RDIR: Capturing Temporally-Invariant Representations of Multiple Objects in Videos

Piotr Zieliński, Tomasz Kajdanowicz
Department of Artificial Intelligence
Wrocław University of Science and Technology
p.zielinski@pwr.edu.pl

Abstract

Learning temporally coherent representations of multiple objects in videos is crucial for understanding their complex dynamics and interactions over time. In this paper, we present a deep generative neural network, which can learn such representations by leveraging pretraining. Our model builds upon a scale-invariant structured autoencoder, extending it with a convolutional recurrent module to refine the learned representations through time and enable information sharing among multiple cells in multi-scale grids. This novel approach provides a framework for learning per-object representations from a pretrained object detection model, offering the ability to infer predefined types of objects, without the need for supervision. Through a series of experiments on benchmark datasets and real-life video footage, we demonstrate the spatial and temporal coherence of the learned representations, showcasing their applicability in downstream tasks such as object tracking. We analyze the method's robustness by conducting an ablation study, and we compare it to other methods, highlighting the importance of the quality of objects' representations.

1. Introduction

Human's ability to perceive and understand the visual world allows us to comprehend the compositionality of the scene captured by our eyesight. This cognitive process is based not only on observing the surroundings at a given moment but also on comprehending the temporal variance of the scene, and how the objects move and interact with each other, enabling a deep understanding of visual scenes. The complexity of this natural process is a topic of vivid research [12]. Recent machine learning and computer vision methods aim at learning similar comprehension as a result of supervised learning for particular tasks, such as object detection, instance segmentation, visual question answering, etc. A group of methods allowing for a more general understanding of scenes is often referred to as multi-object representation learning models.

Downstream models trained on object-centric representations are usually easier to train; this approach can also reduce the amount of data required to achieve good performance. However, the success of these algorithms relies heavily on the quality of embeddings produced by the representation learning model [2]. Recent methods, building upon previous developments in this area, extend the image-based approach to videos and infer temporal changes of objects in scenes: their movement, variation of shape, etc. [7, 18, 20, 23, 31]. This makes it possible to capture and understand the underlying dynamics of complex scenes as they change through time.

Recently, the research in the area of multi-object representation learning has been shifting from unsupervised models towards semi- or self-supervised approaches [9, 22, 32]. By incorporating additional knowledge these approaches provide more robust representations and can attend to individual objects in complex scenes more easily. Unfortunately, many of these methods utilize very computationally expensive processing, taking multiple days of training on high-performance GPUs to reach success. Furthermore, the quality of the representations and their temporal and spatial stability have been given insufficient attention, as most models are compared by the quality of scene decomposition (i.e. the accuracy of segmentation masks).

In this paper, we address the challenge of capturing object-centric, temporally, and spatially stable representations in videos. We introduce RDIR, a novel method for multi-object representation learning on videos, which utilizes a recurrent mechanism to provide temporally consistent object representations. It follows the recent shift towards semi- and self-supervised learning, extending a pretrained single-shot multi-scale object detection model with a recurrent mechanism for encoding each object representation without further supervision. By applying a pretrained object detection model, RDIR enables a deeper understanding of detected objects, which can be obtained on

any unannotated dataset, with better scalability and shorter training. Through a series of experiments, we compare it with other multi-object representation learning models for videos, proving the ability to capture stable object representations and showcasing its usability in downstream tasks.

The contributions of the paper are as follows. We introduce a model for learning object-centric representations on videos and explain the theoretical underpinnings. We present a comparison of the performance and the quality of the representations of this model and other state-of-the-art methods for multi-object representation learning. We provide a novel experimental approach for evaluating the temporal and spatial coherence of representations and show how dataset characteristics and model architecture influence the performance of the approach.

2. Related Works

The presented research is done in the multi-object visual representation learning domain, aiming at learning representations of multiple objects visible in a scene.

Multi-object representation learning on images is usually conducted utilizing unsupervised, VAE-based [21, 30] models. The approaches can be categorized into two groups: spatial-attention models and scene-mixture models. Spatial-attention models leverage geometrical figures (usually rectangles) for attending to individual objects, allowing for faster processing and highly interpretable inference. These models can work by iteratively predicting subsequent objects (as in AIR [1]) or predicting all objects with a single-shot model (SPAIR [6] and SPACE [26]). However, due to strong inductive bias, these models cannot attend to objects of irregular shapes and struggle with large objects, often splitting them into meaningless parts. Scene-mixture models such as MONet [4], IODINE [15], Slot Attention [28] or GENESIS [10, 11], do not restrict the shape of masks used to split the scene into parts, enabling the model to infer on more complex scenes. However, these methods are expensive to train and infer, as they attend to the image recurrently (producing one mask at a time or refining representations iteratively) and tend to encapsulate multiple objects in one mask on complex datasets, unable to discover meaningful entities.

RDIR builds on the approach suggested in SSDIR [32], where a spatial-attention model is extended with a multi-scale convolutional encoder and a non-restricted attention box size, allowing the model to attend to objects of varying sizes in one forward pass while preserving lower computational expense comparing to scene-mixture models.

Multi-object representation learning on videos has been tackled by extending image-based approaches for modeling sequential data. A common paradigm involves applying a recurrent LSTM [16] or GRU [5] cell as a module for handling sequences of images. This approach has been used for both spatial-attention models (recurrent [23] and single-shot [7, 18]) and scene-mixture models [8, 31], achieving good results on simple datasets. The results show however that without any supervision, these methods are not able to scale to real-world datasets, struggling with issues similar to their image-based counterparts. What is more, researchers focus on the reconstruction quality and generative capabilities of these models, rarely reviewing or comparing the quality of the representations produced by the model. Another issue is related to the computational expense, increased additionally by the application of recurrent cells.

A unique approach was proposed in SIMONe [20], where a transformer network is used to explicitly factorize latent representations into temporal and per-object latents. By inferring the entire sequence at once, the model can extract split representations referring to how the frame changes over time, and encapsulate each object's appearance in its per-sequence representation. However, this approach cannot scale to real-world datasets due to the fully unsupervised training setup, and it is more computationally complex with the application of a strong feature extractor and a transformer network.

RDIR applies a recurrent mechanism similar to discovery-propagation cells applied for spatial-attention models, but here we do not explicitly track and associate new objects discovered in the subsequent frames. Instead, we apply the recurrent cell in the encoder feature maps, aiming at features refining with the use of hidden state passed through the sequence and emergent representations' association. See Section 3 for more details.

Supervision in multi-object representation learning models is suggested as the key factor for scaling the current methods to real-world datasets. In the Slot Attention for Videos model [22] authors leverage a self-supervised model for training using the optical flow conditioning. Another extension to Slot Attention [9] proposes conditioning the model on depth maps as the target instead of the RGB input frame. Researchers show that these advances enable the model to scale from benchmark datasets to real-world data, allowing it to understand complex scenes better. The analysis of these models' performance was focused on the quality of instance segmentation in the emergent masks, without reviewing the quality of representations. Furthermore, the added supervision requires providing a dataset with depth maps or estimating the optical flow for training.

In [32] authors postulate a semi-supervised approach. Here, the model consists of a single-shot object detector, trained with supervision, which provides spatial attention locations. Authors extend the model with representation en-

coders and fine-tune the model without supervision, showing the improved quality of the latent space and the ability to enforce focus on a selected set of object classes in pictures. This approach requires obtaining a smaller dataset for training an object detection model, which can then be extended and trained on a larger dataset without supervision for learning objects' representations.

RDIR is an extension to the semi-supervised approach, utilizing a staged training protocol for optimal computational expense and fast training. Provided an accurate object detector, it can attend accurately to given object classes and learn their representations without further supervision.

3. The Method

In this section, we describe **Recurrent Detect, Infer, Repeat** (RDIR). It is a structure autoencoder, extending the recent semi-supervised approach for multi-object representation learning with a recurrent encoder network, allowing it to scale to learning representations on videos.

3.1. Background: SSDIR [32]

SSDIR (Single-Shot Detect, Infer, Repeat) is a neural network based on variational autoencoder architecture for multi-object representation learning on images. It extends the well-known single-shot object detection approach and integrates knowledge learned in this task for capturing structured representations of images. Each object in the scene is represented by four latent variables: $\boldsymbol{z}_{where}$ defining the object's position and size, $\boldsymbol{z}_{present}$ - a binary variable indicating the object's presence, $\boldsymbol{z}_{what}$ describing the object's appearance and $\boldsymbol{z}_{depth}$ determining the object's relative depth in the scene.

The encoder uses a pretrained SSD [27] object detector to estimate objects' positions and presence, and re-uses feature maps from a pretrained backbone to produce appearance and depth representations. Then, the representations are forwarded to the decoder and filtered according to the presence variable; each present object's appearance latent is processed by a convolutional decoder, producing per-object images. These images are translated and scaled according to the tight bounding box location [17]. The resulting images are merged using a weighted sum, with $\boldsymbol{z}_{depth}$ as weights.

SSDIR applies the semi-supervised approach to learning representations. The object detection model is trained with supervision on an annotated dataset, and then its weights are transferred to the SSDIR model and re-used as a base for learning the detected objects' representations. This way SSDIR can be enforced to attend to a specified subset of objects visible in the scene.

Thanks to multi-scale grid-based inference and unrestricted objects' sizes, SSDIR attends to objects of varying sizes and positions, and produces scale-invariant representations in a single processing of an image, without the need of extracting glimpses, as previous spatial-attention models did [6, 26]. However, it uses a simple convolutional backbone (VGG11) and cannot scale to larger images. The approach suffers from the problems of other single-shot object detectors, which struggle with densely packed objects, and often produce multiple predictions of the same object due to detecting it on several levels of multi-scale feature maps.

3.2. RDIR

In RDIR we follow the idea of extending a single-shot object detection model with encoding heads for representation learning and then training the model as an autoencoder. The model encodes each image in a sequence into a structured representation, referring to each detected object in the scene, and describing its appearance, location, presence, and depth in the scene. Then, the representations are filtered and processed by the decoder, producing per-object images, which are merged to create the image reconstruction. In RDIR we attempt to model two functions: the encoder $\boldsymbol{P}(\boldsymbol{Z}|\boldsymbol{X})$, mapping the input image to the latent representation, and the decoder $\boldsymbol{P}(\boldsymbol{X}|\boldsymbol{Z})$, reconstructing the input image based on the latent representation.

Latent space is structured to represent each object visible in the scene. By leveraging the single-shot object detection model and the multi-scale feature maps approach, RDIR produces latent representations for multiple grids, referring to parts of the input image. These latent representations consist of four variables:

1. $\boldsymbol{z}_{where} = [\boldsymbol{cx}, \boldsymbol{cy}, \boldsymbol{w}, \boldsymbol{h}] \in \mathbb{R}^4$ - describing each object's position and size (as bounding box center coordinates $\boldsymbol{cx}$, $\boldsymbol{cy}$, width $\boldsymbol{w}$ and height $\boldsymbol{h}$),
2. $\boldsymbol{z}_{present} = \max \boldsymbol{c} \in [0, 1]$ - used to determine if the given cell detected any object (inferred based on the maximum of object's class confidences $\boldsymbol{c}$),
3. $\boldsymbol{z}_{what} \in \mathbb{R}^D$ - D-sized object appearance vector used for reconstructing per-object image,
4. $\boldsymbol{z}_{depth} \in \mathbb{R}$ - relative object depth, used for ordering object reconstructions in the decoder.

Following the semi-supervised approach, RDIR uses a pretrained object detection model to provide $\boldsymbol{z}_{where}$ and $\boldsymbol{z}_{present}$ latents, simplifying the process of object discovery. The remaining latent variables are inferred by the model, trained in the representation learning setup.

The encoder (shown in Fig. 1) is designed to enable capturing spatially and temporally consistent latent representations. It consists of a convolutional backbone based on the architecture proposed in YOLOv4 [3] (CSPDarknet53), which provides significantly improved capability compared

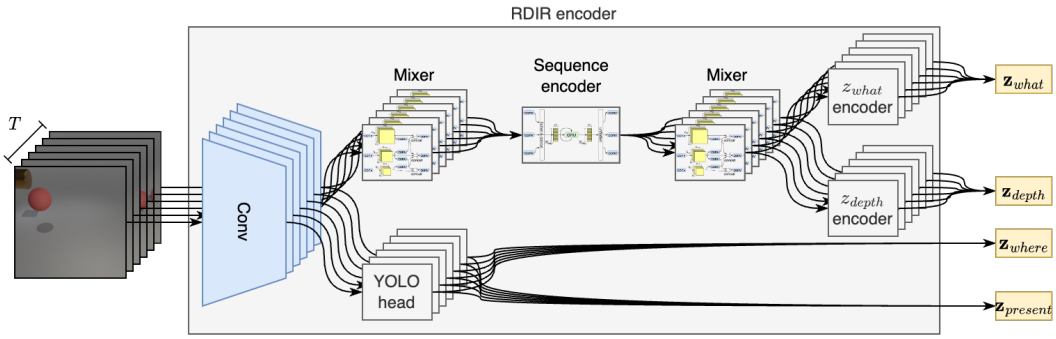

Figure 1. RDIR encoder architecture. Videos, treated as sequences of images, are processed individually by the convolutional backbone (*Conv*). The intermediate features, processed by a pretrained YOLO head, produce $\boldsymbol{z}_{where}$ and $\boldsymbol{z}_{present}$ latents for each cell in multi-scale grids. To infer $\boldsymbol{z}_{what}$ and $\boldsymbol{z}_{depth}$ embeddings, these intermediate features are forwarded to the encoding part of the model. *Mixer* (Fig. 3) enable sharing information across feature maps of all levels of grid resolutions, allowing the *Sequence encoder* (Fig. 2) to infer on scale-invariant representations. Next, another *Mixer* module is provided, transforming the RNN output to share temporal context across grids. Finally, the resulting latent features are used by $\boldsymbol{z}_{what}$ and $\boldsymbol{z}_{depth}$ encoders, producing embeddings for each cell in each grid.

to SSDIR. YOLOv4 is a well-established single-shot object detection model; its backbone prepares multi-scale feature maps, treated as grids of internal representations and used to predict per-cell bounding box coordinates and class confidences. By default, YOLOv4 utilizes three levels of feature grids, each with different resolutions and numbers of channels. RDIR leverages the backbone's capabilities to produce intermediate features for each time-step $t \in 1, ..., T$ of the input sequence and then processes these features with a pretrained YOLO head to infer objects' locations ($\boldsymbol{z}_{where}$) and class confidences (used to create $\boldsymbol{z}_{present}$). Then, RDIR re-uses the feature maps-based approach to learn representations of objects contained within each cell in all grids. We extend the approach proposed by SSDIR and propose significant advances to improve the quality of representations.

The key advance in RDIR is the utilization of the **Sequence encoder** (Fig. 2). Different from other spatial-attention models for learning representations on videos, RDIR assumes implicit modeling of sequential characteristics of videos. This is made possible by the application of grid-based inference. RDIR flattens all grids into a single tensor of shape $[N_{objs}, D]$ (where $N_{objs} = \sum_{i=1}^{3} w_i * h_i$ - the total number of cells in all grids ($i = \{1, 2, 3\}$), and $D = c_l$ - the number of channels used in all feature maps), which allows it to use a recurrent cell for propagating sequential information via its hidden state. After the recurrent cell, the grid structure is reconstructed, preserving the original shape and order of cells. This approach allows the model to consider temporal changes for all cells in each grid before learning objects' representations, without explicit discovery and propagation of objects. However, due to the separation of N_{objs} vectors in the GRU network, it cannot share any information across neighboring cells, which is crucial, especially for moving objects (as the object moves across the image it would activate different cells, which cannot share temporal information). To solve this issue, RDIR utilizes additional dimension-preserving convolutional layers, added before and after the recurrent cell. We leverage the primary characteristic of convolution which allows it to consider each cell's neighborhood, handling the case of sharing information across adjacent parts of the image.

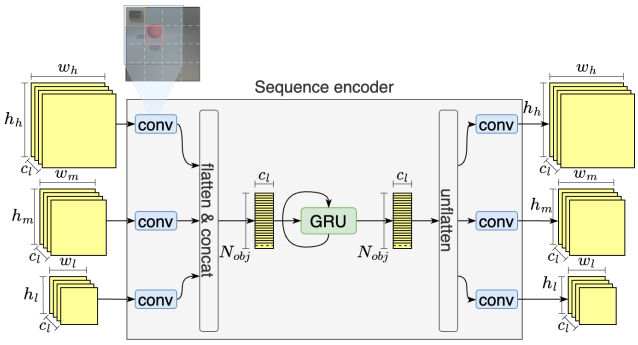

Figure 2. RDIR Sequence encoder processes the feature maps with dimension-preserving *conv layers*, which propagate information from adjacent cells via 2D convolutional layers with larger kernels (at least 3×3). The resulting feature maps are flattened to create stacked per-cell features, processed by a recurrent GRU cell. Next, the feature maps structure is recreated, and final *conv* layers are added to propagate the temporal state from the GRU cell across grids. The output feature maps are of the same shape as the input.

An important part of the encoder is the **Mixer** (Fig. 3. Multi-scale feature maps struggle with sharing information across objects detected on various level grids. This results from the fact, that these grids are extracted earlier in the backbone and do not share context in the lower-level representations, which on the other hand are crucial in the representation learning setup. To allow RDIR to consider information from all levels of feature maps we introduce a

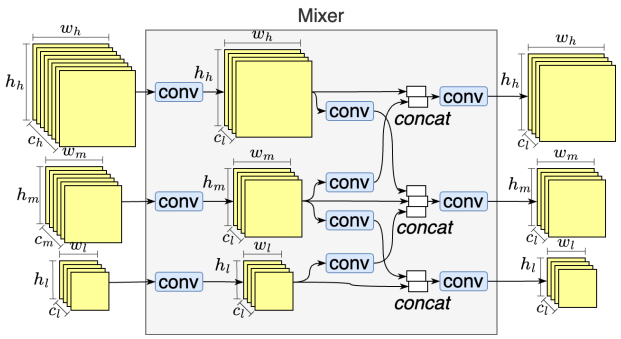

Figure 3. RDIR Mixer, used to share contexts between multi-level feature maps. The first set of *conv* layers (consisting of a 2D convolution layer, batch normalization, and Leaky ReLU activation blocks) unifies the number of channels across all feature maps to c_l. Then, a set of *conv* layers is used to up and down-scale all neighboring feature maps, enabling their concatenation. Concatenated feature maps are once again processed by the final set of *conv* layers, restoring the unified number of channels c_l.

mixing module; it performs up- and down-scaling to match the dimensions of adjacent feature maps. By concatenating the resulting feature maps with the original grids we allow these grids to consider information from other feature map levels, in the relevant part of images, as we preserve the order of cells. RDIR encoder utilizes two mixers: before and after the Sequence encoder. This approach allows the model to share information across feature grids both without the temporal context and after adding it to the GRU cell.

The encoder preserves the capability to scale to images of different shapes, solving this shortcoming of SSDIR. Also, thanks to a much more advanced convolutional backbone, it can be applied to complex datasets with larger input sizes. Similarly to SSDIR, the YOLO backbone and head weights are transferred from a pretrained object detection model and frozen for training, whereas the encoder part is trained together with the decoder without supervision, allowing the model to focus on relevant parts of the image during learning representations.

The decoder architecture is similar to those in other spatial-attention multi-object representation learning models, especially the one used in SSDIR, with several advancements: the ability to use non-max suppression, batch normalization, and adjustable reconstructed objects' size.

During training, the latent representation is filtered using sampled $\boldsymbol{z}_{present}$. We also add several negative examples apart from the detected objects to improve the stability of training and ensure the model can learn the entire data distribution. Then, selected object representations $\boldsymbol{z}_{what}$ are decoded using a convolutional decoder network. The result is transformed to the original image size using a spatial transformer [17]; these reconstructions are merged by sorting with the $\boldsymbol{z}_{depth}$ latent variable.

During inference, sampling $\boldsymbol{z}_{present}$ is replaced with thresholding, and no negative examples are added. We also utilize Non-Max Suppression to remove duplicated objects from the $\boldsymbol{z}_{present}$ latent variable. This allows the model to return a more accurate number of representations, especially in images containing many objects.

Model training is conducted in the setup of a classic autoencoder. In this research, the probabilistic characteristics of the model are not directly utilized, hence we opted for training the model using the mean squared error loss function, minimizing the reconstruction error between the original input and the reconstructed output (Eq. 1, where $\boldsymbol{X}$ refers to the input image and $\boldsymbol{Y}$ is the reconstruction, and N refers to the total number of samples in the dataset). It is worth pointing out, that extending this model to the Variational Autoencoder framework would be trivial.

$$MSE(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{N} \sum_{i}^{N} (X_i - Y_i)^2 \tag{1}$$

In the context of generative modeling, given the noise introduced by the grid-based encoding and injecting negative samples, this can be viewed as maximizing the likelihood of the data under the assumed model [13]. Since RDIR does not model background directly, MSE between input image and reconstruction can be inflated due to the complexity of the background. We address this issue by adding a component calculating MSE on areas detected by the model. The final loss function is presented in Eq. 2:

$$\mathcal{L} = \alpha_r * MSE\left(\boldsymbol{X}, \boldsymbol{Y}\right) + \alpha_{obj} \frac{1}{M} \sum_{i}^{M} MSE\left(\boldsymbol{X}^i_{obj}, \boldsymbol{Y}^i_{obj}\right) \tag{2}$$

where:
α_r is the reconstruction MSE component coefficient,
$\boldsymbol{X}$ is the input image,
$\boldsymbol{Y}$ is the image reconstruction,
α_{obj} is the per-object MSE component coefficient,
M is the number of objects in an image,
$\boldsymbol{X}^i_{obj}$, $\boldsymbol{Y}^i_{obj}$ refer to detected objects' original appearances and reconstructions.

We apply a staged training protocol. The weights of a pretrained object detection model are are transferred to an SSDIR-like model (with the same architecture as RDIR, but without Sequence encoder and the second Mixer), which is trained without supervision to establish good base weights for training RDIR (especially the $\boldsymbol{z}_{what}$ encoder and decoder). Then, the model is extended with the Sequence encoder and the second Mixer and trained as an unsupervised autoencoder. This way we achieve a shorter training time and achieve better performance.

4. Experiments and Discussion

In this section, we present the experimental setup and evaluation methodology used to assess the performance of RDIR. The objective is to evaluate the quality of the representations learned by RDIR while comparing it against baseline multi-object representation learning models. We analyze the consistency of learned representations and review their applicability in downstream tasks. Furthermore, we conduct an ablation study to examine the impact of the Mixer in the encoder of RDIR and the method's robustness to the number of objects in the sequence.

Datasets used in this research include real-world data, as well as images created using simulations. Generating data provides the ability to direct control over the dataset characteristics, necessary to review their influence on the model performance. We use a simple benchmark dataset, created from the MNIST dataset. The multi-scale moving MNIST dataset was generated by pasting a random number of digits (between 2 and 5) into the initial frame and then moving them with a varying speed throughout the sequence, varying the digit size. This dataset includes multiple occlusions and size changes and was used to verify the model's robustness.

The quality of representations was analyzed using the MOT15 dataset [24], a real-life multi-object tracking dataset. It consists of 11 sequences of images, each including multiple moving pedestrians; it is annotated with objects' positions and unique IDs, making it suitable for the task of object tracking. In this context, the COCO dataset [25] was used to train the initial object detection model (considering only the 'person' class).

The MOVi datasets come from the Kubric data generation pipeline [14] and were used to prove the stability of the representations produced by RDIR. See the Supplementary Material for an investigation on this matter.

Baseline methods To evaluate the performance of RDIR, we selected a set of baseline methods. The motivation behind this choice was to select one model from each category of multi-object representation learning models for videos. Therefore we have chosen 4 baseline methods:

- SCALOR [18], a single-scale spatial-attention model, utilizing a recurrent proposal-rejection mechanism for discovering and propagating detected object representations across the sequence,
- PROVIDE [31], a recurrent scene-mixture model, that leverages iterative amortized inference with a 2D-LSTM, incorporating information from previous refinement steps and previous frames,
- SIMONe [20], a transformer-based variational autoencoder with factorized latent representation, separating object attributes from global temporal context,
- SSDIR [32], with upgraded convolutional backbone from YOLOv4 (same as in RDIR) and with the addition of the Mixer, denoted as SSDIR-YOLO; used to verify the improvement of adding the Sequence encoder to the model; these upgrades overcome the major issues of the original approach (weak backbone and limited input image size).

We assume the same size of latent representation for each model. For the experiments on multi-scale moving MNIST dataset, we use a 64×64 input size, but since the size of grids in RDIR and SSDIR-YOLO depends on the input size, we upscale these images to 128×128.

Considering the significantly higher resolution of the MOT15 dataset, the object tracking experiment uses the highest possible resolution (limited by each method's design and GPU capability). Therefore, SCALOR and PROVIDE use their default 64×64 resolution, SIMONe utilizes 128×128 input, and SSDIR-YOLO and RDIR can work on 416×416 input data (thanks to their scalability).

4.1. Experiment 1: Predicting the sum of digits in a sequence

We evaluate quantitatively the quality of learned representations by utilizing them in a digit summation task. We adopt an experimental setup similar to one proposed in [1]: each model is trained on the full multi-scale moving MNIST dataset. Then, we extract the representations of each image for both the train and validation subsets and use them to fit a linear regression model, predicting the sum of digits in the sequence. The target is computed from the ground truth.

Since each of the models produces multiple representations for each image (referring to each detected object or attention mask), we aggregate them across the frame using summing. Then, to produce per-sequence representation we calculate the mean of each per-frame representation. The resulting 64-element vector is used as the feature vector for training a linear regression model. This approach enables direct comparison of methods, which do not explicitly learn representations of individual objects. Scene-mixture-based models (such as PROVIDE and SIMONe) create masks, that overlap multiple objects at once, whereas spatial attention models with fixed object size (such as SCALOR) tend to split large objects into smaller parts. By applying this aggregation we can evaluate these models regardless of the number of objects they infer on, and it allows us to review the information they collect on the entire video.

The quality of the regression model is determined by calculating R^2 metric across three random seeds for the train and val subset. The results are shown in Table 1.

RDIR achieves the best results for the full dataset (with scaling and translation), showing the improvement of apply-

	MNIST (scaling and translation)		MNIST (no scaling)		MNIST (no translation)	
R^2	train	val	train	val	train	val
SCALOR	0.301 ± 0.009	0.294 ± 0.028	0.353 ± 0.018	0.357 ± 0.024	0.288 ± 0.017	0.274 ± 0.010
PROVIDE	0.298 ± 0.258	0.282 ± 0.258	0.349 ± 0.300	0.345 ± 0.312	0.266 ± 0.232	0.253 ± 0.236
SIMONe	0.580 ± 0.02	0.577 ± 0.023	0.652 ± 0.013	0.658 ± 0.070	0.493 ± 0.018	0.478 ± 0.019
SSDIR-YOLO	0.574 ± 0.015	0.573 ± 0.012	0.641 ± 0.013	0.647 ± 0.015	0.396 ± 0.011	0.404 ± 0.019
RDIR	0.579 ± 0.010	0.581 ± 0.002	0.625 ± 0.006	0.636 ± 0.002	0.425 ± 0.009	0.429 ± 0.008

Table 1. Downstream task: regression of the sum of digits in a sequence. RDIR achieves best results on the most complex dataset but is slightly worse than SIMONe and SSDIR in simpler datasets (without scaling or translation). Values are averaged over 3 random seeds.

ing the Sequence encoder over the SSDIR approach. On the other hand, the metrics are comparable to those achieved by running linear regression on SIMONe representation, showing the benefits of its factorized latent space.

It is worth noting, that in the case of a simplified dataset (without scaling or translation), the performance of RDIR is slightly worse than SIMONe and SSDIR. This might be the result of a slight overfitting, as the final metrics are similar to the other models. Nevertheless, the ability to include temporal characteristics is an advantage over the SSDIR, especially in cases where we aim at preserving the high stability of representations. On the other hand, applying recurrent cells instead of a transformer network allows RDIR to be used in online inference, without the need for collecting the entire sequence of images.

The performance of SCALOR and PROVIDE is significantly worse than the other methods. During the research, these models struggled to appropriately discover objects in the scene, yielding a significantly larger number of representations per frame. This shows, that such models struggle to understand images with highly varying scenes and provide valuable object representations.

4.2. Experiment 2: Representations-based object tracking

We explore the application of RDIR representations in real-life scenarios by utilizing them for object tracking. We adopt the following setup: each model is trained on the training subset of the MOT15 dataset. Then, we process the validation dataset to extract objects' representations of each model. We apply a simple object tracker, which matches objects between consecutive frames using the Hungarian algorithm based on the cosine similarity between objects' representations, and keeping track of unique IDs of tracked objects. Finally, we use bounding box predictions of each model and the predicted object ID to evaluate the tracking performance, using TrackEval [19].

The quality of tracking is measured using the HOTA metric [29] across three random seeds for the train and validation subset. The results for each model and the baseline object detection (*YOLO*) are collected in Table 2.

	MOT15	
$HOTA$	train	val
SCALOR	1.452 ± 0.063	1.305 ± 0.041
PROVIDE	0.753 ± 0.038	0.698 ± 0.036
SIMONe	0.495 ± 0.275	0.724 ± 0.597
SSDIR-YOLO	31.774 ± 2.193	20.752 ± 1.190
RDIR	30.582 ± 2.201	20.749 ± 0.344
YOLO	20.100 ± 0.520	14.263 ± 0.525

Table 2. Downstream task: object tracking using learned representations. RDIR and SSDIR-YOLO achieve much better results than the baseline methods. Values are averaged over 3 random seeds.

The performance of SSDIR-YOLO and RDIR are similar; the metrics show a significant improvement over the *YOLO* detection model. Due to mask-based inference in PROVIDE and SIMONe, it is necessary to convert segmentation masks to bounding boxes. We calculate each mask's (i.e. object) center of mass and then build a bounding box using the area of the mask and a predefined $3:1$ ratio of object height (suitable for pedestrian class). Nevertheless, neither of the baseline methods was able to focus on individual objects in the scene, instead dividing it into meaningless parts, which resulted in very low tracking metric.

Leveraging a pretrained object detection model offers the ability to transfer knowledge from a different dataset. In Table 3 we examine the performance of models based on a detector pretrained on the COCO dataset (predicting the 'person' class). Even though the detector is less fitting to this problem (the performance of *YOLO*@COCO is low), the addition of representations learned by SSDIR-YOLO and RDIR yields a big increase in the object tracking metric. On the other hand, a model trained end-to-end (without the pretrained object detection model) behaves similarly to the baselines, not being able to focus on individual objects.

The marginal difference between SSDIR-YOLO and RDIR is worth addressing. Our implementation of SSDIR performs better than the original approach thanks to the improvements proposed in this paper. Even though the performance of SSDIR-YOLO and RDIR is similar, RDIR offers other advantages, especially more stable representations,

	MOT15	
HOTA	train	val
YOLO@MOT15	20.100 ± 0.520	14.263 ± 0.525
YOLO@COCO	8.240 ± 0.154	9.457 ± 0.141
SSDIR@MOT15	31.774 ± 2.193	20.752 ± 1.190
SSDIR@COCO	14.788 ± 0.607	14.029 ± 0.751
RDIR@MOT15	30.582 ± 2.201	20.749 ± 0.344
RDIR@COCO	13.922 ± 0.781	13.752 ± 0.777
RDIR E2E	0.634 ± 0.076	0.582 ± 0.120

Table 3. Downstream task: object tracking using learned representations. '@' denotes the dataset on which the base detection model was trained. The addition of representations significantly improves tracking performance, even for less-fitting object detection models. Values are averaged over 3 random seeds.

	MNIST	
R^2	train	val
RDIR	0.579 ± 0.010	0.581 ± 0.002
no-mixer	0.561 ± 0.023	0.562 ± 0.022
downscaler	0.543 ± 0.027	0.542 ± 0.027

Table 4. Ablation study: the influence of the Mixer module on the quality of representations. Adding the Mixer to RDIR improves the performance of the linear regression model applied to its representations. Values are averaged over 3 random seeds.

which are investigated in a study included in the Supplementary Material.

4.3. Ablation study: influence of the Mixer on model performance

We review the influence of the Mixer module in the RDIR model by training three models, and modifying their architecture. *RDIR* refers to a standard model, *no-mixer* is a model where latent features were not modified (this requires using separate recurrent cells for each feature map, as they use a varying number of channels), whereas *downscaler* only applies channels reduction, to use a single recurrent cell for all feature maps. Then, the digit summation experiment (Sec. 4.1) was repeated; the resulting R^2 metrics were collected in Table 4

It is clear, that applying the Mixer in RDIR improves the performance of the model. Interestingly, a simple reduction of the number of channels, which allows the use of a single recurrent cell does not provide an improvement over three separate recurrent cells for each latent representation.

4.4. Ablation study: influence of the number of objects in the sequence on model performance

We verify RDIR's robustness to varying numbers of objects visible in a sequence by running a trained model on

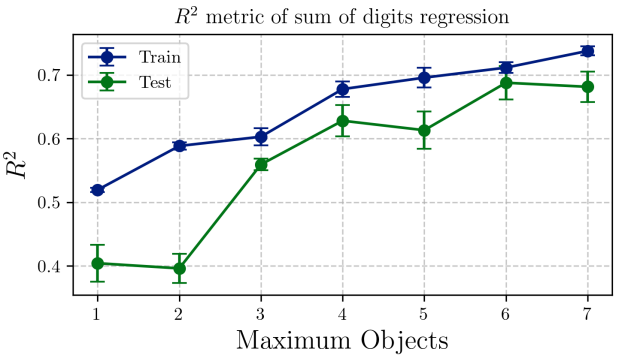

Figure 4. Ablation study: the influence of the number of objects in a sequence on the quality of representations. RDIR can generalize to a larger number of objects, but the performance of the downstream model is worse for one- and two-object sequences.

additionally generated moving multi-scale MNIST datasets, containing from 1 up to 7 objects in each sequence, split into train and validation subsets ($80 : 20$). We follow the digit summation procedure (Sec. 4.1) and evaluate linear regression models trained on the representations. The R^2 metric and standard deviation for each dataset are shown in Fig. 4.

The results demonstrate the robustness of the RDIR latent space across sequences with varying numbers of objects. The model consistently generates versatile embeddings, irrespective of the object count in the sequence. Notably, a decline in performance is observed in sequences containing only one or two objects.

5. Conclusion

In this paper, we presented RDIR, a novel method for capturing stable representations of multiple objects on videos from a pretrained object detector. We showed how its latent space can be used by means of applying representations in downstream tasks. We reviewed the ability to transfer knowledge from other datasets into similar tasks and showed the improvement gained from representations inferred by RDIR. We also examined the sensitivity of the model to the dataset characteristics, showing its robustness to a larger number of objects.

The research could be continued in several aspects. The model could be trained on a larger unannotated dataset by transferring knowledge from another dataset and applied in demanding downstream tasks. Another interesting direction would be reviewing a more advanced way of modeling object interactions, as they are crucial for a profound understanding of the scene. Finally, the approach could be based on an instance segmentation model, leading to more granular masks of objects in the scene.

References

[1] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in Neural Information Processing Systems*, (Nips):3233–3241, 2016. 2, 6

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, aug 2013. 1

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. 3

[4] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. pages 1–22, 2019. 2

[5] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 2

[6] Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 3412–3420, 2019. 2, 3

[7] Eric Crawford and Joelle Pineau. Exploiting spatial invariance for scalable unsupervised object tracking. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 1, 2

[8] Antonia Creswell, Rishabh Kabra, Christopher P. Burgess, and Murray Shanahan. Unsupervised object-based transition models for 3d partially observable environments. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 2

[9] Gamaleldin F. Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C. Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2

[10] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2020. 2

[11] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. GENESIS-V2: Inferring Unordered Object Representations without Iterative Refinement. *arXiv preprint arXiv:2104.09958*, 2021. 2

[12] Russell A. Epstein and Chris I. Baker. Scene perception in the human brain. *Annual Review of Vision Science*, 5(1):373–397, 2019. PMID: 31226012. 1

[13] Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. 5

[14] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 6

[15] Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loie Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:4317–4343, 2019. 2

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

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2017–2025, Cambridge, MA, USA, 2015. MIT Press. 3, 5

[18] Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *Proceedings of ICLR 2020*. OpenReview.net, 2020. 1, 2, 6

[19] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020. 7

[20] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 1, 2, 6

[21] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2

[22] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[23] Adam Roman Kosiorek, Hyunjik Kim, Ingmar Posner, and Yee Whye Teh. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems*, 2018. 1, 2

[24] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942. 6

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 6

[26] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020. 2, 3

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing. 3

[28] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. 2

[29] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020. 7

[30] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. 2

[31] Polina Zablotskaia, Edoardo A. Dominici, Leonid Sigal, and Andreas M. Lehrmann. Provide: a probabilistic framework for unsupervised video decomposition. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 2019–2028. PMLR, 27–30 Jul 2021. 1, 2, 6

[32] Piotr Zieliński and Tomasz Kajdanowicz. Learning scale-invariant object representations with a single-shot convolutional generative model. In Derek Groen, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot, editors, *Computational Science – ICCS 2022*, pages 613–626, Cham, 2022. Springer International Publishing. 1, 2, 3, 6