34th International Conference on Artificial Neural Networks (ICANN), Kaunas, Lithuania, 2025.

VideoPCDNet: Video Parsing and Prediction with Phase Correlation Networks

Noel José Rodrigues Vicente[1], Enrique Lehner[1], Angel Villar-Corrales[1,2,3]✉, Jan Nogga[1,2,3] and Sven Behnke[1,2,3]

[1] Autonomous Intelligent Systems group, University of Bonn, Germany
[2] Lamarr Institute for Machine Learning and Artificial Intelligence
[3] Center for Robotics, University of Bonn, Germany
villar@ais.uni-bonn.de

Abstract. Understanding and predicting video content is essential for planning and reasoning in dynamic environments. Despite advancements, unsupervised learning of object representations and dynamics remains challenging. We present VideoPCDNet, an unsupervised framework for object-centric video decomposition and prediction. Our model uses frequency-domain phase correlation techniques to recursively parse videos into object components, which are represented as transformed versions of learned object prototypes, enabling accurate and interpretable tracking. By explicitly modeling object motion through a combination of frequency domain operations and lightweight learned modules, VideoPCDNet enables accurate unsupervised object tracking and prediction of future video frames. In our experiments, we demonstrate that VideoPCDNet outperforms multiple object-centric baseline models for unsupervised tracking and prediction on several synthetic datasets, while learning interpretable object and motion representations.

Keywords: Object-centric video prediction, object-centric learning, phase-correlation networks, unsupervised learning

1 Introduction

Humans naturally interpret dynamic scenes by segmenting them into discrete objects and tracking their interactions over time. Recent works in object-centric video prediction imitate this cognitive process by learning unsupervised object representations and modeling their dynamics and interactions using recurrent neural networks [29] (RNNs) or transformers [25,28,3]. However, these methods typically incur high computational costs, require large amounts of training data, and produce models whose internal representations remain largely opaque.

In this work, we propose VideoPCDNet, an unsupervised video decomposition and prediction model that extends the Phase-Correlation Decomposition Network (PCDNet) [23] to the video domain. Unlike conventional object-centric methods, which rely on high-dimensional latent spaces to capture object representations and dynamics, our proposed VideoPCDNet represents objects as transformed versions of a set of learned object prototypes and explicitly encodes their motion using phase differences. This design not only enables efficient

object tracking and scene parsing, but also yields representations that are inherently interpretable. Our model recursively parses a video sequence into its object components, leading to an accurate and robust object tracking even under challenging motion dynamics and occlusions. Furthermore, VideoPCDNet explicitly models object motion by combining phase-correlation techniques with lightweight learned modules in order to efficiently forecast future object states and video frames with a minimal number of trainable parameters.

In our experiments, we demonstrate that VideoPCDNet outperforms multiple baseline models for unsupervised object tracking and future frame video prediction on several synthetic datasets while also learning interpretable object and motion representations. Our work thus demonstrates that integrating frequency-domain processing with object-centric representation learning can yield a more efficient and interpretable framework for video prediction.

2 Related Work

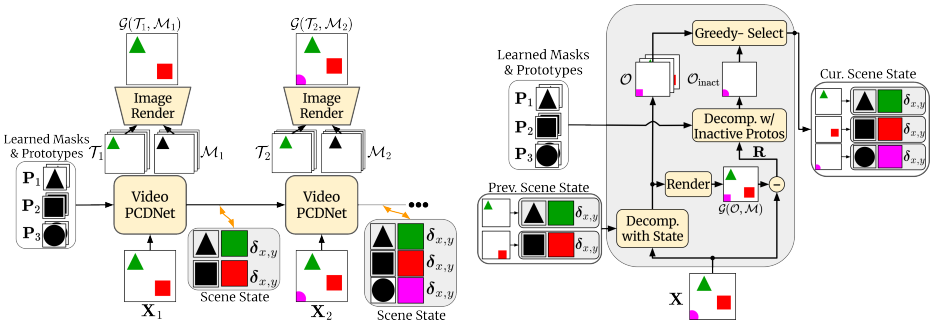
2.1 Object-Centric Learning

Object-centric representation learning. Object-centric learning methods aim to decompose an image or video into a set of object components in an unsupervised manner. These objects can be represented as unconstrained latent vectors (often called slots) [2,26,15,13,14], factored latent variables [11,21], spatial mixture models [10,5], or explicit object prototypes [23,16]. The learned object representations benefit multiple downstream tasks, such as learning behaviors for robotic manipulation [18,24] and unsupervised segmentation [14,5].

Object-centric video prediction. Object-centric video prediction aims to model object dynamics and interactions to forecast future object states and frames. Recently, several methods propose to model object dynamics using slot-based representations and different architectural priors, including RNNs [29,19] or transformers [25,28,3]. In contrast, VideoPCDNet leverages phase-correlation as a strong inductive bias to model object motion interpretably and with minimal learnable parameters.

2.2 Phase-Correlation Networks

Phase-correlation networks are a class of neural networks that incorporate the differentiable phase-correlation technique [1] to estimate transformation parameters, such as translations, between two signals by analyzing the phase differences in their Fourier transforms. Integrating this operation into neural networks often leads to interpretable and compact models for image and video tasks. PCDNet [23] uses the phase correlation method to align a set of learned object prototypes with input images, enabling unsupervised object-centric image decomposition. In the video domain, Frequency Domain Transformer Networks (FDTN) [6] compute phase differences between consecutive video frames to model linear motion, and recursively apply the inferred motion to forecast future video frames.



(a) VideoPCDNet Pipeline (b) VideoPCDNet Object Parsing

Fig. 1: Overview of VideoPCDNet. Our model recursively parses a video sequence $\mathbf{X}_i$ into its object components, which are represented as transformed versions of a set of learned object prototypes $\mathcal{P}$. VideoPCDNet maintains an interpretable scene state, which encodes the objects present in the scene and their properties, thus enabling interpretable object prediction and tracking.

Several works extended this approach for motion segmentation [7,9], modeling object rotations and scale variations [27], stochastic video prediction [8] or learning relational motion [17]. Unlike previous methods, which treat the scene's motion holistically or merely separate foreground from background, our proposed VideoPCDNet parses a video sequence into individual object components and explicitly models each object's motion.

3 VideoPCDNet

We propose VideoPCDNet, illustrated in Fig. 1, a novel framework that extends the phase-correlation decomposition network PCDNet [23] for object-centric video parsing and prediction. Given a sequence of images $\mathbf{X}_{1:C}$, our method recursively parses each frame into N independent object components $\mathcal{O} = \{\mathbf{O}_1, ..., \mathbf{O}_N\}$, where each of these objects corresponds to a transformed version of a prototype from a set of learned object prototypes $\mathcal{P} = \{\mathbf{P}_1, ..., \mathbf{P}_P\}$. To allow for a temporally consistent object-centric representation, each object $\mathbf{O}_i$ is represented by a state $\mathbf{s}$ that encodes interpretable attributes such as shape, color, and position, enabling robust object tracking over time (Sec. 3.2). A key component in VideoPCDNet is the *Motion Module* (Sec. 3.3), which leverages phase correlation in the frequency domain to predict future object states, which can be used to render a video frames.

3.1 Preliminaries: PCDNet

PCDNet [23] is an unsupervised decomposition method that parses images into distinct object-centric components, which are represented as transformed versions of a set of learned object prototypes $\mathcal{P}$. Each object prototype $\mathbf{P}_j$ is learned along with its corresponding mask $\mathbf{M}_j$, which is used to model depth ordering

and occlusions. At its core, PCDNet leverages a differentiable phase-correlation mechanism, known as PC-Cell, to align learned prototypes with objects in the input image $\mathbf{X}$ by estimating spatial translations in the frequency domain. Specifically, the PC-Cell computes the cross-correlation in the frequency domain between an object prototype and the input image, producing a localization matrix $\mathbf{L}$ that encodes potential object locations as correlation peaks:

$$\Delta\theta = \left(\frac{\mathcal{F}(\mathbf{X}) \odot \overline{\mathcal{F}(\mathbf{P})}}{||\mathcal{F}(\mathbf{X}) \odot \overline{\mathcal{F}(\mathbf{P})}||} \right), \quad (1) \qquad \mathbf{L} = \mathcal{F}^{-1}(\Delta\theta), \quad (2)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier and Inverse Fourier transforms, respectively, and $\overline{\mathcal{F}(\mathbf{P})}$ denotes the complex conjugate of $\mathcal{F}(\mathbf{P})$. From this matrix, the most prominent peaks are extracted, which represent the spatial shifts (δ_x, δ_y) that best align the corresponding prototype $\mathbf{P}$ to the scene.

Originally, PCDNet handles color information via a separate Color Module, which adjusts prototype colors only after spatial alignment, disregarding color cues during the decomposition process. In contrast, our framework incorporates color as an additional feature for decomposition. To achieve this, we discretize the image colors using k-means clustering, converting the image into a multi-channel representation where each channel corresponds to distinct color cluster.

Our refined PC-Cell operates by computing channel-wise phase-correlation (Eq. (1)) between the object prototypes and this multi-channel image representation. Each correlation peak is thus represented by a triplet $(\mathbf{c}, \delta_x, \delta_y)$, identifying the strongest responding color channel $\mathbf{c}$ and corresponding shift parameters (δ_x, δ_y). Using the Fourier shift theorem, the transformed prototypes $\mathcal{T} = \{\mathbf{T}_1, ...\mathbf{T}_{|\mathcal{T}|}\}$, denoted as object templates, are computed as:

$$\mathbf{T} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{P}) \cdot \exp(-i2\pi(\delta_x \mathbf{f}_x + \delta_y \mathbf{f}_y))\right), \quad (3)$$

where $\mathbf{f}_x$ and $\mathbf{f}_y$ denote the frequencies along the horizontal and vertical directions, respectively.

Finally, PCDNet employs an iterative greedy algorithm, described in Algorithm 1, in order to select the transformed prototypes $\mathcal{T}$ and their corresponding transformed masks $\mathcal{M}$ that best represent the image. This greedy algorithm iteratively selects the object template that, combined with the previously selected templates, minimizes the reconstruction error using Eq. (4). The object templates and masks are composed using Eq. (5) to reconstruct the images, such that the first selected object $(\mathbf{T}_1)$ corresponds to the one closest to the viewer, whereas the last selected object $(\mathbf{T}_N)$ is located the furthest from the viewer; thus inherently modeling relative depth ordering between objects.

$$\mathbf{E}(\mathbf{X}, \mathcal{T}) = ||\mathbf{X} - \mathcal{G}(\mathcal{T}, \mathcal{M})||, \quad (4)$$

$$\mathcal{G}(\mathcal{T}, \mathcal{M}) = \mathbf{T}_{i+1} \odot (1 - \mathbf{M}_i) + \mathbf{T}_i \odot \mathbf{M}_i \;\; \forall\, i \in \{N, ..., 1\}. \quad (5)$$

PCDNet is trained end-to-end by minimizing an image reconstruction error, as well as two regularization costs to enforce sparsity in the object prototypes and smooth object masks.

Algorithm 1 Greedy Object and Mask Selection for Reconstruction

procedure GREEDYSELECT($\mathbf{X}, \mathcal{T}, \mathcal{M}$, `max_objs`)
  $\mathcal{S} \leftarrow \varnothing$
  while $|\mathcal{S}| <$ `max_objs` **do**
    for all $j \notin \mathcal{S}$ **do**
      Let $\mathcal{S}' \leftarrow \mathcal{S} \cup \{j\}$
      Compute the reconstruction $\mathcal{G}(\mathcal{T}_{\mathcal{S}'},\ \mathcal{M}_{\mathcal{S}'})$ (Eq. (5))
      Compute error $\mathbf{E}_j = ||\mathbf{X} - \mathcal{G}(\mathcal{T}_{\mathcal{S}'},\ \mathcal{M}_{\mathcal{S}'})||$ (Eq. (4))
    $j^* \leftarrow \arg\min_{j \notin \mathcal{S}} \mathbf{E}_j$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^*\}$
  return $\mathcal{T}_{\mathcal{S}},\ \mathcal{M}_{\mathcal{S}}$

3.2 Video Processing with PCDNet

State Representation and Alignment While PCDNet successfully parses images into their object components, extending this algorithm for video processing introduces several challenges, such as consistent object tracking.

Our proposed VideoPCDNet addresses these issues by recursively tracking and updating a consistent scene state that determines the objects present in the scene, as well as their main attributes. Namely, VideoPCDNet computes for each object k in the scene a state $\mathbf{s}_k = (\mathbf{c}_k, \mathbf{P}_k, \mathbf{Z}_k)$ that represents the object color $\mathbf{c}_k$, appearance $\mathbf{P}_k$, and center of mass $\mathbf{Z}_k$. This state representation enables VideoPCDNet to obtain a temporally consistent tracking of the objects in the scene by conditioning and aligning the decomposition process with the current state. Given the object representations from a video frame and the current VideoPCDNet state, we define the cost of matching each parsed object i to each tracked object j as:

$$C_{i,j} = \lambda_{\mathbf{c}}\ ||\mathbf{c}_i - \mathbf{c}_j|| + \lambda_{\mathbf{P}}||\mathbf{P}_i - \mathbf{P}_j|| + \lambda_{\mathbf{Z}}||\mathbf{Z}_i - \mathbf{Z}_j||, \tag{6}$$

where $\lambda_{\mathbf{c}}$, $\lambda_{\mathbf{P}}$ and $\lambda_{\mathbf{Z}}$ denote weights for each representation. The parsed objects are then aligned to the VideoPCDNet state according to the Hungarian algorithm. Stable identifiers are maintained by retaining the track IDs of matched objects, dropping those of unmatched ones, and assigning new IDs to newly detected objects; thus achieving temporally consistent object tracking.

Two-Stage Decomposition To improve the efficiency and robustness for object-centric video decomposition, VideoPCDNet leverages a two-stage approach, which is detailed in Algorithm 2. In the first stage, our framework parses the current observation using the object prototypes present in the VideoPCDNet state, reducing interference from inactive prototypes (i.e. those not present in the scene state) and improving the decomposition temporal consistency and efficiency. However, this first stage lacks the ability to parse objects that were not represented in the state, such as objects entering the scene or previously occluded.

This limitation is addressed in the second stage, which refines the initial scene parsing if its reconstruction error exceeds a certain threshold. The second stage

Algorithm 2 Two-Stage Object-Centric Video Parsing Algorithm

procedure TwoStageSelect($\mathbf{X}$, $\mathcal{T}_{ext}$, $\mathcal{M}_{ext}$, `max_objs`, `err_thr`, $\mathcal{O}_{\text{prev}}$, $\mathcal{P}$)
  # Stage 1: Parse image with state
  $(\mathcal{T}, \mathcal{M}) \leftarrow$ CreateCandidatesWithState($\mathbf{X}$, $\mathcal{O}_{\text{prev}}$)
  $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{T}_{ext}, \quad \mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}_{ext}$
  $(\mathcal{O}, \mathcal{M}) \leftarrow$ GreedySelect($\mathbf{X}$, $\mathcal{T}$, $\mathcal{M}$, `max_objs`)
  # Compute residual error
  $\mathbf{R} \leftarrow \mathbf{X} - \mathcal{G}(\mathcal{O}, \mathcal{M})$
  if $\|\mathbf{R}\| \leq$ `err_thr` **then**
    $(\mathcal{O}, \mathcal{M}) \leftarrow$ AlignWithPrevious($\mathcal{O}, \mathcal{M}, \mathcal{O}_{\text{prev}}$)
    return $(\mathcal{O}, \mathcal{M})$
  # Stage 2: Parse residual with inactive prototypes
  $\overline{\mathcal{O}}_{\text{prev}} = \mathcal{P} \setminus \mathcal{O}_{\text{prev}}$
  $(\mathcal{T}_{\text{inact}}, \mathcal{M}_{\text{inact}}) \leftarrow$ CreateCandidatesWithState($\mathbf{R}, \overline{\mathcal{O}}_{\text{prev}}$)
  $(\mathcal{O}_{\text{inact}}, \mathcal{M}_{\text{inact}}) \leftarrow$ GreedySelect($\mathbf{R}$, $\mathcal{T}_{\text{inact}}$, $\mathcal{M}_{\text{inact}}$, `max_objs`)
  $\mathcal{O}_{\text{all}} \leftarrow \mathcal{O} \cup \mathcal{O}_{\text{inact}}$, $\mathcal{M}_{\text{all}} \leftarrow \mathcal{M} \cup \mathcal{M}_{\text{inact}}$
  # Select best object and mask candidates from both stages
  $(\mathcal{O}, \mathcal{M}) \leftarrow$ GreedySelect($\mathbf{X}$, $\mathcal{O}_{\text{all}}$, $\mathcal{M}_{\text{all}}$, `max_objs`)
  # Align selected objects with state
  $(\mathcal{O}, \mathcal{M}) \leftarrow$ AlignWithPrevious($\mathcal{O}, \mathcal{M}, \mathcal{O}_{\text{prev}}$)
  return $\mathcal{O}, \mathcal{M}$

performs a more exhaustive processing of the scene, where the residual error from the first stage is measured, and candidate object templates are computed using the previously inactive prototypes in order to minimize such residual. Finally, the best object candidates among the first and second stages are selected to reconstruct the image, and aligned to update the scene state.

To further enhance robustness in challenging scenarios, such as partial occlusions, VideoPCDNet integrates externally generated object templates ($\mathcal{T}_{ext}$) and masks ($\mathcal{M}_{ext}$) derived from predicted object states. By concatenating these external object candidates with internal ones, VideoPCDNet improves object decomposition consistency and accuracy, particularly in scenarios where the internal localization mechanisms alone may fail.

3.3 Object-Centric Motion Prediction

The object-centric state used in VideoPCDNet enables interpretable forecasting of future object states by modeling object phase differences.

Given two seed frames $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$, we decompose these images into their aligned object components $\mathcal{O}_{t-1}$ and $\mathcal{O}_t$. For each object, we compute with Eq. (1) the phase difference $\Delta\theta$ between the two time-steps, which encodes the object's translation speed. Future object states can then be predicted by adding the estimated phase differences:

$$\hat{\mathbf{O}}_{t+1} = \mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{O}_t) \cdot \exp(i2\pi\Delta\theta)\right). \tag{7}$$

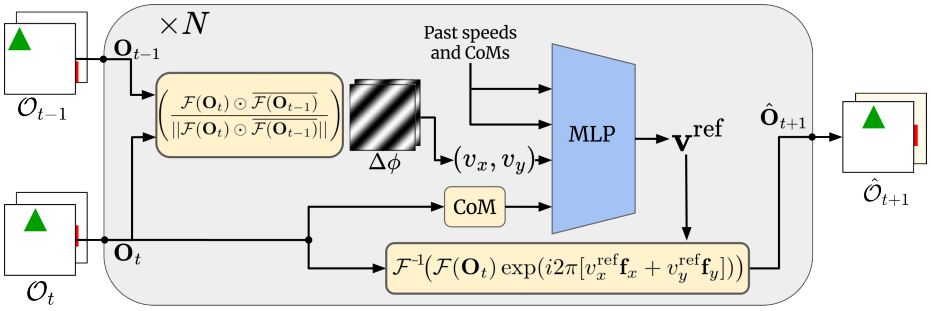


Fig. 2: Illustration of the *Motion Module* in VideoPCDNet. Future object states are predicted leveraging frequency domain operations and a learned module.

Future video frames are then rendered by combing all predicted object states as described in Sec. 3.1.

This simple, yet effective, approach for object-centric video prediction leads to temporal consistent and interpretable predictions. However, it suffers from several limitations, such as only modeling linear motion or lacking robustness to imperfect phase differences.

To address these shortcomings, we propose to enhance our prediction approach with a learnable module, which refines the estimated object velocities, thus allowing VideoPCDNet to accurately and robustly forecast future object states, as well as dealing with heavy occlusions and non-linear motion.

The refinement of velocities is performed by the *Motion Module*, depicted in Fig. 2. This module is an MLP shared among all objects in the scene, which jointly processes a temporal history of object features to produce a refined object velocity. More precisely, the Motion Module receives the estimated object velocities and center of mass from the last time-steps, and outputs a refined velocity prediction $\mathbf{v}^{\text{ref}} = (v_x^{\text{ref}}, v_y^{\text{ref}})$. These refined velocities are then converted into phase differences $\Delta\theta^{\text{ref}}$:

$$\Delta\theta^{\text{ref}} = 2\pi(v_x^{\text{ref}}\mathbf{f}_x + v_y^{\text{ref}}\mathbf{f}_y). \tag{8}$$

The future object states are then predicted through Eq. (7) using the refined phase differences, ensuring that the model captures non-linear motion and performs a robust prediction for every object.

4 Experiments

4.1 Experimental Setup

Datasets: We evaluate VideoPCDNet on two synthetic datasets with varied object appearance and motion: Sprites-MOT and Dynamics-MOT.

Sprites-MOT [11] features 64×64 frames with up to three 11×11 objects, selected from four shapes, moving linearly while entering and leaving the scene. This dataset is used to benchmark unsupervised object-centric tracking [26].

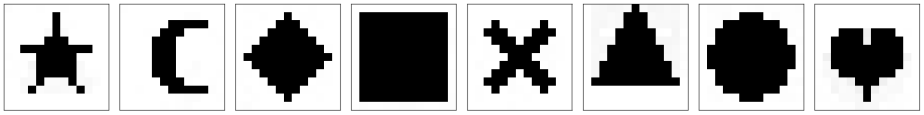


Fig. 3: Learned prototypes on Dynamics-MOT. All objects are discovered.

Dynamics-MOT is a self-generated dataset of 30-frame 64×64 videos. Each sequence features three 11×11 objects, selected from a set of eight shapes, moving and bouncing off image boundaries. To evaluate different motion dynamics, we generate two variants: *Dynamics-MOT Bouncing* features objects moving in linear trajectories, whereas in *Dynamics-MOT Parabolic* the objects follow curved, projectile-like trajectories. These datasets serve as benchmarks to measure VideoPCDNet's ability to model more complex object dynamics.

Training: We train VideoPCDNet in two stages. First, our framework is trained for object-centric decomposition, enabling the model to learn robust object prototypes and masks. Figure 3 shows the object prototypes learned from Dynamics-MOT, where VideoPCDNet discovers all eight distinct shapes present in the dataset. In the second stage, the object prototypes and masks are frozen and the motion module is trained for forecasting future object states—predicting seven future frames given three seed frames. Overall, VideoPCDNet remains lightweight, with only 19,200 learnable parameters. In Appendix A, we analyze and evaluate the interpretability of its internal representations, including object appearance, position and velocity.

4.2 Unsupervised Object Tracking

We evaluate VideoPCDNet for unsupervised object tracking on the Sprites-MOT dataset following the evaluation protocol described in [26] and compare it with multiple object-centric baselines. The results, listed in Table 1, show that VideoPCDNet outperforms competing methods, demonstrating a superior tracking accuracy (MOTA) and precision (MOTP). These findings demonstrate that tracking objects using prototype representations leads to superior accuracy and robustness.

Table 1: Unsupervised object tracking performance on the Sprites-MOT benchmark. Our proposed VideoPCDNet model achieves the best tracking performance. Best two results are highlighted in boldface and underlined, respectively.

Method	MOTA↑	MOTP↑	MD↑	MT↑	Match↑	ID S.↓	FPs↓
VideoPCDNet (ours)	**95.4**	**94.1**	<u>94.6</u>	<u>92.3</u>	**96.1**	1.0	**0.8**
SCALOR [13]	<u>94.9</u>	80.2	**96.4**	**93.2**	<u>95.9</u>	1.7	<u>1.0</u>
ViMON[26]	92.9	<u>91.8</u>	87.7	87.2	95.0	**0.2**	2.1
OP3 [22]	89.1	78.4	92.4	91.8	<u>95.9</u>	<u>0.4</u>	6.8
TBA [11]	79.7	71.2	83.4	80.0	87.8	2.6	8.1
MONet [2]	70.2	89.6	92.4	50.4	75.3	20.3	5.1

Table 2: Ablation Study. We evaluate the effect of each component in VideoPCDNet. Best two results are highlighted in boldface and underlined.

	Method	MOTA↑	MOTP↑	MD↑	MT↑	Match↑	ID S.↓	FPs↓
1	PCDNet [23]	77.8	73.8	90.8	65.8	80.2	15.0	2.4
2	+ object align	93.7	<u>95.1</u>	<u>95.1</u>	88.1	94.6	2.1	<u>1.0</u>
3	+ object state only	84.8	**95.4**	79.6	74.7	85.6	2.4	**0.8**
4	+ two-stage	<u>94.3</u>	94.5	**95.9**	<u>89.1</u>	<u>95.3</u>	<u>1.9</u>	<u>1.0</u>
5	+ external templates	**95.4**	94.1	94.6	**92.3**	**96.1**	**1.0**	**0.8**

4.3 Ablation Study

We evaluate the tracking performance of different VideoPCDNet variants in order to quantify the effect of each component in our framework. The results are listed in Table 2. **(1)** the PCDNet baseline, which naively parses each frame individually into its object components, achieves the weakest tracking performance among all variants. **(2)** Aligning the object decomposition as described in Sec. 3.2 allows VideoPCDNet to consistently track objects across frames, significantly improving the tracking performance. **(3)** Using only the object prototypes present in the state leads to high precision and few false positives. However, it decreases the detection and tracking accuracy as the model cannot correctly represent objects entering the scene. **(4)** The two-stage approach described in Sec. 3.2 addresses this limitation. **(5)** Finally, the full VideoPCDNet, which additionally incorporates external templates by predicting current object states from past observations, achieves the best tracking performance, demonstrating the effectiveness of each enhancement in boosting overall tracking robustness.

4.4 Video Prediction

We further evaluate the capability of VideoPCDNet for modeling object dynamics and predicting future video frames. We compare our method with two

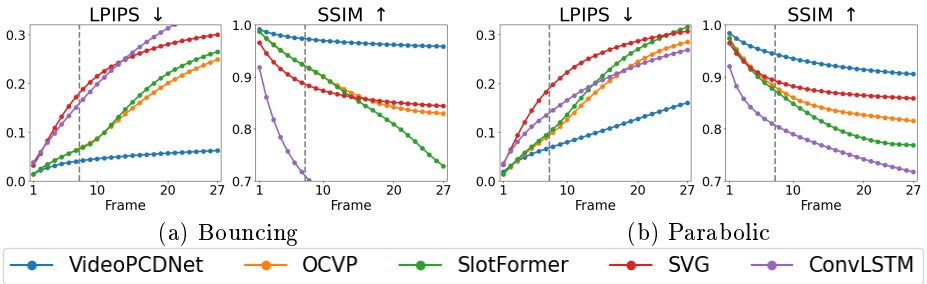
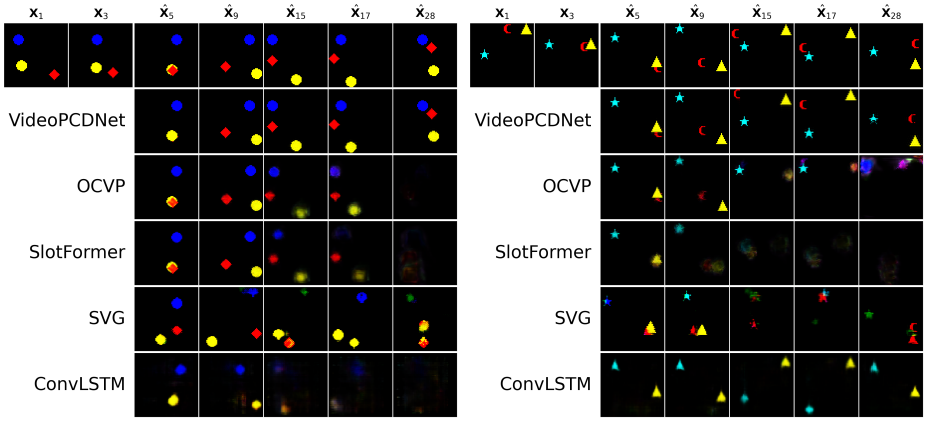


Fig. 4: Video prediction results on the Dynamics-MOT dataset with (a) Bouncing and (b) Parabolic object motion. The vertical bar indicates the prediction horizon used during training. VideoPCDNet outperforms all baselines.



(a) Dynamics-MOT Bouncing (b) Dynamics-MOT Parabolic

Fig. 5: Video prediction rollouts on the Dynamics-MOT dataset with (a) Bouncing and (b) Parabolic object motion. VideoPCDNet accurately models the object motion and predicts up to 28 future frames, whereas the baselines suffer from vanishing objects and prediction artifacts.

holistic video prediction methods: ConvLSTM [20] and SVG [4], as well as two object-centric video prediction baselines: SlotFormer [28] and OCVP [25].

Fig. 4 shows the video prediction results on the Dynamics-MOT dataset with Bouncing (4a) and Parabolic (4b) object motion, respectively. Given three seed frames, we predict the subsequent 27 video frames and measure the SSIM and LPIPS metrics. The results show that object-centric methods, which explicitly model object dynamics (i.e. VideoPCDNet, SlotFormer and OCVP), outperform their holistic counterparts. Furthermore, VideoPCDNet consistently achieves the best performance, especially for longer prediction horizons.

Fig. 5 provides a qualitative comparison illustrating predictions from each method for scenes with bouncing (5a) and parabolic (5b) object motion. In both cases, VideoPCDNet accurately predicts the shape, position, and motion of objects across all time horizons with minimal distortion; whereas the baseline methods progressively degrade in quality, with the non-object-centric SVG and ConvLSTM models exhibiting significant blurriness and visual artifacts in later frames. These results highlight VideoPCDNet's capability to handle complex non-linear object motions while preserving detailed visual and structural information throughout extended prediction periods. Further qualitative video prediction rollouts are shown in Appendix B.

5 Conclusion

We introduced VideoPCDNet, a novel unsupervised framework for object-centric video parsing and prediction that extends phase-correlation networks to the temporal domain. VideoPCDNet decomposes video frames into interpretable object

components, which are represented as transformed versions from a set of learned object prototypes. Our model leverages a two-stage approach for recursively parsing a video sequence into their object components, leading to an accurate and robust object tracking even under challenging motion dynamics and occlusions. Furthermore, VideoPCDNet explicitly models the object motion through phase-correlation techniques in order to efficiently forecast future object states and video frames with less than $20,000$ trainable parameters. In our evaluations, we demonstrate that VideoPCDNet achieves state-of-the-art performance for unsupervised object tracking on the Sprites-MOT benchmark. Furthermore, VideoPCDNet accurately forecasts object states over long prediction horizons, outperforming multiple existing video prediction baselines.

Acknowledgement This work was funded by grant BE 2556/16-2 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG).

References

1. Alba, A., Aguilar-Ponce, R.M., Vigueras-Gómez, J.F., Arce-Santana, E.: Phase correlation based image alignment with subpixel accuracy. In: Mexican International Conference on Artificial Intelligence (MICAI). pp. 171–182. Springer (2012)
2. Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: MONet: Unsupervised scene decomposition and representation. arXiv:1901.11390 (2019)
3. Daniel, T., Tamar, A.: DDLP: Unsupervised object-centric video prediction with deep dynamic latent particles. T. on Machine Learning Research (TMLR) (2024)
4. Denton, E., Fergus, R.: Stochastic video generation with a learned prior. In: International Conference on Machine Learning (ICML) (2018)
5. Engelcke, M., Jones, O.P., Posner, I.: GENESIS-V2: Inferring unordered object representations without iterative refinement. In: International Conference on Neural Information Processing Systems (NeurIPS) (2021)
6. Farazi, H., Behnke, S.: Frequency domain transformer networks for video prediction. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), (2019)
7. Farazi, H., Behnke, S.: Motion segmentation using frequency domain transformer networks. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) (2020)
8. Farazi, H., Behnke, S.: Intention-aware frequency domain transformer networks for video prediction. In: Int. Conf. on Artificial Neural Networks (ICANN) (2022)
9. Farazi, H., Nogga, J., Behnke, S.: Local frequency domain transformer networks for video prediction. In: Int. Joint Conference on Neural Networks (IJCNN) (2021)
10. Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: Int. Conference on Machine Learning (ICML) (2019)
11. He, Z., Li, J., Liu, D., He, H., Barber, D.: Tracking by animation: Unsupervised learning of multi-object attentive trackers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
12. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2**(1), 193–218 (1985)

13. Jiang, J., Janghorbani, S., De Melo, G., Ahn, S.: SCALOR: Generative world models with scalable object representations. In: International Conference on Learning Representations (ICLR) (2019)
14. Kipf, T., Elsayed, G.F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., Greff, K.: Conditional Object-Centric Learning from Video. In: International Conference on Learning Representations (ICLR) (2022)
15. Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Int. Conference on Neural Information Processing Systems (NeurIPS) (2020)
16. Monnier, T., Vincent, E., Ponce, J., Aubry, M.: Unsupervised Layered Image Decomposition into Object Prototypes. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
17. Mosbach, M., Behnke, S.: Fourier-based video prediction through relational object motion. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) (2021)
18. Mosbach, M., Niklas Ewertz, J., Villar-Corrales, A., Behnke, S.: SOLD: Reinforcement learning with slot object-centric latent dynamics. In: International Conference on Machine Learning (ICML) (2025)
19. Nakano, A., Suzuki, M., Matsuo, Y.: Interaction-based disentanglement of entities for object-centric world models. In: International Conference on Learning Representations (ICLR) (2023)
20. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems (NeurIPS) (2015)
21. Stanic, A., Van Steenkiste, S., Schmidhuber, J.: Hierarchical relational inference. In: Conference on Artificial Intelligence (AAAI). pp. 9730–9738 (2021)
22. Veerapaneni, R., Co-Reyes, J.D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., Levine, S.: Entity abstraction in visual model-based reinforcement learning. In: Conference on Robot Learning (CoRL). pp. 1439–1456. PMLR (2020)
23. Villar-Corrales, A., Behnke, S.: Unsupervised image decomposition with phase-correlation networks. In: International Conference on Computer Vision Theory and Applications (VISAPP) (2022)
24. Villar-Corrales, A., Behnke, S.: Playslot: Learning inverse latent dynamics for controllable object-centric video prediction and planning. In: International Conference on Machine Learning (ICML) (2025)
25. Villar-Corrales, A., Wahdan, I., Behnke, S.: Object-centric video prediction via decoupling of object dynamics and interactions. In: IEEE International Conference on Image Processing (ICIP) (2023)
26. Weis, M.A., Chitta, K., Sharma, Y., Brendel, W., Bethge, M., Geiger, A., Ecker, A.S.: Benchmarking unsupervised object representations for video sequences. Journal of Machine Learning Research (JMLR) **22**(183), 1–61 (2021)
27. Wolter, M., Yao, A., Behnke, S.: Object-centered fourier motion estimation and segment-transformation prediction. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN) (2020)
28. Wu, Z., Dvornik, N., Greff, K., Kipf, T., Garg, A.: SlotFormer: Unsupervised visual dynamics simulation with object-centric models. In: International Conference on Learning Representations (ICLR) (2023)
29. Zoran, D., Kabra, R., Lerchner, A., Rezende, D.J.: Parts: Unsupervised segmentation with slots, attention and independence maximization. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10439–10447 (2021)

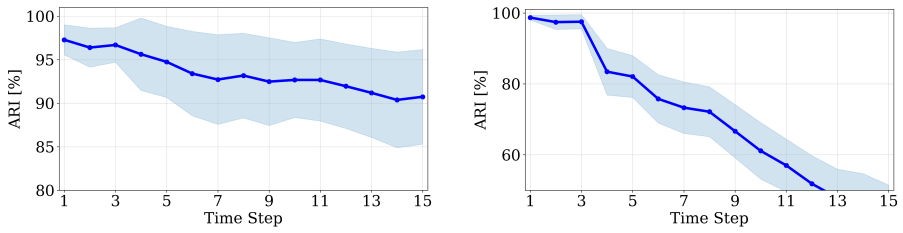
A Interpretability

One of the key properties of VideoPCDNet is its ability to parse dynamic scenes into an interpretable representation of object appearances, color, velocity, and position. In Sec. 4.1, we supported this claim with a qualitative visualization of the learned object prototypes. In this appendix, we complement these visual insights with further qualitative and quantitative evaluations of interpretability. We focus on three indicators of interpretability: the quality of predicted object segmentations (Sec. A.1), the accuracy of predicted object positions (Sec. A.2), and the visualization of complete object trajectories (Sec. A.3).

A.1 Object Segmentations

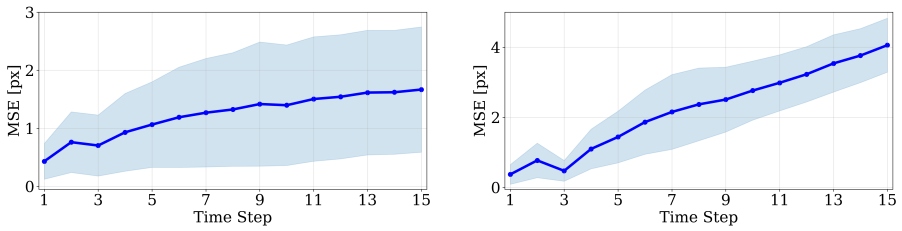
To quantitatively assess the interpretability of the model's object-centric representations, we evaluate the accuracy of the predicted object masks produced by VideoPCDNet using the Adjusted Rand Index (ARI) [12], computed against the ground-truth segmentation masks. ARI is a clustering metric that measures the similarity between two set assignments, ignoring label permutations. It ranges from 0 (random assignment) to 1 (perfect segmentation).

Figure 6 presents the ARI scores of predicted object masks across prediction time-steps, averaged across 300 test sequences. We show the average ARI score as well as its standard deviation. We observe that VideoPCDNet generally achieves high ARI scores throughout the prediction horizon, indicating that object shapes are accurately predicted and their identities are consistently preserved. On the more challenging Dynamics-MOT Parabolic dataset, prediction errors can compound over time, leading to a lower ARI score for longer prediction horizons. The high ARI scores over time indicate that VideoPCDNet produces accurate and robust object-level decomposition in a predictive setting.



(a) Dynamics-MOT Bouncing (b) Dynamics-MOT Parabolic

Fig. 6: Accuracy of predicted object masks over time on the Dynamics-MOT dataset with Bouncing (left) and Parabolic (right) motion. Higher ARI indicates better agreement with ground-truth object segmentation. VideoPCDNet predicts accurate object masks as shown by the strong segmentation consistency across time-steps.



(a) Dynamics-MOT Bouncing (b) Dynamics-MOT Parabolic

Fig. 7: Accuracy of predicted object positions over time on the Dynamics-MOT dataset with Bouncing (left) and Parabolic (right) motion. VideoPCDNet maintains low position error across future time-steps, demonstrating accurate and consistent object-centric motion prediction in both scenarios.

A.2 Object Position

To further assess the interpretability of the learned representations, we evaluate whether VideoPCDNet can accurately model and track object positions over time. In the initial decomposition stage, VideoPCDNet estimates object positions by identifying correlation peaks in the phase difference $\Delta\theta$. During prediction, VideoPCDNet leverages frequency-domain operations and the learned motion module to compute the object velocities $(v_x^{\text{ref}}, v_y^{\text{ref}})$, which are then used to forecast future object positions.

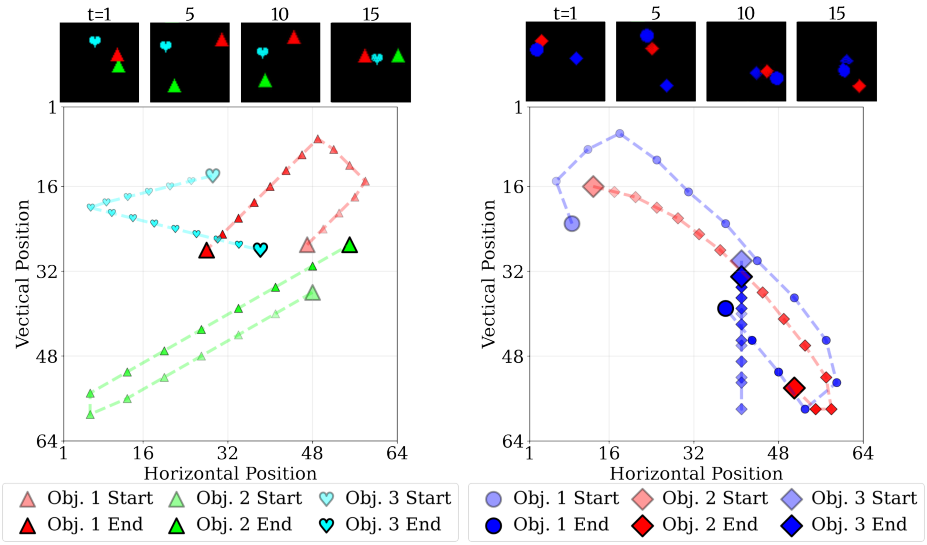
Figure 7 reports the mean squared error between predicted and ground-truth object positions, averaged across all objects and 300 sequences. VideoPCDNet achieves consistently low and stable errors across prediction horizons for both linear and parabolic motion scenarios. This strong performance—despite the absence of explicit object identity and position supervision—demonstrates that our model learns accurate object-specific position representations.

A.3 Visualization of Object Trajectories

To provide a comprehensive view of VideoPCDNet's interpretability, we visualize the complete object trajectories predicted by our model for a prediction horizon of 15 frames. Fig. 8 shows the predicted trajectories for each object in two distinct Dynamics-MOT sequences with linear (Fig. 8a) and parabolic (Fig. 8b) motion.

In both scenarios, VideoPCDNet accurately captures the corresponding motion patterns and correctly predicts the bouncing behavior when objects collide with image boundaries. Notably, in the challenging parabolic motion scenario, the predicted trajectories maintain non-linear paths, accurately capturing both the horizontal velocity component and the gravitational acceleration effect.

These results highlight the interpretability of VideoPCDNet's object-centric approach, where complex scene dynamics are decomposed into meaningful and tractable object trajectories that correspond to intuitive motion patterns.



(a) Dynamics-MOT Bouncing (b) Dynamics-MOT Parabolic

Fig. 8: Predicted trajectories on Dynamics-MOT with Bouncing (left) and Parabolic (right) motion. VideoPCDNet outputs accurate object trajectories over 15 frames, predicting the corresponding motion patterns and the bouncing behavior when objects collide with image boundaries.

B Evaluation on Space-Invaders Dataset

To demonstrate the applicability of our method to datasets with more visually challenging objects, we evaluate VideoPCDNet on the Space-Invaders dataset, which presents more complex visual patterns. We employ two seed frames to predict the subsequent eight.

The Space-Invaders dataset consists of video sequences mimicking the classic Atari game, featuring multiple aliens and a spaceship. We render the dataset with six distinct alien sprites, which move from the top of the frame towards the bottom with linear or parabolic velocity, whereas the spaceship moves sideways. The more complex object appearances and non-linear velocities makes this dataset more challenging that the original Sprites-MOT and Dynamics-MOT benchmarks.

B.1 Learned Prototypes

Fig. 9 demonstrates VideoPCDNet's ability to discover meaningful object prototypes from the Space-Invaders dataset. Our model successfully learns seven distinct prototypes, including six different alien shapes and the spaceship.

The diversity of these learned prototypes demonstrates that VideoPCDNet can effectively decompose the dataset into meaningful object-centric representations, extending beyond the simple geometric shapes of synthetic datasets.

Fig. 9: Object prototypes learned on the Space-Invaders dataset.

B.2 Video Parsing and Prediction

Figs. 10–12 provide qualitative results demonstrating VideoPCDNet's video parsing and prediction capabilities across different sequences from the Space-Invaders dataset. Each figure shows VideoPCDNet's predicted frames and the corresponding predicted semantic segmentation masks. In all sequences, our model successfully maintains object identity and appearance while accurately predicting their trajectories. Furthermore, the semantic segmentation masks demonstrate that VideoPCDNet preserves sharp object boundaries and distinct object identities throughout the prediction horizon, even as objects move across the scene, suffer from overlap, or interact with boundaries.

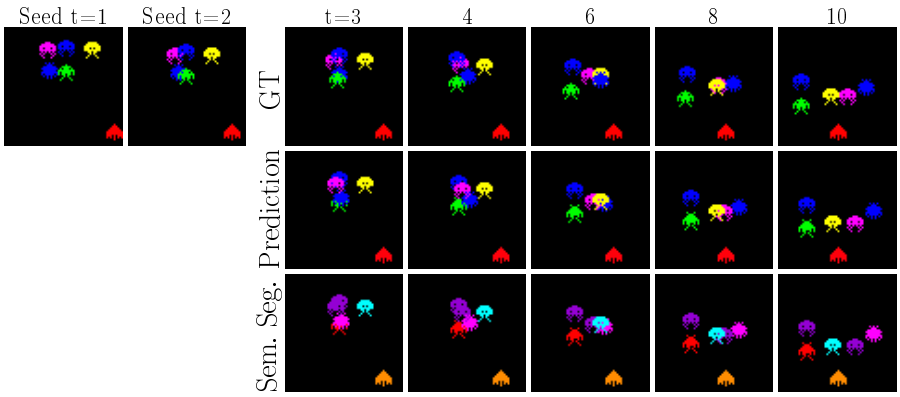


Fig. 10: Qualitative results on Space-Invaders dataset showing ground truth with two seed frames (top), VideoPCDNet predictions (middle), and semantic segmentation (bottom, colors from Fig. 9).

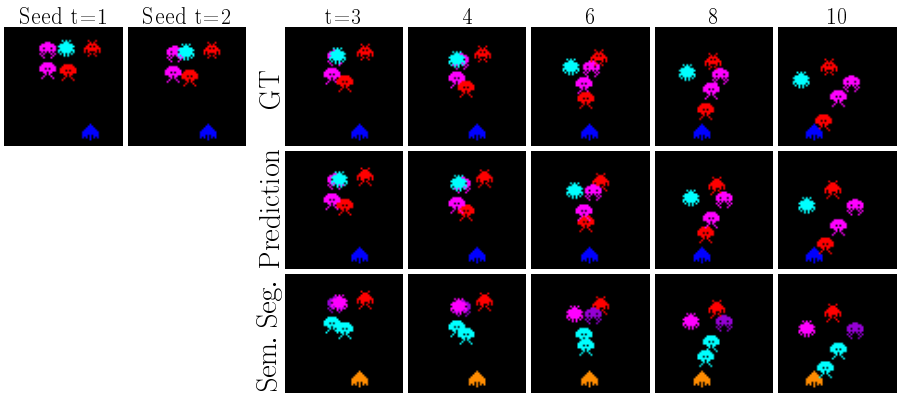


Fig. 11: Qualitative results on Space-Invaders dataset showing ground truth with two seed frames (top), VideoPCDNet predictions (middle), and semantic segmentation (bottom, colors from Fig. 9).

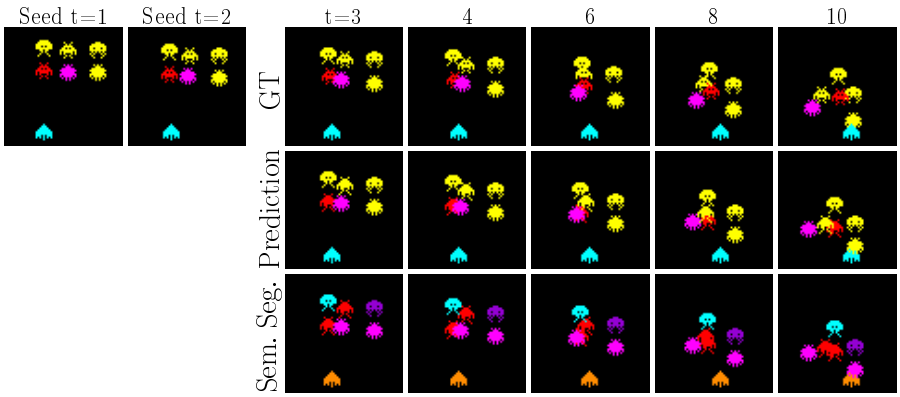


Fig. 12: Qualitative results on Space-Invaders dataset showing ground truth with two seed frames (top), VideoPCDNet predictions (middle), and semantic segmentation (bottom, colors from Fig. 9).