BIG GANS ARE WATCHING YOU: TOWARDS UNSUPERVISED OBJECT SEGMENTATION WITH OFF-THE-SHELF GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Since collecting pixel-level groundtruth data is expensive, unsupervised visual understanding problems are currently an active research topic. In particular, several recent methods based on generative models have achieved promising results for object segmentation and saliency detection. However, since generative models are known to be unstable and sensitive to hyperparameters, the training of these methods can be challenging and time-consuming.

In this work, we introduce an alternative, much simpler way to exploit generative models for unsupervised object segmentation. First, we explore the latent spaces of the publicly available unsupervised models, such as BigBiGAN and StyleGAN2, and reveal the "segmenting" latent directions that can be used to obtain saliency masks for GAN-produced images. These masks are then used to train a discriminative segmentation model. Being very simple and easy-to-reproduce, our approach outperforms the state-of-the-art on common benchmarks in the unsupervised setting. All the code and the pretrained models are available online[1].

1 INTRODUCTION

Deep convolutional models are a core instrument for visual understanding problems, including object localization (Zhou et al., 2016; Choe et al., 2020), saliency detection (Wang et al., 2019), segmentation (Long et al., 2015) and others. Deep CNNs, however, require a large amount of high-quality training data to fit a huge number of learnable parameters. In practice, obtaining pixel-level labeling is expensive, since it requires labor-intensive human efforts. Therefore, much research attention has currently focused on weakly-supervised and unsupervised approaches for pixel-level tasks, such as segmentation (Xia & Kulis, 2017; Ji et al., 2019; Chen et al., 2019; Bielski & Favaro, 2019).

An emerging line of research on unsupervised segmentation exploits generative models as a tool for image decomposition. Namely, recent works (Chen et al., 2019; Bielski & Favaro, 2019) have designed training protocols that include generative adversarial networks (GANs), to solve the foreground object segmentation without human labels. Given the promising results and the fact that the GANs' performance is steadily improving, this research direction will be likely developed in the future.

In practice, however, training high-quality generative models is challenging. This is especially the case for GANs, which training can be both time-consuming and unstable. Moreover, the models in Chen et al. (2019); Bielski & Favaro (2019) 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. To this end, we propose an alternative way to exploit GANs for unsupervised segmentation, which does not train a separate generative model for each task. Instead, we use publicly available GANs to produce synthetic images equipped with segmentation masks, which can be obtained automatically.

Our work is partially inspired by the findings from Voynov & Babenko (2020) that has shown that the latent space of BigGAN (Brock et al., 2018) possess the direction "responsible" for the background removal, which can be used to produce training data for saliency detection. However, approaches using BigGAN samples cannot be considered fully unsupervised, since BigGAN is trained with

[1] https://github.com/gans-are-watching/iclr2021_submit

supervision from the Imagenet class labels. Moreover, BigGAN cannot naturally adapt to a particular segmentation task, which can be biased from Imagenet, where BigGAN was trained.

In this paper, we prove that large off-the-shelf GANs can segment images, being completely unsupervised. Namely, we reveal "segmenting" directions in both state-of-the-art publicly available unsupervised GANs: BigBiGAN (Donahue & Simonyan, 2019), which is an unsupervised GAN trained on the Imagenet (Deng et al., 2009); and a very recent StyleGAN2 model (Karras et al., 2020). 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. Note, our approach allows for a straightforward way to tune hyperparameters. Since synthetic data is unlimited, its hold-out subset can be used as validation. As another technical contribution, we propose a simple yet effective approach to adapt off-the-shelf GANs to produce synthetic data for the the particular segmentation task at hand. Despite its simplicity, this adaptation dramatically increases segmentation performance.

Our work confirms the promise of using GANs to produce synthetic training data, which is a long-standing goal of research on generative modeling. We show that the approach outperforms the existing unsupervised alternatives for object segmentation and saliency detection. Furthermore, it performs on par with weakly-supervised methods for object localization, being fully unsupervised.

The main contributions of our paper are the following:

1. We introduce an alternative line of research on using GANs for unsupervised object segmentation. In a nutshell, we advocate the usage of high-quality synthetic data produced by publicly available unsupervised GANs to train discriminative segmentation models.
2. We propose a procedure that employs off-the-shelf GANs to produce synthetic data that is appropriate for a particular segmentation task. Being fast and simple, this procedure substantially increases the effectiveness of our protocol.
3. We show that our method outperforms the state-of-the-art in most operating points. Given its simplicity and reproducibility, the method can serve as a useful 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). 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 completely unsupervised, since they typically rely on the pretrained classification or segmentation networks. In contrast, in this work, we focus on the methods that do not require any source of external supervision.

Generative models for object segmentation. The recent line of completely unsupervised methods (Chen et al., 2019; Bielski & Favaro, 2019) employs generative modeling to decompose the image into the object and the background. In a nutshell, these methods exploit the idea that the object 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 is known to be unstable, time-consuming, and sensitive to hyperparameters.

In contrast, our approach avoids these disadvantages, being much simpler and easier to reproduce. In essence, we propose to use the "inner knowledge" of the off-the-shelf large-scale GAN to produce the saliency masks for synthetic images and use them as a supervision for discriminative models.

Latent spaces of large-scale GANs. Our study is inspired by the recent findings from Voynov & Babenko (2020). This work introduces an unsupervised technique that discovers the directions in the GAN latent space corresponding to interpretable image transformations. Among its findings, Voynov & Babenko (2020) demonstrates that the large-scale conditional GAN (BigGAN Brock et al. (2018)) possesses a "background removal" direction that can be used to obtain saliency masks. However, this direction was discovered only for BigGAN that was trained under the supervision from the image class labels. For unconditional GANs, such a direction 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. In this paper, we show that this supervision is not necessary, contributing novel knowledge to the general trend to unsupervised learning.

3 METHOD

3.1 EXPLORING THE LATENT SPACES OF UNSUPERVISED GANS.

In this section we investigate the latent spaces of two state-of-the-art unsupervised GANs: *(i)* the recent BigBiGAN model (Donahue & Simonyan, 2019) trained on the Imagenet (Deng et al., 2009) without labels and its parameters are available online[2]; *(ii)* the StyleGAN2 model (Karras et al., 2020) trained of the LSUN-Church dataset. In both models, a generator G maps the samples $z \sim \mathcal{N}(0, \mathbb{I})$ from the latent space $\mathbb{R}^d$ into the image space $G : z \to I$. BigBiGAN is also equipped with an encoder $E : I \to z$ that was trained jointly with the generator and maps images to the latent space.

We explore the latent spaces of both models via a recent unsupervised technique (Voynov & Babenko, 2020) that identifies interpretable directions in the latent space of a pretrained GAN. By moving a latent code z in these directions, one can achieve different image transformations, such as image zooming or translation. Formally, given an image corresponding to a latent code z, one can modify it via shifting the code in an interpretable direction h. Then a modified image $G(z{+}h)$ can be generated. Importantly, h operates consistently over the whole latent space, i.e. for all z, shifting results in the same type of transformation. As the first step of our study, we apply the technique from Voynov & Babenko (2020) to the generators of BigBiGAN and StyleGAN2 to explore the potential of their latent spaces. In a nutshell, Voynov & Babenko (2020) seeks to learn K directions in the latent space $h_1, \dots, h_K$ such that the effects of the corresponding image transformations are "disentangled". More formally, the sets of pairs $\{G(z), G(z{+}h_i) | z \sim \mathcal{N}(0, \mathbb{I})\}$ with different $i{=}1, \dots, K$ aims to be easy to distinguish from each other by a CNN classifier, which is trained jointly with $h_1, \dots, h_K$.

We use the authors' implementation[3] with default hyperparameters and the number of directions $K{=}120$. For StyleGAN2, the *W*-space was explored. After learning converged, we inspect the directions manually and filter out the interpretable ones, notably this process takes about two minutes by a single person.[4] Several revealed directions are visualized in Figure 5 and Figure 6 in Appendix.

Compared to the results from Voynov & Babenko (2020) for the supervised BigGAN, the latent spaces of BigBiGAN and StyleGAN2 do not possess any directions that have clear "background removal" effect. However, they both possess directions that have different effects on the object and background pixels. The BigBiGAN's corresponding transformation "Saliency lighting" is presented on Figure 1 and we refer to this direction as h_{bg}. As one can see, moving along it makes the object pixels lighter, while the background pixels become darker. Therefore, despite BigBiGAN is completely unsupervised, its latent space can be used to obtain saliency masks for generated images. Technically, we produce a binary saliency mask M for an image $G(z)$ by comparing its intensity with the "shifted"

[2] `https://tfhub.dev/deepmind/bigbigan-resnet50/1`

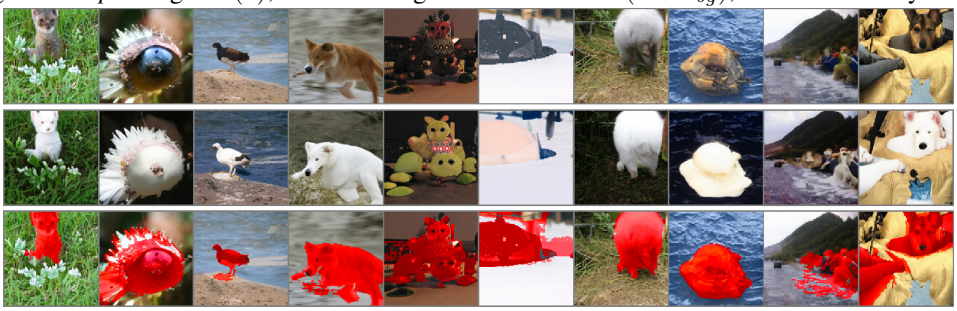
[3] `https://github.com/anvoynov/GANLatentDiscovery`

[4] While, rigorously speaking, this inspection introduces a minor amount of human supervision, in the experiments below we show that this amount is much smaller compared to existing baselines, which use hold-out labeled sets to set hyperparameters. Moreover, in Appendix we describe a technique that reduces the number of "segmenting" directions candidates up to only a few ones.

image $M = [G(z+h_{bg}) > G(z)]$ after grayscale conversion. As a shift magnitude, we always use $||h_{bg}||=5$. Examples of generated images and the corresponding masks are provided on Figure 2.

Figure 1: Samples of latent shift along the saliency lighting direction.

Figure 2: *Top:* images $G(z)$; *Middle:* images after the shift $G(z + h_{bg})$; *Bottom*: saliency masks



3.2 Adaptation to the particular segmentation task.

Since BigBiGAN was trained on the Imagenet, sampling the latent codes from the standard Gaussian distribution $z \sim \mathcal{N}(0, \mathbb{I})$ will result in the synthetics distribution 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 at hand $\mathcal{I}=\{I_1, \dots, I_N\}$, 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 $\{E(I_1), \dots, E(I_N)\} \subset \mathbb{R}^{120}$ and sample the codes from the neighborhood of these representations. Formally, the samples have the form $\{E(I_i)+\alpha\xi \mid i \sim \mathcal{U}\{1, N\}, \xi \sim \mathcal{N}(0, I)\}$. Here α denotes the neighborhood size and it should be larger for small $\mathcal{I}$ to prevent overfitting. In particular, we use $\alpha=0$ for Imagenet and $\alpha=0.2$ for all other cases. In the experimental section, we demonstrate that this simple and efficient modification of data generation process results in dramatic performance boost.

3.3 Improving saliency masks.

Here we describe a few simple heuristics that increase the quality of the masks for the particular segmentation task. The ablation of each component is presented in Section A.1.

Mask size filtering. Since some of the BigBiGAN-produced images are low-quality and do not contain clear objects, the corresponding masks can result in a very noisy supervision. To avoid this, we apply a 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 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 $M_1, \dots, M_k$. 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.

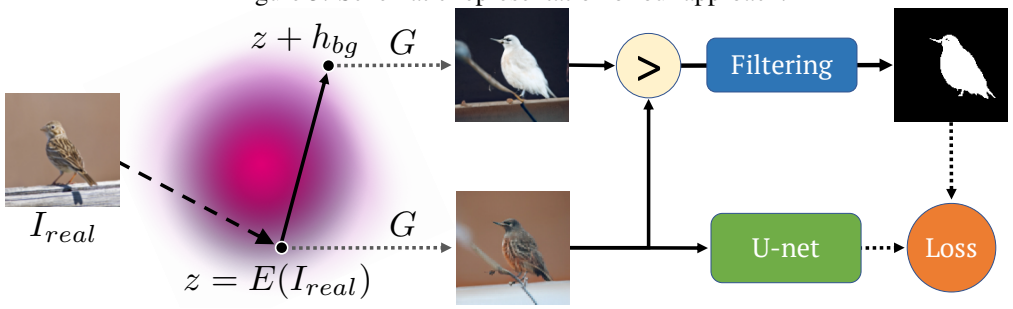
We present samples of images rejected by each filtering step in Figure 7 in Appendix.

3.4 Training 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 3. In all our experiments we employ a simple U-net architecture Ronneberger et al. (2015). We train U-net on the synthetic dataset with 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 extremely simple, 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. Training with on-line synthetic data generation takes approximately seven hours on two Nvidia 1080Ti cards.

Figure 3: Schematic representation of our approach.



4 Experiments

The goal of this section is 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 $F_\beta = \frac{(1+\beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}$. 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_β. We use β=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 11788 photographs of birds with segmentation masks. We follow Chen et al. (2019), and use 10000 images for our training subset and 1000 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 8189 images of flowers equipped with saliency masks generated automatically via the method developed for flowers. With this dataset, we do not apply the mask area filter in our method, 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.
- **BigBiGAN** is our method where the latent codes are sampled from $z \sim \mathcal{N}(0, \mathbb{I})$. For Flowers dataset we found benefitical 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 images preprocessing by central crop and resize 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.

Method	CUB-200-2011			Flowers		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy
PerturbGAN	—	0.380	—	—	—	—
ReDO	—	0.426	0.845	—	0.764	0.879
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.

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. Therefore, we use the most 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. The results of the comparison 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 the unsupervised scenario. Several qualitative segmentation samples are provided on Figure 4.

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**
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.

Figure 4: *Top:* Images from the DUTS-test dataset. *Middle:* Groundtruth masks. *Bottom:* Masks produced by the E-BigBiGAN method.

4.3 Weakly-supervised object localization (WSOL)

A closely related to the segmentation problem is the object localization, where for a given image one has to provide a bounding box instead of a segmentation mask. In this section, we demonstrate that our unsupervised method performs on par with the weakly-supervised state-of-the-art. To compare with the previous literature, we use the numbers from the very recent evaluation paper by Choe et al. (2020) that reviews a large number of existing WSOL methods and reports actual state-of-the-art. We employ exactly the same evaluation protocols as in Choe et al. (2020) and compare the prior works with our E-BigBiGAN method, which samples z from the latent codes of Imagenet-train images, as described in Section 3.3. The comparison results are provided in Table 3.

Evaluation metrics. For the WSOL problem we use the following metrics (Choe et al., 2020):

- **MaxBoxAcc** (Russakovsky et al., 2015; Zhou et al., 2016). For an image I_n, let us have a predicted mask s_n and a set of ground truth bounding boxes $B_n^{(i)}$ for $i = 1, \dots, m$ (some datasets can provide several bounding boxes per image). Let us select a threshold $\tau \in [0, 1]$ and denote c_n^τ the largest (in terms of the area) connected component of the mask s_n binarized with threshold τ. Let us denote with $\text{box}(c_n^\tau)$ the minimal bounding box containing the set c_n^τ. Then we define

$$\text{BoxAcc}(\tau) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{\text{IoU}(\text{box}(c_n^\tau), B_n^{(j)}) \geq 0.5} \tag{1}$$

  where $B_n^{(j)}$ corresponds to the ground truth bounding box with the maximal IoU with $\text{box}(c_n^\tau)$ and N denotes the number of images. Then the final metrics MaxBoxAcc is the maximum of $\text{BoxAcc}(\tau)$ over all thresholds τ.

- **PxAP** Achanta et al. (2009). Let us have a predicted mask s_n and ground truth mask t_n. For a threshold $\tau \in [0, 1]$ we define a pixel precision and recall

$$\text{P}_\tau = \frac{1}{N} \sum_{n=1}^{N} \frac{|\{s_n \geq \tau\} \cap \{t_n = 1\}|}{|\{s_n \geq \tau\}|}; \quad \text{R}_\tau = \frac{1}{N} \sum_{n=1}^{N} \frac{|\{s_n \geq \tau\} \cap \{t_n = 1\}|}{|\{t_n = 1\}|} \tag{2}$$

  We average both values over all images and then PxAP is defined as the area under curve of the pixel precision-recall curve.

Datasets. We use the following benchmarks for weakly-supervised object localization.

- **Imagenet** (Russakovsky et al., 2015). For evaluation we use $10,000$ validation images. The dataset contains several annotated bounding boxes for each image.
- **Caltech-UCSD Birds 200-2011** (Wah et al., 2011). For evaluation we use $5,794$ test images.
- **OpenImages** (Choe et al., 2020) contains a subset of OpenImages instance segmentation dataset Benenson et al. (2019). For evaluation we use $5,000$ randomly selected images from 100 classes as in Choe et al. (2020).

Method	Imagenet (MaxBoxAcc)	CUB (MaxBoxAcc)	OpenImages (PxAP)
Previous SOTA (Choe et al., 2020)	0.654	0.781	0.630
E-BigBiGAN	0.614	0.742	0.638

Table 3: The comparison of E-BigBiGAN to the WSOL state-of-the-art. For E-BigBiGAN we report the mean values over 10 independent runs. Despite being completely unsupervised, E-BigBiGAN performs on par with the WSOL methods, which were trained under more supervision.

4.4 Is BigGAN's supervision necessary for the segmentation performance?

In Table 4 we compare our method with the approach proposed in Voynov & Babenko (2020). Though the last is not fully unsupervised as it is based on the conditional Imagenet-BigGAN, it is interesting to compare the performance of supervised and unsupervised GANs of the same architecture. 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 the weaker version of our method that uses the prior latent distribution without any filtering (see Table 5, 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 fully unsupervised. This results are quite surprising, since BigGAN has remarkably higher generated images quality with the Fréchet Inception Distance (FID) of 10.2 facing 23.3 for BigBiGAN.

Method	ECSSD			DUTS			DUT-OMRON		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy	maxF_β	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 4: Comparison of our method with the weakly-supervised BigGAN-based approach.

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 allows to automatically produce 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 common benchmarks, we demonstrate that our method achieves superior performance compared to existing unsupervised competitors.

We also highlight the fact that the state-of-the-art generative models, such as BigBiGAN, can be successfully used to generate training data for yet another computer vision task. We expect that other problems such as semantic segmentation can also benefit from the usage of GAN-produced data in the weakly-supervised or few-shot regimes. Since the quality of GANs will likely improve in the future, we expect that the usage of synthetic data will become increasingly widespread.

REFERENCES

Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 1597–1604. IEEE, 2009.

Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11700–11709, 2019.

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

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

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

Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2014.

Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. 2020.

Runmin Cong, Jianjun Lei, Huazhu Fu, Qingming Huang, Xiaochun Cao, and Chunping Hou. 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.

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

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

Fang Guo, Wenguan Wang, Jianbing Shen, Ling Shao, Jian Yang, Dacheng Tao, and Yuan Yan Tang. Video saliency detection using object proposals. *IEEE transactions on cybernetics*, 48(11): 3159–3170, 2017.

Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9865–9874, 2019.

Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 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.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020.

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

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

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *Advances in Neural Information Processing Systems*, pp. 204–214, 2019.

Maria-Elena Nilsback and Andrew Zisserman. Delving into the whorl of flower segmentation. In *BMVC*, volume 2007, pp. 1–10, 2007.

Houwen Peng, Bing Li, Haibin Ling, Weiming Hu, Weihua Xiong, and Stephen J Maybank. Salient object detection via structured matrix decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):818–832, 2016.

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

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

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

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

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 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.

Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. 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.

Wenguan Wang, Jianbing Shen, Ling Shao, and Fatih Porikli. Correspondence driven saliency transfer. *IEEE Transactions on Image Processing*, 25(11):5025–5034, 2016.

Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, and Haibin Ling. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019.

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

Xide Xia and Brian Kulis. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*, 2017.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 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.

Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1155–1162, 2013.

Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3166–3173, 2013.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Dingwen Zhang, Junwei Han, and Yu Zhang. 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.

Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard Hartley. 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.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014a.

Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014b.

A Appendix

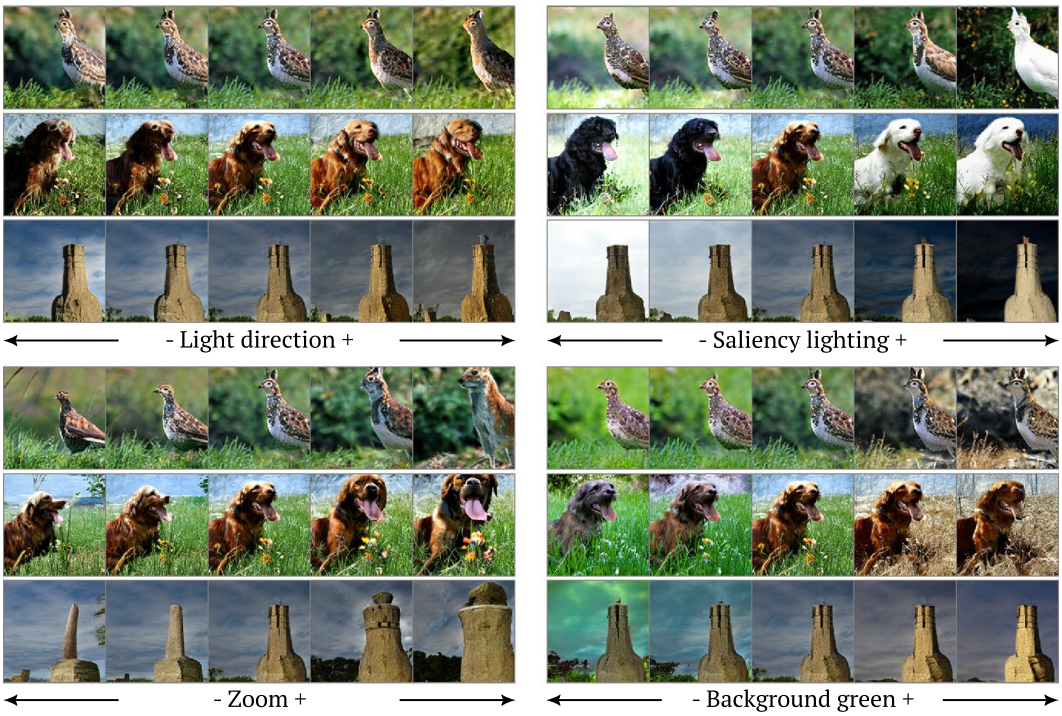


Figure 5: Examples of interpretable directions discovered in the BigBiGAN latent space.

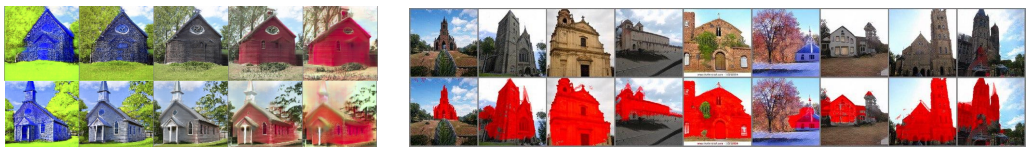


Figure 6: Latent directions and saliency masks from the StyleGAN2 trained on the LSUN-Church dataset.

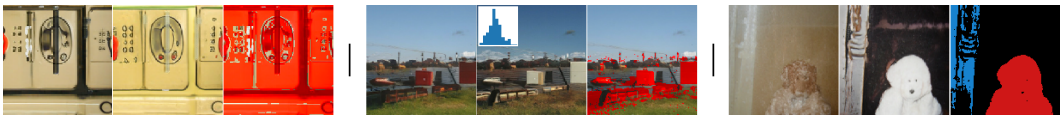


Figure 7: Examples of mask improvement. *Left:* sample rejected by the mask size filter. *Middle:* sample rejected by the histogram filtering. *Right block*: mask pixels removed by the connected components filter are shown in blue and the remaining mask pixels are shown in red.

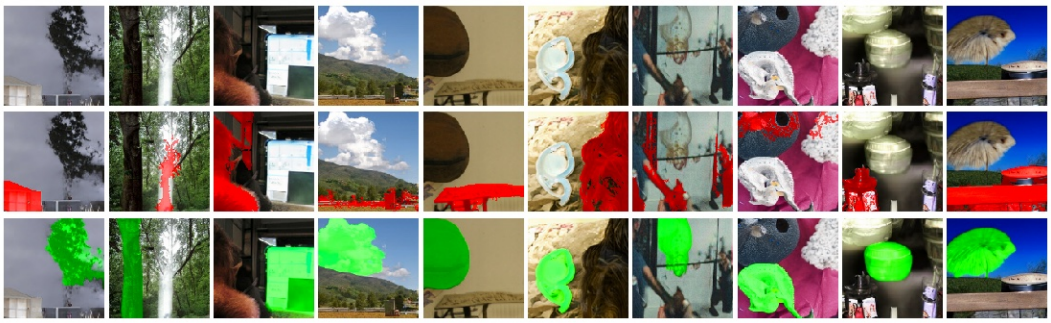


Figure 8: Failure cases of masks generation. *Top*: BigBiGAN samples; *Middle*: masks produced by latent manipulations; *Bottom*: masks produced by the supervised saliency model.

A.1 Ablation.

In Table 5 we demonstrate the impacts 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. The most significant performance impact comes from using the latent codes of the real images from the Imagenet.

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 5: Impact of different components in the E-BigBiGAN pipeline.

A.2 Synthetic Data Quality

In this section, we compare the generated synthetic saliency masks with the real ones. First, we address the question of the consistency of the generated data with the real one. We use the SOTA publicly available saliency model[5] to evaluate the quality of our synthetic saliency masks used for the best E-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 compared to the performance of our scheme on the real datasets. We attribute such behavior to the fact that our synthetic data is often noisy and it is more beneficial to train on difficult noisy data and to test on refined clean data rather than otherwise. Examples of masks with the strongest disagreement with the supervised model are provided in Figure 8. Typically, failures correspond to low-quality samples and the samples that do not contain well-defined salient objects (e.g. landscapes).

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

Our pipeline is generally based on the two steps: first, image generation, second, saliency mask synthesizing. To address the question of what is the bottleneck of the final model performance: the generated images quality, or the masks, we perform the following experiment. Using the same SOTA 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's generator. That is, we take the same images as in our best method and form the masks with a model pretrained on the real data. Then we train a U-net binary segmentation model guided by this data following the protocol described in Section 3.4. The comparison of the original model with the $\mathrm{U}^2\mathrm{Net}$-guided model is presented in Table 6. Notably, the $\mathrm{U}^2\mathrm{Net}$-guided model performs better, though does not demonstrate a break-through outperformance. This suggests that the bottleneck of the proposed method remains in the generated images quality instead of the mask generation approach.

Method	ECSSD			DUTS			DUT-OMRON		
	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy	maxF_β	IoU	Accuracy
E-BigBiGAN with $\mathrm{U}^2\mathrm{Net}$	**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 6: Comparison of masks generation with supervised $\mathrm{U}^2\mathrm{Net}$-guided synthetic labeling

B GANs LATENT SEGMENTATION

We have shown that on pair with BigGAN, the BigBiGAN has a latent direction responsible for image segmentation. This remains true for other state-of-the-art generative models. As an additional experiment, we explored the latent space of StyleGAN2 (Karras et al. (2020)) trained on LSUN-Church dataset (Yu et al. (2015)) and following Voynov & Babenko (2020) successfully reveal directions that have different effects on foreground / background pixels drastically increasing foreground red channel while keeping background colors closer to the original. The direction and examples of saliency masks for LSUN-Church are visualized in Figure 6. So, both StyleGAN2 and BigBiGAN can differentiate between object/background, being unsupervised. Note, however, that the BigBiGAN's domain is much broader, and its synthetic data can be used for a wider range of tasks.

B.1 MODELING A SEGMENTING DIRECTION

Thus different models reveal a latent direction responsible for segmentation, though all of them act in a different manner. Here we provide a principled framework to identify such segmenting directions automatically.

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 pixel $G(z)_{x,y}$ we have

$$G(z+h)_{x,y} = A_{i(x,y)}(G(z)_{x,y}),\ i(x,y) \in \{1,2\} \tag{3}$$

that is, the latent shift acts on each pixel as one of the two fixed maps. Intuitively, this definition formalizes that there are two different ways generated image pixels transform after the latent code shift. Notably, all the shifts presented on the Figure 9 satisfies (3) up to nonlinear tail.

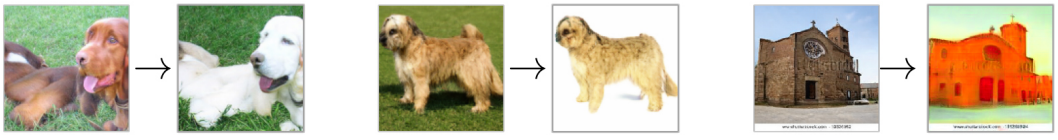


Figure 9: Segmentation directions shifts. *Left*: BigBiGAN; *center*: BigGAN (Voynov & Babenko, 2020); *right*: StyleGAN2

Now we address the problem to find these affine $A_1,\ A_2$ given a latent direction h of a generator G. In addition, we also show how to rank h with respect to the suitability for a segmentation mask generation task.

Given two affine operators A_1, A_2; for a pair of a pixel intensities c, c' let us define the map

$$S_{A_1,A_2}(c, c') = \underset{A \in \{A_1,A_2\}}{\arg\min} \left(\|A(c) - c'\|_2\right) \cdot c \tag{4}$$

that is S_{A_1,A_2} applies one of the operators that maps c closer to c'. We extend this action to the generated images space by setting $(S_{A_1,A_2} \cdot G(z))_{x,y} = S_{A_1,A_2}(G(z)_{x,y}, G(z+h)_{x,y})$ Let us define the restoration loss:

$$\mathcal{L}_h(A_1, A_2) = \mathbb{E} \sum_{x,y} \|S_{A_1,A_2} \cdot G(z)_{x,y} - G(z+h)_{x,y}\|_2 \tag{5}$$

here we sum over all pixels of an image $G(z)$. This quantity indicates how good can we approximate the map $\sigma_h : G(z) \to G(z+h)$ by applying A_1, A_2 pixelwise. Once this map can be represented in a form of equation 3, the quantity $\mathcal{L}_h$ possesses the global minimum equal to 0. Also if σ_h is not all the same per-pixel affine operator for all of the pixels, this minimum is unique.

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

$$\mathcal{L}_h(A_1, A_2) \to \min_{A_1,A_2} \tag{6}$$

These operators also define a binary segmentation of a generated image by assigning (x, y)-pixel class equal to the quantity

$$\underset{i \in \{1,2\}}{\arg\min} \|A_i \cdot G(z)_{x,y} - G(z+h)_{x,y}\|_2) \tag{7}$$

Thus a pixel class is defined by the operator that better mimics σ_h in that pixel.

B.2 Exploring segmentation directions

Given G, we wish to find a segmenting direction. We start with a set of interpretable latent directions $h_1, \ldots, h_K$. For each h_k we optimize (6) with the stochastic gradient descent. As the number of learnable parameters is small and equal to 24, we use mini-batch 4 and 200 steps of Adam optimizer with a learning rate 0.005. The optimization converges rapidly and we have not observed any benefits in a larger batch or greater number of steps. Overall this optimization takes a few minutes on the Nvidia-1080Ti card. For each h_k we receive a pair of affine operators $A_1^{(h_k)}, A_2^{(h_k)}$. The optimized restoration quality loss $\mathcal{L}_{h_k}$ reflects how good a particular transform σ_{h_k} can be approximated by two pixelwice operators.

In practice, this ranking is not sufficient to distill directions suitable for segmentation. A transform σ_k may induce almost identical $A_1^{(h_k)}$ and $A_2^{(h_k)}$ once it can be defined by a single operator, for instance, uniform global lighting. In that case, the masking based on these operators becomes to be noisy and uncorrelated with the saliency. To overcome this, let us define the mean distance $D_{h_k} = \|A_1^{(h_k)} \cdot G(z)_{x,y} - A_1^{(h_k)} \cdot G(z)_{x,y}\|_2$ between the pixels of generated images. This quantity measures the difference between $A_1^{(h_k)}$ and $A_1^{(h_k)}$. It appears that the segmenting directions always demonstrate high D_{h_k} value.

There is another pitfall coming from a latent h_k if it acts as an unique operator $G(z + h_k)_{x,y} = A \cdot G(z)_{x,y}$ with A the same for all pixels. Once so, the optimization process of $\mathcal{L}_{h_k}$ may focus on a single operator applied to all the pixels while ignoring the second. In that case the $\mathcal{L}_{h_k}$ quantity could be low while D_{h_k} high, producing a mask with all pixels assigned to the same class. We exclude these directions by the area filtration: we ignore all the shifts h_k such that the classes assigned by (7) are all the same for at least 99% of pixels.

In a nutshell, the segmenting direction search takes the form:

1 find interpretable directions $h_1, \ldots, h_K$;

2 for each direction compute $\mathcal{L}_{h_k}, D_{h_k}$ and the operators $A_1^{(h_k)}, A_2^{(h_k)}$;

3 discard directions that assign the same class for at least 99% of pixels;

4 take direction with the highest D_{h_k} and lowest $\mathcal{L}_{h_k}$

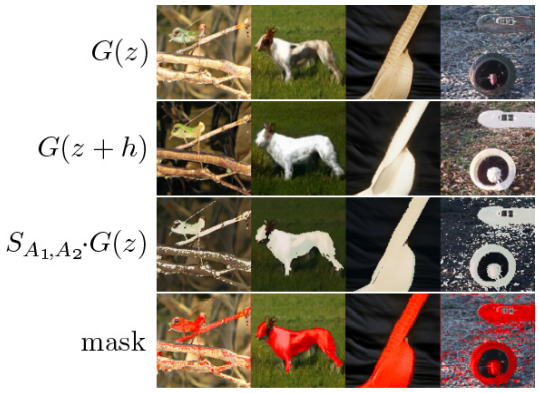


Figure 10: *Rows from top to bottom*: generated image; shifted image; shifted image approximation by pixelwice operators A_1, A_2; mask generated by these apperators assignement.

in practice, we perform the last step by: first, take top-20% with the highest D_{h_k} and, second, considering the top-10 with the lowest $\mathcal{L}_{h_k}$. Commonly all of these directions are applicable for synthetic saliency generation, though may induces different final performance.

Now we illustrate how this technics can be applied for the BigBiGAN segmenting direction search, though this remains valid for other models. After the rectification by Voynov & Babenko (2020) applied, we recieve 120 latent directions. Then we scale them by a multiplier 5 as the unit-length latent shifts commonly induce a minor image transformation leading to noisy $\mathcal{L}_h$ optimization process. In our experiments, the saliency lighting direction h_{bg} receives the highest $D_{h_{bg}}$ value with a relativelty small restoration loss $\mathcal{L}_{h_{bg}}$. On the Figure 10 we illustrate the images $S_{A_1,A_2} \cdot G(z)$ that approximate the shifted $G(z + h_{bg})$ by the pixelwice operators A_1, A_2 that minimize $\mathcal{L}_{h_{bg}}$. In assumption that the pixels are normalized in a range $[0, 1]$, the operators A_1, A_2 correspondent to the background saliency direction h_{bg} have the form:

$$A_1(c) = \begin{pmatrix} 0.13 & -0.12 & 0.06 \\ 0.01 & 0.00 & 0.04 \\ 0.02 & -0.20 & 0.22 \end{pmatrix} \cdot c + \begin{pmatrix} 0.78 \\ 0.76 \\ 0.69 \end{pmatrix}; \; A_2(c) = \begin{pmatrix} 0.31 & -0.05 & 0.05 \\ 0.04 & 0.19 & 0.06 \\ 0.01 & -0.06 & 0.31 \end{pmatrix} \cdot c - \begin{pmatrix} 0.1 \\ 0.15 \\ 0.19 \end{pmatrix}$$

That is 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, it commonly means that the intensity of the pixels handled by the first operator increases while the intensity of the pixels handled by the second decreases. Thus, we use this simple heuristic to generate the saliency masks in purpose to simplify the method.