Autoregressive Unsupervised Image Segmentation

Yassine Ouali, Céline Hudelot, Myriam Tami

Autoregressive Unsupervised Image Segmentation

Yassine Ouali, Céline Hudelot and Myriam Tami

Université Paris-Saclay, CentraleSupélec, MICS, 91190, Gif-sur-Yvette, France
{yassine.ouali,celine.hudelot,myriam.tami}@centralesupelec.fr

Abstract. In this work, we propose a new unsupervised image segmentation approach based on mutual information maximization between different constructed views of the inputs. Taking inspiration from autoregressive generative models that predict the current pixel from *past* pixels in a raster-scan ordering created with masked convolutions, we propose to use different *orderings* over the inputs using various forms of masked convolutions to construct different *views* of the data. For a given input, the model produces a pair of predictions with two valid orderings, and is then trained to maximize the mutual information between the two outputs. These outputs can either be low-dimensional features for representation learning or output clusters corresponding to semantic labels for clustering. While masked convolutions are used during training, in inference, no masking is applied and we fall back to the standard convolution where the model has access to the full input. The proposed method outperforms current state-of-the-art on unsupervised image segmentation. It is simple and easy to implement, and can be extended to other visual tasks and integrated seamlessly into existing unsupervised learning methods requiring different views of the data.

Keywords: Image segmentation, Autoregressive models, Unsupervised learning, Clustering, Representation learning.

1 Introduction

Supervised deep learning has enabled great progress and achieved impressive results across a wide number of visual tasks, but it requires large annotated datasets for effective training. Designing such fully-annotated datasets involves a significant effort in terms of data cleansing and manual labeling. It is especially true for fine-grained annotations such as pixel-level annotations needed for segmentation tasks, where the annotation cost per image is considerably high [5,17]. This hurdle can be overcome with unsupervised learning, where unknown but useful patterns can be extracted from the easily accessible unlabeled data. Recent advances in unsupervised learning [22,27,7,37], that closed the performance gap with its supervised counterparts, make it a strong possible alternative.

Recent works are mainly interested in two objectives, unsupervised representation learning and clustering. Representation learning aims to learn semantic

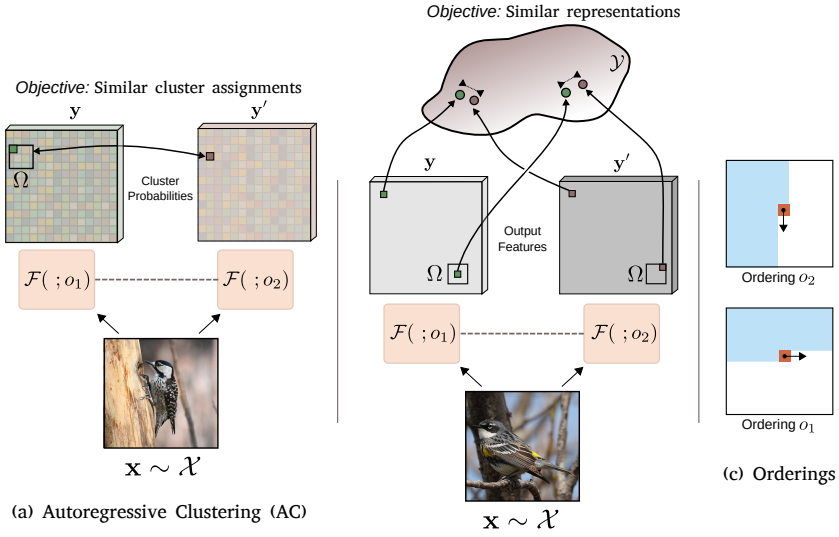

Fig. 1. Overview. Given an encoder-decoder type network $\mathcal{F}$ and two valid orderings (o_1, o_2) as illustrated in (c). The goal is to maximize the Mutual Information (MI) between the two outputs over the different *views*, *i.e.* different *orderings*. (a) For Autoregressive Clusterings (AC), we output the cluster assignments in the form of a probability distribution over pixels, and the goal is to have similar assignments regardless of the applied ordering. (b) For Autoregressive Representation Learning (ARL), the objective is to have similar representations at each corresponding spatial location and its neighbors over a window of small displacements Ω.

features that are useful for down-stream tasks, be it classification, regression or visualization. In clustering, the unlabeled data points are directly grouped into semantic classes. In both cases, recent works showed the effectiveness of maximizing Mutual Information (MI) between different *views* of the inputs to learn useful and transferable features [22,37,42,13] or discover clusters that accurately match semantic classes [27,21].

Another line of study in unsupervised learning is generative modeling. In particular, for image modeling, generative autoregressive models [36,35,41,9], such as PixelCNN, are powerful generative models with tractable likelihood computation. In this case, the high-dimensional data, *e.g.*, an image $\mathbf{x}$, is factorized as a product of conditionals over its pixels. The generative model is then trained to predict the current pixel x_i based on the past values $x_{\leq i-1}$ in a raster scan fashion using masked convolutions [35] (Fig. 3 (a)).

In this work, instead of using a single left to right, top to bottom ordering, we propose to use several orderings obtained with different forms of masked convolutions and attention mechanism. The various *orderings* over the input pixels, or the intermediate representations, are then considered as different *views*

of the input image*, and the model is then trained to maximize the MI between the outputs over these different views.

Our approach is generic, and can be applied for both clustering and representation learning (see Fig. 1). For a clustering task (Fig. 1 (a)), we apply a pair of distinct orderings over a given input image, producing two pixel-level predictions in the form of probability distribution over the semantic classes. We then maximize the MI between the two outputs at each corresponding spatial location and its intermediate neighbors. Maximizing the MI helps avoiding degeneracy (*e.g.*, uniform output distributions) and trivial solutions (*e.g.*, assigning all of the pixels to the same cluster). For representation learning (Fig. 1 (b)), we maximize a lower bound of MI between the two output feature maps over the different *views*.

We evaluate the proposed method using standard image segmentation datasets: Potsdam [14] and COCO-stuff [5], and show competitive results. We present an extensive ablation study to highlight the contribution of each component within the proposed framework, and emphasizing the flexibility of the method.

To summarize, we propose following contributions: **(i)** a novel unsupervised method for image segmentation based on autoregressive models and MI maximization; **(ii)** various forms of masked convolutions to generate different orderings; **(iii)** an attention augmented version of masked convolutions for a larger receptive field, and a larger set of possible orderings; **(iv)** an improved performance above previous state-of-the-art on unsupervised image segmentation.

2 Related Works

Autoregressive models. Many autoregressive models [32,15,35,41,38,9,10] for natural image modeling have been proposed. They model the joint probability distribution of high-dimensional images as a product of conditionals over the pixels. PixelCNN [35,36] specifies the conditional distribution of a sub-pixel (*i.e.*, a color channel of a pixel) as a full 256-way softmax, while PixelCNN++ [41] uses a mixture of logistics. In both cases, masked convolutions are used to process the initial image $\mathbf{x}$ in an autoregressive manner. In Image [38] and Sparse [10] transformers, self-attention [44] is used over the input pixels, while PixelSNAIL [9] combines both attention and masked convolutions.

Clustering and unsupervised representation learning. Recent works in clustering aim at combining traditional clustering algorithms [19] with deep learning, such as using K-means style objectives when training deep nets training [6,18,12]. However, such objective can lead to trivial and degenerate solutions [6]. IIC [27] proposed to use a MI based objective which is intrinsically more robust to such trivial solutions. Unsupervised learning of representations [22,1,37,16] rather aims to train a model, mapping the unlabeled inputs into some lower-dimensional space, while preserving semantic information and discarding instance-specific details. The pre-trained model can then be fine-tuned on a down-stream task with fewer labels.

*Throughout the paper, a *view* refers to the application of a given *ordering*. Both are used interchangeably.

Unsupervised learning and MI maximization. Maximizing MI for unsupervised learning is not a new idea [19,2], and recent works demonstrated its effectiveness for unsupervised learning. For representation learning, the training objective is to maximize a lower bound of MI over continuous random variables between distinct views of the inputs. These views can be the input image and its representation [23], the global and local features [22], the features at different scales [1], a sequence of extracted patches from an image in some fixed order [37] or different modalities of the image [42]. For a clustering objective, with discrete random variables as outputs, the exact MI can be maximized over the different views, *e.g.*, IIC [27] maximizes the MI between the image and its augmented version.

Unsupervised Image Segmentation. Methods that learn the segmentation masks entirely from data with no supervision can be categorized as follows: (1) GAN based methods [8,4] that extract and redraw the main object in the image for object segmentation. Such methods are limited to only instances with two classes, a foreground and a background. The proposed method is more generalizable and is independent of the number of ground-truth classes; (2) Iterative methods [24] consisting of a two-step process. The features produced by a CNN are first grouped into clusters using spherical K-means. The CNN is then trained for better feature extraction to discriminate between the clusters. We propose an end-to-end method simplifying both training and inference; (3) MI maximization based methods [27] where the MI between two views of the same instance at the corresponding spatial locations is maximized. We propose an efficient and effective way to create different views of the input using masked convolutions. Another line of work consists of leveraging the learned representations of a deep network for unsupervised segmentation, *e.g.*, CRFs [29] and deep priors [30].

3 Method

Our goal is to learn a representation that maximizes the MI, denoted as I, between different views of the input. These views are generated using various orderings, capturing different aspects of the inputs. Formally, let $\mathbf{x} \sim \mathcal{X}$ be an unlabeled data point, and $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ be a deep representation to be learned as a mapping between the inputs and the outputs. For clustering, $\mathcal{Y}$ is the set of possible clusters corresponding to semantic classes, and for representation learning, $\mathcal{Y}$ corresponds to a lower-dimensional space of the output features. Let $(o_i, o_j) \in \mathcal{O}$ be two orderings o_i and o_j obtained from the set of possible and valid orderings $\mathcal{O}$ (Fig. 2). For two outputs $\mathbf{y} \sim \mathcal{F}(\mathbf{x}; o_i)$ and $\mathbf{y}' \sim \mathcal{F}(\mathbf{x}; o_j)$, the objective is to maximize the predictability of $\mathbf{y}$ from $\mathbf{y}'$ and vice-versa, where $\mathcal{F}(\mathbf{x}; o_i)$ corresponds to applying the learning function $\mathcal{F}$ with a given ordering o_i to process the image $\mathbf{x}$. This objective is equivalent to maximizing the MI between the two encoded variables:

$$\max_{\mathcal{F}} I(\mathbf{y}; \mathbf{y}') \tag{1}$$

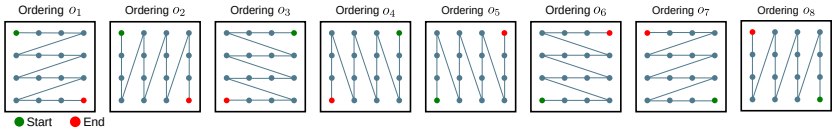

Fig. 2. Raster-scan type orderings.

We start by presenting different forms of masked convolutions to generate various raster-scan orderings, and propose an attention augmented variant (Section 3.1). We then formulate the training objective for maximizing Eq. (1) (Section 3.2). We finally conclude with a flexible design architecture for the function $\mathcal{F}$ (Section 3.3).

3.1 Orderings

Masked Convolutions In neural autoregressive modeling [35,41,9], for an input image $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ with 3 color channels, a raster-scan ordering is first imposed on the image (see Fig. 2, ordering o_1). Such an ordering, where the pixel x_i only depends on the pixels that come before it, is maintained using masked convolutions

Our proposition is to use all 8 possible raster-scan type orderings as the set of valid orderings $\mathcal{O}$ as illustrated in Fig. 2. A simple way to obtain them is to use a single ordering o_1 with the standard masked convolution (Fig. 3 (a)), along with geometric transformations g (*i.e.*, image rotations by multiples of 90 degrees and horizontal flips), resulting in 8 versions of the input image. We can then maximize the MI between the two outputs, *i.e.*, $I(\mathbf{y}; g^{-1}(\mathbf{y}'))$ with $\mathbf{y}' \sim \mathcal{F}(g(\mathbf{x}); o_j)$. In this case, since the masked weights are never trained, we cannot fall-back to the normal convolution where the function $\mathcal{F}$ has access to the full input during inference, greatly limiting the performance of such approach.

This point motivates our approach. Our objective is to learn all the weights of the masked convolution during training, and use an unmasked version during inference. This can be achieved by using a normal convolution, and for a given ordering o_i, we mask the corresponding weights during the forward pass to construct the desired view of the inputs. Then in the backward pass, we only update the unmasked weights and the masked weights remain unchanged. In this case, all of the weights will be learned and we will converge to a normal convolution given enough training iterations. During inference, no masking is applied, giving the function $\mathcal{F}$ full access to the inputs.

A straight forward way to implement this is to use 8 versions of the standard masked convolution to create the set $\mathcal{O}$ (Fig. 3 (d)). However, for each forward pass, the majority of the weights are masked, resulting in a reduced receptive field and a fewer number of weights will be learned at each iteration, leading to some disparity between them.

Given that we are interested in a discriminative task, rather than generative image modeling where the access to the current pixel is not allowed. We start

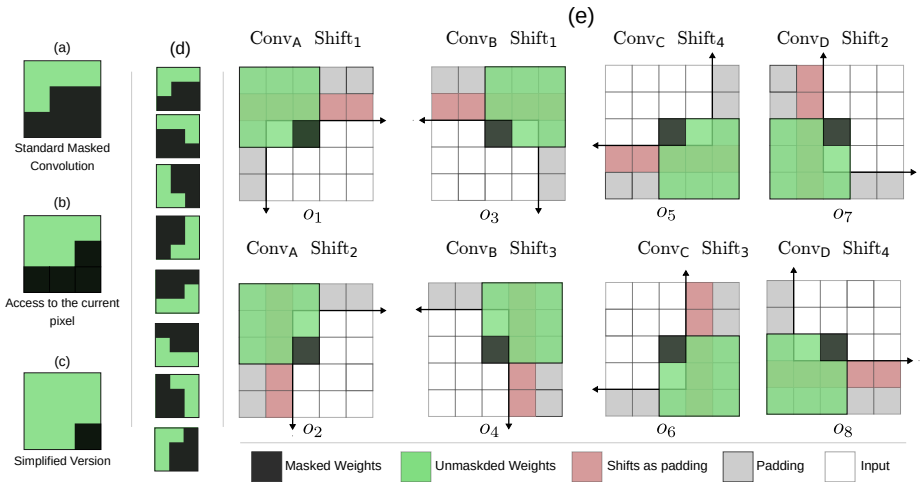

Fig. 3. Masked Convolutions. (a) Standard masked convolution used in autoregressive generative modeling, yielding an ordering o_1. (b) A relaxed version of standard masked convolution where we have access to the current pixel at each step. (c) A simplified version of masked convolution with a reduced number of masked weights. (d) The 8 versions of the standard masked convolution to construct all of the possible raster-scan type orderings. (e) The proposed types of masked convolutions with the corresponding shifts to obtain all of the 8 desired raster-scan types orderings. $F = 3$ in this case.

by relaxing the conditional dependency, and allow the model to have access to the current pixel, reducing the number of masked locations by one (Fig. 3 (b)). To further reduce the number of masked weights, for an $F \times F$ convolution, instead of masking the lower rows, we can simply shift the input by the same amount and only mask the weights of the last row. We thus reduce the number of masked weight from $\lfloor F^2/2 \rfloor$ (Fig. 3 (b)) to $\lfloor F/2 \rfloor$ (Fig. 3 (c)). With four possible masked convolutions: $\{\text{Conv}_\text{A}, \text{Conv}_\text{B}, \text{Conv}_\text{C}, \text{Conv}_\text{D}\}$ and four possible shifts:[†] $\{\text{Shift}_1, \text{Shift}_3, \text{Shift}_2, \text{Shift}_4\}$, we can create all of 8 raster-scan orderings as illustrated in Fig. 3 (e). The proposed masked convolutions do not introduce any additional computational overhead, neither in training, nor inference, making them easy to implement and integrate into existing architectures with minor changes.

Attention Augmented Masked Convolutions As pointed out by [35], the proposed masked convolutions are limited in terms of expressiveness since they create blind spots in the receptive field (Fig. 6). In our case, by applying dif-

[†] *e.g.*, for Shift_1 and a 3×3 convolution, an image of spatial dimensions $H \times W$ is first padded on the top resulting in $(H + 1) \times W$, the last row is then cropped, going back to $H \times W$.

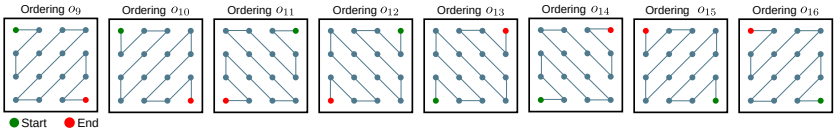

Fig. 4. Zigzag type orderings.

ferent orderings, we will have access to all of the input $\mathbf{x}$ over the course of training, and this *bug* can be seen as a *feature* where the blind spots can be considered as an additional restriction. This restricted receptive filed, however, can be overcome using the self-attention mechanism [44]. Similar to previous works [45,46,3], we propose to add attention blocks to model long range dependencies that are hard to access through standalone convolutions. Given an input tensor of shape (H, W, C_{in}), after reshaping it into a matrix $X \in \mathbb{R}^{HW \times C_{in}}$, we can apply a masked version of attention [44] in a straight forward manner. The output of the attention operation is:

$$A = \text{Softmax}((QK^\top) \odot \mathbf{M}_{o_i})V \tag{2}$$

with $Q = XW_q$, $K = XW_k$ and $V = XW_v$, where $W_q, W_k \in \mathbb{R}^{C_{in} \times d}$ and $W_v \in \mathbb{R}^{C_{in} \times d}$ are learned linear transformations that map the input X to queries Q, keys K and values V, and $\mathbf{M}_{o_i} \in \mathbb{R}^{HW \times HW}$ corresponds to a masking operation to maintain the correct ordering o_i.

The output is then projected into the output space using a learned linear transformation $W^O \in \mathbb{R}^{d \times C_{in}}$ obtaining $X_{\text{att}} = AW^O$. The output of the attention operation X_{att} is concatenated channel wise with the input X, and then merged using a 1×1 convolution resulting in the output of the attention block.

Zigzag Orderings. Using attention gives us another benefit, we can extend the set of possible orderings to include zigzag type orderings introduced in [9] (Fig. 4). With zigzag orderings, the outputs at each spatial location will be mostly influenced by the values of the corresponding neighboring input pixels, which can give rise to more semantically meaningful representations compared to that of raster-scan orderings. This is done by simply using a mask $\mathbf{M}_{o_i}$ corresponding to the desired zigzag ordering o_i. Resulting in a set $\mathcal{O}$ of 16 possible and valid orderings o_i with $i \in \{1, \ldots, 16\}$ in total. See Fig. 5 for an example.

3.2 Training Objective

In information theory, the MI $I(X;Y)$ between two random variables X and Y measures the *amount of information* learned from the knowledge of Y about X and vice-versa. The MI can be expressed as the difference of two entropy terms:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3}$$

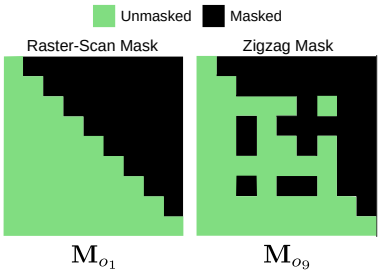

Fig. 5. Attention Masks. Examples of the different attention masks $\mathbf{M}_{o_i}$ of shape $HW \times HW$ applied for a given ordering o_i. With $HW = 9$.

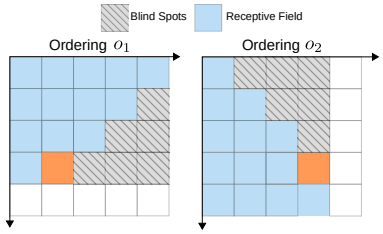

Fig. 6. Blind Spots. Blind spots in the receptive field of pixel ■ as a result of using a masked convolution for a given ordering o_i.

Intuitively, $I(X;Y)$ can be seen as the reduction of uncertainty in one of the variables, when the other one is observed. If X and Y are independent, knowing one variable exposes nothing about the other, in this case, $I(X;Y) = 0$. Inversely, if the state of one variable is deterministic when the state of the other is revealed, the MI is maximized. Such an interpretation explains the goal behind maximizing Eq. (1). The neural network $\mathcal{F}$ must be able to preserve information and extract semantically similar representations regardless of the applied ordering o_i, and learn representations that encode the underlying shared information between the different views. The objective can also be interpreted as having a regularization effect, forcing the function $\mathcal{F}$ to focus on the different views and subparts of the input $\mathbf{x}$ to produce similar outputs, reducing the reliance on specific objects or parts of the image.

Let $p(\mathbf{y}, \mathbf{y}')$ be the joint distribution produced by sampling examples $\mathbf{x} \sim \mathcal{X}$ and then sampling two outputs $\mathbf{y} \sim \mathcal{F}(\mathbf{x}; o_i)$ and $\mathbf{y}' \sim \mathcal{F}(\mathbf{x}; o_j)$ with two possible orderings o_i and o_j. In this case, the MI in Eq. (1) can be defined as the Kullback–Leibler (KL) divergence between the joint and the product of the marginals:

$$I(\mathbf{y}, \mathbf{y}') = D_{\mathrm{KL}}(p(\mathbf{y}, \mathbf{y}') \| p(\mathbf{y})p(\mathbf{y}')) \tag{4}$$

To maximize Eq. (4), we can either maximize the exact MI for a clustering task over discrete predictions, or a lower bound for an unsupervised learning of representations over the continuous outputs. We will now formulate the loss functions $\mathcal{L}_{\mathrm{AC}}$ and $\mathcal{L}_{\mathrm{ARL}}$ of both objectives for a segmentation task.

Autoregressive clustering (AC). In a clustering task, the goal is to train a neural network $\mathcal{F}$ to predict a cluster assignment corresponding to a given semantic class $k \in \{1, \ldots, K\}$ with K possible clusters at each spatial location. In this case, the encoder-decoder type network $\mathcal{F}$ is terminated with K-way softmax, outputting $\mathbf{y} \in [0,1]^{H \times W \times K}$ of the same spatial dimensions as the input. Concretely, for a given input image $\mathbf{x}$ and two valid orderings $(o_i, o_j) \in \mathcal{O}$, we forward pass the input through the network producing two output probability

distributions $\mathcal{F}(\mathbf{x}; o_i) = p(\mathbf{y}|\mathbf{x}, o_i)$ and $\mathcal{F}(\mathbf{x}; o_j) = p(\mathbf{y}'|\mathbf{x}, o_j)$ over the K clusters and at each spatial location. After reshaping the outputs into two matrices of shape $HW \times K$, with each element corresponding to the probability of assigning pixel x_l with $l \in \{1, \ldots, HW\}$ to cluster k, we can compute the joint distribution $p(\mathbf{y}, \mathbf{y}')$ of shape $K \times K$ as follows:

$$p(\mathbf{y}, \mathbf{y}') = \mathcal{F}(\mathbf{x}; o_i)^\top \mathcal{F}(\mathbf{x}; o_j) \tag{5}$$

The marginals $p(\mathbf{y})$ and $p(\mathbf{y}')$ can then be obtained by summing over the rows and columns of $p(\mathbf{y}, \mathbf{y}')$. Similar to IIC [27], we symmetrize $p(\mathbf{y}, \mathbf{y}')$ using $[p(\mathbf{y}, \mathbf{y}') + p(\mathbf{y}, \mathbf{y}')^\top]/2$ to maximize the MI in both directions. The clustering loss $\mathcal{L}_{AC}$ in this case can be written as follows:

$$\mathcal{L}_{\text{AC}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} \left[\mathbb{E}_{p(\mathbf{y},\mathbf{y}')} \log \frac{p(\mathbf{y}, \mathbf{y}')}{p(\mathbf{y})p(\mathbf{y}')} \right] \tag{6}$$

In practice, instead of only maximizing the MI between two corresponding spatial locations, we maximize it between each spatial location and its intermediate neighbors over small displacements $\mathbf{u} \in \Omega$ (see Fig. 1). This can be efficiently implemented using a convolution operation as demonstrated in [27].

Autoregressive representation learning (ARL). Although the clustering objective in Eq. (6) can also be used as a pre-training objective for $\mathcal{F}$, Tschannen *et al.* [43] recently showed that maximizing the MI does not often results in transferable and semantically meaningful features, especially when the down-stream task is a priori unknown. To this end, we follow recent representation learning works based on MI maximization [37,22,1,42], where a lower bound estimate of MI (*e.g.*, InfoNCE [37], NWJ [34]) is maximized between different views of the inputs. These estimates are based on the simple intuitive idea, that if a critic f is able to differentiate between samples drawn from the joint distribution $p(\mathbf{y}, \mathbf{y}')$ and samples drawn from the marginals $p(\mathbf{y})p(\mathbf{y}')$, then the true MI is maximized. We refer the reader to [43] for a detailed discussion.

In our case, with image segmentation as the target down-stream task, we maximize the InfoNCE estimator [37] over the continuous outputs. Specifically, with two outputs $(\mathbf{y}, \mathbf{y}') \in \mathbb{R}^{H\times W\times C}$ as C-dimensional feature maps. The training objective is to maximize the infoNCE based loss $\mathcal{L}_{\text{ARL}}$:

$$\mathcal{L}_{\text{ARL}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}} \left[\log \frac{e^{f(\mathbf{y}_l, \mathbf{y}'_l)}}{\frac{1}{N}\sum_{m=1}^{N} e^{f(\mathbf{y}_l, \mathbf{y}'_m)}} \right] \tag{7}$$

For an input image $\mathbf{x}$ and two outputs $\mathbf{y}$ and $\mathbf{y}'$. Let $\mathbf{y}_l$ and $\mathbf{y}'_m$ correspond to C-dimensional feature vectors at spatial positions l and m in the first and second outputs respectively. We start by creating N pairs of feature vectors $(\mathbf{y}_l, \mathbf{y}'_m)$, with one positive pair drawn from the joint distribution and $N-1$ negative pairs drawn from the marginals. A positive pair is a pair of feature vectors corresponding to the same spatial locations in the two outputs, *i.e.*, a pair $(\mathbf{y}_l, \mathbf{y}'_m)$ with $m = l$. The negatives are pairs $(\mathbf{y}_l, \mathbf{y}'_m)$ corresponding to two distinct spatial

positions $m \neq l$. In practice, we also consider small displacements Ω (Fig. 1) when constructing positives. Additionally, the negatives are generated from two distinct images, since two feature vectors might share similar characteristics even with different spatial positions. By maximizing Eq. (7), we push the model $\mathcal{F}$ to produce similar representations for the same spatial location regardless of the applied ordering, so that the critic function f is able to give high matching scores to the positive pairs and low matching to the negatives. We follow [22] and use separable critics $f(\mathbf{y}, \mathbf{y}') = \phi_1(\mathbf{y})^\top \phi_2(\mathbf{y}')$, where the functions ϕ_1/ϕ_2 non-linearly transform the outputs to a higher vector space, and $f(\mathbf{y}_l, \mathbf{y}'_m)$ produces a scalar corresponding to a matching score between the two representations at two spatial positions l and m of the two outputs.

Note that both losses $\mathcal{L}_{\text{AC}}$ and $\mathcal{L}_{\text{ARL}}$ can be applied interchangeably for both objectives, a case we investigate in our experiments (Section 4.1). For $\mathcal{L}_{\text{AC}}$, we can consider the clustering objective as an intermediate task for learning useful representations. For $\mathcal{L}_{\text{ARL}}$, during inference, K-means [28] algorithm can be applied over the outputs to obtain the cluster assignments.

3.3 Model

The representation $\mathcal{F}$ can be implemented in a general manner using three sub-parts, *i.e.*, $\mathcal{F} = h \circ g_{ar} \circ d$, with a feature extractor h, an autoregressive encoder g_{ar} and a decoder d. With such a formulation, the function $\mathcal{F}$ is flexible and can take different forms. With h as an identity mapping, $\mathcal{F}$ becomes a fully autoregressive network, where we apply different orderings directly over the inputs. Inversely, if g_{ar} is an identity mapping, $\mathcal{F}$ becomes a generic encoder-decoder network, where h plays the role of an encoder. Additionally, h can be a simple convolutional stem that plays an important role in learning local features such as edges, or even multiple residual blocks [20] to extract higher representations. In this case, the orderings are applied over the hidden features using g_{ar}. g_{ar} is similar to h, containing a series of residual blocks, with two main differences, the proposed masked convolutions are used, and the batch normalization [25] layers are omitted to maintain the autoregressive dependency, with an optional attention block. The decoder d can be a simple conv1×1 to adapt the channels to the number of cluster K, followed by bilinear upsampling and a softmax operation for a clustering objective. For representation learning, d consists of two separable critics ϕ_1/ϕ_2, which are implemented as a series of conv$3 \times 3 -$ BN $-$ ReLU and conv1×1 for projecting to a higher dimensional space. See sup. mat. for the architectural details.

4 Experiments

Datasets. The experiments are conducted on the newly established and challenging baselines by [27]. Potsdam [14] with 8550 RGBIR satellite images of size 200×200, of which 3150 are unlabeled. We experiment on both the 6-labels variant (roads and cars, vegetation and trees, buildings and clutter) and Potsdam-3,

a 3-label variant formed by merging each of the pairs. We also use COCO-Stuff [5], a dataset containing *stuff* classes. Similarly, we use a reduced version of COCO-Stuff with 164k images and 15 coarse labels, reduced to 52k by taking only images with at least 75% stuff pixel. In addition to COCO-Stuff-3 with only 3 labels, sky, ground and plants.

Table 1. AC Ablations. Ablations studies conducted on Potsdam (POS) and Potsdam-3 (POS3) for Autoregressive Clusterings. We show the pixel classification accuracy (%).

	Network $\mathcal{F} = h \circ g_{ar} \circ d$ h	g_{ar}	POS	POS3
	Random		28.5	38.2
$\mathcal{F}_1$	Id	5 Res. blocks	39.3	56.3
$\mathcal{F}_2$	Stem	5 Res. blocks	46.4	**66.4**
$\mathcal{F}_3$	Res. block	4 Res. blocks	**47.9**	64.5
$\mathcal{F}_4$	5 Res. blocks	Id	35.1	63.4
$\mathcal{F}_5$	ResNet-18	Id	40.7	51.9

(a) **Variation of $\mathcal{F}$.**

$\lvert\mathcal{O}\rvert$	POS	POS3
2	43.2±2.19	59.5±5.12
4	45.6±3.22	63.55±3.52
8	**46.4**	**66.4**

(b) **Number of orderings.**

Orderings Raster-Scan	Zigzag	Attention	POS	POS3
✓	×	×	45.2	61.0
✓	×	✓	47.9	66.3
×	✓	✓	47.8	**66.5**
✓	✓	✓	**49.3**	65.4

(c) **Attention.**

Sampling o_i	POS	POS3
Random	46.4	**66.4**
No Rep.	48.6	64.8
Hard	**48.9**	65.2

(d) **Sampling of o_i.**

Type	Transf.	POS	POS3
None	-	46.4	66.4
Photometric	Col. Jittering	47.9	65.5
Geometric	Flip	46.7	68.0
Geometric	Rot.	**48.5**	68.3
Geo. & Pho.	All	**48.5**	**68.3**

(e) **Transformations.**

p	POS	POS3
0	46.4	**66.4**
0.1	**47.9**	64.7
0.2	46.9	65.1

(f) **Dropout.**

Evaluation Metrics. We report the pixel classification Accuracy (Acc). For a clustering task, with a mismatch between the learned and ground truth clusters. We follow the standard procedure and find the best one-to-one permutation to match the output clusters to ground truth classes using the Hungarian algorithm [31]. The Acc is then computed over the labeled examples.

Implementation details. The different variations of $\mathcal{F}$ are trained using ADAM with a learning rate of 10^{-5} to optimize both objectives in Eqs. (6) and (7). The training is conducted on NVidia V100 GPUs, and implemented using the PyTorch framework [39]. For more experimental details, see sup. mat.

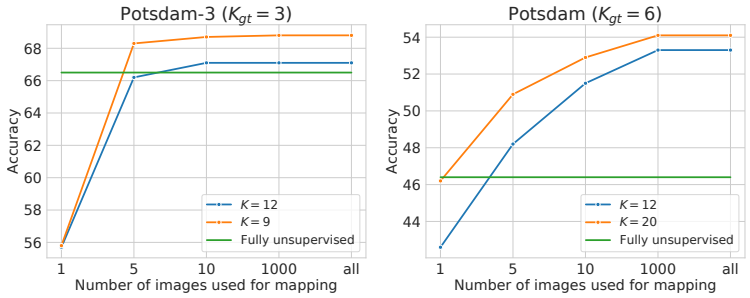

Fig. 7. Overclustering. The Acc obtained when using a number of output clusters greater than the number of ground truth classes $K > K_{gt}$. With variable number of images used to find the best many-to-one matching between the outputs and targets.

4.1 Ablation Studies

We start by performing comprehensive ablation studies on the different components and variations of the proposed method. Table 1 and Fig. 7 show the ablation results for AC, and Table 2 shows a comparison between AC and ARL, analyzed as follows:

Variations of $\mathcal{F}$. Table 2a compares different variations of the network $\mathcal{F}$. With a fixed decoder d (*i.e.*, a 1×1 Conv followed by bilinear upsampling and softmax function), we adjust h and g_{ar} going from a fully autoregressive model ($\mathcal{F}_1$) to a normal decoder-encoder network ($\mathcal{F}_4$ and $\mathcal{F}_5$). When using masked versions, we see an improvement over the normal case, with up to 8 points for Potsdam, and to a lesser extent for Potsdam-3 where the task is relatively easier with only three ground truth classes. When using a fully autoregressive model ($\mathcal{F}_1$), and applying the orderings directly over the inputs, maximizing the MI becomes much harder, and the model fails to learn meaningful representations. Inversely, when no masking is applied ($\mathcal{F}_4$ and $\mathcal{F}_5$), the task becomes comparatively simpler, and we see a drop in performance. The best results are obtained when applying the orderings over low-level features ($\mathcal{F}_2$ and $\mathcal{F}_3$). Interestingly, the unmasked versions yield results better than random, and perform competitively with 3 output classes for Potsdam-3, validating the effectiveness of maximizing the MI over small displacements $\mathbf{u} \in \Omega$. For the rest of the experiments we use $\mathcal{F}_2$ as our model.

Attention and different orderings. Table 2c shows the effectiveness of attention. With a single attention block added at a shallow level, we observe an improvement over the baseline, for both raster-scan and zigzag orderings, and their combination, with up to 4 points for Potsdam. In this case, given the quadratic complexity of attention, we used an output stride of 4.

Data augmentations. For a given training iteration, we pass the same image two times through the network, applying two different orderings at each forward pass. We can, however, pass a transformed version of the image as the second input. We investigate using photometric (*i.e.*, color jittering) and geometric (*i.e.*,

Table 2. Comparing ARL and AC. We compare ARL and AC on a clustering task (left). And investigate the quality of the learned representations by freezing the trained model, and reporting the test Acc obtained when training a linear (center) and non-linear (right) functions trained on the labeled training examples.

Clustering		
Method	POS	POS3
Random CNN	28.5	38.2
AC	**46.4**	**66.4**
ARL	45.1	57.1

Linear Evaluation		
Method	POS	POS3
AC	**23.7**	**41.4**
ARL	**23.7**	38.5

Non-Linear Evaluation		
Method	POS	POS3
AC	**68.0**	**81.8**
ARL	47.6	63.5

rotations and H-flips) transformations. For geometric transformations, we bring the outputs back to the input coordinate space before computing the loss. Results are shown in Table 2e. As expected, we obtain relative improvements with data augmentations, highlighting the flexibility of the approach.

Dropout. To add some degree of stochasticity to the network, and as an additional regularization, we apply dropout to the intermediate activations within residual blocks. Table 2f shows a small increase in Acc for Potsdam.

Orderings. Until now, at each forward pass, we sample a pair of possible orderings with replacement from the set $\mathcal{O}$. With such a sampling procedure, we might end-up with the same pair of orderings for a given training iteration. As an alternative, we investigate two other sampling procedures. First, with no repetition (No Rep.), where we choose two distinct orderings for each training iteration. Second, using hard sampling, choosing two orderings with opposite receptive fields (*e.g.*, o_1 and o_6). Table 2d shows the obtained results. We see 2 points improvement when using hard sampling for Potsdam. For simplicity, we use random sampling for the rest of the experiments. Additionally, to investigate the effect of the number of orderings (*i.e.*, the cardinality of $\mathcal{O}$), we compute the Acc over different choices and sizes of $\mathcal{O}$. Table 2b shows best results are obtained when using all 8 raster-scan orderings. Interestingly, for some choices, we observe better results, which may be due to selecting orderings that do not share any receptive fields, as the ones used in hard sampling.

Overclustering. To compute the Acc for a clustering task using linear assignment, the output clusters are chosen to match the ground truth classes $K = K_{gt}$. Nonetheless, we can choose a higher number of clusters $K > K_{gt}$, and then find the best many-to-one matching between the output clusters and ground truths based a given number of labeled examples. In this case, however, we are not in a fully unsupervised case, given that we extract some information, although limited, from the labels. Fig. 7 shows that, even with a very limited number of labeled examples used for mapping, we can obtain better results than the fully unsupervised case.

AC and ARL To compare AC and ARL, we apply them interchangeably on both clustering and representation learning objectives. In clustering, for ARL, after PCA Whitening, we apply K-means over the output features to get the cluster assignments. In representation learning, we evaluate the quality of the learned representations using both linear and non-linear separability as a proxy

Table 3. Unsupervised image segmentation. Comparison of AC with state-of-the-art methods on unsupervised segmentation.

	COCO-Stuff-3	COCO-Stuff	Potsdam-3	Potsdam
Random CNN	37.3	19.4	38.2	28.3
K-means [40]	52.2	14.1	45.7	35.3
SIFT [33]	38.1	20.2	38.2	28.5
Doersch 2015 [11]	47.5	23.1	49.6	37.2
Isola 2016 [26]	54.0	24.3	63.9	44.9
DeepCluster 2018 [6]	41.6	19.9	41.7	29.2
IIC 2019 [27]	72.3	27.7	65.1	45.4
AC	**72.9**	**30.8**	**66.5**	**49.3**

for disentanglement, and as a measure of MI between representations and class labels. Table 2 shows the obtained results.
Clustering. As expected, AC outperforms ARL on a clustering task, given that the clusters are directly optimized by computing the exact MI during training.
Quality of the learned representations. Surprisingly, AC outperforms ARL on both linear and non-linear classifications. We hypothesize that unsupervised representation learning objectives that work well on image classification, fail in image segmentation due to the dense nature of the task. The model in this case needs to output distinct representations over pixels, rather than the whole image, which is a harder task to optimize. This might also be due to using only a small number of features (*i.e.*, N pairs) for each training iteration.

4.2 Comparison with the state-of-the-art

Table 3 shows the results of the comparison. AC outperforms previous work, and by a good margin for harder segmentation tasks with a large number of output classes (*i.e.*, Potsdam and COCO-Stuff), highlighting the effectiveness of maximizing the MI between the different orderings as a training objective. We note that no regularization or data augmentation were used, and we expect that better results can be obtained by combining AC with other procedures as demonstrated in the ablation studies.

5 Conclusion

We presented a novel method to create different *views* of the inputs using different *orderings*, and showed the effectiveness of maximizing the MI over these views for unsupervised image segmentation. We showed that for image segmentation, optimizing over the discrete outputs MI works better for both clustering and unsupervised representation learning, due to the dense nature of the task. Given the simplicity and ease of adoption of the method, we hope that the proposed approach can be adapted for other visual tasks and used in future works.
Acknowledgments. We gratefully acknowledge the support of Randstad corporate research chair, Saclay-IA platform of and Mésocentre computing center.

References

1. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: Advances in Neural Information Processing Systems. pp. 15509–15519 (2019) 3, 4, 9
2. Becker, S., Hinton, G.E.: Self-organizing neural network that discovers surfaces in random-dot stereograms. Nature **355**(6356), 161–163 (1992) 4
3. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Le, Q.V.: Attention augmented convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3286–3295 (2019) 7
4. Bielski, A., Favaro, P.: Emergence of object segmentation in perturbed generative models. In: Advances in Neural Information Processing Systems. pp. 7256–7266 (2019) 4
5. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1209–1218 (2018) 1, 3, 11
6. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 132–149 (2018) 3, 14
7. Caron, M., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised pre-training of image features on non-curated data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2959–2968 (2019) 1
8. Chen, M., Artières, T., Denoyer, L.: Unsupervised object segmentation by redrawing. In: Advances in Neural Information Processing Systems. pp. 12705–12716 (2019) 4
9. Chen, X., Mishra, N., Rohaninejad, M., Abbeel, P.: Pixelsnail: An improved autoregressive generative model. In: International Conference on Machine Learning. pp. 864–872 (2018) 2, 3, 5, 7
10. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019) 3
11. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1422–1430 (2015) 14
12. Fard, M.M., Thonet, T., Gaussier, E.: Deep k-means: Jointly clustering with k-means and learning representations. arXiv preprint arXiv:1806.10069 (2018) 3
13. Federici, M., Dutta, A., Forré, P., Kushman, N., Akata, Z.: Learning robust representations via multi-view information bottleneck. arXiv preprint arXiv:2002.07017 (2020) 2
14. Gerke, M.: Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen) (2014) 3, 10
15. Germain, M., Gregor, K., Murray, I., Larochelle, H.: Made: Masked autoencoder for distribution estimation. In: International Conference on Machine Learning. pp. 881–889 (2015) 3
16. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018) 3
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014) 1
18. Haeusser, P., Plapp, J., Golkov, V., Aljalbout, E., Cremers, D.: Associative deep clustering: Training a classification network with no labels. In: German Conference on Pattern Recognition. pp. 18–32. Springer (2018) 3

19. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the american statistical association **67**(337), 123–129 (1972) 3, 4
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 10
21. He, Z., Xu, X., Deng, S.: k-anmi: A mutual information based clustering algorithm for categorical data. Information Fusion **9**(2), 223–233 (2008) 2
22. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018) 1, 2, 3, 4, 9, 10
23. Hu, W., Miyato, T., Tokui, S., Matsumoto, E., Sugiyama, M.: Learning discrete representations via information maximizing self-augmented training. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1558–1567. JMLR. org (2017) 4
24. Hwang, J.J., Yu, S.X., Shi, J., Collins, M.D., Yang, T.J., Zhang, X., Chen, L.C.: Segsort: Segmentation by discriminative sorting of segments. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7334–7344 (2019) 4
25. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015) 10
26. Isola, P., Zoran, D., Krishnan, D., Adelson, E.H.: Learning visual groups from co-occurrences in space and time. arXiv preprint arXiv:1511.06811 (2015) 14
27. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9865–9874 (2019) 1, 2, 3, 4, 9, 10, 14
28. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. arXiv preprint arXiv:1702.08734 (2017) 10
29. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 1543–1547. IEEE (2018) 4
30. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 1543–1547. IEEE (2018) 4
31. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955) 11
32. Larochelle, H., Murray, I.: The neural autoregressive distribution estimator. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 29–37 (2011) 3
33. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004) 14
34. Nguyen, X., Wainwright, M.J., Jordan, M.I.: Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory **56**(11), 5847–5861 (2010) 9
35. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in neural information processing systems. pp. 4790–4798 (2016) 2, 3, 5, 6
36. Oord, A.v.d., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759 (2016) 2, 3
37. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 1, 2, 3, 4, 9

38. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4055–4064. PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018) 3
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017) 11
40. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research **12**(Oct), 2825–2830 (2011) 14
41. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517 (2017) 2, 3, 5
42. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. arXiv preprint arXiv:1906.05849 (2019) 2, 4, 9
43. Tschannen, M., Djolonga, J., Rubenstein, P.K., Gelly, S., Lucic, M.: On mutual information maximization for representation learning. arXiv preprint arXiv:1907.13625 (2019) 9
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 3, 7
45. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018) 7
46. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. arXiv preprint arXiv:1805.08318 (2018) 7