effective way of resolving spatial ambiguities and learning scene (object) representations. However, as most existing OCRL works, e.g. GQN (Eslami et al., 2018) and MulMON (Nanbo et al., 2020) are built upon either a static scene assumption, they often do *not* handle well dynamic scenes in both training and testing. This motivates the research of unsupervised OCRL in dynamic-scene setting.

From a Static Observer's View Most existing unsupervised OCRL methods that target dynamic scenes (Hsieh et al., 2018; Kosiorek et al., 2018; Jaques et al., 2020; Lin et al., 2020) fall in the category of *single-view-dynamic-scene* scenarios, where the scene objects are moving and the observer (camera) is fixed. For simplicity, these models often omit the viewpoint variable V shown in Figure 1.1 and 2.3 (right). In fact, as Hsieh et al. (2018); Kosiorek et al. (2018); Jaques et al. (2020) all employed the "paste stickers on canvas" rendering schemes of Jaderberg et al. (2015); Eslami et al. (2016) and their experiments were on 2D MNIST digits, it is unclear whether or not these models can generalize to 3D scenes. Although Lin et al. (2020) construct explicit transition models of 3D object dynamics based on video observations, the inferred scene object representations are still implicit. In this case, the models cannot perform novel-view synthesis and thus do *not* support explicit evaluation of how well the inferred representations capture the 3D scene spatial information.

From a Moving Observer's View Endowing machines with the ability to reason about viewpoints and scene dynamics is particularly important. Achieving so will allow us to **1)** evaluate explicitly the represented 3D spatial structures at any time of a dynamic event, and **2)** handle representation learning in the general *moving-view-dynamic-scene* setting which appear commonly in real-world applications (Grauman et al., 2022). Singh et al. (2019) proposed a non-object-centric framework, i.e. T-GQN, that models the spatial representation learning at each time step as a stochastic process and transitions between these time-stamped stochastic processes with a state machine. Although handles representation learning in the scenario of *moving-view-dynamic-scene* and learns time-dependent scene representations, it **1)** cannot attain object-level scene factorization, and **2)** typically requires multi-view data at each time step (as so-called "context") to disentangle the "coincidentally" entangled scene motions and camera motions (i.e. *temporal entanglement*, discussed in Chapter 5) during training.

To learn *dynamics-aware* object-centric scene representations, Nanbo et al. (2021) proposed DyMON as *the first unsupervised OCRL framework that targets the moving-view-dynamic-scene setting* (discussed in Chapter 5). By factorizing the generative effects of the scene object motions and the observer motions, DyMON can describe the motions of each object and represents their representations as functions of time. Similarly, a concurrent work, SIMONe (Kabra et al., 2021), also investigated time-dependent OCRL representations like DyMON. However, as the "time-varying elements" defined in SIMONe refer to the *cross-frame pixel changes* caused by the viewpoint changes, i.e. the scenes are still assumed static, SIMONe is more similar to MulMON (Nanbo et al., 2020) rather than DyMON.

Chapter 3

MulMON: Multi-view Multi-Object Network

Learning object-centric representations of multi-object scenes is a promising approach towards machine intelligence, facilitating high-level reasoning and control from visual sensory data. However, current approaches for *unsupervised object-centric scene representation* are incapable of aggregating information from multiple observations of a scene. As a result, these "single-view" methods form their representations of a 3D scene based only on a single 2D observation (view). Naturally, this leads to several inaccuracies, with these methods falling victim to single-view spatial ambiguities. To address this, we propose *The Multi-View and Multi-Object Network (MulMON)*—a method for learning accurate, object-centric representations of multi-object scenes by leveraging multiple views. In order to sidestep the main technical difficulty of the *multi-object-multi-view* scenario—maintaining object correspondences across views—MulMON iteratively updates the latent object representations for a scene over multiple views. To ensure that these iterative updates do indeed aggregate spatial information to form a complete 3D scene understanding, MulMON is asked to predict the appearance of the scene from novel viewpoints during training. Through experiments we show that MulMON better-resolves spatial ambiguities than single-view methods—learning more accurate and disentangled object representations—and also achieves new functionality in predicting object segmentations for novel viewpoints.

This chapter is an extended version of the paper "*Learning Object-Centric Representations of Multi-Object Scenes from Multiple Views*" (Nanbo et al., 2020), published at *Neural Information Processing Systems* (2020).

3.1 Introduction

Traditional VAEs (Kingma and Welling, 2013; Rezende et al., 2014) use a "single-object" or "flat" random vector representations that fail to obtain compositional interpretations of natural scenes, i.e. the existence of interchangeable objects with common properties. As a result, "multi-object" or object-centric representations have emerged as a promising approach to scene understanding, improving sample efficiency and generalization for many downstream applications like relational reasoning and control (Carlos et al., 2008; Mnih et al., 2015; Bapst et al., 2019; Mambelli et al., 2022). However, recent progress in unsupervised *object-centric scene representation learning* (OCRL) has been limited to "single-view" methods which form their representations of 3D scenes based only on a single 2D observation (view). As a result, these methods form inaccurate representations that fall victim to single-view spatial ambiguities (e.g. occlusions and *optical illusions*) and fail to capture 3D spatial structures.

To address this, we present MulMON (Multi-View and Multi-Object Network)—an unsupervised method for learning object-centric scene representations from multiple views. Using a spatial mixture model (Greff et al., 2017) and iterative amortized inference (Marino et al., 2018), MulMON sidesteps the main technical

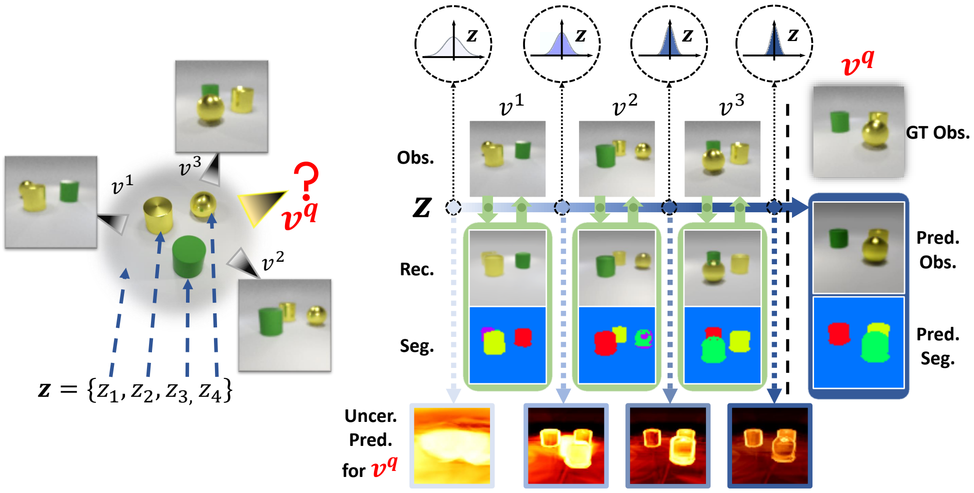


Figure 3.1: **Left:** *Multi-object-multi-view* setup. v^q denotes the query viewpoint, while z_k denotes“slot” k, i.e. the latent object representation of a scene object. **Right:** MulMON overview. Starting with a standard normal prior, MulMON iteratively refines the posterior over $\mathbf{z}$ over multiple views, each time reducing its uncertainty about the scene——as illustrated by the darkening, white-to-blue arrow. Within-view “inner loop” iterations are depicted by the green arrows and boxes. Cross-view “outer-loop” iterations are depicted by the white-to-blue arrows and boxes. At the bottom, we have visualised MulMON's reduction in uncertainty about $\mathbf{z}$ in image space, where each image shows the per-pixel variance of MulMON's predicted observation from query viewpoint v^q. MulMON's final predictions for v^q (observation and segmentation) are shown to the right of the vertical dotted line.

difficulty of the *multi-object-multi-view* scenario—maintaining object correspondence across views—by iteratively updating the latent object representations for a scene over multiple views, each time using the previous iteration's posterior as the new prior. To ensure that these iterative updates do indeed aggregate spatial information, rather than simply overwrite, MulMON is asked to predict the appearance of the scene from novel viewpoints during training. Given images of a *static* scene from several viewpoints, MulMON forms an object-centric representation, then uses this representation to predict the appearance and object segmentations of that scene from unobserved viewpoints. Through experiments we demonstrate that:

- MulMON better-resolves spatial ambiguities than single-view methods like IODINE (Greff et al., 2019), while providing all the benefits of object-centric representations that *non-object-centric* methods like GQN (Eslami et al., 2018) lack, e.g. object segmentations and manipulations (see §3.5).
- MulMON accurately captures 3D scene information (rotation along the vertical axis) by integrating spatial information from multiple views (see §3.5.3).
- MulMON achieves both inter- and intra-object disentanglement—enabling both single-object and single-object-property scene manipulations (see §3.5.3).
- MulMON represents the first feasible solution to the *multi-object-multi-view problem*, permitting new functionality like viewpoint-queried object-segmentation (see see §3.5.2).

3.2 Problem: Multi-View OCRL

We discussed in §1.1 that the problem of OCRL can be formulated as a factorization of $p(z_1, z_2, ..., z_K|x)$, where x is a single RGB image in a single-view setting (Burgess et al., 2019; Greff et al., 2019), $z_k \in \mathbb{R}^D$ is the value of a single object factor Z_k, and K is the number of object *slots* (greater than the actual number of objects). With this in mind, we can define a more general object-centric scene representation learning problem as that of learning a representation of an up-to-K object scene based on T uncalibrated observations from random viewpoints, where the scenes are static and assumed to be a spatial configuration that is independent of the observer. Formally, this involves factorizing the posterior $p(z_1, z_2, ..., z_K|x^1, x^2, ..., x^T)$. Since each 2D observation x^t of the 3D scene must be associated with a viewpoint v^t, we specify the problem as that of computing $p(z_1, z_2, ..., z_K|x^1, x^2, ..., x^T, v^1, v^2, ..., v^T)$ or compactly, $p(\mathbf{z}\,|\{x^t, v^t\})$, where

$v^t \in \mathbb{R}^J$ is the viewpoint sample associated with the image sample x^t. Note that both the $\{Z_1, Z_2, ..., Z_K\}$ and the $\{(X^t, V^t)\}$ are exchangeable.

3.3 Method

Our goal is to learn structured, object-centric scene representations that accurately capture the spatial structure of 3D multi-object scenes, and to do this by leveraging multiple 2D views. Key to achieving this is **1)** an outer loop that iterates over views, aggregating information while avoiding the object matching problem, and **2)** a training procedure that ensures that these outer loops are indeed used to form a complete 3D understanding of the scene, rather than just overwriting each other. We detail **1)** in §3.3.1, and **2)** in §3.3.4. Additionally, we describe the viewpoint-conditioned generative model and iterative inference procedure in §3.3.2 and §3.3.3 respectively.

3.3.1 Iterating Over Views

Cross-View Iterations (The Outer Loop) For a static scene, we consider that the latent scene representation $\mathbf{Z} = \{Z_k\}$ is updated sequentially in T steps as the T observations are obtained one-by-one from $t = 1$ to $t = T$, where t denotes the updating step. This suggests that $\mathbf{Z}^t$ is obtained by updating $\mathbf{Z}^{t-1}$ using a new observation X^t, taken at the viewpoint V^t (see the green box in Figure 3.2, left). Therefore, by making an assumption that $\mathbf{Z} = \mathbf{Z}^t$ for any integer $t \in [1, T]$ (i.e. considering Z^t the best $\mathbf{Z}$ by far at time t), we can compute the target multi-view posterior in a recursive form as:

$$p(\mathbf{z} | x^{1:T}, v^{1:T}) = p(\mathbf{z}^0) \prod_{t=1}^{T} p(\mathbf{z}^t | x^t, v^t, \mathbf{z}^{t-1}), \tag{3.1}$$

where $\mathbf{z}^{t-1} = \{z_k^{t-1}\}$ is the value(s) of the scene (object) representation(s) $\mathbf{Z}^{t-1} = \{Z_k^{t-1}\}$ *before* observing the image x^t at viewpoint v^t, $\mathbf{z}^t$ the latent value(s) afterwards, and $p(\mathbf{z}^0)$ the initial guess, which is assumed to be a standard isotropic Gaussian: $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The formulation in Eqn. 3.1 turns the multi-view problem into a recursive single-view problem and, in theory, enables online learning of scenes from an infinitely large number of observations without causing memory overflow.

Within-View Iterations (The Inner Loop) As shown in Figure 3.2, MulMON consists of a scene-representation inference model and a viewpoint-conditioned generative model. In each iteration, the inference model starts with a prior assumption about the latent object representations, i.e. $\mathbf{Z} = \mathbf{Z}^{t-1} = \{Z_k^{t-1}\}$, and approximates the target posterior distribution $p(\mathbf{z}^t = \{z_k^t\} | x^t, v^t, \mathbf{z}^{t-1})$ after observing x^t at viewpoint v^t. The approximation, as mentioned in §3.1, is handled by iterative amortized inference (Marino et al., 2018) and the approximate posterior is carried to the next iteration as the new prior. In other words, a single iteration is a single-view process that updates the latent object-centric scene representations $\mathbf{Z} = \mathbf{Z}^{t-1} = \{Z_k^{t-1}\}$ using an image sample x^t and the associated viewpoint v^t. We call the single-view iterative process the *inner loop*, and the cross-view Bayesian updating process (see Eqn. 3.1) the *outer loop*.

3.3.2 Generative Model

We model image observations X^t with a spatial Gaussian mixture model (Williams and Titsias, 2004; Greff et al., 2017), similar to MONet (Burgess et al., 2019) and IODINE (Greff et al., 2019), only we condition such generation on the viewpoint variable V^t. We write the viewpoint-conditioned generative likelihood as:

$$p_\theta(x^t | \mathbf{z}^t, v^t) = \prod_{i=1}^{M} \sum_{k=1}^{K} p_\theta(C_i^t = k | z_k^t, v^t) \cdot p_\theta(x_{ik}^t | z_k^t, v^t), \tag{3.2}$$

where x_{ik}^t are the RGB values in image t at a pixel location i that pertain to object k, $p_\theta(x_{ik}^t | z_k^t, v^t)$ is the Gaussian density function parametrized by a neural network θ, and m_{ik} is the mixing coefficient for object k and pixel i, i.e. the probability that pixel i is assigned to the k-th object. More formally, $m_{ik} = p_\theta(C_i^t = k | z_k^t, v^t)$,

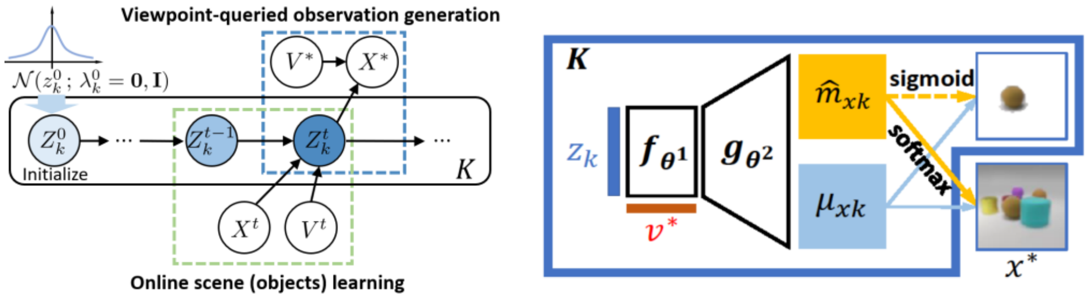


Figure 3.2: **Left:** Graphical view of MulMON's cross-view iterations. The two core components, a viewpoint-conditioned generative model (§3.3.2) and an inference model (§3.3.3), are shown in blue and green boxes respectively. **Right:** Viewpoint-conditioned generative model. To generate a viewpoint-conditioned observation, samples of the K latent latent object representation, i.e. $\{z_k\}$, will be first transformed w.r.t. a viewpoint v^* using the function f_{θ^1} and then passed through a decoder g_{θ^2} to render a viewpoint-queried observation x^*. As shown, the unknown parameters of the spatial Gaussian mixture (see Eqn. 3.2), i.e. the pixel-wise Gaussian means μ_{xk} and mixing probabilities ($\mathbf{softmax}(\hat{m}_{xk})$), are output by g_{θ^2}, and image observations x^* are essentially sampled from the spatial Gaussian mixture.

where C_i^t is a categorical random variable and $C_i^t = k$ represents the event that pixel i is assigned to the k-th object. This is an important property for object segmentation, as it implies that every pixel in x^t must be explained by one and only one object. Together, the M mixing coefficients for object k (one per pixel) form a soft object segmentation mask $m_k = p_\theta(C^t = k | z_k^t, v^t)$. We assume all pixel values x_{ik}^t are independent given the corresponding latent object representation z_k^t and viewpoint v^t, and simplify computations by using a fixed variance $\sigma^2 = 0.01$ for all pixels. We refer the reader to Appendix 3C for the implementation details of Eqn. 3.2. In practice, we split the parameters θ into two pieces, θ^1 and θ^2, in order to handle the viewpoint-queried neural transformation and observation-generation separately in two consecutive stages. That is, we first transform the K latent object representations $\boldsymbol{z}^t$ w.r.t. a viewpoint v^q using the function f_{θ^1}, then we pass the output through a decoder g_{θ^2} in order to render a viewpoint-queried observation x^q. We illustrate this process in Figure 3.2 (right) and Algorithm 1.

3.3.3 Inference

Though Eqn. 3.1 simplifies the inference problem by breaking the computation of the *multi-view* OCRL posterior $p(\mathbf{z}|\{(x^t, v^t)\})$ into a recursive computation of a series of *single-view* OCRL posteriors $p(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{t-1})$, exact inference of $p(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{t-1})$ is still intractable. Similar to IODINE (Greff et al., 2019), we apply iterative amortized inference (Marino et al., 2018) to approximate the intractable target posterior. However, unlike IODINE, which always initializes the prior from a standard Gaussian, the inference model of MulMON takes an approximate posterior obtained from previous observations (except for $\mathbf{Z}^0$) as the prior. In this case, we approximate the intractable posterior with $q_{\boldsymbol{\lambda}}(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{t-1})$, where $\boldsymbol{\lambda} = \{\lambda_k\} = \{(\mu_k, \sigma_k)\}$ parametrizes a set of object-centric Gaussian distributions in the latent space. We denote the number of iterations for the inner loop with L, and each iteration is indexed by l. The parameter update in the iterative inference is thus:

$$z_k^{t(l)} \overset{k}{\sim} q_{\lambda_k^{(l)}}(z_k^t|x^t, v^t, z_k^{t-1}) \tag{3.3}$$

$$\lambda_k^{(l+1)} \overset{k}{\leftarrow} \lambda_k^{(l)} + f_\Phi(z_k^{t(l)}, x^t, v^t, \mathbf{a}(\cdot)), \tag{3.4}$$

where the refinement function f_Φ, with trainable parameter Φ, is modeled by a recurrent neural network, e.g. LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014). The $\overset{k}{\sim}$ and $\overset{k}{\leftarrow}$ operators denote parallel operations over K independent object slots. The same auxiliary inputs as that of IODINE are computed to refine the posteriors. These computations are handled by a function $\mathbf{a}(\cdot)$, namely the "auxiliary function", which takes in the refinement function's inputs along with the posterior parameter $\lambda_k^{(l)}$.

Algorithm 1: MulMON at Test Time: Online Scene Learning

```
Input: Trained parameters $\Phi$, $\theta$
Hyperparams: $K$, $\sigma^2 = 0.01$, $L$      Init: $\boldsymbol{\lambda}^0 = \{\lambda_k^0\} \leftarrow \{(\mu_k = \mathbf{0}, \sigma_k = \mathbf{I})\}$;
/* The outer loop for scene learning */
for $t = 1$ to $T$ do
    Access a scene observation $(x^t, v^t)$;
    $\boldsymbol{\lambda}^{prior} = \boldsymbol{\lambda}^{t(0)} \leftarrow \boldsymbol{\lambda}^{t-1}$;
    /* The inner loop for observation aggregation */
    for $l = 0$ to $L-1$ do
        $\mathbf{z}^{t(l)} \sim \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)})$;
        $\{\mu_{xk}^{(l)}, \hat{m}_{xk}^{(l)}\} \leftarrow g_{\theta^2}(f_{\theta^1}(\mathbf{z}^{t(l)}, v^t))$;
        $\{m_k^{(l)}\} \leftarrow \mathbf{softmax}(\{\hat{m}_{xk}^{(l)}\})$;
        /* The spatial Gaussian mixture */
        $p_\theta(x^t | \mathbf{z}^{t(l)}, v^t) \leftarrow \sum_k m_k^{(l)} \mathcal{N}(x_k^t; \mu_{xk}^{(l)}, \sigma^2 \mathbf{I})$;
        if $l == 0$ then
            $\mathcal{L}_{\mathcal{T}}^{(l)} \leftarrow -\log p_\theta(x^t | \mathbf{z}^{t(l)}, v^t)$;
        else
            $\mathcal{L}_{\mathcal{T}}^{(l)} \leftarrow \mathcal{D}_{\mathrm{KL}}[\mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)}) || \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{prior})] - \log p_\theta(x^t | \boldsymbol{z}^{t(l)}, v^t)$;
        $\boldsymbol{\lambda}^{t(l+1)} \leftarrow \boldsymbol{\lambda}^{t(l)} + f_\Phi(z_k^{t(l)}, x^t, v^t, \mathbf{a}(\cdot))$;
    $\boldsymbol{\lambda}^t \leftarrow \boldsymbol{\lambda}^{t(l+1)}$;
```

3.3.4 Training

MulMON learns the decoder parameters θ and the refinement network parameters Φ by minimizing $\mathcal{D}_{\mathrm{KL}}[q_{\boldsymbol{\lambda}}(\mathbf{z}|x^{1:T}, v^{1:T}) || p_\theta(\mathbf{z}|x^{1:T}, v^{1:T})]$ for every *I.I.D.* data sample $(x^{1:T}, v^{1:T}) \sim P_{data}$. Such minimization is equivalent to maximizing the evidence lower bound (i.e. the ELBO, denoted as $\mathcal{L}$) in variational inference (see §2.1.3). However, besides maximizing the iterative ELBO like IODINE, we also simulate novel viewpoint-queried generation in the training process (similar to GQN, see Eslami et al. 2018). By asking MulMON to predict the appearance of a scene from unobserved viewpoints during training, we ensure that the iterative updates are indeed used to aggregate spatial information across views, as a complete 3D scene understanding is required to perform well. More formally, we randomly partition the set of T scene observations $\{(x^t, v^t)\}$ into two subsets $\mathcal{T}$ and $\mathcal{Q}$, with $n \sim \mathcal{U}(1,5)$ observations in $\mathcal{T}$ and the remaining $T-n$ observations in $\mathcal{Q}$. We perform scene learning on $\mathcal{T}$ and novel viewpoint-queried generation on $\mathcal{Q}$. We thus derive the MulMON ELBO (for one scene sample) as:

$$\mathcal{L} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbf{E}_{q_{\boldsymbol{\lambda}}(\mathbf{z}^t|\cdot)}[\log p_\theta(x^t|\mathbf{z}^t, v^t)] + \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q},\, t\sim\mathcal{T}} \mathbf{E}_{q_{\boldsymbol{\lambda}}(\mathbf{z}^t|\cdot)}[\log p_\theta(x^q|\mathbf{z}^t, v^q)] \\ - \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbf{IG}(\mathbf{z}^t, x^t; v^t, \mathbf{z}^{t-1}) \tag{3.5}$$

where $\mathbf{IG}$ is the information gain (aka. Bayesian surprise), the operation $|\cdot|$ computes the cardinality of a discrete set, and $q_{\boldsymbol{\lambda}}(\boldsymbol{z^t}|\cdot)$ is an abbreviation of the variational posterior $q_{\boldsymbol{\lambda}}(\mathbf{z}^t|x^t, v^t, \mathbf{z}^{t-1})$, from which we sample $\mathbf{z}^t$ by applying ancestral sampling. In practice, we use an efficient approximation of the information gain, i.e. an approximate $\mathbf{IG}$: $\mathcal{D}_{\mathrm{KL}}[q_{\boldsymbol{\lambda}}(\mathbf{z}|x^t, v^t, x^{1:t-1}, v^{1:t-1})||q_{\boldsymbol{\lambda}}(\mathbf{z}|x^{1:t-1}, v^{1:t-1})]$. Note that: **1)** making $\mathcal{T}$ and $\mathcal{Q}$ mutually complementary w.r.t. each scene data sample is to reduce the chance of MulMON memorizing the observed images (even though which is very low in a long run); **2)** using a fixed number of observations could harm the model's robustness at test time, hence why we randomly partition the observations into size-varying sets $\mathcal{T}$ and $\mathcal{Q}$ during training, i.e. we train the model with varying number of observations. See Appendix 3A for full details of the training algorithm of MulMON.

3.4 Related Work

Single-Object-Single-View **(SOSV)** A growing number of recent unsupervised learning advances have come in the form of "disentanglement" models (Higgins et al., 2017; Chen et al., 2016; Kim and Mnih, 2018; Mathieu et al., 2019) that explore feature-level decompositions by encouraging e.g. independence among latent variables. However, most of these models focus on a single view of a single object that has been placed in front of some background (e.g. dSprites, CelebA, 3D Chairs). As a result, they fail to **i)** capture object-level compositional structures that richly exist in natural scenes, and **ii)** accurately capture 3D scene information (i.e. resolve single-view spatial ambiguities and estimate, for example, rotation along the vertical axis).

***Multi-Object-Single-View* (MOSV)** To avoid the additional computational complexity of factorizing or segmenting the scene objects into explicit *multi-object* representations, many works have used pre-segmented images (Yao et al., 2018). However, this comes at the cost of decreased representational power (good object representation requires good object segmentation (Greff et al., 2019)) and a reliance on annotated data. In addition, these works struggle in a multi-view scenario where pre-segmented images require consistent multi-frame object registration and tracking, since the segmentation and representation models work independently. More recently, several works (Eslami et al., 2016; Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020b; Lin et al., 2019) have succeeded in approximating the target posterior $p(z_1, z_2, ..., z_K|x)$ within the VAE framework, achieving impressive unsupervised object-level scene factorization. However, being single-view models, they fall victim to single-view spatial ambiguities. As a result, they fail to accurately capture the scene's 3D spatial structure, causing problems for object-level segmentation. To overcome this and learn object-based representations that accurately capture 3D spatial structures, MulMON essentially extends these models to the multi-view scenario.

***Single-Object-Multi-View* (SOMV)** Recent unsupervised scene representation learning models, e.g. GQN (Eslami et al., 2018) and EGQN (Tobin et al., 2019), have shown success in aggregating multi-view scene observations into a single-slot representation that accurately captures the global spatial structure of the 3D scene. As a result, they can predict the appearance of a scene from unobserved viewpoints. However, being single-slot or “global-structure” models, they fail to achieve explicit object-level scene understanding in multi-object scenes, and as a result, miss out on the aforementioned benefits of object-centric scene representations. To overcome this, MulMON essentially extends these models to the case of multi-object representations. In addition, several works have sought explicit 3D representations either in the latent space (Rezende et al., 2016) or output space (Wu et al., 2016; Arsalan Soltani et al., 2017). However, due to the complexity of 1) working with explicit 3D object representations and 2) maintaining object correspondences across views, these works have been limited to single-object scenes (often quite simple, with “floating” objects placed in front of a plain background).

Multi-Object Scenes in Videos While some works in multi-object discovery and tracking in videos appear to be MOMV models (Kosiorek et al., 2018; Hsieh et al., 2018), they in fact work with one view per scene (abiding strictly by our definition of a scene in §3.2) and are only capable of dealing with binarizable MNIST-like images.

3.5 Experiments

Our experiments are designed to demonstrate that MulMON is a feasible solution to the MOMV problem, and to demonstrate that MulMON learns better representations than the MOSV and SOMV models by resolving spatial ambiguity. To do so, we compare the performance of MulMON against two baseline models, IODINE (Greff et al., 2019) (MOSV) and GQN (Eslami et al., 2018) (SOMV), in terms of segmentation, viewpoint-queried prediction (appearance and segmentation) and disentanglement (inter- and intra-object). To best facilitate these comparisons, we created two new datasets called CLEVR6-MultiView (abbr. CLE-MV) and CLEVR6-Augmented (abbr. CLE-Aug) which contain

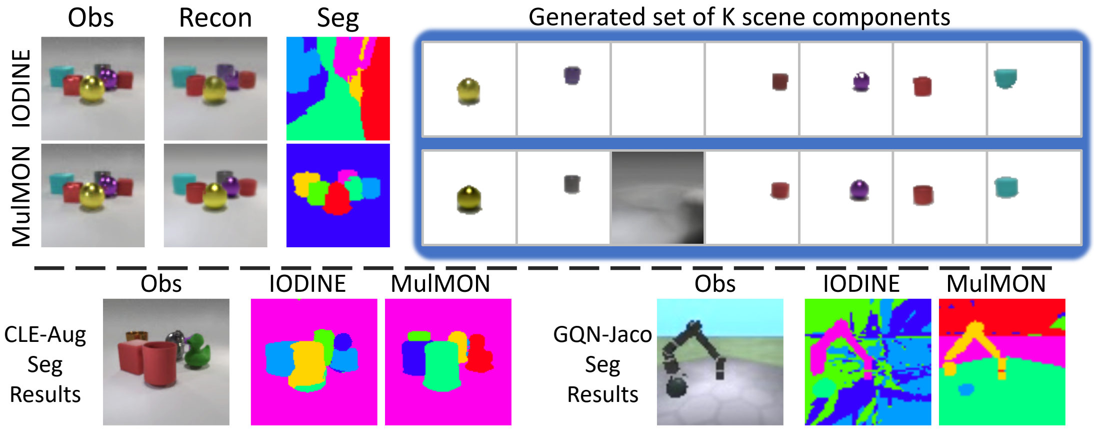


Figure 3.3: Qualitative comparison of MulMON vs. IODINE in terms of scene segmentation performance. **Top Left:** Reconstruction and segmentation comparison on a CLE-MV data sample. **Top Right:** Individual masked-object generation using each object's representation independently (see Appendix 3C for the computation details). **Bottom:** Segmentation performance on CLE-Aug and GQN-Jaco data samples (specific colors arbitrary).

ground-truth segmentation masks and shape descriptions (e.g. colors, materials, etc.). The CLE-MV dataset is a multi-view, observation-enabled variant (10 views per scene) of the CLEVR6 dataset (Johnson et al., 2017; Greff et al., 2019). The CLE-Aug adds more complex shapes (e.g. horses, ducks, and teapots etc.) to the CLE-MV environment. In addition, we compare the models on the GQN-Jaco dataset (Eslami et al., 2018) and use the GQN-Shepard-Metzler7 dataset (Eslami et al., 2018) (abbr. Shep7) for a specific ablation study. We train all models using an Adam optimizer with an initial learning rate 0.0003 for 300k gradient steps. In addition, all experiments were run across five different random seeds to simulate scenarios of different observation orders and view selections. For more details about the four datasets and model implementations see Appendix 3B and 3C respectively.

3.5.1 Scene Factorization

The ability of MulMON to perform scene object decomposition in the scene learning phase is crucial for learning object-centric scene representations. We evaluate its segmentation ability by computing mean-intersection-over-union (mIoU) scores between the output and the GT masks. However, since the segments produced by IODINE and MulMON are unordered, GT masks and object segmentation masks need to first be one-to-one registered for each scene. We solve this matching problem by first computing every possible object pairs of GT object masks and outputs, then, for each GT object mask, we find the output object mask that gives the highest IoU score. Table 3.1a shows that MulMON outperforms IODINE in object segmentation. The qualitative comparison in Figure 3.3 shows that IODINE captures each object well independently but fails to understand the spatial structure along depth directions (3D) – as described by the Categorical distribution (see §3.3.2). Note that IODINE's poor segmentation performance is mostly due to its poor handling of the background, i.e. its tendency to split up the background. Although the background is often considered a less-important "object", correct handling of the background demonstrates better spatial-reasoning ability. Together, all of these results suggest that MulMON learns better single-object representations and spatial structures by overcoming spatial ambiguities. It is also worth noting that both Table 3.1a and Figure 3.3 show a significant difference in IODINE's scene factorization performances on the CLE-MV data and the CLE-Aug data. I.e. IODINE factorizes CLE-Aug scenes better than CLE-MV scenes. We suspect the reason why IODINE factorizes CLE-Aug better is that CLE-Aug objects show less geometric symmetry than CLE-MV objects, which eases the identification of the compositional structure. Studying how geometric symmetry affects object factorization and model identification would be an interesting future work.

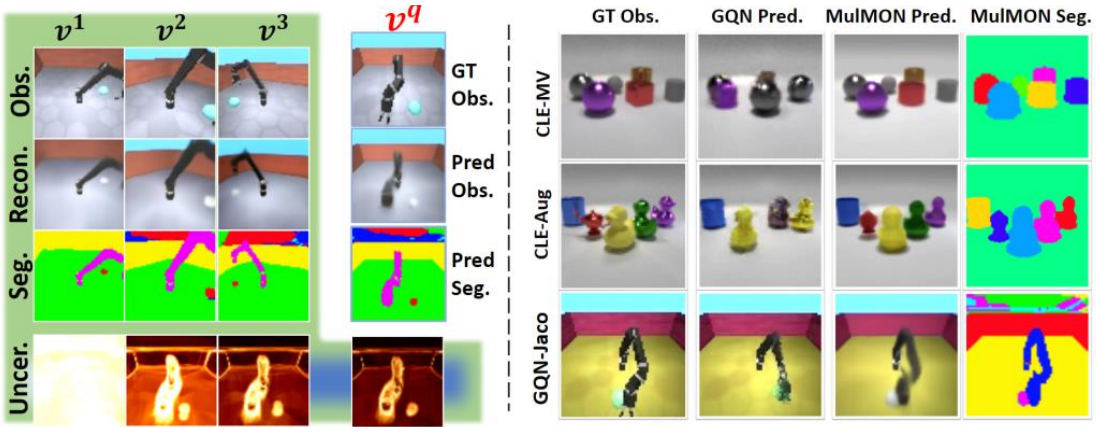


Figure 3.4: Qualitative results on novel-viewpoint prediction. **Left:** Working example of MulMON (columns show model after addition of new views), including uncertainty reduction (bottom row) across views, for a GQN-Jaco sample. **Right:** Qualitative comparison between MulMON and GQN.

3.5.2 Novel-viewpoint Prediction

MulMON can predict both observations and segmentation for novel viewpoints. This is the major advantage of our model (a MOMV model) over the MOSV and SOMV models in scene understanding. For our evaluation of online scene learning, each model is provided with 5 observations of each scene and then asked to predict both the observation and segmentation for randomly-selected novel viewpoints. We compute the root-mean-square error (RMSE) and mIoU as quality measures of the predicted observation and segmentation respectively. Table 3.1c shows that MulMON outperforms GQN on novel-view observation prediction. Table 3.1b shows that MulMON is the only model that can predict the object segmentation for novel viewpoints—and it does so with a similar quality to the original object segmentation (compare with Table 3.1a). However, as shown in Figure 3.4, GQN tends to capture more pixel details than MulMON, albeit at the risk of predicting wrong spatial configurations.

Models	CLE-MV	CLE-Aug
GQN	N/A	N/A
IODINE	0.1891 ± 0.0000	0.5137 ± 0.0007
MulMON	**0.7852 ± 0.0008**	**0.7076 ± 0.0004**

(a) Object Segmentation (mIoU)

Models	CLE-MV	CLE-Aug
GQN	N/A	N/A
IODINE	N/A	N/A
MulMON	**0.7845 ± 0.0011**	**0.6860 ± 0.0006**

(b) Predicted Object Segmentation (mIoU)

Models	CLE-MV	CLE-Aug	GQN-Jaco
GQN	0.1426 ± 0.0002	0.1482 ± 0.0001	0.1675 ± 0.0013
IODINE	N/A	N/A	N/A
MulMON	**0.0464 ± 0.0004**	**0.0733 ± 0.0003**	**0.1607 ± 0.0018**

(c) Predicted Observation RMSE (pixel avg.)

Models	Disent.	Compl.	Inform.
GQN	N/A	N/A	N/A
IODINE	0.47 ± 0.00	0.60 ± 0.01	0.67 ± 0.01
MulMON	**0.65 ± 0.01**	**0.73 ± 0.01**	**0.78 ± 0.00**

(d) Disent. on CLE-MV (DCI)

Table 3.1: Quantitative comparisons of MulMON, IODINE and GQN. "N/A" denotes cases where a model is *unable* to perform a task. In tables (a), (b) and (d), higher is better and 1 is best. For table (c), lower is better and 0 is best.

3.5.3 Disentanglement Analysis

To evaluate how well MulMON performs disentanglement at both the inter-object level and the intra-object level, we run disentanglement analyses on the representations learned by MulMON. For our qualitative analysis, we pick one of K objects in a scene, and traverse one dimension (by varying the value of the variable) of the learned object-representation at a time. Figure 3.5 (left) shows **1)** MulMON's intra-object disentanglement, encoding interpretable features in different latent dimensions; and **2)** MulMON's inter-object disentanglement, allowing single-object manipulation without affecting other objects in the scene. Figure 3.5 (right) shows that MulMON captures 3D information (vertical-axis rotation) and broadcasts consistent manipulations of this 3D information to different views. For our quantitative analysis, we employ the method of Eastwood and Williams (2018), i.e. DCI (see Appendix 3D.1), to compare the representations learned by each model on the CLE-MV and CLE-Aug datasets. As shown in Table 3.1d, MulMON learns object representations that are more disentangled,

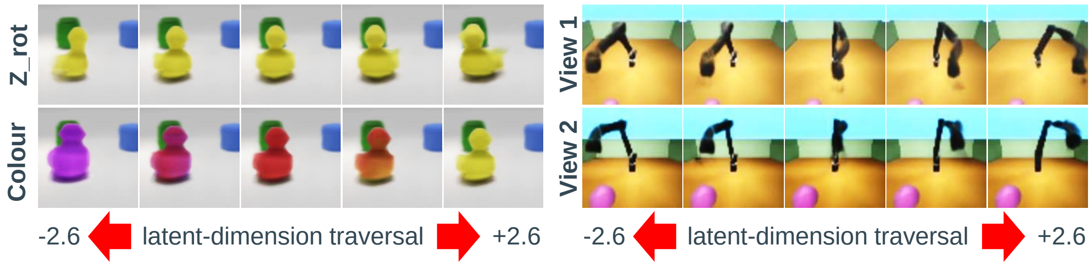


Figure 3.5: Single-object manipulations via latent traversals. **Left:** Traversing two dimensions of the duck's latent representation (one per row). Top row cropped for visual clarity. **Right:** For two different views (one per row), we manipulate the dimension of the learned representation that appears to capture vertical-axis rotation.

complete (compact) and informative (about ground-truth object properties). See Appendix 3D for further details.

3.5.4 Ablation Study

The Effects of The Number of Observations T We consider the number of observations T the most important hyperparameter of MulMON as the key insight of MulMON is to reduce multi-object spatial uncertainty by aggregating information across multiple observations. To visualize the effect of T on MulMON's performance, we plot MulMON's uncertainty about the scene as a function of T during testing. Specifically, for a given scene and ordering of the observations, we **1)** draw 10 samples from the approximate variational posterior $q_{\boldsymbol{\lambda}}(\mathbf{z}\,|x^{1:t}, v^{1:t})$ at each $t \in \mathbb{N}^{[1,T]}$, **2)** obtain the corresponding viewpoint-queried observation predictions using the 10 latent samples (see Section 3.3.2 and the right figure of Figure 3.2), **3)** compute the pixel-wise empirical variance over these observation predictions and average them over all scenes in the dataset and sampling 5 random view orderings (5 different random seeds). We show in Figure 3.6 (left) that MulMON effectively reduces the spatial uncertainty/ambiguity (measured by the average pixel-wise variance σ^2) by leveraging multiple views. In particular, Mul-

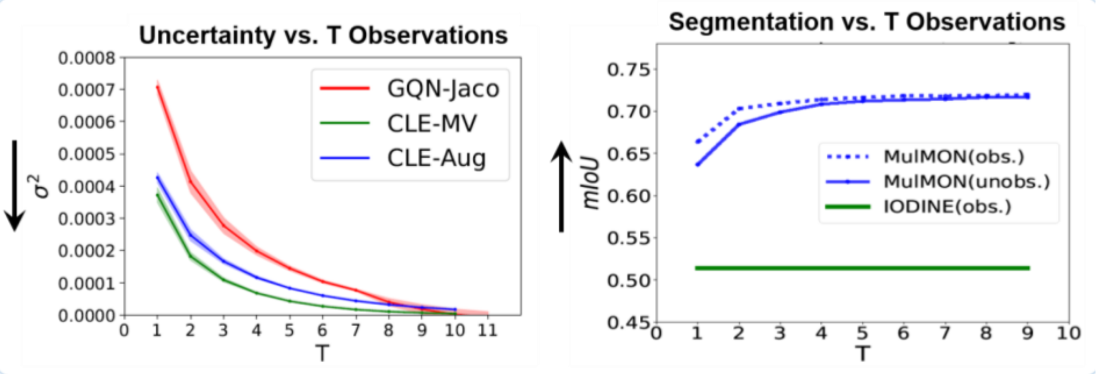


Figure 3.6: **Left:** Spatial uncertainty vs. T, where σ^2 (lower is better) denotes the average pixel-wise variance defined in the text. We show consistent results (trends) on three different datasets. **Right:** Scene factorization performance (segmentation mIoU, higher is better) vs. T on the CLE-Aug data. "MulMON (obs.)" and "IODINE (obs.)" tags MulMON and IODINE's performance in segmenting the observed images, respectively. "MulMON (unobs.)" tags MulMON's performance in predicting the segmentations for the unobserved (queried) views.

MON's uncertainty is rapidly reduced after only a small number of observations T. Moreover, we also show in Figure 3.6 (right) that the scene factorization performance improves as more observations are aggregated. In addition to the effect of T, we also study the effects of two other important hyperparameters, namely the globally-fixed number of object slots K and the coefficient of information gain **IG** (in the MulMON ELBO). For details on these further ablation studies, we refer the reader to Appendix 3D.

3.6 Conclusion

We have presented MulMON—a method for learning accurate, object-centric representations of multi-object scenes by leveraging multiple views. We have shown that MulMON's ability to aggregate information across multiple views does indeed allow it to better-resolve spatial ambiguity (or uncertainty) and better-capture 3D spatial structures, and as a result, outperform state-of-the-art models for unsupervised object segmentation. We have also shown that, by virtue of addressing the more complicated multi-object-multi-view scenario, MulMON achieves new functionality—the prediction of both appearance and object segmentations for novel viewpoints. The proposed design for multi-view uncertainty reduction and learning accurate scene representation can be useful in downstream tasks that involve active scene exploration and environment interaction. As all scenes in this paper are static, future work may look to extend MulMON to dynamic multi-object scenes.

Acknowledgements

This research was partly funded by the *Trimbot2020* project, which contributes to the European Union Horizon 2020 programme. The authors would like to thank Prof. C. K. I. Williams for his valuable advice that helped to improve this work and acknowledge the GPU computing support from Dr. Zhibin Li's Advanced Intelligence Robotics Lab at the University of Edinburgh.

Note: we use the same notations in the Appendix as that in the main chapter.

Appendix 3A. Training Algorithms of MulMON

We show the training algorithms of MulMON in Algorithm 2 & 3, where the **evaluate_likelihood** function is essentially Eqn. 3.2) and the **image_render** function is discussed in Appendix 3C. Implementation Details. All variables appeared in the algorithms are defined in the main chapter.

Algorithm 2: MulMON Training Algorithm

```
Data Training data $\mathbf{D} = \{(x^{1:T}, v^{1:T})\}_{1:N}$ (I.I.D. scenes)
Init trainable parameters $\Phi^{(0)}$, $\theta^{(0)}$, step count $s = 0$;
repeat
    Sample a mini batch $\{(x^{1:T}, v^{1:T})^{(m)}\}_{1:M} \sim \mathbf{D}$, where $M \leq N$;
    /* The below loop can go parallel as tensor operations */
    for $(x^{1:T}, v^{1:T})$ in $\{(x^{1:T}, v^{1:T})\}_M$ do
        $\mathcal{L}_m \leftarrow \textbf{SingleSampleELBO}((x^{1:T}, v^{1:T}), \Phi^{(s)}, \theta^{(s)})$;
    $\mathcal{L} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_m$;
    /* update */
    $\Phi^{(s+1)} \leftarrow \textbf{optimizer}(\mathcal{L}, \Phi^{(s)})$ ;
    $\theta^{(s+1)} \leftarrow \textbf{optimizer}(\mathcal{L}, \theta^{(s)})$;
    $s \leftarrow s + 1$;
until $\Phi, \theta$ converge;
```

Appendix 3B. Data Configurations

In this section, we discuss our **data configurations**. We show samples of the used datasets in Figure 3.7.

CLEVR-MultiView & CLEVR-Augmented We adapt the Blender environment of the original CLEVR datasets (Johnson et al., 2017) to render both

```
Algorithm 3: SingleSampleELBO
Input: A scene sample $(x^{1:T}, v^{1:T})$, trainables $\Phi$, $\theta$
Hyperparams $K$, $\sigma^2$, $L$
Data $\mathcal{T} = \{(x^t, v^t)\}, \mathcal{Q} = \{(x^q, v^q)\} \xleftarrow{random\ partition\ T} (x^{1:T}, v^{1:T})$
Init $\boldsymbol{\lambda}^0 = \{\lambda_k^0\} \leftarrow \{(\mu_k = \mathbf{0}, \sigma_k = \mathbf{I})\}$;
/* The outer loop for scene learning                                  */
for $t = 1$ to $|\mathcal{T}|$ do
    Access a scene observation $(x^t, v^t)$;
    $\boldsymbol{\lambda}^{prior} = \boldsymbol{\lambda}^{t(0)} \leftarrow \boldsymbol{\lambda}^{t-1}$;
    /* The inner loop for observation aggregation                     */
    for $l = 0$ to $L - 1$ do
        $\mathbf{z}^{t(l)} \sim \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)})$;
        $x^t \leftarrow$ render_image($\mathbf{z}^{t(l)}, v^t; \theta$) ;
        $p_\theta(x^t | \mathbf{z}^{t(l)}, v^t) \leftarrow$ evaluate_likelihood($x^t; \mathbf{z}^{t(l)}, v^t, \theta$) ;
        if $l == 0$ then
            $\mathcal{L}_{\mathcal{T}}^{(l)} \leftarrow -\log p_\theta(x^t | \mathbf{z}^{t(l)}, v^t)$;
        else
            $\mathcal{L}_{\mathcal{T}}^{(l)} \leftarrow \mathcal{D}_{\mathrm{KL}}[\mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)}) || \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{prior})] - \log p_\theta(x^t | \mathbf{z}^{t(l)}, v^t)$;
        $\boldsymbol{\lambda}^{t(l+1)} \leftarrow \boldsymbol{\lambda}^{t(l)} + f_\Phi(z_k^{t(l)}, x^t, v^t, \mathbf{a}(\cdot))$;
    $\boldsymbol{\lambda}^t \leftarrow \boldsymbol{\lambda}^{t(l+1)}$;
    $\mathcal{L}_{\mathcal{T}}^t \leftarrow \frac{2l+2}{L^2+L} \sum_l \mathcal{L}_{\mathcal{T}}^{(l)}$;
/* Viewpoint-queried prediction                                       */
for $(x^q, v^q)$ in $\mathcal{Q}$ do
    $\mathbf{z}^t \sim \mathcal{N}(\mathbf{z}^t; \boldsymbol{\lambda}^t)$;
    $x^q \leftarrow$ render_image($\mathbf{z}^t, v^q$) ;
    $p_\theta(x^q | \mathbf{z}^t, v^q) \leftarrow$ evaluate_likelihood($x^q; \mathbf{z}^t, v^q, \theta$) ;
    $\mathcal{L}_{\mathcal{Q}}^q \leftarrow -\log p_\theta(x^q | \mathbf{z}^t, v^q)$;
/* Compute the MulMON ELBO                                            */
$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum_t \mathcal{L}_{\mathcal{T}} + \frac{1}{|\mathcal{Q}|} \sum_q \mathcal{L}_{\mathcal{Q}}$;
Output: $\mathcal{L}$
```

datasets. We make a scene by randomly sampling $3 \sim 6$ rigid shapes as well as their properties like poses, materials, colors etc.. For the CLEVR-MultiView

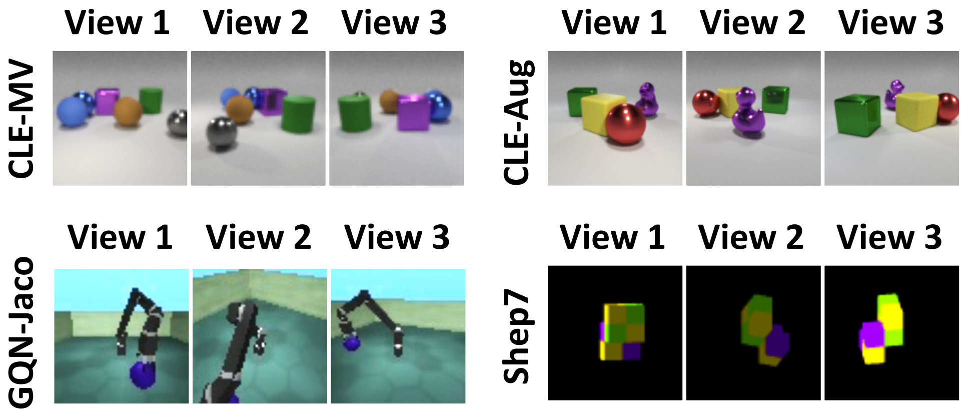


Figure 3.7: Examples of the four dataset used in this work.

(CLE-MV) dataset, we sample shapes from three categories: cubes, spheres, and cylinders, which are the same as the original CLEVR dataset. For the CLEVR-Augmented (CLE-Aug), we add more shape categories into the pool: mugs, teapots, ducks, owls, and horses. We render 10 image observations for each scene and save the 10 camera poses as 10 viewpoint vectors. We use resolution 64×64 for the CLE-MV images and 128×128 for the CLE-Aug images. All viewpoints are at the same elevation level but different azimuth with their focuses locked at the scene center. We thus parametrize a viewpoint 3-D viewpoint vector as $(\cos\alpha, \sin\alpha, r)$, where α is the azimuth angle and r is the distance to the scene center. In addition, we save the object properties (e.g. shape categories, materials, and colors etc.) and generate objects' segmentation masks for quantitative evaluations. CLEVR-MultiView (CLE-MV) contains 1500 training scenes, 200 testing images. CLEVR-Augmented (CLE-Aug) contains 2434 training scenes and 500 testing scenes.

GQN-Jaco We use a mini subset of the original GQN-Jaco dataset (Eslami et al., 2018) in our paper. The original GQN-Jaco contains 4 million scenes, each of them contains 20 image observations (resolution: 64×64) and 20 corresponding viewpoint vectors (7D). To reduce the storage memory and accelerate the training, we randomly sample $2,000$ scenes for training and 500 scenes for testing. Also, for each scene, we use only 11 observations (viewpoints) that are randomly sampled from the 20 observations of the original dataset.

GQN-Shepard-Metzler7 Same as the GQN-Jaco dataset, we make a mini GQN-Shepard-Metzler7 dataset (Eslami et al., 2018) (Shep7) by randomly selecting 3000 scenes for training and 200 for testing. Each scene contains 15 images observations (resolution: 64×64) with 15 corresponding viewpoint vectors (7D). We use Shep7 to study the effect of K on our model.

Appendix 3C. Implementation Details

In this section, we introduce the implementation details of our experiments. We show our *training configurations* in Table 3.2 and the *architectural design* of MulMON in Tables 3.3, 3.4, & 3.5

Table 3.2: Training Configurations

Type	MulMON, IODINE, GQN
Optimizer	Adam
Initial learning rate η_0	$3e^{-4}$
Learning rate at step s	$\max\{0.1\eta_0 + 0.9\eta_0 \cdot (1.0 - s/6e^5), 0.1\eta_0\}$
Total gradient steps	300000
Batch Size$^{.}$	8 for CLE-MV, CLE-Aug, 16 for GQN-Jaco, 12 for Shep7

* GQN scheduler with a faster attenuation rate

Decoder-Output Processing For a single view of a scene, our decoder g_θ outputs K $3 \times H \times W$ RGB values (i.e. $\{x_k\}$ as in Eqn. 3.2 of the main chapter) along with K $1 \times H \times W$ mask logits (denoted as $\{\hat{m}_k\}$). H and W are the image sizes, i.e. height and width. In this section, we detail the computation of rendering K individual scene components' images, segmentation masks, and

Table 3.3: Model State Space Specifications

Type	CLE-MV	CLE-Aug	GQN-Jaco	Shep7
z_dims	16	16	32	16
v_dims	3	3	7	7

z_dims: the dimension of a latent representation

v_dims: the dimension of a viewpoint vector

Table 3.4: MulMON Refinement Network with Trainable Parameters 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*z_dims		
	LSTMCell	128		
	Linear	128	Linear	output $\Delta\lambda$

z_dims: the dimension of a latent representation

v_dims: the dimension of a viewpoint vector

* see Greff et al. (2019) for more details about the auxiliary inputs

LSTMCell channels: the dimensions of the hidden states

reconstructed scene images. We compute the individual scene objects' images as:

$$x_k \overset{k}{\leftarrow} \mathbf{sigmoid}(\hat{m}_k) \cdot x_k.$$

As shown in Figure 3.8, this overcomes mutual occlusions of the objects since the **sigmoid** functions do not impose any dependence on K objects. We compute the segmentation masks as:

$$m_k \overset{k}{\leftarrow} \mathbf{softmax_k}(\hat{m}_k).$$

Table 3.5: MulMON Decoder Network with Trainable Parameters θ

Parameters	Type	Channels (out)	Activations.	Descriptions
θ^1 (view transformer)	Input	z_dims+ v_dims		$z_k \sim \mathcal{N}(z_k; \lambda_k)$, v
	Linear	512	Relu	
	Linear	z_dims	Linear	$\tilde{z}_k = f_{\theta_1}(z_k, v)$
θ^2 (Generator)	Input	z_dims		$\tilde{z}_k = f_{\theta_1}(z_k, v)$
	Broadcast	z_dims+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	rgb mean (μ_{xk}) + mask logits ($\hat{m}_k$)

z_dims: the dimension of a latent representation

v_dims: the dimension of a viewpoint vector

*: see spatial broadcast decoder in Watters et al. (2019)

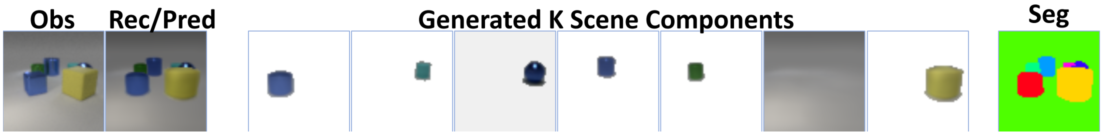


Figure 3.8: Example output of the decoding process. **Left to right:** GT scene observation, (predicted/reconstructed) scene observation, generated K individual scene components (white background for visual clarity), segmentation map. The generated scene components overcome/impute occlusions (e.g. the purple glossy sphere).

To generate binary segmentation masks, we take **argmax** operation over the K $\hat{m}$ at every pixel location and encode the maximum indicator (indices) using one-hot codes. We render a scene image using a composition of all scene objects as:

$$x = \sum_k \mathbf{softmax_k}(\hat{m}_k) \cdot x_k = \sum_k m_k \cdot x_k$$

Appendix 3D. Additional Results

3D.1 Disentanglement Analysis

Table 3.6: Disent. on CLE-Aug (DCI, higher is better)

Models	Disent.	Compl.	Inform.
GQN	N/A	N/A	N/A
IODINE	0.54	0.48	0.21
MulMON	**0.63**	**0.54**	**0.58**

To compare quantitatively the intra-object disentanglement achieved by MulMON and IODINE, we employ the framework and metrics (DCI) of Eastwood and Williams (2018). Specifically, let $\mathbf{z}^{GT}$ be the values of ground-truth generative factors $\mathbf{Z}^{GT}$ of a multi-object scene, where each $Z_k^{GT} \in \mathbf{Z}^{GT}$ defines an object and each segment, e.g. a single dimension $Z_{ki}^{GT} \subset Z_k^{GT}$, defines an object feature, e.g. color. Following Eastwood and Williams (2018), we learn a mapping from $\mathbf{Z}$ to $\mathbf{Z}^{GT}$ with random forests in order to quantify the disentanglement, completeness and informativeness of the learned object representations. The results on the CLE-MV dataset is presented in §3.5.3, and here we present the results on the CLE-Aug dataset. As shown in Table 3.6, MulMON again outperforms IODINE, learning representations that are more disentangled, complete and informative (about ground-truth factor values). It is worth noting the significant gap in informativeness in Table 3.6. This strongly indicates that the object representations learned by MulMON are more accurate, i.e. capturing object properties better.

3D.2 Ablation Study

Novel-View Prediction vs. The Number of Observations T We discussed in §3.5.4 that the spatial uncertainty decrease (Figure 3.6, left) suggests a boost

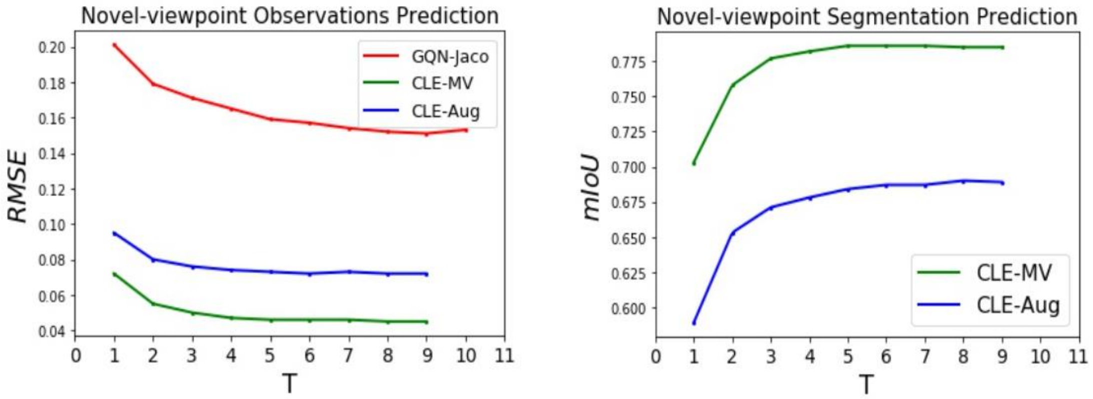


Figure 3.9: Ablation study of the effects of T. **Left:** the performance of novel-view synthesis improves as T increases. **Right:** the segmentation prediction performance boosts as T increases.

of task performance (see Figure 3.9 & 3.6 (right)). This suggests that MulMON does leverage multi-view exploration to learn more accurate scene representations (than a single-view configuration), which also explains the performance gap (between IODINE and MulMON) shown in Figure 3.6 (right). Here we show more detailed results on models' performance vs T in novel-view synthesis. We employ mIoU (mean intersection-over-union) and RMSE (root-mean-square error) to measure MulMON's performance on observation prediction and segmentation prediction respectively. To further demonstrate the advantage that MulMON has over both IODINE and GQN, we compare their performance in terms of both segmentation and novel-view appearance prediction, as a function of the number of observations given to the models. Figure 3.10 shows that: **1)** MulMON significantly outperforms IODINE even with a single view, likely due to a superior 3D scene understanding gained during training (figures on the left), **2)** Despite the more difficult task of achieving object-level segmentation, MulMON closely mirrors the performance of GQN in predicting the appearance of the scene from unobserved viewpoints (figures on the right), **3)** MulMON achieves similar performance in scene segmentation from observed and unobserved viewpoints, and the difference diminishes as the number of views increase (see dashed lines vs.

solid lines in the left-hand figures).

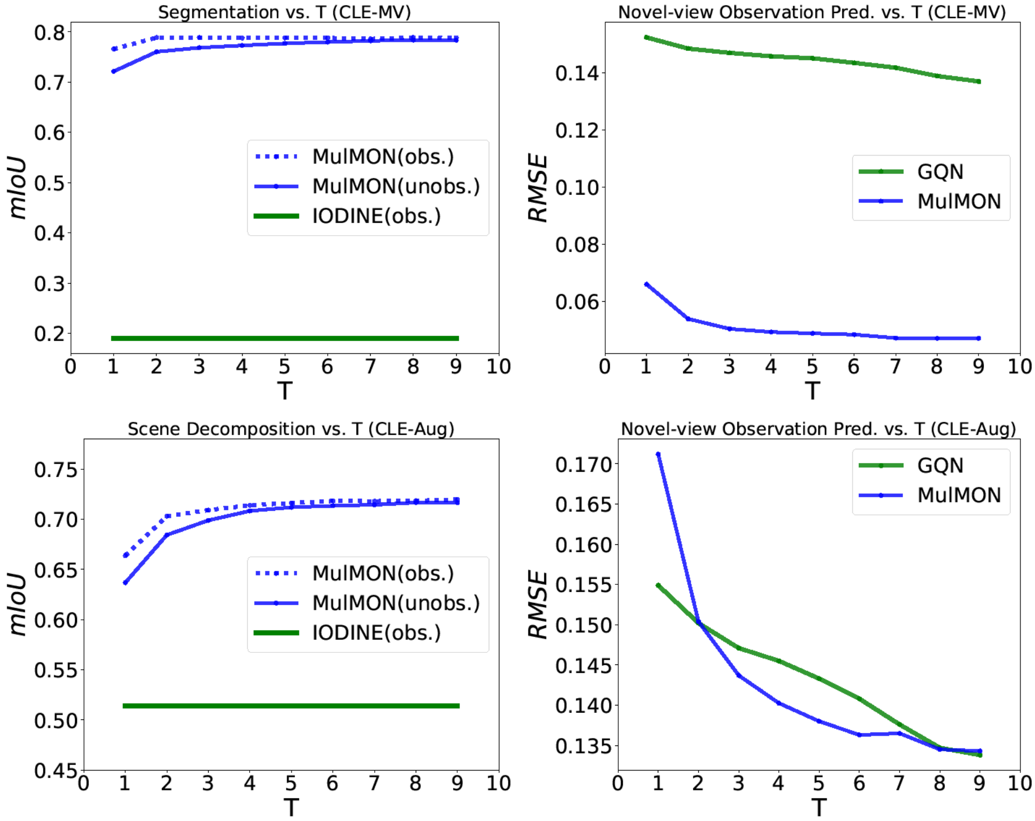


Figure 3.10: Performance comparison w.r.t. number of observations T. (Top left) Segmentation performance vs. number of observations T on CLE-MV dataset. Note that “obs” means that MulMON reconstructs the observed images (scene appearances) and “unobs” means that MulMON predicts the appearance of the scene from unoberserved viewpoints. (Top right) Novel-viewpoint appearance prediction performance vs. number of observations given to the models on CLE-MV dataset. (Bottom left) Segmentation performance vs. number of observations T on CLE-MV dataset. (Bottom right) RMSE of appearance predictions for unobserved viewpoints vs. number of observations on CLE-Aug dataset.

Task Performance vs. The Number of The Object Slots K Although an explicit assumption about the number of objects in a scene is not required for MulMON, selecting an appropriate K (i.e. the number of object slots) is crucial

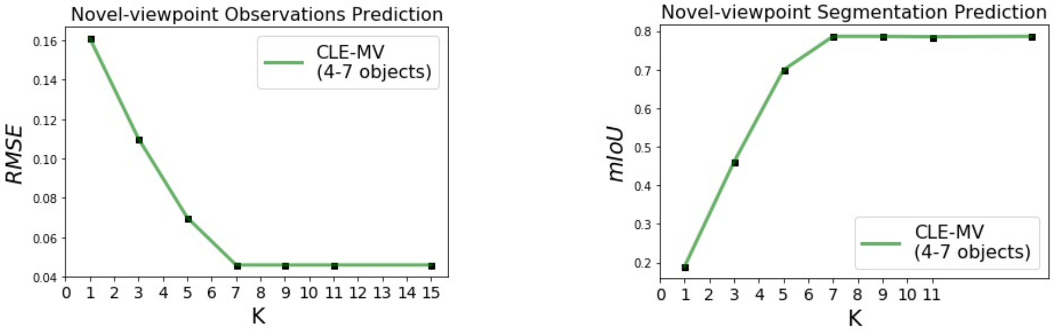


Figure 3.11: The effect of K. *An insufficient number* of the object "slots" leads to higher RMSE (left) and lower mIoU (right). Using a large K, larger than the "sufficient" number (i.e. 7 in this case, 6 objects + 1 background), does not improve the task performance significantly.

to have MulMON work correctly. In the main chapter, we discussed that "K needs to be sufficiently larger than the number of scene objects" and we show the experimental support here. We trained our model on CLE-MV, where each scene contains 4 to 7 objects including the background, with K set to 9, and tested the model on novel-viewpoint prediction task using various K. Figure 3.11 shows that, for both observation prediction and segmentation prediction tasks, the model's performance improves as K increases until reaching a critical point at $K = 7$, which is the maximum quantity of scene objects in the dataset. Therefore, one should select a K that is always greater or equal to the maximum number of objects in a scene. When this condition is satisfied, further increases in K will mostly not affect MulMON's performance. However, subtle cases in terms of K's selection do exist. As shown in Figure 3.12, instead of treating the Shep7 scene as a combination of a single object and the background, MulMON performs as a part segmenter that discovers the small cubes and their spatial composition. This is because, in the training phase of MulMON, the amortized parameters Φ and θ are trained to capture the object features (possibly disentangled) shared across all the objects in the whole dataset instead of each scene with specific

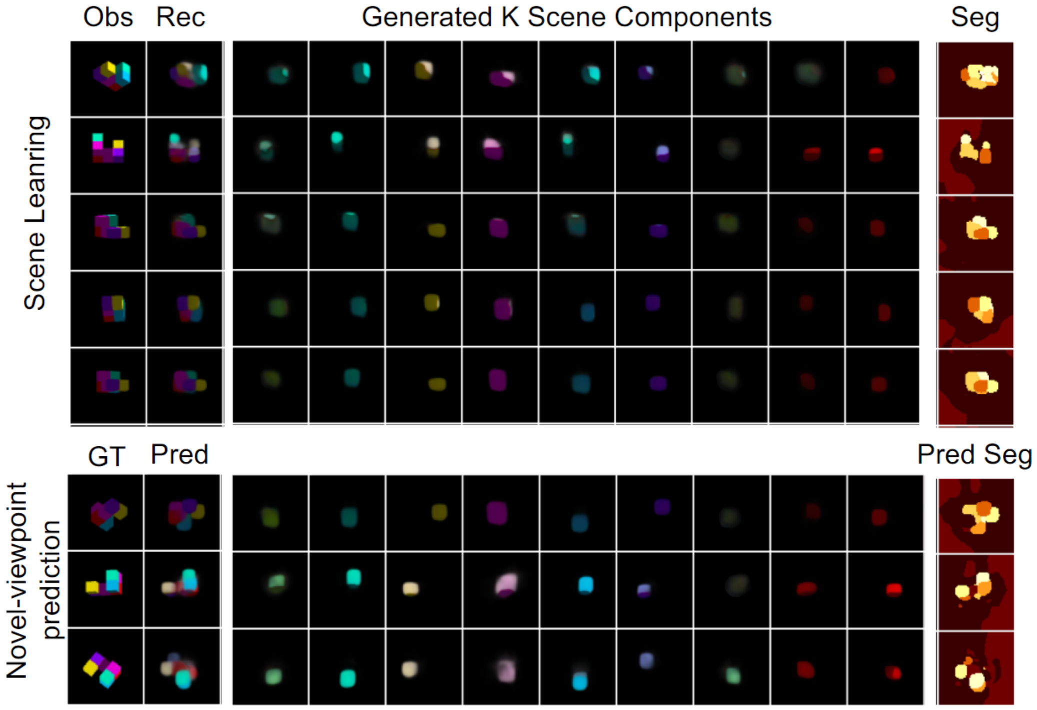


Figure 3.12: MulMON on Shep7. MulMON treats an Shep7 object as composition of parts (cubes) instead of a complete object.

objects. These shared object features are what drives the segmentation process of MulMON. In Shep7, what is being shared are the cubes, the object itself is a spatial composition of the cubes. The results on Shep7 shown in Figure 3.12 illustrate the granularities and subjectiveness of perceptual grouping and leaves much space for us to study in the future.

Scene Learning vs. The IG Coefficient In the MulMON ELBO, we fix the coefficient of the information gain at 1. In testing, we consider this coefficient controls the scene learning rate (uncertainty reduction rate). We denote the coefficient as $\alpha_{\mathbf{IG}}$ hereafter. According to the MulMON ELBO (to maximize), the negative sign of the **IG** term suggests that a larger value of the coefficient leads to less information gain (spatial exploration). To verify this, we try four different $\alpha_{\mathbf{IG}}$ (0.1, 1.0, 10.0 and 100.0) and track the prediction uncertainty as observations are acquired (same as our ablation study of T). The results in Figure 3.13 verifies our assumption the scene learning rate: larger $\alpha_{\mathbf{IG}}$ leads to slower

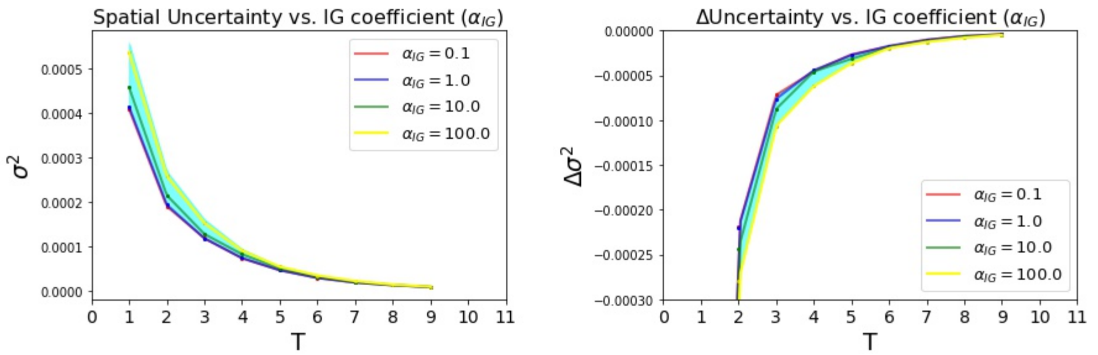


Figure 3.13: Scene learning rate vs. $\mathbf{IG}$ coefficient (denoted as $\alpha_{\mathbf{IG}}$). **Left:** Uncertainty reduction gets slightly slower when we increase $\alpha_{\mathbf{IG}}$. **Right:** The computed uncertainty change rate or scene learning rate (lower means slower) shows larger $\alpha_{\mathbf{IG}}$ slightly slows the scene learning.

scene learning and vice versa.

3D.3 Compositional Generalization

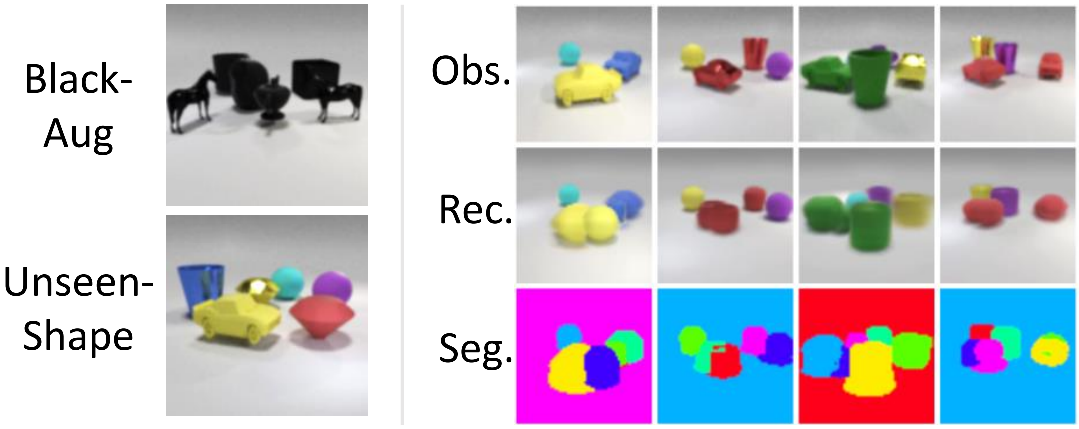


Figure 3.14: **Left:** Example images of the modified CLE-Aug, i.e. BlackAug & UnseenShape. **Right:** Qualitative results of MulMON on the UnseenShape data.

To evaluate MulMON's compositional generalization ability, we trained MulMON, IODINE and GQN on CLE-Aug. Then, we compared their performance on *O.O.D.* data from CLE-MV and two modified CLE-Aug datasets—namely, Black-Aug and UnseenShape (see Figure 3.14, left). Black-Aug keeps all the configurations of CLE-Aug objects, except the shapes are painted black. We created UnseenShape by replacing all the shapes of CLE-Aug with novel shapes (cups, cars, spheres, and diamonds) while keeping all the other settings.

Table 3.7: MulMON's generalization performance.

Tasks	Models	CLE-Aug (train)	CLE-MV	Black-Aug	UnseenShape
Seg.	IODINE	0.51 ± 0.001	0.61 ± 0.002	0.50 ± 0.006	0.51 ± 0.004
(mIoU)	MulMON	**0.71 ± 0.000**	**0.71 ± 0.004**	**0.67 ± 0.002**	**0.64 ± 0.004**
Pred.Obs	GQN	0.15 ± 0.000	**0.15 ± 0.001**	**0.24 ± 0.003**	**0.17 ± 0.002**
(RMSE)	MulMON	**0.07 ± 0.000**	0.16 ± 0.002	0.26 ± 0.002	0.21 ± 0.006
Disent.	IODINE	0.54, 0.48, 0.21	0.14, 0.12, 0.26	0.2, 0.26, 0.27	0.13, 0.12, 0.26
(D,C,I)	MulMON	**0.63, 0.54, 0.68**	**0.52, 0.48, 0.63**	**0.55, 0.55, 0.66**	**0.5, 0.47, 0.67**
Pred.Seg (mIoU)	MulMON	**0.69 ± 0.001**	**0.71 ± 0.004**	**0.68 ± 0.005**	**0.60 ± 0.005**

Table 3.7 shows the comparison results of "*non-object-centric* model (GQN) vs. MulMON" and "*single-view model* (IODINE) vs. MulMON" in terms of generalization. We can see from Table 3.7 that MulMON, as a *multi-view-object-centric* model, generalizes best in most of the subtasks. The disentanglement and segmentation comparisons between IODINE and MulMON suggests that the multi-view scheme does allow an OCRL model to discover the compositional structures around objects. Though Figure 3.14 (right) does show that MulMON indeed leverages the knowledge of composition to handle the UnseenShape data, it it is surprising to see that GQN generalizes slightly better than MulMON in the novel-view observation prediction task—which seems to contradict our assumption that the understanding of compositionality leads to better generalization. In fact, though GQN is *not* an OCRL model, it does somewhat discover the feature-level compotitional structures (Eslami et al., 2018) like most disentanglement models (Higgins et al., 2017; Kim and Mnih, 2018). As the *O.O.D.* data like Black-Aug and UnseenShape are "out-of-distribution" in the sense of fea-

tures, we conjecture that GQN (feature→scene composition) is more sensitive to feature-level compositionality than MulMON (feature→object→scene composition) hence generalizes better in feature changes. Our speculation is backed by the generalization experiments of IODINE in object-level scene compositional changes—Greff et al. (2019) trained IODINE on scenes with $3-5$ objects and discovered that, as an OCRL model, IODINE generalized well in scenes with more than 5 objects (see (Greff et al., 2019), Figure 9). These generalization results along with the Shep7 results (see Figure 3.12) leads us to an interesting direction of future research—*granularity* of compositionality.

3D.4 Random Scene Generation

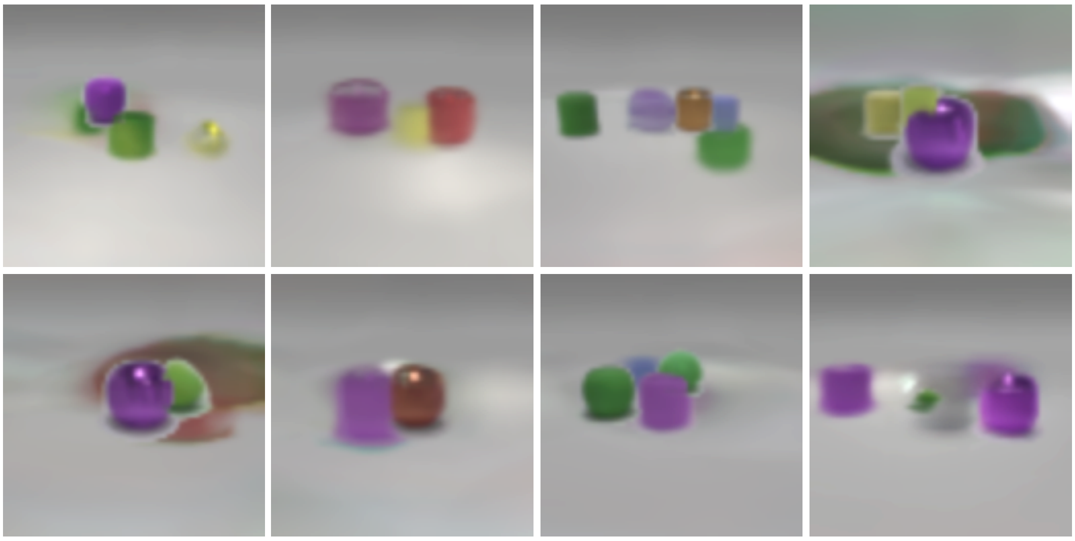


Figure 3.15: Random scene generation samples.

As a generative model, MulMON can generate random scenes by composing independently-sampled objects. However, to focus on forming accurate, disentangled representations of multi-object scenes, we must assume objects are *I.I.D.* and thus ignore inter-object dependence—e.g. two objects can appear at the same location. We nevertheless show random scene examples generated by MulMON (trained on the CLE-MV dataset) in Figure 3.15. We can see that MulMON generates mostly good object samples by randomly composing different features but does not take into account the global layout of the objects. As a result, the generated scene images contain odd backgrounds and are somewhat fuzzy in terms of occlusions. This can be aided by modeling the scene global prior using the confounder variable(s) C (see Figure 2.3 in §2.2.3) as suggested by Reddy et al. (2022), or modeling the inter-object dependence like Engelcke et al. (2019), or combining both top-down and bottom-up inference schemes (Emami et al., 2021).

Chapter 4

LDS: Latent Duplicate Representation Suppression

Generative object-centric scene representation learning is crucial for structural visual scene understanding. Built upon variational autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014), current approaches infer a set of latent object representations to interpret a scene observation (e.g. an image) under the assumption that each part (e.g. a pixel) of a scene observation must be explained by one and only one object of the underlying scene. Despite the impressive performance these models achieved in unsupervised scene factorization and representation learning, we show empirically that they often produce duplicate scene object representations which directly harms the scene factorization performance. In this chapter, we address the issue by introducing a differentiable prior that explicitly forces the inference to suppress duplicate latent object representations. The extension is evaluated by adding it to three different unsupervised scene factorization approaches. The results show that the models trained with the proposed method not only outperform the original models in scene factorization and have fewer duplicate representations, but also achieve better variational posterior approximations than the original models.

This chapter is an extended version of the paper "*Duplicate Latent Representation Suppression for Multi-object Variational Autoencoders*" (Nanbo and Fisher, 2021), published at *The British Machine Vision Conference* (2021).

4.1 Introduction

Variational autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014) have become a powerful tool for unsupervised visual scene understanding and representation learning. As a particular type of latent-variable generative model, a VAE model not only inherits the ability to explain scene observations (e.g. images) by learning a marginal distribution $p(x;\theta)$ over the observations $X \in \mathbb{R}^M$ but it also allows to describe and represent the observed scenes in a more compact latent space $\mathbf{Z} \in \mathbb{R}^D$ $(D \ll M)$ for simplicity and efficiency. A rising trend in VAE research is to treat a multi-object scene as a composition of objects (a.k.a. scene components), i.e. a scene representation $\mathbf{Z}$ is a set of scene object representations $\mathbf{Z} = \{Z_k\}$, where each Z_k corresponds to one and only one object in the scene. These object-based scene representation learning models are often referred to as the *multi-object VAEs*, they are called *component VAEs* (abbr. CompVAEs) in this chapter for simplicity.

By making an assumption that each pixel of a scene image observation must be explained by one and only one object in the scene, recent CompVAE advances (Burgess et al., 2019; Greff et al., 2019; Nanbo et al., 2020) show great success in *unsupervised image segmentation* and *object-centric representation learning* (OCRL). In these models, this assumption acts as a constraint to force different scene object components, i.e. different $Z_k \in \{Z_k\}$, to capture different image pixels—we expect these models to infer a set of distinctive Z_k. However, all three CompVAEs investigated here (i.e. MONET (Burgess et al., 2019), IODINE (Greff et al., 2019), MulMON (Nanbo et al., 2020)) can infer duplicate latent object representations (see Figure 4.1 for an example), which violates the implicit assumption and thus harms their performance in scene factorization (or image segmentation).

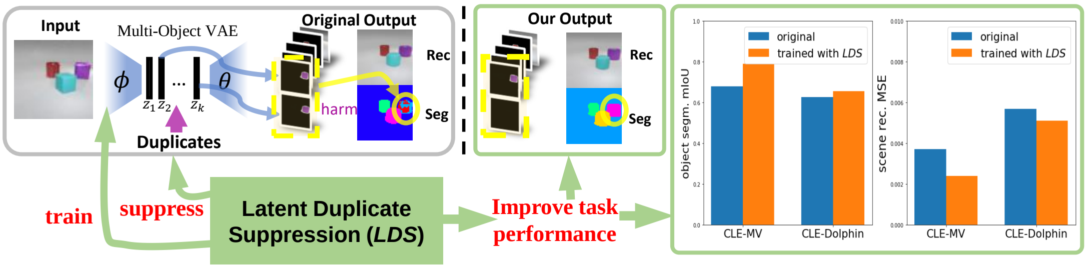


Figure 4.1: **Top left (the grey box):** The state-of-the-art unsupervised scene factorization and image segmentation approaches, i.e. multi-object VAE models, often infer duplicate latent object representations that harm the scene object segmentation performance. **Bottom left (the green diagram):** We propose a differentiable latent-duplicate-suppression prior (abbr. *LDS*) to train better multi-object-VAE inference networks that suppress the duplicates. **Middle & right:** Multi-object VAEs that trained with the proposed *LDS* achieves better scene object segmentation (e.g. higher mIoU on 2 datasets) and observation reconstruction performance (lower MSE).

In this chapter, we refer to the issues raised by inferring duplicate latent representations as the *uniqueness issues* and the implicit assumption of $\forall Z_k \in \{Z_k\}$ being unique as the *uniqueness assumption*. To address the *uniqueness issues*, we propose a differentiable prior, namely for *latent duplicate suppression* (abbr. *LDS*), to train the CompVAEs' inference network to suppress duplicates while making inference at test time. The *LDS* prior essentially implements the *uniqueness assumption* — two identical components cannot appear in the same scene representation set $\mathbf{Z}$, i.e. penalizing highly-similar latent object representation pairs during training.

In our experiments, we train two representative single-view CompVAEs, i.e. MONET (Burgess et al., 2019) and IODINE (Greff et al., 2019), and one multi-view CompVAE, i.e. MulMON (Nanbo et al., 2020), with *LDS* as the experimental group and train the same models without *LDS* as the control group. We show the effectiveness of training CompVAEs' with *LDS* in suppressing scene factorization duplicates and achieving better variational approximation by comparing the performance of the two groups of models. We claim and demonstrate that training a CompVAE with the proposed *LDS* prior enables the CompVAE to: **1)** Produce better scene factorizations with fewer duplicate objects (see §4.4.2). **2)** Learn better scene representations that supports better scene observation reconstructions (see §4.4.2). **3)** Achieve better variational posterior approximation, i.e. decrease the *inference gap* (Cremer et al., 2018) (see §4.4.3).

4.2 Method

Our goal is to enable CompVAEs' inference networks to suppress duplicates when making inferences at test time. Our approach is to introduce a differentiable prior, i.e. the *LDS* prior, as an additional constraint to train the CompVAEs' inference models. In §4.2.1, we briefly review the general construction of CompVAEs. In §4.2.2, we present the *LDS* prior and how to train a CompVAE model with it. In §4.2.3, we discuss CompVAEs' suboptimality and define a measure for the comparison of two posterior approximations.

4.2.1 Background

Similar to VAEs, a CompVAE model often consists of a generative model and an inference model. The generative likelihood of a scene image observation in a CompVAE is often modeled as a spatial Gaussian mixture (Williams and Titsias, 2004; Greff et al., 2017) parametrized by θ:

$$p_\theta(x|\{z_k\}) = \prod_{i=1}^{M} \sum_{k=1}^{K} p_\theta(C_i = k|z_k) \cdot \mathcal{N}(x_{ik}; g_\theta(z_k), \sigma^2), \tag{4.1}$$

where i indexes a pixel location (M in total) and x_{ik} is the RGB value of the k-th object at the location i. RGB values are samples of $\mathcal{N}(x_{ik}; g_\theta(z_k), \sigma^2)$ where $g_\theta(\cdot)$ is a decoder network and the standard deviation σ is set to a fixed value, e.g. $\sigma = 0.1$, for all pixels. The generated K RGB values x_{ik} compete to explain a location i as an instance of object k. The objects and their likelihoods, i.e. the mixing coefficients, are captured by a categorical distribution $p_\theta(C_i = k|z_k)$, where $C_i = k$ denotes the event of object k winning the "competition". Note that this formulation is similar to that used in MulMON (Nanbo et al., 2020), but that approach investigated multi-view problems, where viewpoints V were taken as conditions. We refer the reader to Appendix 4B for more details about the image-generating process.

To tackle the problems of scene factorization and OCRL, the inference model of a CompVAE infers a joint posterior of the scene objects $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_K\}$. Although CompVAEs encode a fixed number (K) of object slots for the inferred object representations, they do not make any assumption about the number of objects in a scene. Ideally, one can use as many object slots as possible—leaving some redundant slots unused. In practice, a K that is slightly larger than the number of scene objects is often chosen for efficient computation. Though such practice sidesteps the assumption of knowing the number of scene objects, we discovered that CompVAEs often use the redundant slots to create duplicates. Based on the independence assumption commonly used for scene object factorization, the inference problem is solved by computing a tractable variational approximation:

$$q_\Phi(\mathbf{z}|x) = q_\Phi(z_1, z_2, \ldots, z_k|x) = \prod_{k=1}^{K} q_\Phi(z_k|x, *), \tag{4.2}$$

where Φ denotes the trainable amortized parameters Kingma and Welling (2013) and $*$ denotes other conditions (e.g. $z_{1:k-1}$, see Engelcke et al. 2019). Note that Eqn. 4.2 describes a CompVAE inference process in a general form that holds for many existing CompVAE variants.

4.2.2 Latent Duplicate Suppression

The goal of the proposed *LDS* prior is to penalize duplicates during the training process so the trained models infer fewer duplicate object representations during testing. In other words, we want to train a Φ that better suppresses duplicates. Because CompVAEs use fixed numbers (K) of object slots for the inferred latent representations, we can easily construct a fixed-size pair-wise similarity matrix, $\Sigma \in \mathbb{R}^{K \times K}$ using a kernel function. In this chapter, we use the cosine kernel function to compute the similarities between any two latent object representations. If we write the a set of object latent values in a matrix form (with exchangeable horizontal entries) as $S = [z_1, z_2, ..., z_K]^T \in \mathbb{R}^{K \times D}$, the computation of the similarity matrix can be written as: $\Sigma = SS^T/(||S_r|| \cdot ||S_c^T||)$, where $||S_r||$ and $||S_c^T||$ compute the Euclidean norms for matrix S and S^T's row and column vectors respectively. The self-similarities of the inferred objects are captured by the constructed Σ's diagonal elements, and the mutual similarities are captured by Σ's off-diagonal elements. To suppress duplicates, we need to penalize high off-diagonal similarities, i.e. by maximizing the *LDS* prior:

$$\mathcal{L}_{LDS}(\{z_k\}; \Phi) = \sum_{h=1}^{K} \sum_{,j=1,h \neq j}^{K} \log \mathcal{N}(\Sigma_{h,j}; 0, \sigma^2). \tag{4.3}$$

The log normal density regulates its measure to a smaller range and σ (which models small variation in the similarity values) is fixed globally at 0.1. As both VAEs and CompVAEs are variational Bayesian models, their training relies on maximizing their evidence lower bounds (abbr. ELBO, denoted as $\mathcal{L}_{ELBO}(x; \Phi, \theta)$) w.r.t. the two trainable parameters Φ and θ. Taking a CompVAE model, we thus train it by maximizing:

$$\mathcal{L}(x;\Phi,\theta) = \mathcal{L}_{ELBO}(x;\Phi,\theta) + \lambda \cdot \mathcal{L}_{LDS}(\{z_k\};\Phi), \tag{4.4}$$

where λ is a Lagrange multiplier (set to default: 1). In general, combining Eqn. 4.1 & 4.2 leads to a general formulation of CompVAE ELBO: $\mathcal{L}_{ELBO}(x;\Phi,\theta) = \mathbf{E}_{q_\Phi(\{z_k\}|x)}[\log p_\theta(x|\{z_k\})] - \mathcal{D}_{\mathrm{KL}}(q_\Phi(\{z_k\}|x)|p_\theta(\{z_k\}))$. However, the exact formulations for a specific CompVAE is model-dependent. For example, MulMON (Nanbo et al., 2020) uses a multi-view ELBO. It is important to note that, though it is possible to apply the *LDS* prior in the inference stage as a post-processing technique, we use it only in training. This is because post-inference suppression implies hard decisions on filtering the inferred representations, which risks mistakenly deleting important explanatory components.

4.2.3 CompVAE Suboptimality Measure

In this chapter, we use superscripts + and 0 on a variable to indicate if it is related to the experimental group (CompVAEs trained with *LDS* prior) or the control group (original CompVAEs). To validate that after suppressing duplicate object representations, the CompVAE models less often violate the *uniqueness assumption* and approximates better the variational posterior $p(\{z_k\}|x)$, i.e. $q_{\Phi^+}(\{z_k\}|x)$ becomes a better approximation than $q_{\Phi^0}(\{z_k\}|x)$ with respect to $p(\{z_k\}|x)$, we need a measure to quantify approximation qualities and thus support model comparisons. Through the derivation of VAEs' ELBO (Kingma and Welling, 2013), a gap between the observed evidence $\log p_\theta(x)$ and the ELBO $\mathcal{L}_{ELBO}(x;\Phi,\theta)$ is illustrated:

$$\mathcal{D}_{\mathrm{KL}}(q_\Phi(\mathbf{z}|x)\|p_\theta(\mathbf{z}|x)) = \log p_\theta(x) - \mathcal{L}_{ELBO}(x;\Phi,\theta) \geq 0. \tag{4.5}$$

This is referred to as the *inference gap* of VAEs (Cremer et al., 2018), which provides a quantitative measure of how good is an approximation. Similarly, we formulate $\mathcal{G} = \mathcal{D}_{\mathrm{KL}}(q_\Phi(\{z_k\}|x) \| p_\theta(\{z_k\}|x))$ as the approximation quality measure for a CompVAE. Therefore, by comparing $\mathcal{G}^+$ and $\mathcal{G}^0$ we can determine if the experimental group reaches better suboptimality than the control group.

In practice, because $\log p_\theta(x)$ is inaccessible, $\mathcal{G}$ is not computable (see Eqn. 4.5). We thus approximate $\log p_\theta(x)$ with a Monte Carlo estimate — the importance weighting estimate (Burda et al., 2016), where the sample size (denoted as B) is set to 500. Therefore, we can compute the *inference gap* $\mathcal{G}$ as:

$$\begin{aligned} \mathcal{G} &= \mathcal{D}_{\mathrm{KL}}(q_\Phi(\{z_k\}|x) \| p_\theta(\{z_k\}|x)) \\ &= \mathbf{E}_{\mathbf{z}^1,\ldots,\mathbf{z}^b \sim q_\Phi(\mathbf{z}|x)}[\log \frac{1}{B} \sum_{b=1}^{B} \frac{p_\theta(x, \mathbf{z}^b)}{q_\Phi(\mathbf{z}|x)}] - \mathcal{L}_{ELBO}(x; \Phi, \theta) \geq 0. \end{aligned} \tag{6}$$

to simplify the discussion hereafter, we define a measure *inference gap drop* (denoted as $\Delta\mathcal{G}^+$) using $\mathcal{G}^0$ and $\mathcal{G}^+$: $\Delta\mathcal{G}^+ = \mathcal{G}^0 - \mathcal{G}^+$. In general, a positive $\Delta\mathcal{G}^+$ suggest a smaller gap is achieved and thus provides better approximation, a negative $\Delta\mathcal{G}^+$ suggests the opposite. In our experiments, we use $\Delta\mathcal{G}^+$ as an important metric for our model suboptimality analysis (see §4.4.3).

4.3 Related Work

Our work lies in the research area of unsupervised scene factorization and representation learning. Earlier works in this area like the Attend-Infer-Repeat (AIR) model (Eslami et al., 2016) and its variants (Hsieh et al., 2018; Kosiorek et al., 2018) perform object-centric scene factorization by sequentially searching for one object at a time in the image plane until all objects in the image are captured. As these models do not target a 3D understanding of a scene (without assuming a known 3D render), they cannot resolve occlusions and handle images with complex backgrounds. The problem is overcome by recent advances (Burgess et al., 2019; Engelcke et al., 2019; Greff et al., 2019; Nanbo et al., 2020) that the pixel-level compositions of scene objects, i.e. each pixel needs to be explained by one and only one scene component. This line of work is referred to as the *scene-mixture* models by Lin et al. (2019) as they all use the spatial mixture models (Williams and Titsias, 2004; Greff et al., 2017) to explain the image observations of scenes. This allows the models to reason about depth and occlusions which are essential for 3D understanding.

Our work is also related to relational reasoning works that are built upon Comp-VAEs. We discuss them in two categories: implicit and explicit relational reasoning. Although aforementioned works such as Engelcke et al. (2019); Greff et al. (2019); Nanbo et al. (2020) do not explicitly reason about relationships, the discovery of scene objects suggests mutual dependence of each other. **These models violate the implicitly-introduced *uniqueness assumption* and thus cannot suppress duplicate object representations, while we aim at fixing these issues in this work.** There are unsupervised scene factorization models that handle explicit relations among the inferred objects, e.g. R-NEM (van Steenkiste et al., 2018), STOVE (Kossen et al., 2020) and G-SWM (Lin et al., 2020). They focus on dynamics modeling and define "relations" as the interactions of the scene objects and thus differ from the problem solved in this chapter, which concerns relations between the inferred representations in a global layout sense. A recent work, i.e. GENESIS (Engelcke et al., 2019), which models the global layout of scene objects explicitly, is perhaps the closest to us in terms of scene-object relational reasoning.

The proposed work is related to the *duplicate removal* or *non maximum suppression* (abbr. NMS) idea that is widely used across many computer-vision tasks such as edge detection (Rosenfeld and Thurston, 1971) and feature extraction (Lowe, 2004). Among all the applications, NMS's usage in object detection is the closest to ours, where duplicate detection candidates will be removed or suppressed (Rothe et al., 2014; Bodla et al., 2017) based on a quantifiable criterion, e.g. detection confidence. However, as NMS in these models works as a post-processing technique so it cannot handle the mistakes a model made in the inference stage. Also, the violation of the *uniqueness assumption* by the aforementioned CompVAEs can lead to a worse variational approximation of the VAE posterior (Cremer et al., 2018), which is worse than what the traditional duplicate-removal techniques achieve.

4.4 Experiments

Our experiments are based on two datasets: CLE-MV (Nanbo et al., 2020) and Dolphin. The Dolphin dataset is synthesized using CLE-MV's graphics engine by adding more complex and general shapes (e.g. dolphins, horses, ducks, etc.). There are in total 1700 and 3631 different scenes in the CLE-MV and the Dolphin datasets respectively and each scene consists of 3-6 objects including the background (a trivial object). As there are 10 image observations (with size 64×64) taken from 10 different viewpoints, both the two datasets support multi-view tasks. We thus randomly select 1500 scenes (15000 images) from CLE-MV and 3000 scenes (30000 images) from Dolphin to make the training sets. At test time, we sample 160 unseen scenes (i.e. 1600 images) from CLE-MV and 200 unseen scenes (2000 images) from Dolphin, where "unseen scenes" denote scenes that are not in the training sets. Note that we use multi-view CLEVR datasets instead of the original CLEVR (Johnson et al., 2017) because we want to show that the proposed method works for both single-view and multi-view scenarios. For the experiments, we use three baseline CompVAE models including two single-view models, i.e. MONet and IODINE, and a multi-view model MulMON, and create our experimental group with the three CompVAEs trained with the proposed *LDS* prior. We train all models using the same training specifications as that of the experimental group except for removing the *LDS* prior. We thus study

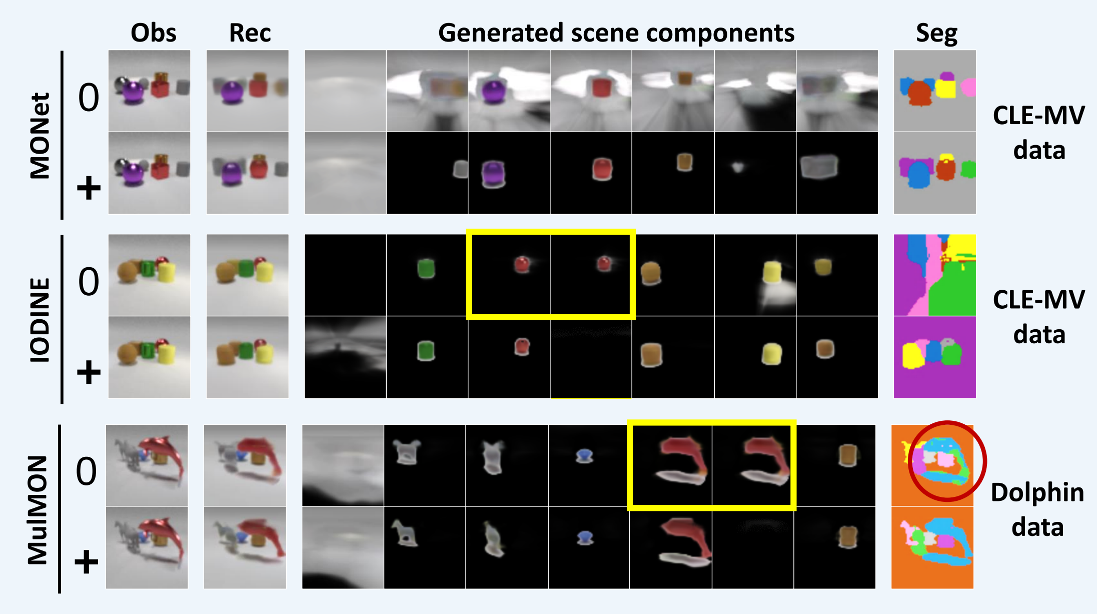


Figure 4.2: Qualitative comparisons between the experimental group (tagged with "+") and the control group (tagged with "0"). The Obs column is a source image, Rec is the corresponding reconstructed image based on the inferred representation. The next 7 columns show the independent generation of the inferred scene components (order not important). The Seg column shows the pixel label for the component with highest probability (pixel color is not important). **Top:** Training with *LDS* aids the original MONet model which suffers from local minima: obtains fair factorization and reconstruction while fails to learn clean object geometries and thus generates noisy scene components whereas MONet$^+$ produces cleaner inferred components. **Middle:** Training with *LDS* aids IODINE: resolves duplicates (circled in yellow) and fixes the weak background segmentation, as shown by the large colored regions in the Seg column, which is a known issue of IODINE (Greff et al., 2019). **Bottom:** Training with *LDS* allows MulMON to suppress duplicates and thus produce a better segmentation map. (Colored boxes and circles highlight the duplicates and failures caused by them.)

and demonstrate the effectiveness by comparing the two groups in various aspects. We refer the reader to the Appendix for the ablation study and the model specifications.

4.4.1 Duplicate Suppression

The first set of experiments justify the proposed *LDS* methods by demonstrating its effects on suppressing duplicates. We ran both the control-group and experimental group models on the 200 CLE-MV test scenes (2000 images) to get two quantitative measures: **1)** the average pair-wise similarities (see Eqn. 4.3, denoted as $\overline{\mathbf{Sim}}$) among all the inferred latent object representations, **2)** the percentage of images for which object duplicates were inferred. To better visualize the effect of the proposed *LDS* on reducing latent-object-representation similarities, we used the difference between average pair-wise similarities of the control- and experimental-group models, i.e. $\Delta\overline{\mathbf{Sim}}^{+} = \overline{\mathbf{Sim}}^{0} - \overline{\mathbf{Sim}}^{+}$, where a positive $\Delta\overline{\mathbf{Sim}}^{+}$ suggests positive effect of *LDS* in suppressing latent object replicates. For the second measure, we randomly picked 100 images and counted the total number of image cases where duplicates were produced. The results in Figure 4.3 suggest that the proposed *LDS* prior works effectively reduces latent-object-representation similarities and suppresses duplicate representations.

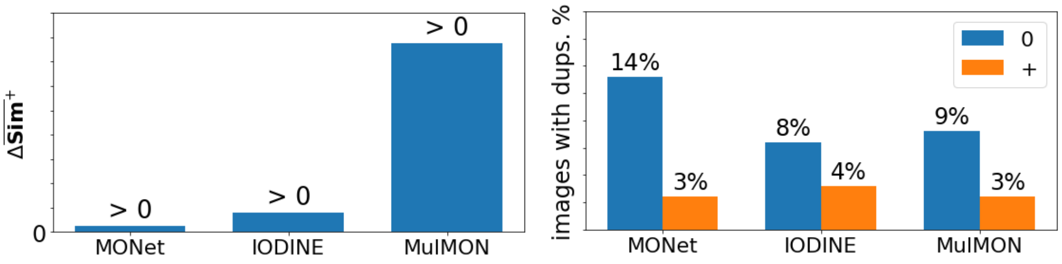


Figure 4.3: Effectiveness of the proposed *LDS* in duplicate suppression. **Left** All of the three tested CompVAEs give positive $\Delta\overline{\mathbf{Sim}}^{+}$ values, where positive $\Delta\overline{\mathbf{Sim}}^{+}$ suggests smaller similarities (i.e. improvements) of the experimental-group (trained with *LDS*) latent object representations than that of the control group. **Right** Direct comparison between the experimental- (tagged with $^{+}$) and control-groups (tagged with ^{0}) in duplicate suppression. The lower percentages when using LDS mean fewer duplicates and thus effective duplicate suppression.

4.4.2 Task Performance

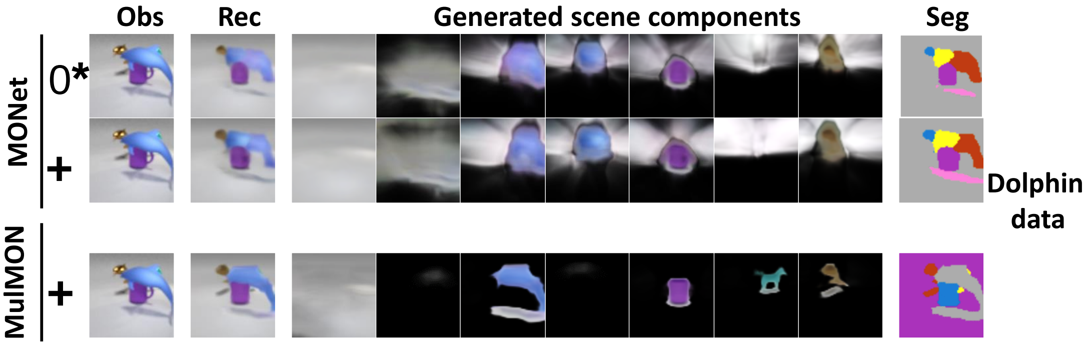
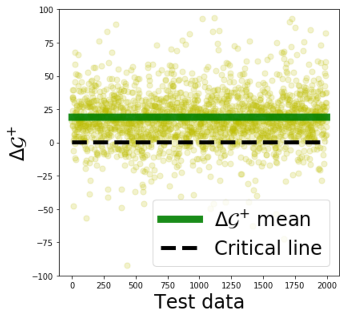


Figure 4.4: A partial-failure example from the "outlier" model (MONet^0) on Dolphin (tagged with "$\star$" in Table 4.1). **Top:** The model produces good factorization but fails badly to learn good-quality object representations and thus shows noisy generations. The proposed *LDS* fails to fix it. **Bottom:** A good example shown by a model that achieves similar quantitative performance (MulMON$^+$).

		CLE-MV		Dolphin	
Models	*LDS*	MSE↓	mIoU↑	MSE↓	mIoU↑
MONet	0	0.0037 ± 0.0002	0.6806 ± 0.0072	$\star\mathbf{0.0059 \pm 0.0002}$	$\star\mathbf{0.6620 \pm 0.0070}$
	+	$\mathbf{0.0024 \pm 0.0002}$	$\mathbf{0.7899 \pm 0.0092}$	0.0063 ± 0.0005	0.6567 ± 0.0077
IODINE	0	$\mathbf{0.0016 \pm 0.0002}$	0.1911 ± 0.0042	0.0054 ± 0.0001	0.3501 ± 0.0043
	+	0.0020 ± 0.0001	$\mathbf{0.7252 \pm 0.0054}$	$\mathbf{0.0050 \pm 0.0002}$	$\mathbf{0.6224 \pm 0.0052}$
MulMON	0	0.0019 ± 0.0001	0.7834 ± 0.0046	0.0055 ± 0.003	0.6246 ± 0.0056
	+	$\mathbf{0.0019 \pm 0.0001}$	$\mathbf{0.7911 \pm 0.0043}$	$\mathbf{0.0051 \pm 0.0002}$	$\mathbf{0.6556 + \pm 0.0027}$

Table 4.1: Quantitative comparisons between the experimental group (tagged with "+") and the control group (tagged with "0"). All results are averaged over five different random seeds. $\star$ denotes the most significant case where *LDS* does not generate obvious improvements which we will discuss in the text.

Scene Factorization The biggest advantage of CompVAEs over traditional VAEs in visual scene understanding is that they can perform unsupervised scene factorization, which directly links to observation segmentation. Therefore, we



Pearson Correlations Coefficients	
Reconstruction MSE vs. $\Delta\mathcal{G}^+$	-0.2874
Segmentation mIoU vs. $\Delta\mathcal{G}^+$	0.2019

Table 4.2: **Left figure:** The effect of *LDS* on the variational approximation quality: yellow dots represents the $\Delta\mathcal{G}^+$ for each test data sample (2000 test images), and the green line is the mean $\Delta\mathcal{G}^+$, which is the change in the ELBO (evidence lower bound) value from Eqn. 4.5. Positive values are improvements. Observe that most dots lie above the "no improvement" line at 0, demonstrating that *LDS* generally produces improvements. **Right table**: The correlation between the task performance and the *inference gap*. 0 suggests no correlation and +1/-1 denotes the strongest positive/negative correlation. The right table exhibits a negative correlation between the *inference gap drop* $\Delta\mathcal{G}^+$ (ie. bigger $\Delta\mathcal{G}^+$ correlates with lower errors) and the reconstruction errors and a positive correlation between the segmentation accuracy (mIoU) and $\Delta\mathcal{G}^+$ (ie. bigger $\Delta\mathcal{G}^+$ correlates with better segmentation).

compared the scene object decomposition performance between the experimental group (CompVAEs trained with *LDS*) and control group (original CompVAEs) on scene object decomposition task. Because both the CLE-MV and Dolphin datasets are synthesized with the ground-truth segmentation maps, we can thus compute the *mean intersection over union* (mIoU) score as the performance measure. To solve the bipartite matching problem as the output object masks (in a list) are not in the same order as the GT masks, we used the Hungarian matching algorithm to find the best match that maximizes the mIoU score for a scene. Table 4.1 shows that the experimental group, i.e. CompVAEs trained with the proposed *LDS* prior, results in similar or improved performance compared to the control group over most models and datasets. Figure 4.2 demonstrates the effectiveness of the proposed *LDS* prior in reducing duplicates and aiding CompVAEs' local minimas. We also examined the "outlier model", i.e. MONet$^+$ trained on Dolphin, and some output samples are shown in Figure 4.4. For the outlier model, even though the quantitative measures are improved, the model still suffers from the local minima. We also consider this a failure instance of the proposed *LDS* as it does *not* aid the model like it does to MONet trained on the Dolphin dataset (see Figure 4.2).

Scene Reconstruction Reconstruction quality reflects the representation-learning quality of a VAE model. Hence, we compared the experimental group and the control group also on reconstruction quality using the *mean squared error* (MSE) between the observation image and the reconstruction image as our quantitative measure. The MSE was computed from the RGB vector distances, where color values are on a $[0, 1]$ scale. Table 4.1 shows that the proposed *LDS* improves not only the scene factorization but also the scene reconstruction. This suggests the proposed *LDS* helps CompVAEs to learn better scene representations.

4.4.3 Suboptimality Analysis

As shown in Figure 4.2, the proposed *LDS* prior not only suppressed the object replicates, but it also fixed several issues (uniqueness and degenerated inference) that exist in the original CompVAEs, improving scene reconstruction quality. The suboptimality analysis presented in this section gives a better understanding of how the proposed *LDS* helps to improve the task performance. To verify our hypothesis that the proposed *LDS* reduced the violation of the *uniqueness assumption* and thus achieved a better variational approximation of the target posterior $p_\theta(\mathbf{z} = \{z\}|x)$ and improved the task performance, we studied: **1)** the effect of the proposed *LDS* on the variational approximation quality, and **2)** the correlation between the task performance (mIoU) and the variational approximation quality. We evaluated the variational approximation quality by computing the *inference gap drop* $\Delta\mathcal{G}^{+}$ (see §4.2.3) for the 2000 test images from the CLE-MV dataset and averaged the $\Delta\mathcal{G}^{+}$ over 2000 samples to obtain the mean $\Delta\mathcal{G}^{+}$. Table 4.2 (left figure) shows the drop $\Delta\mathcal{G}^{+}$ of these 2000 test samples and their mean. As illustrated, MONet trained with the proposed *LDS* produces a positive drop $\Delta\mathcal{G}^{+}$ — the proposed *LDS* reduces the *inference gap* and is thus a better approximation than the original model. We computed the Pearson correlation coefficients between the task performance measures, i.e. MSE (for reconstruction) and mIoU (for segmentation), and $\Delta\mathcal{G}^{+}$ on the 2000 test samples. As shown in Table 4.2 (right table) an increased *inference gap drop* $\Delta\mathcal{G}^{+}$ does indeed decrease the reconstruction error (negative correlation) and increase the segmentation accuracy (positive correlation). However, we also admitted that a better suboptimality does not completely explain the performance boost given the degrees of correlation shown in Table 4.2. It thus requires future investigations on other potential causes of the performance boost.

4.5 Conclusion

In this chapter, we present a differentiable prior that leverages similarity measures to regulate the object-centric latent representations inferred by multi-object VAEs, i.e. CompVAEs. Despite its simplicity, we demonstrate its effectiveness in fixing known issues, namely the *uniqueness issues*, of the multi-object VAE models — inferring duplicate object representations. We ascribe the *uniqueness issues* to the violation of the *uniqueness assumption* that is implicitly introduced by the scene-mixture-model assumption, i.e. each part of an scene observation (e.g. a pixel) must be explained by one and only scene object. Therefore, we demonstrate through experiments that, by suppressing duplicates, better variational approximation and task performance can be achieved. Regarding the future research, we are particularly interested in modelling more flexible and possibly learnable similarity functions, e.g. a similarity measure that can distinguish explicitly the inter-object correlations' effect on each dimension of an object's latent representation and thus weight them accordingly.

Acknowledgements

The authors would like to thank the reviewers of the original paper for the insightful suggestions. This research was partly funded by the Trimbot2020 project, which contributes to the European Union Horizon 2020 programme.

Note: we use the same notations in the Appendix as that in the main chapter.

Appendix 4A. Implementation Details

Training specifications Table 4.3, 4.4 & 4.5 gives the training configurations of MONet, IODINE and MulMON respectively. Note that **1)** for IODINE and MulMON, which use iterative inference modules, we apply LDS per iterative step to compute their ELBOs during training, and **2)** for all CompVAEs, we apply LDS only in their training times.

Model Architecture Specifications As discussed in the main chapter, we use three existing CompVAE models as our baselines and build our contributions on top of these architectures. It is important to use the same architectures as the original papers. However, we found it difficult to use a latent dimension of 64 as in Greff et al. (2019) for the CLEVR-based datasets as it trains too slowly, over one week for one run on two RTX2080TI, we thus reduced the dimension of IODINE to 16 for our IODINE. As constructing the proposed *LDS* prior requires no model architecture design and architecture parameter tweaking, we refer to the original papers of MONet (Burgess et al., 2019), IODINE (Greff et al., 2019), and MulMON (Nanbo et al., 2020) for the architecture details.

Appendix 4B. CompVAE Rendering Process

Figure 4.5 shows the CompVAE rendering process we used to produce all qualitative results presented in this chapter. **Importantly, we used softmax functions to compute the compositional probabilities of each components, i.e. the mixing probabilities in Eqn. 4.1, to render the whole scene, and sigmoid functions to render independent objects**. However, one might also

Table 4.3: Training Configurations For MONet^0 and MONet$^+$

Type	the trainings of MONet^0 and MONet$^+$
Optimizer	RMSprop
Initial learning rate η_0	$3e^{-4}$
Batch size	40 (unit: images)
Learning rate at step s	N/A
Total gradient steps	$600k$
Gradient-norm clipping	5.0
log-normal likelihood strength	1.0
KL (Gaussian prior) strength β	0.5
KL (attention prior) strength	0.5
LDS (MONet$^+$ only) strength	0.5

Table 4.4: Training Configurations of IODINE^0 and IODINE$^+$

Type	the trainings of IODINE^0 and IODINE$^+$
Optimizer	Adam
Initial learning rate η_0	$1e^{-4}$
Batch size	8
Learning rate at step s	$\star\max\{0.1\eta_0 + 0.9\eta_0 \cdot (1.0 - s/1e^6), 0.1\eta_0\}$
Total gradient steps	$600k$
Gradient-norm clipping	5.0
inference iterations (Greff et al., 2019)	5
log-normal likelihood strength	1.0
KL (Gaussian prior) strength β	1.0
LDS (IODINE$^+$ only) strength	1.0

$\star$: same scheduler as GQNs'.

see independent component rendering with other functions in the related literature, e.g. IODINE (Greff et al., 2019) uses a linear mapping of x_k to render independent components.

Table 4.5: Training Configurations of MulMON^0 and MulMON$^+$

Type	the trainings of MulMON^0 and MulMON$^+$
Optimizer	Adam
Initial learning rate η_0	$2e^{-4}$
Batch size	8
Learning rate at step s	$\star\max\{0.1\eta_0 + 0.9\eta_0 \cdot (1.0 - s/1e^6), 0.1\eta_0\}$
Total gradient steps	$600k$
Gradient-norm clipping	5.0
inference iterations (Greff et al., 2019)	5
log-normal likelihood strength	1.0
KL (Gaussian prior) strength β	1.0
LDS (IODINE$^+$ only) strength	1.0

$\star$: same scheduler as GQNs'.

Appendix 4C. Additional Results

4C.1 Ablation Study

The ablation study focuses on two hyperparameters: **1)** the standard deviation σ used in the *LDS* prior (see §4.2.2 of the main chapter) and **2)** the number of object slots K. The former relates to the precision of the similarity measure and the latter determines the size of the similarity matrix constructed in the *LDS* computation, i.e. it relates to the scalability of *LDS*. We do the ablation study with only MONet and on only the CLE-MV dataset for computation efficiency. We select 4 different σ to train MONet and compare their performance on the scene reconstruction and the scene factorization tasks. Figure 4.6 shows no significant performance loss in tasks by changing σ from the default value, 0.1, to other values. A future investigation will be further increasing σ until it is sufficiently close to a uniform distribution and thus breaks the *LDS* prior. Moreover, the performance might get boosted in some cases. For the object-slot quantity K, we first train MONet with $K = 7$ and $K = 9$ respectively and test them with 7,9, 11, 15 object slots. Figure 4.6 shows: **1)** the models trained with $K = 7$ and

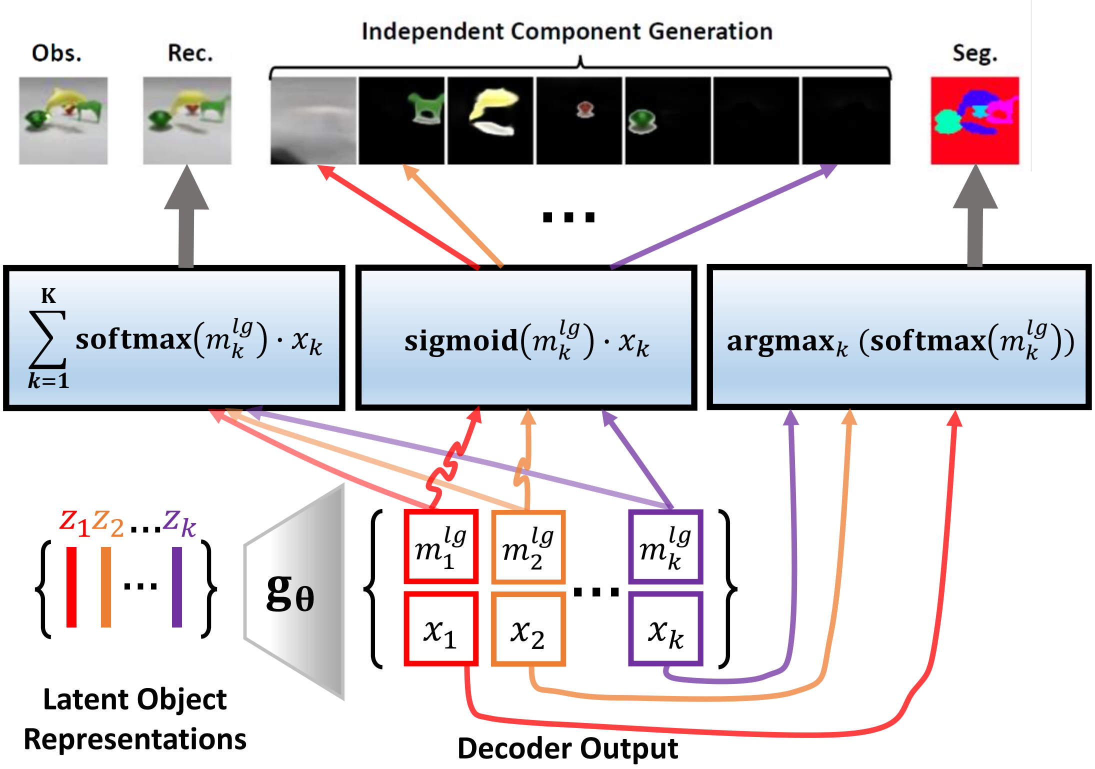

Obs.
Rec.
Independent Component Generation
Seg.
$\sum_{k=1}^{K}$ softmax(m_k^{lg}) · x_k
sigmoid(m_k^{lg}) · x_k
argmax$_k$ (softmax(m_k^{lg}))
$z_1 z_2 \ldots z_k$
g_θ
m_1^{lg}
m_2^{lg}
m_k^{lg}
x_1
x_2
x_k
Latent Object Representations
Decoder Output

Figure 4.5: Overview of a CompVAE rendering process. **Bottom left:** The rendering process starts by inputting a set of inferred latent object representations into the generator network g_θ. **Bottom middle:** The generator g_θ outputs a raw mask ($m_k^{lg} \in \mathbb{R}^{H\times W\times 1}$) and a color pool ($x_k \in \mathbb{R}^{H\times W\times 3}$). **Top & middle row:** The decoder output is then passed into three different functions to get different render results. All computations are defined pixel-wise but executed in parallel.

$K = 9$ have very similar performance in both tasks and **2)** testing with a different K does not cause a significant performance drop.

4C.2 GENESIS on the CLE-MV Data

We tested GENESIS (Engelcke et al., 2019) on the CLE-MV data to assess how well the inference redundancy problems are handled by the autoregressive model of GENESIS. The experiment was conducted on top of the official implementation

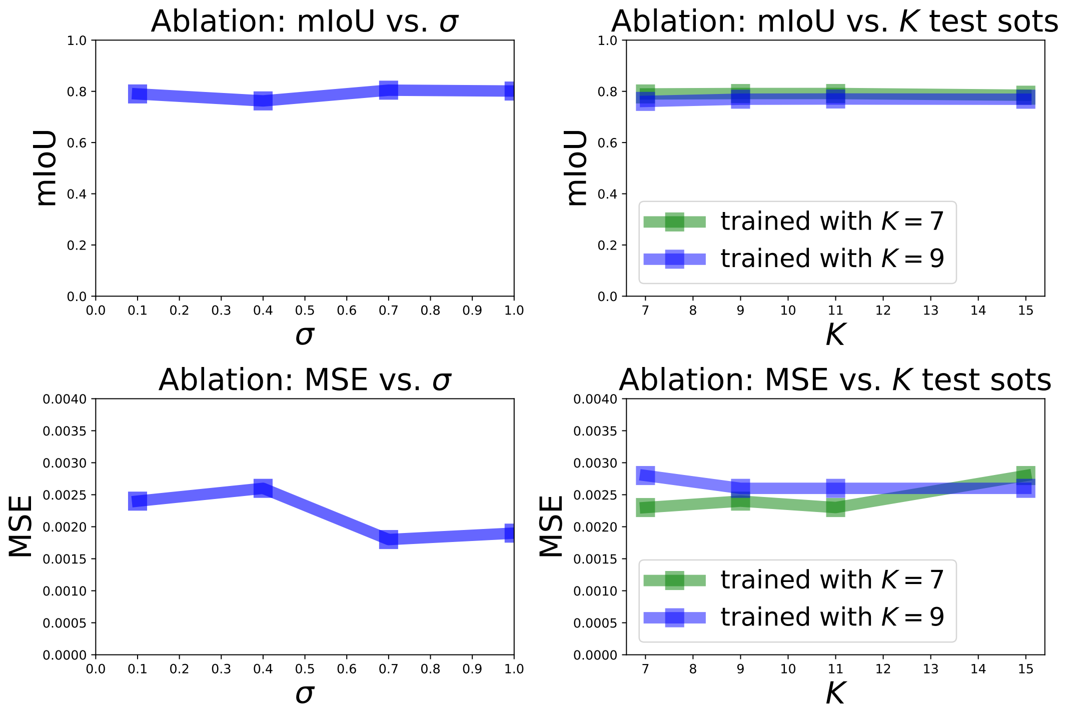


Figure 4.6: Ablation study results. **Top left:** Scene decomposition performance vs. *LDS* prior precision (σ). **Top right:** Scene decomposition performance vs. the number of object slots used in training and testing (K). **Bottom left:** Scene observation reconstruction performance vs. *LDS* prior precision (σ). **Bottom right:** Scene observation reconstruction performance vs. the number of object slots used in training and testing (K).

of GENESIS[1] with strict abidance of its original hyperparameter configurations. However, as shown in Figure 4.7, GENESIS failed to factorise CLE-MV scenes correctly—it treats a CLE-MV scene observation (i.e. an image) as a big and flat object that contains all the content. As a result, it produces wrong image segmentation. A possible reason could be that GENESIS represents the autoregressive conditioning of object discovery in the latent space (i.e. $z_k \,|\, z_{1:k-1}$) instead of the image space as that of MONet—a successive object mask conditions directly on all the previous obtained masks (i.e. $m_k \,|\, m_{1:k-1}$). According to Emami et al. (2021), this could introduce a more severe global information leaking issue. In

[1] https://github.com/applied-ai-lab/genesis.git

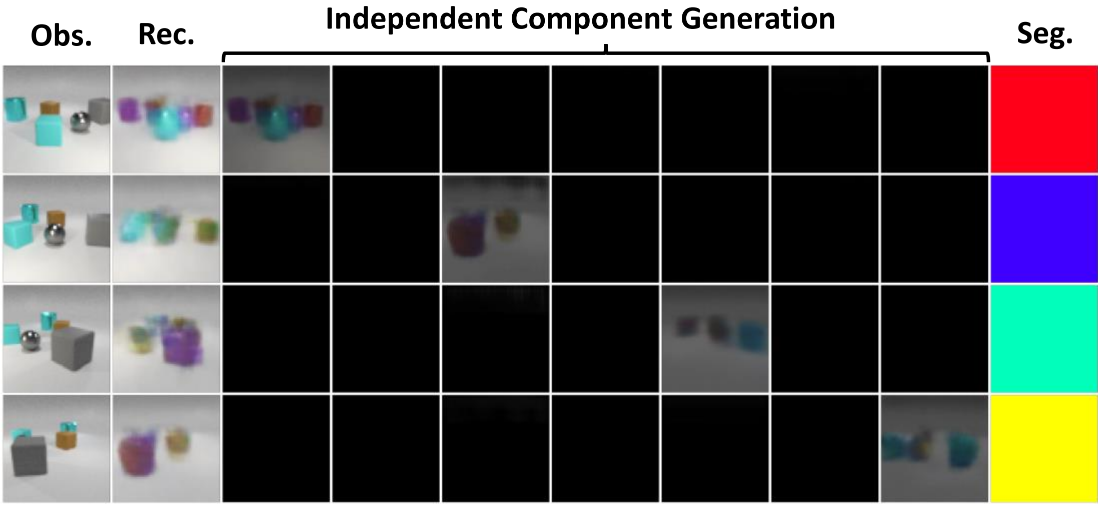


Figure 4.7: Qualitative results of GENESIS on the CLE-MV dataset.

general, future study is needed to better understand the practical limitations and their causes in GENESIS.

4C.3 Real-image Experiments

To demonstrate that the proposed LDS can efficiently perform duplicate suppression on real images, we conducted comparison experiments between Comp-VAEs that are trained with and without LDS priors on the a collected real-image dataset. For simplicity, we chose only MONet for this comparison because MONet suffers the latent-duplicate issue the most among the three investigated Comp-VAE variants (see Figure 4.3, right).

Real-image Dataset We created such dataset by randomly placing 2−4 cubes (of different colours) on white table top and taking photos with a webcam that is mounted on a moving robot arm. We created 109 scenes in total and for each scene we captured 20−30 images from different viewing angles. We show the hardware platform setup in Figure 4.8.

Results Figure 4.9 shows that the original MONet$^{0^\star}$ infers redundant white table components. Although MONet^{+} demonstrates a slight performance drop in

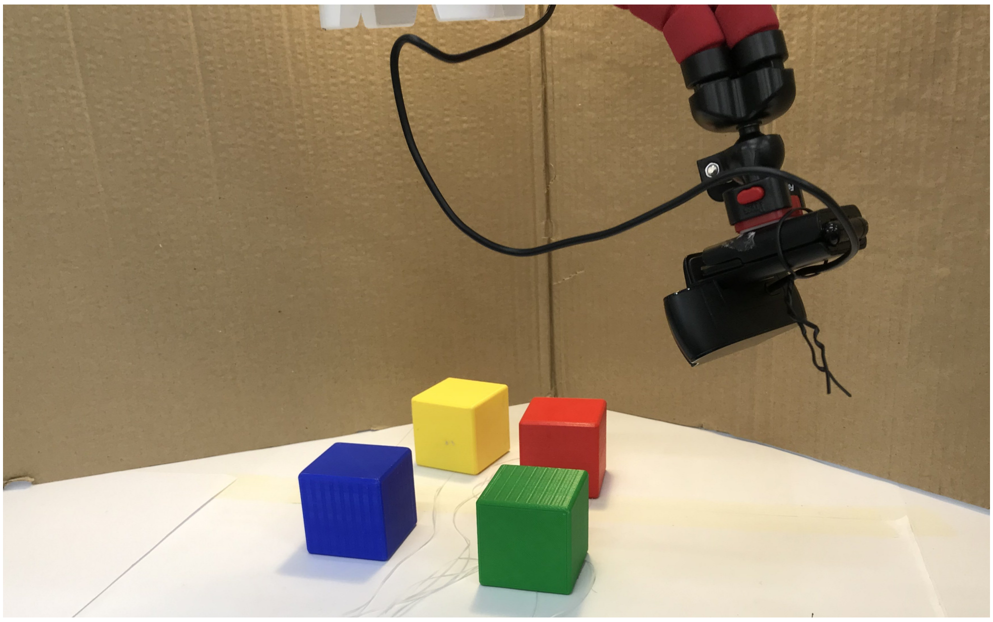


Figure 4.8: Hardware platform for real-image dataset recording.

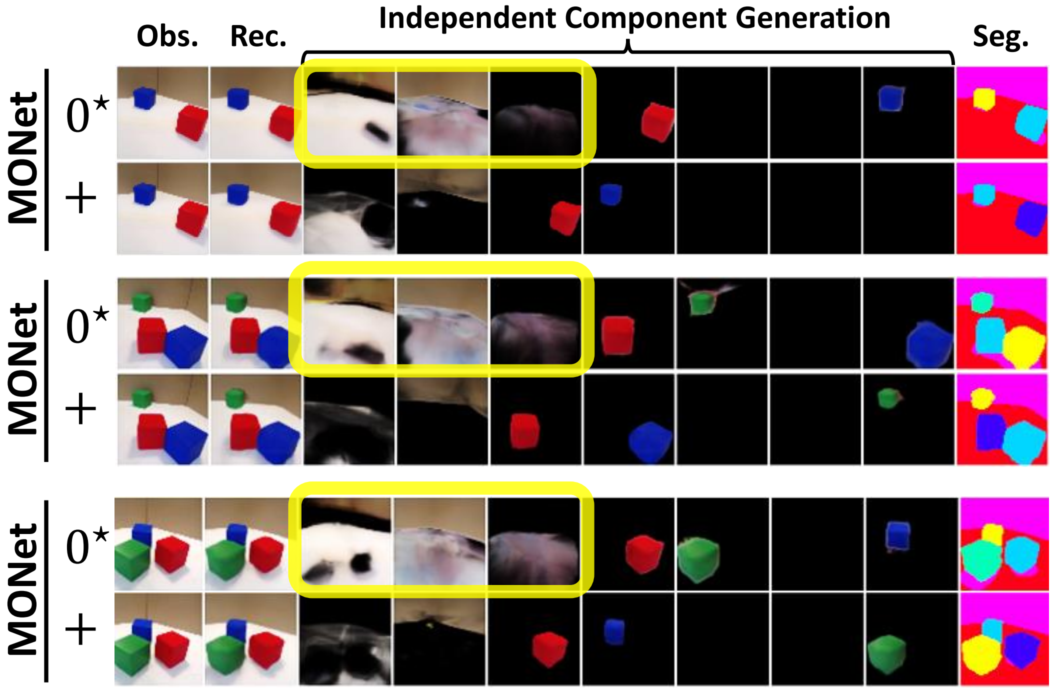


Figure 4.9: Qualitative results of MONet on real images. Symbols "$0^\star$" and "+" tag models that trained with and without LDS respectively. Yellow circles highlight duplicated or partially duplicated components.

handling occlusions (e.g. renders the independent table component worse than MONet$^{0^\star}$), it does suppress the duplicate table finding issues of MONet$^{0^\star}$. Also, we see that MONet^{+} produces cleaner segmentation results than MONet$^{0^{\star}}$. Compared with synthetic data, real images often come from complex distributions and thus exhibit significant larger pixel variances (due to uncontrolled lighting, materials, etc.), complicating the training of a generative model. This also explains why neither MONet$^{0^{\star}}$ nor MONet^{+} model the independent table (always partially occluded) distribution properly. In conclusion, LDS is an effective addition to CompVAEs on real data and can potentially serve as a useful tool in some real applications.

Chapter 5

DyMON: Dynamics-aware Multi-Object Network

Learning object-centric scene representations is essential for attaining structural understanding and abstraction in 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.

This chapter is an extended version of the paper "*Object-Centric Representation Learning with Generative Spatial-Temporal Factorization*" (Nanbo et al., 2021), published at *Neural Information Processing Systems* (2021).

5.1 Introduction

Object-centric representation learning (OCRL) promises improved interpretability, generalization, and data-efficient learning on various downstream tasks like reasoning (Janner et al., 2019; Van Steenkiste et al., 2019) and planning (Mnih et al., 2015; Carlos et al., 2008; Zadaianchuk et al., 2021). It aims at discovering compositional structures around objects from the raw sensory input data, i.e. a *binding problem* (Greff et al., 2020), where the *segregation* (Revonsuo and Newman, 1999; Greff et al., 2020) (i.e. factorization)[1] 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 (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020b) or a fixed-view video (Lin et al., 2020). 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. (2020) 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 limits a model's potential in real-world applications.

In this chapter, we target an unexplored problem—unsupervised object-centric la-

[1] We make no distinction between the terms "segregation" and "factorization" in this thesis. Also, we consider "image segmentation" an application of "object factorization".

tent representation learning in *multi-view-dynamic-scene* scenarios. In this chapter, we consider "multi-view" observations as the product of a moving ego-centric observer. I.e. there is one and only one observer in the scene, even though multiple moving observers can also generate multi-view observations (see Singh et al. 2019). 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 a scene are moving at the same time. To aggregate 3D object information from consecutive observations, an agent needs not only to handle the cross-view object correspondence problem (Nanbo et al., 2020) but also to factorize the independent effects of the scene dynamics and observer motions in the observations. 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 chapter, we refer to the relationship between the scene spatial structures and the viewpoints as the *temporal entanglement* because the temporal association of them complicates the recovery of the *independent causal mechanism* (Schölkopf et al., 2021), or an equivalence, around the scenes and the observers.

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 §5.3). 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 (this still leaves the intermediate case for future work). 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 §5.5). **(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 §5.5.1). **(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 §5.5.1).

5.2 Problem: Temporal Entanglement

The dynamic nature of the world suggests that the spatial configuration of a scene $\mathbf{Z}^t$, i.e. a set of objects $\{Z_k^t\}_{1:K}$ (where $Z_k^t \in \mathbb{R}^D$), and an observer $V^t \in \mathbb{R}^J$ are bound to the specific time t that an observation $X^t \in \mathbb{R}^M$ $(M \gg D)$ is taken. Let x^t and v^t denote the samples (specific values) of X^t and V^t, respectively, we define $\mathbf{x} = \{(x^t, v^t)\}_{1:T}$ [2] as a specific data sample, e.g. a sequence or set of T multi-view images with their associating viewpoints, from the observational data $\mathbf{D}$. Assuming $\mathbf{Z}^t$ is also observable (provided in the data) for now, we augment a scene data sample as $\mathbf{x}_a = \{(x^t, v^t, \mathbf{z}^t)\}_{1:T}$ and focus on describing

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

the relationships among $\mathbf{Z}$, V, and X. In general, we assume that $\mathbf{Z}$ and V are independent such that $P(\mathbf{Z}, V) = P(\mathbf{Z})P(V)$ (as discussed in §1.1), but they nevertheless become dependent when an observation X is given. In this case, inferring the actual scene object motions $\mathbf{Z}^t$ or the observer's motions V^t requires an agent to correctly disentangle the generative effects of $\mathbf{Z}$ and V upon X, i.e. describing $P(X|\operatorname{do}(\mathbf{Z}))$ and $P(X|\operatorname{do}(V))$[3].

In this chapter, we aim to train a generative model, $P(X|\mathbf{Z}, V)$, that correctly captures $P(X|\operatorname{do}(\mathbf{Z}))$ and $P(X|\operatorname{do}(V))$. Under our assumption about the causal graph: $\mathbf{Z} \rightarrow X \leftarrow V$ (with independent $\mathbf{Z}$ and V, see 1.1), estimating $P(X|\operatorname{do}(\mathbf{Z}))$ and $P(X|\operatorname{do}(V))$ is equivalent to estimating: $P(X|\mathbf{Z})$ and $P(X|V)$, respectively. Therefore, if we can sample $\mathbf{Z}$ and V with its associated X, e.g. sampling $(X, \mathbf{Z})$ independently of V or sampling (X, V) independently of $\mathbf{Z}$, we can estimate $P(X|\mathbf{Z})$ by marginalizing V, i.e. $P(X|\mathbf{Z}) = \sum_V P(X|\mathbf{Z}, V)P(V)$ and, similarly, estimate $P(X|V) = \sum_{\mathbf{Z}} P(X|\mathbf{Z}, V)P(\mathbf{Z})$. In GQN (Eslami et al., 2018) and MulMON (Nanbo et al., 2020), where scenes are assumed static, we can treat an augmented observational scene sample as $\mathbf{x}_a = \{(x^t, v^t), \mathbf{z}^t\}_{1:T}$—i.e. $\mathbf{z}^t$ and $(x^{t'}, v^{t'}) \sim \{(x^t, v^t)\}_{1:T}$ can be accessed independently hence marginalizing $\mathbf{Z}^t$ is possible. With recurring values v^t across different scenes samples[4] $\mathbf{x}_a \sim \mathbf{D}$, marginalizing V^t is also possible. However, in more general settings where scenes can be dynamic, the joint-sample indicator $(\cdot)$ in $\mathbf{x}_a = \{(x^t, v^t, \mathbf{z}^t)\}_{1:T}$ forbids drawing $(x^{t'}, v^{t'}) \sim \{(x^t, v^t)\}_{1:T}$ independently of $\mathbf{z}^t$—i.e. $\mathbf{Z}^t$ and V^t appear to be entangled along the temporal axis. In this case, without further assumptions, learning the disentangled causal effects of $\mathbf{Z}$ and V from the observational data is seemingly impossible. In this chapter, we refer to this issue as *temporal entanglement* in view of the temporal implication of the $(\cdot)$ indicator.

[3] The $\operatorname{do}(\cdot)$ operators perform interventions, see (Pearl, 2012)

[4] Some V values recur because GQN and MulMON used finite sample spaces for V, whose size is much smaller than that of the scenes $\mathbf{Z}$ (i.e. the size of $\mathbf{D}$).

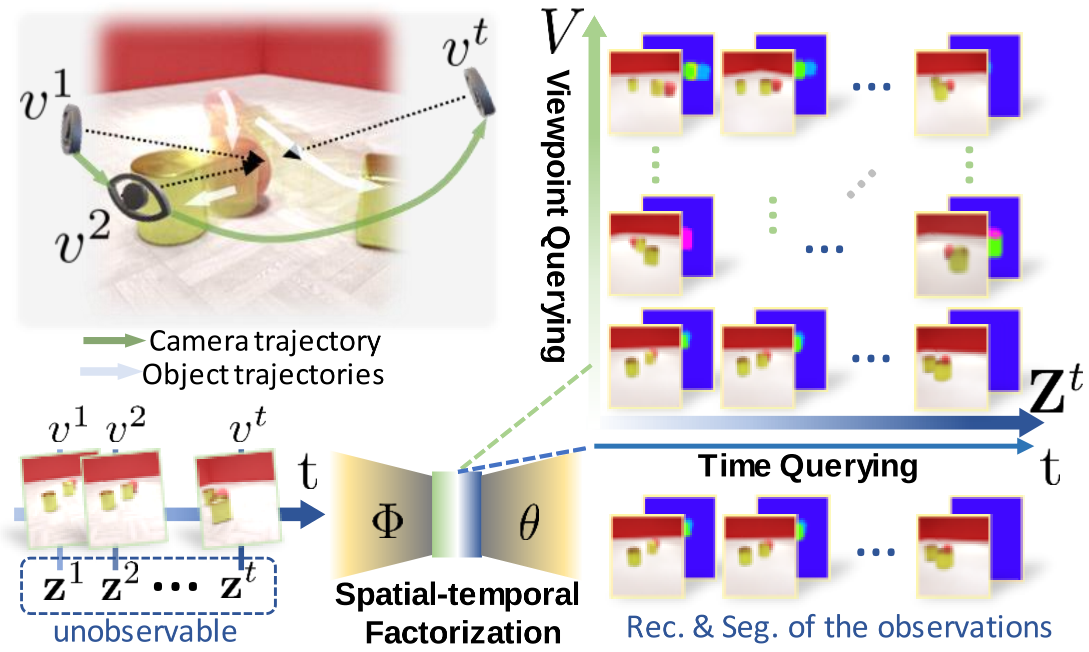


Figure 5.1: **Top Left:** *Multi-view-dynamic-scene* setup. A sample v^t denotes the spatial configuration (e.g. position, orientation, etc.) of an observer at a specific time t. A latent sample $\mathbf{z}^t$ describes the objects and their spatial configuration at a specific time t. We highlight one 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 in the scene observation, an image sequence (see *bottom left*). **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*).

5.3 Method: 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 §5.3.1), and **2)** scene spatial factorization (see §5.3.2). We discuss the training of DyMON in §5.3.3.

5.3.1 Temporal Disentanglement

The key to resolving *temporal entanglement*, i.e. temporal disentanglement, is to enable sampling (X^t, V^t) independently of $\mathbf{Z}^t$, and $(X^t, \mathbf{Z}^t)$ 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, along the temporal axis. In this chapter, 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 the scenes and observers with two independent dynamical systems:

$$\mathbf{Z}^{t+\Delta t} - \mathbf{Z}^t = \overline{f_{\mathbf{Z}}}(\mathbf{Z}^t, t)\Delta t \ , \quad V^{t+\Delta t} - V^t = \overline{f_V}(V^t, t)\Delta t, \tag{5.1}$$

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, though each component $Z^t \in \mathbf{Z}^t$ should, in theory, capture both the shape and pose dynamics of an object, we do not consider deformable objects, whose shapes can change temporally, in this chapter. 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 §5.3.3). In testing, DyMON treats them equally.

5.3.2 Spatial Object Factorization

DyMON tackles scene spatial decomposition in a similar way to MulMON (Nanbo et al., 2020) using a generative model and an inference model. The generative likelihood function of a single image observation is modelled with a spatial Gaussian mixture (Williams and Titsias, 2004; Greff et al., 2017):

$$p_\theta(x^t | \mathbf{z}^t = \{z_k^t\}, v^t) = \prod_{i=1}^{M} \sum_{k=1}^{K} p_\theta(C_i^t = k | z_k^t) \cdot \mathcal{N}(x_{k,i}^t; g_\theta(z_k^t, v^t), \sigma^2 \mathbf{I}), \tag{5.2}$$

where i indexes a pixel location (M in total) and RGB values (e.g. $x^t_{k,i}$) that pertain to an object k are sampled from a Gaussian distribution $\mathcal{N}(x^t_{k,i}; g_\theta(z^t_k, v^t), \sigma^2\mathbf{I})$ whose mean is determined by the decoder network $g_\theta(\cdot)$ (defined in §1.1) 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* MulMON (see §3.3.1) 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:

$$p(\mathbf{z}^t = \{z^t_k\}|x^{\leqslant t}, v^{\leqslant t}) \approx q_\Phi(\mathbf{z}^t\,|x^{\leqslant t}, v^{\leqslant t}) = q(\mathbf{z}^0)\prod_t q_\Phi(\mathbf{z}^t\,|x^t, v^t, \mathbf{z}^{<t}), \tag{5.3}$$

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* (Marino et al., 2018) with amortization function Φ (modelled with neural networks). Refer to Appendix 5B. for full details about the generative and inference models of DyMON.

5.3.3 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 Algorithm 4) where accessing $(x^{t'}, v^{t'}) \sim \{(x^t, v^t)\}_{1:T}$ samples independently of $\mathbf{z}^t$ samples is possible. Similar to MulMON (Nanbo et al., 2020), 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:

$$
\begin{aligned}
\mathcal{L} =& \mathbf{ELBO} + \beta \cdot \mathbf{LL}_{query} \qquad (5.4)\\
=& \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} (\mathbf{E}_{q_\Phi(\mathbf{z}^t|\cdot)}[\log p_\theta(x^t|\mathbf{z}^t, v^t)] - \mathcal{D}_{\mathrm{KL}}[q_\Phi(\mathbf{z}^t|x^{\leqslant t}, v^{\leqslant t})||q_\Phi(\mathbf{z}^{<t}|x^{<t}, v^{<t})])\\
&+ \beta \cdot \frac{1}{|\mathcal{T}| \cdot |\mathcal{Q}|} \sum_{t \in \mathcal{T}} \sum_{t_q \in \mathcal{Q}} \mathbf{E}_{q_\Phi(\mathbf{z}^t|\cdot)}[\log p_\theta(x^q|\mathbf{z}^t, v^q)], \qquad (5.5)
\end{aligned}
$$

where $\mathcal{T}$ and $\mathcal{Q}$ record the times when DyMON performs inference and v^t viewpoint-queried generation, $|\cdot|$ returns the cardinality of a set, β is the weighting coefficient. We construct $\mathcal{T}$ by sampling t (either regularly or irregularly) with a random walk through $[1, T] \subseteq \mathbb{N}$, 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 Algorithm 4, 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.

Assignment Function and Batching As the samplers of T and Q behave differently for SCFO and FCSO data (see Algorithm 4), we need to determine if a data sample $\mathbf{x} \sim \mathbf{D}$ is an SCFO sample or an FCSO sample. Under **A2**, we

Algorithm 4: DyMON Training Algorithm

```
Input: Training data $\mathbf{D}$ (I.I.D. scenes)
Hyperparams $|\mathcal{Q}|, (\beta_{FCSO}, \beta_{SCFO}), (\Delta t, \Delta\tau)$ ;    // $\Delta t > \Delta\tau > 2$
Init trainable params: $\Phi, \theta$ ; prior: $\boldsymbol{\lambda}^0 = \{(\mu_k = \mathbf{0}, \sigma_k = \mathbf{I})\}$;
repeat
    Sample $\mathbf{x} = \{(x^t, v^t)\}_{1:T} \sim \mathbf{D}$ ;    // a seq of (RGB imgs, viewpts)
    if assign$(\mathbf{x}; \mathbf{D}) == FCSO$ 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
    $\mathcal{T} =$ rand_walk_t$(\mathrm{s} = 1, \mathrm{e} = T, \mathrm{step\_dist} = \mathcal{U}\{\Delta t_{\mathbf{z}} - 2, \Delta t_{\mathbf{z}} + 2\})$ ;
    $(x, v)$, $t$, $\boldsymbol{\lambda}^t$, $\boldsymbol{ELBO}$, $\mathbf{LL}_{query}$, $=$ $\mathbf{x}[1]$, $1$, $\boldsymbol{\lambda}^0$, $0$, $0$;
    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$_{\Phi,\theta}(x, v, \boldsymbol{\lambda}^t)$ ;
            $\mathbf{z}^\mathbf{t} \sim \mathcal{N}(\mathbf{z}^t; \boldsymbol{\lambda}^t)$ ;    // sample updated $\mathbf{z}^t$
            $\mathcal{Q} = \{t_q\} =$ randint$(\mathrm{dist} = \mathcal{U}\{t - \Delta t_{\mathbf{z}}/2, t + \Delta t_{\mathbf{z}}/2\}, \mathrm{size} = |\mathcal{Q}|)$;
            for $t_q \in \mathcal{Q}$ do
                $(x^q, v^q) = \mathbf{x}[t_q]$;
                $\boldsymbol{LL}_{query} += (1/(|\mathcal{Q}| \cdot |\mathcal{T}|)) \cdot \log p_\theta(x^q | \mathbf{z}^t, v^q)$ ;    // query $v = v^q$
            $\boldsymbol{ELBO} += (1/|\mathcal{T}|) \cdot \boldsymbol{ELBO}^{(t)}$;
        $t += 1$;
    $\mathcal{L} = \boldsymbol{ELBO} + \beta \cdot \boldsymbol{LL}_{query}$ ;
    $\theta, \Phi \xleftarrow{update}$ optimizer$_{\mathbf{max}}(\mathcal{L}, \theta, \Phi)$;
until $\theta, \Phi$ converge;
```

consider any dataset consisting of only a mix (i.e. no fast moving camera and objects, nor no stationary camera and objects) 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, $\mathbf{assign}(\mathbf{x}; \mathbf{D})$ (as shown in Algorithm 4). 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.

5.4 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 (Kingma and Welling, 2013), early success was shown by AIR (Eslami et al., 2016) that searches for one object at a time in image regions. AIR and most of its successors (Kosiorek et al., 2018) generally treat objects as flat pixel patches and the image generation process as “paste flat objects on canvas” using a spatial transformer (Jaderberg et al., 2015). Without further assumptions about the generator (e.g. using a pre-defined graphics renderer like (Yao et al., 2018)), they often cannot summarize well scene spatial properties that are suitable for 3D spatial reasoning and manipulation. To overcome this, most recent advances (Burgess et al., 2019; Greff et al., 2019; Lin et al., 2019; Engelcke et al., 2019; Locatello et al., 2020b; Engelcke et al., 2021) model a single 2D image with a spatial Gaussian mixture model (Williams and Titsias, 2004; Greff et al., 2017) that allows explicit modeling of background and occlusions. Our work has close relationship to IODINE (Greff et al., 2019): we handle the object-wise inference from an image observation at each time point using the *iterative amortized inference* (Marino et al., 2018) design and capture the compositional generative process with a spatial Gaussian mixture model. However, IODINE and the aforementioned methods still suffer from single-view ambiguities like occlusions or optical illusions—i.e. they cannot form accurate 3D scene representations.

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. (2018) demonstrated that they do reduce the spatial uncertainty and enable explicit 3D knowledge evaluation—novel-view prediction. By combining GQN (Eslami et al., 2018) and IODINE (Greff et al., 2019), Nanbo et al. (2020) 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 (Chen et al., 2021) 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 (Hsieh et al., 2018; Kosiorek et al., 2018; Jaques et al., 2020), 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. (2020), i.e. GSWM, where they modeled relative depth information and pair-wise interactions of 3D object patches. However, as GSWM does *not* support viewpoint reasoning like GQN (Eslami et al., 2018) and MulMON (Nanbo et al., 2020), it is unknown if these models really learn about 3D. 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 (Singh et al., 2019) 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, notable distinctions between the problems that T-GQN and DyMON are targeting: **1)** T-GQN does *not* infer object-level scene factorization and **2)** a typical T-GQN situation 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 (Pearl et al., 2009; Peters et al., 2017; Suter et al., 2019; Schölkopf et al., 2021) and *independent component analysis* (abbr. ICA) (Comon, 1992; Hyvärinen and Pajunen, 1999; Hyvarinen and Morioka, 2016). Unlike traditional disentanglement representation learning works (Higgins et al., 2017; Kim and Mnih, 2018; Locatello et al., 2019a) that aim at feature-level disentanglement, in this chapter, 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 trends of neural radiance fields (Mildenhall et al., 2020; Martin-Brualla et al., 2021; Pumarola et al., 2020) are relevant to our work in the sense of 3D scene representations using multi-view images. However, from a *vision-as-Bayesian-inference* (Yuille and Kersten, 2006) 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.

5.5 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 5C.3 for details), in this chapter. 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-$|\overline{f_{\mathbf{z}}}|$, see Appendix 5C.1) 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. 57600 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 (Lin et al., 2020), and MulMON (Nanbo et al., 2020), 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 the Appendix for full details on experimental setups, and ablation studies and more qualitative results.

5.5.1 Space-Time Querying

DyMON takes in a sequence of RGB image observations of a scene along with their associating viewpoints, i.e. $\mathbf{x} = \{(x^t, v^t)\}_{1:T}$, and infers a set of latent scene representations $\{\mathbf{z}^t\}_{1:T}$ that associate with the observation time stamps t. Re-

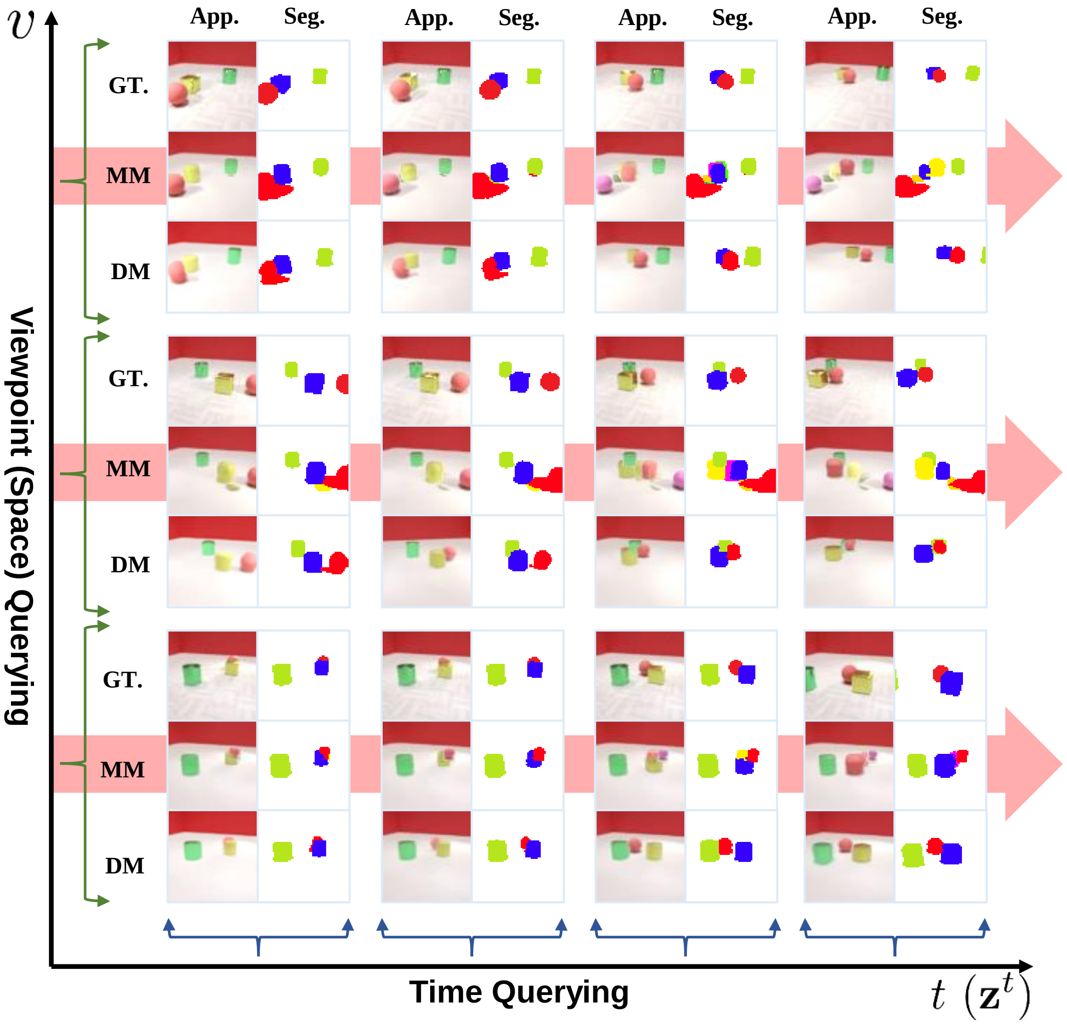


Figure 5.2: Qualitative results of spatial-temporal factorization. The GT rows show the true scene. The "MM" and "DM" entries are the scene re-rendered from the corresponding models, i.e. MulMON and DyMON respectively. The vertical row pairs show the results from viewpoint changes and the horizontal 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).

call that DyMON can disentangle the mixed effects of the scene motions and observer motions only if the learned mechanism $P(X|\mathbf{Z}, V)$ correctly captures

$P(X|\operatorname{do}(\mathbf{Z}))$ and $P(X|\operatorname{do}(V))$ (see §5.2). To evaluate if DyMON successfully recovered the underlying *independent generative mechanism* (or an equivalence) from the training data, we have DyMON perform both *viewpoint-queried* and *time-queried* predictions of *scene appearances* and *segmentations* using the inferred scene representations. In other words, we evaluate how well DyMON can answer counterfactual questions about the spatial and temporal scene structures. We show the results with the below two demonstrations:

Novel-view Prediction at Arbitrary Times In this experiment, we took the inferred latent scene representations $\{\mathbf{z}^t\}_{1:T}$ and the learned generative mechanism $P(X|\mathbf{Z},V)$, and checked the scene spatial structures at arbitrary times from arbitrary viewpoints. Specifically, we fixed $\mathbf{Z}$ to a value of interest $\mathbf{z} \in \{\mathbf{z}^t\}_{1:T}$ and manually set the viewpoint V to arbitrary values $v \in \mathbf{supp}(V)$ in the generative model—i.e. formally, $P(X|\operatorname{do}(\mathbf{Z}=\mathbf{z}),\operatorname{do}(V=v))$ from a causal perspective. Similarly, we also queried about the spatial state of a dynamic scene at time t from a specific viewpoint by fixing the viewpoint and manually inputting $\mathbf{z}^t$ at arbitrary times t to the generative function. We conducted this experiment was trained on the DR-Lvl.3 data and show the prediction results that are queried by space-time tuples, $(V,\ \mathbf{Z})$, in Figure 5.2.

Dynamics Replay of Scenes & Objects From Arbitrary Viewpoints In this experiment, we gave DyMON a sequence of image observations of a dynamic scene as input, and had 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 5.3 (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 5.3 (right) where one can

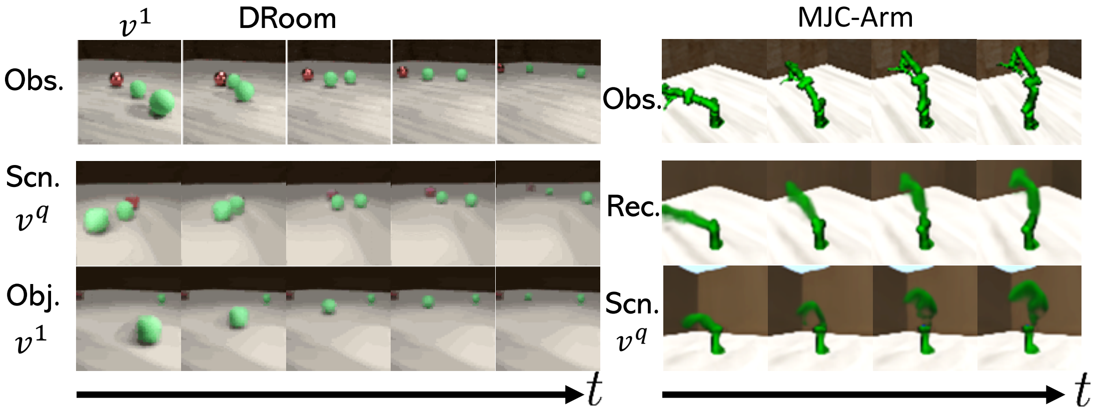


Figure 5.3: **Left:** DyMON performing dynamics replays on the DRoom dataset, where the first row is the observation sequence input to DyMON, second and third rows show replays of the scene dynamics (all objects' original motions) and object dynamics (just the foreground green ball moves) respectively 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).

see that DyMON not only replays object dynamics as global position changes, it also captures object local motions.

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 5D.4 for the results.

5.5.2 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.

DyMON vs. Dynamic Scenes In this experiment, we evaluate DyMON's performance in the *multi-view-dynamic-scene* setting in comparison to MulMON. We discussed in chapter 3 that MulMON can also recover the *independent generative*

	MSE↓		mIoU↑	
Models	Obs.Rec.	Nv.Obs.	Obs.Seg.	Nv.Seg.
MulMON	0.011 ± 0.001	**0.019 ± 0.002**	0.511 ± 0.001	0.461 ± 0.062
DyMON	**0.004 ± 0.001**	0.021 ± 0.002	**0.717 ± 0.000**	**0.508 ± 0.065**

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

	MSE↓		mIoU↑	
Models	Obs.Rec.	Nv.Obs.	Obs.Seg.	Nv.Seg.
MulMON	**0.006 ± 0.001**	**0.012 ± 0.005**	**0.583 ± 0.080**	**0.538 ± 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	**0.014 ± 0.011**	**0.682 ± 0.107**

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

Table 5.1: 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.

mechanism around scenes $\mathbf{Z}$ and an observer V, but it strictly requires **static-scene** training data. 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 trained DyMON on the DR-Lvl.3 subset and MulMON on the DR0-$|\overline{f_{\mathbf{z}}}|$ subset (because MulMON *cannot* train on dynamic-scene data) and conducted comparison across the three DRoom dynamic-scene subsets (i.e. DR-Lvl.1 $\sim$ 3). Table 5.1a 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 5.2 and observe that MulMON's performance declines along the

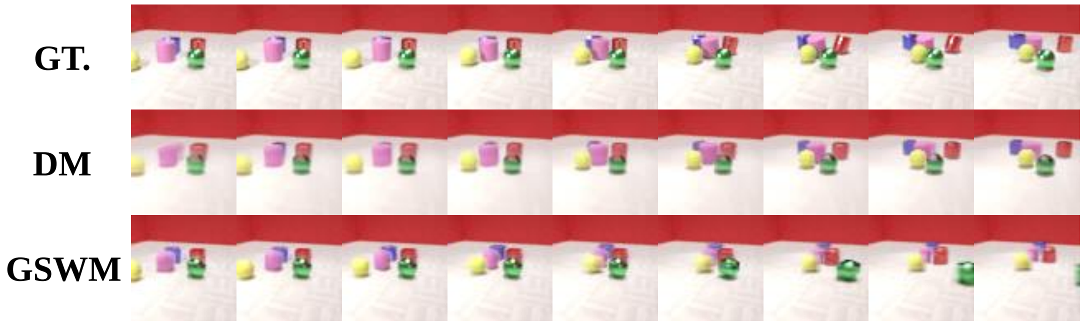


Figure 5.4: **Left:** Qualitative comparisons of DyMON and GSWM on reconstructing the DR0-$|\overline{f_v}|$ scenes. The GT rows show the actual observations of a dynamic scene, and the "DM" and "GSWM" rows show observation reconstruction results of DyMON and GSWM, respectively.

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 In this experiment, we evaluate how well DyMON handles *multi-view-static-scene* scenarios in comparison with a specialized model, i.e. MulMON. We train and test both DyMON nad MulMON on the DR0-$|\overline{f_{\mathbf{z}}}|$ subset w.r.t. reconstructions and segmentations of both the observed and unobserved views. Table 5.1b 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. 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 (Lin et al., 2020), 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-$|\overline{f_v}|$ subset and measure the reconstruction quality of the observations. Table 5.1c 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 5.4 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 5.3 & 5.4), 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 (Singh et al., 2019) 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 5D.3 for the results and discussions).

5.5.3 Robustness vs. Assumption Violations

As discussed in §5.3.1, we enable the training of DyMON on *multi-view-dynamic-scene* by proposing two assumptions that favor: **i)** high frame-rate image sequences and **ii)** significant difference between the speeds of an observer and scene objects, respectively. In this experiment, we assess the robustness of DyMON against violations of our assumptions. As the DR-Lvl.1 $\sim$ 3 datasets were created with three different settings of average scene-observer speed differences (the differences increase from DR-Lvl.1 to DR-Lvl.3, see Appendix 5C.1 for more details), we used these three datasets to simulate three different levels of violations (violation levels decrease from DR-Lvl.1 to DR-Lvl.3). 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 decreased levels of assumption violations in Figure 5.5. As shown, **1)** the space-time querying performance boost (by ~ 0.008, see Figure 5.5, left) as the level of violation decreases within training (from DR-Lvl.1 to DR-Lvl.3); **2)** the space-time querying performance boost for all three DyMONs as we increase the magnitude of scene-observer speed

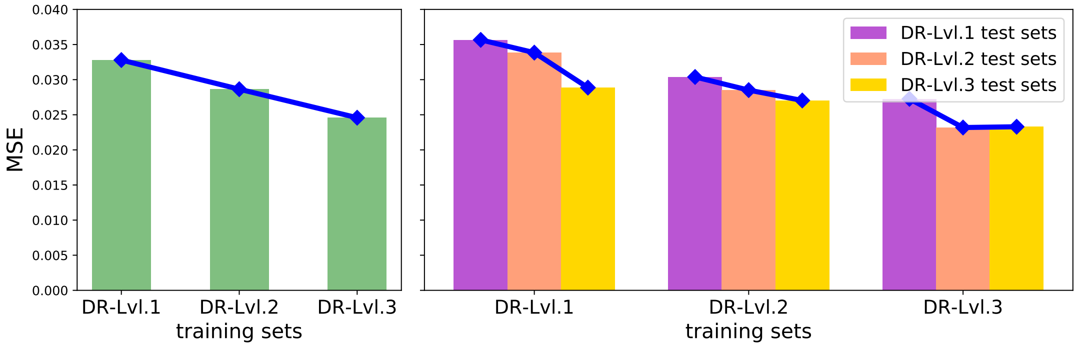


Figure 5.5: Space-time-queried scene appearance prediction MSE (*lower is better*): DyMON vs. levels of assumption violations (three violation levels correspond to three DRoom subsets: DR-Lvl.1 $\sim$ 3). The violation levels decrease from DR-Lvl.1 to DR-Lvl.3. **Left:** Average prediction MSEs (across all the DR-Lvl.1 $\sim$ 3 testing data) achieved by three DyMONs that are trained on the DR-Lvl.1 $\sim$ 3 training data, respectively. MSE reduces when *training* on datasets with larger scene-observer speed differences. **Right:** The performance (MSEs) of the three DyMONs on each of the three DR-Lvl.1 $\sim$ 3 testing data. MSE reduces when *testing* on datasets with larger scene-observer speed differences.

differences. These results suggest that **1)** DyMON can handle complex *multi-view-dynamic-scene* environments to certain degrees and **2)** more distinguishable scene-observer dynamics should lead to better performance.

5.6 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 chapter focuses on representing the spatial scene configurations at every specific time point, i.e. DyMON does not model dynamics explicitly, it cannot predict the future evolution of scenes, which leaves space for future exploration.

Acknowledgements

The authors would like to thank Prof. C. K. I. Williams and Cian Eastwood for their valuable discussions, the reviewers of the original paper for the insightful suggestions. The first author would like to acknowledge the School of Informatics, the University of Edinburgh for providing his PhD scholarship. This research was also partly supported by the *Trimbot2020* project, which contributes to the European Union Horizon 2020 programme.

Note: we use the same notations in the Appendix as that in the main chapter.

Appendix 5A. Algorithms

5A.1 Iterative inference algorithm

Algorithm 5: 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
    $\mathbf{z^{t(l)}} \sim \mathcal{N}(\mathbf{z}^{t(l)}; \boldsymbol{\lambda}^{t(l)})$ ;          // sample from a prior---make a guess
    $p_\theta(x^{t(l)}|\mathbf{z}^{t(l)}, v) = g_\theta(\mathbf{z^{t(l)}}, v)$ ;          // 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 $\boldsymbol{ELBO}^t, \boldsymbol{\lambda}^{t(l)} = \{(\mu_k^{t(l)}, \sigma_k^{t(l)})\}$
```

5A.2 Testing algorithm

Algorithm 6: 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 $\boldsymbol{\lambda}^t = \{(\mu_k^t, \sigma_k^t)\}$ ;
```

Appendix 5B. Implementation Details

5B.1 Training configurations

We show the training configurations used in this chapter in Table 5.2.

Table 5.2: 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	DyMON vs. GSWM: 300000 (for both) DyMON vs. MulMON: 200000 (for both)
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	

5B.2 Model implementation

We show the designs of the generative mapping function g_θ and the refinement function in Table 5.3 &. 5.4 respectively. After obtaining a set of K RGBM outputs from this function, i.e. $\{(\mu_{xk}, \hat{m}_{xk})\}$ (see Table 5.3), we render (i.e. compose) an image as: $x = \sum_k \mathbf{softmax}(\hat{m}_{xk}) \cdot x_k$, where $x_k \sim \mathcal{N}(x_k; \mu_{xk}, 0.1^2\mathbf{I})$,

Appendix 5C. Datasets

5C.1 DRoom (DynamicRoom)

Simulation Environment We created the DRoom simulation on the top of the CLEVR Blender environment (Johnson et al., 2017)[5]. Like other multi-

[5] https://github.com/facebookresearch/clevr-dataset-gen (Accessed: 2021-06-02)

Table 5.3: Generator function g_θ

Parameters	Type	Channels (out)	Activations.	Descriptions
θ^1 (projection)	Input	$D + d$		$\mathbf{z}^t \sim \mathcal{N}(\mathbf{z}^t; \boldsymbol{\lambda}^t)$, v^t
	Linear	256	Relu	
	Linear	D	Linear	$\tilde{\mathbf{z}}^t = g_{\theta_1}(\mathbf{z}^t, v^t)$
θ^2 (rendering)	Input	D		$\tilde{z}_k^t = g_{\theta_2}(z_k^t, v^t)$
	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 Watters et al. (2019)

Stride= 1 set for all Convs.

Table 5.4: 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 (Greff et al., 2019) for details

LSTMCell/GRUCell channels: the dimensions of the hidden states

object datasets[6], 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 cen-

[6] https://github.com/deepmind/multi_object_datasets (Accessed: 2021-06-02)

trifugal force field within a fixed center and range on the ground across all DRoom datasets. In this chapter, we sample the magnitude of the force using: $\textbf{random.choice}(\text{vals} = 8500 \times \{0, 0.1, 0.2, ..., 1\}, \text{probs} = \mathcal{C}at(...))$, 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: $\textbf{random.choice}_{azi}(\text{vals} = 5.0\,degs \times \{0, 0.1, 0.2, ..., 1\}, \text{probs} = \mathcal{C}at_{azi}(...))$ and $\textbf{random.choice}_{ele}(\text{vals} = 1.0\,degs \times \{0, 0.1, 0.2, ..., 1\}, \text{probs} = \mathcal{C}at_{ele}(...))$, which suggests that we can control the speed of the camera by inputting different $\mathcal{C}at_{azi}(...))$ and $\mathcal{C}at_{ele}(...))$.

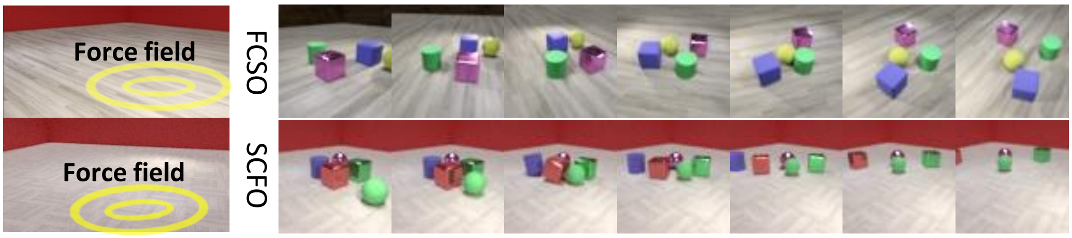


Figure 5.6: **Left:** DRoom simulation environment setup where yellow rings denote the force fields. **Right:** One *fast-camera-slow-object* (FCSO) sample (**top row**) and *slow-camera-fast-object* (SCFO) sample (**bottom row**). Both are randomly selected from the DR-Lvl.3 dataset.

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 Cartesian coordinates. The sampler specifications, i.e. the categorical distributions $\mathcal{C}at(...)$), used to generate the five DRoom subsets are listed in Table 5.5. 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 5.7.

Table 5.5: 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	$\{0, 0, 0, 0, 0, 0.02, 0.08, 0.15, 0.35, 0.35, 0.05\}$	$\{0.3, 0.3, 0.28, 0.11, 0.01, 0, 0, 0, 0, 0, 0\}$

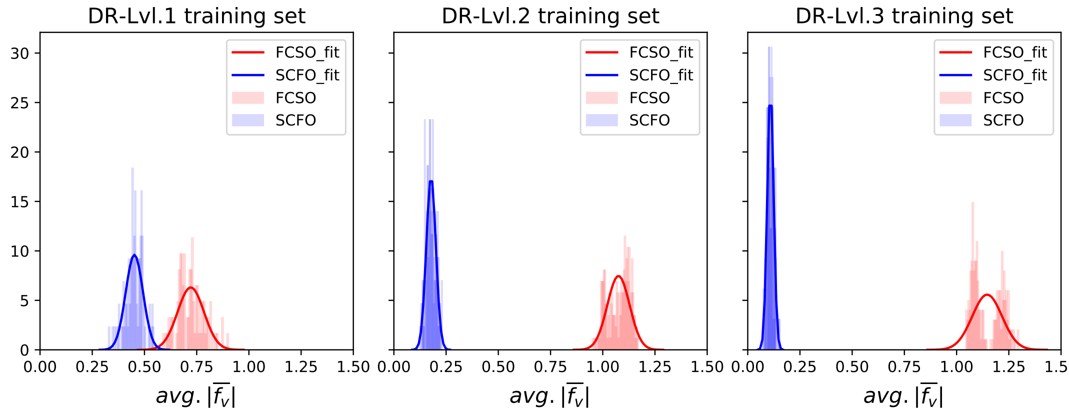


Figure 5.7: Visualization of the data assignment results on the DR-Lvl.1 $\sim$ 3 datasets.

5C.2 MJC-Arm (Mujoco-Arm)

Simulation Environment The environment is built with MuJoCo physics simulator (Todorov et al., 2012), and the Franka Emika robot arm with a Barret hand attached to 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:

$$\textbf{random.choice}_{FCSO}(\{0, 0.1, 0.2, ..., 1\}, \text{probs} = \{0.34, 0.34, 0.25, 0.07, 0.0, ..., 0.0\})$$

$$\textbf{random.choice}_{SCFO}(\{0, 0.1, 0.2, ..., 1\}, \text{probs} = \{0.0, ..., 0.0, 0.07, 0.25, 0.34, 0.34\})$$

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.

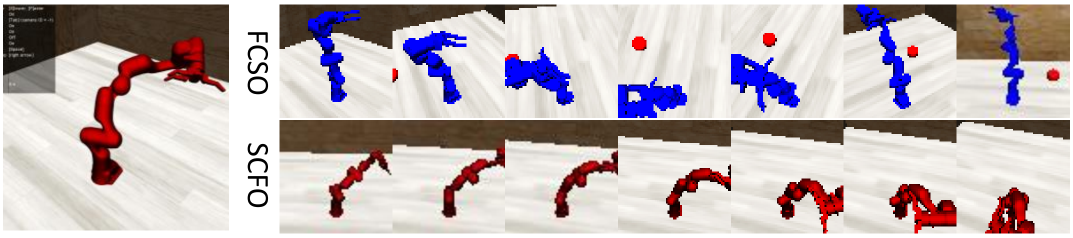


Figure 5.8: **Left:** Mujoco simulation environment. **Right:** One *fast-camera-slow-object* (FCSO) sample (**top row**) and *slow-camera-fast-object* (SCFO) sample. Both are randomly selected from the MJC-Arm dataset.

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 trial, 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.

5C.3 Real-World Data (CubeLand)

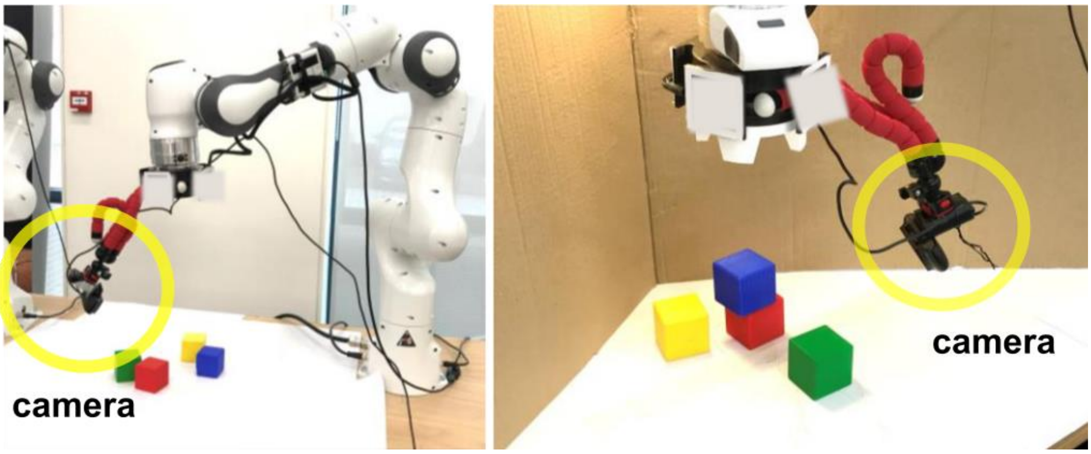


Figure 5.9: CubeLand data-collection platform.

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 unnecessary background clutter, a bicolor data collection environment was set up with white surface and brown walls. A camera was mounted on the end effector of Franka arm (a robotic arm with 7 D.O.F.) as shown in Figure 5.9. The end effector had a fixed motion, i.e., it only rotated back and forth 120 degrees. The cubes were connected by 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 5.10). 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 and the radius of the assembly (center of the end effector to the camera) spanned 14.5 cm and 19.5 cm, respectively.

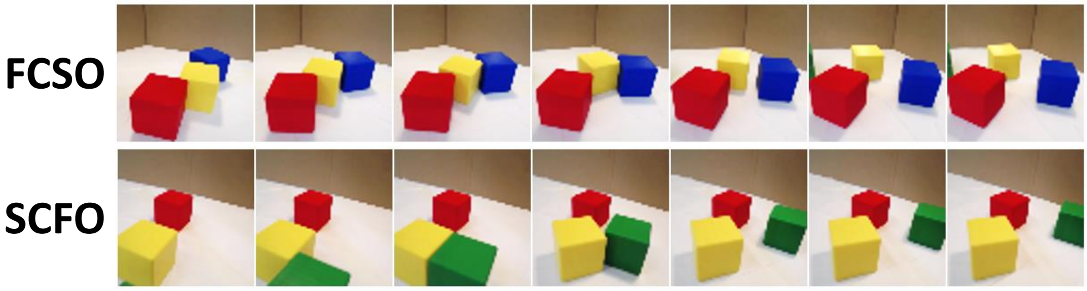


Figure 5.10: CubeLand data samples. **Top:** a fast camera, slow objects (FCSO) data sample. **Bottom:** a slow camera, fast objects (SCFO) data sample.

Appendix 5D. Additional Results

5D.1 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. 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 5.11). To analyze the independent effects of $\Delta t_{\mathbf{z}}$ and Δt_v, we "squeezed" 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 5.11 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 relatively difficult. One might run pre-analysis before training, e.g. visually look several sequences,

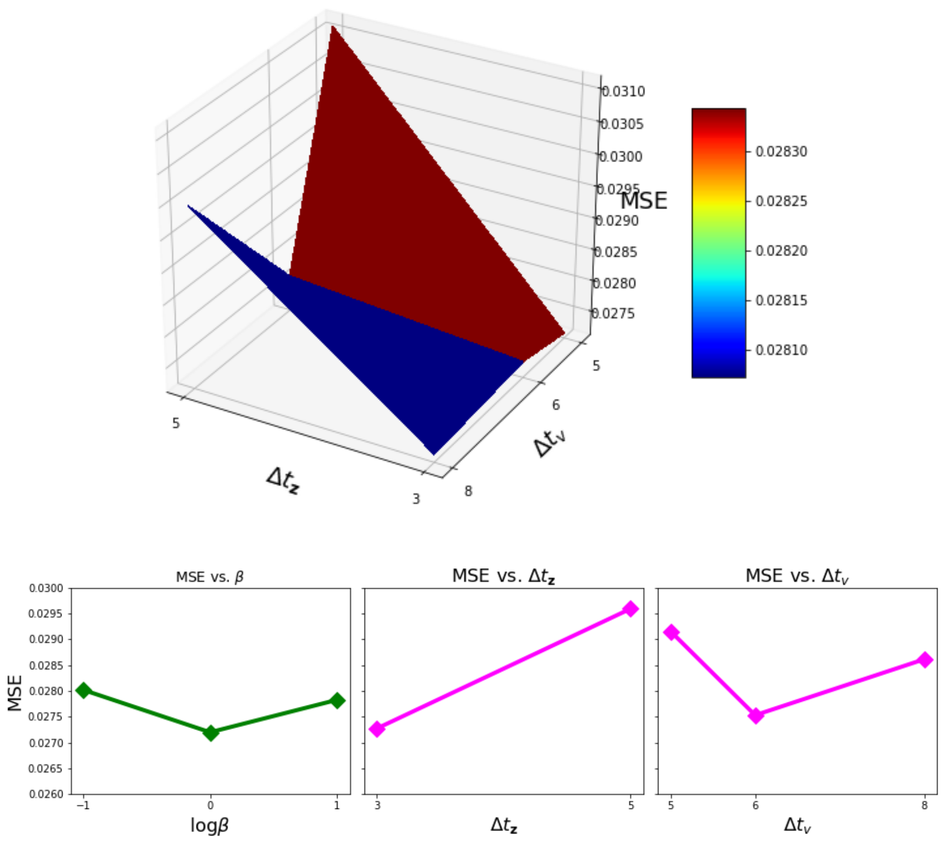


Figure 5.11: Ablation study results. **Top:** Space-time queried view synthesis MSE vs. nested $\Delta t_\mathbf{z}$ and Δt_v. **Bottom left:** MSE vs. different β (in $\log_2$ space). **Bottom middle:** MSE vs. different $\Delta t_\mathbf{z}$ (MSE computed by averaging across different Δt_v). **Bottom right:** MSE vs. different Δt_v (MSE computed by averaging across different $\Delta t_\mathbf{z}$).

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 relatively insensitive results in the bottom left of Figure 5.11.

5D.2 T-GQN Results

We used the official implementation of T-GQN[7] and trained a T-GQN on the DR-Lvl.3 data. Although the training has converged (see Figure 5.13), we observe

[7] https://github.com/singhgautam/snp

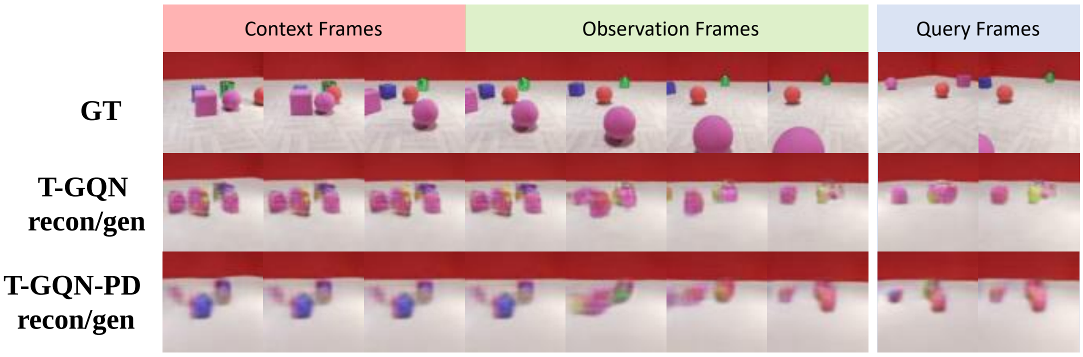


Figure 5.12: Qualitative results of T-GQN on DR-Lvl.3 test data.

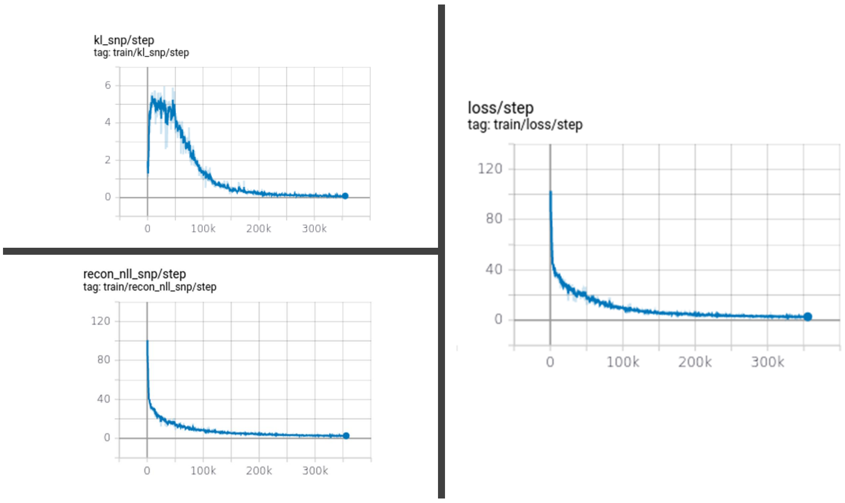


Figure 5.13: T-GQN training curves. We train t-GQN on our DRoom data until it converges.

that it fails to represent the underlying 3D scenes (see Figure 5.12) 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.

5D.3 Additional Qualitative Results

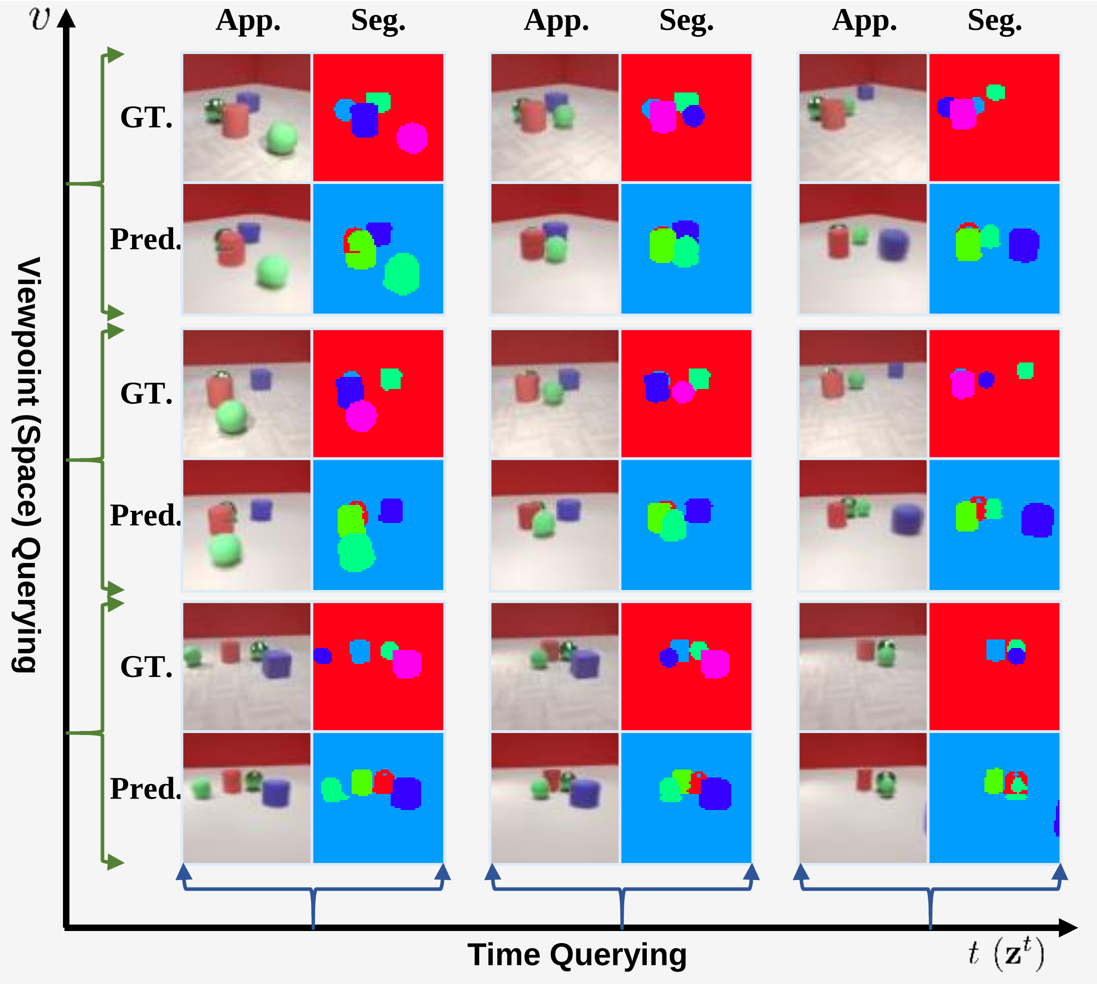


Figure 5.14: Spatial-temporal factorization results of a DRoom scene.

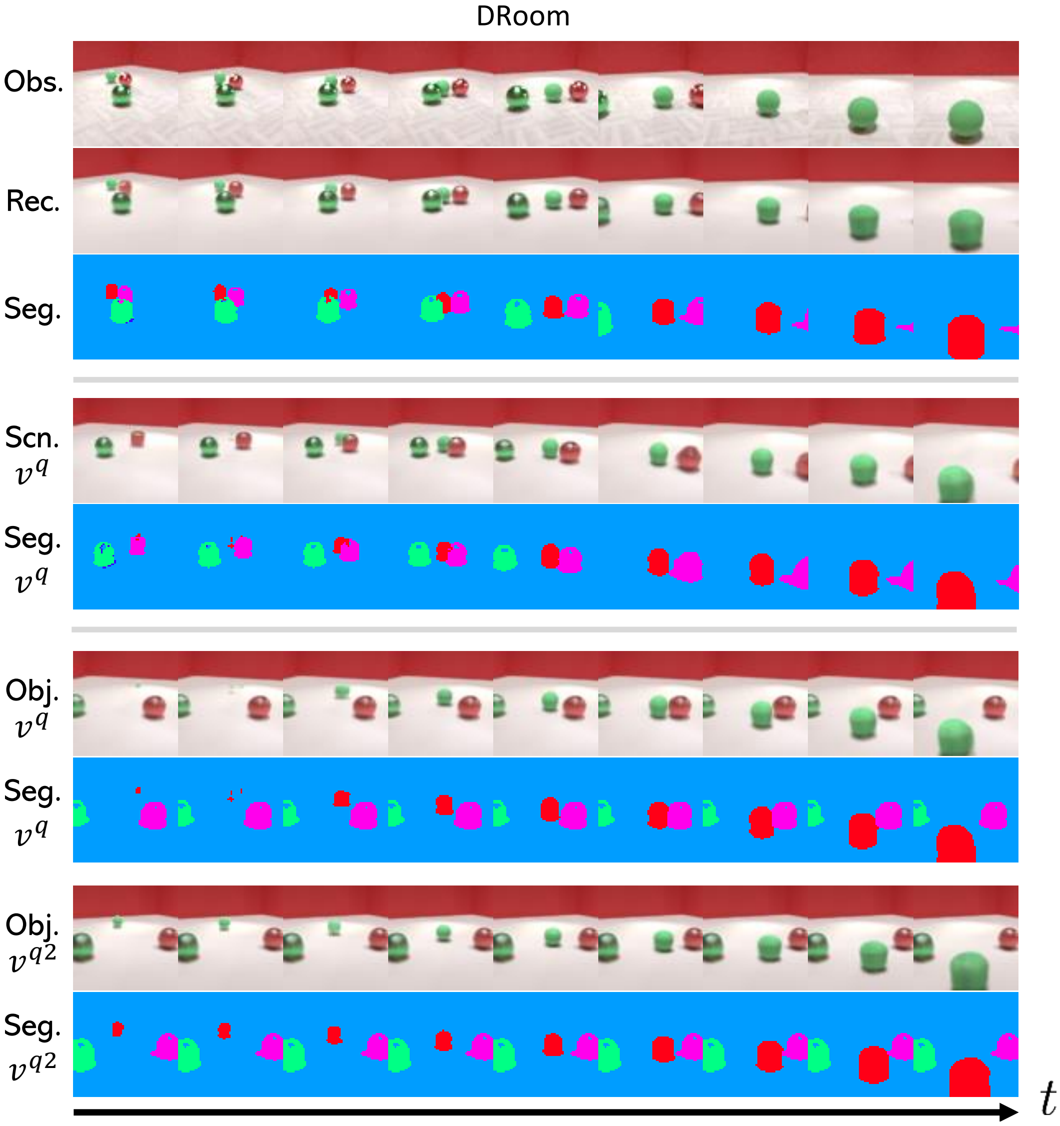


Figure 5.15: Dynamics replay of a DRoom scene.

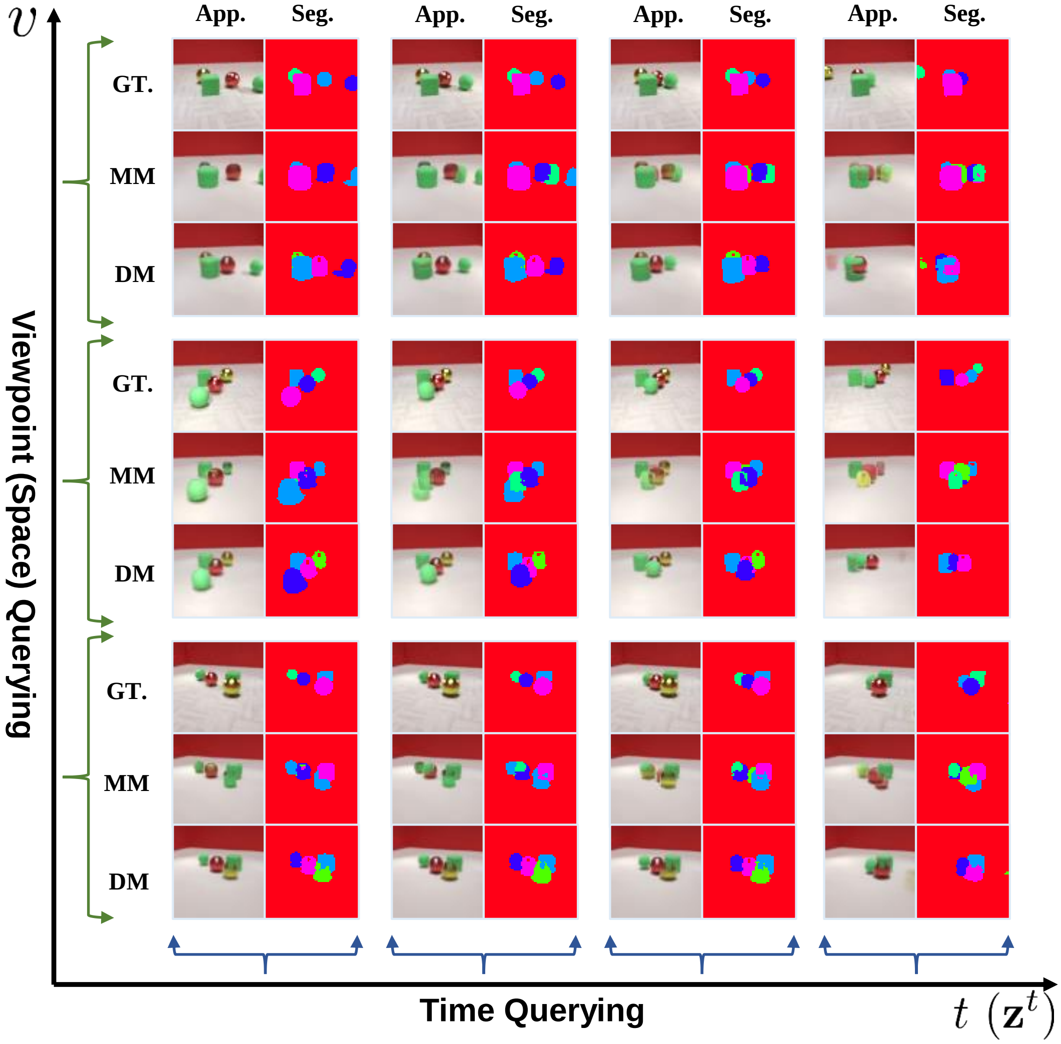


Figure 5.16: Qualitative comparisons: DyMON vs. MulMON in spatial-temporal factorization (on DRoom). We train DyMON on DR-Lvl.3 and train MulMON on DR0-$|\overline{f_{\mathbf{z}}}|$.

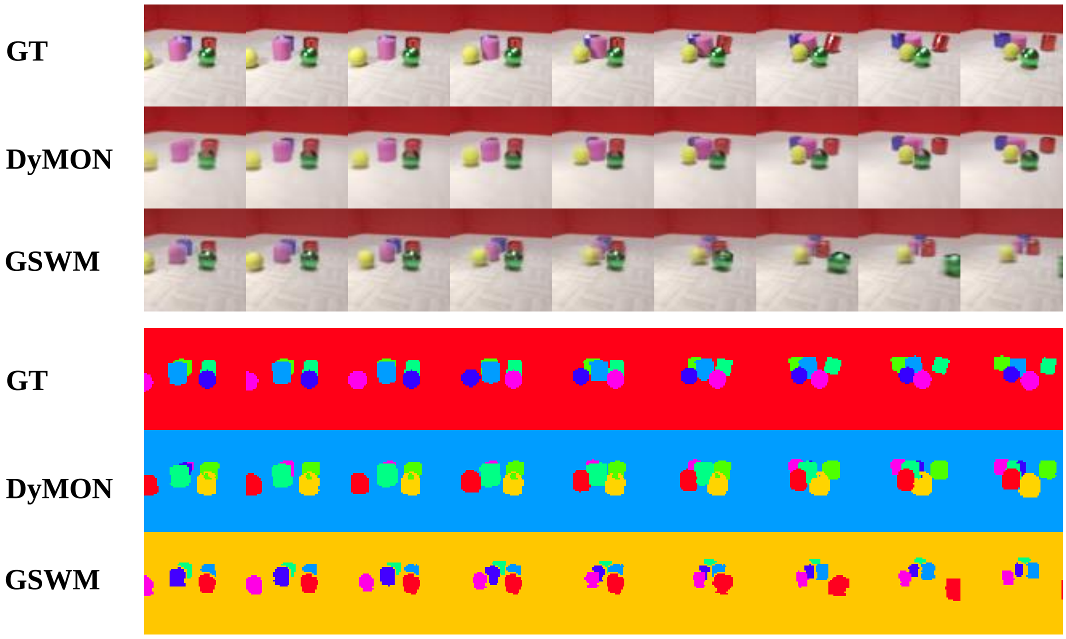


Figure 5.17: Qualitative comparisons of DyMON and GSWM on DR0-$|\overline{f_v}|$. **Top:** reconstruction performance. **Bottom:** segmentation performance (we observe that DyMON outperforms GSWM in segmenting scenes).

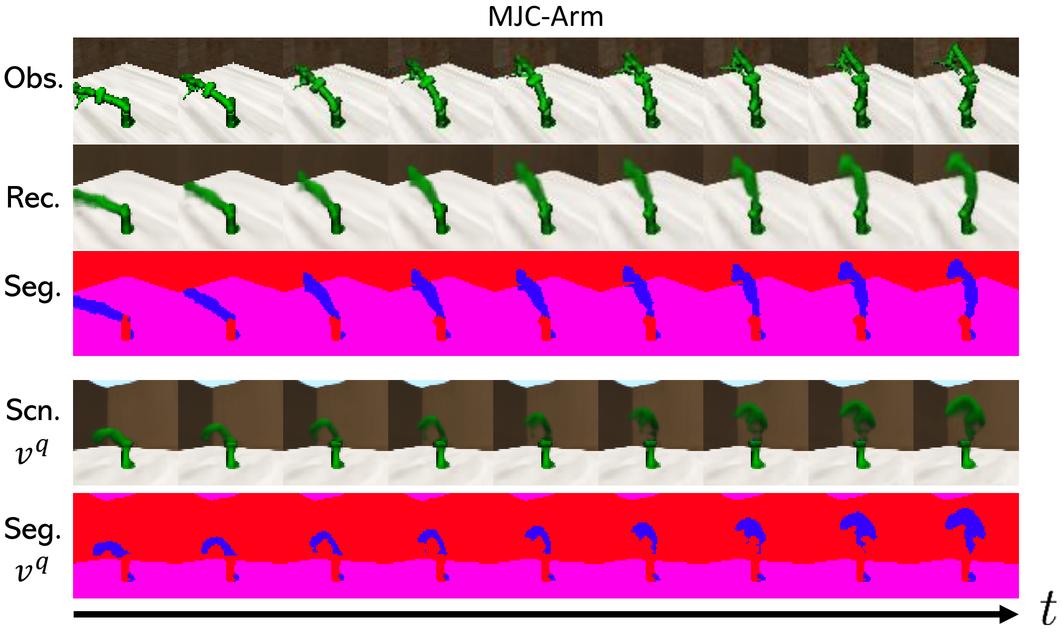


Figure 5.18: Dynamics replay of a MJC-Arm scene.

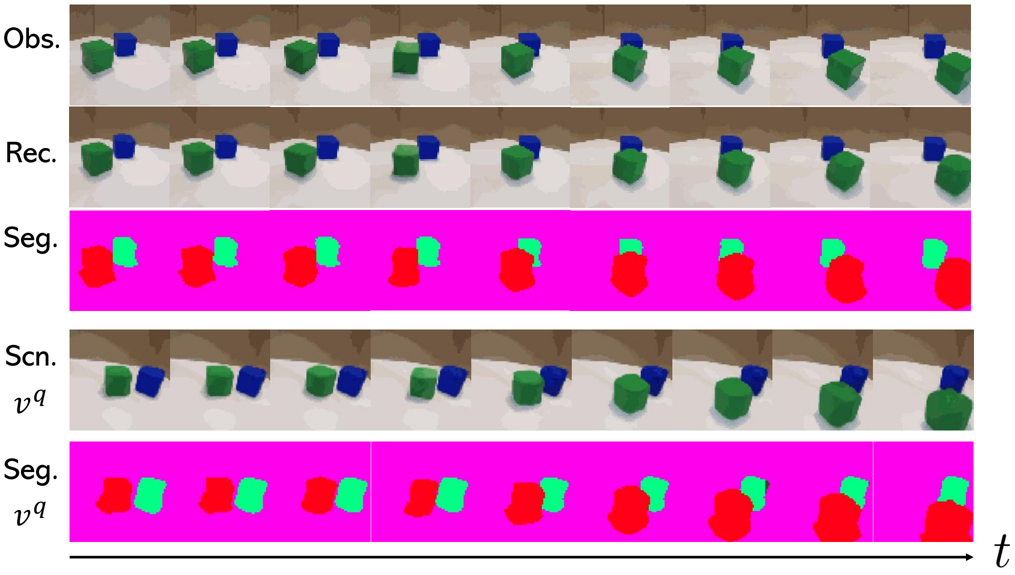


Figure 5.19: Dynamics replay of a real scene (i.e. CubeLand data). We conduct experiments on real-world data to show DyMON's potential for real-world applications.

Chapter 6

Discussions

In this thesis, we have explored the idea of empowering machines to understand visual compositionality, which holds great promise in improving the **i)** *systematic generalization* and **ii)** *representation interpretability* limitations of most existing machine learning systems. Specifically, we established our study in the scenarios of *object-centric representation learning* (abbr. OCRL). In this case, our goal is to address the *factorization* problem such that the artificial systems can uncover the compositional generative structures around objects that underlie the scene observation data. We have investigated three common issues in object-centric factorization, i.e. **i)** *single-view ambiguities* (see chapter 3), **ii)** *latent-representation duplicates* (see chapter 4), and **iii)** *temporal-structure entanglement* (see chapter 5), and proposed three methods that effectively handle these issues. Moreover, we have approached these three issues from a generative perspective *without* any supervision.

6.1 Summary Of Contributions

In **Chapter 3**, we presented **MulMON** as a method for learning accurate, object-centric representations of multi-object scenes by leveraging multiple views.

We have shown that MulMON's ability to aggregate information across multiple views does indeed allow it to better-resolve spatial ambiguity (or uncertainty) and better-capture 3D spatial structures, and as a result, outperform state-of-the-art models for unsupervised object segmentation. We have also shown that, by virtue of addressing the more complicated multi-object-multi-view scenario, MulMON achieves new functionality—the prediction of both appearance and object segmentations for novel viewpoints, i.e. it has learned about objects and 3D structure. We believe the object-wise multi-view uncertainty reduction design of MulMON can make it a promising element in downstream tasks that involve environment exploration and interaction.

In **Chapter 4**, we presented a **differentiable prior (the *LDS* prior)** that leverages similarity measures to regulate the object-centric latent representations inferred by multi-object VAEs, i.e. CompVAEs. Despite its simplicity, we have demonstrated its effectiveness in fixing known issues, namely the *uniqueness issues*, of the multi-object VAE models — inferring duplicate object representations. We ascribed the *uniqueness issues* to the violation of the *uniqueness assumption* that is implicitly introduced by the scene-mixture-model assumption, i.e. each part of a scene observation (e.g. a pixel) must be explained by one and only one scene object. Therefore, we have demonstrated through experiments that suppressing duplicates can lead to better variational approximation and task performance.

In **Chapter 5**, we presented the **Dynamics-aware Multi-Object Network (DyMON)** as a method for learning object-centric representations in a *multi-view-dynamic-scene* setting. We have made two weak assumptions that allow 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.* We have shown through our experiments that such *spatial-temporal factorization* permits querying the predictions of scene appearances and segmentations across both space and time. We believe the ability to perform counterfactual reasoning about space and time (as well as objects) is essential for building world models (Ha and Schmidhuber, 2018) and training intelligent agents.

6.2 Limitations and Future Works

The Severe Gap To Real-world Applications The study of *generative object-centric representation learning* is still in an early stage. Though the existing methods, including the three methods published in this thesis, have demonstrated great success in well-controlled environments, e.g. CLEVR (Johnson et al., 2017) and Mujoco (Todorov et al., 2012), they are far from being ready for real-world applications with *texture-rich* scenes (see an example in Greff et al. 2019). According to Eslami et al. (2018); Greff et al. (2019), such limitation is likely resulting from the insufficient computational capacity for generative modeling of high-dimensional data with complex structures. Regarding this limitation, we consider observation complexity reduction by ignoring trivial details (e.g. using super-pixel methods Achanta et al. 2012) a promising direction to explore. However, it is often tricky to answer "what details should be ignored"—this connects to the discussions of *granularity* (see the last paragraph of this section).

Dynamics Modeling and Extrapolation As discussed in chapter 5, DyMON focuses on capturing the spatial status of the scene objects at every specific time point rather than describing their full trajectories. As a result, although DyMON can predict a scene's appearances and segmentations at arbitrary past times (querying across time) and from arbitrary viewpoints (querying across space), it cannot predict the future evolution of scenes. In other words, DyMON supports interpolating query points within the observed time but *not* extrapolating.

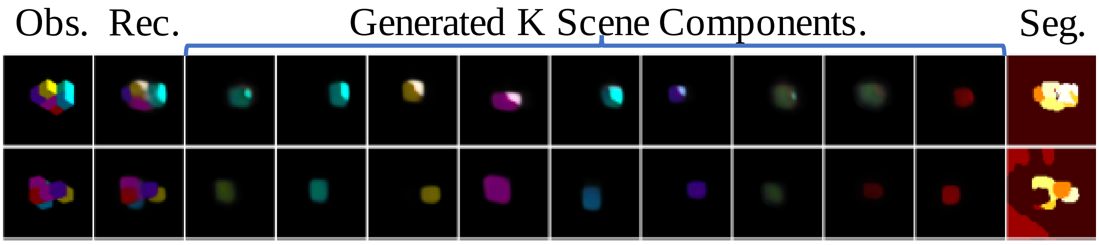


Figure 6.1: MulMON treats a Shep7 object as a composition of parts (cubes) instead of one object. Adapted from Figure 3.12.

Enabling a model to describe the scene objects' trajectories and interactions is necessary for *physics understanding* (Jaques et al., 2020; Lin et al., 2020) and *intuitive physics planning* (Janner et al., 2019; Smith et al., 2019). If we treat "predicting the future evolution of scenes" as an *initial-value* problem (Coddington and Levinson, 1955), the extrapolation limitation can be potentially resolved by fitting ordinary differential equations, e.g. using Neural ODE (Chen et al., 2018b), to describe the scene object dynamics in the representation space.

On The Granularity of Compositional Structures We have shown in chapter 3 that MulMON successfully uncovered both *object-level* and *feature-level* compositional structures. However, as shown in Figure 6.1, MulMON seems to fail in disentangling "objects" with a "desired" granularity—i.e. it treats a Shep7 object as a composition of parts (cubes) instead of one object. This raises an intriguing question: "what levels of granularities are desired?", which is closely connected to the long-standing discussions of *objectness*, i.e. the definition of "object" (Caesar et al., 2018; Kosiorek, 2020), but more general. The "objectness" discussion often implies a setting where only one level of granularity is of interest, while we are interested in modeling compositional structures at multiple granularities. For example, in the settings of OCRL, one potential future exploration can be shaping an interpretable latent space in the hierarchy of "features$\rightarrow$ $\overset{\circlearrowleft}{\text{objects}}$ $\rightarrow$scenes".

6.3 Societal Impact

In this thesis, we presented methods for learning object-centric representations of multi-object scenes. Object-centric scene representations can support many downstream tasks, such as autonomous scene exploration, object segmentation (tracking) and scene synthesizing.

Autonomous scene exploration has real-world applications in exploring hazardous environments, mines, potential bomb threats, nuclear waste zones. This could have societal impacts through increased worker safety or potential military (mis)uses.

Object detection and tracking has real-world applications in tracking people in CCTV footage, detecting buildings from aerial footage, and spotting potential hazards for autonomous vehicles. Potential societal impacts include safer autonomous vehicles and unwanted/increased surveillance.

Finally, scene synthesizing has applications in automated scene modeling for computer games. This further transitions society away from labor-intensive tasks to higher-level cognitive tasks. This could have both positive (more time for cognitive tasks) and negative (less employment) impacts on society.

Bibliography

R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

A. Arsalan Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1511–1519, 2017.

J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó. The slam problem: a survey. *Artificial Intelligence Research and Development*, pages 363–371, 2008.

V. Bapst, A. Sanchez-Gonzalez, C. Doersch, K. L. Stachenfeld, P. Kohli, P. W. Battaglia, and J. B. Hamrick. Structured agents for physical construction. In *International Conference on Machine Learning*, pages 464–474, 2019.

D. Barber. *Bayesian reasoning and machine learning.* Cambridge University Press, 2012.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *International Conference on Learning Representations*, 2020.

I. Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5561–5569, 2017.

L. Buesing, T. Weber, Y. Zwols, S. Racaniere, A. Guez, J.-B. Lespiau, and N. Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

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.

H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

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.

R. B. Cattell. A biometrics invited paper. factor analysis: An introduction to essentials i. the purpose and underlying models. *Biometrics*, 21(1):190–215, 1965.

C. Chen, F. Deng, and S. Ahn. Roots: Object-centric representation and rendering of 3d scenes. *Journal of Machine Learning Research*, 22:259–1, 2021.

R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in Neural Information Processing Systems*, 31, 2018a.

R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018b.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 29, 2016.

K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

E. A. Coddington and N. Levinson. Theory of ordinary differential equations. 1955.

P. Comon. Independent component analysis, 1992.

P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. doi: 10.1109/34.927467.

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61 (1):38–59, 1995.

C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018.

B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

R. Dang-Nhu. Evaluating disentanglement of structured representations. In *International Conference on Learning Representations*, 2021.

G. Desjardins, A. Courville, and Y. Bengio. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*, 2012.

T. DeVries, M. A. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14304–14313, 2021.

A. Didolkar, A. Goyal, R. Ke, C. Blundell, P. Beaudoin, N. M. O. Heess,

M. Mozer, and Y. Bengio. Neural production systems. In *Advances in Neural Information Processing Systems*, 2021.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

S. Duan, L. Matthey, A. Saraiva, N. Watters, C. P. Burgess, A. Lerchner, and I. Higgins. Unsupervised model selection for variational disentangled representation learning. *arXiv preprint arXiv:1905.12614*, 2019.

C. Eastwood and C. K. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.

P. Emami, P. He, S. Ranka, and A. Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *International Conference on Machine Learning*, pages 2970–2981. PMLR, 2021.

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.

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

S. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, G. E. Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in Neural Information Processing Systems*, 29, 2016.

S. A. Eslami, D. Jimenez 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.

B. Esmaeili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. Meent. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2525–2534. PMLR, 2019.

B. Fruchter. Introduction to factor analysis. 1954.

G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

A. Goyal, A. Lamb, P. Gampa, P. Beaudoin, S. Levine, C. Blundell, Y. Bengio, and M. Mozer. Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. In *International Conference on Learning Representations*, 2020.

R. E. Grandy. Understanding and the principle of compositionality. *Philosophical Perspectives*, 4:557–572, 1990.

K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

K. Greff, A. Rasmus, M. Berglund, T. Hao, H. Valpola, and J. Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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

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.

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

L. Gresele, J. Von Kügelgen, V. Stimper, B. Schölkopf, and M. Besserve. Independent mechanism analysis, a new concept? *Advances in Neural Information Processing Systems*, 34:28233–28248, 2021.

D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. ISBN 0521540518.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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.

I. Higgins, N. Sonnerat, L. Matthey, A. Pal, C. P. Burgess, M. Bošnjak, M. Shanahan, M. Botvinick, D. Hassabis, and A. Lerchner. Scan: Learning hierarchical compositional visual concepts. In *International Conference on Learning Representations*, 2018.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, page 1735–1780, 1997.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

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.

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

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

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

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

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.

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

D. Jimenez Rezende, S. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. *Advances in Neural Information Processing Systems*, 29, 2016.

J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

R. Kabra, D. Zoran, G. Erdogan, L. Matthey, A. Creswell, M. Botvinick, A. Lerchner, and C. Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34:20146–20159, 2021.

H. Kamp and B. Partee. Prototype theory and compositionality. *Cognition*, 57 (2):129–191, 1995.

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

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

T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2019.

V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002.

A. Kosiorek. Learning object-centric representations, 2020.

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.

A. R. Kosiorek, H. Strathmann, D. Zoran, P. Moreno, R. Schneider, S. Mokrá, and D. J. Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pages 5742–5752. PMLR, 2021.

J. Kossen, K. Stelzner, M. Hussing, C. Voelcker, and K. Kersting. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations*, 2020.

T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, and V. Mansinghka. Picture: A probabilistic programming language for scene perception. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4390–4399, 2015a.

T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. *Advances in Neural Information Processing Systems*, 28, 2015b.

A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018.

K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

B. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR, 2018.

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Y. Liao, S. Donne, and A. Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018.

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.

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*, pages 6140–6149. PMLR, 2020.

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, 2019a.

F. Locatello, M. Tschannen, S. Bauer, G. Rätsch, B. Schölkopf, and O. Bachem. Disentangling factors of variation using few labels. 2019b.

F. Locatello, B. Poole, G. Rätsch, B. Schölkopf, O. Bachem, and M. Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020a.

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*, 2020b.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

D. Mambelli, F. Träuble, S. Bauer, B. Schölkopf, and F. Locatello. Compositional multi-object reinforcement learning with linear relation networks. In *ICLR 2022 Workshop on Generalizable Policy Learning in Physical World*, 2022.

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

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019.

B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38 (4):1–14, 2019.

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.

T. P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Massachusetts Institute of Technology, 2001.

M. Minsky. *Society of mind.* Simon and Schuster, 1988.

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.

S. Mohamed and B. Lakshminarayanan. Learning in implicit generative models. In *International Conference on Learning Representations*, 2017.

G. E. Moran, D. Sridhar, Y. Wang, and D. M. Blei. Identifiable variational autoencoders via sparse decoding. *arXiv preprint arXiv:2110.10804*, 2021.

P. Moreno, C. K. Williams, C. Nash, and P. Kohli. Overcoming occlusion with inverse graphics. In *European Conference on Computer Vision*, pages 170–185, 2016.

K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

L. Nanbo and R. B. Fisher. Duplicate latent representation suppression for multi-object variational autoencoders. In *The 32nd British Machine Vision Conference*, 2021.

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.

L. Nanbo, M. A. Raza, W. Hu, Z. Sun, and R. Fisher. Object-centric representation learning with generative spatial-temporal factorization. In *Advances in Neural Information Processing Systems*, 2021.

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. Ieee, 2011.

M. Niemeyer and A. Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021.

J. Pearl. The do-calculus revisited. *arXiv preprint arXiv:1210.4852*, 2012.

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

E. Penner and L. Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

T. Porter and T. Duff. Compositing digital images. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 253–259, 1984.

R. A. Potamias, S. Ploumpis, and S. Zafeiriou. Neural mesh simplification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18583–18592, 2022.

S. J. Prince, J. H. Elder, J. Warrell, and F. M. Felisberti. Tied factor analysis for face recognition across large pose differences. *IEEE Transactions on pattern analysis and machine intelligence*, 30(6):970–984, 2008.

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.

A. G. Reddy, B. G. L, and V. N. Balasubramanian. On causally disentangled representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

P. Reizinger, L. Gresele, J. Brady, J. von Kügelgen, D. Zietlow, B. Schölkopf,

G. Martius, W. Brendel, and M. Besserve. Embrace the gap: Vaes perform independent mechanism analysis. *arXiv preprint arXiv:2206.02416*, 2022.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

A. Revonsuo and J. Newman. Binding and consciousness. *Consciousness and cognition*, 8(2), 1999.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014.

D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems*, pages 4996–5004, 2016.

K. Ridgeway and M. C. Mozer. Learning deep disentangled embeddings with the f-statistic loss. *Advances in Neural Information Processing Systems*, 31, 2018.

M. Rolinek, D. Zietlow, and G. Martius. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12406–12415, 2019.

E. Rosch and C. B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.

A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971.

R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer, 2014.

J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4(6):863–879, 1992.

J. Schmidhuber. On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. *arXiv preprint arXiv:1511.09249*, 2015.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1255–1262, 2012.

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. doi: 10.1109/JPROC.2021.3058954.

A. Shapiro. Identifiability of factor analysis: Some results and open problems. *Linear Algebra and its Applications*, 70:1–7, 1985.

P. Sharma, A. Tewari, Y. Du, S. Zakharov, R. Ambrus, A. Gaidon, W. T. Freeman, F. Durand, J. B. Tenenbaum, and V. Sitzmann. Seeing 3d objects in a single image via self-supervised static-dynamic disentanglement. *arXiv preprint arXiv:2207.11232*, 2022.

Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

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

E. E. Smith and D. N. Osherson. Conceptual combination with prototype concepts. *Cognitive science*, 8(4):337–361, 1984.

K. Smith, L. Mei, S. Yao, J. Wu, E. Spelke, J. Tenenbaum, and T. Ullman. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. *Advances in neural information processing systems*, 32, 2019.

K. Stelzner, K. Kersting, and A. R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

R. Strudel, R. Garcia, I. Laptev, and C. Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23(12):1–83, 2022.

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, 2019.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61 (3):611–622, 1999.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

J. Tobin, W. Zaremba, and P. Abbeel. Geometry-aware neural rendering. *Advances in Neural Information Processing Systems*, 32, 2019.

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. doi: 10.1109/IROS.2012.6386109.

F. Träuble, E. Creager, N. Kilbertus, F. Locatello, A. Dittadi, A. Goyal, B. Schölkopf, and S. Bauer. On disentangled representations learned from correlated data. In *International Conference on Machine Learning*, pages 10401–10412. PMLR, 2021.

M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.

S. Van Steenkiste, F. Locatello, J. Schmidhuber, and O. Bachem. Are disentangled representations helpful for abstract visual reasoning? *Advances in Neural Information Processing Systems*, 32, 2019.

H. Von Helmholtz. Treatise on physiological optics vol. iii. 1867.

A. Voynov and A. Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020.

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.

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.

J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, 2016.

Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets:

A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019.

S. Yao, T. M. Hsu, J.-Y. Zhu, J. Wu, A. Torralba, B. Freeman, and J. Tenenbaum. 3d-aware scene manipulation via inverse graphics. *Advances in Neural Information Processing Systems*, 31, 2018.

I. Yildirim, M. Janner, M. Belledonne, C. Wallraven, W. Freiwald, and J. Tenenbaum. Causal and compositional generative models in online perception. In *CogSci*, 2017.

A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021a.

H.-X. Yu, L. Guibas, and J. Wu. Unsupervised discovery of object radiance fields. In *International Conference on Learning Representations*, 2021b.

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

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

S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.