Object Segmentation Without Labels with Large-Scale Generative Models

Andrey Voynov [1] **Stanislav Morozov** [1] **Artem Babenko** [1]

Abstract

The recent rise of unsupervised and self-supervised learning has dramatically reduced the dependency on labeled data, providing effective image representations for transfer to downstream vision tasks. Furthermore, recent works employed these representations in a fully unsupervised setup for image classification, reducing the need for human labels on the fine-tuning stage as well. This work demonstrates that large-scale unsupervised models can also perform a more challenging object segmentation task, requiring neither pixel-level nor image-level labeling. Namely, we show that recent unsupervised GANs allow to differentiate between foreground/background pixels, providing high-quality saliency masks. By extensive comparison on standard benchmarks, we outperform existing unsupervised alternatives for object segmentation, achieving new state-of-the-art. Our model and implementation are available online[2].

1. Introduction

Reducing the reliance on labeled data is a long-standing goal of machine learning research. The recent studies on unsupervised and self-supervised learning for both discriminative (Chen et al., 2020b; He et al., 2020; Caron et al., 2020; Grill et al., 2020) and generative (Donahue & Simonyan, 2019; Chen et al., 2020a) models have demonstrated that one does not need labeled data on the pretraining stage to produce image representations for typical computer vision problems. While these representations require some labeled data to finetune to a particular downstream task, recent works (Van Gansbeke et al., 2020; Zheltonozhskii et al., 2020) exploit these representations to solve the image classification problem without labels at all.

This paper employs state-of-the-art unsupervised generative models to perform label-free object segmentation, where groundtruth pixel-level labels are expensive to collect because of labor-intensive human efforts. This problem currently receives much research attention and is typically addressed by methods based on GANs (Chen et al., 2019; Bielski & Favaro, 2019; Benny & Wolf, 2020). However, training high-quality GANs can be both time-consuming and unstable. Moreover, the protocols in (Chen et al., 2019; Bielski & Favaro, 2019; Benny & Wolf, 2020) typically include a large number of hyperparameters that are tricky to tune in the completely unsupervised setup when a labeled validation set is not available. In contrast, we propose an alternative, much simpler approach that does not require adversarial training or heavy hyperparameter tuning for each particular segmentation task.

Our work is partially inspired by the findings from Voynov & Babenko (2020), which has shown that the latent space of BigGAN (Brock et al., 2019) possess the direction "responsible" for the background removal, and this direction can be used to produce training data for saliency detection. However, the approach (Voynov & Babenko, 2020) is not unsupervised since (i) BigGAN is trained with known Imagenet labels, therefore, it is not an unsupervised model; (ii) it requires manual inspecting of several latent transformations.

This paper eliminates external supervision mentioned above and demonstrates that large off-the-shelf GANs can segment images, being completely unsupervised. As a main technical novelty, we introduce an automatic procedure identifying the "segmenting" latent directions in the pretrained GANs. This procedure reveals such directions in the state-of-the-art publicly available BigBiGAN (Donahue & Simonyan, 2019), which is trained on the Imagenet (Deng et al., 2009) without labels. These directions allow to distinguish object/background pixels in the generated images, providing decent segmentation masks. These masks are then used to supervise a discriminative U-Net model (Ronneberger et al., 2015), which is stable and easy to train. As another advantage, our approach also provides a straightforward way to tune hyperparameters. Since an amount of synthetic data is unlimited, its hold-out subset can be used as validation.

Our work confirms the promise of using GANs to produce synthetic training data, which is a long-standing goal of research on generative modeling. In extensive experiments,

[1]Yandex, Moscow, Russia. Correspondence to: Andrey Voynov <an.voynov@yandex.ru>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[2]https://github.com/anvoynov/BigGANsAreWatching

we show that the approach often outperforms the existing unsupervised alternatives for object segmentation and saliency detection. Our results provide additional evidence to the common trend that more data and larger models often can reduce the requirements of human labels.

Overall, the contributions of our paper are the following:

1. We propose to perform unsupervised object segmentation using off-the-shelf Imagenet-pretrained GANs.
2. We introduce an automatic method to identify "segmenting" directions in the GAN latent space.
3. We show that our method outperforms the state-of-the-art in most operating points. Given its simplicity, the method can serve as a baseline in the future.

2. Related work

In this paper, we address the binary object segmentation problem, i.e., for each pixel, we aim to predict if it belongs to the object or the background. This problem is typically referred to as saliency detection (Wang et al., 2019) and foreground object segmentation (Chen et al., 2019; Bielski & Favaro, 2019; Benny & Wolf, 2020). While most prior works propose fully-supervised or weakly-supervised methods, we focus on the most challenging unsupervised setup, where only a few approaches have been developed.

Existing unsupervised approaches. Before the rise of deep learning models, a large number of "shallow" unsupervised techniques were developed (Zhu et al., 2014b; Jiang et al., 2013; Peng et al., 2016; Cong et al., 2017; Cheng et al., 2014; Wei et al., 2012). These earlier techniques were mostly based on hand-crafted features and heuristics, e.g., color contrast (Cheng et al., 2014), or certain background priors (Wei et al., 2012). Often these approaches also utilize traditional computer vision routines, such as super-pixels (Yang et al., 2013; Wang et al., 2016), object proposals (Guo et al., 2017), CRF (Krähenbühl & Koltun, 2011). These heuristics, however, are not completely learned from data, and the corresponding methods are inferior to the more recent "deep" approaches.

Regarding unsupervised deep models, several works have recently been proposed by the saliency detection community (Wang et al., 2017b; Zhang et al., 2018; 2017; Nguyen et al., 2019). Their main idea is to combine or fuse the predictions of several heuristic saliency methods, typically using them as a source of noisy groundtruth for deep CNN models. However, these methods are not entirely unsupervised since they rely on the supervised-pretrained classification or segmentation networks or utilize a limited number of labeled data (Zhang et al., 2020). In contrast, in this work, we focus on the methods that do not require labeled data.

Generative models for object segmentation. The recent line of unsupervised methods (Chen et al., 2019; Bielski & Favaro, 2019) employs generative modeling to decompose the image into the object/background. In a nutshell, these methods exploit the idea that the object's location or appearance can be perturbed without affecting image realism. This inductive bias is formalized in the training protocols, which include learning of GANs. Therefore, for each new segmentation task, one has to perform adversarial learning, which can be unstable, time-consuming, and sensitive to hyperparameters. In contrast, our approaches avoid these disadvantages, being much simpler and easier to reproduce. In essence, we propose to use the "inner knowledge" of pretrained large-scale GANs to produce the saliency masks.

Latent spaces of large-scale GANs. Recent study (Voynov & Babenko, 2020) has shown that the latent space of BigGAN (Brock et al., 2019) can be used to obtain saliency masks for synthetic images. However, such an ability was discovered only for BigGAN trained under the supervision from the image class labels. For unconditional GANs, it was not discovered in (Voynov & Babenko, 2020), hence, it is not clear if the supervision from the class labels is necessary for the GAN latent space to distinguish between object/background pixels. This paper shows that this supervision is not necessary, contributing novel knowledge to the general trend to unsupervised learning.

3. Latent Segmenters in Unsupervised GANs

Voynov & Babenko (2020) has shown that the BigGAN's latent space contains a direction h_{bg} responsible for a background removal: once a latent code z of a generated image $G(z)$ is shifted by h_{bg}, the background pixels of the shifted image $G(z + h_{bg})$ become white, while the foreground ones remain almost unchanged. As we will show, the latent spaces of other large-scale GANs also have directions that have different effects on background/foreground pixels. The following section provides a principled framework to identify such "segmenting" directions automatically.

3.1. Modeling a segmenting direction

Formally, we consider a latent shift h to be a *segmenting direction* if there are two affine operators $A_1, A_2 : \mathbb{R}^3 \to \mathbb{R}^3$ such that for each latent z and pixel $G(z)_{x,y}$ we have

$$G(z + h)_{x,y} = A_{i(x,y)}(G(z)_{x,y}),\ i(x, y) \in \{1, 2\} \quad (1)$$

that is, the latent shift acts on each pixel as one of the two fixed maps. Intuitively, this definition formalizes our desire that the latent segmenter should affect the background/foreground pixels differently.

Now we explain how to find these affine operators $A_1,\ A_2$ for a given latent direction h from the generator G. In

addition, we also show how to identify the directions that are the most appropriate for an object segmentation task. Given two affine operators A_1, A_2, for a pair of pixel intensities c, c' let us define a map

$$S_{A_1,A_2}(c,c') = \underset{A\in\{A_1,A_2\}}{\arg\min} \left(\|A(c) - c'\|_2\right) \cdot c \quad (2)$$

that is S_{A_1,A_2} chooses an operator that maps c closer to c' and applies it to c. We extend this action to the generated images space by setting $(S_{A_1,A_2} \cdot G(z))_{x,y} = S_{A_1,A_2}(G(z)_{x,y}, G(z+h)_{x,y})$. Then we define the restoration loss:

$$\mathcal{L}_h(A_1,A_2) = \mathop{\mathbb{E}}_{z} \sum_{x,y} \| \left(S_{A_1,A_2} \cdot G(z)\right)_{x,y} - G(z+h)_{x,y}\|_2 \quad (3)$$

here we sum over all pixels of an image $G(z)$. This quantity indicates how good one can approximate the map $\sigma_h : G(z) \to G(z+h)$ by choosing the optimal A_1, A_2 for each pixel. If this map can be represented in a form of (1), the quantity $\mathcal{L}_h$ possesses a global minimum equal to 0.

Thus, for a given direction h one can find the optimal A_1, A_2 by solving

$$A_1, A_2 = \text{argmin}_{A_1,A_2} \mathcal{L}_h(A_1,A_2) \quad (4)$$

These operators also define a binary segmentation of a generated image by assigning a label computed as

$$\underset{i\in\{1,2\}}{\arg\min} \|A_i \cdot G(z)_{x,y} - G(z+h)_{x,y}\|_2 \quad (5)$$

for each pixel (x, y).

3.2. Exploring segmenting directions

Now we explain how to identify the segmenting direction in the latent space of a given pretrained GAN. First, we find a set of interpretable directions using the technique from (Voynov & Babenko, 2020) with the default hyperparameters. This results in a set of latent directions $h_1, \dots, h_N$. For each h_k we then optimize (4) with the stochastic gradient descent. Since the number of learnable parameters is only 24 and the loss is averaged over all image pixels, we use a small mini-batch of four images and 200 steps of Adam optimizer with a learning rate 0.005. The optimization converges rapidly, and we did not observe any benefits from larger batches or larger numbers of steps. Overall, this optimization takes a few minutes on the Nvidia-1080ti GPU card. Thus, for each h_k we obtain a pair of affine operators $(A_1^{(h_k)}, A_2^{(h_k)})$. The optimal loss value $\mathcal{L}_{h_k}$ indicates how good a particular transform σ_{h_k} can be approximated by two pixelwise affine operators. In practice, this ranking is not sufficient to identify directions suitable for segmentation as the transforms σ_k may induce almost identical or a global lighting transformation with $A_1^{(h_k)}$ close to $A_2^{(h_k)}$. If so, the masking based on these operators becomes noisy and inadequate for downstream tasks. To overcome this issue, for each h_k we compute the mean distance $D_k = \|A_1^{(h_k)} \cdot G(z)_{x,y} - A_2^{(h_k)} \cdot G(z)_{x,y}\|_2$ over the pixels of generated images. Intuitively, the direction that induces the most distant A_1, A_2 should produce adequate segmentation masks for synthetic images.

We apply the above approach to the Imagenet-pretrained unsupervised BigBiGAN, which parameters are available online. After using the method from (Voynov & Babenko, 2020), we extract 120 latent directions to serve as candidates to be the latent segmenters. We also scale them by a multiplier 5 as the unit-length latent shifts commonly induce minor image transformation leading to noisy restoration loss $\mathcal{L}_h$ optimization process. After solving the optimization problem (4) and computing D_k values for each h, we choose the direction with the highest D_k among the best-70% in terms of the restoration loss. We use this direction as a latent segmenter to produce the saliency masks.

On the Figure 1 we plot the optimal values of the restoration loss $\mathcal{L}_{h_k}$ and the operators mean distance D_k for all the candidate directions. Notably, the background/foreground direction, which we utilize for the saliency generation, has the highest mean distance D_k while possessing low restoration loss $\mathcal{L}_{h_k}$. On the Figure 2 we illustrate the images $S_{A_1,A_2} \cdot G(z)$ that approximate the shifted $G(z+h)$ by the pixelwise operators A_1, A_2 that minimize $\mathcal{L}_{h_{bg}}$. In our experiments, the operators A_1, A_2 corresponding to the background saliency direction have a form:

$$A_1(c) = \begin{pmatrix} 0.13 & -0.12 & 0.06 \\ 0.01 & 0.00 & 0.04 \\ 0.02 & -0.20 & 0.22 \end{pmatrix} \cdot c + \begin{pmatrix} 0.78 \\ 0.76 \\ 0.69 \end{pmatrix};$$

$$A_2(c) = \begin{pmatrix} 0.31 & -0.05 & 0.05 \\ 0.04 & 0.19 & 0.06 \\ 0.01 & -0.06 & 0.31 \end{pmatrix} \cdot c - \begin{pmatrix} 0.1 \\ 0.15 \\ 0.19 \end{pmatrix}$$

The first operator performs aggressive lightening, while the second one downscales the channels and applies a minor negative shift resulting in a darkening. In practice, the intensity of the pixels handled by the first operator increases while the intensity of the pixels handled by the second one decreases. This direction, along with the corresponding operators, is used to produce the saliency masks as shown in Figure 2. For efficiency, we set the generated image mask at pixel (x, y) to be equal to $[\|G(z+h_{bg})_{x,y}\| > \|G(z)_{x,y}\|]$ as in practice it appears to be a decent approximation of (5).

3.3. Adaptation to the particular segmentation task.

Since BigBiGAN was trained on the Imagenet, sampling the latent codes from the standard Gaussian distribution

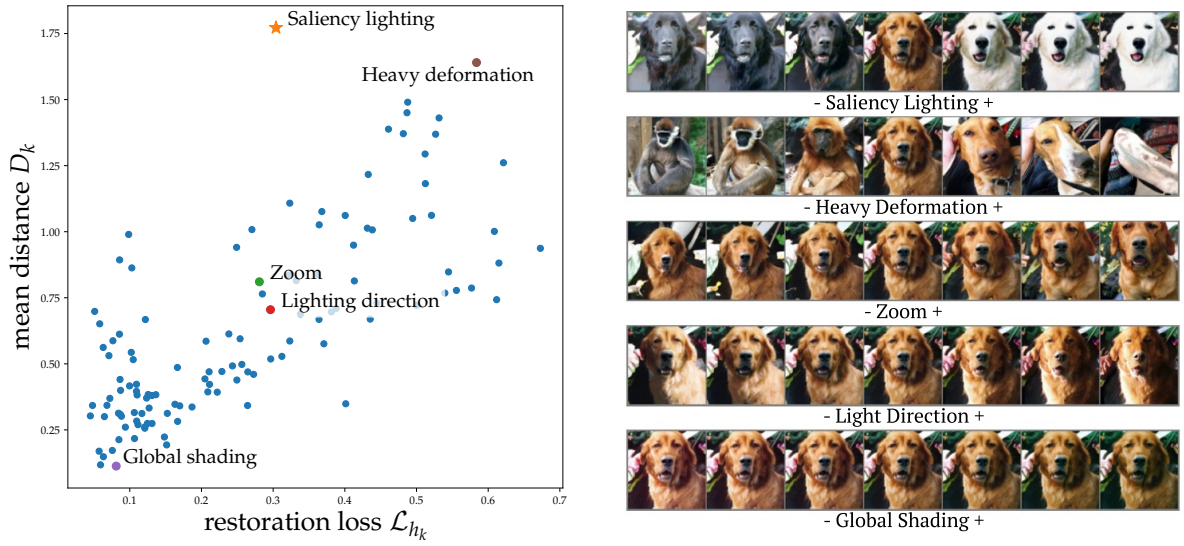

Figure 1. Left: BigBiGAN latent directions restoration loss values and the operators A_1, A_2 dissimilarity. *Right*: examples of generated image transformations induced by moving a latent z along some of directions. The central image in each row corresponds to the original sample $G(z)$.

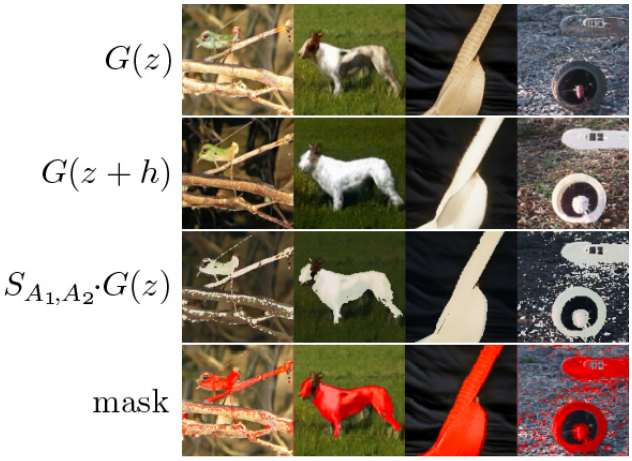

Figure 2. From top to bottom: generated image; shifted image; shifted image approximation by pixelwise operators A_1, A_2; mask generated by these operators assignment.

$z \sim \mathcal{N}(0, \mathbb{I})$ will result in the distribution of the synthetic data that resembles the Imagenet. However, this distribution can be suboptimal for the particular segmentation task. To mitigate this issue, we introduce a simple additional step in the process of synthetic data generation. To make the distribution of generated images closer to the particular dataset $\mathcal{I}=\{I_1, \ldots, I_N\}$, we sample z from the latent space regions that are close to the latent codes of $\mathcal{I}$. To this end, we use the BigBiGAN encoder to compute the latent representations $\{E(I_1), \ldots, E(I_N)\} \subset \mathbb{R}^{120}$ and sample the codes from the neighborhood of these representations. Formally, the samples have a form:

$$\{E(I_i)+\alpha\xi \mid i \sim \mathcal{U}\{1, N\}, \xi \sim \mathcal{N}(0, I)\} \qquad (6)$$

Here α denotes the neighborhood size, and it should be larger for small $\mathcal{I}$ to prevent overfitting. In particular, we use α=0 for Imagenet and α=0.2 for all other cases. In the experimental section, we demonstrate that this simple and efficient modification of the data generation process results in a dramatic performance boost.

3.4. Improving saliency masks.

Here we describe a few simple heuristics that increase the masks' quality for the particular segmentation task. The ablation of each component is presented in Section 4.4.

Mask size filtering. Since some of the BigBiGAN-produced images are low-quality and do not contain well-defined objects, the corresponding masks can result in very noisy supervision. To alleviate this, we apply simple filtering that excludes the images where the ratio of foreground pixels exceeds 0.5.

Histogram filtering. Since $G(z+h_{bg})$ should have mostly dark and light pixels, we filter out the images that are not contrastive enough. Formally, we compute the intensity histogram with 12 bins for the grayscaled $G(z+h_{bg})$. Then we smooth it by taking the moving average with a window of 3 and filter out the samples that have local maxima outside the first/last buckets of the histogram.

Connected components filtering. For each generated mask M we group the foreground pixels into connected (by edges) groups forming clusters $M_1, \ldots, M_k$. Assuming that M_1 is the cluster with the maximal area, we exclude all the clusters M_i with $|M_i| < 0.2 \cdot |M_1|$. This technique allows to remove visual artifacts from the synthetic data.

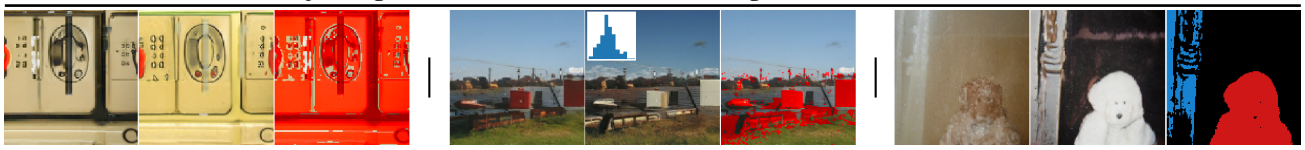

Figure 3. Examples of mask improvement. *Left:* the sample rejected by the mask size filter. *Middle:* the sample rejected by the histogram filtering. *Right block*: mask pixels removed by the connected components filter are shown in blue and the remaining mask pixels are shown in red.

We present samples of images rejected by each filtering step in Figure 3.

3.5. Training the model on synthetic data

Given a large amount of synthetic data, one can train one of the existing image-to-image CNN architectures in the fully supervised regime. The whole pipeline is schematically presented in Figure 4. In all our experiments, we employ a standard U-net architecture (Ronneberger et al., 2015). We train U-net on the synthetic dataset with the Adam optimizer and the binary cross-entropy objective applied on the pixel level. We perform $12 \cdot 10^3$ steps with batch 95. The initial learning rate equals 0.001 and is decreased by 0.2 on step $8 \cdot 10^3$. During inference, we rescale an input image to have a size 128 along its shorter side. Compared to existing unsupervised alternatives, the training of our model is straightforward and does not include a large number of hyperparameters. The only hyperparameters in our protocol are batch size, learning rate schedule, and a number of optimizer steps, and we tune them on the hold-out validation set of synthetic data. We set the batch size to guarantee the maximal GPU memory utilization. Figure 5 reports the segmentation quality on the hold-out synthetic data for the different hyperparameter values. Notably, they have a minor affect on the model quality. Figure 5 demonstrates that the optimal hyperparameters chosen for synthetic data are typically optimal for real datasets as well. Training with online synthetic data generation takes approximately seven hours on two Nvidia 1080Ti cards.

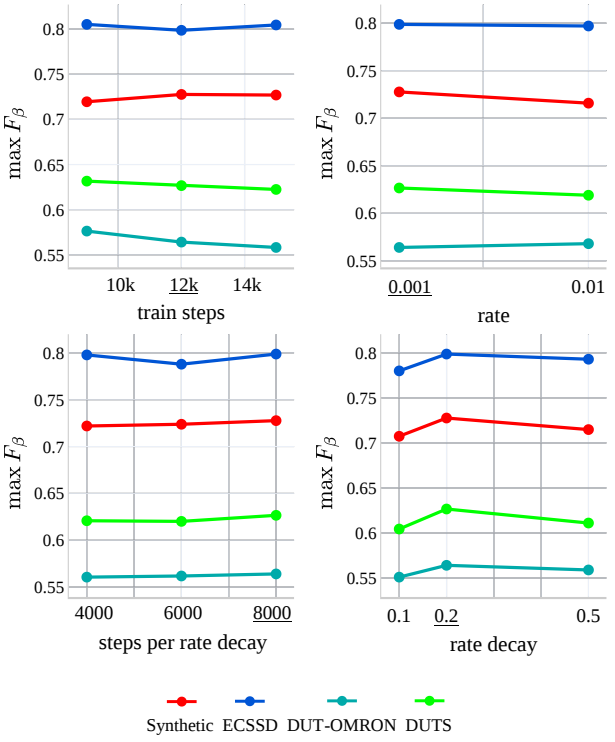

Figure 5. Model performance for different hyperparameters. We take the hyperparameters that performs best on the synthetic data.

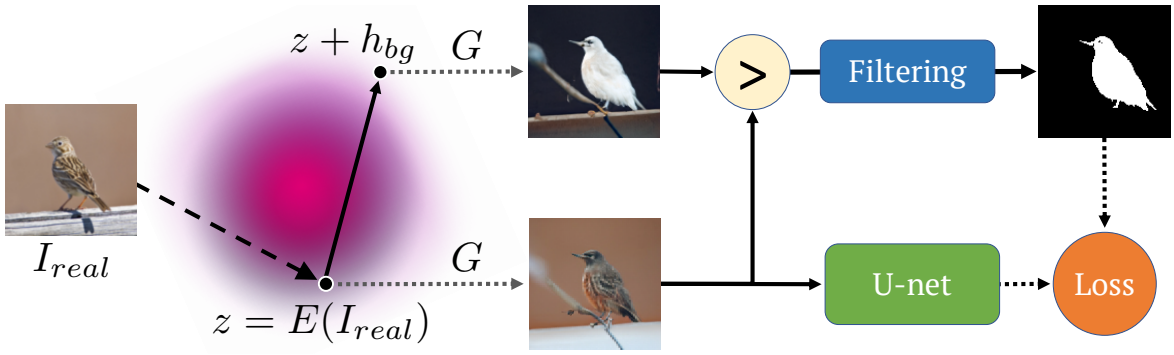

Figure 4. Schematic representation of our approach. First, we generate an image $G(z)$ and the shifted image $G(z + h_{bg})$. Then these images induce a synthetic mask. This mask is passed through a filtering (Section 3.4) and a U-net segmentation model learns to predict it, given the original generated image.

4. Experiments

This section aims to confirm that the usage of GAN-produced synthetic data is a promising direction for unsupervised saliency detection and object segmentation. To this end, we extensively compare our approach to the existing unsupervised counterparts on the standard benchmarks.

Evaluation metrics. All the methods are compared in terms of the three measures described below.

- **F-measure** is an established measure in the saliency detection literature. It is defined as $F_\beta = \frac{(1+\beta^2) Precision \times Recall}{\beta^2 Precision + Recall}$. Here Precision and Recall are calculated based on the binarized predicted masks and groundtruth masks as $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, where TP, TN, FP, FN denote true-positive, true-negative, false-positive, and false-negative, respectively. We compute F-measure for 255 uniformly distributed binarization thresholds and report its maximum value $max\ F_\beta$. We use $\beta = 0.3$ for consistency with existing works.
- **IoU** (Intersection over Union) is calculated on the binarized predicted masks and groundtruth as $IoU(s, m) = \frac{\mu(s \cap m)}{\mu(s \cup m)}$, where μ denotes the area. The binarization threshold is set to 0.5.
- **Accuracy** measures the proportion of pixels that have been correctly assigned to the object/background. The binarization threshold for masks is set to 0.5.

Since the existing literature uses different benchmark datasets for saliency detection and object segmentation, we perform a separate comparison for each task below.

4.1. Object segmentation.

Datasets. We use two following datasets from the literature of segmentation with generative models.

- **Caltech-UCSD Birds 200-2011** (Wah et al., 2011) contains 11,788 photographs of birds with segmentation masks. We follow (Chen et al., 2019), and use 10,000 images for our training subset and 1,000 for the test subset from splits provided by (Chen et al., 2019). Unlike (Chen et al., 2019), we do not use any images for validation and simply omit the remaining 788 images.
- **Flowers** (Nilsback & Zisserman, 2007) contains 8,189 images of flowers equipped with saliency masks generated automatically via the method developed for flowers. We do not apply the mask area filter in our method with this dataset, as it rejects most of the samples.

On these two datasets, we compare the following methods:

- **PerturbGAN** (Bielski & Favaro, 2019) segments an image based on the idea that object location can be perturbed without affecting the scene realism. For comparison, we use the numbers reported in (Bielski & Favaro, 2019).
- **ReDO** (Chen et al., 2019) produces segmentation masks based on the idea that object appearance can be changed without affecting image quality. For comparison, we report the numbers from (Chen et al., 2019). Note, (Chen et al., 2019) use hold-out labeled sets to set hyperparameters.
- **OneGAN** (Benny & Wolf, 2020) which simultaneously learning a conditional image generator, foreground extraction and segmentation, clustering, and object removal and background completion. Note that (Benny & Wolf, 2020) uses bounding boxes from image annotations to cut background patches, which is a source of weak supervision.
- **BigBiGAN** is our method where the latent codes are sampled from $z \sim \mathcal{N}(0, \mathbb{I})$. For Flowers dataset we

Method	CUB-200-2011			Flowers		
	max F_β	IoU	Accuracy	maxF_β	IoU	Accuracy
PerturbGAN	—	0.380	—	—	—	—
ReDO	—	0.426	0.845	—	0.764	0.879
OneGAN	—	0.555	—	—	—	—
BigBiGAN	0.794	0.683	0.930	0.760	0.540	0.765
E-BigBiGAN (w/o z-noising)	0.750	0.619	0.918	0.814	0.689	0.874
E-BigBiGAN (with z-noising)	**0.834**	**0.710**	**0.940**	**0.878**	**0.804**	**0.904**
std	0.005	0.007	0.002	0.001	<0.001	<0.001

Table 1. The comparison of unsupervised object segmentation methods. For our model, we report the performance averaged over ten runs. For the best model, we also report the standard deviation values.

Method	ECSSD			DUTS			DUT-OMRON		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy
HS	0.673	0.508	0.847	0.504	0.369	0.826	0.561	0.433	0.843
wCtr	0.684	0.517	0.862	0.522	0.392	0.835	0.541	0.416	0.838
WSC	0.683	0.498	0.852	0.528	0.384	0.862	0.523	0.387	**0.865**
DeepUSPS	0.584	0.440	0.795	0.425	0.305	0.773	0.414	0.305	0.779
BigBiGAN	0.782	0.672	0.899	0.608	0.498	0.878	0.549	0.453	0.856
E-BigBiGAN	**0.797**	**0.684**	**0.906**	**0.624**	**0.511**	**0.882**	**0.563**	**0.464**	0.860

Table 2. The comparison of unsupervised saliency detection methods. For BigBiGAN and E-BigBiGAN we report the mean values over 10 independent runs.

found it beneficial to generate the saliency masks by thresholding the shifted image $G(z + h_{bg})$ with its mean value. Thus, for Flowers the masks are generated as $M = [G(z + h_{bg}) > \text{mean}(G(z + h_{bg}))]$.

- **E-BigBiGAN (w/o z-noising)** is our method where the latent codes of synthetic data are sampled from the outputs of the encoder E applied to the train images of the dataset at hand.

- **E-BigBiGAN (with z-noising)** same as above with latent codes sampled from the vicinity of the embeddings with the neighborhood size α set to 0.2.

Following the prior works, we apply image preprocessing by extracting a central crop and resizing it to 128×128. The comparison results are provided in Table 1, which demonstrates the significant advantage of our scheme. Note, since both datasets in this comparison are small-scale, z-noising considerably improves the performance, increasing the diversity of training images.

4.2. Saliency detection.

Datasets. We use the following established benchmarks for saliency detection. For all the datasets, groundtruth pixel-level saliency masks are available.

- **ECSSD** (Shi et al., 2015) contains 1,000 images with structurally complex natural contents.

- **DUTS** (Wang et al., 2017a) contains 10,553 train and 5,019 test images. The train images are selected from the ImageNet detection train/val set. The test images are selected from the ImageNet test, and the SUN dataset (Xiao et al., 2010). We always report the performance on the DUTS-test subset.

- **DUT-OMRON** (Yang et al., 2013) contains 5,168 images of high content variety.

Baselines. While there are a large number of papers on unsupervised deep saliency detection, all of them employ pretrained supervised models in their training protocols.

Figure 6. Top: Images from the DUTS-test dataset. *Middle:* Groundtruth masks. *Bottom:* Masks produced by the E-BigBiGAN method.

Method	ECSSD			DUTS			DUT-OMRON		
	$maxF_\beta$	IoU	Accuracy	$maxF_\beta$	IoU	Accuracy	$maxF_\beta$	IoU	Accuracy
(Voynov & Babenko, 2020)	0.778	0.648	0.904	0.604	0.478	**0.889**	0.56	0.444	**0.878**
BigBiGAN (base)	0.737	0.626	0.859	0.575	0.454	0.817	0.498	0.389	0.758
E-BigBiGAN	**0.797**	**0.684**	**0.906**	**0.624**	**0.511**	0.882	**0.563**	**0.464**	0.860

Table 3. Comparison of our method with the weakly-supervised BigGAN-based approach.

Method	ECSSD			DUTS			DUT-OMRON		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy
Base	0.737	0.626	0.859	0.575	0.454	0.817	0.498	0.389	0.758
+Imagenet embeddings	0.773	0.657	0.874	0.616	0.483	0.832	0.533	0.413	0.772
+Size filter	0.781	0.670	0.900	0.62	0.499	0.871	0.552	0.443	0.842
+Histogram	0.779	0.670	0.900	0.621	0.503	0.875	0.555	0.450	0.850
+Connected components	**0.797**	**0.684**	**0.906**	**0.624**	**0.511**	**0.882**	**0.563**	**0.464**	**0.860**

Table 4. Impact of different components in the E-BigBiGAN pipeline.

For example, DeepUSPS (Nguyen et al., 2019) uses a segmentation model pretrained on CityScapes dataset (Cordts et al., 2016). Therefore, we mostly use the recent "shallow" methods HS (Yan et al., 2013), wCtr (Zhu et al., 2014a), and WSC (Li et al., 2015) as the baselines. These three methods were chosen based on their state-of-the-art performance reported in the literature and publicly available implementations. To compare with deep saliency detection models, we also add DeepUSPS (Nguyen et al., 2019) to the list of baselines. However, to perform a fair comparison, we train the DeepUSPS model without pretraining on the CityScapes dataset. The results for all methods are reported in Table 2. In this table, BigBiGAN denotes the version of our method where the latent codes of synthetic images are sampled from $z \sim \mathcal{N}(0, \mathbb{I})$. In turn, in E-BigBiGAN, z are sampled from the latent codes of Imagenet-train images for all three datasets. Since the Imagenet dataset is large enough, we do not employ z-noising in this comparison.

As one can see, our method mostly outperforms the competitors by a considerable margin, which confirms the promise of using synthetic imagery in unsupervised scenarios. Note that DeepUSPS shows weak results using the same amount of supervision as our method. Several qualitative segmentation samples are provided on Figure 6.

4.3. Is BigGAN's supervision necessary for the segmentation performance?

In Table 3 we compare our method with the approach proposed in (Voynov & Babenko, 2020). Though this method is not fully unsupervised, it is interesting to compare synthetic from supervised and unsupervised GANs. (Voynov & Babenko, 2020) utilized the "background removal" direction in the BigGAN's latent space to generate foreground / background masks. As BigGAN has no encoder, we also compare it with a weaker version of our method that uses the prior latent distribution without any filtering (see Table 4, first line). Notably, even without any adaptation to the particular dataset and filtering, our method performs on par with the "supervised" one. Enriched with the adaptation step, our approach outperforms (Voynov & Babenko, 2020) while being unsupervised. These results are quite surprising since BigGAN has remarkably higher generation quality with the Fréchet Inception Distance (FID) of 10.2 facing 23.3 for BigBiGAN.

4.4. Ablation.

In Table 4 we demonstrate the impact of individual components in our method. First, we start with a saliency detection model trained on the synthetic data pairs $\{G(z), M = [G(z+h_{bg}) > G(z)]\}$ with $z \sim \mathcal{N}(0, I)$. Then we add one by one the components listed in Section 3.3 and Section 3.4. The most significant performance impact comes from using the latent codes of the real images from the Imagenet.

4.5. Synthetic Data Quality

This section compares the quality of saliency masks obtained with our method with the real ones. First, we evaluate the consistency of the real and generated masks. We use the SOTA publicly available saliency model[1] to evaluate the quality of our synthetic masks used for the best E-

[1] https://github.com/NathanUA/U-2-Net

Method	ECSSD			DUTS			DUT-OMRON		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy
E-BigBiGAN with U^2Net	**0.813**	0.674	**0.911**	**0.654**	**0.525**	**0.906**	**0.663**	**0.559**	**0.915**
E-BigBiGAN	0.797	**0.684**	0.906	0.624	0.511	0.882	0.563	0.464	0.860

Table 5. Comparison of masks generation with supervised U^2Net-guided synthetic labeling

BigBiGAN run with the Imagenet embeddings. The model results in 0.412 IoU and 0.720 accuracy on 10^5 random samples, which is lower than our scheme's performance on the real datasets. We attribute such behavior to the fact that our synthetic data is often noisy, and the model trained on human-provided masks is not robust to the noise. In other words, it is more beneficial to train on difficult, noisy data and to test on refined, clean data rather than otherwise.

To understand whether the performance bottleneck of our method is the quality of generated images or the quality of masks, we perform the following experiment. Using the same SOTA supervised model $\mathrm{U}^2\mathrm{Net} : \mathbb{R}^{3\times128\times128} \to \{0,1\}^{128\times128}$ as above, we take randomly sampled latent z formed by the E-BigBiGAN pipeline and form the dataset of the pairs $\{G(z),\ \mathrm{U}^2\mathrm{Net}(G(z))\}$ were G is the BigBiGAN generator. In other words, we take the same images as in our best method and form the masks with a high-quality saliency model pretrained on the real data. Then we train a U-net segmentation model on this data following the protocol described in Section 3.5. The comparison of the original model with the U^2Net-guided model is presented in Table 5. Notably, the U^2Net-guided model performs better, though it does not demonstrate a break-through outperformance on two datasets out of three. This result indicates that the bottleneck of our method mainly lies in the quality of generated images rather than the mask generation approach.

4.6. Independent Masks Estimation

In principle, the optimization problem (4) that induces the segmentation masks could be solved independently for each generated image. We have tried to optimize the operators A_1, A_2 independently for each generated sample $G(z)$ during the synthetic data generation. In this case, the resulting masks appear to be almost the same as masks produced by the operators optimized for all synthetic images simultaneously. In Figure 7 we present the masks generated in the original protocol (red) and the masks generated with the per-sample optimization (green). Mutual IoU and accuracy of these masks computed over 128 randomly generated samples is 0.86 and 0.96 respectfully, therefore, they are very similar. Even though the latent shift may indeed act with a different strength, the optimal operators' indices from Equation 2 seem to be rather persistent. We should also note that this per-sample training protocol would require an enormous extra time as single batch generation takes approximately 30 seconds.

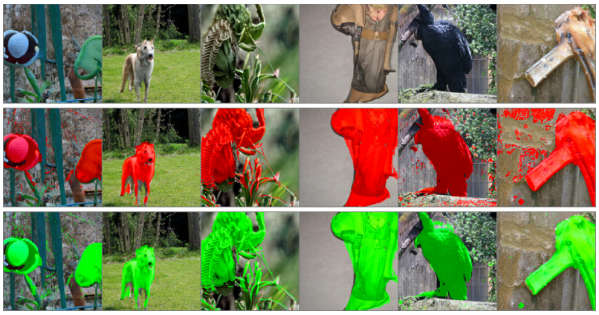

Figure 7. First row: generated images $G(z)$; *second row*: masks generated with operators A_1, A_2 that optimize the common objective from Equation 3; *third row*: masks generated with the operators A_1, A_2 independently optimized for each sample.

5. Conclusion

In our paper, we continue the line of works on unsupervised object segmentation with the aid of generative models. While the existing unsupervised techniques require adversarial training, we introduce an alternative research direction based on the high-quality synthetic data from the off-the-shelf GAN. Namely, we utilize the images produced by the BigBiGAN model, which is trained on the Imagenet dataset. Exploring BigBiGAN, we have discovered that its latent space semantics automatically create the saliency masks for synthetic images via latent space manipulations. We propose to use the BigBiGAN's encoder to fit this pipeline for a particular dataset. As shown in experiments, this synthetic data is an excellent source of supervision for discriminative computer vision models. The main feature of our approach is its simplicity and reproducibility since our model does not rely on a large number of components/hyperparameters. On several standard benchmarks, we demonstrate that our method achieves superior performance compared to existing unsupervised competitors.

References

Benny, Y. and Wolf, L. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. *ECCV*, 2020.

Bielski, A. and Favaro, P. Emergence of object segmentation in perturbed generative models. In *Advances in Neural Information Processing Systems*, pp. 7254–7264, 2019.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chen, M., Artières, T., and Denoyer, L. Unsupervised object segmentation by redrawing. In *Advances in Neural Information Processing Systems*, pp. 12705–12716, 2019.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020a.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020b.

Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H., and Hu, S.-M. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2014.

Cong, R., Lei, J., Fu, H., Huang, Q., Cao, X., and Hou, C. Co-saliency detection for rgbd images based on multi-constraint feature matching and cross label propagation. *IEEE Transactions on Image Processing*, 27(2):568–579, 2017.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Donahue, J. and Simonyan, K. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pp. 10541–10551, 2019.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Guo, F., Wang, W., Shen, J., Shao, L., Yang, J., Tao, D., and Tang, Y. Y. Video saliency detection using object proposals. *IEEE transactions on cybernetics*, 48(11): 3159–3170, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., and Li, S. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2083–2090, 2013.

Krähenbühl, P. and Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pp. 109–117, 2011.

Li, N., Sun, B., and Yu, J. A weighted sparse coding framework for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5216–5223, 2015.

Nguyen, T., Dax, M., Mummadi, C. K., Ngo, N., Nguyen, T. H. P., Lou, Z., and Brox, T. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *Advances in Neural Information Processing Systems*, pp. 204–214, 2019.

Nilsback, M.-E. and Zisserman, A. Delving into the whorl of flower segmentation. In *BMVC*, volume 2007, pp. 1–10, 2007.

Peng, H., Li, B., Ling, H., Hu, W., Xiong, W., and Maybank, S. J. Salient object detection via structured matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):818–832, 2016.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Shi, J., Yan, Q., Xu, L., and Jia, J. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):717–729, 2015.

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. Scan: Learning to classify images without labels. In *European Conference on Computer Vision*, pp. 268–285. Springer, 2020.

Voynov, A. and Babenko, A. Unsupervised discovery of interpretable directions in the gan latent space. *arXiv preprint arXiv:2002.03754*, 2020.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., and Ruan, X. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136–145, 2017a.

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., and Ruan, X. Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 136–145, 2017b.

Wang, W., Shen, J., Shao, L., and Porikli, F. Correspondence driven saliency transfer. *IEEE Transactions on Image Processing*, 25(11):5025–5034, 2016.

Wang, W., Lai, Q., Fu, H., Shen, J., and Ling, H. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019.

Wei, Y., Wen, F., Zhu, W., and Sun, J. Geodesic saliency using background priors. In *IEEE, ICCV*, 2012.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492. IEEE, 2010.

Yan, Q., Xu, L., Shi, J., and Jia, J. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1155–1162, 2013.

Yang, C., Zhang, L., Lu, H., Ruan, X., and Yang, M.-H. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3166–3173, 2013.

Zhang, D., Han, J., and Zhang, Y. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4048–4056, 2017.

Zhang, D., Tian, H., and Han, J. Few-cost salient object detection with adversarial-paced learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12236–12247. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/8fc687aa152e8199fe9e73304d407bca-Paper.pdf.

Zhang, J., Zhang, T., Dai, Y., Harandi, M., and Hartley, R. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9029–9038, 2018.

Zheltonozhskii, E., Baskin, C., Bronstein, A. M., and Mendelson, A. Self-supervised learning for large-scale unsupervised image clustering. *arXiv preprint arXiv:2008.10312*, 2020.

Zhu, W., Liang, S., Wei, Y., and Sun, J. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014a.

Zhu, W., Liang, S., Wei, Y., and Sun, J. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014b.