Appendix: Learning Foreground-Background Segmentation from Improved Layered GANs

Yu Yang[1], Hakan Bilen[2], Qiran Zou[1], Wing Yin Cheung[1], Xiangyang Ji[1]
[1] Tsinghua University, BNRist [2] The University of Edinburgh
yang-yu16@mails.tsinghua.edu.cn, hbilen@ed.ac.uk,
{zouqr19,zhangyx20}@mails.tsinghua.edu.cn, xyji@tsinghua.edu.cn

A. Network Details

Latent codes The public code $\mathbf{z}$ is continuous and randomly drawn from a normal distribution $N(0,1)^{d_z}$. The private code $\mathbf{c}$ is designed as two-level codes, which is also known as parent and child code in FineGAN [14] and OneGAN [1]. In particular, $\mathbf{c}$ is discrete and randomly drawn from a categorical distribution $\text{Cat}(d_c)$, where d_c denotes the number of categories. These categories are grouped into super categories so that every $\mathbf{c}$ can be associated with a super private code $\mathbf{c}_{\text{sup}}$. We use a very simple grouping mechanism that group categories with consecutive indices and the group size is varied dependent on datasets as in FineGAN [14] and OneGAN [1].

Generator The structure of our generator (Fig. 1(a)) is adapted from FineGAN [14] which is tailored for generating images of fine-grained categories. Compared to the original structure, the GRU activation function is replaced with ReLU to save some computation without sacrificing too much performance. It is noteworthy that only $\mathbf{c}_{\text{sup}}$ and $\mathbf{z}$ influences the generation process of foreground masks, while $\mathbf{c}_{\text{sup}}$, $\mathbf{c}$, and $\mathbf{z}$ have impacts on the foreground appearance, *i.e.* RGB value. The weights of all the convolutional and linear layers are initialized with orthogonal matrix [13].

Perturbation The perturbation $\mathcal{T}$ operates on any given image or mask, which is decomposed into three consecutive elementary transformation, isotropic scaling, rotation, and translation. Formally,

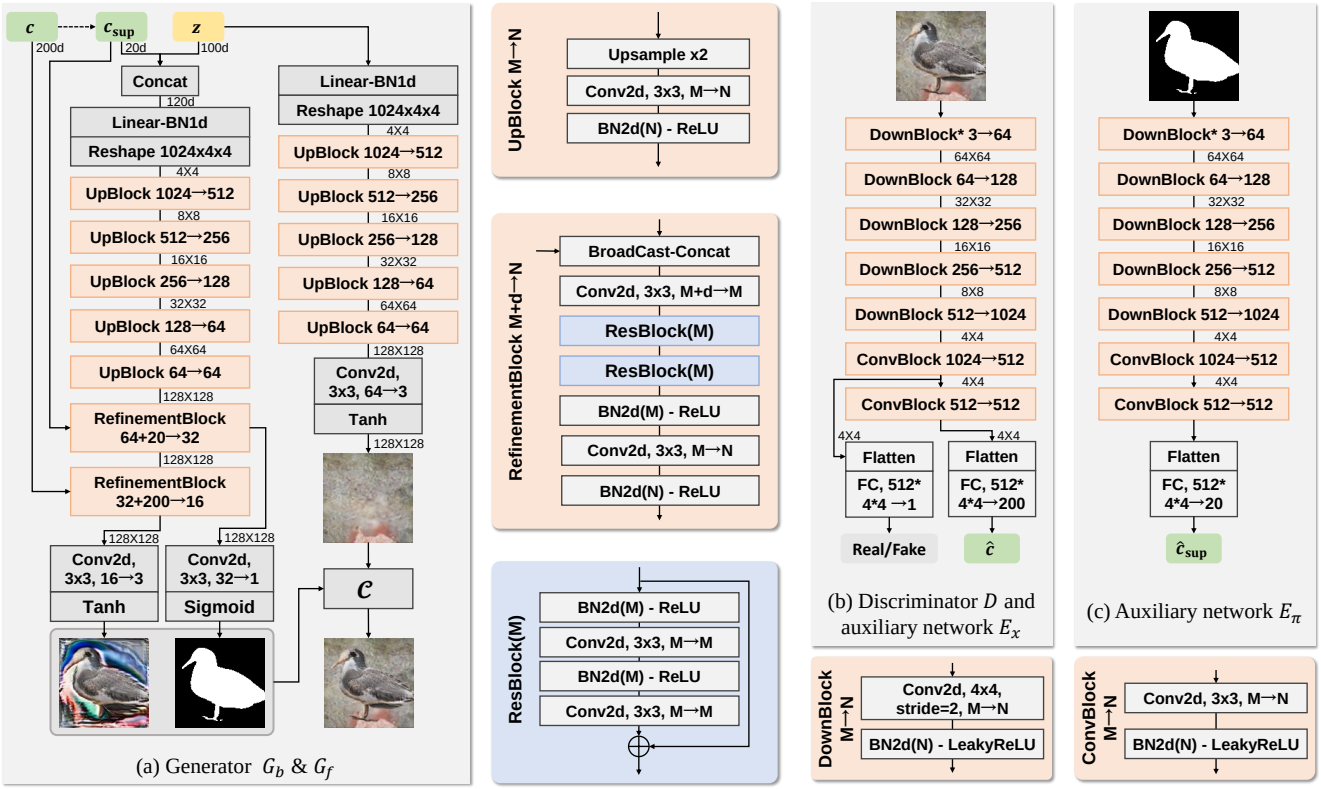
$$\mathcal{T}(\mathbf{x})(u,v,1) = \mathbf{x}(\boldsymbol{A}^{-1}(u,v,1)), \boldsymbol{A} = \boldsymbol{T}(t_x,t_y)\,\boldsymbol{R}(\alpha)\,\boldsymbol{S}(s), \tag{1}$$

$$\text{where } \boldsymbol{S}(s) = \begin{bmatrix} 2^s & 0 & 0 \\ 0 & 2^s & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{T}(t_x,t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{R}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

and the parameters of perturbation, s, t_x, t_y, and α, are uniformly drawn from a small range. Besides the above geometric perturbation, we also use background contrast jittering as in PerturbGAN [2].

Discriminator The structure of our discriminator is presented in Fig. 1(b). Spectral normalization [10] is employed to stabilize the training. The weights of all the convolutional and linear layers are initialized with orthogonal matrix [13].

Auxiliary networks for mutual information maximization Fig. 1(b)(c) presents the structures of our auxiliary networks for mutual information maximization. Following previous work [4, 9, 14, 1], E_x shares backbone with D and bifurcates at the top layers. E_x takes as input an image and output a categorical distribution which is considered as a approximation of the posterior distribution $p(\mathbf{c}|\mathbf{x})$. E_π has a similar role to E_x but it approximates $p(\mathbf{c}_{\text{sup}}|\boldsymbol{\pi})$ as $\mathbf{c}_{\text{sup}}$ controls the generation of masks.



(a) Generator G_b & G_f

(b) Discriminator D and auxiliary network E_x

(c) Auxiliary network E_π

Figure 1. **Network structures**. Discriminator D and the auxiliary network E_x shares the backbone parameters and bifurcates at the top layers. "DownBlock*" denotes a DownBlock without BN layer.

B. Dataset Details

Caltech-UCSD Birds 200-2011 (CUB) This dataset contains 11,788 bird images of 200 categories. It is split into 10,000 images for training, 788 images for validation, and 1,000 images for testing, as in [3]. All of the images only contain single instance and the interested objects are rarely occluded. The ground truth segmentation masks of the foreground object are annotated by humans and provided in the official release.

Stanford Dogs This dataset contains 20,580 dog images of 120 categories. It is split into 12,000 images for training, and 8,580 images for testing by official release [6]. Most images contain single instance and a few images contain occluded objects and irrelevant objects like human may appear in the image. The ground truth segmentation masks are approximated with a Mask R-CNN pretrained on MSCOCO[1].

Stanford Cars This dataset contains 16,185 car images of 196 categories. It is split into 8,144 images for training, and 8,041 images for testing by official release [8]. All if the images only contain single instance. The ground truth segmentation masks are approximated with a Mask R-CNN pretrained on MSCOCO, same model as used in Stanford Dogs.

Amazon Picking Challenge (APC) This dataset[2] was created for evaluating 6D poses of objects in the warehouse environment. The authors also released an object segmentation training set which is used to pretrain the segmentation networks. We use the "object segmentation training dataset"[3] for training our models. This dataset comprises images containing a single challenge object appearing either on a shelf or in a tray. Each object is shot as a series of scenes from different poses on both

[1]We use the detectron2 library (https://github.com/facebookresearch/detectron2) and the model R101-FPN (https://dl.fbaipublicfiles.com/detectron2/COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x/138205316/model_final_a3ec72.pkl)

[2]https://vision.princeton.edu/projects/2016/apc/

[3]http://3dvision.princeton.edu/projects/2016/apc/downloads/training.zip

the shelf and in the red tray. Following GENESIS-V2 [5], we randomly select 10% of scenes for validation, 10% of scenes for testing and the rest for training using the script provided in GENESIS-V2 official source code[4]. The raw images are resized and center-cropped, resulting in 128×128 images. The ground truth segmentation masks are provided in the official release. The ground truth masks are generated by certain automatic methods instead of annotated by humans. Therefore, these ground truth masks are quite noisy.

C. Training Details

C.1. Training Layered GANs

Data augmentation The real images augmented with horizontal flip and random resized crop. This augmentation is used to increase the variation in training set and shown helpful to support perturbed composition [2].

Optimization We employ Adam [7] optimizer with initial learning rate as 0.0002 and (β_1, β_2) as $(0.5, 0.999)$ train our layered GAN. The batch size is 32. The training process lasts until 1,000,000 images are seen by discriminator. Both the generator and the discriminator operate at 128×128 resolution.

Hyperparameters for each dataset Table 1 lists the specific hyperparameters used for each dataset.

	d_c	group size	scale s	shift t_x, t_y	rotation α	bg contrast	γ_{mi}	γ_{bg}
CUB	200	10	$[-0.2, 0.]$	$[-16\text{px}, 16\text{px}]$	$[-15°, 15°]$	0	1	2
Stanford Dogs	120	10	$[-0.2, 0.]$	$[-16\text{px}, 16\text{px}]$	$[-15°, 15°]$	0	2	1
Stanford Cars	196	14	$[-0.2, 0.]$	$[-16\text{px}, 16\text{px}]$	$0°$	$[0.7, 1.3]$	1	2
APC	200	10	$[-0.2, 0.]$	$[-16\text{px}, 16\text{px}]$	$[-15°, 15°]$	0	2	0

Table 1. **Hyperparameters** for training layered GANs on CUB, Stanford Dogs, Stanford Cars, and APC.

C.2. Alternate Training

The training of layered GANs and the training of segmentation network is alternated for 5 rounds. In each round, the layered GANs are trained until discriminator has seen 2,000,000 images and the segmentation network is trained for 2,000 steps. Segmentation network is not involved into regularizing layered GANs until the second round. We employ a U-Net [11] as segmentation network which is trained with Adam optimizer and the learning rate is 0.001 and the batch size is 32. Random color augmentation is used in training segmentation networks.

C.3. Training Segmentation Networks

A U-Net [11] segmentation network is trained from the synthetic dataset with Adam optimizer. The initial learning rate is 0.001 and batch size is 32. The training duration is 12,000 steps. Random color augmentation is used. The During inference, following [3, 12], the input image and ground truth masks are rescaled and center cropped to 128×128.

D. More Results

Stanford Cars					
	γ_{mi}				
	0.001	0.01	0.1	0.5	1.0
FID ↓	16.0	14.8	14.9	**14.6**	14.7
bg-FID ↑	145.0	134.4	162.9	168.2	**180.3**
IoU ↑	45.2	49.0	54.8	**59.7**	58.7

Stanford Dogs					
	γ_{mi}				
	0.1	0.5	1.0	5.0	10.0
FID ↓	47.3	36.8	**34.5**	46.5	52.1
bg-FID ↑	46.9	106.6	113.3	115.6	**117.5**
IoU ↑	36.8	58.0	62.1	**64.8**	57.1

Table 2. **Ablation study with respect to mutual information maximization** on Stanford Cars and Stanford Dogs at 64×64 resolution. γ_{mi} denotes the loss weight of mutual information maximization. The alternate training is disabled.

[4] https://github.com/applied-ai-lab/genesis

Ablation study w.r.t. mutual information maximization We further present the ablation study with respect to mutual information maximization on Stanford Cars and Stanford Dogs in Table 2 at resolution 64×64. It can be concluded that an appropriate γ_{mi} is essential to achieve optimal performance, while a wide range of γ_{mi} can assure the success of learning. Notably, the appropriate γ_{mi} might be different from dataset to dataset. For example, on Stanford Cars, even when the γ_{mi} is as low as 0.001, the segmentation is still learned though with low performance (IoU 45.2). However, on Stanford Dogs and CUB, a very low γ_{mi} (*e.g.* 0.1) would lead to the failure of disentangling foreground and background.

References

[1] Yaniv Benny and Lior Wolf. Onegan: Simultaneous unsupervised learning of conditional image generation, foreground segmentation, and fine-grained clustering. In *ECCV*, 2020.

[2] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. In *NeurIPS*, 2019.

[3] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *NeurIPS*, 2019.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

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

[6] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR Workshop*, 2011.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshop*, 2013.

[9] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *ICML*, 2020.

[10] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[12] Pedro Savarese, Sunnie SY Kim, Michael Maire, Greg Shakhnarovich, and David McAllester. Information-theoretic segmentation by inpainting error maximization. *arXiv preprint arXiv:2012.07287*, 2020.

[13] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[14] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019.