# Unsupervised Image Decomposition with Phase-Correlation Networks

Angel Villar-Corrales[a] and Sven Behnke[b]
*Autonomous Intelligent Systems, University of Bonn, Germany*

Keywords: Object-Centric Representation Learning, Unsupervised Image Decomposition, Frequency-Domain Neural Networks, Phase Correlation.

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. Code and models to reproduce our experiments can be found in https://github.com/AIS-Bonn/Unsupervised-Decomposition-PCDNet.

## 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., 2021; 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 autoencoders (Kingma and Welling, 2014) (VAEs) with spatial attention mechanisms (Burgess et al., 2019; Crawford and 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 which is characteristic of deep neural networks (DNNs).

To address the aforementioned issues, we propose a novel image decomposition framework: the ***P**hase-Correlation **D**ecomposition **Net**work* (PCDNet). Our method assumes that an image is formed as an arrangement of multiple objects, each belonging to one of a finite number of different classes. Following this assumption, 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 to estimate the transformation parameters that best align a prototype to its corresponding object in the image. The PC Cell localizes the object prototype in the image by applying the phase-correlation method (Alba et al., 2012), i.e., 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 the estimated phase shift in the frequency domain.

[a] https://orcid.org/0000-0001-5805-2098
[b] https://orcid.org/0000-0002-5040-7525

Villar-Corrales, A. and Behnke, S.
Unsupervised Image Decomposition with Phase-Correlation Networks.
DOI: 10.5220/0010919100003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 224-235
ISBN: 978-989-758-555-5; ISSN: 2184-4321

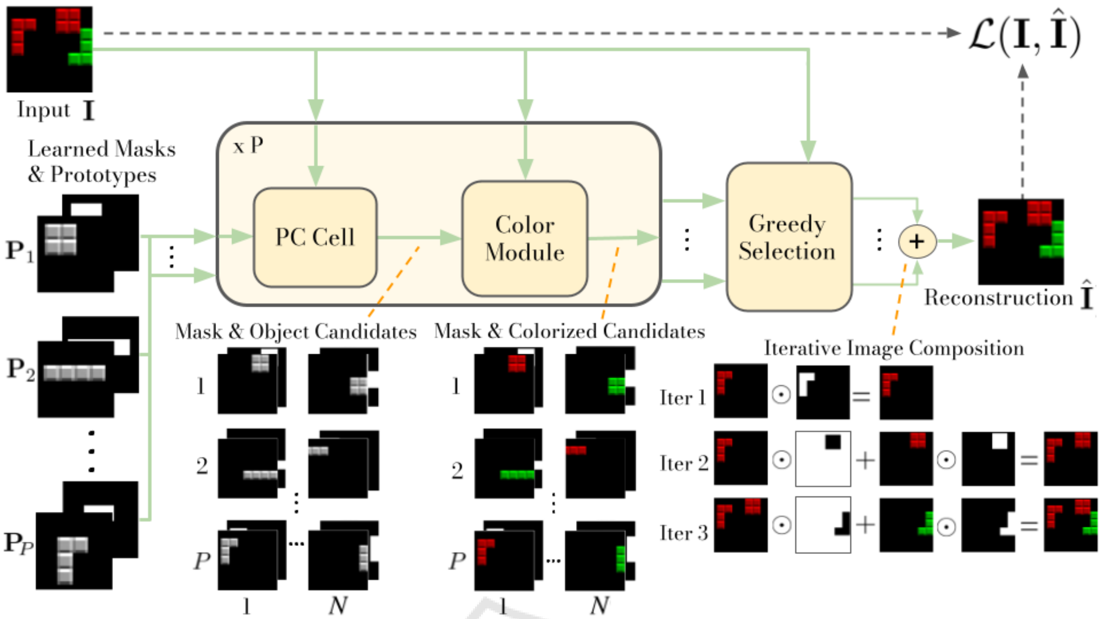

$\mathcal{L}(\mathbf{I}, \hat{\mathbf{I}})$
Input $\mathbf{I}$
Learned Masks & Prototypes
$\mathbf{P}_1$
$\mathbf{P}_2$
$\mathbf{P}_P$
x P
PC Cell
Color Module
Greedy Selection
Reconstruction $\hat{\mathbf{I}}$
Mask & Object Candidates
Mask & Colorized Candidates
Iterative Image Composition
Iter 1
Iter 2
Iter 3
1
2
$P$
1
$N$

Figure 1: PCDNet decomposition framework. First, the Phase Correlation (PC) Cell estimates the $N$ translation parameters that best align each learned prototype to the objects in the image, and uses them to obtain $(P \times N)$ 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.

PCDNet is trained by first decomposing an image into its object components, and then reconstructing the input by recombining the estimated object components following the “dead leaves” model approach, i.e., as a superposition of different objects. The strong inductive biases introduced by the network structure 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., 2021; Locatello et al., 2020; Goyal et al., 2020) decompose the image into unconstrained per-object latent representations.

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

Despite these recent advances in unsupervised object-centric learning, most existing methods rely on DNNs and 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 suffer from the inherent lack of interpretability characteristic of DNNs. Our proposed method exploits the strong inductive biases introduced by our scene composition model in order to decompose an image into object-centric components without the need for deep encoders, 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), as well as video sequences (Jojic and Frey, 2001). More recently, several works (Yang et al., 2017; Lin et al., 2018; Zhang et al., 2020; Aksoy et al., 2017; Arandjelović and 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 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.

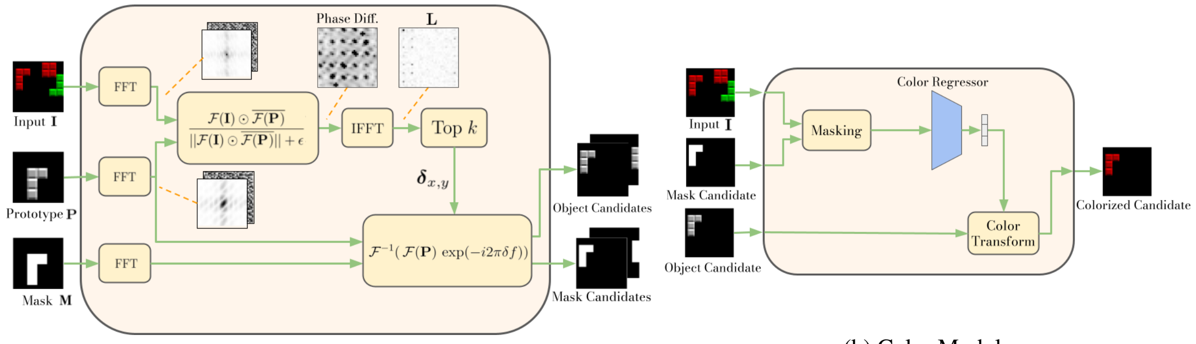
### 2.3 Frequency-domain Neural Networks

Signal analysis and manipulation in the frequency domain is one of the most widely used tools in the field of signal processing (Proakis and Manolakis, 2004). However, frequency-domain methods are not so developed for solving computer vision tasks with neural networks. They mostly focus on specific applications such as 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) and motion segmentation (Farazi and 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 our image decomposition model: PCDNet. Given an input image $\mathbf{I}$, PCDNet aims at its decomposition into $N$ independent objects $O = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_N\}$. 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 ($N_{\max}$).


Phase Diff.
L
Input I
Prototype P
Mask M
FFT
FFT
FFT
$\frac{\mathcal{F}(\mathbf{I}) \odot \overline{\mathcal{F}(\mathbf{P})}}{||\mathcal{F}(\mathbf{I}) \odot \overline{\mathcal{F}(\mathbf{P})}|| + \epsilon}$
IFFT
Top $k$
$\delta_{x,y}$
$\mathcal{F}^{-1}(\mathcal{F}(\mathbf{P}) \exp(-i2\pi\delta f))$
Object Candidates
Mask Candidates
Color Regressor
Input I
Masking
Mask Candidate
Object Candidate
Color Transform
Colorized Candidate

(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 and aligns the color channels of a translated object prototype.

Inspired by recent works in prototypical learning and clustering (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 and 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 **I** and an object prototype **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 **L** is found by applying the inverse FFT ($\mathcal{F}^{-1}$) on the normalized phase differences:

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

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 $\varepsilon$ is a small constant to avoid division by zero. Finally, the estimated relative pixel displacement ($\delta_{x,y} = (\delta_x, \delta_y)$) can then be found by locating the correlation peak in **L**:

$$\delta_{x,y} = \arg\max(\mathbf{L}) \; . \qquad (2)$$

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 $N_{\text{max}}$ correlation values from **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 to align the object prototypes and the corresponding alpha masks to the objects in the image. Given the translation parameters $\delta_x$ and $\delta_y$, an object prototype is shifted using

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

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

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 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 neural network, which learns the color parameters (one for gray-scale and three for RGB images). Finally, these learned parameters are applied to the translated object prototypes with a channel-wise affine transform. Further details about the color module are given in Appendix 5.

### 3.3 Greedy Selection

The PC Cell and color modules produce $T = N_{\max} \times P$ 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 the maximum number of objects and the number of object candidates ($T^{N_{\max}}$), 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 $T \times N_{\max}$, 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, 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. This 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 object ($\mathbf{T}_N$) is located the furthest from the viewer:

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

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

An example of this image composition is displayed in Figure 1. This reconstruction approach inherently models relative depths, allowing for a simple, yet effective, modeling of 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 reconstruction error ($\mathcal{L}_{MSE}$), while regularizing the prototypes to with respect to the $\ell_1$ norm ($\mathcal{L}_{L1}$), and enforcing smooth alpha masks with a total variation regularizer (Rudin and Osher, 1994) ($\mathcal{L}_{TV}$). Specifically, we minimize the following loss function:

$$\mathcal{L} = \mathcal{L}_{MSE} + \lambda_{L1}\, \mathcal{L}_{L1} + \lambda_{TV}\, \mathcal{L}_{TV}\,, \tag{6}$$

$$\mathcal{L}_{MSE} = ||\mathbf{I} - \mathcal{G}(\mathcal{T}')||_2^2, \tag{7}$$

$$\mathcal{L}_{L1} = \frac{1}{P} \sum_{\mathbf{P} \in \mathcal{P}} ||\mathbf{P}||_1\,, \tag{8}$$

$$\mathcal{L}_{TV} = \frac{1}{P} \sum_{\mathbf{M} \in \mathcal{M}} \sum_{i,j} |\mathbf{M}_{i+1,j} - \mathbf{M}_{i,j}| + |\mathbf{M}_{i,j+1} - \mathbf{M}_{i,j}|\,, \tag{9}$$

where $\mathcal{T}'$ are the object candidates selected by the greedy algorithm, $\mathcal{P}$ are the learned object prototypes, and $\mathcal{M}$ the corresponding alpha masks. Minimizing Equation (6) decreases the reconstruction error between the combination of selected candidates ($\mathcal{G}(\mathcal{T}')$) and the input image while keeping the prototypes compact, and the alpha masks smooth.

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

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) | 99.5 | 229,188 | 18.36 |
| ULID (Monnier et al., 2021) | 99.6 | 659,755 | 52.31 |
| IODINE (Greff et al., 2019) | 99.2 | 408,036 | 16.64 |
| PCDNet (ours) | **99.7** | **28,130** | **58.96** |

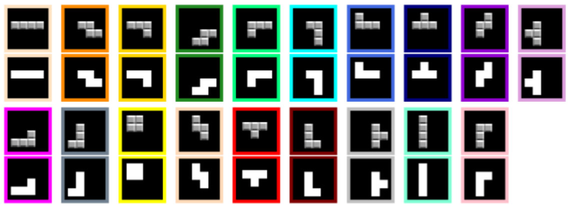


Figure 3: Object prototypes (top) and alpha masks (bottom) learned on the Tetrominoes dataset. Our model is able to discover in an unsupervised manner all 19 pieces.

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 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 all training images, 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). A detailed report of the hyper-parameters used is given in Appendix 5.

### 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 configurations and have one of six random colors.

Figure 3 displays the 19 learned object prototypes and their corresponding alpha 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.

For a fair quantitative comparison with previous works, we evaluate our PCDNet model for object segmentation using the Adjusted Rand Index (Hubert and 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). 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) (ULID).

Table 1 summarizes the evaluation results for object discovery on the Tetrominoes dataset. We observe that PCDNet outperforms SOTA models, achieving 99.7% 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

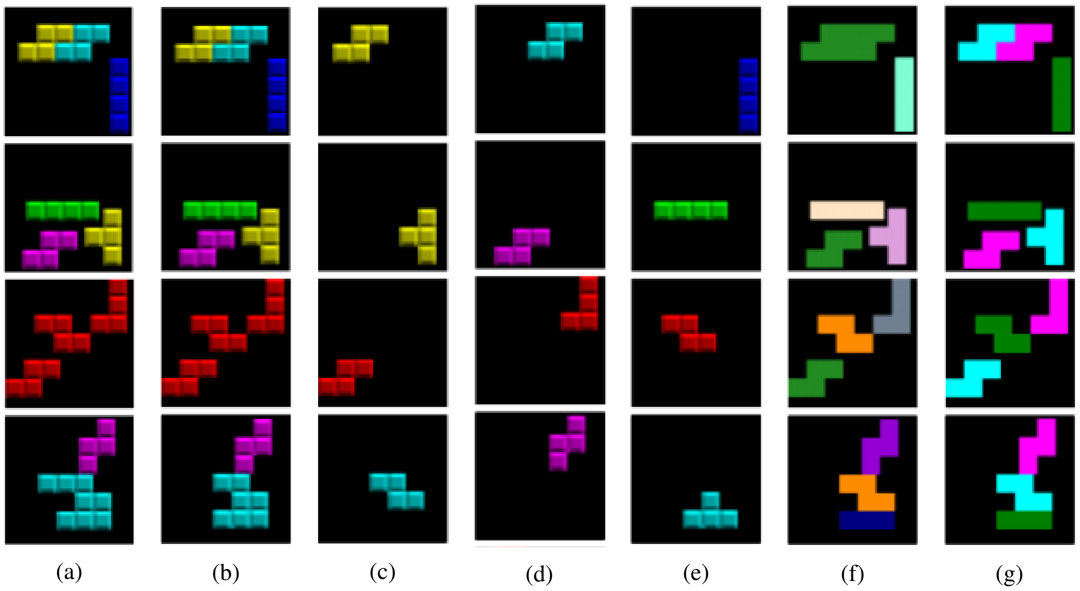


Figure 4: Qualitative decomposition and segmentation results on the Tetrominoes dataset. Last row shows a failure case. **(a)**: Original image. **(b)**: PCDNet Reconstruction. **(c)-(e)**: Colorized transformed object prototypes. **(f)**: Semantic segmentation masks. Colors correspond to the prototype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

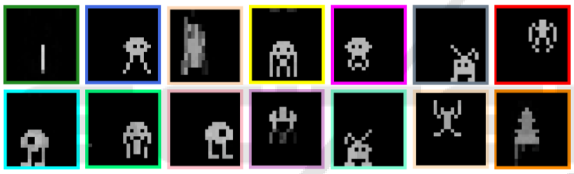


Figure 5: Object prototypes learned on the Space Invaders dataset. PCDNet discovers prototypes corresponding to the different elements from the game, e.g., aliens and ships.

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 $20 \times 20$.

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. In Figure 6 (d) one can observe that one of the slots simultaneously encodes information from several different objects. SPACE successfully decomposes the image into different object components, which are recognized as foreground objects. Nevertheless, the reconstructions appear 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 5.

## 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 (Alexiadis, 2006). We decompose each frame into up to 33 different objects, belonging to one of 30 learned vehicle prototypes.

Figure 7 depicts qualitative results on the NGSIM dataset. We see how PCDNet is applicable to real-world data, accurately reconstructing the input image, while learning prototypes for different types of vehicles. Interestingly, we notice how PCDNet learns the car shade as part of the prototype. This is a reasonable observation, since the shades are projected towards the bottom of the image throughout the whole video.

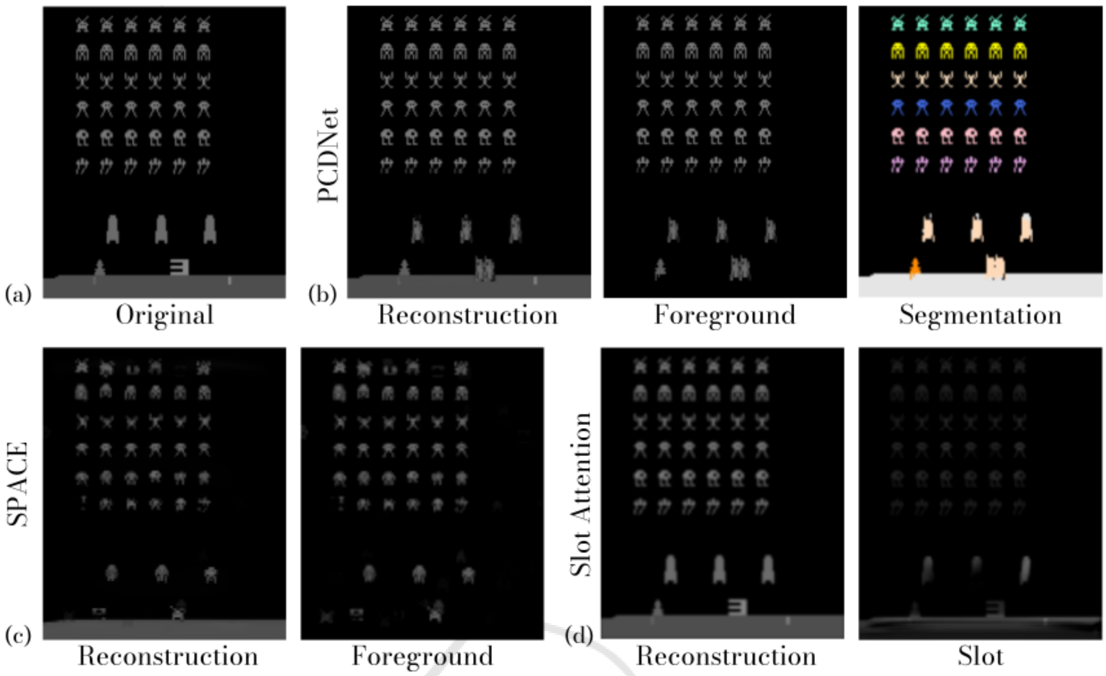


Figure 6: Comparison of different object-centric models on the Space Invaders dataset. PCDNet is the only one among the compared methods which successfully decomposes the image into accurate object components, and that has semantic knowledge of the objects. The color of each object corresponds to the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

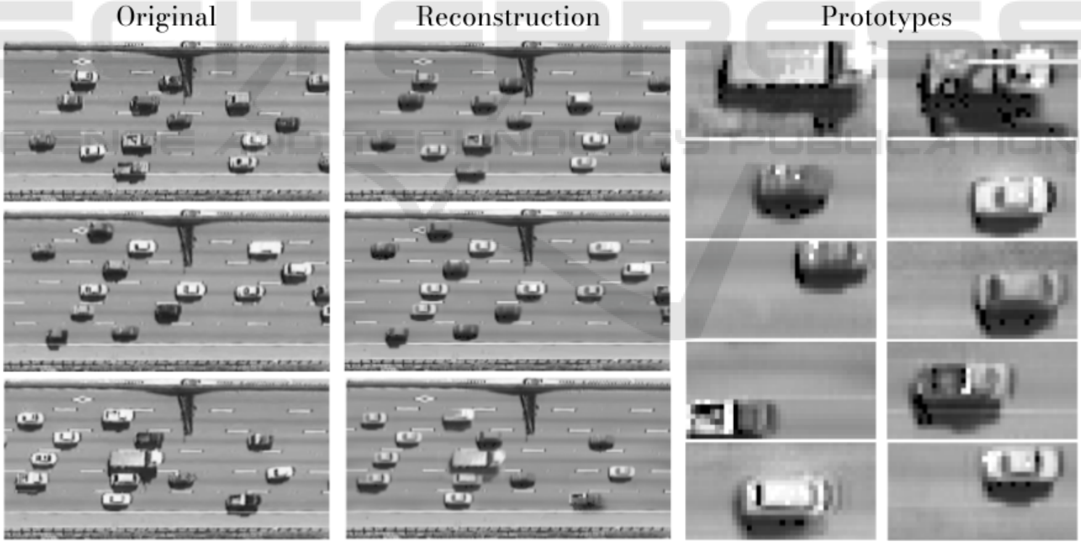


Figure 7: Object discovery on the NGSIM dataset. PCDNet learns different vehicle prototypes in an unsupervised manner.

## 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.

## ACKNOWLEDGMENTS

This work was funded by grant BE 2556/18-2 (Research Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG).

## REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.

Aksoy, Y., Aydin, T. O., Smolić, A., and Pollefeys, M. (2017). Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)*, 36(2):1–19.

Alba, A., Aguilar-Ponce, R. M., Vigueras-Gómez, J. F., and Arce-Santana, E. (2012). Phase correlation based image alignment with subpixel accuracy. In *Mexican International Conference on Artificial Intelligence*, pages 171–182. Springer.

Alexiadis, V. (2006). Video-based vehicle trajectory data collection. In *Transportation Research Board 86th Annual Meeting*.

Arandjelović, R. and Zisserman, A. (2019). Object discovery with a copy-pasting GAN. *arXiv:1905.11369*.

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

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *CoRR*, abs/1804.03599.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. (2019). Monet: Unsupervised scene decomposition and representation. *arXiv:1901.11390*.

Crawford, E. and Pineau, J. (2019). Spatially invariant unsupervised object detection with convolutional neural networks. In *Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3412–3420.

Engelcke, M., Jones, O. P., and Posner, I. (2021). Genesis-v2: Inferring unordered object representations without iterative refinement. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. (2020). Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations (ICLR)*.

Eslami, S. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. (2016). Attend, infer, repeat: Fast scene understanding with generative models. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

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

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

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

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. (2020). Recurrent independent mechanisms. In *International Conference on Learning Representations (ICLR)*.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. (2019). Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning (ICML)*, pages 2424–2433.

Gueguen, L., Sergeev, A., Kadlec, B., Liu, R., and Yosinski, J. (2018). Faster neural networks straight from JPEG. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 31, pages 3933–3944.

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

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

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

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

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

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

Ko, J. H., Mudassar, B., Na, T., and Mukhopadhyay, S. (2017). 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)*.

Kosiorek, A. R., Kim, H., Posner, I., and Teh, Y. W. (2018). Sequential attend, infer, repeat: Generative modelling of moving objects. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

Kosiorek, A. R., Sabour, S., Teh, Y. W., and Hinton, G. E. (2019). Stacked capsule autoencoders. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

Kumar, N., Verma, R., and Sethi, A. (2017). Convolutional neural networks for wavelet domain super resolution. *Pattern Recognition Letters*, 90:65–71.

Kurin, V., Nowozin, S., Hofmann, K., Beyer, L., and Leibe, B. (2017). The Atari grand challenge dataset. *arXiv:1705.10998*.

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

Lin, C.-H., Yumer, E., Wang, O., Shechtman, E., and Lucey, S. (2018). St-gan: Spatial transformer generative adversarial networks for image compositing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9455–9464.

Lin, Z., Wu, Y.-F., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations (ICLR)*.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, pages 4114–4124.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. (2020). Object-centric learning with slot attention. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

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

Mathieu, M., Henaff, M., and LeCun, Y. (2014). Fast training of convolutional networks through FFTs. In *International Conference on Learning Representations (ICLR)*.

Monnier, T., Groueix, T., and Aubry, M. (2020). Deep transformation-invariant clustering. In *International Conference on Neural Information Processing Systems (NeurIPS)*.

Monnier, T., Vincent, E., Ponce, J., and Aubry, M. (2021). Unsupervised Layered Image Decomposition into Object Prototypes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

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

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *International Conference on Neural Information Processing Systems Workshops (NeurIPS-W)*.

Proakis, J. G. and Manolakis, D. G. (2004). Digital signal processing. *PHI Publication: New Delhi, India*.

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

Rudin, L. I. and Osher, S. (1994). Total variation based image restoration with free local constraints. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 31–35. IEEE.

Sbai, O., Couprie, C., and Aubry, M. (2020). Unsupervised image decomposition in vector layers. In *IEEE International Conference on Image Processing (ICIP)*, pages 1576–1580. IEEE.

Stanic, A., Van Steenkiste, S., and Schmidhuber, J. (2021). Hierarchical relational inference. In *Conference on Artificial Intelligence (AAAI)*, pages 9730–9738.

Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J., and Levine, S. (2020). Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning (CoRL)*, pages 1439–1456. PMLR.

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

Weis, M. A., Chitta, K., Sharma, Y., Brendel, W., Bethge, M., Geiger, A., and Ecker, A. S. (2021). Benchmarking unsupervised object representations for video sequences. volume 22, pages 1–61.

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

Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.-K., and Ren, F. (2020). Learning in the frequency domain. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1740–1749.

Yang, J., Kannan, A., Batra, D., and Parikh, D. (2017). LR-GAN: Layered recursive generative adversarial networks for image generation. *arXiv:1703.01560*.

Zhang, Y., Tsang, I. W., Luo, Y., Hu, C.-H., Lu, X., and Yu, X. (2020). Copy and paste GAN: Face hallucination from shaded thumbnails. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7355–7364.

# APPENDIX

## Model and Training Details

Table 2: Hyper-parameter values used for each dataset.

| Param. | Tetrominoes | Space Invaders | NGSIM |
|---|---|---|---|
| LR | 0.003 | 0.001 | 0.013 |
| Scheduler | 0.1 / 5 | 0.1 / 5 | 0.6 / 2 |
| $\lambda_{L1}$ | $10^{-3}$ | 0 | $10^{-5}$ |
| $\lambda_{TV}$ | $10^{-3}$ | 0 | $10^{-2}$ |
| Batch size | 64 | 3 | 3 |

### Training Details

We train our experiments with an NVIDIA RTX 3090 GPU with 24 GB RAM using the Adam (Kingma and Ba, 2015) update rule. Additionally, we use a learning rate scheduler that linearly decreases the learning rate. We determine the values of our hyper-parameters using Optuna (Akiba et al., 2019)[1]. The selected hyper-parameter values for each dataset are listed on Table 2. We report the learning rate (LR), learning rate scheduler parameters (LR factor / epochs), batch size, and regularizer weights ($\lambda_{L1}$ and $\lambda_{TV}$).

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%.

### Color Module

The color module, depicted in Figure 2b, 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 (Kosiorek et al., 2019). Despite applying the same affine channel transform, our method differs in the way the color parameters are computed.

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

| Layer | Dimension |
|---|---|
| Input | $3 \times H \times W$ |
| Conv. $(3 \times 3)$ + ReLU | $12 \times H \times W$ |
| Batch Norm. | $12 \times H \times W$ |
| Conv. $(3 \times 3)$ + ReLU | $12 \times H \times W$ |
| Batch Norm. | $12 \times H \times W$ |
| Global Avg. Pooling | $12 \times 1 \times 1$ |
| Flatten | 12 |
| Fully Connected | 3 |

### Greedy Selection Algorithm

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

## Qualitative Results

Figure 8 displays PCDNet segmentation results on the Space Invaders dataset. Figure 9 depicts further qualitative comparisons on the Space Invaders dataset between PCDNet, SPACE (Lin et al., 2020) and Slot Attention (Locatello et al., 2020). Figure 10 depicts several object prototypes and their corresponding alpha masks learned by PCDNet on the NGSIM dataset.

Algorithm 1: Greedy Selection Algorithm.

```
procedure GREEDY SELECTION ALGORITHM
Inputs:
    $\mathbf{I} \leftarrow$ Input image
    $\mathcal{T} = [\mathbf{T}_1, ..., \mathbf{T}_T] \leftarrow$ Object Candidates
    $N_{\max} \leftarrow$ Max. number of objects
Returns:
    $\mathcal{O} = [\mathbf{O}_1, ..., \mathbf{O}_{N_{\max}}] \leftarrow$ Selected objects
Algorithm:
    $\mathcal{O} \leftarrow [\,]$
    for $i$ in range $[1, N_{\max}]$ 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 $\mathcal{O}$
```

[1] Hyper-parameter ranges and further details can be found in https://github.com/AIS-Bonn/Unsupervised-Decomposition-PCDNet

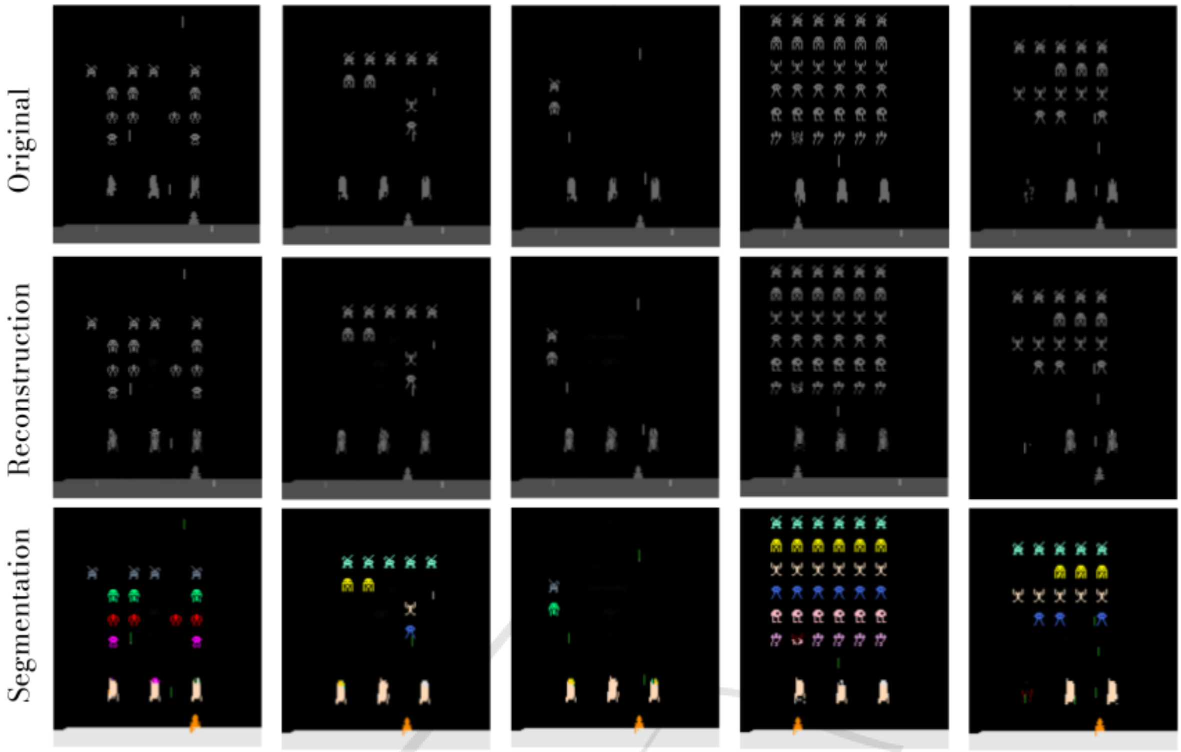


Figure 8: Additional PCDNet unsupervised segmentation qualitative results on the Space Invaders dataset.

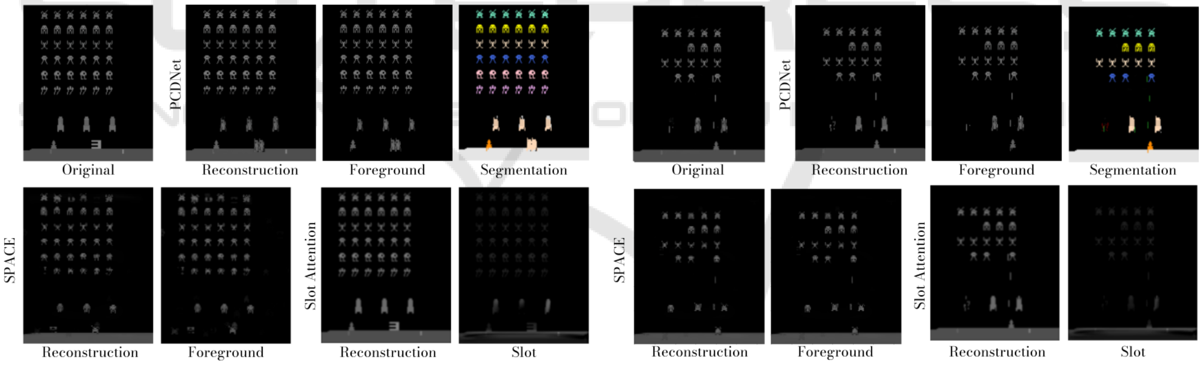


Figure 9: Additional qualitative comparison on the Space Invaders dataset.

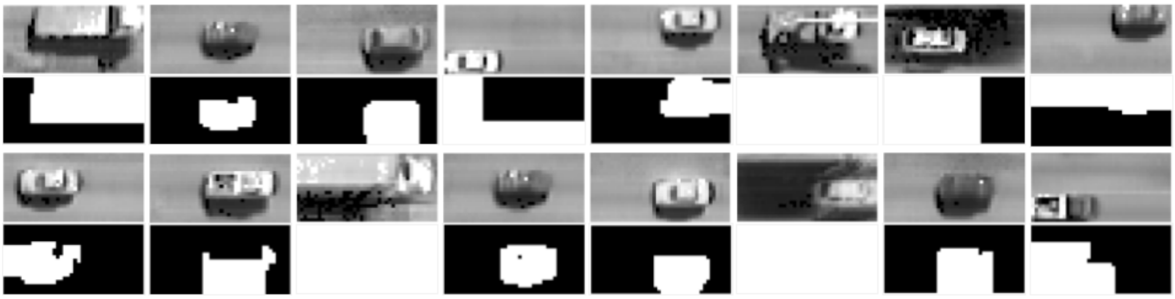


Figure 10: Several vehicle prototypes (top) and their corresponding alpha masks (bottom) learned on the NGSIM dataset.