Action-conditioned disentanglement of agent and objects for video prediction in robotic tasks

Manuel Serra Nunes
mserranunes@isr.tecnico.ulisboa.pt

Abstract— **Based on past experience, our brain is capable of anticipating imminent sensory inputs to simplify routine tasks such as reading or driving. Inspired by this notion, studying the prediction of sensory signals may turn out to be an important stepping stone on the path towards intelligent systems.**

In the last five years the effort for materializing this idea by predicting video frames has intensified, yet, current solutions are still hampered by the rapid increase in difficulty that comes with the size of the frames and the prediction horizon. Two ways of mitigating these problems are to disentangle sources of information and to add the future actions of an agent as an extra input, with the later adding the extra benefit of allowing the use of the video prediction model in planning tasks. Naturally, in this type of application the ability of the model to grasp the implications of each action is of critical importance.

With this in consideration, this work can be summarized in the following contributions: (i) we propose a new method that evaluates video prediction models based on their ability to guide action decisions; (ii) we design an autoencoder model that separates agent information from information of objects, using knowledge of future actions and the inherent structure of video data; and (iii) we investigate whether separately predicting object information, conditioned on the actions, can improve the state of the art in video prediction.

Keywords: Video Prediction, Disentangled representations, Robotics, Benchmarking, Predictive Coding, Deep Learning

I. Introduction

A crucial step in problem solving - as we are taught from a young age - is to find a point of view that offers a new, simpler angle on how to approach the task at hand. For example, dividing 210 by 6 is a trivial problem using long division, however, it becomes incredibly more difficult if instead the question is presented as dividing the Roman numerals CCX by VI [1]. Most people would start by finding a better representation for the information and convert the numbers from Roman to Arabic notation. In general, a good representation for available information is one that makes subsequent tasks easier [1].

It turns out that such a notion may be rooted not only in the way we are encouraged to think by our professors but also in human nature. A proposition of especial interest to this thesis is the possibility that the human brain learns representations of the surrounding world in a prediction based fashion. In the past two decades, prediction oriented theories of the human brain have emerged, posing the idea that, in the early stages of visual processing, neurons work to reduce redundancy in the transmitted signals by only relaying information that is not easily predicted.

In particular, the Predictive Coding theory of Rao and Ballard [2] proposes that at each level of processing exist two types of neuron populations: representation neurons that, based on an internal world model, predict activity at the lower level and error neurons, computing the mismatch between the prediction coming from the upper level via feedback connections and the actual neuronal response at that level. Forward connections then convey the prediction error to the upper layer for the world model to be updated, progressively allowing better estimates of future sensory inputs. Under the light of Predictive Coding, a visual scene is perceived when the brain's prediction successfully matches the incoming signal, which may only happen after online corrections to the initial guess.

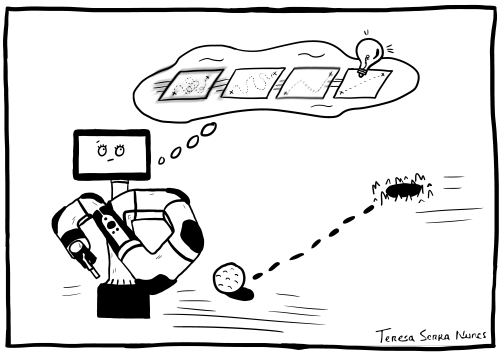

Fig. 1: A sawyer robot imagining different possible outcomes of executing a sequence of actions in the hypothetical task of pushing a ball to a whole.

Nowadays, the ideas of Rao and Ballard have matured into a theory of brain functioning and perception usually referred to as Predictive Processing (PP). This theory supports the idea that brains are predictive engines that are continuously trying to anticipate the structure of the incoming sensory signal based on past experience. For this theory to be feasible, however, an additional wrinkle must be taken into account and that is prediction driven learning. In the earlier stages of their lives, the way humans learn about the world is characterized by observation, experimentation and trial-and-error. These constitute a way of interacting with the world and learning what to expect at any given situation while having in our senses a self-supervisory signal that is free, rich and continuously available for comparison. This allows the optimization of a predictive model which in turn leads to an ever improving grasp of the surrounding world [3]: a baby that is capable of anticipating what is about to happen when her mom opens the fridge or predicting what she will see and hear as a stack of block toys starts leaning in one direction has already collected a lot of knowledge about the world and how to act within it.

Finally, this notion of learning by experimentation, leads to the consideration that sensori prediction must somehow be connected with action decision and the motor cortex, an idea that is core of Karl Friston's Free Energy Principle [4]. Specifically, Friston argues that biological organisms are defined by their natural tendency to resist disorder [5], *i.e.*, by seeking states that allow them to remain within physiological bounds. In doing that, agents associate beneficial states with low entropy (or uncertainty) and work to minimize surprise or, in other words, the likelihood of undesirable states. This interpretation entails that, while in Predictive Processing updating the internal model of the world was the single mechanism available for reducing the error between sensory predictions and

actual observations, action is now regarded as a complementary way of reducing such an error, by selectively acting upon the world in ways that result in desirable observations and bring the internal model's predictions about.

The most straightforward way of translating these ideas to an artificial system is through video prediction as, for a model to be capable of anticipating imminent visual inputs, it must have acquired some internal knowledge of how its world works. Furthermore, when an agent's actions are taken into consideration, it becomes possible for the model to imagine the possible futures that result of the different available actions and to select the most convenient action sequence, as illustrated in Fig. 1. In these applications, the ability to select the best possible action is very dependent on how well the Video Prediction (VP) model can anticipate future observations based on the robot's actions and the current status of the scene. Having a metric that can rank video prediction models based on how well they perform as a forward model is therefore of fundamental importance.

A second question raised by the availability of the actions is whether these, along with the sequential structure of the video, can be used to disentangle (separate) the sources of visual information related with the agent from those related with external objects present in the scene. The importance of this question, along with our solution, is described in chapter IV. Finally, we intend to explore how having disentangled representations of the agent's actions and of the objects in the scene can be beneficial for a video prediction model. In summary, this work has the following contributions:

- we propose a novel action-based quality assessment metric for robotic VP models;
- we apply the metric on several different models and quantitatively rank them from a robotic perspective;
- we propose a new method of disentangling agent and object information for video prediction in robotic tasks;
- we quantitatively compare our approach with other methods both in visual quality and ability to be used in planning tasks.

In addition, we provide the implementation of both our proposed metric[1], for use in VP models that we did not consider, and of our propsed disentangled prediction model[2].

II. Theoretical Background & Related Work

A. Deep learning models for video prediction

Early work in anticipation of visual information includes sensori-motor networks [6], which emulate the interaction between the visual and motor systems in organisms to predict future visual stimulus. In [7], sensori-motor networks are applied to small image patches to predict the next time step's stimulus. However, when the problem is extended to more generic settings involving observations of a complete scene and longer temporal sequences, more complex models become necessary, making deep neural networks the predominant solution for VP in the last few years.

Recently, a significant research effort has been made on finding data efficient methods that can leverage the structure of image data. Convolutional neural networks are a kind of neural network designed to solve problems in which the data is structured in a grid, which has made it an ubiquitous model in image and video frame processing. The name convolutional stems from the fact that these networks use the convolution of the layer input with the weights, instead of the matrix multiplication used in fully connected networks. This modification is not only motivated by the improvement in data efficiency, but also by the notion that most meaningful features in an image - such as edges and corners - are sparse interactions between pixels in the same local region. [8].

However, the fact that the output of a Convolutional Neural Network (CNN) is solely based on the most recent input and not on the history of observed inputs constitutes a limitation in problems like video prediction, in which the information contained in the sequence of frames is crucial for a good performance. In contrast, Recurrent neural networks (RNNs) exploit sequential information by having a recurrent cell that continually receives as input new elements $\boldsymbol{x}_t$ of a sequence. Within this cell, there is an internal feedback loop - the hidden state $\boldsymbol{h}$ - which works as a memory, allowing past information to persist, which makes Recurrent Neural Networkss (RNNs) a crucial tool in video prediction.

A final model widely used in video prediction is the autoencoder, which is an unsupervised approach to learning a low dimensional feature representation of the factors of variation in the training data. In other words, the aim is to obtain a hidden vector whose values specify the characteristics necessary to distinguish examples in the data. The autoencoder approach to solving this problem is based on learning an identity mapping while imposing an information bottleneck at an intermediate representation. With an image as example, we start by using a function f - the encoder - to map the image $\boldsymbol{x}$ to a low dimensional vector $\boldsymbol{h} = f(\boldsymbol{x})$. A second function g - the decoder - takes $\boldsymbol{h}$ as input and maps it to the original high dimensional space of the image, obtaining $\hat{\boldsymbol{x}} = g(\boldsymbol{h})$. If we further impose that the reconstruction $\hat{\boldsymbol{x}}$ should be as similar as possible to the original image $\boldsymbol{x}$, by optimizing $\ell_2(\boldsymbol{x}, \hat{\boldsymbol{x}})$, the small size of $\boldsymbol{h}$ will force the encoder to prioritize conveying the main characteristics describing the data, while the decoder will learn to closely reconstruct the data from those characteristics.

B. Video prediction

In the last 5 years, research in video prediction has intensified, propelled by the advances in deep learning described in the previous section, as well as improved computational resources and the collection of large scale video datasets. Today, VP models can, for the most part, be classified in one of two types of approach: pixel motion models and latent variable models. Additionally, they can be deterministic or stochastic, and can be conditioned on an agent's actions.

The introduction of the concept of pixel motion by Finn *et al*. [9], Xue *et al*. [10] and De Brabandere *et al*. [11], is perhaps one of the most meaningful contributions to VP, as it liberates the system from having to predict every pixel from scratch by instead modelling pixel motion from previous images and applying it to the most recent observation. The main challenge with this approach lies in modeling the motion of the pixels and combining the predictions for each pixel/object into a single predicted image. The advantages are evident, though: by focusing prediction on the moving pixels, the dimensionality of the problem is greatly reduced. Also, since the model is based on the prediction of independent pixel movement and does not rely on any *a priori* physical model of the world, it has the ability to generalize what is learned during training to novel objects, never seen before.

Rather than focusing on this approach, some authors have opted for pixel based approaches that use latent variables to reason about all the information contained in the video [12] [13]. Even though this may seem an unpromising solution to video prediction, different interpretations of the base architecture can lead to improved results over pixel motion models. Latent variable models are implemented

[1] https://github.com/m-serra/action-inference-for-video-prediction-benchmarking

[2] https://github.com/m-serra/adr

with autoencoders and start by obtaining a low dimensional feature representation $\boldsymbol{h}$ for each observed frame. Typically, an RNN is then used to project this representation into the future, taking into account the past sequence of $\boldsymbol{h}$ vectors and finally, for each predicted $\hat{\boldsymbol{h}}$ that the RNN outputs, a decoder network is used to construct a future frame.

This type of architecture poses two main difficulties: on the one hand the encoder is tasked with selecting the essential factors of variation of the scene and not only expressing them using a small set of real numbers, but also in a way that the decoder can interpret. On the other hand - and unlike pixel motion models - the decoder has to directly generate every pixel of the predicted frames. Still, if the model learns to produce good, low dimensional representations of the data, it is possible to obtain crisp predictions for much longer into the future. This is because, as long as the $\boldsymbol{h}$ vectors are consistent in time and the decoder can interpret them, every new frame will be constructed in the same conditions, preventing the accumulation of errors. This is unlike pixel motion models, which rely on the actual pixel values from the most recent frame (whether it was observed or predicted) to construct the new one, meaning that errors can easily compound and lead to blurry predictions.

A problem that is common to these approaches to video prediction is the existence of factors of variation in the environment that are not observable. These include, for example, the spin of a falling ball that makes it bounce in unpredictable directions, the weight of a poked object or, when using a fixed camera angle, the depth of different objects. For regions affected by these aspects, the models should ideally be able to give their best guess and output the most likely pixel values. However, in trying to minimize the ℓ_2 loss for the predicted frame, the VP models have a tendency to output an average of the different possible modes, a result that will never contain the true value of the pixel. To solve this problem, some authors propose the conditioning of the VP model on latent random variables [14] [15]. This way, during training, the models can learn to associate these variables with the unobservable aspects of the environment so that at test time, when the variables are sampled from a prior distribution, a decision for one of the possible outcomes is induced.

C. Video prediction based planning

Action planning is perhaps the most important application of VP models. This is because, by acting upon the environment an agent can observe the inner workings of its world, which in turn elicits better sensory prediction and enables further, more complex interactions. In this type of tasks, VP models serve as forward models, *i.e*., as a function that, given the current state and action, outputs the next state.

This is explored in the work of Ha *et al*. [16], with the introduction of an architecture that learns a policy for solving OpenAI Gym [17] reinforcement learning problems using an encoder of observed video frames and a MDN-RNN to predict future visual codes, given current and past observations and executed actions. In a real world robotic scenario, Finn and Levine [9] use a VP model to continuously sample the expected future given different sequences of actions. The sequence that maximizes the likelihood of the robot achieving the goal of pushing an object to a specified location is selected at each time step to be executed. This type of model-based control is an active area of research in Reinforcement Learning (RL) and large-scale datasets of robotic experiment such as RobotNet [18], should allow future breakthroughs.

III. Evaluating Video Prediction Models

A. The human standpoint

A common trend in VP models is the evaluation of model performance based on metrics designed to mirror human perception of quality in image and video, *i.e*., Quality of Experience (QoE). This is a subjective concept, which depends not only on the data fidelity of the reconstructed image or video but also on the personal experience and expectations of the viewer [19]. The standard measure for QoE is the Mean Opinion Score (MOS) which is the average quality rating, given by a sample of viewers. QoE prediction is an active area of research in which proposed methods are usually compared to the Peak Signal to Noise Ratio (PSNR) benchmark. PSNR is a logarithmic measure of the mean squared error between a distorted image and the original. Its mathematical simplicity and convenient optimization properties make it one of the most popular metrics for image quality [20]. However, PSNR compares images pixel by pixel, not taking into account the content, leading to pathological cases [19] in which it fails at approximating human judgement.

An alternative metric that addresses this problem is the Structural Similarity (SSIM) Index [21], which is founded on the principle that signals that are close in space have strong dependencies between each other and that the human visual system is highly adapted for extracting this structural information. SSIM indices are calculated using a sliding window which produces an index map. This index is 1 if the structure of corresponding patches of the two images is the same and the final SSIM score corresponds to the average of the index map. More recently, Learned Perceptual Image Patch Similarity (LPIPS) metrics, based on learned features of neural networks such as VGG have shown remarkable capabilities as a perceptual distance metric [22].

Inspired by the developments in image generation, methods that are specifically designed for assessing realism in generated video have also been proposed [23]. The Fréchet Video Distance (FVD) [24] accounts for visual quality, temporal coherence, and diversity by measuring the distance between the distribution that originated the observed data and the distribution from which the predicted video is generated, instead of comparing pixels or image patches.

B. The robotic standpoint

Despite all the metrics described in section III-A being widely used in state of the art VP research, we argue that they are not necessarily adequate in action oriented applications such as robotic planning. Instead we propose a new angle on the problem where the key idea is that the quality of the VP model should be measured by how well it can guide an agent in deciding its actions from the predicted frames.

We start by assuming that the better the dynamics representation of the agent is at encoding action features, the better it will be for planning actions based on the expected outcome. Under this assumption, the problem turns into evaluating how well a VP model is encoding action features and assigning it a score based on such evaluation.

With this in mind, we hypothesise that the ability to observe a sequence of predicted frames and infer the executed actions should be an indication that the VP model is correctly encoding action features. To better illustrate this idea, first consider a failure case: if the VP model generates a sequence of predicted frames that do not correspond to the actions executed by the robot, then no action inference model can recognize the correct set of actions from the

predicted images, resulting in a low action inference score. On the other hand, if the VP model understands the consequences of the input actions, then the frames it predicts should correctly reflect the action and its consequences, allowing an inference model to recognize the actual executed actions and attain a high score.

To assess the quality of models, we first train a simple convolutional neural network to infer the actions executed between every two frames using predicted videos. The actions are assumed to be continuous and multidimensional, to be representative of most robotic control action-spaces. Each pair of frames is concatenated along the channels dimension and given to the network as input, as illustrated in fig. 2. Because action dynamics should not change over time, model parameters are shared across all time steps of a sequence. While a RNN would typically have been useful for learning the sequence of executed actions, we choose to input a window of two frames at a time, cutting off any temporal correlation between actions. This forces the inference model to identify actions from the frames instead of focusing on learning the temporal action distribution. The option for a window size of two frames is due to the fact that in the selected dataset the robot's actions are randomly updated every two frames. For datasets with different conditions, however, the window size parameter can control the temporal information received by the network without shifting the attention of the model from the frames, and it is expected that bigger windows should result in better action inference.

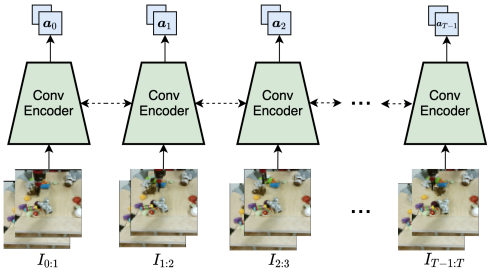

Fig. 2: Action inference network. At each step the network receives a pair of frames and outputs a recognized action.

IV. Disentangling Agent and Objects for Video Prediction

The notion of learning representations of the data that facilitate the realization of subsequent tasks is central in deep learning and video prediction. Autoencoders, for example, learn to discover and represents the underlying factors generating the data. Typically, these factors of variation are described as being a set of sources interacting in complex ways. Ideally, these sources could be *disentangled* and varied independently of one another to produce a change in the associated characteristics of the observed data [25]. In practice however, obtaining a representation in which the underlying causes of the data are factorized in this fashion is a difficult task. Nonetheless, an independent understanding of each factor of variation in the data translates in a significantly easier solution to any subsequent task, such as classification or prediction [26], making this one of the most important problems in deep learning. In particular, having separate representations for different sources of information allows the processing of each one independently and the prediction of only the most relevant parts of the scene, which may represent a solution for the dimensionality problem of video prediction.

A. Disentangling agent movement

With the intent of separating visual information that is part of an agent from the content and movable objects in the scene, we ask the question: *Can the movement of a robot be reconstructed up to time step $t+k$ from the content information observed at time step t and the actions commanded to the robot between t and $t+k$?*

Solving this challenge requires a model that is capable of both identifying which parts of an image are related to the robot and of transforming that information according to the specified action. Inspired by previous work on how to disentangle content information from pose (motion) in videos [27], we propose a solution, which starts by setting the constraint that, for the most part, the content of a video remains the same during the a video sequence, but can change from video to video. Hence, we adopt a loss term that encourages the content representation $\boldsymbol{h}_c$ to be the same when built from frames of the same video, which has the effect of making it time-invariant:

$$\mathcal{L}_{similarity}(E_c) = ||E_c(\boldsymbol{x}^{t-c:t}) - E_c(\boldsymbol{x}^{t+m-c:t+m})||_2^2, \quad (1)$$

where E_c represents the content encoder, $\boldsymbol{x}^t$ the video frame at time step t and m is a time gap chosen at random. We consider time step t the present moment while $t-c$ is the first frame in the video, with c being the number of past frames given to the content encoder as context.

While in previous work [27] movement requires an adversarial loss term to remove any content information, we have a simplified task at this stage because we're only concerned with the agent's own movement, which is determined by the commanded actions. Using actions has the benefit of content information being separated from movement information by default. So, to answer the initial question only one more loss term is needed, imposing that the model should be able to reconstruct the future frame $\boldsymbol{x}^{t+k}$ from the content representation at time t and the action representation

$$\mathcal{L}_{reconstruction}(E_c, A, D_a) = ||D_a(\boldsymbol{h}_c^{t-c:t}, \boldsymbol{h}_a^{t+k}) - \boldsymbol{x}^{t+k}||_2^2, \quad (2)$$

with k being another random time gap, $\boldsymbol{h}_a^{t+k}$ being the output of the action encoder $\boldsymbol{h}_a^{t+k} = A(\boldsymbol{a}^{t-c:t+k})$ and D_a being the decoder that generates the predicted frame $\hat{\boldsymbol{x}}^{t+k}$ from the content and action representations. The complete training objective is then

$$\mathcal{L} = \mathcal{L}_{reconstruction}(E_c, A, D_a) + \alpha \mathcal{L}_{similarity}(E_c) \quad (3)$$

and we refer to this model as Action-conditioned Disentangled Representations - Agent Only (ADR-AO).

As previously discussed, real world environments are characterized by complex dynamics that often depend on unobservable and unpredictable factors. A patent example observed in ADR-AO is concerned with the difficulty of perceiving depth from a single $2D$ observation of the scene, often leading to blurry reconstructions, as illustrated in Fig. 4. To solve this issue, we extend the model to learn a set of latent variables $\boldsymbol{z}_a$ that the model can associate with the unobservable factors, allowing it to decide on one of the possible outcomes for these factors. This is achieved with the help of a distribution $q_\phi(\boldsymbol{z}_a^t | \boldsymbol{a}^{0:t+k}, \boldsymbol{x}^t)$, that is trained both with a reconstruction term and with the regularizing constraint that it should remain close to a prior distribution $p(\boldsymbol{z}_a)$ chosen to be a Gaussian $\mathcal{N}(0, \boldsymbol{I})$. The final extended objective function becomes

$$\begin{aligned}\mathcal{L} =& \mathcal{L}_{reconstruction}(E_c, A, D_a) + \alpha \mathcal{L}_{similarity}(E_c) \\ &+ \beta \mathrm{D_a}_{KL}(q_\phi(\boldsymbol{z}_a | \boldsymbol{a}^{0:t+k}) || p(\boldsymbol{z}_a)).\end{aligned} \quad (4)$$

To condition the on the sequence of executed actions, we use an LSTM that at each time step receives the content representation and

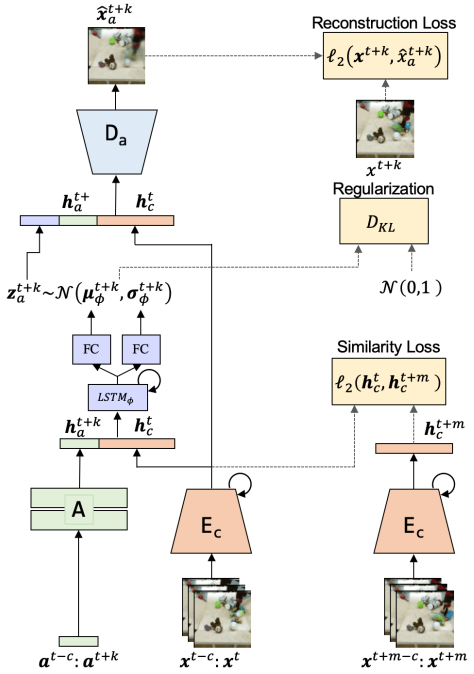

Fig. 3: Stochastic ADR-AO.

the current time step's action representation $\boldsymbol{h}_a^{t+k}$ and outputs the parameters of a multivariate Gaussian distribution from which $\boldsymbol{z}_a$ is sampled. This extended architecture is summarized in Fig. 3.

It is worth noting that unlike common practice in stochastic VP models [15] [14], where at test time the latent variables must be sampled from the prior $p(\boldsymbol{z}_a)$, in our model the approximate posterior distribution $q_\phi(\boldsymbol{z}_a^t | \boldsymbol{a}^{0:t+k}, \boldsymbol{x}^t)$ can be used at test time, as the LSTM is only conditioned on the future actions, that are known. This allows our model to use the latent variables to model hidden factors of variation without creating random futures that diverge from the truth, as is the case in [15].

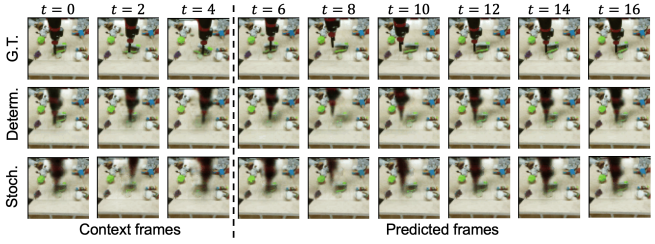

Fig. 4: To minimize the error of predicting whether the arm passes in front or behind the green stapler, the deterministic model averages the two modes, resulting in a blurry prediction. On the other hand, the stochastic model can make a decision on the most likely mode, placing the arm in front of the stapler and preserving most of the stapler's shape. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/stochasticity_experiment/`.

B. Disentangling object information

Inspired by the predictive views of the human brain described in section I, we argue that a VP system used in a robotic context should focus on the external, difficult to predict, consequences of the robot's actions, rather than on the foreseeable self movements of the robot. This shift in attention can be made possible with the help of the ADR-AO model proposed in the previous section. Because ADR-AO is trained to only be aware of the planned motion of the agent, its predictions show objects that end up getting displaced in the ground truth video being ignored and left in their original position (see Fig. 5). This means that the error between ground truth frames $\boldsymbol{x}^t$ and frames generated with ADR-AO $\hat{\boldsymbol{x}}_a^t$, *i.e.*,

$$\boldsymbol{x}_{err}^t = \boldsymbol{x}^t - \hat{\boldsymbol{x}}_a^t, \tag{5}$$

produces an image dominated by information of the objects that moved during the video, while the background and the movement of the agent - which have already been predicted - get suppressed, as illustrated in Fig. 5. In practice, this cue on the object information can then be leveraged by a model to learn a representation for the objects that ADR-AO fails to perceive, in a mechanism that draws parallelism with the forward propagation of yet unexplained error in Predictive Coding. Importantly, this information is obtained in a self-supervised way, without the need for data pre-processing or human annotation to obtain object locations and identities, as is common in most state of the art work [28] [29].

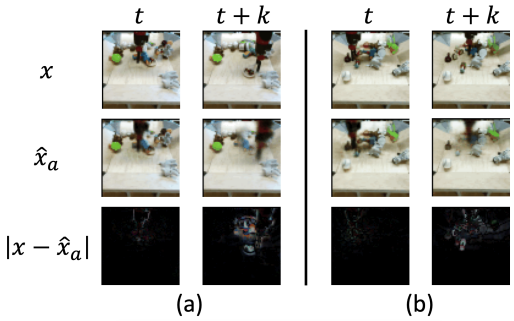

Fig. 5: Top row: ground truth frames. Middle row: predictions with ADR-AO, where objects are left in their initial position. Bottom row: the error between the two rows is dominated by object information. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/error_images/`.

Having a separate cue on object information, we can now train an autoencoder to obtain a low dimensional representation of the objects $\boldsymbol{h}_o$. We aim at reconstructing a video frame at time step $t+k$ from the content information of c context frames $\boldsymbol{x}_{t-c:t}$ and a sequence of actions $\boldsymbol{a}_{t-c:t+k}$ to be performed by the agent. While the goal might be the same, the approach differs in the task of the autoencoder: whereas for ADR-AO the decoder reconstructs its best approximation of $\boldsymbol{x}_{t+k}$, to learn the feature representation of the objects the new autoencoder is now trained to reconstruct the error $\boldsymbol{x}_{err}$. As such, the future frame $\boldsymbol{x}_{t+k}$ is also needed as an extra input, to obtain the error image. If, ideally, the reconstruction $\hat{\boldsymbol{x}}_{err}$ is sufficiently close to the original error image, the future frame $\hat{\boldsymbol{x}}_{pred}^{t+k}$ can be obtained as

$$\hat{\boldsymbol{x}}_{pred}^{t+k} = \hat{\boldsymbol{x}}_a^{t+k} + \hat{\boldsymbol{x}}_{err}^{t+k}. \tag{6}$$

In practice, however, we verify that better results are obtained if $\boldsymbol{x}_{err}$ is split into its positive and negative components with each one being reconstructed separately:

$$\boldsymbol{x}_{err} = \boldsymbol{x}_{err+} - \boldsymbol{x}_{err-} = max\{0, \boldsymbol{x}^t - \hat{\boldsymbol{x}}_a^t\} - max\{0, \hat{\boldsymbol{x}}_a^t - \boldsymbol{x}^t\}. \tag{7}$$

Because at this stage ADR-AO is considered to have been pre-trained, we reuse its action A and content E_c encoders to obtain

low dimensional feature representations of the actions $\boldsymbol{h}_a$ and of the content $\boldsymbol{h}_c$, which aid the decoder D_o in the reconstruction of the error image. Yet, the trainable parameters of both E_c and A are frozen, and only E_o and D_o are trained. The model of Fig. 6 - which we call ADR - is trained with three different reconstruction loss terms. The first term determines that the obtained frame $\hat{\boldsymbol{x}}_{pred}$ should be as close as possible to the corresponding ground truth frame, in terms of ℓ_2 distance. The second and third loss terms are concerned with the positive and negative components of the error reconstruction and encourage their ℓ_2 distance to the input error images to be as small as possible. The need for these two extra loss terms was verified empirically, as they encourage sparsity in the decoder's output and provide an additional cue for the reconstruction, preventing the model from overfitting to the training data. The complete loss function is presented in equation (8):

$$\begin{aligned}\mathcal{L}(E_o, D_o) =& ||\boldsymbol{x}^{t+k} - \hat{\boldsymbol{x}}_{pred}^{t+k})||_2^2 + \alpha(||\boldsymbol{x}_{err+}^{t+k} - \hat{\boldsymbol{x}}_{err+}^{t+k})||_2^2 \\ &+ ||\boldsymbol{x}_{err-}^{t+k} - \hat{\boldsymbol{x}}_{err-}^{t+k})||_2^2)\end{aligned} \tag{8}$$

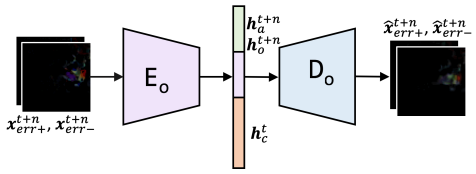

Fig. 6: ADR architecture with action and content encoders omitted for simplicity.

C. Disentangled video prediction

While the first stage of the model, ADR-AO, only requires knowledge of the ground truth frames at training time, for computing the loss function, the complete model, ADR, also makes use of the ground truth for obtaining the error images that serve as input to the E_o module. While this poses no problem when the task is limited to reconstruction, in a practical video prediction problem the ground truth frames would not be available, meaning that only ADR-AO could be directly used for video prediction.

To predict video using ADR, a new module that can acquire knowledge of the temporal evolution of the object representations $\boldsymbol{h}_o$ needs to be added. For that, we opt for the use of an LSTM, as illustrated in the diagram of Fig. 7. At each time step t, the LSTM receives a complete representation of the current frame, which is the concatenation of the content, action and object representations, and is asked to output the object representation for the next time step $(t+1)$. During the temporal period corresponding to the context frames, the $\boldsymbol{h}_o$ vector received by the LSTM can be obtained from the observed frames, allowing the initialization of the hidden state. However, once the model runs out of context frames, the predicted vectors $\hat{\boldsymbol{h}}_o$ must be fed back to the LSTM, instead of vectors coming from E_o, effectively allowing it to unroll imagined futures. Finally, to obtain the predicted error images that allow the construction of $\hat{\boldsymbol{x}}_{pred}$, D_o is used. We refer to this updated framework as Action-conditioned Disentangled Representations - Video Prediction (ADR-VP).

An important benefit of having disentangled sources of information is highlighted by this architecture: because content vectors are trained to be constant in time and action representations are obtained from commanded actions to which the model has access (even future ones), the LSTM can focus on only predicting information related to the objects, reducing the complexity of its task. Finally,

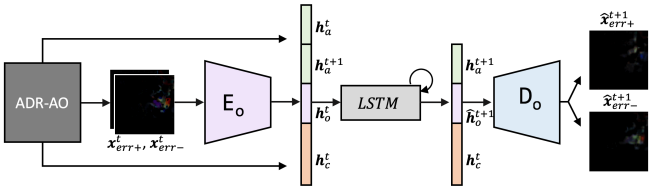

Fig. 7: Video prediction with ADR.

it is important to note that to predict the object representation for the next time step, the LSTM should receive both the current and the next action vectors as input. This allows the LSTM to predict the movement of the objects conditioned on a known imminent movement of the agent, whereas otherwise it would have to guess object movement based on an unknown trajectory of the arm.

Despite having tested different solutions for training ADR-VP, inluding teacher forcing [1] and scheduled sampling [30], we were not able to train the model to a point in which it successfully predicts object movement. There may be several factors contributing to this result, in particular the dataset and the architecture. BAIRs robot push dataset has the particularity of the robotic arm moving randomly around the container, which possibly impacts the training of our model. This type of movement causes video sequences in the dataset to often have no interaction between the robotic arm and the objects contained in the box. Furthermore, even in sequences in which objects are displaced, the movement is typically only seen in a subset of the frames, with other frames showing no object movement. This implies an imbalance in the dataset in terms of the number of frames in which objects move, versus the number of frames in which they are static, possibly leading the network to learn that predicting no movement minimizes the expected loss. In comparison, our chosen baseline, DRNET, which also trains an LSTM for video prediction with teacher forcing, is tested on the KTH dataset and succeeds in predicting video frames. This dataset, however, is composed of videos of human actions such as walking and waving the arms up and down, where movement is consistent and present from frame to frame, which possibly avoids the aforementioned issue.

The other effect of the random actions of BAIRs robotic arm is unpredictability. Even though the uncertainty in the arm's trajectory is offset by the knowledge of the actions that will be commanded in the future, this kind of movement may still have some reflection in the consistency of the movement of the objects. In particular, because the arm's trajectory can change from frame to frame, so can the objects' trajectory, cutting of the temporal correlation that would allow the LSTM to produce good predictions.

Still, there have been video prediction models tested on BAIR dataset, such as CDNA, Stochastic Variational Video Prediction (SV2P) and Stochastic Video Generation with Learned Prior (SVG-LP). However, as demonstrated in the results chapter, quantitative and qualitative results obtained with these models show that the only one that successfully predicts object movement is SVG-LP, with the others simply blurring them out. Even so, SVG-LP is a stochastic action-free model, leading most of its predictions to diverge from the actual observed future. In contrast, the fact that ADR-VP is conditioned on the actions, constrains it to only predict futures that are close to the truth, which possibly results in increased difficulty during training.

Finally, the reason for the poor predictive capabilities of ADR-VP may be related with its architecture and the complexity of the the 3

stage training procedure. This not only makes the training process difficult, due to increased run times and number of parameters to be tuned but also increases the chances of error accumulation from on stage to the other. Another factor introducing complexity in the LSTM's is the dimensionality of $\boldsymbol{h}_o$. While DRNET uses the same scheme to predict $\boldsymbol{h}_p$ vectors of size 24, we set the d$|\boldsymbol{h}_o|$ to 128. This represents a considerable increase in complexity which we try to account for by increasing the number of trainable parameters in the LSTM.

V. Experiments and Results

A. Experimental setup

We conduct our experiments using the BAIR robot push dataset [31] which consists of a robotic arm pushing a collection of objects on a table. Videos are 30 frames long, with 64×64 images. The dataset also provides the commanded action sequences, a 4-dimensional array representing the joint velocities and whether the gripper is open or closed, and a 3-dimensional array representing the Cartesian coordinates of the gripper. A characteristic of the BAIR which has a particular effect on the results is the fact that joint velocities are only updated every two frames. In practice, this alternating nature results in the action inference network not experiencing all types of actions the same way, therefore becoming better fit to some situations than others. For this reason, action inference results are presented separately for odd and even frames.

In our experiments we focus on action-conditioned video prediction models. We select 1) CDNA: a deterministic model based on pixel-motion modelling [32], 2) SAVP: which also models pixel motion but introduces variational and adversarial terms to the loss, to try to improve prediction quality and account for the variability in the environment [33], 3) a variant of SAVP in which the adversarial term is suppressed, 4) SV2P: an extension of CDNA conditioned on stochastic variables [14], 5) and finally we test SVG-LP: the stochastic, action-free model of [15]. All models were pre-trained by the respective authors with exception of CDNA. At training time, they receive 2 context frames and actions (with the exception of SVG-LP which is action-free) and predict video up to time step 12.

B. Evaluating video prediction from a robotic standpoint

To test whether our proposed action inference metric provides new insights that are useful for selecting the best video prediction model to be used in a planning task, we start by doing a forward pass over the entire training set and saving the generated predictions, for each of the models being tested. While the models were trained to predict until time step 12, they are now asked to generalize until step 30.

Having a dataset of predictions for each VP model, the action inference network is trained on the 28 frame long predictions (the 2 context frames are not considered). In our experiments, we define the actions being inferred as the displacements Δx and Δy of the robot's gripper along the x and y axis, between every two time steps. The ground truth targets for the actions are directly extracted from the BAIR dataset gripper state sequences by subtracting consecutive temporal positions for both axis. This results in an action sequence of length 27 for each 28-frame predicted video.

We start by evaluating the selected group of VP models on some of the traditionally used metrics described in section III-A and on the recently proposed FVD. As opposed to the methodology adopted by some of the previous work [33], [15] in which 100 possible futures are sampled and the best score of the group is reported, we choose to sample a single time, in order to better approximate the conditions of a robot planning actions. This approach has especial impact on action-free models like SVG-LP, that are exposed to greater uncertainty. Regarding the action-conditioned models, the results displayed in Fig. 8 are in line with previous reports, indicating that models have better performance when conditioned on both actions and stochastic variables, as is the case with SAVP-VAE and SV2P. On the other hand, the addition of an adversarial loss term seems to affect performance negatively, which reflects on SAVP having a lower PSNR/SSIM score than a deterministic model like CDNA despite the high visual appeal of the predicted frames. The FVD values for the test set predictions are presented in Table I.

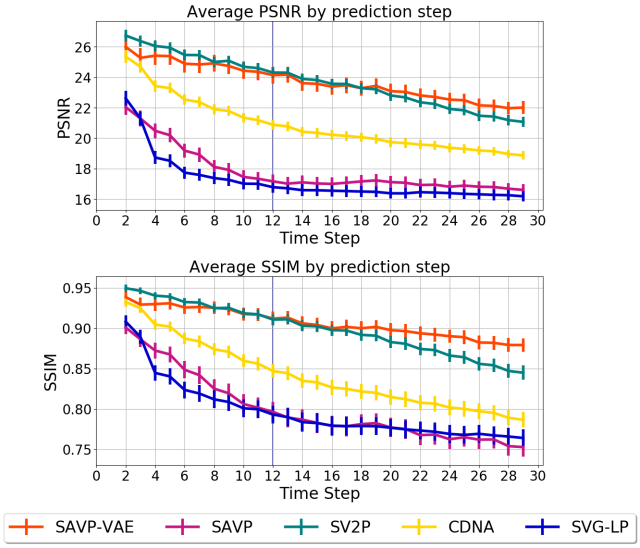

Fig. 8: Average PSNR and SSIM over the test set with 95% confidence interval. Results were reproduced with modification from [32], [15], [33].

For each VP model's predictions dataset, the action inference model that produces the best validation score during training is selected. To measure how well it can identify the executed actions, we compute the Mean Absolute Error (MAE) between inferred and ground truth actions. In our experiments MAE is computed along the 256 test examples for each time step and the evolution of the metric over time is reported in Fig. 9. The most immediate characteristic in the temporal evolution of action inference that arises from an initial analysis of Fig. 9 is that the temporal downgrade artefact in performance observed in PSNR and SSIM is not manifested in the capacity of the model to recognize the actions. This quality of the metric stems from the fact that the parameters of the inference model are shared across all time steps, a choice based on the fact that action dynamics do not change over time and therefore VP models should have a consistent action encoding for all time steps. For this reason, a VP model that encodes actions in a consistent manner should allow the inference network to better learn how to recognize actions and will therefore display stable MAE values across time, as is verified for SAVP and SAVP-VAE. On the other hand, because video predictions made by CDNA have changing dynamics, starting with good resolution and transitioning to blurry images as time advances, it is difficult for the action inference model to learn to identify actions.

The performance of the action inference on predictions made by different models indicates, based on Fig. 9 and on Table I, that the model that is better encoding action features and would therefore

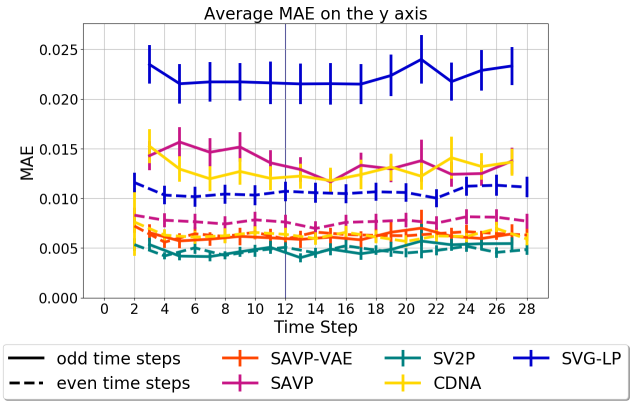

Fig. 9: Average MAE results with 95% confidence interval for predictions made by different VP models in the y axis. Similar results are obtained for the x axis. Odd and even time steps are shown separately.

be the most suited in robotic planning problems is SV2P, closely followed by SAVP-VAE, implying that conditioning on stochastic variables is beneficial but the introduction of the adversarial loss for better image quality removes attention from optimal encoding of action features. These models even outperform the ground truth oracle, supporting the argument that the stochastic variables should be accounting for non observable aspects of the scene and that some blurring of the background may actually help the inference network focus on the action features. On the other hand, the action-free SVG-LP model has an MAE value of 0.163 which corresponds to the variance of the data. This indicates, that the inference model is unable to identify the actions and limits itself to predicting a constant average. The origin of this result is that an action-free stochastic model from which a single prediction is sampled, may produce a future that is different from the ground truth, causing recognized actions to not match the targets and preventing the model from learning a meaningful mapping during training.

In general, and as reported by [24], PSNR and SSIM present a very high correlation as both of them are based on frame by frame comparisons with the original data. Furthermore, because most VP models use an ℓ_2 term in the loss function, these are biased metrics. We also verify that multiple ranking changes occur between our proposed score and FVD, including SV2P scoring the best in action recognition while having an FVD value close to that of SVG-LP, which for being action-free has the lowest score under our metric. These results show that the ability to recognize actions from predicted images doesn't necessarily correlate with previously proposed metrics and that action inference may offer a valuable perspective for choosing the best model in a planning scenario.

C. Disentangling agent and object information

In analysing how the proposed model fares in natural scenarios like the one of BAIR dataset, we start by addressing its disentanglement capabilities. Here, the intention is to understand if ADR-AO is capable of distinguishing the body of the agent (in this case the robotic arm) from other bodies in the environment. Knowing that the learned representations should be (1) informative about the reconstructed frame in the sense that it allows a correct recovery of the content and agent pose, (2) flexible enough to sample different possible (and possibly unseen) scenarios in a planning task, and (3) generalizable to new domains with a short adaptation period. Different experiments were implemented to verify these conditions.

We start by evaluating the reconstruction capabilities of ADR-AO. Because the intention is to learn representations for self information and content, an ideal result for BAIR dataset would be for these to allow the reconstruction of the future frame $\boldsymbol{x}^{t+k}$ with the robotic arm correctly positioned and all the objects in their original pose. Figure 10 shows two predicted sequences obtained using ADR-AO, where 5 frames were given as input and the next 15 predicted. The model succeeds in isolating visual information that is part of the agent (in this case the robotic arm), as demonstrated by its ability to correctly translate the arm, despite some loss in sharpness. Furthermore, the fact that the predicted movement matches the ground truth trajectory of the arm means that the model is successfully learning a mapping between commanded actions and the resulting arm position in the visual field. We also verify that the limited information contained in the inputs prevents the model from learning to predict the movement of the objects, as intended. It nevertheless predicts the existence of some change in the pixels that are part of these object, resulting in some blurriness.

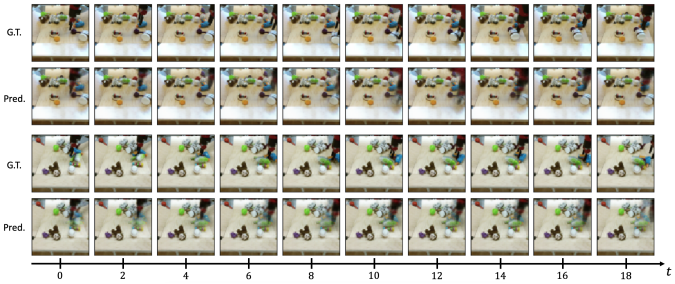

Fig. 10: ADR-AO predictions on BAIR dataset. While the movement of the arm is well predicted, displaced objects are left in their original position. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/adr_ao`.

With the intent of the designed model being its use in robotic planning tasks where different possible action sequences and their corresponding future can be sampled, it is important to determine whether the proposed model is capable of generating previously unseen agent movements. To do so, we train ADR-AO using only the Cartesian position of the gripper as input actions, allowing artificial and previously unseen arm movements to be handcrafted by setting a sequence of positions where the arm should move to. This experiment is illustrated in Fig. 11), where the original actions of a video sequence of the dataset are replaced by a new, infinity shaped trajectory. The model is able to generate a future that matches the new trajectory while maintaining the original background. This confirms that, if the object movement can later be modelled, ADR-AO may be used for planning tasks.

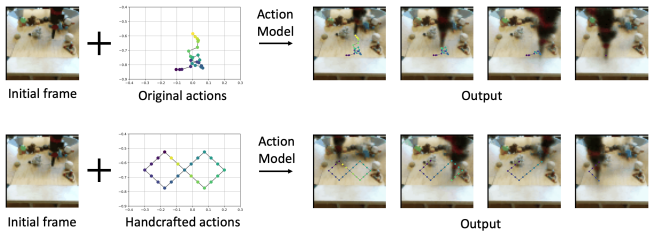

Fig. 11: Prediction with handcrafted action sequence. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/disent_self_movement/`

Finally, we investigate whether the representations learned on

BAIR dataset are generalizable to new environments, given some short adaptation. For that, ADR-AO pre-trained on BAIR dataset is fine-tuned on 800 examples of the Google Push Dataset, which despite constituting a related scenario, has a different robot, objects, camera angle and reference for the joint angles and gripper position. As stated in the literature [34], disentangled representations should facilitate tasks such as transfer and few-shot learning, so a good performance on a new dataset with such few adaptation examples should be a positive indication of the content and self information sources being separated. The ability to adapt to the Google Push Dataset is illustrated in Fig. 12. From a first glance at the sequence it is apparent that the model succeeds in reconstructing the background with remarkable detail, despite the limited number of observations of the environment and the objects it contains. On the other hand, the visual information of the robotic arm is well predicted in terms of trajectory but somewhat blurred.

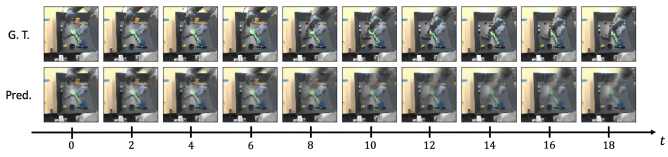

Fig. 12: Generalization to Google Push Dataset. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/google_push/`

Having modelled the movement of the agent we now turn to the objects in the scene. As described in section IV-B, the separate object representation is obtained as a consequence of ADR-AOs prediction error, which we consider to be dominated by object information, therefore requiring no further steps for disentanglement. As such we restrict the analysis to a qualitative assessment of the reconstruction ability of ADR.

In general, we observe that the model succeeds in capturing the object information conveyed in the error signal, enough for the true movement of the objects to be perceived in a frame by frame reconstruction. This is illustrated by the example of Fig. 13, where a group of objects is pushed and their movement, which was absent in the output of ADR-AO, is now apparent. There is however a significant amount of blurriness in the moved objects, which is likely introduced by ADR-AO as an attempt to minimize the reconstruction loss in areas around objects, which is supported by the example. Another aspect worth highlighting is how ADR can sometimes show some difficulty in modelling the negative change of the object, *i.e.*, in removing the object from its initial position. Additional examples can be seen online in the *url* of Fig. 13.

These results demonstrate that the broad movement of the objects is well reconstructed, which represents a first step towards object information being well modelled - one of the main faults of existing video prediction models. Nevertheless, the aforementioned challenges show that there is still room for improvement and better object reconstruction should result in increased consistency in the $\boldsymbol{h}_o$ vector, which in turn should facilitate the training of ADR-VP.

D. Video Prediction

To evaluate the video prediction capabilities of ADR we test the model both on visual quality metrics (PSNR, SSIM and FVD) and in our proposed action inference metric. We now opt to present only the results for three best models previously found (CDNA, SAVP-VAE and SV2P). Additionally, we show the performance of both the deterministic and stochastic versions of ADR-AO and of ADR trained on top of the stochastic ADR-AO.

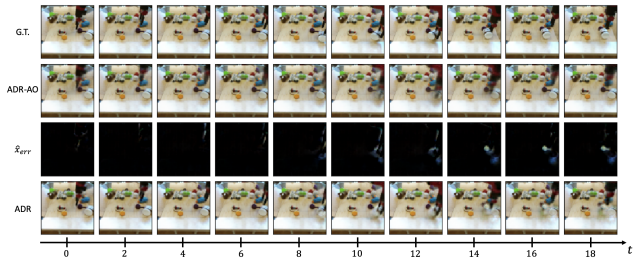

Fig. 13: Reconstruction with ADR. Better seen online at `https://web.ist.utl.pt/ist181063/vp_examples/adr`.

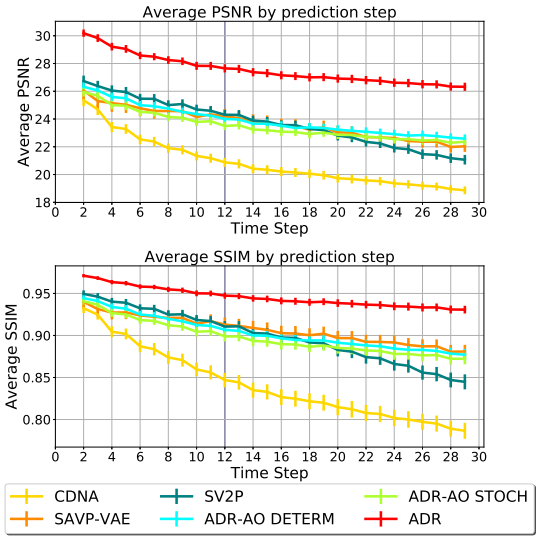

Fig. 14: Model comparison in terms of PSNR and SSIM. The vertical blue line indicates the training horizon.

Starting with an analysis of the models' performance in terms of PSNR and SSIM, presented in Fig. 14, there are two aspects to highlight. The first one is that both the deterministic and stochastic versions of ADR-AO obtain a performance that matches the performance of the two best VP models available in the literature. On another note, we verify that the deterministic version of ADR-AO marginally outperforms the stochastic version. Because both of these metrics are very correlated with the ℓ_2 loss, this is a result that was expected as, during training, the stochastic version has an extra loss term to optimize for, removing focus from the ℓ_2 reconstruction term. On the other hand, the deterministic ADR-AO is allowed to minimize the reconstruction loss as much as it can, even if - as seen in Fig. 4 - this results in objects disappearing. It is important to consider that unlike the other VP models, ADR-AO is explicitly designed not to consider object information, making these two models the floor of the performance that our complete model can achieve. Furthermore, the fact that in spite of this characteristic, ADR-AO still achieves state of the art results is a confirmation that existing VP models fail at conveying the future of object information.

The second aspect of Fig. 14 worth being highlighted is the added performance that comes with considering object information, as shown by ADR. Even though ADR is not in practice a VP model, as it requires access to ground truth video frames, it still provides proof of the importance of modelling object movement and represents a ceiling on the performance that ADR-VP may achieve. As such,

results for a working ADR-VP should be between ADR-AO and ADR, prospecting that further work on ADR-VP can lead to an improvement of the state of the art in video prediction.

As a second way of measuring model performance, we test how well the predicted frames allow an action inference model to recognize the agent's actions, with the results measured in terms of MAE and presented in Fig. 15 and in table I.

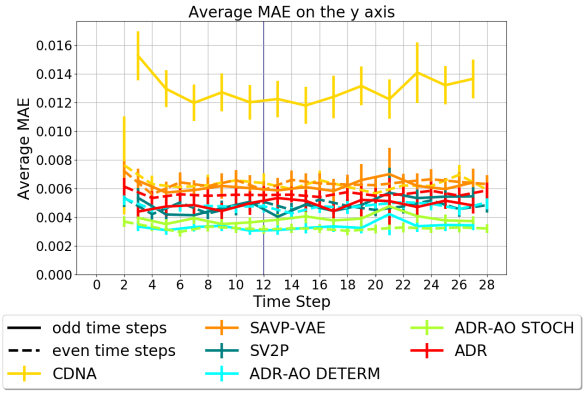

Fig. 15: Model comparison in terms of action inference for the y axis. Results for the x axis are similar. The vertical blue line indicates the training horizon.

The results indicate that the best action recognition was achieved in frames predicted by the stochastic version of ADR-AO, closely followed by the deterministic ADR-AO and SV2P, with the separation between models being better perceived in the values of table I. Interestingly, while in PSNR and SSIM the stochastic ADR-AO model did worse than the deterministic, it is now the opposite. This is a result that further demonstrates that simply minimizing the reconstruction error is not beneficial for downstream tasks such as classification and planning and that a prediction that is plausible is preferable to one that blurs out uncertain aspects of the scene.

A final aspect to take into consideration is how ADR predictions don't allow a better performance of the action inference model, when compared to the agent only versions. This suggests that keeping the background static while moving the arm is helpful for the inference model. A possible explanation for this result is that, because the model only has to correctly recognize the movement of the arm, the removal of other dynamic information allows it to focus on its task. This idea is further supported by the oracle results (where the action inference model was run on ground truth frames), which are both worse than the results achieved by some of the VP models and very close to those achieved by ADR.

TABLE I: FVD and MAE values for each VP model.

Model	FVD Value	MAE Value
CDNA	943.5	0.0111
SAVP	738.3	0.0092
SAVP-VAE	**409.8**	0.0056
SV2P	691.1	0.0049
SVG-LP	728.2	0.0160
ADR-AO Determ.	793.0	0.0049
ADR-AO Stoch.	857.2	**0.0046**
ADR	801.1	0.0057
Oracle	0.0	0.0058

VI. Conclusions and Future Work

In this work we started by proposing a solution for the problem of assessing video prediction models from the standpoint of a robot planning actions. We demonstrated the limitations of evaluating video prediction models solely based on their ability to generate realistic looking frames, instead, arguing that the choice of video prediction models used in planning tasks should be based on complementary metrics, so that both visual quality and the ability of the model to understand the consequences of the robot's actions are considered. A shortcoming of our action-inference metric is that it only takes into account the actions of the robot and should therefore be extended to also consider the objects in the scene. Doing that may require the introduction of better datasets that include states of the environment other than gripper position, in particular objects' positions and speeds.

In our second contribution, we tested whether the sequential structure of video and the agent's actions can be used to disentangle sources of information in video. Our proposed solution takes advantage of these inductive biases to separate visual information related with the agent, from information of the objects present in the scene. In the future it would be interesting to consider new scenarios that include a moving camera and third person agent's who can generate movement that is uncorrelated with the agent.

Finally, we explored the possibility of improving state of the art in video prediction by building a model based on the previously obtained representations. In our approach, we try to shift attention from the agent's own movement, which is foreseeable, to the movement of the objects, which has greater uncertainty and is more difficult to model. We include two video prediction models: ADR-AO and ADR-VP. ADR-AO manages to match, and in some cases improve, the state of the art while explicitly disregarding object information. Such a result is a confirmation of the inability of existing models to predict object dynamics and highlights the importance of our proposal to focus on object information. However, we failed to train ADR-VP to the point where it succeeds in object prediction. The most immediate work direction is therefore fixing ADR-VP, possibly with a simpler architecture, trainable end-to-end and testing it on more diverse large scale datasets such as RoboNet [18]. The proposed models should then be tested on planning tasks similar to the one proposed in [9], and compared to alternative video prediction models. This would allow both the study of the correlation between our action-inference metric with the success rate in planning tasks, and to determine whether our proposed solution for video prediction is in fact better suited for planning tasks.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[3] A. Clark, *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford University Press, 2015.

[4] K. Friston, J. Kilner, and L. Harrison, "A free energy principle for the brain," *Journal of Physiology-Paris*, vol. 100, no. 1-3, pp. 70–87, 2006.

[5] K. Friston, "The free-energy principle: a unified brain theory?" *Nature reviews neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.

[6] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," *Nature Reviews Neuroscience*, vol. 12, no. 12, p. 739, 2011.

[7] R. Santos, R. Ferreira, Â. Cardoso, and A. Bernardino, "Sensorimotor networks vs neural networks for visual stimulus prediction," in *4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, 2014, pp. 287–292.

[8] F.-F. Li, A. Karpathy, J. Johnson, and S. Yeung. (2017) Stanford CS231n: Convolutional neural networks for visual recognition. [Online]. Available: http://cs231n.stanford.edu/

[9] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[10] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/6552-visual-dynamics-probabilistic-future-frame-synthesis-via-cross-convolutional-networks.pdf

[11] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 667–675.

[12] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning (ICML)*, 2015, pp. 843–852.

[13] L. Castrejon, N. Ballas, and A. Courville, "Improved conditional vrnns for video prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7608–7617.

[14] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *CoRR*, vol. abs/1710.11252, 2017. [Online]. Available: http://arxiv.org/abs/1710.11252

[15] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," *CoRR*, vol. abs/1802.07687, 2018. [Online]. Available: http://arxiv.org/abs/1802.07687

[16] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[17] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[18] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[19] S. Winkler and P. Mohandas, "The evolution of video quality measurement: From psnr to hybrid metrics," *IEEE transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, 2008.

[20] Q. Huynh-Thu and M. Ghanbari, "The accuracy of psnr in predicting video quality for different video scenes and frame rates," *Telecommunication Systems*, vol. 49, no. 1, pp. 35–48, 2012.

[21] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.

[22] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[23] P. Bhattacharjee and S. Das, "Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 4268–4277.

[24] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Towards accurate generative models of video: A new metric & challenges," *arXiv preprint arXiv:1812.01717*, 2018.

[25] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[26] Y. Bengio, "Deep learning of representations: Looking forward," in *International Conference on Statistical Language and Speech Processing*. Springer, 2013, pp. 1–37.

[27] E. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Advances in neural information processing systems*, 2017, pp. 4414–4423.

[28] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," in *Conference on Robot Learning*, 2020, pp. 100–109.

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

[30] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015, pp. 1171–1179.

[31] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," *Conference on Robot Learning (CoRL)*, 2017.

[32] C. Finn, I. J. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *CoRR*, vol. abs/1605.07157, 2016. [Online]. Available: http://arxiv.org/abs/1605.07157

[33] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv preprint arXiv:1804.01523*, 2018.

[34] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning (ICML)*, 2019.