LABEL-EFFICIENT SEMANTIC SEGMENTATION WITH DIFFUSION MODELS

Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, Artem Babenko Yandex Research

Abstract

Denoising diffusion probabilistic models have recently received much research attention since they outperform alternative approaches, such as GANs, and currently provide state-of-the-art generative performance. The superior performance of diffusion models has made them an appealing tool in several applications, including inpainting, super-resolution, and semantic editing. In this paper, we demonstrate that diffusion models can also serve as an instrument for semantic segmentation, especially in the setup when labeled data is scarce. In particular, for several pretrained diffusion models, we investigate the intermediate activations from the networks that perform the Markov step of the reverse diffusion process. We show that these activations effectively capture the semantic information from an input image and appear to be excellent pixel-level representations for the segmentation problem. Based on these observations, we describe a simple segmentation method, which can work even if only a few training images are provided. Our approach significantly outperforms the existing alternatives on several datasets for the same amount of human supervision.

Keywords： diffusion model , semantic segmentation.

1 Introduction

Denoising diffusion probabilistic models (DDPM)[1][2] have recently outperformed alternative approaches to model the distribution of natural images both in the realism of individual samples and their diversity[3]. These advantages of DDPM are successfully exploited in applications, such as colorization[4] , inpainting[4], super-resolution[5][6](, and semantic editing (Meng et al., 2021), where DDPM often achieve more impressive results compared to GANs. So far, however, DDPM were not exploited as a source of effective image representations for discriminative computer vision problems. While the prior literature has demonstrated that various generative paradigms, such as GANs[7] or autoregressive models[8] , can be used to extract the representations for common vision tasks, it is not clear if DDPM can also serve as representation learners. In this paper, we provide an affirmative answer to this question in the context of semantic segmentation. In particular, we investigate the intermediate activations from the U-Net network that approximates the Markov step of the reverse diffusion process in DDPM. Intuitively, this network learns to denoise its input, and it is not clear why the intermediate activations should capture semantic information needed for high-level vision problems. Nevertheless, we show that on certain diffusion steps, these activations do capture such information, and therefore, can potentially be used as image representations for downstream tasks. Given these observations, we propose a simple semantic segmentation method, which exploits these representations and works successfully even if only a few labeled images are provided. On several datasets, we show that our DDPM-based segmentation method outperforms

the existing baselines for the same amount of supervision. To sum up, the contributions of our paper are: 1. We investigate the representations learned by the state-of-the-art DDPM and show that they capture high-level semantic information valuable for downstream vision tasks.

Diffusion models[1][2] are a class of generative models that approximate the distribution of real images by the endpoint of the Markov chain which originates from a simple parametric distribution, typically a standard Gaussian. Each Markov step is modeled by a deep neural network that effectively learns to invert the diffusion process with a known Gaussian kernel. Ho et al. highlighted the equivalence of diffusion models and score matching[9][8] , showing them to be two different perspectives on the gradual conversion of a simple known distribution into a target distribution via the iterative denoising process. Very recent works[10][3] have developed more powerful model architectures as well as different advanced objectives, which led to the "victory" of DDPM over GANs in terms of generative quality and diversity. DDPM have been widely used in several applications, including image colorization[4] , super-resolution[6], inpainting[11], and semantic editing[4]. In our work, we demonstrate that one can also successfully use them for semantic segmentation. Image segmentation with generative models is an active research direction at the moment, however, existing methods are primarily based on GANs. The first line of works[12][11] is based on the evidence that the latent spaces of the state-of-the-art GANs have directions corresponding to effects that influence the foreground/background pixels differently, which allows producing synthetic data to train segmentation models. However, these approaches are currently able to perform binary segmentation only, and it is not clear if they can be used in the general setup of semantic segmentation. The second line of works[13][14][15][16] is more relevant to our study since they are based on the intermediate representations obtained in GANs. In particular, the method proposed in[13] trains a pixel class prediction model on these representations and confirms their label efficiency. In the experimental section, we compare the method from[13]to our DDPM-based one and demonstrate several distinctive advantages of our solution. Representations from generative models for discriminative tasks. The usage of generative models, as representation learners, has been widely investigated for global prediction[7][8] , and dense prediction problems[13][14][15][16]. While previous works highlighted the practical advantages of these representations, such as out-of-distribution robustness[6] , generative models as representation learners receive less attention compared to alternative unsupervised methods, e.g., based on contrastive learning[8]. The main reason is probably the difficulty of training a high-quality generative model on a complex, diverse dataset. However, given the recent success of DDPM on Imagenet[17] , one can expect that this direction will attract more attention in the future. 2. We design a simple semantic segmentation approach that exploits these representations and outperforms the alternatives in the few-shot operating point. 3. We compare the DDPM-based representations with their GAN-based counterparts on the same datasets and demonstrate the advantages of the former in the context of semantic segmentation.

2 Related works

In this section, we briefly describe the existing lines of research relevant to our work.

2.1 Diffusion models

Diffusion models[1][2]are a class of generative models that approximate the distribution of real images by the endpoint of the Markov chain which originates from a simple parametric distribution, typically a standard Gaussian. Each Markov step is modeled by a deep neural network that effectively learns to invert the diffusion process with a known Gaussian kernel. Ho et al. highlighted the equivalence of diffusion models and score matching[9][18], showing them to be two different perspectives on the gradual conversion of a simple known distribution into a target distribution via the iterative denoising process. Very recent works[10] have developed more powerful model architectures as well as different advanced objectives, which led to the "victory" of DDPM over GANs in terms of generative quality and diversity. DDPM have been widely used in several applications, including image colorization[4] , super-resolution[5] , inpainting[4] , and semantic editing[4]. In our work, we demonstrate that one can also successfully use them for semantic segmentation.

2.2 Image segmentation with generative models

Image segmentation with generative models is an active research direction at the moment, however, existing methods are primarily based on GANs. The first line of works[12][11] is based on the evidence that the latent spaces of the state-of-the-art GANs have directions corresponding to effects that influence the foreground/background pixels differently, which allows producing synthetic data to train segmentation models. However, these approaches are currently able to perform binary segmentation only, and it is not clear if they can be used in the general setup of semantic segmentation. The second line of works[13][14][15][16] is more relevant to our study since they are based on the intermediate representations obtained in GANs. In particular, the method proposed in[13]trains a pixel class prediction model on these representations and confirms their label efficiency. In the experimental section, we compare the method from[13]to our DDPM-based one and demonstrate several distinctive advantages of our solution.

2.3 Representations from generative models for discriminative tasks.

Representations from generative models for discriminative tasks. The usage of generative models, as representation learners, has been widely investigated for global prediction[7][8] , and dense prediction problems[13][14][15][16]. While previous works highlighted the practical advantages of these representations, such as out-of-distribution robustness[6] , generative models as representation learners receive less attention compared to alternative unsupervised methods, e.g., based on contrastive learning[8] . The main reason is probably the difficulty of training a high-quality generative model on a complex, diverse dataset. However, given the recent success of DDPM on Imagenet[17], one can expect that this direction will attract more attention in the future.

3 Method

3.1 Background

Diffusion models transform noise $x_T \sim N(0, I)$ to the sample x_0 by gradually denoising x_T to less noisy samples x_t. Formally, we are given a forward diffusion process:

$$q\left(x_t \mid x_{t-1}\right) := \mathcal{N}\left(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t I\right), \tag{1}$$

for some fixed variance schedule $\beta_1, \ldots, \beta_t$. Importantly, a noisy sample x_t can be obtained directly from the data x_0 :

$$\begin{aligned} q\left(x_t \mid x_0\right) &:= \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t} x_0, \left(1-\bar{\alpha}_t\right) I\right), \\ x_t &= \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1), \end{aligned} \tag{2}$$

where $\alpha_t := 1 - \beta_t, \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. Pretrained DDPM approximates a reverse process:

$$p_\theta\left(x_{t-1} \mid x_t\right) := \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right). \tag{3}$$

In practice, rather than predicting the mean of the distribution in Equation (3), the noise predictor network $\varepsilon_\theta\left(x_t, t\right)$ predicts the noise component at the step t; the mean is then a linear combination of this noise component and x_t. The covariance predictor $\Sigma_\theta\left(x_t, t\right)$ can be either a fixed set of scalar covariances or learned as well (the latter was shown to improve the model quality[10]). The denoising model $\varepsilon_\theta\left(x_t, t\right)$ is typically parameterized by different variants of the UNet architecture[19], and in our experiments we investigate the state-of-the-art one proposed in[3] .

3.2 Extracting representations

For a given real image $x_0 \in \mathbb{R}^{H \times W \times 3}$, one can compute T sets of activation tensors from the noise predictor network $\varepsilon_\theta\left(x_t, t\right)$. The overall scheme for a timestep t is presented in Figure 1. First, we corrupt x_0 by adding Gaussian noise according to Equation (2). The noisy x_t is used as an input of $\varepsilon_\theta\left(x_t, t\right)$ parameterized by the UNet model. The UNet's intermediate activations are then upsampled to $H \times W$ with bilinear interpolation. This allows treating them as pixel-level representations of x_0.

3.3 The intermediate activations from the noise predictor capture semantic information.

For this experiment, we take a few images from the LSUN-Horse and FFHQ datasets and manually assign each pixel to one of the 21 and 34 semantic classes, respectively. Our goal is to understand whether the pixel-level representations produced by DDPM effectively capture the information about semantics. To this end, we train a multi-layer perceptron (MLP) to predict the pixel semantic label from its features produced by one of the 18 UNet decoder blocks on a specific diffusion step t. Note that we consider only the decoder activations because they also aggregate the encoder activations through the skip connections. MLPs are trained on 20 images and evaluated on 20 hold-out ones. The predictive performance is measured in terms of mean IoU.

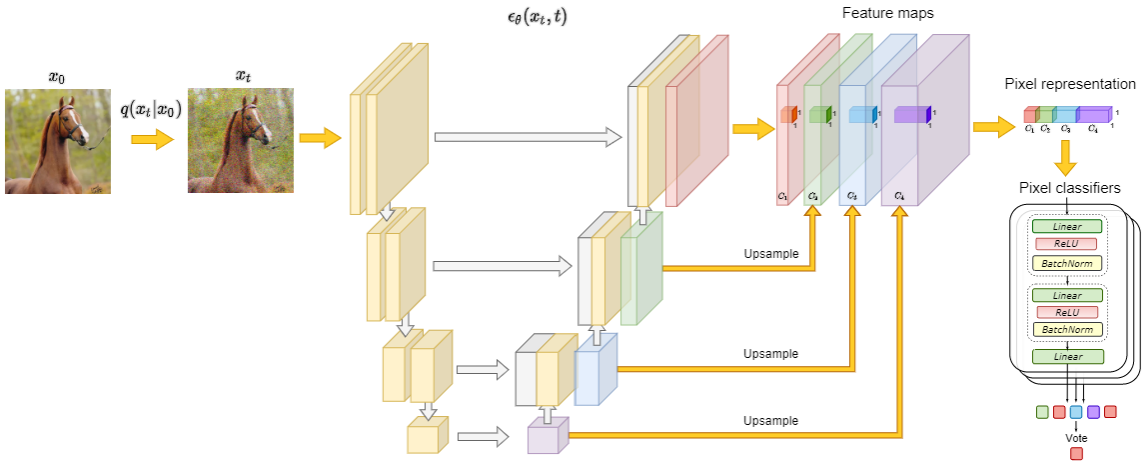

Figure 1: Overview of the proposed method. (1) $x_0 \rightarrow x_t$ by adding noise according to $q\left(x_t \mid x_0\right)$. (2) Extracting feature maps from a noise predictor $\varepsilon_\theta\left(x_t, t\right)$. (3) Collecting pixel-level representations by upsampling the feature maps to the image resolution and concatenating them. (4) Using the pixel-wise feature vectors to train an ensemble of MLPs to predict a class label for each pixel.

4 Reproduce the details

4.1 Main work

4.1.1 Training diffusion model

Here, we mainly do a data migration work. Part of the code of the diffusion model refers to the main code of openai's denoising-diffusion-pytorch, which refers to the denoising formula of the forward process and the denoising formula of the backward process, etc. Then, the training code of our main body, including the code of data processing, It's all self-written. The training part of the code is shown in the figure:

```
class PlModel(pl.LightningModule):
    def __init__(self):
        super(PlModel, self).__init__()
        self.automatic_optimization = False
        self.model = Unet(
            dim=64,
            dim_mults=(1, 2, 4, 8)
            # dim_mults=(1, 1, 2, 2, 4, 4)
        ).cpu()
        # self.model = DenoiseNet()
        self.diffusion = GaussianDiffusion(
            self.model,
            image_size=(256, 256),
            timesteps=1000,  # number of steps
            loss_type='l2'  # L1 or L2
        ).cpu()

    def training_step(self, batch, step):
        optimE = self.optimizers()

        loss = self.diffusion(batch)
        optimE.zero_grad()
        self.manual_backward(loss)
        optimE.step()
        self.log_dict({'loss': loss.detach().cpu().item()}, prog_bar=True)
        sch = self.lr_schedulers()
        sch.step()

        if self.global_step % 1000 == 0:
            self.diffusion.eval()
            sampled_images = self.diffusion.sample(batch_size=4)
            self.logger.experiment.log(
                {'train_{}_{}'.format(self.current_epoch, self.local_rank): [wandb.Image(i) for i in
                                                                             sampled_images.detach().cpu()[:8]]})
            self.diffusion.train()
```

图 2: The part of training code

In our training work, we first downloaded nearly 10,000 cervical cytology pictures supplemented with professional marks of pathologists from the 2022 ”Domain See Cup” Medical Examination Artificial Intelligence Developer Competition. In order to speed up model training, the pictures were randomly cropped into 256*256 format, and the batchsize of the pictures was set to 16. The time step of the diffusion model is set as 1000, that is, there are 1000 noise-adding steps and 1000 de-noising steps. The unet model is selected to predict the noise, and the picture of the noise-adding time at xt and t time are input into unet, where t is in the form of affine transformation rather than directly predicting the mean and variance of the previous moment. The predicted noise can be calculated by mathematical formula to obtain the image output of xt-1 at the previous moment. Here, standard Gaussian noise is used for constraint and l2 loss is selected. Here, our maximum number of iterations is 1000, and the training time of the whole model is about ten days. The main role of this diffusion model is to assist the classifier in label classification, and his can be used to generate cell pictures, that is, to generate a variety of pictures in line with the distribution of this data set. The training data is shown in the figure:

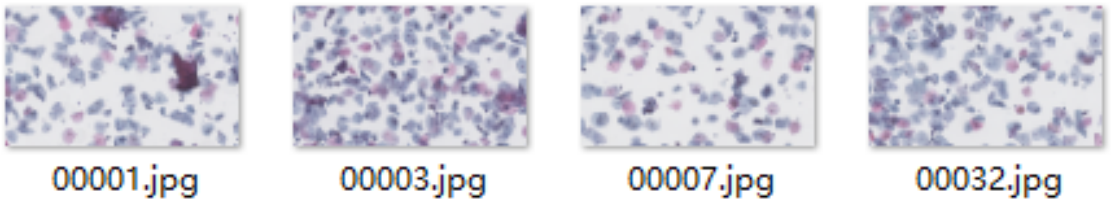

图 3: training data

4.1.2 Classifier for training

Here we refer to the official code of https://github.com/yandex-research/ddpm-segmentation, in order to be able to according to our data migration tasks, we also changed the feature extraction based on recurrence method. Part of the code looks like this:

```
        )

    def forward(self, x, time,save):
        x = self.init_conv(x)
        t = self.time_mlp(time)

        h = []

        for block1, block2, attn, downsample in self.downs:
            x = block1(x, t)
            x = block2(x, t)
            x = attn(x)
            h.append(x)
            x = downsample(x)

        x = self.mid_block1(x, t)
        x = self.mid_attn(x)
        x = self.mid_block2(x, t)

        for block1, block2, attn, upsample in self.ups:
            x = torch.cat((x, h.pop()), dim=1)
            x = block1(x, t)
            b=x.detach().clone()
            save.append(b)
            x = block2(x, t)
            c=x.detach().clone()
            save.append(c)
            x = attn(x)
            d=x.detach().clone()
            save.append(d)
            x = upsample(x)
            f=x.detach().clone()
            save.append(f)

        return self.final_conv(x),save
```

图 4: The part of training code

In this section, we refer to the classifier of the original text, that is, two LRB modules plus a Linear connection layer. The LRB module is composed of Linear,ReLU and BatchNorm. Since all down-sampling features are used here, we directly extract all up-sampling from the classifier module and combine output. Here we mainly trained ten models, each model trained 1000 times. The annotated data set is shown in the figure:

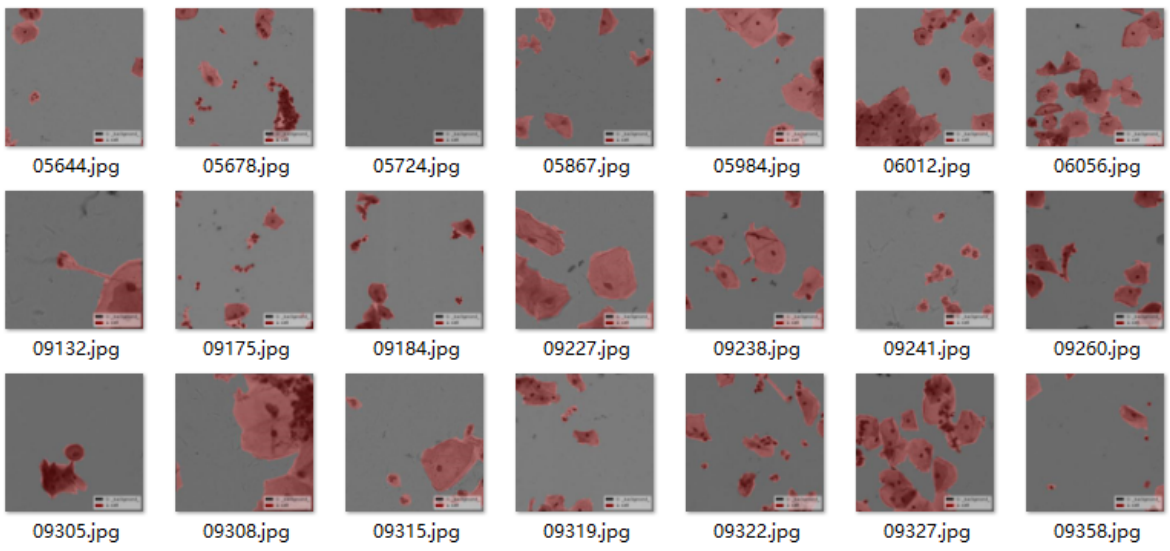

图 5: Annotated dataset

Firstly, 30 images were extracted from the cell data set for random cropping, and images of 256*256 size

were cropped out. Since the input dimension of this classifier was fixed, we could only fix the image size of the input diffusion model. Then, these 30 images were labeled by lableme. One is cell, one is background. Then, we first denoised the image. We selected the denoised image at t=50,100 and 150 moments, sent the denoised image to unet to extract all the features of the upper sampling layer, and finally conducted the supervised training of the classifier.

4.2 Experimental environment construction

The training mainly used the laboratory server, the development inheritor was pycharm, the language was python, and the deep learning framework was pytorch. Five Gpus were required for the training diffusion model, and only one gpu was required for the training classifier model.

5 Analysis of Experimental results

Here, 30 images with size of 256*256 were selected for our test set, and data annotation was carried out on the images. We put the test images into the model and extracted features, entered the classifier for pixel semantic classification, and calculated mIOU from the final output semantic segmentation map and the original annotated segmentation map. The figure below shows the calculated IOU results. It can be seen that the IOU results here are very good, and the task migration is very successful. Another reason is that our data quality is good, the cell map is clear, and the cell distribution is uniform.

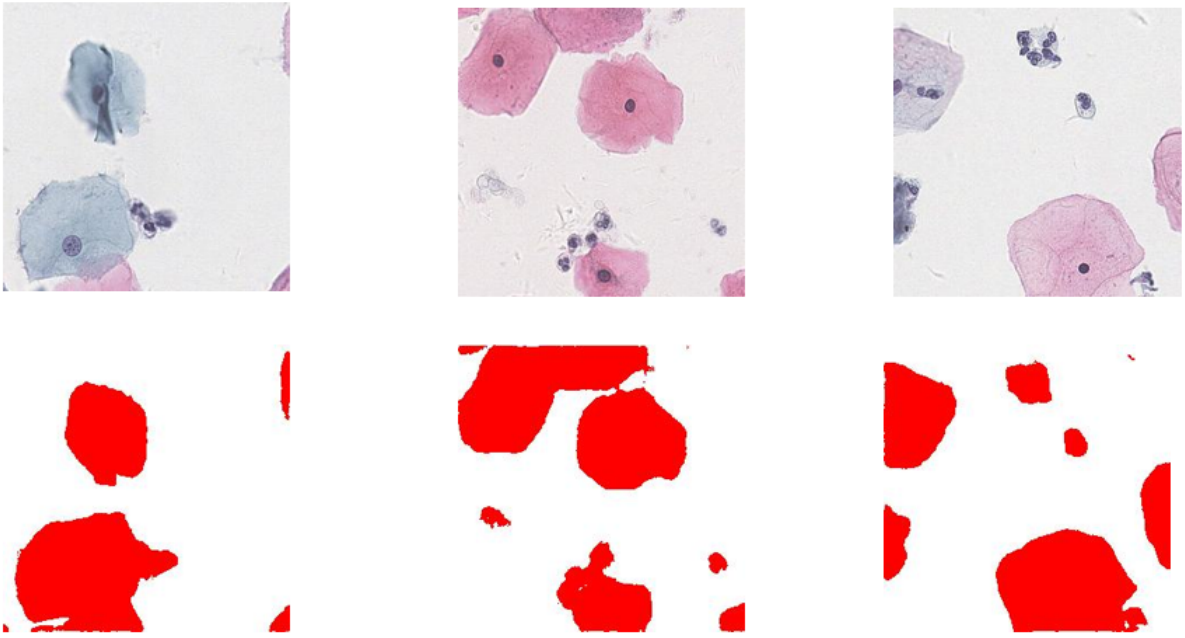

图 6: Experimental segmentation results

图 7: the experiment result of mIOU

6 Summary and Outlook

To sum up: This report mainly reintroduces the work of semantic segmentation by using diffusion model. The first three parts mainly introduce the content of the paper, and the fourth part is one of my main work. Here we carried out data transfer and transferred the segmentation model to nearly 10,000 cervical cancer cytology pictures, and introduced in detail how to train the diffusion model for the migration task. And how to train a classifier. Shortcomings: The test effect of our diffusion model here is not good on the data sets with many segmentation labels such as FFHQ and Bedroom-28. The missegmentation area of the resulting picture is likely to be at the edge of the image, and the segmentation area will show pixel clutter. Future research: Since the diffusion model is good in the field of multi-modes, we can use the segmentation diagram to guide sketch editing. Users can edit the segmentation diagram, and then perform conditional guidance on the segmentation diagram to edit the image.

References

[1] SOHL-DICKSTEIN J, WEISS E, MAHESWARANATHAN N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[C]//International Conference on Machine Learning. 2015: 2256-2265.

[2] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.

[3] DHARIWAL P, NICHOL A. Diffusion models beat gans on image synthesis[J]. Advances in Neural Information Processing Systems, 2021, 34: 8780-8794.

[4] MENG C, SONG Y, SONG J, et al. Sdedit: Image synthesis and editing with stochastic differential equations[J]. arXiv preprint arXiv:2108.01073, 2021.

[5] SAHARIA C, HO J, CHAN W, et al. Image super-resolution via iterative refinement[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[6] LI D, YANG J, KREIS K, et al. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8300-8311.

[7] DONAHUE J, SIMONYAN K. Large scale adversarial representation learning[J]. Advances in neural information processing systems, 2019, 32.

[8] CHEN M, RADFORD A, CHILD R, et al. Generative pretraining from pixels[C]//International conference on machine learning. 2020: 1691-1703.

[9] SONG Y, ERMON S. Generative modeling by estimating gradients of the data distribution[J]. Advances in Neural Information Processing Systems, 2019, 32.

[10] NICHOL A Q, DHARIWAL P. Improved denoising diffusion probabilistic models[C]//International Conference on Machine Learning. 2021: 8162-8171.

[11] MELAS-KYRIAZI L, RUPPRECHT C, LAINA I, et al. Finding an unsupervised image segmenter in each of your deep generative models[J]. arXiv preprint arXiv:2105.08127, 2021.

[12] VOYNOV A, BABENKO A. Unsupervised discovery of interpretable directions in the gan latent space [C]//International conference on machine learning. 2020: 9786-9796.

[13] ZHANG Y, LING H, GAO J, et al. Datasetgan: Efficient labeled data factory with minimal human effort [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 10145-10155.

[14] TRITRONG N, REWATBOWORNWONG P, SUWAJANAKORN S. Repurposing gans for one-shot semantic part segmentation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 4475-4485.

[15] XU Y, SHEN Y, ZHU J, et al. Generative hierarchical features from synthesizing images[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 4432-4442.

[16] GALEEV D, SOFIIUK K, RUKHOVICH D, et al. Learning high-resolution domain-specific representations with a gan generator[C]//Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). 2021: 108-118.

[17] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. 2009: 248-255.

[18] SONG Y, ERMON S. Improved techniques for training score-based generative models[J]. Advances in neural information processing systems, 2020, 33: 12438-12448.

[19] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. 2015: 234-241.