2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561541

# Object-centric Video Prediction without Annotation

Karl Schmeckpeper,*[1] Georgios Georgakis,*[1] and Kostas Daniilidis[1]

*Abstract*— **In order to interact with the world, agents must be able to predict the results of the world's dynamics. A natural approach to learn about these dynamics is through video prediction, as cameras are ubiquitous and powerful sensors. Direct pixel-to-pixel video prediction is difficult, does not take advantage of known priors, and does not provide an easy interface to utilize the learned dynamics. Object-centric video prediction offers a solution to these problems by taking advantage of the simple prior that the world is made of objects and by providing a more natural interface for control. However, existing object-centric video prediction pipelines require dense object annotations in training video sequences. In this work, we present Object-centric Prediction without Annotation (OPA), an object-centric video prediction method that takes advantage of priors from powerful computer vision models. We validate our method on a dataset comprised of video sequences of stacked objects falling, and demonstrate how to adapt a perception model in an environment through end-to-end video prediction training.**

## I. Introduction

Modeling physical interaction is a fundamental agent skill for interacting with the world. This is a challenging skill to learn as it requires understanding the scene's dynamics. For object manipulation scenarios, the challenge is exacerbated by the need to understand the environment at the object level, including agent-object and object-object physical interactions. Addressing this problem using visual sensors offers many advantages. First, high quality cameras are easily accessible and have low size, weight, and power requirements, allowing them to be included on most robotic platforms. Second, there is an abundance of existing data available, allowing powerful deep learning models to be trained. Third, visual observations offer rich information about the environment, including pose, texture, and semantics, that cannot easily be matched by other sensors.

Existing methods typically address this problem via learning an action-conditioned predictive model that infers the changes to the visual scene. These models have been demonstrated mostly through end-to-end deep networks that learn to map pixels and control inputs to future pixels [1]. This paradigm assumes that the model can implicitly learn to visually segment the objects and infer their motion in the scene in spite of the high dimensionality of pixels from the raw image inputs. It does not take advantage of perceptual priors that can be extracted from an observation, forcing the model to function without any aid from existing computer vision methods.

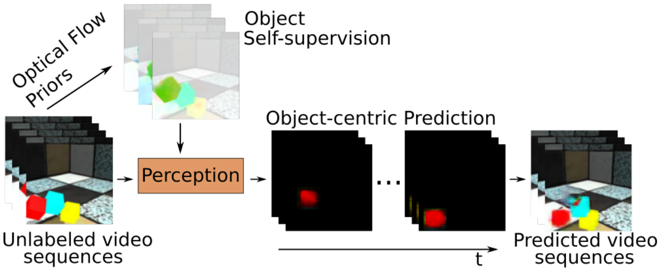
Optical Flow Priors
Object Self-supervision
Perception
Object-centric Prediction
...
Unlabeled video sequences
t
Predicted video sequences

Fig. 1. We seek to model physical interactions with a visual sensor and predict the future states of objects. During training our proposed approach leverages optical flow priors to generate object self-supervision. We segment the image into objects, before predicting the future states of the objects, and using them to generate predictions of the next frames. In testing we predict future video sequences given a single frame as input.

There have been efforts to learn pairwise object interactions by treating a visual scene as a collection of objects, which led to the development of object-centric predictive models. These methods typically assume that labeled object information is readily available at each future time step either in the form of object locations [2], or forces applied on the objects [3]. However, many robotic agents are required to operate in unstructured real-world environments that exhibit increased visual variability with no access to dense labels and often have to generalize to previously unseen objects.

Recent works have demonstrated that utilizing perceptual priors, via powerful computer vision models, reduces sample complexity, enables generalizability across environments, and largely increases performance in visuomotor tasks [4]–[7]. Inspired by these methods, we make the observation that visual motion is a strong cue for objectness [8] and propose a novel object-centric video predictive model that leverages state-of-the-art perception in the form of object instance segmentation and optical flow, and does not require object annotations. The perception model is trained end-to-end with dynamics and image generation models in order to predict a future frame sequence from a single input frame (see Figure 1). The joint training allows for the perception model to fine-tune on the existing environment, while the dynamics and generation models benefit from the rich feature representation encoded in the perception model. This results in an object-centric model that is not restricted only to environments where object level annotations are available, paving the way towards adaptable predictive models for manipulation tasks. Our contributions include: (a) the introduction of a novel prediction model that does not require object level annotations, (b) state-of-the-art results on the Shapestacks [9] dataset which demonstrate the benefits of

* Denotes equal contribution.

[1] The authors are with the GRASP Laboratory, Computer and Information Science Department, Univeristy of Pennsylvania, Philadelphia, PA 19104. E-Mail: karls@seas.upenn.edu

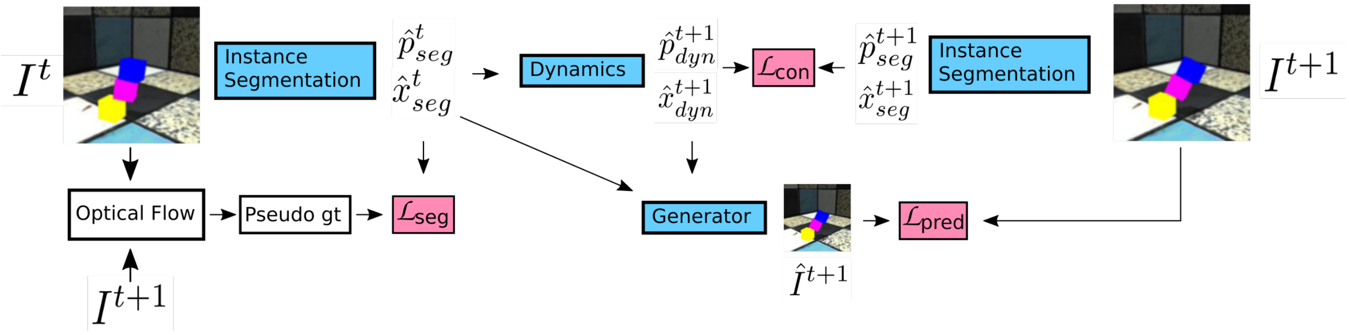

Fig. 2. Overview of our object-centric predictive model for a single training time step. Given $I^t$ we first extract patches $\hat{p}^t_{seg}$ and bounding boxes $\hat{x}^t_{seg}$ from the instance segmentation module which is trained using our generated pseudo ground-truth. $\hat{p}^t_{seg}$ and $\hat{x}^t_{seg}$ are then passed in the dynamics module that predicts future states $\hat{p}^{t+1}_{dyn}$ and $\hat{x}^{t+1}_{dyn}$. We enforce consistency between the predicted future states and the outputs of the instance segmentation on frame $I^{t+1}$. Finally, the future states are used as input to the generator that predicts the future image $\hat{I}^{t+1}$. Blue color designates modules that are trained.

our proposed method, and (c) the ability to fine-tune the perception model in new environments.

## II. Related Works

*a) Video Predictive Models:* Learning to predict the future from raw observations such as images has gained large interest in recent years, especially in the context of learning representations for planning and robotic control [1], [10]–[18]. In the seminal work of Finn et al. [14] an action-conditioned model that predicts future pixel motion was introduced. This was later demonstrated in [1] to work with model predictive control for agents interacting with objects. Ebert et al. [12] tackles the same problem but in a reinforcement learning setting, while [13] attempts to learn a suitable representation to address occlusions in manipulation scenarios. Other works improve the training of these models by leveraging both action-conditioned and action-free data, increasing the amount of data available [15], [18]. Another line of works [19]–[21] are interested in predicting future frames not conditioned on a specific action, where the focus is on learning stochastic models to capture multiple possible futures of a dynamic scene. For example, Lee et al. [19] combined latent variational variable models with adversarially-trained models in order to produce both realistic and diverse future predictions. In contrast to all the aforementioned approaches that learn to predict future frames directly from pixels, we treat a scene as a collection of objects and learn how these objects appear in the future by modeling their physical interactions.

*b) Object-centric Predictive Models:* In order to address the high dimensionality of input pixels, several methods learn structured object-factorized representations for predicting either future images [2], [22]–[25] or object properties [3], [26]–[29]. The success of these methods correlates with their understanding of physical interactions and object dynamics in a scene. Byravan et al. learn an object centric prediction model from depth images [30] and use it for robotic control [31]. The work of Ye et al. [2] predicts future states of independent entities and learns to encode their interactions through a graph neural network. Similarly, Janner et al. [22] follow an object factorization where object representations are organized as pairs before passed through a physics prediction network and a renderer. These object-centric models have been demonstrated in simple planning scenarios such as reaching [31] and pushing objects [23], but have also been illustrated in more complicated tasks such as stacking blocks [22], demonstrating that having access to an explicit object representation aids in planning. Several works in human video prediction rely on predicting the future state of a human skeleton before predicting the video, but these methods do not generalize to arbitrary objects [32], [33].

For predicting object properties, Fragkiadaki et al. [3] learns to predict object velocities in a simulated billiards game by individually modeling the temporal state of each ball. In Battaglia et al. [29], a framework for learning about object relations called interaction networks is introduced, while the work of [28] extends this framework to the visual domain. More recently, Qi et al. [27] explore the use of a region proposal network to extract object representations suitable for predicting future object locations and shapes.

Unlike our proposed method, all of these approaches require specific object related supervision such as object location, orientation, temporal association, velocities, or rely on strong assumptions such as the number of objects in the scene.

## III. Methodology

We propose an object-centric video prediction pipeline that can operate from raw pixels. Our pipeline is made up of three main components: an instance segmenter, which separates an object into entities, a dynamics model, which predicts the future location and state of each entity, and an image generator, which synthesizes the future frame from the predicted entities. We describe each component in detail, as well as how the entire system is trained end-to-end.[1] An overview of our pipeline is shown in Figure 2.

[1] All code is available at `https://github.com/kschmeckpeper/opa`

### A. Instance Segmentation

The first step in our object-centric video prediction method is to segment the input images into a set of entities. We use the popular state-of-the-art instance segmentation method of Mask R-CNN [34] from the Detectron2 [35] implementation. Mask R-CNN is built on top of the Faster R-CNN [36] object detection framework and predicts segmentation masks of detected object instances. The model uses a backbone (e.g., ResNet [37]) to extract image features maps that are initially passed to a region proposal network (RPN) which proposes candidate object bounding boxes. The network is able to compute object specific features through the RoIAlign layer which uses pooling operations to extract the image features corresponding to a specific region of interest. Then, the object features and bounding boxes are passed to separate branches of the network that are responsible to predict object labels, refine the candidate bounding boxes, and estimate the instance segmentation masks. We chose this particular model because of its modularity and its ability to provide object specific features through region of interest pooling operations, but in principal any learnable instance segmentation method can be used in our pipeline. Since Mask R-CNN is typically trained on the COCO [38] dataset, we devise a strategy to adapt it to novel robot manipulation environments.

As mentioned earlier, we assume that the environment we operate in does not offer any ground-truth at the object level. However, in video sequences motion cues often provide reliable location and mask information since all pixels belonging to a rigid object move in unison. To take advantage of this, we employ an optical flow method [39] on consecutive frames of our videos. The magnitude of the predicted flow is thresholded to keep any image areas that have moved more than $1\%$ of the image dimensions. The remaining pixels are organized into connected components in image space. The convex hull of each connected component is then taken as a pseudo ground-truth mask and all masks are defined to belong to a single object class. Figure 3 shows examples of generated pseudo ground-truth masks from optical flow. The idea is to learn what *can* move in a scene as that is an inherent indication of objectness [8], [40]. This also allows us to treat everything else in the scene, such as the floor tiles, as background.

During training we use the generated ground-truth to estimate the Mask R-CNN losses $L_{cls}$, $L_{box}$, $L_{mask}$ and the RPN losses $L_{obj}$, $L_{reg}$. $L_{cls}$ is a classification log loss, $L_{box}$ is a smooth $L_1$ loss for bounding box regression, and $L_{mask}$ is a per-pixel binary cross-entropy loss. Regarding RPN, $L_{obj}$ is a binary log loss, and $L_{reg}$ is identical to $L_{box}$. More details can be found in [34], [36]. We define $L_{seg}$ in our model to be the summation of all these losses. In practice, we pre-train the instance segmentation model on a small set of training sequences before integrating with the rest of the components.

In the forward pass of the model, given an image $I$ the instance segmentation extracts bounding boxes, $\hat{x}_{seg}$, and patches, $\hat{p}_{seg} = (\hat{f}_{seg}, \hat{m}_{seg})$ for each detected object instance, where $\hat{f}_{seg}$ is the feature representation of the object (from RoIAlign) and $\hat{m}_{seg}$ is the mask defined as a foreground probability of each pixel location.

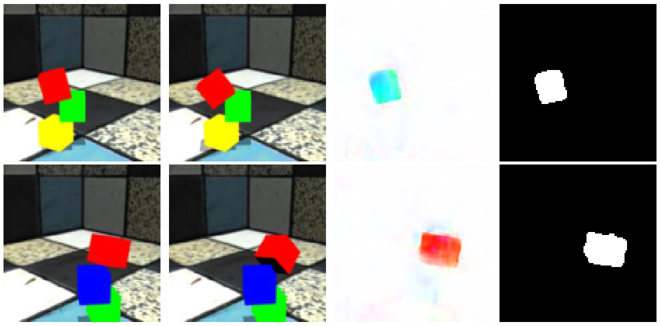

Fig. 3. Pseudo ground-truth masks (last column) generated from optical flow (third column) of consecutive frames (first two columns). Even though this strategy only annotates moving objects, which are at the top of the stack, the instance segmentation learns a good representation and can generalize to static objects, as we show in our experiments.

### B. Dynamics Prediction

Given a set of bounding boxes, $\hat{x}_{seg}$, and patches, $\hat{p}_{seg}$, the dynamics model, $\mathcal{D}$, seeks to predict their state at the next timestep:

$$\hat{x}^{t+1}_{dyn}, \hat{p}^{t+1}_{dyn} = \mathcal{D}(\hat{x}^{t}_{seg}, \hat{p}^{t}_{seg}) \tag{1}$$

Each bounding box and patch is encoded to form a latent representation. The latent representation is passed through a fully-connected neural network to generate the latent encoding at the next time step. The predicted encodings are decoded to get the predicted future location of the bounding box, $\hat{x}^{t+1}_{dyn}$ and its patch, $\hat{p}^{t+1}_{dyn}$.

During training, the algorithm has access to the ground truth images from future timesteps, so we can impose a consistency loss between the entities predicted by the dynamics model and the entities found by running the instance segmentation model on the future frame:

$$\begin{aligned}\mathcal{L}_{con} = &\left\| \hat{m}^{t+1}_{dyn} \odot \hat{f}^{t+1}_{dyn} - \hat{m}^{t+1}_{seg} \odot \hat{f}^{t+1}_{seg} \right\|^2 + \\ &\left\| \hat{x}^{t+1}_{dyn} - \hat{x}^{t+1}_{seg} \right\|^2\end{aligned} \tag{2}$$

To ensure that the consistency loss is applied to corresponding entities, we associate the instance predictions from the input image $I^t$ to those of future image $I^{t+1}$ based on bounding box centroid proximity in image space. This procedure is repeated for any following future images and establishes a constant number of entities throughout a single sequence. Another option for establishing these associations is to warp $\hat{x}_{seg}$ and $\hat{p}_{seg}$ according to optical flow between $I^t$ and $I^{t+1}$. However, we discovered that this was unreliable for long sequences due to artifacts introduced during warping.

### C. Image Generation

The final image is generated as a function of the previous image, $I^t$, the previous bounding boxes $\hat{x}^t_{seg}$ and segmentation masks $\hat{m}^t_{seg}$, and the predicted future patches $\hat{p}^{t+1}_{dyn}$ and bounding boxes $\hat{x}^{t+1}_{dyn}$:

$$\hat{I}^{t+1} = \mathcal{G}(I^t, \hat{x}^t_{seg}, \hat{m}^t_{seg}, \hat{x}^{t+1}_{dyn}, \hat{p}^{t+1}_{dyn}). \tag{3}$$

The image generator $\mathcal{G}$ works by first generating separate images containing pixels for the predicted objects $\hat{I}_{obj}$, pixels for the background $\hat{I}_{back}$, synthetic pixels $\hat{I}_{synth}$, and then compositing them together. We will describe how each image is created and how they are combined in the subsequent paragraphs. First, we define a transformation function $\gamma$, which converts the segmentation masks $\hat{m}^t$ to image coordinates using their corresponding bounding boxes $\hat{x}^t$:

$$\hat{M}^t = \gamma(\hat{m}^t, \hat{x}^t) \tag{4}$$

where $\hat{M}^t$ is a binary mask with the same dimensions as the previous image $I^t$.

The contribution of the objects is made by passing each patch through a convolutional neural network $\mathcal{F}$ that decodes from features to pixels, and then taking the sum of the pixels multiplied by their corresponding masks:

$$\hat{I}^{t+1}_{obj} = \sum_{i=1}^{n} \gamma\left(\hat{m}^{t+1}_{dyn,i} \odot \mathcal{F}(\hat{p}^{t+1}_{dyn,i}), \hat{x}^{t+1}_{dyn}\right) \tag{5}$$

where $n$ is the number of object instances.

Our object-centric video prediction model assumes that any change in the image can be explained by an object, meaning the background pixels, which are copied directly between frames, are all pixels that are not objects. The background mask is therefore found by subtracting all current and previous object masks from a mask of all ones:

$$\hat{M}^{t+1}_{back} = 1 - \sum_{i=1}^{n} \hat{M}^t_{seg,i} - \sum_{i=1}^{n} \hat{M}^{t+1}_{dyn,i}. \tag{6}$$

The background pixels are then the product of the background mask and the image from the previous timestep:

$$\hat{I}^{t+1}_{back} = \hat{M}^{t+1}_{back} \odot I^t. \tag{7}$$

We generate synthetic pixels to fill in the holes between the background mask and the object masks. These are primarily regions of the image that were occluded by an object but are no longer occluded. The mask that controls the locations of the synthetic pixel is given by the following equation:

$$\hat{M}^{t+1}_{synth} = 1 - \hat{M}^{t+1}_{back} - \sum_{i=1}^{n} \hat{M}^{t+1}_{dyn,i}. \tag{8}$$

The synthetic pixels are generated from a convolutional neural network $\Psi$ that takes the object pixels, $\hat{I}_{obj}$, and the background pixels, $\hat{I}_{back}$, as input:

$$\hat{I}^{t+1}_{synth} = \hat{M}^{t+1}_{synth} \odot \Psi(\hat{I}^{t+1}_{obj}, \hat{I}^{t+1}_{back}). \tag{9}$$

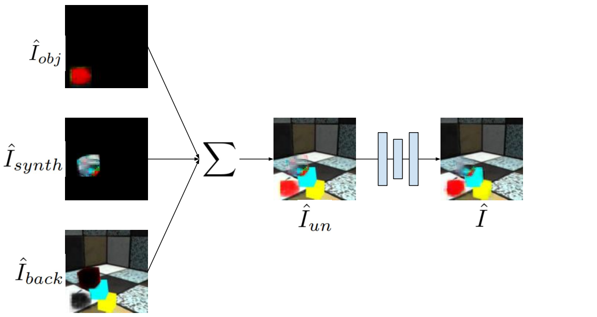

Fig. 4. Image generation. This shows the components of the image generator $\mathcal{G}$. The unrefined image, $\hat{I}_{un}$, is generated by summing together the contributions from the objects, $\hat{I}_{obj}$, the background, $\hat{I}_{back}$, and the synthetic pixels, $\hat{I}_{synth}$. The unrefined image is passed through a neural network to clean up the edges between its component patches, generating the final image, $\hat{I}$.

We then sum together the background pixels, $\hat{I}^{t+1}_{back}$, the object pixels, $\hat{I}^{t+1}_{obj}$, and the synthetic pixels, $\hat{I}^{t+1}_{synth}$, generating an initial unrefined image, $\hat{I}^{t+1}_{un}$:

$$\hat{I}^{t+1}_{un} = \hat{I}^{t+1}_{back} + \hat{I}^{t+1}_{obj} + \hat{I}^{t+1}_{synth}. \tag{10}$$

The unrefined image, $\hat{I}_{un}$, is passed through a convolutional neural network to clean up the edges between its component patches, generating the final image, $\hat{I}$. The image generator process is shown in Figure 4. The final predicted image and the unrefined image are supervised using an L2 loss with the ground truth future image:

$$\begin{aligned}\mathcal{L}_{pred} = &\left\|\hat{I}^{t+1} - I^{t+1}\right\|^2 + \left\|\hat{I}^{t+1}_{un} - I^{t+1}\right\|^2 \\ &+ \alpha \left\|\hat{u}_{seg} \odot \hat{I}^{t+1} - \hat{u}_{seg} \odot I^{t+1}\right\|^2.\end{aligned} \tag{11}$$

The third term is a masking loss that provides attention to foreground pixels, where $\hat{u}_{seg}$ are the predicted binary masks, and $\alpha$ is a weighting hyperparameter. Both the final predicted image and the unrefined predicted image are supervised to ensure that the revising network does not attempt to learn dynamics.

The entire system is jointly optimized to minimize the combined loss:

$$\mathcal{L} = \mathcal{L}_{pred} + c_1\mathcal{L}_{con} + c_2\mathcal{L}_{seg} \tag{12}$$

where $c_1$, and $c_2$ are hyperparamters that weight the relative importance of each part of the loss.

## IV. Experiments

We seek to show that our prediction model can provide good video prediction when trained without any annotations, such as bounding boxes, on the training data. In order to do this, the model must be able to segment the input images and predict the future states of objects. We present video prediction results, and evaluate the internal object detection and object-centric prediction.

**Implementation Details:** We use the Mask R-CNN model with a ResNet50+FPN backbone that is pretrained on the

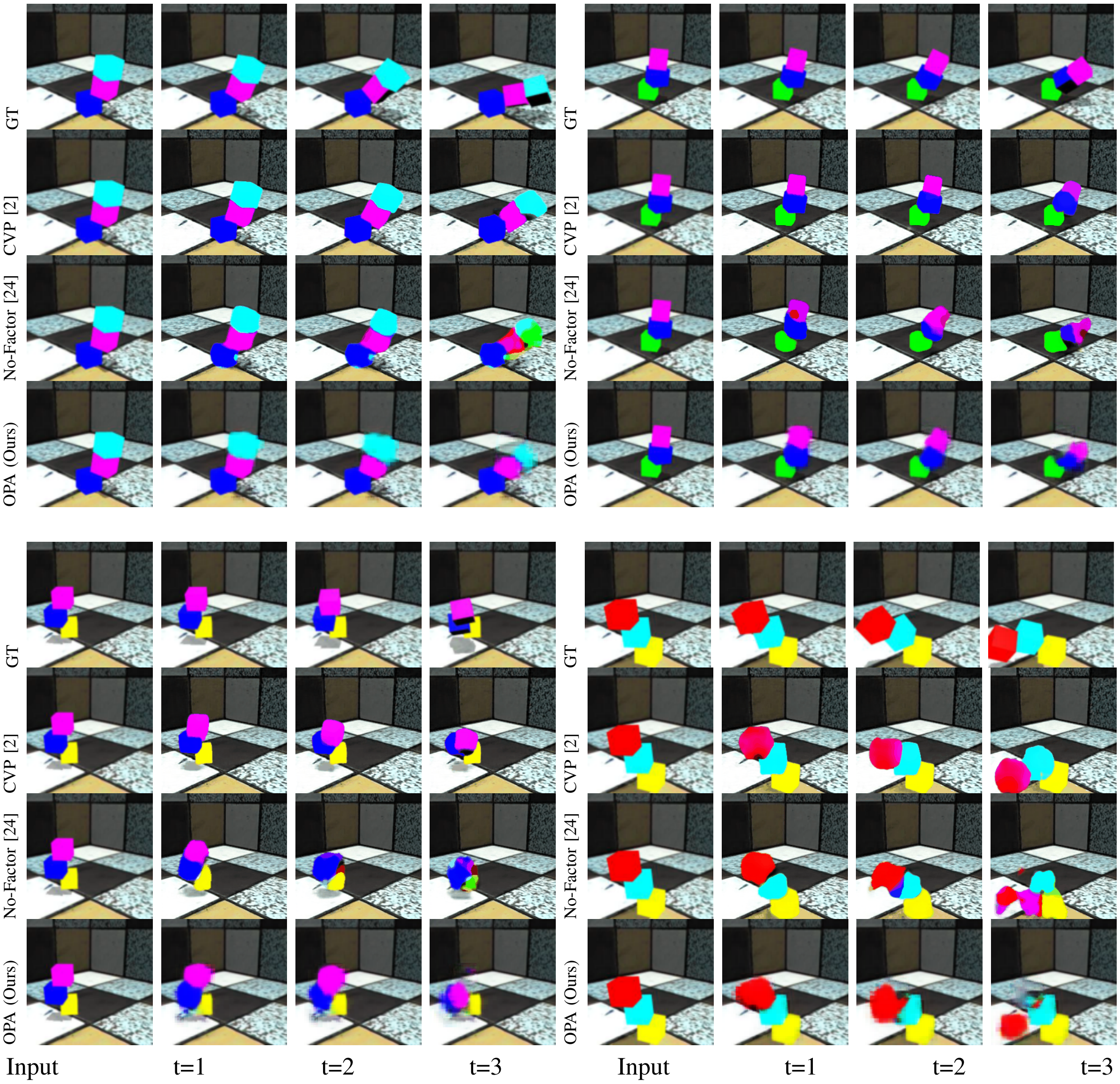

Fig. 5. Qualitative Prediction Results. OPA, our prediction model, is able to achieve better performance than the non-object-centric prediction model from [24]. Our predicted videos retain more consistent object properties, such as color and shape than the competing approach. Additionally, OPA is able to achieve comparable results to CVP [2], despite CVP requiring ground truth annotations including bounding boxes and object tracks. OPA is able to bring the predictive quality of object-centric video prediction to datasets without annotations.

COCO [38] dataset. Our dynamics model $\mathcal{D}$ is implemented as a 4-layer fully connected network, the feature-to-pixel network $\mathcal{F}$ is a 3-layer convolutional network and the synthetic pixel generator network, $\Psi$ is a stacked-hourglass network. Our patch size is 14 pixels. For more information, see the official implementation.

### A. Video Prediction

We perform our video prediction experiments in an extension of the Shapestacks dataset [2], [9]. The environment is made up of an unstable stack of blocks placed in front of the camera. In all experiments, the model is given a single starting frame and must predict the subsequent images. We sample every other frame of the original dataset in order to speed up the videos.

We compare our approach to several existing approaches. Compositional Video Prediction (CVP), a state of the art object-centric video prediction that requires ground truth bounding boxes at each timestep [2]. We additionally compare against the non-object centric approach, which we train both with and without access to the ground-truth bounding boxes [24]. In comparison, our approach requires no annotations or ground-truth bounding boxes, allowing it to learn directly from pixels. For all baselines, we use the implementations provided by [2].

The results of the video prediction experiments are shown in Table I. We evaluate the prediction performance across sequences of length three, given a single input frame, using the Mean Square Error (MSE) and the Learned Perceptual Image Patch Similarity (LPIPS) [41] metrics. Qualitative results are shown in Figure 5.

Our model is able to achieve better prediction results

| Method | MSE (↓) | LPIPS [41] (↓) |
|---|---|---|
| CVP [2] (Requires GT bounding boxes) | $0.010134 \pm 0.000318$ | $0.051941 \pm 0.001432$ |
| No-factor [24] (w/ GT bounding boxes) | $0.012974 \pm 0.000327$ | $0.067548 \pm 0.001381$ |
| No-factor [24] (w/o GT bounding boxes) | $0.010821 \pm 0.000341$ | $0.069826 \pm 0.001524$ |
| OPA (Ours) (w/o GT bounding boxes) | $\mathbf{0.009720 \pm 0.000314}$ | $\mathbf{0.0619602 \pm 0.001567}$ |

TABLE I

MEAN AND STANDARD ERROR FOR PREDICTION ON SHAPESTACKS.

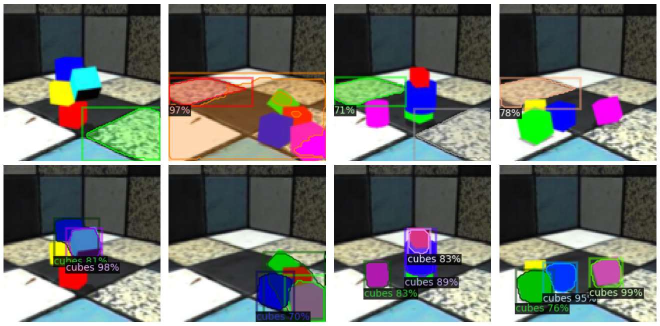

Fig. 6. Mask R-CNN detections before (top row) and after (bottom row) the instance segmentation model was finetuned using our generated pseudo ground-truth.

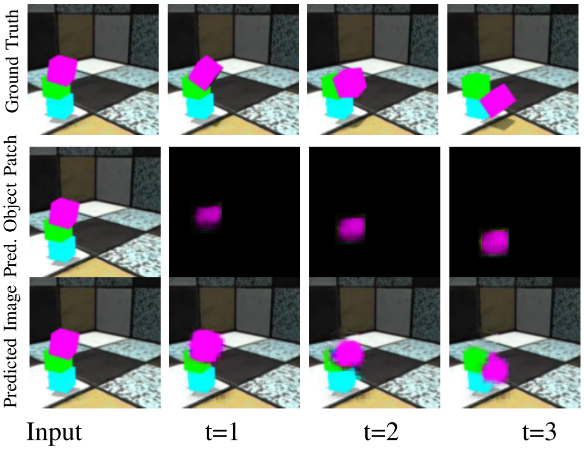

Fig. 7. Object-centric prediction. Our model predicts the future states of individual objects, middle. By predicting the states of multiple objects and reasoning about non-object pixels, our system is able to predict the full image, bottom.

than the non-object-centric no-factor prediction model [24]. Unlike [24] which tackles this problem by learning to map pixels to pixels in an object-agnostic manner, our approach is able to maintain more consistent object properties, such as color and shape.

We treat CVP's performance as the lower bound error that an object-centric model can achieve because it makes use of all possible object annotations during training. We note that our performance is comparable even though we train without ground truth bounding box annotations or object tracks, making it so our approach is much more widely applicable.

### B. Learned Segmentation

We demonstrate that our proposed approach of generating pseudo ground-truth from optical flow cues provides sufficient supervision to fine-tune the instance segmentation model. Figure 6 illustrates Mask R-CNN detections before and after our pseudo ground-truth training strategy. Notice how the model generalizes to multiple objects in each scene even though the pseudo ground-truth from optical flow usually only annotates objects at the top of the stack.

The fact that optical flow priors [39] can provide supervision to the instance segmentation model is actually a confirmation that visual motion can provide strong objectness cues [8], [40], even though it was pre-trained using a significantly different dataset. In contrast, Mask R-CNN, a powerful object instance segmentation model, requires finetuning in the context of our experimental setup.

### C. Object-Centric prediction

Additionally, our prediction pipeline maintains good object representations during prediction. An example of the predicted object is shown in Figure 7. Being able to predict the future states of individual objects is important, not just because it improves the quality of the final predicted image, but also because it allows for easier interfacing with downstream tasks. Many robotic tasks involve interacting with objects, and maintaining an explicit object representation allows the agent to plan on those representations, rather than having to plan in pixel space.

## V. CONCLUSIONS

We present Object-centric Prediction without Annotation (OPA), an approach to training object-centric video prediction models purely from unlabeled video. We have demonstrated our model's advantage towards object-agnostic prediction models and have shown comparable performance to methods that use dense object annotations during training. Object-centric video prediction models offer a promising way to allow robots to learn about the dynamics of the world from cheap video data. They have the ability to handle more difficult dynamics while providing better interfaces for robotic control than pixel-to-pixel prediction models. OPA's ability to learn object-centric video prediction from video without annotations offers the promise of bringing object-centric video prediction to more applications, allowing robots to better understand and anticipate their environments.

## ACKNOWLEDGEMENTS

Research was sponsored by the Army Research Office and was accomplished under grants ARO MURI W911NF-20-1-0080, NSF CPS 2038873, ARL DCIST CRA W911NF-17-2-0181, ONR N00014-17-1-2093, and by the Honda Research Institute.

# References

[1] 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.

[2] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani, "Compositional video prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 353–10 362.

[3] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik, "Learning visual predictive models of physics for playing billiards," *arXiv preprint arXiv:1511.07404*, 2015.

[4] A. Sax, J. O. Zhang, B. Emi, A. Zamir, S. Savarese, L. Guibas, and J. Malik, "Learning to navigate using mid-level visual priors," *Conference on Robot Learning*, 2019.

[5] B. Zhou, P. Krähenbühl, and V. Koltun, "Does computer vision matter for action?" *arXiv preprint arXiv:1905.12887*, 2019.

[6] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[7] G. Georgakis, Y. Li, and J. Kosecka, "Simultaneous mapping and target driven navigation," *arXiv preprint arXiv:1911.07980*, 2019.

[8] B. Xiong, S. D. Jain, and K. Grauman, "Pixel objectness: learning to segment generic objects automatically in images and videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2677–2692, 2018.

[9] O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi, "Shapestacks: Learning vision-based physical intuition for generalised object stacking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 702–717.

[10] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Advances in neural information processing systems*, 2015, pp. 2863–2871.

[11] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," *arXiv preprint arXiv:1704.02254*, 2017.

[12] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[13] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," *arXiv preprint arXiv:1710.05268*, 2017.

[14] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, 2016, pp. 64–72.

[15] K. Schmeckpeper, A. Xie, O. Rybkin, S. Tian, K. Daniilidis, S. Levine, and C. Finn, "Learning predictive models from observation and interaction," *In European Conference on Computer Vision (ECCV)*, 2020.

[16] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[17] M. Zhang, S. Vikram, L. Smith, P. Abbeel, M. Johnson, and S. Levine, "Solar: Deep structured representations for model-based reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7444–7453.

[18] O. Rybkin, K. Pertsch, K. G. Derpanis, K. Daniilidis, and A. Jaegle, "Learning what you can do before doing anything," *arXiv preprint arXiv:1806.09655*, 2018.

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

[20] 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*, 2016, pp. 91–99.

[21] E. L. Denton *et al.*, "Unsupervised learning of disentangled representations from video," in *Advances in neural information processing systems*, 2017, pp. 4414–4423.

[22] 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.

[23] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," in *Conference on Robot Learning*, 2019.

[24] A. Lerer, S. Gross, and R. Fergus, "Learning physical intuition of block towers by example," *arXiv preprint arXiv:1603.01312*, 2016.

[25] S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. Mitra, and A. Vedaldi, "Relate: Physically plausible multi-object scene synthesis using structured latent spaces," *arXiv preprint arXiv:2007.01272*, 2020.

[26] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," *arXiv preprint arXiv:1612.00341*, 2016.

[27] H. Qi, X. Wang, D. Pathak, Y. Ma, and J. Malik, "Learning long-term visual dynamics with region proposal interaction networks," *arXiv preprint arXiv:2008.02265*, 2020.

[28] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, "Visual interaction networks: Learning a physics simulator from video," in *Advances in neural information processing systems*, 2017, pp. 4539–4547.

[29] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, *et al.*, "Interaction networks for learning about objects, relations and physics," in *Advances in neural information processing systems*, 2016, pp. 4502–4510.

[30] A. Byravan and D. Fox, "SE3-nets: Learning rigid body motion using deep neural networks," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 3, pp. 173–180, 2017.

[31] A. Byravan, F. Lceb, F. Meier, and D. Fox, "SE3-Pose-Nets: Structured deep dynamics models for visuomotor control," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3339–3346, 2018. [Online]. Available: https://arxiv.org/pdf/1710.00489.pdf

[32] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang, "Deep Video Generation, Prediction and Completion of Human Action Sequences," nov 2017. [Online]. Available: http://arxiv.org/abs/1711.08682http://dx.doi.org/10.1007/978-3-030-01216-8{_}23

[33] N. Fushishita, A. Tejero-de Pablos, Y. Mukuta, and T. Harada, "Long-Term Video Generation of Multiple Futures Using Human Poses," apr 2019. [Online]. Available: http://arxiv.org/abs/1904.07538

[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[35] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[39] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] A. Dave, P. Tokmakov, and D. Ramanan, "Towards segmenting anything that moves," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.