ORSA-T: Multi-View Object-Centric Scene Representation Learning with Slot Attention and Transformer

Henri Placek, Chris Child, Tillman Weyde
Department of Computer Science
City St George's, University of London, United Kingdom
{Henri.Placek,C.Child,T.E.Weyde}@citystgeorges.ac.uk

***Abstract*—Understanding a scene from multiple, potentially partial views and decomposing it into objects is foundational for human perception and intelligence. Current multi-view object-centric scene representation learning models that use partial views analyze all views at once. This differs from the way humans process visual information and is not compatible with reinforcement learning, where an agent learns about its environment through actions, such as moving to change the viewpoint. In this paper, we propose ORSA-T (Object-centric scene Representation learning with Slot Attention and Transformer), which combines Implicit Slot Attention with an aggregation of previous views by a Transformer and improves the scene representation iteratively based on a sequence of images annotated with viewpoints. The Transformer uses all previous representations and the current update to aggregate scene information, which makes ORSA-T remember objects better and learn more effectively when applied to partial views. In our experiments, ORSA-T predicts and segments images from a new viewpoint better than MulMON, the current SOTA, and ORSA without aggregation connections and Transformer. As ORSA-T learns iteratively to improve its scene representation, it is suitable for use in reinforcement learning.**

I. Introduction

Human perception of environments by decomposition into objects and their relationships, and the improvement of this perception over time and with movement has been studied for decades [1]–[3]. Machine learning of object-centric representations to replicate this process has received much attention recently [4]. The hope is that these models contribute to the development of more human-like AI algorithms, which are more interpretable and generalizable [5], [6], and better facilitate downstream tasks like visual reasoning [7] and reinforcement learning [8], [9].

In recent years, numerous models have emerged that transform an image from a single viewpoint into objects in an unsupervised manner, such as MONet [10], IODINE [11], SPACE [12], Slot Attention [13], Invariant Slot Attention [14], and Implicit Slot Attention [15]. However, this task does not require the model to understand the spatial structure of the scene. Scene rendering, which generates novel views after learning a scene representation from partial views, has also undergone significant development, leading to GQN [16], SRN [17] or NeRF-VAE [18]. However, the representations resulting from these algorithms are not object-centric. Some methods such as ObSuRF [19] learn object-centric scene representations by processing a single image and generate new views from different viewpoints, but they are ineffective on datasets with partial views. Only a few models address learning object-centric representations from multiple viewpoints and they use different set-ups. SIMONe [20] and OCLOC [21] use views without viewpoint annotations. In contrast, ROOTS [22], OSRT [23] and MulMON [24] require both views and viewpoints. SIMONe, OCLOC, ROOTS and OSRT process all the views from which they learn simultaneously, which allows aggregating information from partial views of the scene easily. However, this approach is not like human perception which is serial in nature and these models cannot be used in reinforcement learning, where agents improve their understanding of the environment after every move. Object-centric learning models for videos, such as SAVi [25] or STEVE [26], also utilize multiple views without any notion of viewpoint. While they are capable of tracking objects, they are not able to generate views from unseen viewpoints. MulMON, on the other hand, processes views successively and improves the representation after each view. The downside of

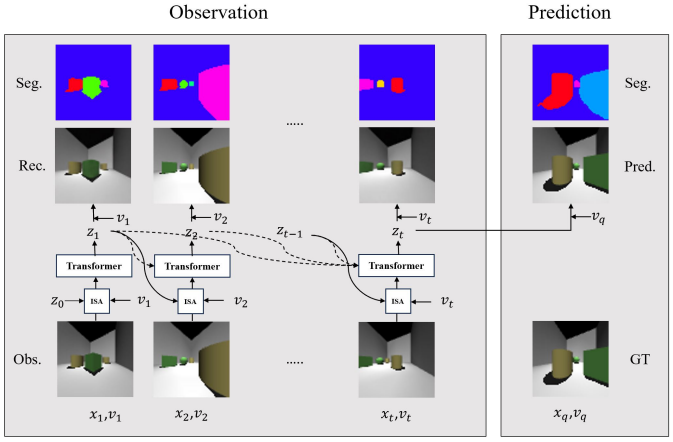

Fig. 1. ORSA-T overview: during observation, our model improves the representation z_{t-1} with each observed view, consisting of the image x_t and viewpoint v_t. It uses the improved representation z_t and v_t to reconstruct and segment x_t. There are two novel elements in ORSA-T. First, we use Implicit Slot Attention (ISA) to update the previous scene representation. Second, we aggregate all previous scene representations and the update with a Transformer. This enables our model to better remember previously seen objects. The model is trained by back-propagating the the image difference loss during observation and prediction. Removing the transformer modules and their connections, represented by dotted lines, yields the diagram of ORSA without Transformer.

the MulMON method is that it is prone to forgetting objects previously seen when using partial views.

We propose here ORSA-T (Object-centric Scene Representation Learning with Slot Attention and Transformer), an unsupervised incremental representation learning model that improves its scene representation with each new view. See Fig. 1 for ORSA-T overview. It factorizes a scene into object representations, enabling it to reconstruct, predict and segment views. Our model has a similar structure to MulMON, but it uses Implicit Slot Attention to update the previous representation using the current input. We extend Implicit Slot Attention with an aggregation step. In this step, all previous scene representations are combined with the update and refined into a new representation using a Transformer [27]. This architecture enables the model to better remember previously seen objects, making it effective for partial-view datasets. It is able to predict views for novel viewpoints unlike object-centric learning models for videos or multi-view models that do not use viewpoints. Because ORSA-T improves representations sequentially for each view, it can be a pretrained module in a reinforcement learning setting, unlike models that process all views at once. The main contributions of this study are: (1) proposing ORSA-T, an unsupervised model that uses transformers to retain information about previously observed objects; (2) evaluating ORSA-T on three datasets and analyzing the impact of transformers; and (3) comparing ORSA-T to ORSA and MulMON.

II. Related work

a) Single-view object representation learning: algorithms have been developed in recent years that learn object-centric representations and produce good segmentations. AIR [28], MONet and GENESIS [29] use a sequential attention mechanism. SPAIR [30] and SPACE [12] rely on the spatial location of objects. IODINE [11], Slot Attention [13], and Invariant Slot Attention [14] start with random or learned representations, and refine them iteratively with different methods.

b) Multi-view scene rendering: these techniques learn a single scene representation from a set of images with viewpoints and generate realistic images from novel viewpoints. Examples of multi-view scene rendering models include GQN [16], SRN [17], Deepvoxels [31], NeRF [32] and NeRF-VAE [18].

c) Objects tracking in videos: this is a multi-view problem, where the scene is dynamic with objects in motion. The objective is to detect, track objects, and reconstruct images. Additionally, some approaches have the capability to predict subsequent video frames. These methods are largely based on single-view object representation learning, such as SQAIR [33] on AIR, VIMON [34] on MONet, and SAVi, SAVi++ [35], SlotFormer [36] and STEVE on Slot Attention.

d) Single-view scene object representation learning: models, such as ObSuRF [19], uORF [37] and COLF [38], learn scene representation from a single view, from which they can create novel views and segment them. All of these models combine Slot Attention to find the representation and NeRF to decode them, only uORF models object and background representations separately.

e) Multi-view scene object representation learning: has two different setups: one where viewpoints are known and another where viewpoints are not known. Algorithms like SIMONe [20] and OCLOC [21] factorize images without viewpoint information into scene object representation and viewpoint representation. While these approaches provide high-quality reconstruction and segmentation, they cannot generate new views, as the viewpoint information is internal to the model. If the viewpoints are known, models like ROOTS, ORST, and MulMON can also generate views from novel viewpoints. ROOTS and OSRT process all views they learn on simultaneously. ROOTS creates a 3D grid and utilises GQN. OSRT encodes the view into features using a Convolutional Neural Network (CNN) [39]. These flattened features are then processed by a Transformer encoder and Slot Attention to obtain scene object representations. MulMON, on the other hand, processes these views sequentially using IODINE and refines the representation in each step.

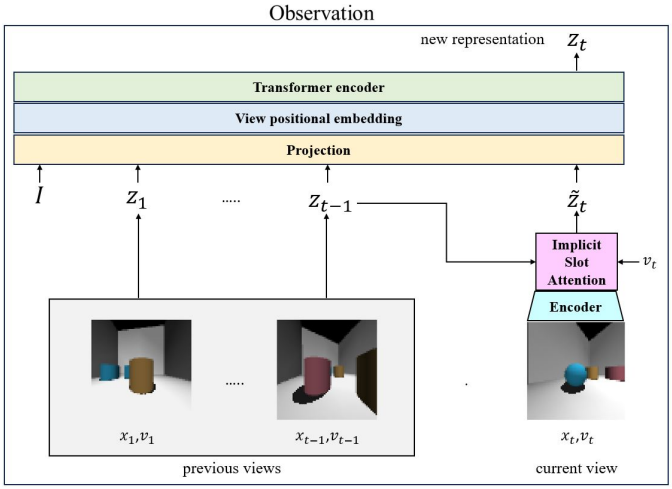

Fig. 2. ORSA-T processing of a single view during observation. First, Implicit Slot Attention produces an *update* $\tilde{z}_t$ from the current image x_t with viewpoint v_t and the previous scene representation z_{t-1}. An initial token I and all previous scene representations $z_1, ..., z_{t-1}$ and the *update* $\tilde{z}_t$ are input to a Transformer that outputs what we call the *refined* representation z_t.

III. Method

Our goal is to transform a set of views, i.e., images with their viewpoints, $\mathcal{T} = \{(x_t, v_t)\}_{t=1}^{T}$ into an object-centric scene representation $\{z^k\}_{k=1}^{K}$, where K is the chosen number of objects, which captures implicitly the 3D structure of the scene. Our method refines the scene representation by iterating through the views. It starts with an initial representation $\{z_0^k)\}_{k=1}^{K}$ or $z_0 \in \mathbb{R}^{K \times D_{slots}}$ where K is the number of vectors and D_{slots} is the representation dimension. Similar to SAVI [25], SAVI++ [35], Invariant Slot Attention [14] and BO-QSA [40], z_0 is not initialised randomly, but is a trainable parameter. Similar to MulMON, for each view, the representation from the previous view or the initial representation is updated using Implicit Slot Attention and then corrected using a Transformer which combines the updated representation for the current view with the representation of all previous views. The objective of this approach is to prevent our model from forgetting previously seen objects. We choose Implicit Slot

Attention [15] which applies implicit differentiation over the original Slot Attention because it gives better, more stable results and a smooth convergence without the need for gradient clipping. Fig. 2 illustrates the processing of a single view, Algorithm 1 shows the full algorithm.

A. Representation update

Our model follows the Slot Attention architecture. The image x_t is encoded into a matrix of feature vectors $f_t \in \mathbb{R}^{N \times D_{ft}}$ with N vectors and D_{ft} feature dimensionality:

$$f_t = Encoder(x_t) \tag{1}$$

The process uses a CNN encoder, adds positional encodings, and flattens the image features. The viewpoint v_t is encoded into $\tilde{v_t}$ using a Multi-Layer Perceptron (MLP):

$$\tilde{v_t} = MLP(v_t) \tag{2}$$

To take into account the 3D nature of the scene, f_t is concatenated with $\tilde{v_t}$ and transformed by an MLP into $\tilde{f_t}$ of the same dimensioninality as f_t:

$$\tilde{f_t} = MLP(concat(f_t, \tilde{v_t})) \tag{3}$$

From the representation of the previous or initial representation z_{t-1} and $\tilde{f_t}$, Implicit Slot Attention returns an *update* $\tilde{z_t} \in \mathbb{R}^{K \times D_{slots}}$:

$$\tilde{z_t} = ImplicitSlotAttention(z_{t-1}, \tilde{f_t}) \tag{4}$$

B. Representation refinement

The input to the refinement for view t is the concatenation of the initial token, the representations of the previous views, if any, and the *update* $[I, z_1, ..., z_{t-1}, \tilde{z_t}] \in \mathbb{R}^{(t+1) \times D_{slots}}$. The refinement uses a standard Transformer encoder module. In order to use the Transformer on the first view x_1, we employ an initial token $I \in \mathbb{R}^{K \times D_{slots}}$, similar to Vision Transformer [41]. The input is linearly projected to get $[Z_1, ..., Z_{t+1}] \in \mathbb{R}^{(t+1)K \times D_{tr}}$ where D_{tr} is the internal dimension of the Transformer:

$$[Z_1, ..., Z_{t+1}] = Linear([I, z_1, ..., z_{t-1}, \tilde{z_t}]) \tag{5}$$

Similarly to Slotformer, we employ positional encoding at the view level, R_i is the sum of Z_i and P_i, where $P_i \in \mathbb{R}^{K \times D_{tr}}$ is the sinusoidal positional encoding of v_i:

$$[R_1, ..., R_{t+1}] = [Z_1, ..., Z_{t+1}] + [P_1, ..., P_{t+1}] \tag{6}$$

$[R_1, ..., R_{t+1}] \in \mathbb{R}^{(t+1)K \times D_{tr}}$ is the input to the Transformer and $[S_1, ..., S_{t+1}] \in \mathbb{R}^{(t+1)K \times D_{tr}}$ its output:

$$[S_1, ..., S_{t+1}] = Transformer([R_1, ..., R_{t+1}]) \tag{7}$$

The *refined* z_t is a linear transformation of S_{t+1}:

$$z_t = Linear(S_{t+1}) \tag{8}$$

C. Representation decoding

To reconstruct an image x_t from a viewpoint v_t, we find the object representation for the view z_t^{view} by applying an MLP to the concatenation of the scene object representation z and the encoded viewpoint $\tilde{v_t}$:

$$z_t^{view} = MLP(concat(z, \tilde{v_t})) \tag{9}$$

Algorithm 1 ORSA-T algorithm for observation and prediction

Trainable Modules : $Encoder, Decoder, ImplicitSlotAttention, Transformer, LayerNorm, MLP \times 5, Linear \times 2$
Trainable parameters : z_0, I
Input: $\mathcal{T} = \{(x_t, v_t)\}_{t=1}^{T}$ //Images and viewpoints
Output: $\mathcal{L}$ //Loss
$\mathcal{O} = \{(x_o, v_o)\}_{o=1}^{O}$ $\mathcal{Q} = \{(x_q, v_q)\}_{q=1}^{Q}$ //Random split
$\mathcal{L}_o = 0$, $\mathcal{L}_q = 0$ //Loss initialization
//* Iteration through observed images
for $o = 1$ **to** O **do**
  //* Image, viewpoint encoding and projection
  $f_o = Encoder(x_o)$
  $\tilde{v_o} = MLP(v_o)$
  $\tilde{f_o} = MLP(concat(f_o, \tilde{v_o}))$ //projection
  $\tilde{f_o} = LayerNorm(\tilde{f_o})$
  //* Representation update
  $\tilde{z_o} = ImplicitSlotAttention(z_{o-1}, \tilde{f_o})$
  //* Representation refinement
  $[Z_1, ..., Z_{o+1}] = Linear([I, z_1, ..., z_{o-1}, \tilde{z_o}])$
  $[R_1, ..., R_{t+1}] = [Z_1, ..., Z_{o+1}] + [P_1, ..., P_{o+1}]$ //positional encoding
  $[S_1, ..., S_{o+1}] = Transformer([R_1, ..., R_{o+1}])$
  $z_o = Linear(S_{o+1})$
  //* Image reconstruction and loss
  $z_o^{view} = MLP(concat(z_o, \tilde{v_o}))$ //projection
  $\tilde{x_o} = Decoder(z_o^{view})$
  $\mathcal{L}_o += \frac{1}{O} \frac{2(O+1-o)}{O+1} MSE(\tilde{x_o}, x_o)$ //Loss update
end for
//* Iteration through queried images
for $q = 1$ **to** Q **do**
  //* Image reconstruction and loss
  $\tilde{v_q} = MLP(v_q)$
  $z_q^{view} = MLP(concat(z_O, \tilde{v_q}))$ //projection
  $\tilde{x_q} = Decoder(z_q^{view})$
  $\mathcal{L}_q += \frac{1}{Q} MSE(\tilde{x_q}, x_q)$ //Loss update
end for
$\mathcal{L}$=$\mathcal{L}_o$+$\mathcal{L}_q$ //Total loss

As in Slot Attention, we implement a spatial broadcast decoder to get the image reconstruction $\tilde{x_t}$ from z_t^{view}. This decoder broadcasts an individual object representation onto a 2D grid with position embeddings, which is decoded by a CNN into an individual mask and RGB values. The combination of the objects' RGB values multiplied by its mask gives the reconstruction:

$$\tilde{x_t} = Decoder(z_t^{view}) \tag{10}$$

D. Training

For the training, the set $\mathcal{T} = \{(x_t, v_t)\}_{t=1}^{T}$ is split into a subset of observed views $\mathcal{O} = \{(x_o, v_o)\}_{o=1}^{O}$ and a subset of queried views $\mathcal{Q} = \{(x_q, v_q)\}_{q=1}^{Q}$. O is drawn randomly, $Q = T \setminus O$ and $\mathcal{T} = \mathcal{O} \cup \mathcal{Q}$.
The model will learn the scene representation by iterating through the observed views and generate the queried views which is the test of the quality of the scene representations.

The training loss $\mathcal{L}$ is the sum of a loss on the observed views $\mathcal{L}_o$ and a loss on the queried views $\mathcal{L}_q$.

$$\mathcal{L} = \mathcal{L}_o + \mathcal{L}_q \tag{11}$$

The observed view loss is a weighted average MSE of reconstruction $\tilde{x_o}$ with respect to the true x_o. In contrast to MulMON, the average is weighted to favor earlier views, as we have noticed that it gives better results during training:

$$\tilde{x_o} = Decoder(z_o^{view}), \quad z_o^{view} = MLP(concat(z_o, \tilde{v_o})) \tag{12}$$

$$\mathcal{L}_o = \frac{1}{O}\sum_{o=1}^{O} \frac{2(O+1-o)}{O+1} MSE(\tilde{x_o}, x_o) \tag{13}$$

The queried view loss is a weighted average MSE between x_q and its reconstruction $\tilde{x_q}$ using the last representation of the learning process z_O.

$$\tilde{x_q} = Decoder(z_q^{view}), \quad z_q^{view} = MLP(concat(z_O, \tilde{v_q})) \tag{14}$$

$$\mathcal{L}_q = \frac{1}{Q}\sum_{q=1}^{Q} MSE(\tilde{x_q}, x_q) \tag{15}$$

IV. Experiments

The goal of our experiments is to evaluate ORSA-T and study the impact of the aggregation and refinement with Transformer in ORSA-T. To assess the influence of the Transformer, we run an ablation study on ORSA. To obtain ORSA, we remove from ORSA-T all the representation refinement processes described in Section III-B. ORSA has a structure similar to MulMON, where IODINE is replaced by Implicit Slot Attention. We also run MulMON to compare its results to those of ORSA-T and ORSA.

A. Dataset creation

We created two datasets, Partial-View and Full-View, to conduct the experiments using MuJoCo [42], which was already used to create datasets for GQN or MONet. The scene consists of a room containing 3 to 4 simple objects (cube, sphere, cylinder) of different colors, similar to CLEVR [43] and CLEVR-MultiView [24]. For both datasets, each scene is rendered from 10 random viewpoints with a resolution of 64x64 pixels, and each image is annotated by its viewpoint. The resolution is the same as CLEVR-MultiView which make results comparable; a higher resolution would have required substantial additional computing resources. Partial-View consists of partial views of the room, which are images covering only part of the scene, where some objects are not in the field of view. Each image contains at least one object. Partial-View has 5000 scenes for training and 500 for testing. Full-View consists of views where all the objects visible. Full-View has 25000 scenes for training and 500 for testing. It contains more scenes than Partial-View due to our models exhibiting extremely poor results in early experimentation with a dataset of similar size to Partial-View. See Fig. 3a,b for examples of both datasets.

B. Experimental setup

a) Baseline: We compare our results with MulMON, which is the only previous object-centric scene representation learning approach utilizing images annotated by viewpoints and improving the representation through iterations across different views.

b) Datasets: In addition to Partial-View, Full-View, as described above, we use CLEVR-MultiView. Full-View is supposedly easier to learn than Partial-View because all objects are present in all images. CLEVR-MultiView is used to evaluate our model on a dataset designed for testing MulMON. It is similar to Full-View in terms of resolution and scene composition. It contains 1500 training scenes and 200 testing scenes. See Fig. 3c for an example of CLEVR-MultiView scene. Fig. 3 illustrates the differences between the datasets we designed and CLEVR-MultiView. Our datasets exhibit wider variations in luminosity, resulting in a less uniform background ranging from black to white and less uniform colors of objects. Additionally, our datasets include more pronounced shadows on objects, which may affect scene segmentation as shadows are not part of the ground truth masks.

c) Training: We train all the models with the number of object representations $K = 7$ (maximum of 6 objects plus background). For all experiments, except MulMON with the CLEVR-MultiView dataset, the dimension of the representation D_{slots} is set to 64 (consistent with Slot Attention and IODINE), and the dimension of the encoded viewpoint $\tilde{v_t}$ is set to 32. For the experiment with MulMON using the CLEVR-MultiView dataset, we use dimensions 16 and 3, respectively, as described in [24]. During training, the number of observed views O is randomly selected between 3 and 7; for testing, the number of observed views is set to 5. We use a higher random number of observed views than MulMON (between 1 and 6) to allow the model to learn from partial views. We train ORSA-T and ORSA for 300,000 gradient steps using the Adam optimizer [44] with a learning rate of 3×10^{-4}, gradient norm clipping set to 5.0, and a batch size of 32. We train MulMON for 200,000 gradient steps using the Adam optimizer [44] with a learning rate of 3×10^{-4} and a batch size of 8. Optimizer, learning rate and its schedule remain unchanged from [24]). We have set up the experiments to compare the models with similar time budgets and memory usage. Each experiment is run with five different random seeds (42, 43, 44, 45, 46).

d) Metrics: We evaluate the models by assessing the quality of the reconstruction and prediction, as well as the quality of the object representation. Reconstruction quality is evaluated using the pixel Mean Squared Error (MSE) on the observed views, while prediction quality is assessed using the queried views. The quality of the object representation or scene factorization is evaluated using mean intersection-over-union (mIOU) on observed views and on queried views. We utilize the MulMON implementation, which employs a Hungarian-style matching algorithm [45] for each image to match ground-truth object masks with the model's object masks used for image reconstruction. The intersection-over-union (IOU) is then computed using the matched masks and averaged. We report the mean and the standard deviation of every metric.

(a) Partial-View **(b) Full-View** **(c) CLEVR-MultiView**

Fig. 3. Example of 3 views from each of the dataset employed. (a) Partial-View, (b) Full-View and (c) CLEVR-MultiView

TABLE I
SUMMARY OF MSE $\times 10^4$ (LOWER IS BETTER, MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR PARTIAL-VIEW DATASET.

MODEL	OBS MSE	QUE MSE
ORSA-T	7.4 ± 1.1	29.0 ± 1.3
ORSA	9.4 ± 0.8	88.8 ± 1.8
MULMON	95.4 ± 5.8	173.4 ± 41.0

TABLE II
SUMMARY OF MIOU (HIGHER IS BETTER, MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR PARTIAL-VIEW DATASET.

MODEL	OBS MIOU	QUE MIOU
ORSA-T	0.87 ± 0.01	0.82 ± 0.01
ORSA	0.79 ± 0.03	0.57 ± 0.02
MULMON	0.45 ± 0.12	0.37 ± 0.10

C. Results on Partial-View dataset

In this subsection, we analyze the results from ORSA-T, ORSA and MulMON on Partial-View. Table I presents the reconstruction and prediction error and Table II the segmentation performance. See Fig. 4 for examples of the outputs of all models.

a) ORSA-T: provides the best results, with the lowest MSE and highest mIOU. It also exhibits the lowest standard deviation (except for the MSE for observed views with ORSA), indicating consistent results across different seeds. The high mIOU, superior to MulMON with CLEVR-Multiview in [24], indicates that ORSA-T effectively factorizes the scene into objects (Fig. 4a).

b) ORSA: exhibits slightly inferior metrics compared to ORSA-T for the observed views, indicating a reduced capability to reconstruct and segment these views. The absence of a Transformer prevents it from learning scene object representation, as shown for the queried views by a threefold increase in MSE and a drop in mIOU from 0.82 to 0.57. The causes of the poor metrics include missing objects, poorly defined object shapes, and object masks covering both objects and background (Fig. 4b).

c) MulMON: fails to learn an effective representation; the MSE for observed views is 13 times larger, and the MSE for queried views is 6 times worse. MulMON reconstructions show blurred objects (Fig. 4c). The metrics also exhibit a large standard deviation, indicating variable performance and the model's inability to handle this type of data reliably.

TABLE III
SUMMARY OF MSE $\times 10^4$ (LOWER IS BETTER, MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR FULL-VIEW DATASET.

MODEL	OBS MSE	QUE MSE
ORSA-T	7.5 ± 0.9	10.3 ± 1.4
ORSA	10.7 ± 3.4	25.5 ± 7.8
MULMON	128.1 ± 29.1	216.4 ± 12.1

TABLE IV
SUMMARY OF MIOU (HIGHER IS BETTER, MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR FULL-VIEW DATASET.

MODEL	OBS MIOU	QUE MIOU
ORSA-T	0.71 ± 0.12	0.71 ± 0.13
ORSA	0.63 ± 0.12	0.58 ± 0.15
MULMON	0.44 ± 0.17	0.34 ± 0.10

D. Results on Full-View dataset

In this subsection, we analyze the impact of a supposedly easier dataset where all the objects are always seen. Table III summarizes the reconstruction and prediction performance, and Table IV the segmentation performance.

a) ORSA-T: delivers the best results. The mean reconstruction MSE is similar to that of the Partial-View dataset, and the mean prediction MSE is three times lower, with both showing low standard deviations. Both mIOUs are equal to 0.71, more than 0.1 lower than the previous dataset, with a high standard deviation (0.12/ 0.13), indicating a significant difference in performance Table V, showing the metrics for all seeds, reveals that the mIOU range is between 0.86 (good segmentation) and 0.53 (poor segmentation). The best seed (Fig. 5a) offers accurate reconstruction, prediction and segmentation. The worst seed (Fig. 6a) offers similar reconstruction and prediction but the segmentation is notably poorer caused by the background.

b) ORSA: exhibits worse performance than ORSA-T: reconstruction MSE is slightly higher, prediction MSE is 2.5 times higher, and it shows significantly higher standard deviation. Both mIOU values are around 0.1 lower compared to ORSA-T, and like ORSA-T, they are characterized by the same high standard deviation. Table VI illustrates the wide range of mIOU for different seeds, ranging from 0.81 to 0.48. Similar to ORSA-T, the comparison between Fig. 5b and Fig. 6b shows the differences in segmentation between seeds.

c) MulMON: shows similar results than with the Partial-View. It is unable to reconstruct, predict or segment. MSE losses are 20 times higher than ORSA-T. See Fig. 5c or Fig. 6c for an illustration of the poor performance.

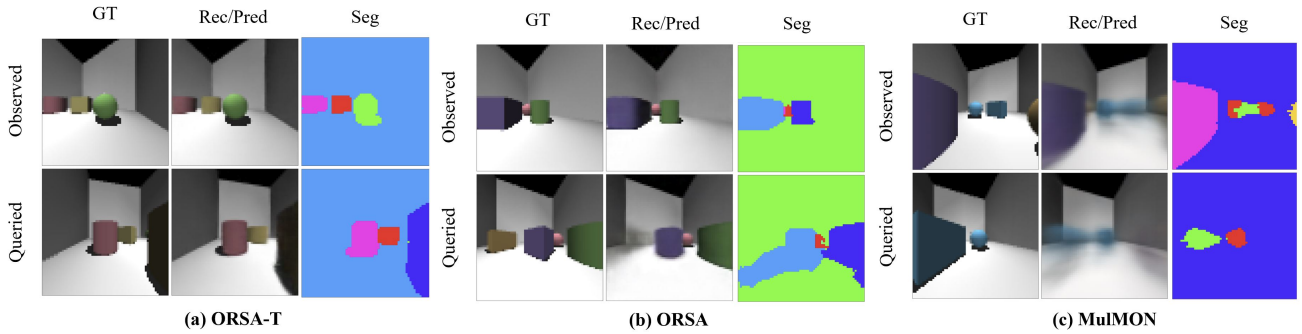

Fig. 4. Example of the performance of the best model of (a) ORSA-T, (b) ORSA and (c) MulMON for Partial-View dataset. ORSA-T has high-quality reconstruction, prediction, and segmentation. ORSA shows similar performance to ORSA-T on observed views but exhibits missing objects and poor segmentation on queried views. MulMON presents blurred reconstructions of objects

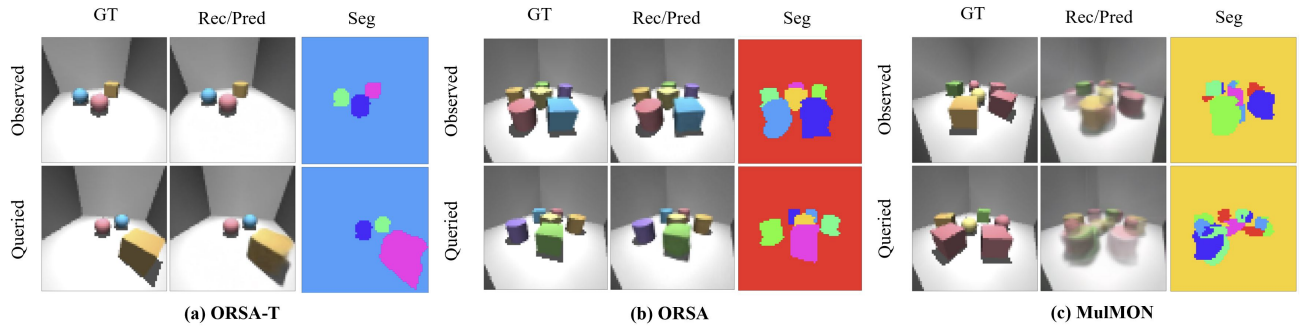

Fig. 5. Example of the performance of the best model of (a) ORSA-T, (b) ORSA and (c) MulMON for Full-View dataset. ORSA-T and ORSA demonstrate similar abilities in terms of reconstruction, prediction, and segmentation. However, ORSA-T exhibits slightly better quantitative performance when comparing Table V and Table VI. MulMON struggles with object reconstruction, prediction and segmentation.

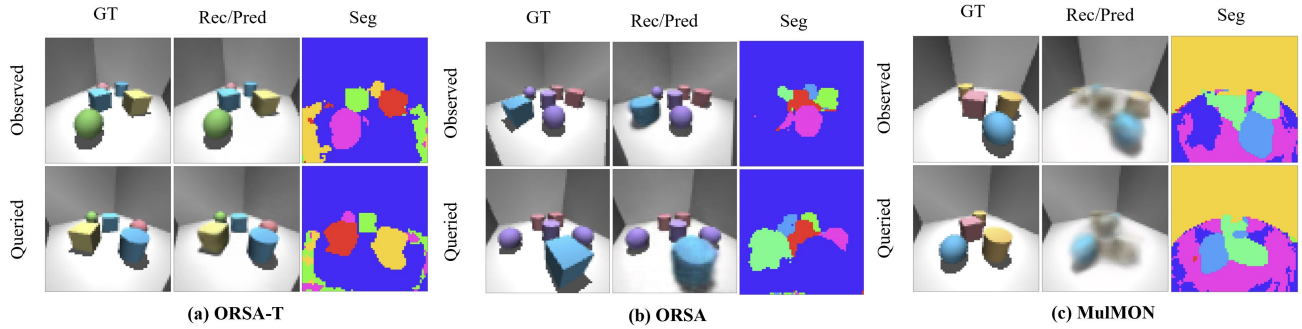

Fig. 6. Example of the performance of the worst model of (a) ORSA-T, (b) ORSA and (c) MulMON for Full-View dataset. ORSA-T and ORSA reconstruct and predict views, but have poor segmentation of objects and background. MulMON struggles with object reconstruction, prediction and segmentation

TABLE V
MSE $\times 10^4$ AND MIOU FOR OBSERVED AND QUERIED VIEWS USING ORSA-T FOR ALL SEEDS FOR FULL-VIEW DATASET.

SEED	OBS MSE	QUE MSE	OBS MIOU	QUE MIOU
42	8.8	12.7	0.54	0.52
43	7.2	9.5	0.83	0.84
44	7.3	8.8	0.68	0.70
45	6.1	9.6	0.87	0.87
46	7.9	11.1	0.64	0.63

TABLE VI
MSE $\times 10^4$ AND MIOU FOR OBSERVED AND QUERIED VIEWS USING ORSA FOR ALL SEEDS FOR FULL-VIEW DATASET.

SEED	OBS MSE	QUE MSE	OBS MIOU	QUE MIOU
42	15.3	34.8	0.51	0.48
43	8.1	20.9	0.73	0.69
44	12.2	29.0	0.49	0.40
45	5.7	12.7	0.81	0.81
46	11.9	30.1	0.61	0.52

E. Results on CLEVR-Multiview dataset

In this subsection, we test our models on one of the MulMON datasets. Table VII summarizes the reconstruction and prediction performance, while Table VIII presents the segmentation performance. As MulMON does not converge on 2 seeds, we also report the results excluding these seeds and label them as 'MulMON converging' in the tables. See Fig. 7 for an illustration of the performance of all models.

a) ORSA-T: exhibits very low standard deviation metrics, with the best prediction MSE and slightly worse reconstruction MSE than ORSA. Although the MSEs are similar to those of

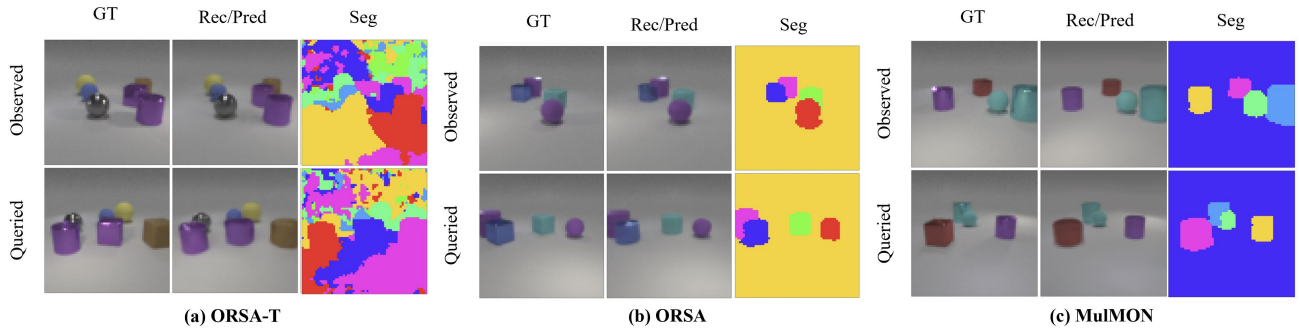

Fig. 7. Example of the performance of the best models for (a) ORSA-T, (b) ORSA, and (c) MulMON on the CLEVR-Multiview dataset. All models are capable of reconstructing and predicting views. ORSA and MulMON can also segment views. However, despite achieving good performance in reconstruction and segmentation compared to other models (Table VII), ORSA-T fails in segmentation

TABLE VII
SUMMARY OF MSE $\times 10^4$ (LOWER IS BETTER, MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR CLEVR-MULTIVIEW DATASET.

MODEL	OBS MSE	QUE MSE
ORSA-T	7.4 ± 0.2	7.6 ± 0.2
ORSA	7.2 ± 0.3	10.8 ± 0.4
MULMON	16.4 ± 7.9	16.5 ± 9.3
MULMON CONVERGING	10.0 ± 0.6	9.0 ± 0.6

TABLE VIII
SUMMARY OF MIOU (MEAN $\pm$ STDDEV ACROSS SEEDS) FOR OBSERVED AND QUERIED VIEWS FOR CLEVR-MULTIVIEW DATASET.

MODEL	OBS MIOU	QUE MIOU
ORSA-T	0.23 ± 0.01	0.22 ± 0.01
ORSA	0.73 ± 0.04	0.71 ± 0.04
MULMON	0.58 ± 0.16	0.60 ± 0.17
MULMON CONVERGING	0.73 ± 0.004	0.77 ± 0.003

other datasets, ORSA-T has difficulty segmenting these images (Fig. 7a), with mIOU just above 0.20.

b) ORSA: has slightly higher standard deviation metrics than ORSA-T. It has the best prediction MSE and a reconstruction MSE slightly worse than ORSA-T. But contrary to ORSA-T, it factorizes the scene more effectively (Fig. 7b), mIOU ranges between 0.71 and 0.73.

c) MulMON: has a very high standard deviation because with 2 of the 5 seeds the model did not converge. With all seeds, the MSEs are twice as high as our models, while the mIOUs are more than 0.1 lower than ORSA. Excluding non-converging models, the MulMON MSEs are around 20% higher than the best results, still indicating good reconstruction (Fig. 7c), and MulMON achieves the best mIOU of 0.73 and 0.77, respectively, compared to 0.73 and 0.71 for ORSA. MulMON achieves these metrics with an object representation dimension of 16 instead of 64 for our models. Note that we trained MulMON with a number of observed views randomly chosen between 3 and 7 instead of 1 to 6, as described in [24].

F. Summary and discussion

When comparing the metrics of ORSA-T and ORSA, the addition of aggregation and refinement with Transformer improves image reconstruction and prediction in nearly all experiments, and enhances the image segmentation. However, ORSA-T is unable to segment images for CLEVR-MultiView. One possible explanation is the small size of the dataset (1500 scenes compared to 25000 for Full-View). This is suggested by ORSA-T and ORSA having MSEs on the training dataset that are less than half of those on the testing datasets, so that some form of overfitting may impact the segmentation.

ORSA-T and ORSA exhibit extreme volatility in results for different seeds with Full-View. Full-View contains less diverse images than Partial, as the camera is moving around the objects. This lack of diversity may hinder the learning of the concept of object. As each view contains all objects, a direction to explore is to reduce the range of observed views for the training to facilitate the backpropagation of losses (We use 3-7, Mulmon 1-6). The small difference in performance between the best-performing ORSA-T (Table V) and ORSA (Table VI) shows that the use of Transformer has less impact than with Partial-View.MulMON seems to struggle with our datasets, possibly due to the differences explained in Section IV-B, such as less uniform colors resulting from variations in luminosity.Its inability to perform well on Full-View, which is based on the CLEVR-MultiView dataset, only emphasizes these differences. Due to the unordered nature of Slot Attention representations, ORSA-T and ORSA cannot effectively track objects during learning for Partial-View and Full-View. A single representation is not consistently associated with a specific object. For example, the first representation could represent a different object for each observed view. Surprisingly, ORSA tracks objects nearly perfectly in CLEVR-MultiView , which further highlights the disparities between CLEVR-MultiView and our datasets.

V. CONCLUSION AND FUTURE WORK

We address the problem of learning object-centric scene representations from multiple views by improving the representation for each new viewpoint using images with partial views of a scene. We introduce ORSA-T, an algorithm for unsupervised scene object representation learning combining Slot-Attention and Transformer. We have created two new datasets to validate our model. Our algorithm was tested on multiple datasets and compared with ORSA (an ablation of ORSA-T without aggregation and refinement by Transformer) and MulMON. ORSA-T's ability to solve scene factorization

using a partial-view dataset has been demonstrated. We have also discovered limitations of our model in segmentation when used with the Full-View dataset, primarily a large standard deviation of the results for different seeds, mostly due to poor segmentation of the background.

Future work should focus on reducing the performance variation across different seeds and on addressing the segmentation problem of the background encountered with CLEVR-MultiView. Ultimately, we aim to integrate a pre-trained ORSA-T model with a reinforcement learning algorithm to attempt to solve simple tasks using only learned representations, for which we have originally designed the Partial-View dataset.

REFERENCES

[1] D. Kahneman, A. Treisman, and B. J. Gibbs, "The reviewing of object files: Object-specific integration of information," *Cognitive Psychology*, vol. 24, no. 2, pp. 175–219, 1992.

[2] E. S. Spelke, S. A. Lee, and V. Izard, "Beyond core knowledge: Natural geometry," *Cogn. Sci.*, vol. 34, no. 5, pp. 863–884, 2010.

[3] S. P. Johnson, "How infants learn about the visual world," *Cogn. Sci.*, vol. 34, no. 7, pp. 1158–1184, 2010.

[4] J. Yuan, T. Chen, B. Li, and X. Xue, "Compositional scene representation learning via reconstruction: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 11 540–11 560, 2023.

[5] J. Tenenbaum, "Building machines that learn and think like people," in *AAMAS 2018*, p. 5.

[6] K. Greff, S. van Steenkiste, and J. Schmidhuber, "On the binding problem in artificial neural networks," *CoRR*, vol. abs/2012.05208, 2020.

[7] Z. Chen, J. Mao, J. Wu, K. K. Wong, J. B. Tenenbaum, and C. Gan, "Grounding physical concepts of objects and events through dynamic visual reasoning," in *ICLR 2021*.

[8] N. Watters, L. Matthey, M. Bosnjak, C. P. Burgess, and A. Lerchner, "COBRA: data-efficient model-based RL through unsupervised object discovery and curiosity-driven exploration," *CoRR*, vol. abs/1905.09275, 2019. [Online]. Available: http://arxiv.org/abs/1905.09275

[9] R. Keramati, J. Whang, P. Cho, and E. Brunskill, "Strategic object oriented reinforcement learning," *CoRR*, vol. abs/1806.00175, 2018.

[10] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. M. Botvinick, and A. Lerchner, "Monet: Unsupervised scene decomposition and representation," *CoRR*, vol. abs/1901.11390, 2019.

[11] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. M. Botvinick, and A. Lerchner, "Multi-object representation learning with iterative variational inference," in *ICML 2019*, vol. 97, pp. 2424–2433.

[12] Z. Lin, Y. 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 *ICLR 2020*.

[13] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," in *NeurIPS 2020*.

[14] O. Biza, S. van Steenkiste, M. S. M. Sajjadi, G. F. Elsayed, A. Mahendran, and T. Kipf, "Invariant slot attention: Object discovery with slot-centric reference frames," in *ICML 2023*, vol. 202, pp. 2507–2527.

[15] M. Chang, T. Griffiths, and S. Levine, "Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation," in *NeurIPS 2022*, vol. 35, pp. 32 694–32 708.

[16] S. M. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. C. Rabinowitz, H. King, C. Hillier, M. M. Botvinick, D. Wierstra, K. Kavukcuoglu, and D. Hassabis, "Neural scene representation and rendering," *Science*, vol. 360, pp. 1204 – 1210, 2018.

[17] V. Sitzmann, M. Zollhoefer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *NeurIPS 2019*, vol. 32.

[18] 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 *ICML 2021*, vol. 139, pp. 5742–5752.

[19] K. Stelzner, K. Kersting, and A. R. Kosiorek, "Decomposing 3d scenes into objects via unsupervised volume segmentation," *CoRR*, vol. abs/2104.01148, 2021.

[20] 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," in *NeurIPS 2021*, vol. 34, pp. 20 146–20 159.

[21] J. Yuan, B. Li, and X. Xue, "Unsupervised learning of compositional scene representations from multiple unspecified viewpoints," in *AAAI 2022*, pp. 8971–8979.

[22] C. Chen, F. Deng, and S. Ahn, "Learning to infer 3d object models from images," *CoRR*, vol. abs/2006.06130, 2020.

[23] M. S. M. Sajjadi, D. Duckworth, A. Mahendran, S. van Steenkiste, F. Pavetic, M. Lucic, L. J. Guibas, K. Greff, and T. Kipf, "Object scene representation transformer," in *NeurIPS 2022*, vol. 35, pp. 9512–9524.

[24] N. Li, C. Eastwood, and R. Fisher, "Learning object-centric representations of multi-object scenes from multiple views," in *NeurIPS 2020*, vol. 33, pp. 5656–5666.

[25] T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, and K. Greff, "Conditional object-centric learning from video," in *ICLR 2022*.

[26] G. Singh, Y.-F. Wu, and S. Ahn, "Simple unsupervised object-centric learning for complex and naturalistic videos," in *NeurIPS 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, pp. 18 181–18 196.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS 2017*, vol. 30.

[28] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, k. kavukcuoglu, and G. E. Hinton, "Attend, infer, repeat: Fast scene understanding with generative models," in *NeurIPS 2016*, vol. 29.

[29] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner, "GENESIS: generative scene inference and sampling with object-centric latent representations," in *ICLR 2020*.

[30] E. Crawford and J. Pineau, "Spatially invariant unsupervised object detection with convolutional neural networks," in *EAAI 2019*, pp. 3412–3420.

[31] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhöfer, "Deepvoxels: Learning persistent 3d feature embeddings," in *CVPR 2019*, pp. 2437–2446.

[32] 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 *ECCV 2020*, vol. 12346, pp. 405–421.

[33] A. Kosiorek, H. Kim, Y. W. Teh, and I. Posner, "Sequential attend, infer, repeat: Generative modelling of moving objects," in *NeurIPS 2018*, vol. 31.

[34] M. A. Weis, K. Chitta, Y. Sharma, W. Brendel, M. Bethge, A. Geiger, and A. S. Ecker, "Benchmarking unsupervised object representations for video sequences," *J. Mach. Learn. Res.*, vol. 22, pp. 183:1–183:61, 2021.

[35] G. Elsayed, A. Mahendran, S. van Steenkiste, K. Greff, M. C. Mozer, and T. Kipf, "Savi++: Towards end-to-end object-centric learning from real-world videos," in *NeurIPS 2022*, vol. 35, pp. 28 940–28 954.

[36] Z. Wu, N. Dvornik, K. Greff, T. Kipf, and A. Garg, "Slotformer: Unsupervised visual dynamics simulation with object-centric models," in *ICLR 2023*.

[37] H. Yu, L. J. Guibas, and J. Wu, "Unsupervised discovery of object radiance fields," in *ICLR 2022*.

[38] C. Smith, H.-X. Yu, S. Zakharov, F. Durand, J. B. Tenenbaum, J. Wu, and V. Sitzmann, "Unsupervised discovery and composition of object light fields," *Transactions on Machine Learning Research*, 2023.

[39] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*, vol. 1681, 1999, p. 319.

[40] B. Jia, Y. Liu, and S. Huang, "Improving object-centric learning with query optimization," in *ICLR 2023*.

[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR 2021*.

[42] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ*, pp. 5026–5033.

[43] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *CVPR 2017*, pp. 1988–1997.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR 2015*.

[45] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.