UNSUPERVISED IMAGE DECOMPOSITION WITH PHASE-CORRELATION NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The ability to decompose scenes into their object components is a desired property for autonomous agents, allowing them to reason and act in their surroundings. Recently, different methods have been proposed to learn object-centric representations from data in an unsupervised manner. These methods often rely on latent representations learned by deep neural networks, hence requiring high computational costs and large amounts of curated data. Such models are also difficult to interpret. To address these challenges, we propose the *Phase-Correlation Decomposition Network* (PCDNet), a novel model that decomposes a scene into its object components, which are represented as transformed versions of a set of learned object prototypes. The core building block in PCDNet is the *Phase-Correlation Cell* (PC Cell), which exploits the frequency-domain representation of the images in order to estimate the transformation between an object prototype and its transformed version in the image. In our experiments, we show how PCDNet outperforms state-of-the-art methods for unsupervised object discovery and segmentation on simple benchmark datasets and on more challenging data, while using a small number of learnable parameters and being fully interpretable.

1 INTRODUCTION

Humans understand the world by decomposing scenes into objects that can interact with each other. Analogously, autonomous systems' reasoning and scene understanding capabilities could benefit from decomposing scenes into objects and modeling each of these independently. This approach has been proven beneficial to perform a wide variety of computer vision tasks without explicit supervision, including unsupervised object detection (Eslami et al., 2016), future frame prediction (Weis et al., 2020; Greff et al., 2019), and object tracking (He et al., 2019; Veerapaneni et al., 2020).

Recent works propose extracting object-centric representations without the need for explicit supervision through the use of deep variational auto-encoders (Kingma & Welling, 2014) (VAEs) with spatial attention mechanisms (Burgess et al., 2019; Crawford & Pineau, 2019). However, training these models often presents several difficulties, such as long training times, requiring a large number of trainable parameters, or the need for large curated datasets. Furthermore, these methods suffer from the inherent lack of interpretability characteristic of deep neural networks (DNNs).

To address the aforementioned issues, we propose a novel image decomposition framework – the ***P**hase-**C**orrelation **D**ecomposition **Net**work* (PCDNet). Our method assumes that an image is formed by a superposition of objects, each belonging to one of a finite number of different classes. Following this assumption, the PCDNet decomposes an image into its object components, which are represented as transformed versions of a set of learned object prototypes.

The core building block of the PCDNet framework is the *Phase Correlation Cell* (PC Cell). This is a differentiable module that exploits the frequency-domain representations of an image and a prototype in order to estimate the transformation parameters that best align a prototype to its corresponding object in the image. First, the PC Cell localizes the object prototype in the image by applying the phase-correlation method (Alba et al., 2012), i.e., by finding the peaks in the cross-correlation matrix between the input image and the prototype. Then, the PC Cell aligns the prototype to its corresponding object in the image by performing a phase shift in the frequency domain.

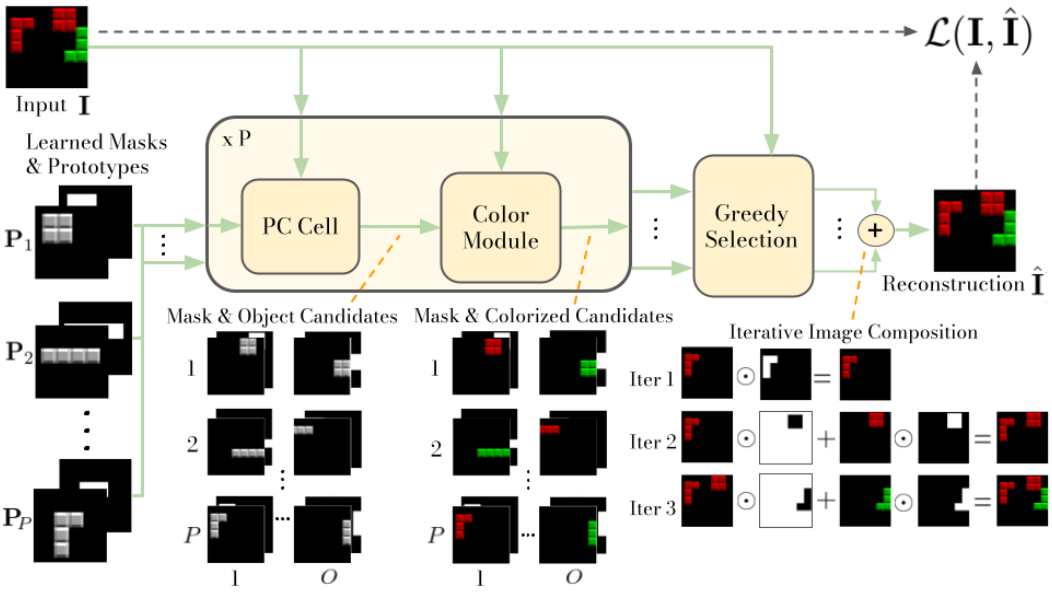


Figure 1: PCDNet decomposition framework. First, the Phase Correlation (PC) Cell estimates the O translation parameters that best align each learned prototype to the objects in the image, and uses them to obtain $(P \times O)$ object and mask candidates. Second, the color module assigns a color to each of the transformed prototypes. Finally, a greedy selection algorithm reconstructs the input image by iteratively combining the colorized object candidates that minimize the reconstruction error.

The PCDNet is trained by first decomposing an image into its object components, and then reconstructing the input by adequately recombining the different components. The strong inductive biases introduced by the network structure and phase-correlation allow our method to learn fully interpretable prototypical object-centric representations without any external supervision while keeping the number of learnable parameters small. Furthermore, our method also disentangles the position and color of each object in a human-interpretable manner.

In summary, the contributions of our work are as follows:

- We propose the PCDNet model, which decomposes an image into its object components, which are represented as transformed versions of a set of learned object prototypes.
- Our proposed model exploits the frequency-domain representation of images so as to disentangle object appearance, position, and color without the need for any external supervision.
- Our experimental results show that our proposed framework outperforms recent methods for joined unsupervised object discovery, image decomposition, and segmentation on benchmark datasets, while significantly reducing the number of learnable parameters, allowing for high throughput, and maintaining interpretability.

2 Related Work

2.1 Object-Centric Representation Learning

The field of representation learning (Bengio et al., 2013) has seen much attention in the last decade, giving rise to great advances in learning hierarchical representations (Paschalidou et al., 2020; Stanic et al., 2021) or in disentangling the underlying factors of variation in the data (Locatello et al., 2019; Burgess et al., 2018). Despite these successes, these methods often rely on learning representations at a scene level, rather than learning in an object-centric manner, i.e., simultaneously learning representations that address multiple, possibly repeating, objects.

In the last few years, several methods have been proposed to perform object-centric image decomposition in an unsupervised manner.

A first approach to object-centric decomposition combines VAEs with attention mechanisms to decompose a scene into object-centric representations. The object representations are then decoded to reconstruct the input image. These methods can be further divided into two different groups depending on the class of latent representations used. On the one hand, some methods (Eslami et al., 2016; Kosiorek et al., 2018; Stanic et al., 2021; He et al., 2019) explicitly encode the input into factored latent variables, which represent specific properties such as appearance, position, and presence. On the other hand, other models (Burgess et al., 2019; Weis et al., 2020; Locatello et al., 2020) decompose the image into unconstrained per-object latent representations.

Recently, several proposed methods (Greff et al., 2019; Engelcke et al., 2020; 2021; Veerapaneni et al., 2020; Lin et al., 2020) exploit parameterized spatial mixture models combined with variational inference in order to decode object-centric latent variables.

Despite these recent advances in unsupervised object-centric learning, most existing methods rely on deep networks and employ expensive attention mechanisms to encode the input images into latent representations, hence requiring a large number of learnable parameters and high computational costs. Furthermore, these approaches also suffer from the inherent lack of interpretability characteristic of DNNs. Our proposed method exploits the strong inductive biases introduced by phase correlation in order to decompose an image into object-centric components without the need for deep encoders, hence using only a small number of learnable parameters, and being fully interpretable.

2.2 Layered Models

The idea of representing an image as a superposition of different layers has been studied since the introduction of the 'dead leaves' model by Matheron (1968). This model has been extended to handle natural images and scale-invariant representations (Lee et al., 2001), or video sequences (Jojic & Frey, 2001). More recently, several works (Yang et al., 2017; Lin et al., 2018; Zhang et al., 2020; Aksoy et al., 2017; Arandjelović & Zisserman, 2019; Sbai et al., 2020) combine deep neural networks and ideas from layered image formation for different generative tasks, such as editing or image composition. However, the aforementioned approaches are limited to foreground/background layered decomposition, or to represent the images with a small number of layers.

The work most similar to ours was recently presented by Monnier et al. (2021). The authors propose a model to decompose an image into overlapping layers, each containing an object from a predefined set of categories. The object layers are obtained with a cascade of spatial transformer networks, which learn transformations that align certain object sprites to the input image.

While we also follow a layered image formation, our PCDNet model is not limited to a small number of layers, hence being able to represent scenes with multiple objects. PCDNet represents each object in its own layer, and uses learned alpha masks to model occlusions and superposition between layers.

2.3 Frequency-Domain Neural Networks

Signal analysis and manipulation in the frequency domain is one of the most widely used tools in the field of signal processing (Proakis & Manolakis, 2004). However, frequency-domain methods are not nearly as developed for solving computer vision tasks with neural networks. They mostly focus on specific applications such as model compression (Xu et al., 2020; Gueguen et al., 2018), image super-resolution and denoising (Fritsche et al., 2019; Villar-Corrales et al., 2021; Kumar et al., 2017), or accelerating the calculation of convolutions (Mathieu et al., 2014; Ko et al., 2017).

In recent years, a particular family of frequency-domain neural networks—the *phase-correlation networks*—has received interest from the research community and has shown promise for tasks such as future frame prediction (Farazi et al., 2021; Wolter et al., 2020) or motion segmentation (Farazi & Behnke, 2020). Phase-correlation networks compute normalized cross-correlations in the frequency domain and operate with the phase of complex signals in order to estimate motion and transformation parameters between consecutive frames, which can be used to obtain accurate future frame predictions requiring few learnable parameters. Despite these recent successes, phase-correlation networks remain unexplored beyond the tasks of video prediction and motion estimation. Our proposed method presents a first attempt at applying phase correlation networks for the tasks of scene decomposition and unsupervised object-centric representation learning.

3 PHASE-CORRELATION DECOMPOSITION NETWORK

In this section, we present a novel image decomposition model: PCDNet. Given an input image $\mathbf{I}$, PCDNet aims at its decomposition into O independent objects $\mathcal{O} = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_O\}$. In this work, we assume that these objects belong to one out of a finite number P of classes, and that there is a known upper bound to the total number of objects present in an image ($O_{\max}$).

Inspired by recent works in prototypical learning and clustering (Li et al., 2021; Monnier et al., 2020), we design our model such that the objects in the image can be represented as transformed versions of a finite set of object prototypes $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_P\}$. Each object prototype $\mathbf{P}_i \in \mathbb{R}^{H,W}$ is learned along with a corresponding alpha mask $\mathbf{M}_i \in \mathbb{R}^{H,W}$, which is used to model occlusions and superposition of objects. Throughout this work, we consider object prototypes to be in gray-scale and of smaller size than the input image. PCDNet simultaneously learns suitable object prototypes, alpha masks and transformation parameters in order to accurately decompose an image into object-centric components.

An overview of the PCDNet framework is displayed in Figure 1. First, the *PC Cell* (Section 3.1) estimates the candidate transformation parameters that best align the object prototypes to the objects in the image, and generates object candidates based on the estimated parameters. Second, a *Color Module* (Section 3.2) transforms the object candidates by applying a learned color transformation. Finally, a *greedy selection algorithm* (Section 3.3) reconstructs the input image by iteratively selecting the object candidates that minimize the reconstruction error.

3.1 PHASE-CORRELATION CELL

The first module of our image decomposition framework is the PC Cell, as depicted in Figure 1. This module first estimates the regions of an image where a particular object might be located, and then shifts the prototype to the estimated object location. Inspired by traditional image registration methods (Reddy & Chatterji, 1996; Alba et al., 2012), we adopt an approach based on phase correlation. This method estimates the relative displacement between two images by computing the normalized cross-correlation in the frequency domain.

Given an image $\mathbf{I}$ and an object prototype $\mathbf{P}$, the PC Cell first transforms both inputs into the frequency domain using the *Fast Fourier Transform* (FFT, $\mathcal{F}$). Second, it computes the phase differences between the frequency representations of image and prototype, which can be efficiently computed as an element-wise division in the frequency domain. Then, a localization matrix $\mathbf{L}$ is found by applying the inverse FFT ($\mathcal{F}^{-1}$) on the normalized phase differences:

$$\mathbf{L} = \mathcal{F}^{-1}\Big(\frac{\mathcal{F}(\mathbf{I}) \odot \overline{\mathcal{F}(\mathbf{P})}}{||\mathcal{F}(\mathbf{I}) \odot \overline{\mathcal{F}(\mathbf{P})}|| + \epsilon}\Big), \tag{1}$$

where $\overline{\mathcal{F}(\mathbf{P})}$ denotes the complex conjugate of $\mathcal{F}(\mathbf{P})$, $\odot$ is the Hadamard product, $||\cdot||$ is the modulus operator, and ϵ is a small constant to avoid division by zero. Finally, the estimated relative pixel displacement ($\boldsymbol{\delta}_{x,y} = (\delta_x, \delta_y)$) can then be found by locating the correlation peak in $\mathbf{L}$:

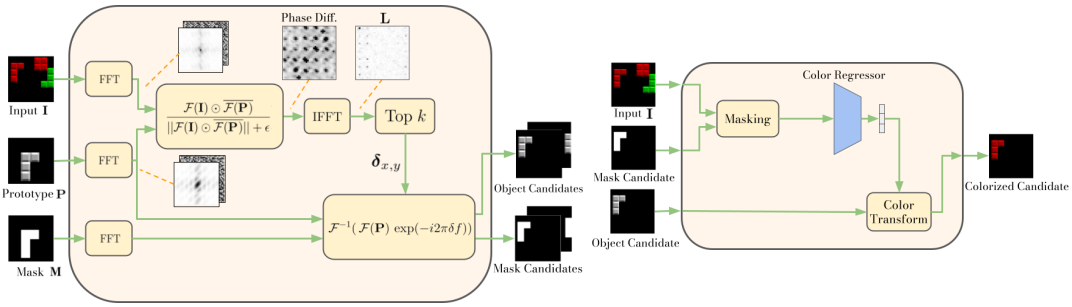
$$\boldsymbol{\delta}_{x,y} = \arg\max(\mathbf{L}) \ . \tag{2}$$

In practical scenarios, we do not know in advance which objects are present in the image or whether there are more than one objects from the same class. To account for this uncertainty, we pick the largest $O_{\max}$ correlation values from $\mathbf{L}$ and consider them as candidate locations for an object.

Finally, given the estimated translation parameters, the PC Cell relies on the Fourier shift theorem in order to align the object prototypes and the corresponding alpha masks to the objects in the image. Given the translation parameters δ_x and δ_y, an object prototype is shifted using

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

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



(a) PC Cell

(b) Color Module

Figure 2: **(a)**: Inner structure of the PC Cell. First, the translation parameters are estimated by finding the correlation peaks between the object prototype and the input image. Second, the prototype is shifted by phase shifting in the frequency domain. **(b)**: The *Color Module* estimates color parameters from the input image and aligns the color channels of a translated object prototype.

Figure 2a depicts the inner structure of the PC Cell, illustrating each of the phase correlation steps and displaying some intermediate representations, including the magnitude and phase components of each input, the normalized cross-correlation matrix, and the localization matrix $\mathbf{L}$.

3.2 Color Module

The PC Cell module outputs translated versions of the object prototypes and their corresponding alpha masks. However, these translated templates need not match the color or texture of the object represented in the image. This issue is solved by the *Color Module*, which is illustrated in Figure 2b. It learns color parameters from the input image, and transforms the translated prototypes according to the estimated color parameters.

Given the input image and the translated object prototype and mask, the color module first obtains a masked version of the image containing only the relevant object. This is achieved through an element-wise product of the image with the translated alpha mask. The masked object is fed to a convolutional network, which learns the color parameters (one for gray-scale and three for RGB images). Finally, these learned color parameters are applied to the translated object prototypes with a channel-wise affine transform. Further details about the color module are given in Appendix B.2.

3.3 Greedy Selection

The PC Cell and color modules produce $T = O_{\max} \times P$ translated and colorized candidate objects ($\mathcal{T} = \{\mathbf{T}_1, ..., \mathbf{T}_T\}$) and their corresponding translated alpha masks ($\{\mathbf{M}_1, ..., \mathbf{M}_T\}$). The final module of the PCDNet framework selects, among all candidates, the objects that minimize the reconstruction error with respect to the input image.

The number of possible object combinations grows exponentially with maximum number of objects and the number of object candidates ($T^{O_{\max}}$), which quickly makes it infeasible to evaluate all possible combinations. Therefore, similarly to (Monnier et al., 2021), we propose a greedy algorithm that selects in a sequential manner the objects that minimize the reconstruction loss. The greedy nature of the algorithm reduces the number of possible object combinations to $T \times O_{\max}$, hence scaling to images with a large number of objects and prototypes.

The greedy object selection algorithm operates as follows. At the first iteration, we select the object that minimizes the reconstruction loss with respect to the input image, and add it to the list of selected objects. Then, for each subsequent iteration, we greedily select the object that, combined with the previously selected ones, minimizes the reconstruction error. The reconstruction error is computed using Equation (4), which corresponds to the mean squared error between the input image ($\mathbf{I}$) and a combination of the selected candidates ($\mathcal{G}(\mathcal{T})$).

The objects are combined recursively in an overlapping manner, as shown in Equation (5), so that the first selected object ($\mathbf{T}_1$) corresponds to the one closest to the viewer, whereas the last selected

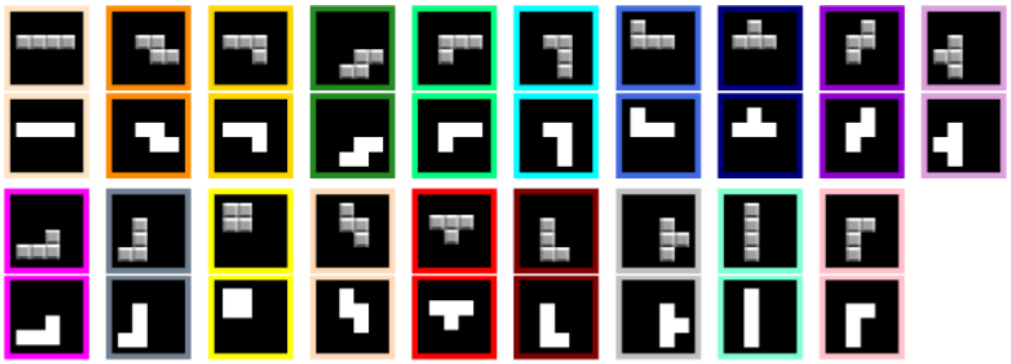


Figure 3: Object prototypes (top) and segmentation alpha masks (bottom) learned on the Tetrominoes dataset. Our model is able to discover in an unsupervised manner all 19 pieces.

object ($\mathbf{T}_O$) is located the furthest from the viewer:

$$\mathbf{E}(\mathbf{I}, \mathcal{T}) = ||\mathbf{I} - \mathcal{G}(\mathcal{T})||_2^2 \tag{4}$$

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

An example of this image composition is displayed in Figure 1. This reconstruction approach inherently models relative depths, hence allowing for a simple, yet effective, modeling of the occlusions between objects.

3.4 Training and Implementation Details

We train PCDNet in an end-to-end manner to reconstruct an image as a combination of transformed object prototypes. The training is performed by minimizing the mean squared error loss function:

$$\mathcal{L}(\mathbf{I}, \mathcal{T}') = ||\mathbf{I} - \mathcal{G}(\mathcal{T}')||_2^2, \tag{6}$$

where $\mathcal{T}'$ corresponds to the object candidates selected by greedy selection algorithm. Namely, minimizing Equation (6) decreases the reconstruction error between the combination of selected object candidates ($\mathcal{G}(\mathcal{T}')$) and the input image.

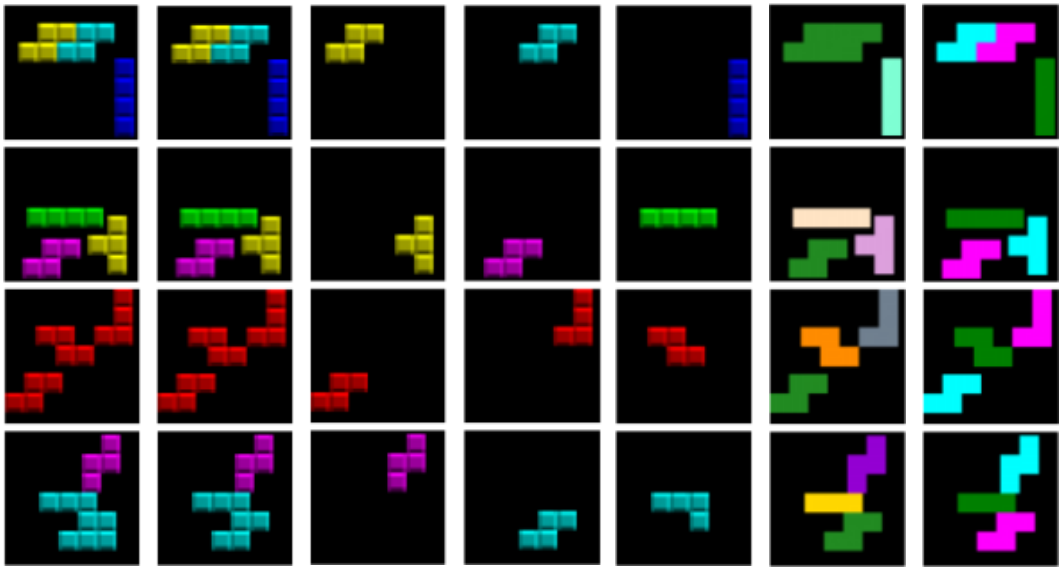
In our experiments, we noticed that the initialization and update strategy of the object prototypes is of paramount importance for the correct performance of the PCDNet model. The prototypes are initialized with a small constant value (e.g., 0.2), whereas the center pixel is assigned an initial value of one, enforcing the prototypes to emerge centered in the frame.

During the first training iterations, we notice that the greedy algorithm selects some object prototypes with a higher frequency that others, hence learning much faster. In practice, this prevents other prototypes from learning relevant object representations, since they are not updated often enough. To reduce the impact of uneven prototype discovery, we add, with a certain probability, some uniform random noise to the prototypes during the first training iterations. This prevents the greedy algorithm from always selecting, and hence updating, the same object prototypes and masks.

In datasets with a background, we add a special prototype to model a static background. In these cases, the input images are reconstructed by overlapping the objects selected by the greedy algorithm on top of the background prototype. This background prototype is initialized by averaging the images from the training set, and its values are refined during training.

4 Experimental Results

In this section, we quantitatively and qualitatively evaluate our PCDNet framework for the tasks of unsupervised object discovery and segmentation. PCDNet is implemented in Python using the PyTorch framework (Paszke et al., 2017). We train our models using the Adam (Kingma & Ba,



(a) (b) (c) (d) (e) (f) (g)

Figure 4: Qualitative decomposition and segmentation results on the Tetrominoes dataset. Last row shows a failure case. **(a)**: Original image. **(b)**: PCDNet Reconstruction. **(c)-(e)**: Colorized transformed object prototypes. **(f)**: Semantic segmentation masks. Colors correspond to the prototype frames in Figure 3. **(g)**: Instance segmentation masks.

Table 1: Object discovery evaluation results on the Tetrominoes dataset. PCDNet outperforms SOTA methods, while using a small number of learned parameters. Moreover, our PCDNet has the highest throughput out of all evaluated methods. For each metric, the best result is highlighted in boldface, whereas the second best is underlined.

Model	**ARI (%)** ↑	**Params** ↓	**Imgs/s** ↑
Slot MLP (Locatello et al., 2020)	35.1	–	–
Slot Attention (Locatello et al., 2020)	<u>99.5</u>	<u>229,188</u>	1.48
ULID (Monnier et al., 2021)	**99.6**	659,755	<u>52.3</u>
IODINE (Greff et al., 2019)	99.2	408,036	11.5
PCDNet (ours)	**99.6**	**28,130**	**59.6**

2015) optimizer with an initial learning rate of $3 \cdot 10^{-3}$. A more detailed report of the used hyperparameters is given in Appendix B [1].

4.1 Tetrominoes Dataset

We evaluate PCDNet for image decomposition and object discovery on the Tetrominoes dataset (Greff et al., 2019). This dataset contains 60.000 training images and 320 test images of size 35×35, each composed of three non-overlapping Tetris-like sprites over a black background. The sprites belong to one out of 19 possible configurations and have one of six random colors.

Figure 3 displays the 19 learned object prototypes and their corresponding alhpa masks from the Tetrominoes dataset. We clearly observe how PCDNet accurately discovers the shape of the different pieces and their tiled texture.

Figure 4 depicts qualitative results for unsupervised object detection and segmentation. In the first three rows, PCDNet successfully decomposes the images into their object components and precisely segments the objects into semantic and instance masks. The bottom row shows an example in which the greedy selection algorithm leads to a failure case.

[1] Source code and pretrained models will be made publicly available upon acceptance of this paper.

Figure 5: Object prototypes learned on the Space Invaders dataset. PCDNet discovers prototypes corresponding to the different elements from the game, including aliens, lasers and ships.

For a fair quantitative comparison with previous works, we evaluate our PCDNet model for object segmentation using the Adjusted Rand Index (Hubert & Arabie, 1985) (ARI) on the ground truth foreground pixels. ARI is a clustering metric that measures the similarity between two set assignments, ignoring label permutations, and ranges from 0 (random assignment) to 1 (perfect clustering).

Table 1 summarizes the evaluation results for object discovery on the Tetrominoes dataset. We compare the performance of our approach with several existing methods: Slot MLP and Slot Attention (Locatello et al., 2020), IODINE (Greff et al., 2019) and Unsupervised Layered Image Decomposition (Monnier et al., 2021) (labeled as ULID in Table 1). The listed results are reproduced using open-source code or taken from the original publication.

From Table 1, we see how our PCDNet model outperforms SOTA models, achieving 99.6% ARI on the Tetrominoes dataset. PCDNet uses only a small percentage of learnable parameters compared to other methods (e.g., only 6% of the parameters from IODINE), and has the highest inference throughput (images/s). Additionally, unlike other approaches, PCDNet obtains disentangled representations for the object appearance, position, and color in a human-interpretable manner.

4.2 Space Invaders Dataset

In this experiment, we use replays from humans playing the Atari game *Space Invaders*, extracted from the Atari Grand Challenge dataset (Kurin et al., 2017). PCDNet is trained to decompose the Space Invaders images into 50 objects, belonging to one of 14 learned prototypes of size 20×20.

Figure 6 depicts a qualitative comparison between our PCDNet model with SPACE (Lin et al., 2020) and Slot Attention (Locatello et al., 2020). Slot Attention achieves an almost perfect reconstruction of the input image. However, it fails to decompose the image into its object components, uniformly scattering the object representations across different slots. SPACE successfully decomposes the image into different object components. Nevertheless, the reconstructions appear severely blurred and several objects are not correct. PCDNet achieves the best results among all compared methods. Our model successfully decomposes the input image into accurate object-centric representations. Additionally, PCDNet learns semantic understanding of the objects. Figure 6 depicts a segmentation of an image from the Space Invaders dataset. Further qualitative results on the Space Invaders dataset are reported in Appendix C.

4.3 NGSIM Dataset

In this third experiment, we apply our PCDNet model to discover vehicle prototypes from real traffic camera footage from the Next Generation Simulation (NGSIM) dataset (NGS). We decompose each frame into 25 different objects, belonging to one of 12 learned vehicle prototypes.

Figure 7 depicts qualitative results on the NGSIM dataset. Despite not reconstructing all objects present in the images, we see how the PCDNet model is applicable to real-world data, learning prototypes and masks for different types of vehicles. Interestingly, we also notice how the PCDNet model learns the car shade as part of the object prototype. This is a reasonable observation, since the shades are projected towards the bottom of the image through the whole duration of the video.

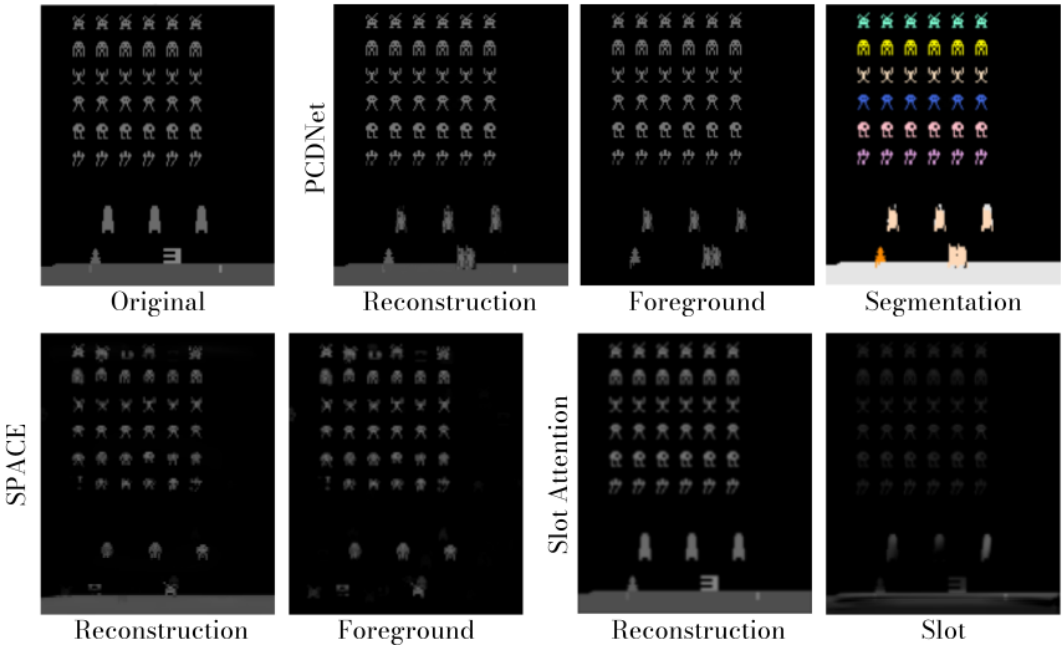


Figure 6: Comparison of different object-centric models on the Space Invaders dataset. PCDNet is the only one among the compared methods which successfully decomposes the input image into accurate object components, and that has semantic knowledge of the object representations. The color of each object corresponds to the frame of the corresponding prototype from Figure 5.

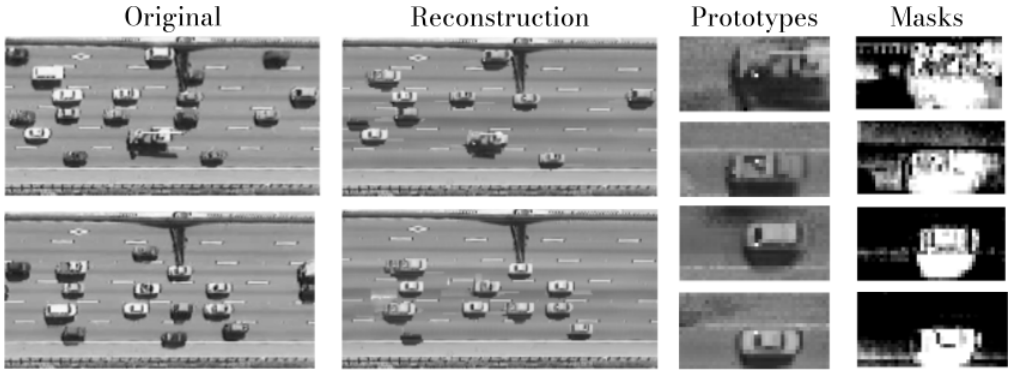


Figure 7: Object discovery on the NGSIM dataset. PCDNet learns prototypes and masks for different types of vehicles in an unsupervised manner.

5 Conclusion

We proposed PCDNet, a novel image composition model that decomposes an image, in a fully unsupervised manner, into its object components, which are represented as transformed versions of a set of learned object prototypes. PCDNet exploits the frequency-domain representation of images to estimate the translation parameters that best align the prototypes to the objects in the image. The structured network used by PCDNet allows for an interpretable image decomposition, which disentangles object appearance, position and color without any external supervision. In our experiments, we show how our proposed model outperforms existing methods for unsupervised object discovery and segmentation on a benchmark synthetic dataset, while significantly reducing the number of learnable parameters, having a superior throughput, and being fully interpretable. Furthermore, we also show that the PCDNet model can also be applied for unsupervised prototypical object discovery on more challenging synthetic and real datasets. We hope that our work paves the way towards further research on phase correlation networks for unsupervised object-centric representation learning.

REFERENCES

Next generation simulation (NGSIM). https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm. Accessed: 2021-06-19.

Yağiz Aksoy, Tunç Ozan Aydin, Aljoša Smolić, and Marc Pollefeys. Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)*, 36(2):1–19, 2017.

Alfonso Alba, Ruth M Aguilar-Ponce, Javier Flavio Vigueras-Gómez, and Edgar Arce-Santana. Phase correlation based image alignment with subpixel accuracy. In *Mexican International Conference on Artificial Intelligence*, pp. 171–182. Springer, 2012.

Relja Arandjelović and Andrew Zisserman. Object discovery with a copy-pasting GAN. *arXiv:1905.11369*, 2019.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *CoRR*, abs/1804.03599, 2018.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv:1901.11390*, 2019.

Eric Crawford and Joelle Pineau. Spatially invariant unsupervised object detection with convolutional neural networks. In *AAAI Conference on Artificial Intelligence*, volume 33, pp. 3412–3420, 2019.

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

Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *arXiv:2104.09958*, 2021.

SM Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2016.

Hafez Farazi and Sven Behnke. Motion segmentation using frequency domain transformer networks. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2020.

Hafez Farazi, Jan Nogga, and Sven Behnke. Local frequency domain transformer networks for video prediction. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2021.

Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3599–3608, 2019.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning (ICML)*, pp. 2424–2433, 2019.

Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from JPEG. *International Conference on Neural Information Processing Systems (NeurIPS)*, 31:3933–3944, 2018.

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

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2015.

Nebojsa Jojic and Brendan J Frey. Learning flexible sprites in video layers. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (CVPR)*, 2001.

Diederik P Kingma and Jimmy Ba. A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Jong Hwan Ko, Burhan Mudassar, Taesik Na, and Saibal Mukhopadhyay. Design of an energy-efficient accelerator for training of convolutional neural networks using frequency-domain computation. In *54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017.

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

Adam R Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E Hinton. Stacked capsule autoencoders. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Neeraj Kumar, Ruchika Verma, and Amit Sethi. Convolutional neural networks for wavelet domain super resolution. *Pattern Recognition Letters*, 90:65–71, 2017.

Vitaly Kurin, Sebastian Nowozin, Katja Hofmann, Lucas Beyer, and Bastian Leibe. The Atari grand challenge dataset. *arXiv:1705.10998*, 2017.

Ann B Lee, David Mumford, and Jinggang Huang. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision*, 41 (1):35–59, 2001.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations (ICLR)*, 2021.

Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9455–9464, 2018.

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

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, pp. 4114–4124, 2019.

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

Georges Matheron. Schéma booléen séquentiel de partition aléatoire. *N-83 CMM, Paris School of Mines publications*, 1968.

Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through FFTs. *International Conference on Learning Representations (ICLR)*, 2014.

Tom Monnier, Thibault Groueix, and Mathieu Aubry. Deep transformation-invariant clustering. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. *arXiv:2104.14575*, 2021.

Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3D objects from a single RGB image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1060–1070, 2020.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. *International Conference on Neural Information Processing Systems Workshops (NeurIPS-W)*, 2017.

John G Proakis and Dimitris G Manolakis. Digital signal processing. *PHI Publication: New Delhi, India*, 2004.

B Srinivasa Reddy and Biswanath N Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

Othman Sbai, Camille Couprie, and Mathieu Aubry. Unsupervised image decomposition in vector layers. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1576–1580. IEEE, 2020.

Aleksandar Stanic, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Hierarchical relational inference. In *AAAI Conference on Artificial Intelligence*, pp. 9730–9738, 2021.

Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning (CoRL)*, pp. 1439–1456. PMLR, 2020.

Angel Villar-Corrales, Franziska Schirrmacher, and Christian Riess. Deep learning architectural designs for super-resolution of noisy images. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1635–1639, 2021.

Marissa A Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S Ecker. Unmasking the inductive biases of unsupervised object representations for video sequences. *arXiv:2006.07034*, 2020.

Moritz Wolter, Angela Yao, and Sven Behnke. Object-centered fourier motion estimation and segment-transformation prediction. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2020.

Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1740–1749, 2020.

Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. LR-GAN: Layered recursive generative adversarial networks for image generation. *arXiv:1703.01560*, 2017.

Yang Zhang, Ivor W Tsang, Yawei Luo, Chang-Hui Hu, Xiaobo Lu, and Xin Yu. Copy and paste GAN: Face hallucination from shaded thumbnails. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7355–7364, 2020.

Table 2: Summary of the datasets used in the paper. Results marked with an asterisk (*) correspond to quantities not specified in the dataset, hence defined by ourselves.

Dataset	**Img. Size**	**Train/Eval**	$O_{\max}$	P	**Proto. Size**
Tetrominoes	35×35	60,000/320	3	19	25×25
NGSIM	98×194	681/-	25*	12*	20×40*
Space Invaders	185×160	69,961/-	50*	14*	20×20*

Table 3: Implementation details of the *Color Module* CNN

Layer	**Size/Ch.**	**Comments**
Input	3	Masked input image
Conv. (3×3) + ReLU	12	Stride=1, Padding=0
Batch Norm.	12	
Conv. (3×3) + ReLU	12	Stride=1, Padding=0
Batch Norm.	12	
Average Pooling	12	
Flatten		
Fully Connected	3	RGB parameters

A Dataset Summary

In this section, we give additional details about the different datasets used throughout this paper. A comprehensive summary is reported in Table 2.

For unsupervised object discovery, we qualitatively and quantitatively evaluate our PCDNet framework on the popular Tetrominoes (Greff et al., 2019) dataset. This collection contains 60,000 training images and 320 test images of size 35×35. Each image is composed of three non-overlapping Tetris-like sprites over a black background. In particular, the sprites belong to one out of 19 possible configurations, and have one out of six colors.

Additionally, we evaluate our model on sequences of the Atari Grand Challenge (Kurin et al., 2017) and the NGSIM (NGS) datasets. Neither of these datasets includes annotated segmentation masks nor bounding boxes, hence we restrict our analysis to a qualitative one.

The Atari Grand Challenge dataset (Kurin et al., 2017) is a large collection of human replays of different popular Atari games. From these, we select several replays corresponding to the game *Space Invaders*. More precisely, we select 69,961 random images of size (185×160). We train our PCDNet model to decompose these images into 50 different objects, belonging to one of the 14 learned prototypes of size (20×20).

The NGSIM dataset (NGS) is a database of driving trajectory and behavioral data, including traffic camera footage on different US Highways. In our work, we use our PCDNet model in order to discover vehicle prototypes from video sequences. Different NGSIM videos display footage of different highways and from several perspectives; therefore, we apply our model to individual videos. The length of the NGSIM video used for testing PCDNet contains 681 frames. We decompose each frame into 25 different objects, belonging to one of 12 learned vehicle prototypes of size (20×40).

B Model and Hyper-Parameter Details

B.1 Training Details

Unless otherwise specified, we train all our experiments with an NVIDIA RTX 3090 GPU with 24 GB RAM using the Adam (Kingma & Ba, 2015) update rule with an initial learning rate of $3 \cdot 10^{-3}$. We use a linear scheduler to reduce the learning rate by a factor of three after every fifth epoch. On the Tetrominoes dataset, we use a batch size of 64 images, whereas on the NGSIM and Space Invaders datasets we use a batch size of four images.

Algorithm 1 Greedy Selection Algorithm

procedure GREEDY SELECTION ALGORITHM
Inputs:
$\mathbf{I} \leftarrow$ *Input image*
$\mathcal{T} = [\mathbf{T}_1, ..., \mathbf{T}_T] \leftarrow$ *Colorized Object Candidates*
$O_{\max} \leftarrow$ *Maximum number of objects in the image*
Returns:
$\mathcal{O} = [\mathbf{O}_1, ..., \mathbf{O}_{O_{\max}}] \leftarrow$ *Selected objects to reconstract the input*
Algorithm:
$\mathcal{O} \leftarrow [\,]$
for o in range [1, $O_{\max}$] **do**
  $\mathbf{E} \leftarrow [\,]$
  for t in range [1, T] **do**
    $\mathbf{E}_t = ||\mathbf{I} - \mathcal{G}(\mathcal{O}, \mathbf{T}_t)||_2^2$
  $q = \arg\min(\mathbf{E})$
  $\mathcal{O} \leftarrow [\mathcal{O}, \mathbf{T}_q]$
return $\mathcal{O}$

The object prototypes are initialized with a constant value of 0.2 and with the center pixel set to one. This enforces the object prototypes to emerge centered. To prevent the greedy algorithm from always selecting the same prototypes during the first iterations, we add uniform random noise $\mathcal{U}[-0.5, 0.5)$ to the prototypes with a probability of 80%.

B.2 COLOR MODULE

The color module, depicted in Figure 2b in the paper, is implemented in a similar fashion to a Spatial Transformer Network (Jaderberg et al., 2015) (STN). The masked image is fed to a neural network, which extracts certain color parameters corresponding to the masked object. The architecture of this network is summarized in Table 3. The extracted color parameters are applied to the translated object prototypes with a channel-wise affine transform. Our color module shares similarities with other color transformation approaches (Monnier et al., 2020; Kosiorek et al., 2019). However, despite applying the same affine channel transform, our method differs in the way the color parameters are computed.

B.3 GREEDY SELECTION ALGORITHM

Algorithm 1 illustrates the greedy selection algorithm used to select the colorized object candidates that best reconstruct the input image.

C QUALITATIVE RESULTS

In this section, we display further qualitative results for some of the evaluated datasets.

Figure 8 depicts the object prototypes learned by PCDNet on the Tetrominoes dataset. Figure 9 depicts the segmentation mask for each of the learned prototypes. It is shown in both figures how PCDNet precisely learns the shapes of all the different pieces as well as their tiled texture. Figure 10 depicts further qualitative results for unsupervised object detection and segmentation on the Tetrominoes dataset. We clearly see how PCDNet selects the correct learned prototypes, estimates the precise locations in the image, and adds the corresponding colors.

Figure 11 shows the learned object prototypes and masks on the Space Invaders datasets. PCDNet is able to discover different elements from the game, including aliens, lasers and ships. The learned masks allow to handle occlusions and superposition of objects. Figures 12 and 13 depict further qualitative comparisons on the Space Invaders dataset between PCDNet, SPACE Lin et al. (2020) and Slot Attention Locatello et al. (2020).

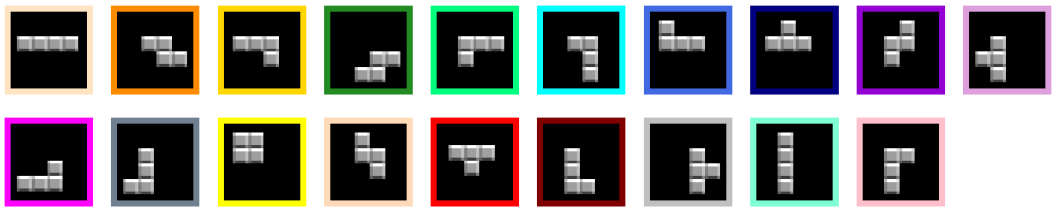


Figure 8: Object prototypes learned on the Tetrominoes dataset. Our model is able to discover in an unsupervised manner all 19 pieces.

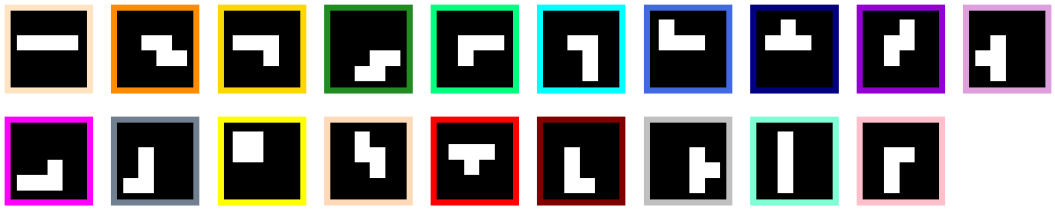
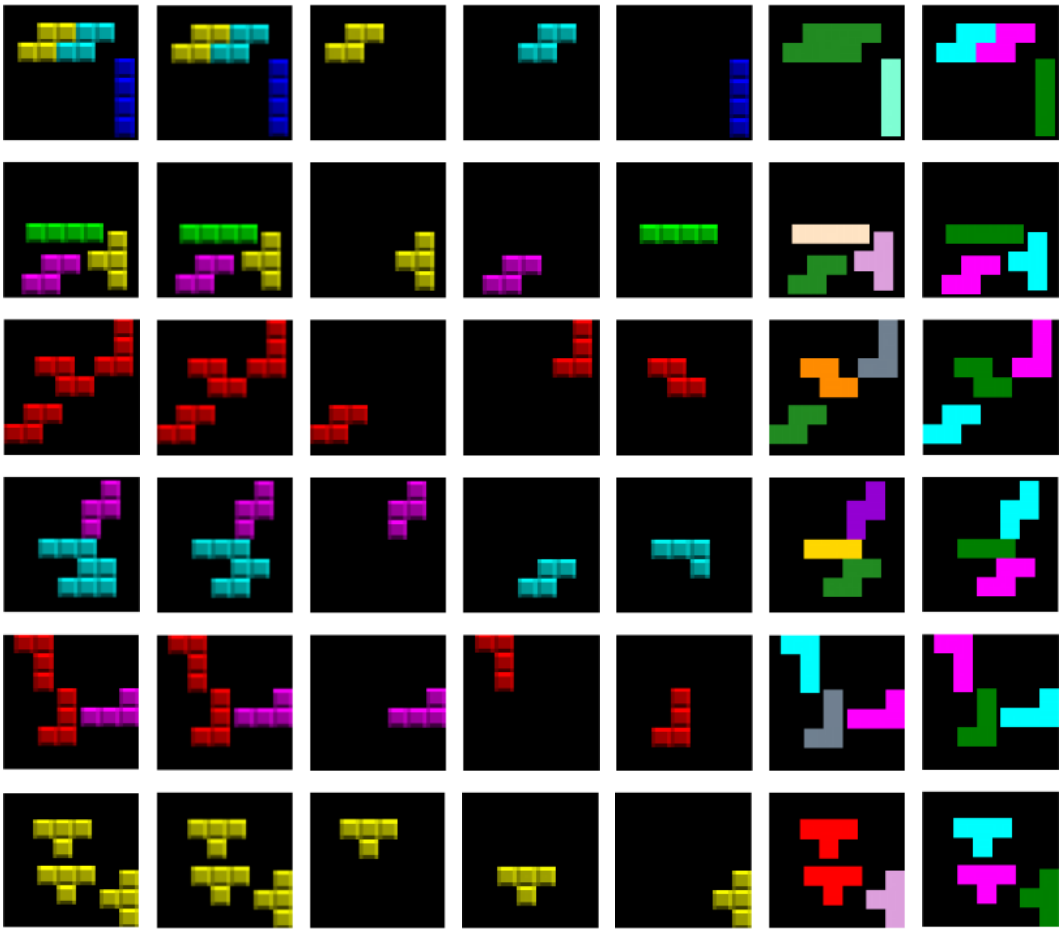


Figure 9: Segmentation masks for the learned Tetrominoes object prototypes.



(a) (b) (c) (d) (e) (f) (g)

Figure 10: Qualitative results on the Tetrominoes dataset. **(a)**: Original image. **(b)**: PCDNet Reconstruction. **(c)-(e)**: Colorized and translated object prototypes selected by PCDNet. **(f)**: Semantic segmentation masks. Colors correspond to the prototype frames in Figure 8. **(g)**: Instance segmentation masks.

Figure 11: Learned prototypes and object masks on the Space Invaders dataset.

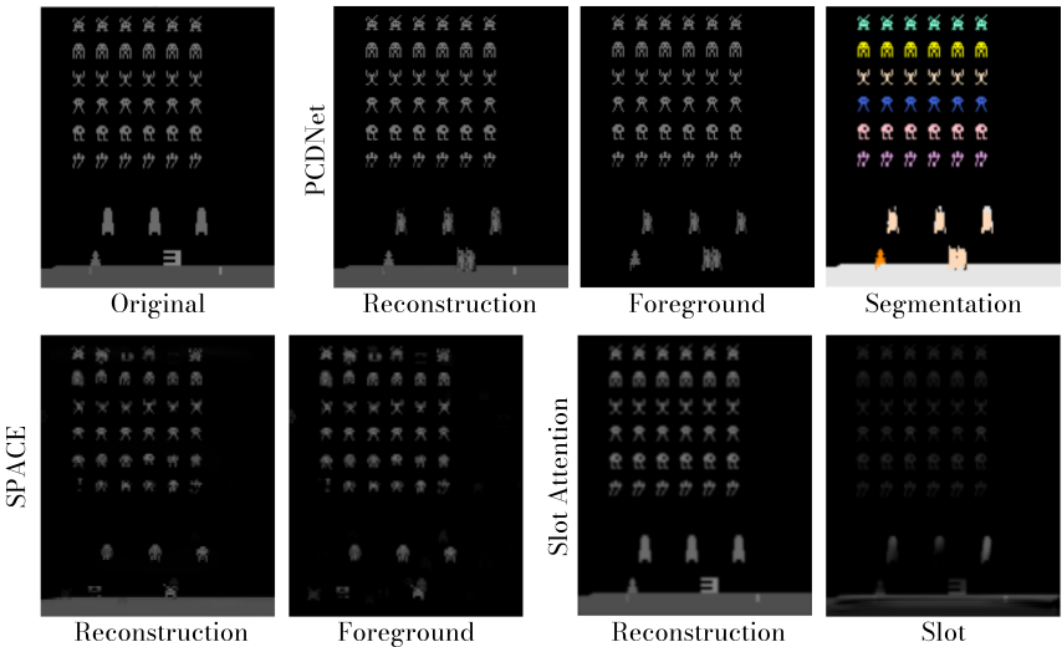


Figure 12: Additional qualitative comparison on the Space Invaders dataset.

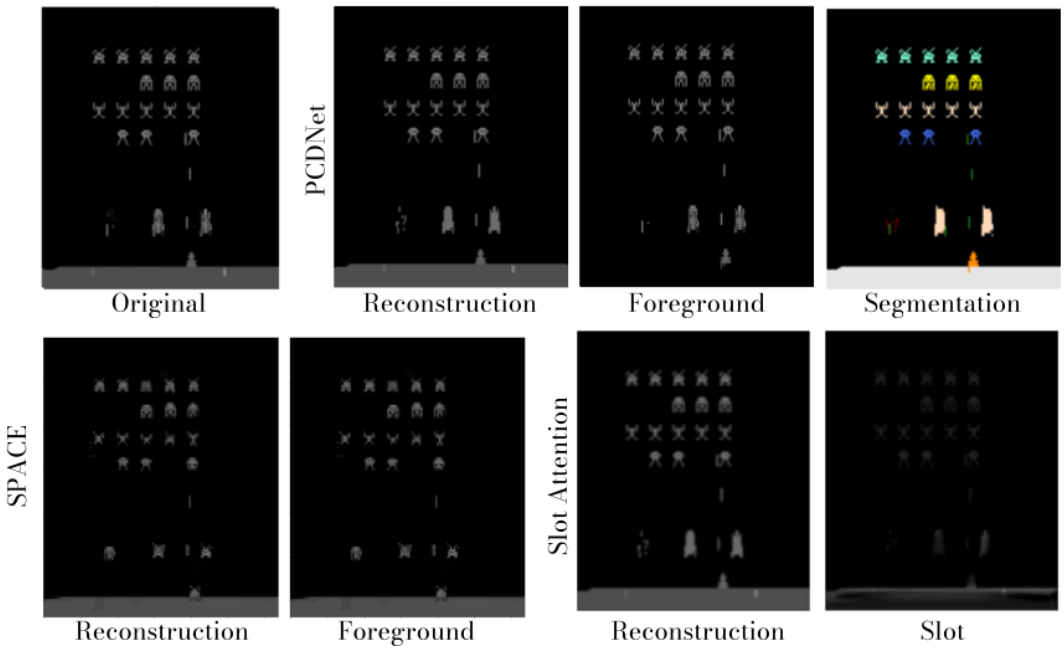


Figure 13: Additional qualitative comparison on the Space Invaders dataset.

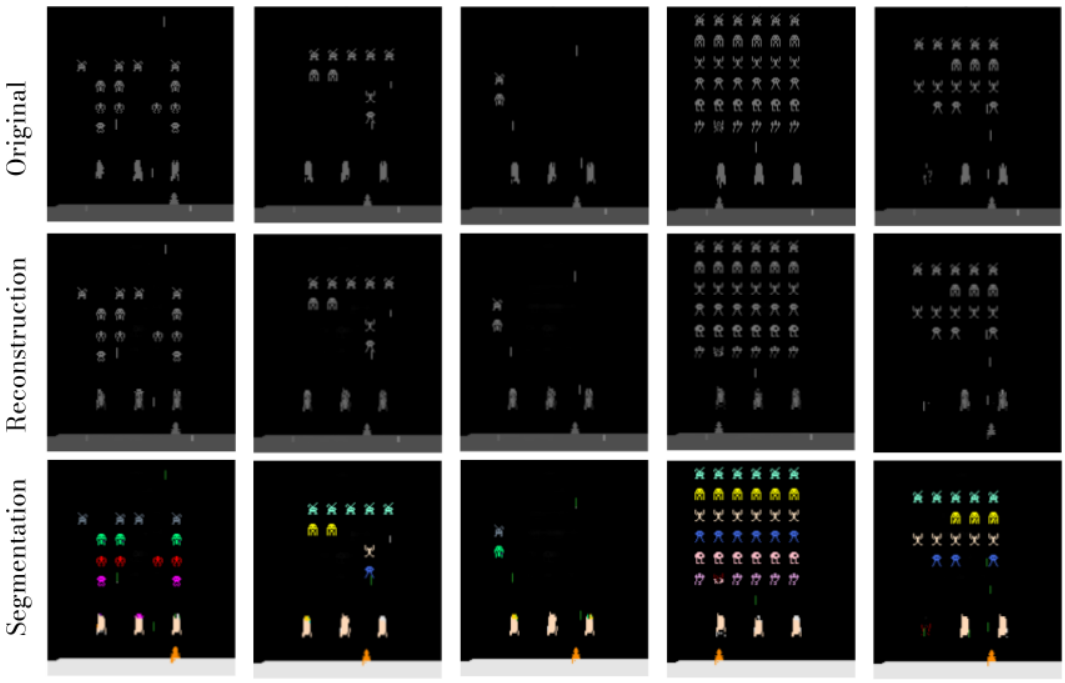


Figure 14: Additional PCDNet unsupervised segmentation qualitative results on the Space Invaders dataset.

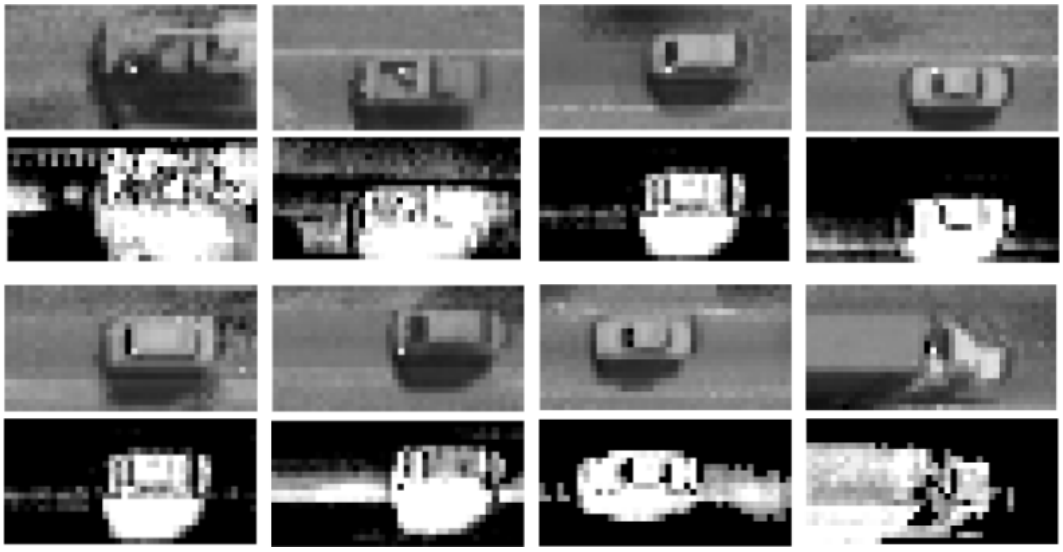


Figure 15: Learned vehicle prototypes and masks on th NGSIM dataset.