LEARNING TO ANNOTATE PART SEGMENTATION WITH GRADIENT MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a method to learn part segmentation for images generated from pre-trained generative models with a few labeled examples. In particular, an annotator is learned to decode generator features into part segmentation masks. In contrast to existing methods that train such an annotator in a supervised learning manner, we formulate the learning problem as a bi-level optimization problem from a teacher-and-student perspective: an annotator is learned to label generated images such that a student which trained from the synthesized data maintains minimum loss on the few labeled examples. We further derive a gradient matching learning objective and propose an iterative algorithm to approximately solve such a bi-level optimization problem. Our method applies to a semi-supervised learning setting and is evaluated on a variety of part segmentation datasets. Experiments demonstrate that our method significantly outperforms other semi-supervised segmentation methods, especially when the amount of labeled examples is limited. Our method achieves performance on par with the supervised learning counterparts yet addresses their shortcomings that labeled examples are required to be generated images. Code will be made public.

1 INTRODUCTION

Deep learning approaches to computer vision tasks usually demand tremendous number of annotated data. Nonetheless, in particular domain, only a few labelled examples are available due to limited budget or lack of expertised annotators. On the contrary, deep generative models, especially generative adversarial networks (GANs), continue to show its ability to synthesize photorealistic images and disentangle factors of variation, though they are only learned from a collection of images, without any annotations. TODO: emphasize more on disentangled representations in generative models. This motivates us to study *if annotations for synthetic images can be learned from a few labelled examples.*

We are not the first to consider this problem. Zhang et al. (2021) propose DatasetGAN where a well trained GAN is equipped with a label branch that decodes the generative features into annotations of target form, *e.g*. segmentation masks or keypoint heatmap. The same idea is also exploited in RepurposeGAN (Tritrong et al., 2021). This label branch can be efficiently trained with only a few labelled synthetic images yet maintain the ability to produce annotations for *any* synthetic images, leading to an infinite labelled data generator. The paradigm of DatasetGAN (Zhang et al., 2021) and RepurposeGAN (Tritrong et al., 2021), however, has one main disadvantage that the labelled examples are restricted to images synthesized by the exact generative models. This restriction impedes DatasetGAN (Zhang et al., 2021) from being applied to a wider scenario due to the following two concerns. First, synthetic images are often illegible in some details, making the annotating process difficult and error-prone. Second, annotating examples needs to be repeated if the generator is changed, leading to inconvenience in updating the generator.

In this paper, we propose to optimize the label branch via matching gradients induced by synthetic data and labelled examples. Gradient matching is proposed and utilized in the recent work on dataset condensation (DC) (Zhao et al., 2021; Zhao & Bilen, 2021) to condense a large set of training data into a synthetic set. Gradient matching pulls close the gradients of network parameters induced by these two datasets along an optimization trajectory and thereby results in a condensed dataset that is equivalent to source dataset in the effect of training a neural network. Inspired by this, we apply the principle of gradient matching in our problem and successfully learn the label branch. However,

our work maintains the following notable difference compared to DC. First, DC mainly focus on the image classification tasks while our work focus on the image segmentation tasks. Second, DC learns to *synthesize* a small dataset which could represent a large dataset, whereas our work learns to *annotate* infinite number of synthetic images from a few labelled examples.

Our methods address the aforementioned issues of DatasetGAN. In the paradigm of gradient matching, the restriction of labelled examples is relaxed to a broader range which varies from synthetic images and realistic in-domain images to even out-of-domain images. We demonstrate our methods in semi-supervised segmentation tasks on several datasets. Our methods achieve competitive performance compared to DatasetGAN yet apply to more scenarios like cross-domain semi-supervised learning. Furthermore, our methods outperform other semi-supervised segmentation methods with a large margin. Finally, due to the reliance on generative models, our methods also inherit their limitation to object-centric or simple scene datasets. Such limitation is hopefully addressed with the progress of generative modelling in the future.

2 Related Work

Semi-supervised learning Semi-supervised learning (SSL) augments the training of neural networks from a small labelled dataset with a large set of unlabelled data. A rich body of work regularize networks on unlabelled data with consistency regularization. In particular, networks are required to make consistent prediction over previous training iterations (Laine & Aila, 2017; Tarvainen & Valpola, 2017; Izmailov et al., 2018), noisy inputs (Miyato et al., 2018), or augmented inputs (Berthelot et al., 2019b;a; Sohn et al., 2020). This paradigm is also interpreted as a teacher-and-student framework where teacher produces pseudo labels for unlabelled data to supervise the training of students. Pseudo labels can be enhanced with heuristics (Lee et al., 2013; Berthelot et al., 2019a) or optimized towards better generalization (Pham et al., 2021; Li et al., 2019). Not limited to image classification tasks, the above ideas are also successfully applied to semantic segmentation tasks (Hung et al., 2018; Ouali et al., 2020; Ke et al., 2020; French et al., 2020). Our work can also be interpreted in the teacher-and-student framework. However, in contrast to the above methods that adopt homogeneous networks for teacher and student, our work employ generative models as teachers and discriminative models as students.

Other SSL approaches explore the utility of unlabelled data from representation learning perspective. In particular, the network is first optimized on the unlabelled data with self-supervised learning pretext tasks (Pathak et al., 2016; Noroozi & Favaro, 2016; Gidaris et al., 2018; He et al., 2020) to gain powerful and versatile representations, and then finetuned on the labelled data with target tasks (Chen et al., 2020; Zhai et al., 2019). In addition, generative models, which can be typically trained without labels, are shown to learn disentangled representations. Therefore, other works (Li et al., 2021; Zhang et al., 2021; Tritrong et al., 2021) decode generative features into dense annotations such as segmentation masks and keypoint heatmap to facilitate SSL. Our work is most related to DatasetGAN (Zhang et al., 2021) and RepurposeGAN (Tritrong et al., 2021) but has less restriction compared to them.

Disentangled representation learning Disentangled representation learning receives a lot of attention since it can benefit xxx A rich body of work show that generative models such as variational autoencoder (VAE) (Kingma & Welling, 2014) and generative adversarial networks (GANs) (Goodfellow et al., 2014) learns disentangled representations which is manifested by controllable generation process (Goetschalckx et al., 2019; Shen et al., 2020; Karras et al., 2019; Plumerault et al., 2020; Jahanian et al., 2020; Voynov & Babenko, 2020; Spingarn-Eliezer et al., 2021; Härkönen et al., 2020). Such advantages are further exploited to save manual efforts on annotation (Zhang et al., 2021; 2020; Tritrong et al., 2021). We follow the recent trend on harnessing the disentangled representation gained by GANs and show that manual labels can be reduced for segmentation tasks.

Part Semantic Segmentation Within various part-based object representations such as keypoint, pose and edge maps, part semantic segmentation provide dense form of feature for both rigid and non-rigid objects. It is of great significance in tremendous computer-vision tasks such as human parsing (Dong et al., 2014; Nie et al., 2018; Fang et al., 2018; Gong et al., 2017), pose estimation (Yang & Ramanan, 2011; Shotton et al., 2011; Xia et al., 2017) and 3D object understanding (Yi et al., 2016; Song et al., 2017). The challenge may arise from varying characteristics of object parts due to

viewpoint changes, occlusion and lack of clear boundaries for certain pairs of parts. Existing datasets manage to alleviate the problem to some extent. CelebAMask-HQ (Lee et al., 2020a) is a large-scale face image dataset annotated with masks of face components. Chen et al. (2014) provides additional part segmentation upon PASCAL VOC dataset (Everingham et al., 2010). Those datasets meet the demands of latest data-driven and deep neural network based method (Lee et al., 2020b; Wang et al., 2015; Wang & Yuille, 2015). However, a well-curated part segmentation dataset containing fine-grained pixel-wise annotation usually requires time-consuming manual labeling work, which holds back the development of this field. Latest work (Zhang et al., 2021; Li et al., 2021; Tritrong et al., 2021) leverage a generative model as an infinite data generator for semi-supervised learning and gain promising results. Nonetheless, the generated data is restricted to the pretrained domain of the generative model and result in limited generality and robustness. In this work, we propose a method utilizing features of pretrained generative model and adopt gradient matching approach to bridge the gap of datasets of different domains.

Gradient matching Modern optimization methods for deep neural networks are based on the gradients of target task loss with respect to network parameters backpropagated through network. Gradients contain rich information such that training data might be recovered from gradients (Zhao et al., 2020; Yin et al., 2021) or a large dataset can be condensed (Zhao et al., 2021; Zhao & Bilen, 2021). In our work, gradients are further explored for learning labels for synthetic images. TODO: Mention Tim's paper here.

3 METHOD

3.1 PROBLEM SETUP

Let G denotes a generative model trained from a dataset $\mathcal{D}_{tot}$ without labels. For example, G can be trained via adversarial learning (Goodfellow et al., 2014) such that it can generate photo-realistic images. The generation process is formulated as

$$\mathbf{x}, \mathbf{h} = G(\mathbf{z}), \tag{1}$$

where $\mathbf{z}$ denotes a random variable drawn from a prior distribution, $\mathbf{x}$ denotes the generated image, and $\mathbf{h}$ denotes the hidden features in G. Following DatasetGAN Zhang et al. (2021), an annotator A_ω is established to decode generative features $\mathbf{h}$ into annotations $\mathbf{y}$ of target form, *e.g.* segmentation masks,

$$\mathbf{y} = A_\omega(\mathbf{h}), \tag{2}$$

where ω parameterizes the annotator. Let $\mathcal{D}_l$ denotes a labelled dataset for target tasks. Our problem is to train annotator A_ω from $\mathcal{D}_l$ such that A_ω can annotate images synthesized from a fixed G.

DatasetGAN & RepurposeGAN Zhang et al. (2021) and Tritrong et al. (2021) train annotators in a direct supervised learning manner. First, a set of synthetic images generated from G are selected and annotated by humans. Generative features associated with the synthetic images are also stored. Such features and manual labels constitute the labelled dataset $\mathcal{D}_l = \{(\mathbf{x}_i, \mathbf{h}_i, \hat{\mathbf{y}}_i)\}_{i=1}^N$. Second, annotators are trained by minimizing the loss computed with predicted annotations and ground truth annotations,

$$\omega = \arg\min_\omega \mathbb{E}_{(\cdot,\mathbf{h},\hat{\mathbf{y}})\in\mathcal{D}_l} \ell(A_\omega(\mathbf{h}), \hat{\mathbf{y}}), \tag{3}$$

where ℓ denotes the loss function for specific tasks, *e.g.* per-pixel cross-entropy function for segmentation tasks. This method, however, requires $\mathcal{D}_l$ to be a *synthetic* and *generator-specific* set, which has following two defects. (i) Details might be illegible in synthetic images, which increases the difficulty and error rate of annotation. (ii) Extra efforts need to be paid to annotate images if the generator is changed.

Teacher and student Our problem can be alternatively formulated from teacher and student perspective as in Li et al. (2019); Pham et al. (2021). Concretely, G generates infinite number of synthetic images and A_ω serves as a teacher producing the labels. We further introduce a student network $S_\theta : \mathbf{x} \mapsto \mathbf{y} = S_\theta(\mathbf{x})$ parameterized with θ learning from the pseudo-labelled synthetic dataset to perform the target task. The optimization problem of student network is formulated as

$$\theta = \arg\min_\theta \mathbb{E}_{\mathbf{h},\mathbf{x}} \ell(S_\theta(\mathbf{x}), A_\omega(\mathbf{h})). \tag{4}$$

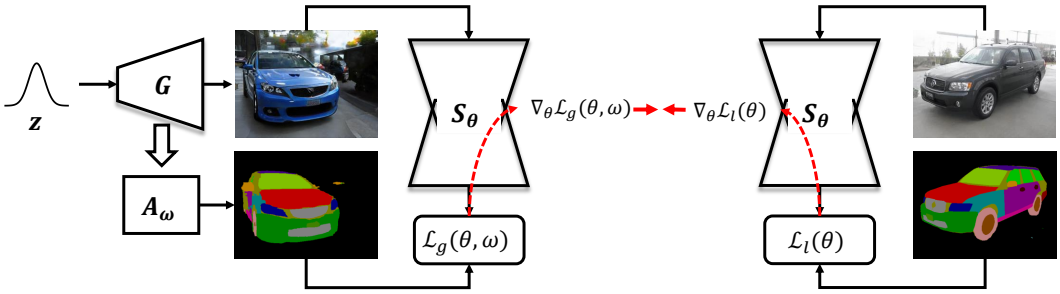

Figure 1: **Illustration** of our methods. TODO: Figure needs to be beautified.

It is notable that, as G is fixed, θ in Equation 4 only depends on the ω, denoted as $\theta(\omega)$, and therefore the objective is denoted as $\mathcal{L}_g(\theta, \omega)$. The aim of learning annotators is to teach a student network to gain high performance on the labelled dataset $\mathcal{D}_l$, which can be formulated as the following bi-level optimization problem

$$\begin{aligned} \min_\omega \quad & \mathcal{L}_l(\theta(\omega)) = \mathbb{E}_{(\mathbf{x},\hat{\mathbf{y}})\in\mathcal{D}_l} \ell(S_\theta(\mathbf{x}), \hat{\mathbf{y}}), \\ \text{where} \quad & \theta(\omega) = \arg\min_\theta \mathcal{L}_g(\theta, \omega). \end{aligned} \tag{5}$$

Compared to DatasetGAN, this optimization problem does not require $\mathcal{D}_l$ to be a generated or generator-specific dataset, but is more cumbersome to be optimized. Unlike Li et al. (2019); Pham et al. (2021) that appeal to meta-learning approaches, we circumvent the bi-level optimization by deriving a learning objective in the form of gradient matching as described in the next section.

3.2 Training Annotators with Gradient Matching

TODO: Cite Tim's paper. We train annotator with gradient matching (Fig. 1), which can overcome the aforementioned limitations of DatasetGAN (Zhang et al., 2021) and circumvent the bi-level optimization in Equation 5. First, the optimization of student network in Equation 4 is approximated with one-step gradient descent,

$$\theta_{t+1}(\omega) = \theta_t - \eta \nabla_\theta \mathcal{L}_g(\theta_t, \omega), \tag{6}$$

where t denotes the timestep of iteration and η denotes the learning rate. By plugging this equation into $\mathcal{L}_l(\theta(\omega))$ and unfolding it with Taylor expansion and omitting the higher order term, we have

$$\begin{aligned} \mathcal{L}_l(\theta_{t+1}(\omega)) &= \mathcal{L}_l(\theta_t - \eta \nabla_\theta \mathcal{L}_g(\theta_t, \omega)) \\ &= \mathcal{L}_l(\theta_t) - \eta \nabla_\theta^\top \mathcal{L}_l(\theta_t) \nabla_\theta \mathcal{L}_g(\theta_t, \omega) + \dots \\ &\approx \mathcal{L}_l(\theta_t) - \eta \nabla_\theta^\top \mathcal{L}_l(\theta_t) \nabla_\theta \mathcal{L}_g(\theta_t, \omega). \end{aligned} \tag{7}$$

In this way, the bi-level optimization problem, Equation 5, turns into

$$\max_\omega \mathcal{L}_{gm}(\theta_t, \omega) = \nabla_\theta^\top \mathcal{L}_l(\theta_t) \nabla_\theta \mathcal{L}_g(\theta_t, \omega). \tag{8}$$

To this end, the learning objective to optimize annotator is established, which is in the form of gradient matching. Since dot product can be considered as measuring similarity between two vectors, Equation 8 can be understood as driving $\nabla_\theta \mathcal{L}_g(\theta_t, \omega)$, the gradients induced by teacher, close to $\nabla_\theta \mathcal{L}_l(\theta_t)$, the gradients induced by labelled data.

TODO: Discuss variants of learning objectives including inner product and cosine similarity.

Algorithm It is noteworthy that the above derivation is based on *one-step* gradient descent approximation and the objective $\mathcal{L}_{gm}(\theta_t, \omega)$ relates to the student parameter θ_t. TODO: It is unclear why we should keep updating student network. Hence, to train annotator towards meeting the aim of Equation 5, an optimization trajectory $\{\theta_t\}_{t=0}^T$ should be considered. In other words, the optimal ω should be independent to θ and the annotator should produce annotations which can be universally employed to teach any student network. On the other hand, the optimization of student network (Equation 4) depends on the annotator. To this end, we propose to alternately train annotator and student as presented in Algorithm 1.

Algorithm 1: Train annotators with gradient matching.

Inputs :

G	trained generator
$\mathcal{D}_l$	set of labeled examples
ω, A_ω	initial annotator parameters, annotator
θ, S_θ	initial segmentor parameters, segmentor
T and B	total number of optimization steps and batch size

for $t = 1$ **to** T **do**

// update teacher

$\mathcal{B}_l \leftarrow \{(\mathbf{x}_i, \hat{\mathbf{y}}_i) \sim \mathcal{D}_l\}_{i=1}^{B}$ // sample a batch of labeled examples

$\nabla_\theta \mathcal{L}_l(\theta) \leftarrow \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta \ell(S_\theta(\mathbf{x}_i), \hat{\mathbf{y}}_i)$ // compute gradients w.r.t. θ

$\mathcal{B}_g \leftarrow \{(\mathbf{x}_i, A_\omega(\mathbf{h}_i))\}_{i=1}^{B}$ // sample a batch of synthetic data

$\nabla_\theta \mathcal{L}_g(\theta, \omega) \leftarrow \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta \ell(S_\theta(\mathbf{x}_i), A_\omega(\mathbf{h}_i))$ // compute gradients w.r.t. θ

$\mathcal{L}_{gm}(\theta, \omega) \leftarrow \nabla_\theta^\top \mathcal{L}_l(\theta) \nabla_\theta \mathcal{L}_g(\theta, \omega)$ // compute gradient matching loss

$\omega \leftarrow \omega - \eta_\omega \nabla_\omega \mathcal{L}_{gm}(\theta, \omega)$ // update annotator parameters

// update student

$\mathcal{B}_g \leftarrow \{(\mathbf{x}_i, A_\omega(\mathbf{h}_i))\}_{i=1}^{B}$ // sample a minibatch of synthetic data

$\mathcal{L}_g(\theta, \omega) \leftarrow \frac{1}{B} \sum_{i=1}^{B} \nabla_\theta \ell(S_\theta(\mathbf{x}_i), A_\omega(\mathbf{h}_i))$ // compute loss on synthetic data

$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_g(\theta, \omega)$ // update segmentor parameters

end

Output : annotator A_ω and segmentor S_θ

3.3 Implementation Details

Network structures We mainly use StyleGAN2 (Karras et al., 2020b) as our generator since this method shows state-of-the-art synthesis quality on single-object-class datasets and disentangled generative representations. For experiments on CelebA, Pascal-Horse and Pascal-Aeroplane, we train StyleGAN2-ADA (Karras et al., 2020a) on the training set at 256×256 resolution. For experiments on Car-20, Face-34, and Cat-16, we employ publicly available pre-trained StyleGAN (Karras et al., 2019) and StyleGAN2 (Karras et al., 2020b;a) models. More details are available in the appendix.

Our annotator is a neural network that takes as input the generator features and output segmentation masks. In contrast to DatasetGAN (Zhang et al., 2021) and RepurposeGAN (Tritrong et al., 2021) where feature maps are upsampled to full resolution and concatenated and fed into either multiple layer perceptrons (MLPs) or a convolutional neural networks (CNNs), we fuse the multi-scale feature maps with a feature pyramid network (FPN) (Lin et al., 2017) structure and decode the fused features into segmentation masks with consecutive convolutional layers. This FPN structure save a lot of GPU footprints and further allow us to sample multiple full images to form one mini-batch. Details about the architecture are available in the appendix.

Training Batch size, optimizer, training duration, hyperparameters, loss function etc.

4 Experiments

4.1 Setup

Datasets We benchmark our methods on several part segmentation datasets, **CelebA**, **Pascal-Horse**, and **Pascal-Aeroplane**. CelebA (Liu et al., 2015) is a large scale human face dataset from which 30,000 images are annotated with up to 19 part classes, denoted as CelebAMask-HQ (Lee et al., 2020a). Following Li et al. (2021), we only consider a subset of 8 face classes and use the first

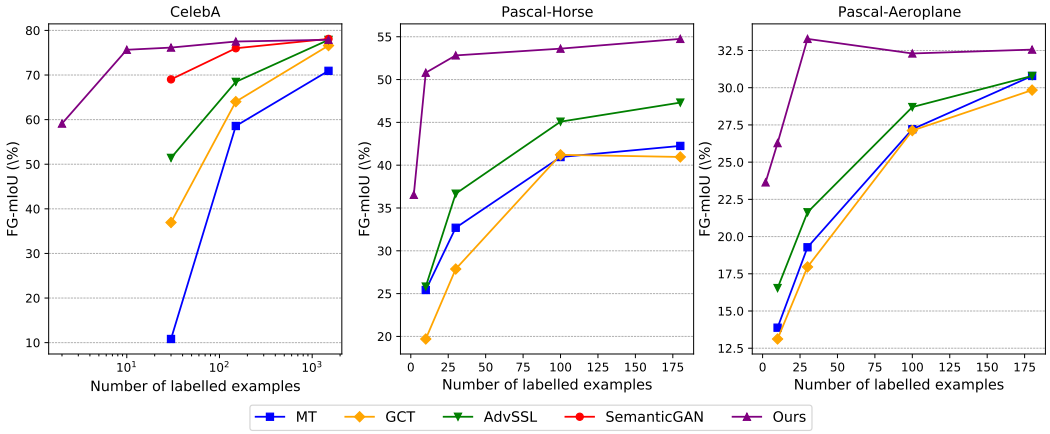

Figure 2: **Benchmark** on CelebA (*Left*), Pascal-Horse (*Middle*), and Pascal-Aeroplane (*Right*). Our method is compared to mean teacher (MT) (Tarvainen & Valpola, 2017; Li et al., 2020), Guided Collaborative Training (GCT) (Ke et al., 2020), Adversarial Semi-supervised Learning (AdvSSL) (Hung et al., 2018), and SemanticGAN (Li et al., 2021) when varied number of labelled examples are employed.

28,000 images in CelebAMask-HQ as unlabeled set, the last 500 images as test set, and the rest 1500 images as labeled set. We construct Pascal-Horse and Pascal-Aeroplane by taking horse images and aeroplane images from Pascal Part dataset Chen et al. (2014) which provides images of 20 object categories annotated with parts. The selected images are further cropped according to bounding box annotations. Full process details are available in the appendix. We finally have 180 images as labeled set, 34 images as validation set, and 225 images as test set in Pascal-Horse, and 180 images as labeled set, 78 images as validation set and 269 images as test set in Pascal-Aeroplane. In Pascal-Horse and Pascal-Aeroplane, additional images of horse and airplane from LSUN (Yu et al., 2015) are taken as unlabeled set. Furthermore, we conduct comprehensive experiments and make thorough comparison to DatasetGAN (Zhang et al., 2021) on **Car-20**, **Cat-16**, and **Face-34** , which are originally released by Zhang et al. (2021) and used to evaluate DatasetGAN. Car-20, Cat-16, and Face-34 are manually annotated 20, 16, and 34 part classes for car, cat, and human face, respectively. Car-20 contains 16 images as training set, 10 images as test set; Cat-16 contains 30 images as training set, 20 images as test set; Face-34 contains 16 images as training set, 20 images as test set.

Evaluation We use mean intersection over union (mIoU) to evaluate the performance of our segmentation network. In CelebA, Pascal-Horse, and Pascal-Aeroplane, we use validation set to select checkpoints and report mIoU across all foreground classes, denoted as "FG-mIoU", on test set. In Car-20, Cat-16 and Face-34, we follow the setting of DatasetGAN (Zhang et al., 2021) and report the cross validation mIoU across all classes including background on test set.

4.2 Semi-Supervised Part Segmentation

Benchmark results As our methods have flexible restrictions on the source of labeled examples, we benchmark our methods on datasets consisting of realistic images with different number of labeled examples. Following SemanticGAN (Li et al., 2021), our methods are compared against SSL methods that have codes available [1], including Mean Teacher (MT) (Tarvainen & Valpola, 2017; Li et al., 2020), Guided Collaborative Training (GCT) (Ke et al., 2020), an adversarial-learning-based semi-supervised segmentation method (AdvSSL) (Hung et al., 2018), as well as SemanticGAN (Li et al., 2021). We benchmark under different number of labeled examples and present the results in figure 2. Our methods are shown to significantly outperform other methods. Notably, when the number of labeled examples reduces, the performance of our methods drop slightly while performance of other methods decreases drastically. We attribute the reason to our utilization of generator features which are highly disentangled. SemanticGAN (Li et al., 2021) also exploits the generative features

[1] We adopt the implementation in `https://github.com/ZHKKKe/PixelSSL`.

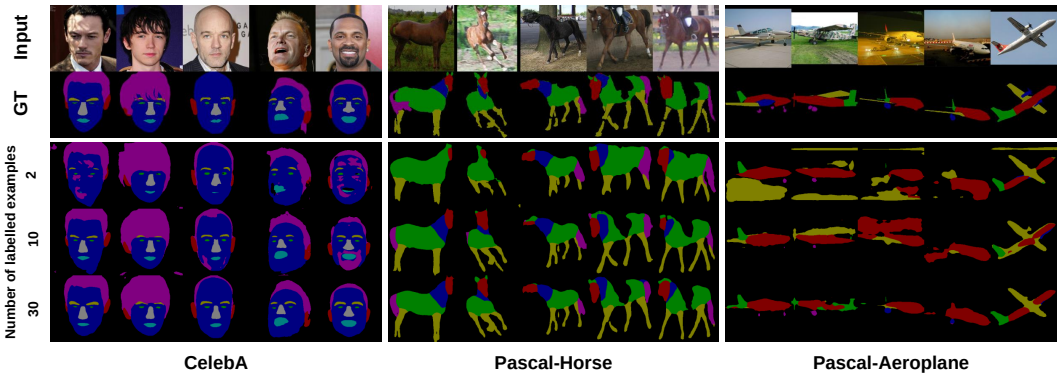

Figure 3: **Qualitative results** of our method on CelebA, Pascal-Horse, and Pascal-Aeroplane. The segmentation results of our models trained from 2, 10, and 30 labeled examples are presented. TODO: Results of aeroplane are not good.

Methods	G	A	S	Car-20	Face-34	Cat-16
TL (Zhang et al., 2021)	-	-	DeepLab	33.91 ± 0.57	45.77 ± 1.51	21.58 ± 0.61
SSL (Zhang et al., 2021)	-	-	DeepLab	44.51 ± 0.94	48.17 ± 0.66	24.85 ± 0.35
DatasetGAN	StyleGAN	MLP	DeepLab	67.53 ± 2.58	54.55 ± 0.25	32.63 ± 0.68
	StyleGAN	MLP	U-Net	66.27 ± 2.75	53.84 ± 0.41	31.36 ± 0.76
	StyleGAN	FPN	DeepLab	59.91 ± 2.36	47.08 ± 0.41	32.61 ± 0.31
	StyleGAN	FPN	U-Net	56.91 ± 3.42	44.52 ± 0.22	30.86 ± 0.36
Ours	StyleGAN	FPN	DeepLab	60.55 ± 2.78	53.03 ± 0.45	32.31 ± 0.37
	StyleGAN	FPN	U-Net	61.83 ± 2.53	53.96 ± 0.21	31.76 ± 0.50
	StyleGAN2	FPN	DeepLab	60.40 ± 1.27	54.27 ± 0.40	34.30 ± 0.29
	StyleGAN2	FPN	U-Net	61.14 ± 2.55	53.31 ± 0.14	31.99 ± 0.58

Our retrained segmentation networks (G: StyleGAN2; A: FPN)

Source S	Downstream S	Car-20	Face-34	Cat-16
U-Net	U-Net	59.33 ± 3.15	54.22 ± 0.15	32.23 ± 0.41
U-Net	DeepLab	61.39 ± 2.78	54.31 ± 0.20	33.55 ± 0.39
DeepLab	U-Net	60.40 ± 2.32	54.95 ± 0.45	32.86 ± 0.75
DeepLab	DeepLab	61.26 ± 1.25	54.63 ± 0.37	32.98 ± 0.63

Table 1: **Comparisons to DatasetGAN** on Car-20, Cat-16, Face-34 with different network architectures. TL represents transfer learning; SSL represents semi-supervised learning. The results of TL and SSL are originally reported by Zhang et al. (2021). The performance is evaluated with mean intersection over union (mIoU(%)) across all part classes plus a background class.

to produce segmentation masks but its performance still degrades faster than our methods as the number of labeled examples decreases. We believe it is due to the adversarial learning employed by SemanticGAN which requires more labels than our methods to achieve fair performance. Qualitative results are shown in figure 3 and more results are available in the appendix.

Comparison to DatasetGAN Since DatasetGAN is very related to our methods, we conduct comprehensive experiments to make comparison. To ablate the complication of network structures, DatasetGAN and our methods are evaluated on Car-20, Cat-16, and Face-34 with different network structures. Quantitative and qualitative results are presented in Table 1 and figure 4. First, we test the effect of annotator architecture. It is noticed that the segmentation performance of DatasetGAN drops as the MLP annotator is replaced with FPN annotator. This suggests that although FPN structure saves some GPU footprints, it compromises the segmentation performance. Second, our methods outperforms DatasetGAN when same network architecture is used: StyleGAN as generator, FPN as annotator and DeepLab/U-Net as segmentation networks. Third, since our method does not require that generator should match the labeled examples, we further employ the up-to-date version of

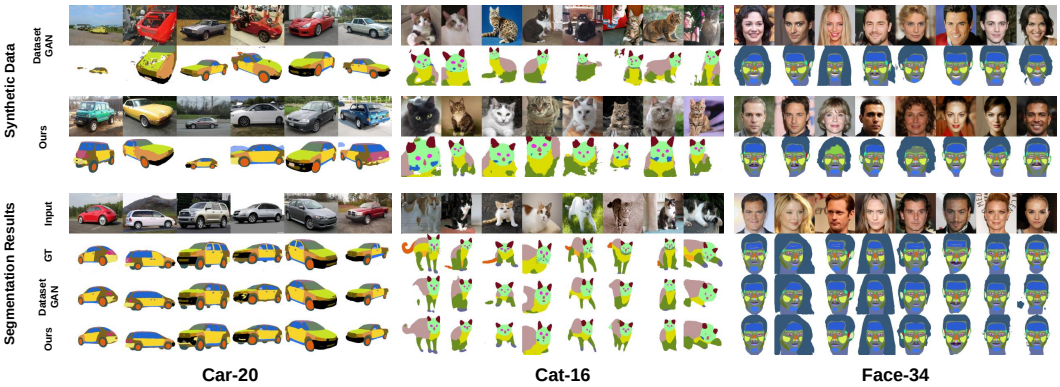

Figure 4: **Qualitative comparison** of our method to DatasetGAN on Car-20, Cat-16, and Face-34.

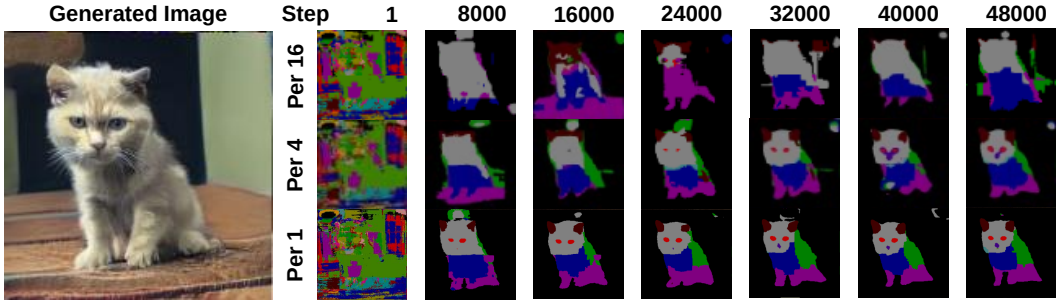

Figure 5: The evolution of learned annotations when student network is updated every 1/4/16 steps. TODO: Update the visualization.

StyleGAN2 and it can be observed that the performance is increased when more powerful pre-trained generator is used. Finally, our methods could adapt to different segmentation architecture, and generally, DeepLab would gives slightly higher performance than U-Net.

We further use our trained annotator to generate a synthetic segmentation dataset from which downstream segmentation networks could be trained. Table 1 shows the results when different source networks, which are used for training annotator with gradient matching, and different downstream networks are used. First, our learned synthetic datasets can be used to train different structure of downstream segmentation networks: DeepLab and U-Net. Results show that downstream segmentation networks can achieve good performance, which indicates that our learned synthetic dataset is not architecture-specific. Second, DeepLab as source network generally produce higher performance than U-Net as source network. This suggests that in our method the student network actually influences the training of teacher (*i.e.* annotator).

4.3 Ablation Study

The frequency of updating student networks Our methods train teacher network and student network alternately. We change the frequency of updating student network for every 1/4/16 steps and the evolution of a learned annotation example is presented in figure 5. It shows that annotator learns faster when student network is updated more frequently. Therefore, our algorithm update annotator and student equal frequently by default.

The parameters to match gradients As computing gradient matching objective requires back-propagation through student networks, this procedure may consume a lot of computer resources as nowadays neural networks tend to be deep and big. We study if this drawback can be alleviated by only matching gradients of partial network parameters. In particular, we conduct experiments on Cat-16 using StyleGAN2 as generator, FPN as annotator, and DeepLab as segmentation network. We ablate the parameters for gradient matching from bottom blocks to top blocks. The results are

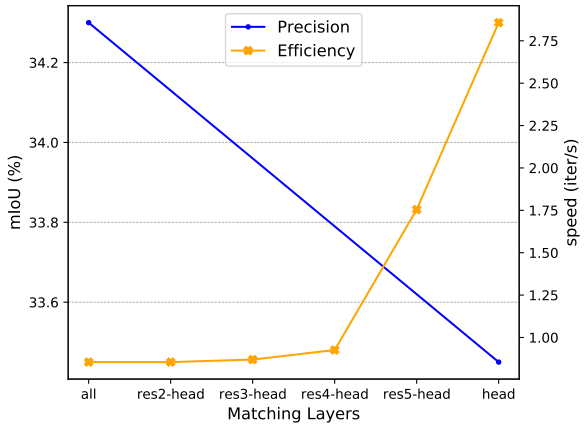

Figure 6: The performance (mIoU (%)) and training speed (steps/s) of our methods on Cat-16 with respect to blocks for matching gradients. The training speed is calculated on a single RTX 2080 Ti GPU.

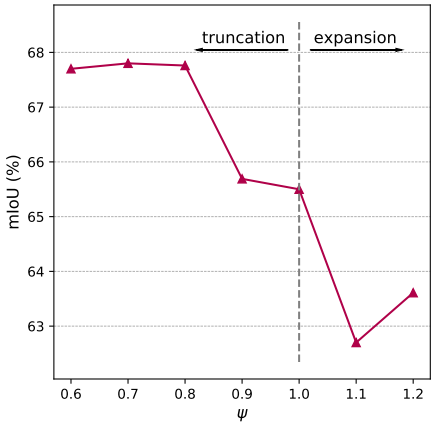

Figure 7: The performance (mIoU (%)) of our methods on Car-20 with respect to the truncation coefficient ψ. TODO: Add some

shown in figure 6, where `Resx` denotes the x-th block in the backbone network and `Head` denotes segmentation head block. It can be seen that by only matching gradients of partial parameters, the training speed can be improved but the segmentation performance is compromised. It is notable that even only the gradients of `Head` are matched, the segmentation performance only drop slightly. Therefore, to trade off the training speed and performance, we only match the gradients of `Head` by default.

The impact of generator Truncation tricks, which shrink the prior distribution support set of latent variables in generation process, are widely used to tune the quality of generated samples or trade off sample diversity and fidelity. We study if this trick can affect our methods. figure 7 shows the performance variation w.r.t. the truncation coefficient, denoted as ψ, where greater ψ typically indicates higher diversity but less fidelity. We find that our methods require a trade-off between diversity and fidelity and a proper ψ (*e.g.* 0.7, 0.8) is needed to achieve optimal performance.

5 Conclusion

We propose a gradient-matching-based method to learn to decode segmentation masks from generative features for images generated from pre-trained generative models with a few labeled examples. Unlike existing methods that train such a decoder in supervised learning way, our methods train such a decoder by matching gradients of a student network induced by the synthetic data and the labelled examples. Our methods mitigate the limitation of existing methods that labeled examples should be generated by the generator yet not compromise the performance on semi-supervised part segmentation tasks. Notably, our methods significantly outperform other semi-supervised semantic segmentation methods on a variety of semi-supervised part segmentation benchmarks. As progress in generative models continues to be made, future work would be using more powerful pre-trained generative models and applying our methods to more variety of computer vision tasks and low-supervision settings.

Reproducibility Statement

We provide all the details about the datasets, network structure, and training strategy in the appendix. Code and trained models to reproduce our main results will be made public.

REFERENCES

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014.

Jian Dong, Qiang Chen, Xiaohui Shen, Jianchao Yang, and Shuicheng Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Jianwen Xie, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. *arXiv preprint arXiv:1805.04310*, 2018.

Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, 2019.

Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Wei Chih Hung, Yi Hsuan Tsai, Yan Ting Liou, Yen-Yu Lin, and Ming Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020.

P Izmailov, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *ICLR*, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020a.

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

Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *ECCV*, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020b.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML workshop*, 2013.

Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *CVPR*, 2021.

Wei-Hong Li, Chuan-Sheng Foo, and Hakan Bilen. Learning to impute: A general framework for semi-supervised learning. *arXiv preprint arXiv:1912.10364*, 2019.

Xiaomeng Li, Lequan Yu, Hao Chen, Chi-Wing Fu, Lei Xing, and Pheng-Ann Heng. Transformation-consistent self-ensembling model for semisupervised medical image segmentation. *TNNLS*, 2020.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 2018.

Xuecheng Nie, Jiashi Feng, and Shuicheng Yan. Mutual learning to adapt for joint human parsing and pose estimation. In *ECCV*, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *CVPR*, 2020.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *CVPR*, 2021.

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *ICLR*, 2020.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020.

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 2020.

Yafei Song, Xiaowu Chen, Jia Li, and Qinping Zhao. Embedding 3d geometric features for rigid object part segmentation. In *ICCV*, 2017.

Nurit Spingarn-Eliezer, Ron Banner, and Tomer Michaeli. Gan steerability without optimization. *ICLR*, 2021.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation. In *CVPR*, 2021.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *ICML*, 2020.

Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015.

Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *ICCV*, 2015.

Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *CVPR*, 2017.

Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ToG*, 2016.

Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *CVPR*, 2021.

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.

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, 2019.

Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *ICLR*, 2020.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021.

Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *ICML*, 2021.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *ICLR*, 2021.

A Appendix

You may include other additional sections here.