Learning Object-Centric Dynamic Modes from Video and Emerging Properties

Armand Comas Massague COMASMASSAGUE.A@NORTHEASTERN.EDU
Christian Fernandez-Lopez* FERNANDEZLOPEZ.C@NORTHEASTERN.EDU
Sandesh Ghimire* S.GHIMIRE@NORTHEASTERN.EDU
Haolin Li LI.HAOLIN@NORTHEASTERN.EDU
Mario Sznaier MSZNAIER@COE.NEU.EDU
Octavia Camps CAMPS@COE.NEU.EDU
Northeastern University

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

One of the long-term objectives of Machine Learning is to endow machines with the capacity of structuring and interpreting the world as we do. This is particularly challenging in scenes involving time series, such as video sequences, since seemingly different data can correspond to the same underlying dynamics. Recent approaches seek to decompose video sequences into their composing objects, attributes and dynamics in a self-supervised fashion, thus simplifying the task of learning suitable features that can be used to analyze each component. While existing methods can successfully disentangle dynamics from other components, there have been relatively few efforts in learning parsimonious representations of these underlying dynamics. In this paper, motivated by recent advances in non-linear identification, we propose a method to decompose a video into moving objects, their attributes and the dynamic modes of their trajectories. We model video dynamics as the output of a Koopman operator to be learned from the available data. In this context, the dynamic information contained in the scene is encapsulated in the eigenvalues and eigenvectors of the Koopman operator, providing an interpretable and parsimonious representation. We show that such decomposition can be used for instance to perform video analytics, predict future frames or generate synthetic video. We test our framework in a variety of datasets that encompass different dynamic scenarios, while illustrating the novel features that emerge from our dynamic modes decomposition: Video dynamics interpretation and user manipulation at test-time. We successfully forecast challenging object trajectories from pixels, achieving competitive performance while drawing useful insights.

Keywords: Koopman operator, Non-linear identification, Dynamics-constrained learning, Representation learning, Video manipulation

1. Introduction

Unsupervised learning of symbolic representations from high dimensional data poses a great challenge to current machine intelligence. As humans, our cognitive model of the world is based on segregation of reality into abstract categories, or symbols. We construct novel behaviours by dynamic reuse of familiar symbols Greff et al. (2020). In visual scenes, we can think of objects and their attributes as one valid set of symbolic representations.

*. These authors contributed equally to this work

There have been numerous efforts to model images from an object-centric perspective without supervision Eslami et al. (2016); Greff et al. (2019); Locatello et al. (2020); Burgess et al. (2019); Lin et al. (2020b). These works use inductive biases encoded in their architecture to decompose a static scene into its objects and their attributes.

As a natural extension, recent research has tackled unsupervised video decomposition Kosiorek et al. (2018); Denton and Birodkar (2017); Comas et al. (2020); Hsieh et al. (2018); Kossen et al. (2020); He et al. (2019); Jiang et al. (2020); Kipf et al. (2020); Jaques et al. (2021); Watter et al. (2015); Karl et al. (2016). A key challenge here is tracking the decomposed objects across time while learning to define what these objects are. Many of these works also attempt to model the intrinsic dynamics of the objects in the scene, and thus how the dynamic scene will look like in the future. While attempting to decompose the video into symbolic representations, these works rely on black-box neural architectures to model dynamics. The usual choices are Recurrent Neural Network (RNN) or Graph Neural Netowrk (GNN) architectures. Interpreting dynamics from RNN models is almost unfeasable. While GNNs are principled and object-centric, their formulation still relies on opaque architectures for the dynamical model. Thus, while their frame predictions are often accurate, the used dynamical models lack a user-interpretable formulation: even though a user can potentially manipulate objects and their attributes in these models, there is no apparent controllable way to manipulate their dynamics or to understand their underlying propagation mechanism.

In this work, we advocate for decomposition not only from a visual perspective, but also from a dynamics perspective. We argue that decomposition leads to user interpretability. We want a user to be able to manipulate object dynamic behaviour in a controlled manner, directly on the pixel domain.

With this objective, we propose to combine neural network-based architectures with Koopman operator theory, which is based on the insight that a finite-dimensional nonlinear system can be transformed into an infinite-dimensional linear dynamical system, with a linear operator $\mathcal{K}$. In this framework, the dynamical system can be understood as a composition of first and second order impulse responses, i.e. “Dynamic Modes" (DM), which can be extracted from a linear (Koopman) operator through spectral decomposition.

We introduce Object-centric Koopman-based Interpretable Decomposition (`OKID`), which is posed as a step forward towards dynamics interpretability through decomposition. Our method is self-supervised and (**1**) uses an attention-based tracking method to learn video representations, factorized into moving objects and their attributes; (**2**) decomposes the discovered trajectories into linear dynamic modes by finding a mapping to the Koopman space; (**3**) learns a global Koopman operator that characterizes the underlying dynamics of the training data; and hence (**4**) performs video decomposition, prediction and interpolation. Our main contributions are:

(i) **We propose `OKID`, that jointly discovers object representations and performs attribute and dynamics decomposition in a self-supervised setting**. `OKID` uses a novel approach to learn a modular dynamical system from data by means of Koopman theory. While Koopman theory has been employed before for the dynamics and interaction prediction of multiple object trajectories, we propose a novel end-to-end method that can do so from pixels.

(ii) **We provide simple video manipulation techniques that unveil novel properties for a video prediction model**. We illustrate this in our experiments. The proposed framework allows for easy video dynamics interpretation and manipulation at test-time. Although we do not improve the state of the art in the task of video prediction, we remain competitive generating visual results of good quality and plausible object trajectories, while enabling interpretability.

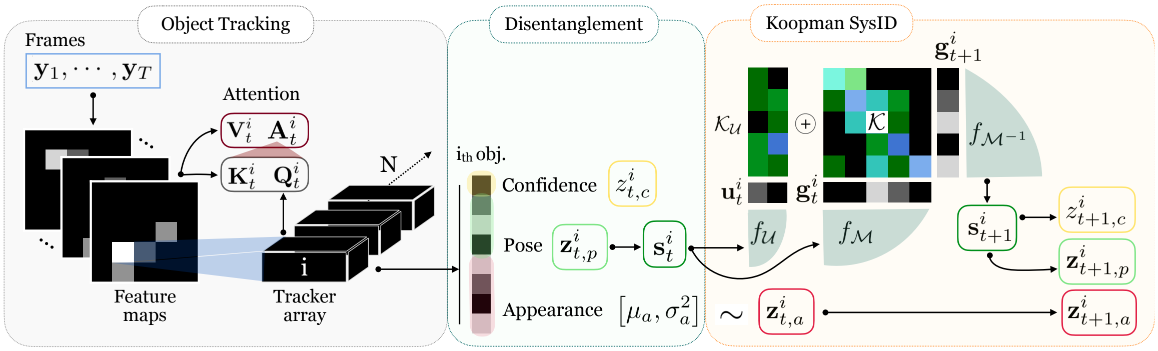

Figure 1: Architecture of OKID. Left: an attention-based recurrent tracker decomposes the scene into its objects. Center: the object representations are disentangled into Confidence, Appearance and Pose. Right: Pose is modeled and forecasted by using a Koopman embedding. Latent representations are later used to reconstruct or predict frames.

2. Background

Consider a time-invariant dynamical system of a single object on $\Re^n$ of the form $\mathbf{s}_{t+1} = \Phi(\mathbf{s}_t)$. where $\mathbf{s}_t \in \Re^n$ is the state of the system at time t, and $\Phi : \Re^n \rightarrow \Re^n$ is a, potentially non-linear, function that defines the temporal transition of the states.

The fundamental insight of Koopman operator theory is that finite-dimensional nonlinear dynamics can be transformed to an infinite-dimensional linear dynamical system by appropriately chosing a Hilbert space of observables $\mathbf{g}$ Koopman (1931); Mezic (2005) and seeking an operator $\mathcal{K}$ that propagates $\mathbf{g}$ one step ahead:

$$\mathbf{g}_t = f_\psi(\mathbf{s}_t), \quad \mathbf{g}_{t+1} = \mathcal{K}\mathbf{g}_t, \tag{1}$$

$$f_\psi \circ \Phi(\mathbf{s}_t) = \mathcal{K} f_\psi(\mathbf{s}_t), \tag{2}$$

where f_ψ is the mapping from the state space to the observable space $\mathbf{g}_t$ and $\circ$ denotes the composition operator. While Koopman operator theory applies for an infinite dimensional Hilbert observable space, here we make a simplified assumption that it holds for some finite dimensional subspace, $\Re^m$ of observables, *i.e.*, $\mathbf{g}_t \in \Re^m$, $f_\psi : \Re^n \rightarrow \Re^m$, $\mathcal{K} : \Re^m \rightarrow \Re^m$.

The eigenfunctions $\psi_j(\mathbf{s}_t) : \Re^n \rightarrow \Re^m$ of the Koopman operator satisfy $\mathcal{K}\psi_j(\mathbf{s}_t) = \lambda_j \psi_j(\mathbf{s}_t)$, where $\lambda_j \in \mathbb{C}$ is the corresponding eigenvalue. Eigenfunctions are hard to find, and some algorithms have been proposed to tackle the challenge. The most widely used are the dynamic mode decomposition (DMD) Schmid (2008) and its extension to nonlinear observables, the extended DMD (EDMD) Williams et al. (2015).

Previous research used hand-crafted functions to model the observable space. Some recent approaches use deep neural networks to represent the observable space Lusch et al. (2018); Morton et al. (2019); Li et al. (2020); Xiao et al. (2021); Azencot et al. (2020). Neural networks have the advantage of being universal approximators, and are effective in finding the Koopman invariant subspace. Koopman methodology is data-driven, model-free and can discover the underlying dynamics and control of a given system from data alone Proctor et al. (2018). The operator $\mathcal{K}$ is usually found by linear regression given historical data or by end-to-end gradient-descent-based optimization.

In some cases, Koopman is employed in presence of control inputs. There are different approaches to introducing inputs to Koopman (*e.g.* Proctor et al. (2018); Li et al. (2020)). In our case, inputs model forces external to an object, originated from object-environment interactions. Therefore, inputs will depend both on the state of the object and the environment's geometry. Thus, the counterpart of equation 1 is:

$$\mathbf{g}_t = f_\psi\left(\mathbf{s}_t\right), \quad \mathbf{u}_t = f_\mathcal{U}\left(\mathbf{s}_t\right), \tag{3}$$

$$\mathbf{g}_{t+1} = \mathcal{K}\mathbf{g}_t + \mathcal{K}_\mathcal{U}\mathbf{u}_t \tag{4}$$

Here, $\mathbf{u}_t$ depends on the current state $\mathbf{s}_t$ (closed-loop control). It is expected to be sparse, and low-dimensional. $\mathcal{K}_\mathcal{U} : \Re^v \rightarrow \Re^m$ is the input Koopman operator. We usually define dimension v such that $v << m$.

3. Related Work

Video decomposition. DRNET Denton and Birodkar (2017) is an early work that decomposes video into a static component (content) and a dynamic component (pose). This is a key idea adopted by several subsequent works, such as DDPAE Hsieh et al. (2018) and SQAIR Kosiorek et al. (2018). These methods are not designed to be scalable or explicitly model interactions. SCALOR Jiang et al. (2020) targeted scalability by massive parallelization to handle hundreds of objects in a scene. STOVE Kossen et al. (2020) adds on by modeling interactions with a graph neural network. By using a Markov model in the latent space, STOVE applies inference in the series. G-SWM Lin et al. (2020a) unifies the abilities of previous models, it is scalable and handles interactions. ODDN Tang et al. (2022), distills object dynamics and models their interactions in an unsupervised way using a clustering type approach. Our method differs from the previous methods substantially in the dynamics model. Finally, TBA He et al. (2019) presents a method that tracks by decomposition, disentanglement and generation of objects, however it does not provide a forecasting mechanism.

Koopman Operator. Koopman Operator theory has been successfully used to disentangle the dynamic modes in complex dynamical systems using decomposition techniques like DMD Schmid (2008), and Extended DMD (EDMD)Williams et al. (2015). Leveraging the fact that Koopman operator based methods require a rich family of functions to generate a mapping, recent works have used deep neural networks to approximate the eigenvectors associated to the Koopman operator Lusch et al. (2018); Azencot et al. (2020); Morton et al. (2019); Li et al. (2020); Xiao et al. (2021); Otto and Rowley (2019). This idea has found applications in fluid dynamics Morton et al. (2018); Azencot et al. (2020), atomic and molecular scale dynamics Xie et al. (2019); Mardt et al. (2017), chaotic systems Brunton et al. (2017) and traffic dynamics Xie et al. (2019). However, to the best of our knowledge, video sequences have not still been targeted. Proctor et al. (2018) generalized Koopman theory for control inputs, giving rise to other methods, such as Li et al. (2020), that use Koopman theory to model object dynamics and interactions from inputs and coordinate-based states.

4. Method

Video often presents multiple objects in motion that generate a complex dynamical scene. In the pixel space, dynamics are highly complex. But object trajectories are often simpler. For our approach, we decompose the scene into its moving objects. We track and identify N objects across input frames, and assign a set of 3 variables to each one of them. Each object representation will be disentangled into the following categories:

- **Pose** $\mathbf{z}_{t,p} \in (-1, 1)^4$: Indicates the parameters for a 2D affine spatial transformation of an object; x and y centroid coordinates, scale and ratio, scaled to the range $(-1, 1)$ with a $\texttt{Tanh}(\cdot)$ activation.
- **Appearance** $\mathbf{z}_{t,a} \in \Re^A$: A vector containing information about an object's appearance.
- **Confidence** $z_{t,c} \in (0, 1)$: Probability indicating the certainty of an object being correctly modeled.

As shown in Fig.1, the architecture consists of: 1) an object tracking block, 2) a Koopman SysID block and 3) a rendering block (not illustrated). We first track objects in a video by means of attention. The tracking block estimates the above attributes from pixels. A state $\mathbf{s}$ based on the pose is then embedded into the observable space with a Koopman mapping. The Koopman operator $\mathcal{K}$ is then used to propagate the trajectory in time. We decode the Koopman predictions back into the attribute space and render a new scene with the estimated set of object attributes. We reconstruct and predict video scenes, and use the ground-truth video to train the entire model end-to-end. We assume that there is no background. Next, we describe in more detail the main modules that form our model.

4.1. Architecture

Tracking. We encode each video frame $(\mathbf{y}_1, \cdots, \mathbf{y}_T)$ with a convolutional encoder, and obtain a feature map as $\mathbf{H}_{t,y} = f_{\texttt{enc}}([\mathbf{y}_t, \texttt{PE}])$, where $\texttt{PE}$ is a positional encoding. We then track objects across frames by using an array of trackers. N trackers are initialized, where N is an *upper-bound* on the expected number of objects in every scene. In our implementation, we use trackers based on He et al. (2019), which we modified to allow for forecasting and stochastic appearance modeling. We define the tracker recurrent updates as:

$$\begin{aligned} &\mathbf{h}^i_{t,tr} = f_{\texttt{tr}}\left(\mathbf{h}^i_{t-1,tr}, \mathbf{H}_{t,y}\right), \quad \left[z^i_{t,c}, \mathbf{z}^i_{t,p}, \mathbf{d}^i_{t,a}\right] = \texttt{FC}\left(\mathbf{h}^i_{t,tr}\right) \\ &\mathbf{z}^i_{t,a} \sim \mathcal{N}(\mu_a, \sigma_a^2), \quad \left[\mu_a, \sigma_a^2\right] = \texttt{FC}(\mathbf{d}^i_{t,a}), \end{aligned} \tag{5}$$

where $f_{\texttt{tr}}(\cdot)$ is an attention-based tracking function described in equation 6. We implement the appearance latent vector $\mathbf{z}^i_{t,a}$ to be the only stochastic variable in this setup, given its inherent complexity. It is sampled from a Gaussian distribution and trained as in a VAE framework. $z^i_{t,c}$ is softly binarized with $\texttt{Sigmoid}(\cdot)$.

Tracker updates. The tracking function, $f_{\texttt{tr}}(\cdot)$ is based on recurrent updates:

$$\mathbf{h}^i_{t,tr} = \texttt{GRU}_{\texttt{tr}}\left(\mathbf{h}^i_{t-1,tr}, \mathbf{u}^i_t\right), \mathbf{u}^i_t = \mathbf{A}^i_t \mathbf{V}_t, \quad \mathbf{A}^i_t = \texttt{Softmax}\left(\beta^i_t \mathbf{Q}^i_t \mathbf{K}^T_t\right) \tag{6}$$

where a $\texttt{GRU}$ cell is used to update the hidden state of each tracker $\mathbf{h}^i_{t,tr}$ and $\mathbf{u}^i_t$ is obtained by soft-attention mechanism, relying on the Query, Key, Value triad, where value and key are set as $\mathbf{V}^i_t = \mathbf{K}^i_t = \mathbf{H}_{t,y}$, and the Query is obtained from $\mathbf{h}^i_{t-1,tr}$ using a fully connected layer.

We need a mechanism to provide information across trackers, while preserving the identity of each object through time. Trackers interact through external memory by using interface variables. We describe the tracker in more detail in the technical report Comas et al. (2023).

Koopman Embedding. We employ Koopman theory to model the dynamics. We argue that finding dynamic modes will introduce benefits to our model, as we discuss in Section 5.
We define the state as a concatenation of T_S delayed instances (from now on referred to as delayed coordinates) of the pose vector $\mathbf{z}_p^i$ for the i^{th} object:

$$\mathbf{s}_t^i = \left[\mathbf{z}_{t,p}^i, \mathbf{z}_{t-1,p}^i, \ldots, \mathbf{z}_{t-T_S+1,p}^i\right] \tag{7}$$

Therefore, $\mathbf{s}_t^i \in \mathbb{R}^{4T_s}$. T_s indicates our prior belief on the number of time-steps needed to model dynamics with an Auto-Regressive (AR) approach.

The observables $\mathbf{g}_t^i$ are obtained through the Koopman mapping $f_{\mathcal{M}} : \Re^{4\times T_s} \to \Re^{\mathcal{M}}$ (equivalent to equation 1). $\mathcal{M}$ and $\mathcal{U}$ make reference to the spaces of observables and inputs, respectively. We recover the original state by approximating the inverse function $f_{\mathcal{M}^{-1}} \approx f_{\mathcal{M}}^{-1}$ with a deterministic Auto-Encoder (AE) architecture. We use an MLP to map states to observables in the Koopman space, and vice-versa. If external forces are present, we also model inputs $\mathbf{u}_t$ as a non-linear mapping $f_{\mathcal{U}} : \Re^{4T_s} \to (-1, 1)^{\mathcal{U}}$ from the state space. Note that we set $f_{\mathcal{U}} : \Re^{4T_s} \to \{0\}^{\mathcal{U}}$ if we assume that the objects' dynamics are not affected by the environment:

$$\hat{\mathbf{s}}_{t+1}^i = f_{\mathcal{M}^{-1}} \circ \left(\mathcal{K} \circ f_{\mathcal{M}}\left(\mathbf{s}_t^i\right) + \mathcal{K}_{\mathcal{U}} \circ f_{\mathcal{U}}\left(\mathbf{s}_t^i\right)\right) = f_{\mathcal{M}^{-1}}\left(\mathcal{K}\mathbf{g}_t^i + \mathcal{K}_{\mathcal{U}}\mathbf{u}_t^i\right). \tag{8}$$

We refer to the estimated states as $\hat{\mathbf{s}}_T^i$. We define the Koopman operators $\mathcal{K} : \Re^{\mathcal{M}} \to \Re^{\mathcal{M}}$ and $\mathcal{K}_{\mathcal{U}} : \Re^{\mathcal{U}} \to \Re^{\mathcal{M}}$ as parameter matrices. We provide a discussion of the role of $\mathbf{u}_t$ in the technical report Comas et al. (2023). The pose vector $\hat{\mathbf{z}}_{t+1,p}^i$ is recovered by keeping the first stacked coordinates of the estimated state $\hat{\mathbf{s}}_{t+1}^i$, and limited to the range $(-1, 1)$. Once trained, the eigendecomposition of the Koopman operator $\mathcal{K}$ provides us with useful insights of the scene dynamics.

4.2. Training

Given the video sequence $(\mathbf{y}_1, \cdots, \mathbf{y}_T)$, its generative distribution is given by:

$$p(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}) = \prod_{i=1}^{N} p_{rec}(\mathbf{y}_{1:T}^i|\mathbf{z}_{1:T}^i) \prod_{i=1}^{N} p_{pred}(\mathbf{y}_{1:T}^i|\mathbf{z}_{1:K}^i), \tag{9}$$

where $\mathbf{z}_t^i = \left[z_{t,c}^i, \mathbf{z}_{t,p}^i, \mathbf{z}_{t,a}^i\right]$. Note that we both reconstruct the whole sequence with length T and predict it from K initial frames ($K < T$). For our experiments, K will coincide with the number of delayed coordinates T_s. Given the inferred latent variables, we reconstruct and predict $\mathbf{y}_i^t$ for each object sequentially. We first generate the object in the center with resolution $R = C \times h \times w$, given the appearance $\mathbf{z}_{t,a}^i$. The decoder $f_{\text{dec}} : \Re^A \to \Re^R$ is a CNN. We then apply a spatial transformer $\mathcal{T}$ to rescale and place the object according to the pose $\mathbf{z}_{t,p}^i$. For each object, the generative model is: $p(\mathbf{y}_t^i|\mathbf{z}_{t,a}^i) = \mathcal{T}(f_{\text{dec}}(\mathbf{z}_{t,a}^i); \mathbf{z}_{t,p}^i) \circ z_{t,c}^i$. Similarly to the VAE framework, we train the model by maximizing the evidence lower bound (ELBO).

We use self-supervision for reconstructing the input $\mathbf{y}_{1:T}$ and predicting that same input from few initial conditions $(1 : K)$. While the ELBO is enough to perform prediction, we find that adding regularizers helps achieving some of the desired features. Thus, our objective $\mathcal{J}_\omega$ with respect to the trainable weights ω is:

$$\mathcal{J}_\omega = \min_\omega \left[-\text{ELBO}+\lambda L_{\text{Reg}}\right] \tag{10}$$

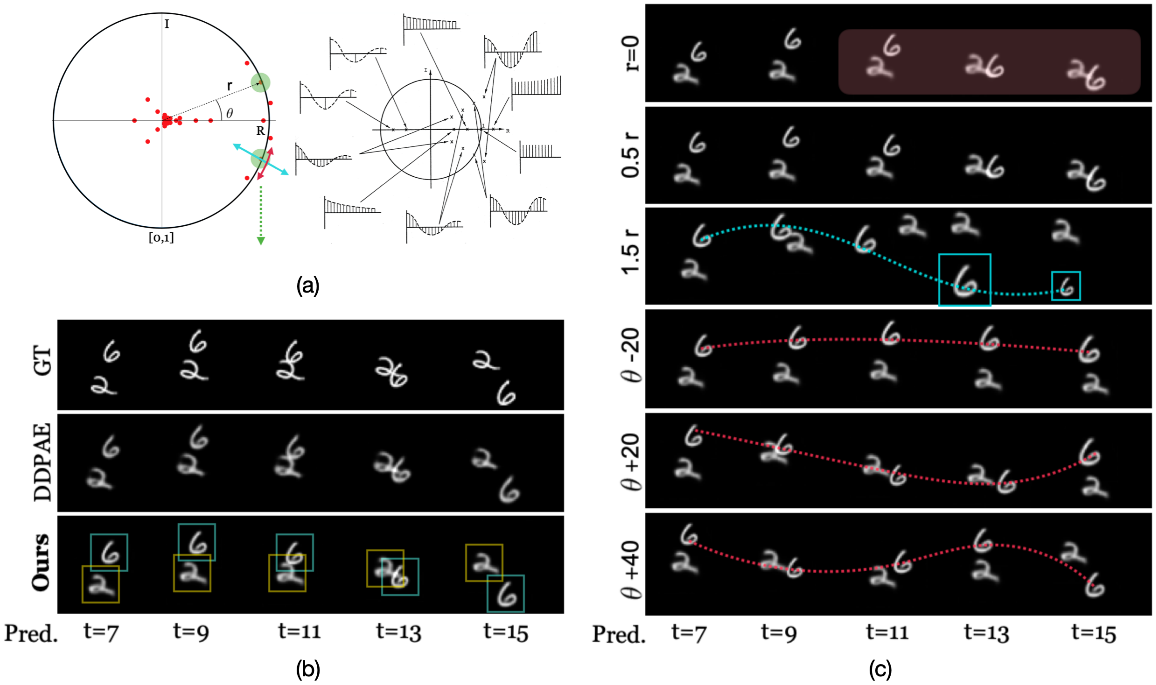

Figure 2: **(a)** right: Illustration of the eigenvalue behaviors of matrix $\mathcal{K}$ Phillips and Nagle (2007) visualized on the complex-plane, left: Learned Koopman matrix eigenvalues for 3D to 2D projection experiment. In green we highlight the eigenvalue pair that we will modify in (c) to visualize the effect of manipulations. **(b)**: Predictions: Ground Truth; our main baseline DDPAE; OKID with the object decomposition bounding boxes; **(c)** by rows: 2 variations to the radius of the highlighted eigenvalue, 3 variations to the angle. Blue line shows the effect of changing the radius and red line shows that of changing the angle.

In order to achieve the desired properties in the Koopman space, we regularize the objective with $L_{\texttt{Reg}}$, as described next. We enforce the observables $\mathbf{g}$ and states $\mathbf{s}$ to be consistent before and after the Koopman embedding. This is done through an autoencoding loss. We might also expect the presence of sparse inputs $\mathbf{u}$, as in the Collision case of Moving MNIST experiments. In that case, we enforce sparsity with its convex surrogate, the ℓ_1 norm. Hence, the full regularization term is given by:

$$L_{\text{reg}} = \lambda_{\texttt{fit}} L_{\texttt{fit}} + \lambda_{\texttt{AE}} L_{\texttt{AE}} + \lambda_{\mathcal{U}} L_{\mathcal{U}} \tag{11}$$

$$L_{\texttt{AE}} = \left\| \mathbf{s}^{1:N}_{2:T} - \hat{\mathbf{s}}^{1:N}_{2:T|1:T-1} \right\|_1, L_{\texttt{fit}} = \left\| \mathbf{g}^{1:N}_{2:T} - \hat{\mathbf{g}}^{1:N}_{2:T|1:T-1} \right\|_2^2, L_{\mathcal{U}} = \left\| \hat{\mathbf{u}}^{1:N}_{1:T} \right\|_1,$$

Here, $\|\cdot\|_1$ and $\|\cdot\|_2^2$ indicate the ℓ_1 and ℓ_2 losses respectively. $\|\cdot\|_1$ is used to enforce sparsity in $\hat{\mathbf{u}}^{1:N}_{1:T}$. Finally, the λs are the weights applied to each one of the loss terms. Details about the ELBO objective and the frame generation can be found in the technical report Comas et al. (2023).

5. Experiments

We ran a set of experiments to evaluate the ability of our framework to model dynamic scenarios and illustrate its interpretability and manipulation capabilities. For this purpose, we tested OKID on variations of the Moving-MNIST dataset, which allows us to test our model in dynamic scenarios that range from simple to complex.

Our baselines are established state-of-the-art methods for decomposed self-supervised video generation: DDPAE Hsieh et al. (2018) (selected as the closest to OKID in terms of architecture), DRNET Denton and Birodkar (2017) and SCALOR Jiang et al. (2020). All baselines model dynamics with LSTM-like modules.

Evaluation Metrics. Our quantitative results will be measured in terms of per frame Mean Square Error (MSE), Mean Absolute Error (MAE), Structural Similarity (SSIM) and the Perceptual Similarity Metric (LPIPS) Zhang et al. (2018).

5.1. Moving MNIST Experiments

Moving MNIST Srivastava et al. (2015) is a synthetic dataset consisting of two digits with size 28×28 moving independently in a 64×64 frame. Each training sequence is generated on-the-fly by sampling MNIST digits and synthesizing trajectories according to a definition of the motion. Our model uses 10k samples for training, 1k for validation and 2k for testing. We simulate 4 scenarios:

- ***Circular motion***: We generate a fairly simple dataset, for which we know the expected results. We sample randomly initial coordinates (x_0, y_0), radius R and the angular step length. We generate the motion with equation: $x = R\cos(t) + x_0$, $y = R\sin(t) + y_0$, where t increases linearly with a slope given by the angular step length. Finally, we constrain the motion to the dimensions of the frame. From $T = 3$ input frames, we predict 17 frames.

- ***Cropped circular motion***: We mask the 29 top rows of the circular motion case, simulating a partially cropped frame. Note that an object of size 28×28 can be completely occluded.

- ***Inelastic/Super-elastic collisions***: We sample initial coordinates (x_0, y_0) and angle θ and let the object collide against the frame boundaries with fixed velocity. We increase the complexity of the case by simulating an inelastic response from the left and top boundaries $(\times 0.8)$ and a super-elastic response from the right and bottom boundaries $(\times 1.25)$. We generate chunks of $T = 13$ frames as input. From the first 3 frames, we predict 10 frames.

- ***3D to 2D motion projection***: Finally, we generate trajectories that lay withing the cube $[-1, 1]^3$. Those are generated randomly following the steps described in the technical report. We project the trajectory to the $xy-$ axis, and simulate depth with the relative sizes of the objects. From 6 input frames, we predict 10.

Quantitative Results. A quantitative general overview of the experiments is given in Table 1. Looking at the perceptual metrics, OKID often outperforms all baselines in terms of LPIPS, and is on par with DDPAE's SSIM. DDPAE is the closest to ours in terms of decomposition and frame generation. The key difference is the dynamics modeling. DDPAE uses a concatenation of LSTMs for reconstructing and predicting the pose. On the absolute MSE and MAE metrics, OKID performance remains competitive while providing a novel set of useful features:

Model Reduction. We propose to test our architecture after removing all but n eigenvalues of $\mathcal{K}$. n was chosen to ensure we do not discard eigenvalues with module higher than 0.5. As we see in Table 1, the results are close to the prediction with the full matrix $\mathcal{K}$. This indicates that our model was able to capture the motion of the digits with 3 to 4 conjugate pairs of eigenvalues for the studied cases.

Table 1: Quantitative comparison of all methods for the four scenarios. We evaluate reconstruction, prediction, and prediction with model reduction (ne denotes that we keep only the top n eigenvalues). Top to bottom shows results of different methods on several datasets. Our method performs similarly to the best RNN-based baseline and outperforms the rest of baselines in prediction.

Model	MSE ↓			MAE ↓			SSIM ↑			LPIPS ↓		
Circular	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)
DDPAE	**40.13**	**71.96**	/	**123.94**	**162.64**	/	**0.87**	**0.82**	/	0.19	0.21	/
OKID	59.97	84.23	84.64	139.92	168.83	169.96	0.86	**0.82**	0.82	**0.15**	**0.17**	0.17
Cropped circular	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)
DDPAE	**35.04**	**54.95**	/	**98.48**	**118.90**	/	**0.88**	**0.85**	/	0.21	0.23	/
OKID	47.80	59.26	59.32	108.28	120.04	120.15	**0.88**	0.86	0.86	**0.17**	**0.19**	0.19
Collision	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)	Rec	Pred	Pred(6e)
DRNET	109.01	214.49	/	218.14	339.58	/	0.75	0.60	/	0.30	0.40	/
SCALOR	**13.24**	329.63	/	**50.02**	494.41	/	**0.95**	0.28	/	0.02	0.48	/
DDPAE	49.53	**93.52**	/	146.65	**199.57**	/	0.84	0.76	/	0.23	0.25	/
OKID	59.53	103.63	110	155.07	205.31	212.38	0.83	**0.77**	0.76	**0.19**	**0.21**	0.22
3D to 2D proj	Rec	Pred	Pred(8e)	Rec	Pred	Pred(8e)	Rec	Pred	Pred(8e)	Rec	Pred	Pred(8e)
DRNET	80.31	136.06	/	172.43	248.11	/	0.77	0.66	/	0.32	0.41	/
SCALOR	**7.58**	233.18	/	**32.05**	377.34	/	**0.96**	0.32	/	**0.02**	0.45	/
DDPAE	20.99	**43.72**	/	57.08	**86.87**	/	0.94	**0.89**	/	0.11	**0.14**	/
OKID	36.32	45.63	49.48	78.27	93.57	101.10	0.89	0.87	0.86	0.17	0.17	0.20

Interpreting $\mathcal{K}$: Circular Motion Experiments. Figure 3 depicts the eigendecomposition of the learned $\mathcal{K}$ for the *Circular Motion Experiments*. We know that the motion is expected to be sinusoidal in both x and y. Therefore, a correct Koopman mapping would not need to be non-linear to capture the dynamics in this case. A sinusoid can be modelled by a linear operator with a complex pair of eigenvectors in the unit circle. We can see that pair, together with real eigenvalues that present no oscillation. We observe also that the learned model is stable as no eigenvalue has greater magnitude than 1. The model has learned a very similar operator $\mathcal{K}$ for both the cropped and the complete version. This indicates that OKID is learning consistent dynamics across datasets when they share the same motion. It also suggests that the model is able to impute a trajectory when the data is missing. Qualitative results can be seen in the technical report Comas et al. (2023).

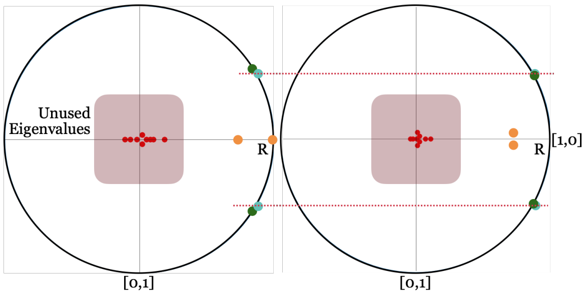

Figure 3: **Eigenvalues of the learned matrix $\mathcal{K}$ in the complex plane** for both *circular* experiments. Left: Complete image, right: Cropped image. The learned eigenvalues are very similar, while one of them has been trained with corrupted data. Yellow points in real axis are associated to non-oscillating dynamic modes. In red, eigenvalues close to 0 have very weak effect on the dynamics.

Interpreting $\mathcal{K}$: 3D to 2D Projection Experiments. Figure 2 shows the qualitative performance in terms of prediction of `OKID` and the result of several interventions. It illustrates how single or conjugated pairs of eigenvalues impact on the dynamics, for the *3D to 2D Projection Experiments*. This dataset is challenging because it entangles linear motion across dimensions by means of projection. It also encompasses digit size variations. We can see that the predictions are accurate and sharp and the learned dynamics are correct. The model correctly disentangles the two objects that appear in the scene, and models their dynamics.

We manipulate the eigenvalues of $\mathcal{K}$ highlighted in green (Fig. 3(a)) by changing their module r or their angle θ. Shadowed in red, we see that this particular eigenvalue pair has higher effect in the latter part of the trajectory. If we increase its module above 1, we observe an increase on the intensity of the variations, leading to strong oscillation at the end of the sequence (see size of the object). This happens because the system is now unstable. If we vary the angle of the eigenvalue pair with respect to the real axis, we see variations in terms of frequency. When we subtract 20 degrees to the angle, it becomes almost 0 resulting into an almost flat trajectory. When we increase that angle, we see digits oscillating with higher frequency.

Inputs: Collision Experiments. For the *Collision Experiments* scenario, the challenge is the use of inputs $\mathbf{u}_t$. Every collision against the frame boundaries applies a force to the object, that modifies its dynamics. Therefore, we model the effect of the environment as in Equation 8, allowing the inputs to be non-zero, and forcing them to be sparse and low-dimensional ($\mathbf{u}_t \in \Re^4$) to avoid overfitting. This generates sharp objects and captures correctly the dynamics.

Additional experiments, including more qualitative samples, single object manipulation and the effect of having more trackers than objects in the scene, can be found in Comas et al. (2023).

6. Conclusion

We propose a self-supervised method for video prediction by means of scene decomposition into objects, their attributes and scene dynamic modes. We embed the dynamic object-centric attributes into a space where dynamics behave linearly, by means of a Koopman embedding. This enables the use of linear algebra techniques to interpret and manipulate the scene dynamics. Through careful experiments we show that decomposition into dynamic modes is indeed possible and carries predictive power. We include examples with inputs and object interactions. We also provide insights into the dynamics of objects (*interpretability*) through the analysis of eigenvalues of the learned Koopman operator. Finally, we show how to manipulate the model to obtain arbitrarily higher temporal resolution, backward prediction, model reduction and alter the dynamic behavior of a targeted object.

Our model has room for improvement. `OKID`'s current implementation does not support background modeling and an upper-bound of objects (number of trackers) N must be defined *a priori*. This is detrimental to the performance, as the number of trackers used has a considerable effect on the iteration time and memory requirements. We intend to address these limitations in the future.

The Technical Report can be found in Comas et al. (2023).

7. Acknowledgements

This work was supported in part by NSF grants IIS-1814631 and CNS–2038493, AFOSR grant FA9550-19-1-0005, and ONR grant N00014-21-1-2431.

References

Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent koopman autoencoders. In *International Conference on Machine Learning (ICML)*, 2020.

S. Brunton, B. W. Brunton, J. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8, 2017.

C. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, I. Higgins, M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *a Computer Research Repository (CoRR)*, 2019.

Armand Comas, Chi Zhang, Zlatan Feric, Octavia Camps, and Rose Yu. Learning disentangled representations of videos with missing data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Armand Comas, Christian Fernandez, Sandesh Ghimire, Haolin Li, Mario Sznaier, and Octavia Camps. Technical report. bit.ly/l4dc-tech-report, 2023.

Emily L Denton and vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning (ICML)*, 2019.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *ArXiv*, abs/2012.05208, 2020.

Z. He, J. Li, Daxue Liu, Hangen He, and D. Barber. Tracking by animation: Unsupervised learning of multi-object attentive trackers. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1318–1327, 2019.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Miguel Jaques, Michael Burke, and Timothy M Hospedales. Newtonianvae: Proportional control and goal identification from pixels via physical latent spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4454–4463, 2021.

Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations (ICLR)*, 2020.

B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences of the United States of America*, 17 5:315–8, 1931.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jannik Kossen, Karl Stelzner, Marcel Hussing, Claas Voelcker, and Kristian Kersting. Structured object-aware physics prediction for video modeling and planning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=B1e-kxSKDH.

Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. In *International Conference on Learning Representations (ICLR)*, 2020.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *ICML*, 2020a.

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 (ICLR)*, 2020b.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Bethany Lusch, J. N. Kutz, and S. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9, 2018.

Andreas Mardt, L. Pasquali, Hao Wu, and F. Noé. Vampnets for deep learning of molecular kinetics. *Nature Communications*, 9, 2017.

I. Mezic. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

Jeremy Morton, Antony Jameson, Mykel J Kochenderfer, and Freddie Witherden. Deep dynamical modeling and control of unsteady fluid flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jeremy Morton, F. Witherden, and Mykel J. Kochenderfer. Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2019.

Samuel E. Otto and Clarence W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846.

Charles L Phillips and H Troy Nagle. *Digital control system analysis and design*. Prentice Hall Press, 2007.

J. Proctor, S. Brunton, and J. N. Kutz. Generalizing koopman theory to allow for inputs and control. *SIAM J. Appl. Dyn. Syst.*, 17:909–930, 2018.

P. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2008.

Nitish Srivastava, Elman Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015.

Qu Tang, Xiangyu Zhu, Zhen Lei, and Zhaoxiang Zhang. Object dynamics distillation for scene decomposition and representation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=oJGDYQFKL3i.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

M. Williams, I. Kevrekidis, and C. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25:1307–1346, 2015.

Yongqian Xiao, Xin Xu, and Qianli Lin. Cknet: A convolutional neural network based on koopman operator for modeling latent dynamics from pixels. *ArXiv*, 2021.

Tian Xie, A. France-Lanord, Yanming Wang, Y. Shao-Horn, and J. Grossman. Graph dynamical networks for unsupervised learning of atomic scale dynamics in materials. *Nature Communications*, 10, 2019.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.