

Generator Knows What Discriminator Should Learn in Unconditional GANs

Gayoung Lee¹ Hyunsu Kim¹ Junho Kim¹
 Seonghyeon Kim² Jung-Woo Ha¹ Yunjey Choi¹

¹NAVER AI Lab, ²NAVER CLOVA

{gayoung.lee, hyunsu1125.kim, jhkim.ai,
 kim.seonghyeon, jungwoo.ha, yunjey.choi}@navercorp.com

Abstract

Recent methods for conditional image generation benefit from dense supervision such as segmentation label maps to achieve high-fidelity. However, it is rarely explored to employ dense supervision for unconditional image generation. Here we explore the efficacy of dense supervision in unconditional generation and find generator feature maps can be an alternative of cost-expensive semantic label maps. From our empirical evidences, we propose a new generator-guided discriminator regularization (GGDR) in which the generator feature maps supervise the discriminator to have rich semantic representations in unconditional generation. In specific, we employ an U-Net architecture for discriminator, which is trained to predict the generator feature maps given fake images as inputs. Extensive experiments on multiple datasets show that our GGDR consistently improves the performance of baseline methods in terms of quantitative and qualitative aspects. Code is available at <https://github.com/naver-ai/GGDR>.

1. Introduction

Generative adversarial networks (GANs) have achieved promising results in various computer vision tasks including image [27–29] or video generation [53, 59, 60, 65], translation [7, 20, 31, 34, 73], manipulation [3, 15, 22, 30, 36, 50], and cross-domain translation [18, 35] for the past several years. In GANs, building an effective discriminator is one of the key components for generation quality since the generator is trained by the feedback from the discriminator. Existing studies proposed various methods to make the discriminator learn better representations by data augmentation [26, 69–71], gradient penalty [42, 43, 46], and carefully designed architectures [25, 49].

One simple yet effective way to improve the discriminator is to provide available additional annotations such as class la-

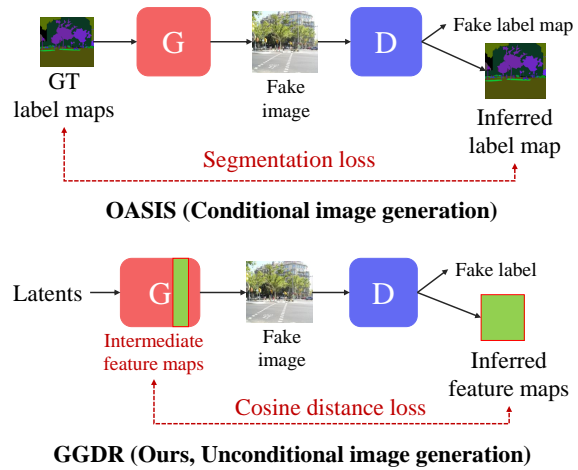


Figure 1. Comparison of how to provide semantic information between OASIS and our method. OASIS enhance the discriminator setting with the ground truth label maps in conditional image generation setting. GGDR, on the other hand, aims at unconditional image synthesis, and uses the generator feature maps instead of human-annotating label maps.

bels [5, 44], pose descriptors [52], normal maps [62], and semantic label maps [20, 41, 45, 56]. Among these annotations, semantic label maps contain rich and dense descriptions about images, and have been frequently used in conditional scene generation. To provide dense semantic information to the discriminator, Pix2pix [20] and SPADE [45] concatenate the label maps with input images, and CC-FPSE [41] uses projection instead of the concatenation to inject the embedding of label maps. OASIS [56] further enhances the discriminator by providing strong supervision using auxiliary semantic segmentation task and achieves better performance.

Despite the success of dense semantic supervision in conditional generation, it has been rarely explored in an unconditional setting. Dense semantic supervision can be useful here as well, as GAN models often struggle when

the data has varied and complex layout images. However, in unconditional generation, most large datasets do not have pairs of images and semantic label maps, since collecting them has a significant human annotation cost. Therefore, unlike the conditional setting, which requires a dense label map for the generator input, unconditional image generation assumes no dense map, and most studies use discriminators that learn only from images.

In this paper, we show that guiding a discriminator using dense and rich semantic information is also useful in unconditional image generation, and propose the method that avoids data annotation costs while utilizing semantic supervision. We propose *generator-guided discriminator regularization (GGDR)* in which the generator feature maps supervise the discriminator to have rich semantic representations. Specifically, we redesign the discriminator architecture in U-Net style, and train the discriminator to estimate the generator feature map when input is a generated image. As shown in **Fig. 1**, GGDR differs from the previous work in that the discriminator is supervised by the generator feature maps instead of human-annotated semantic label maps.

To justify our proposed method, we first compare the generation performance of StyleGAN2 [29] with and without providing ground-truth segmentation maps to the discriminator, and show that utilizing semantic label maps indeed improves the generation performance in an unconditional setting (**Section 2.1**). We then visualize the generator feature maps and show that they contain semantic information rich enough to guide the discriminator, replacing the ground-truth label maps (**Section 2.2**). Utilizing the generator feature maps, GGDR improves the discriminator representation, which is the key component to enhance the generation performance (**Section 3**). We provide thorough comparisons to demonstrate that GGDR consistently improves the baseline models on a variety of data. Our method can be easily attached to any setting without burdensome cost; only 3.7% of the network parameter increased. Our contributions can be summarized as follows:

1. We investigate the effectiveness of dense semantic supervision on unconditional image generation.
2. We show that generator feature maps can be used as an effective alternative of human-annotated semantic label maps.
3. We propose generator-guided discriminator regularization (GGDR), which encourages the discriminator to have rich semantic representation by utilizing generator feature maps.
4. We demonstrate that GGDR consistently improves the state-of-the-art methods on multiple datasets, especially in terms of generation diversity.

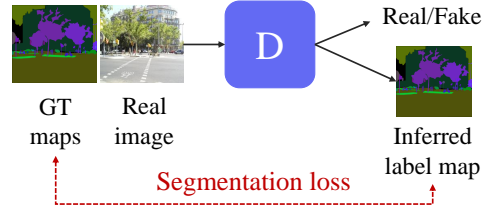


Figure 2. Discriminator architecture with the auxiliary segmentation loss for the preliminary experiments.

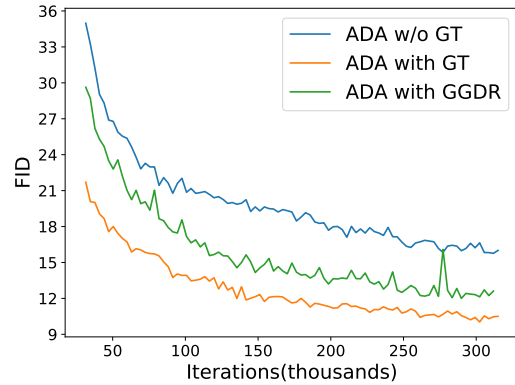


Figure 3. FID scores on ADE20K with and without dense semantic supervision in the preliminary experiments.

2. Dense semantic supervision in unconditional GANs

We first conduct a preliminary experiment using ground-truth segmentation maps to show the efficacy of providing dense semantic supervision for the discriminator (**Section 2.1**). Then, we study whether the generator feature maps can be used as a guide instead of using human annotating ground-truth label maps to avoid expensive manual annotations. We visualize the internal feature maps of the generator and show that they have semantic information rich enough to be used as pseudo-semantic labels (**Section 2.2**).

2.1. Utilization of semantic label maps for discriminator

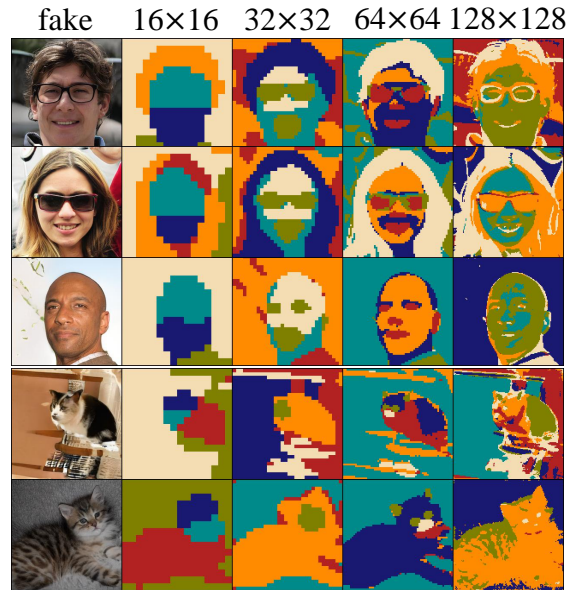
Although it is natural to utilize semantic label maps in conditional image generation [41, 45, 56], it has been still underexplored whether label maps are beneficial for unconditional image generation [26, 28, 29]. We conduct a preliminary experiment to validate the effects of the semantic label maps. We use ADE20K scene parsing benchmark dataset [72] consisting of 20,210 paired images and semantic label map annotations with 150 class labels, which is frequently used to evaluate conditional generation models. We choose StyleGAN2 [29] as our baseline, which is a standard model for unconditional image generation and apply adaptive discriminator augmentation [26]. To provide semantic

supervision for the network, we redesign the discriminator to perform additional segmentation task similar to OASIS [56]. The modified task for the discriminator is described in Fig. 2. The detailed architecture is similar to Fig. 5, except that it upsamples the decoder output until the image size is reached. The decoder in a U-Net style is attached to the discriminator and the segmentation loss is applied to the last layer of the decoder to provide dense supervision. The segmentation loss is the usual cross-entropy loss. Since ground-truth label maps are not available for generated images, we activate the segmentation loss for real images only.

As shown in Fig. 3, the model with the discriminator leveraging the semantic supervision(ADA with GT) outperforms the baseline(ADA w/o GT). As argued in OASIS, the stronger semantic supervision seems to help discriminator learn more semantically and spatially-aware representations and give the generator more meaningful feedback. Our experiment supports that providing additional semantic guide for the discriminator can improve the model performance in unconditional image synthesis. However, dense label maps are rare in datasets for unconditional image synthesis, and it is time-consuming to collect them manually. In the next section, we analyze the feature maps from the generator as an effective alternative for the ground-truth label maps.

2.2. Analysis of generator feature maps

Recent studies have reported that the feature maps of the trained generator of GANs contain rich and dense semantic information [9, 11, 63]. Collins et al. [9] showed that applying k -means clustering to the feature maps of the generator reveals semantics and parts of objects, and used the clusters to edit images. We notice that these feature maps are rich semantic descriptors of the generated images and can be the substitute for the ground truth label maps. To visualize what information is captured in each feature map, we run k -means algorithm on each layer using the batch of generated images. We set $k = 6$ in this experiment. As shown in Fig. 4 (a), the pixels are clustered by the semantic information instead of the low-level features except the last feature map. For example, the hairs of the people have different colors, but are clustered in to the same cluster. The early feature maps show coarse object location, and those from the latter layers contain detailed object parts. The visualized feature maps look like pseudo-semantic label maps and might be regarded as rich descriptions including spatial and semantic information about the images. Therefore, we choose the feature maps of the generator as the substitute for the semantic label maps to guide the discriminator using semantic supervision. The generator feature maps are useful in our case. First, we do not need perfect semantic segmentation maps because our goal is image generation not semantic segmentation. Second, the feature maps are intermediate by-products essential for the generation, so acquiring them is free and does not require



(a) Generator feature map visualization



3k iters (0.8%) 6k iters(1.6%)

(b) Generator feature map in early training phase

Figure 4. Visualization of the generator feature maps using k -means($k = 6$) clustering. (a) StyleGAN2 generator feature map. The visualized feature maps reveal semantically consistent and meaningful regions such as ears in cats. (b) Generated images and their 32×32 feature maps in the early training phase.

additional human annotations.

Dissimilar to previous works [9, 11, 63] that utilize the generator feature map for separate tasks, our method utilizes them during the training to enhance the generation performance itself. Therefore, it is essential to check whether the feature maps from the generator in the middle of training are still semantically meaningful for the guidance. In Fig. 4 (b), we visualize the feature maps of the generator during training to check how early the feature maps become semantically meaningful. Surprisingly, thanks to the powerful modern GANs, we can observe that even in the early stage, the feature maps and the corresponding generated images capture coarse shapes and location of objects. Therefore, we utilize the feature maps from the beginning of training, but for more complex data where the generator needs more iterations to produce meaningful semantics, one may choose when to attach our objective function.

3. Generator-guided discriminator regularization

Based on our observations, we propose *generator-guided discriminator regularization* (GGDR) in which the generator feature maps supervise the discriminator to have rich semantic representations. The overall framework is shown in Fig. 5.

The design of our discriminator D is inspired by that of OASIS [56] where the U-Net encoder-decoder structure is adopted and the last layer predicts semantic label maps. However, unlike OASIS, we leverage feature maps of the generator instead of ground-truth label maps. Thus, there are several differences in the design. First, since the feature maps are not discrete labels anymore, we cannot simply add real/fake class to the decoder output as done in OASIS. Therefore, we separate the decoder and adversarial loss. Next, we use more compact and lighter modules to reduce additional calculation costs. For each layer, we concatenate the output from the decoder and the encoder layer, and run one linear 1×1 convolutional layer with upsampling. We stack the decoder modules until the decoder output has the same resolution with the targeted generator feature map. Although the decoder is compact, it is sufficient to predict the generator feature map as the shared encoder can extract semantic information.

Meanwhile, the encoder part is still shared, and thus it is trained via both semantic and adversarial loss. For the adversarial loss, we adopt the non-saturating adversarial loss [13]:

$$\min_G \max_D \mathcal{L}_{adv}(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (1)$$

Next, we compute the cosine distance loss between the output of the decoder and the target feature map. We use cosine distance loss since it gives a loss within a specified range even between denormalized feature vectors, so it is convenient to scale according to the adversarial loss. Here, we denote $l \in \{1, 2, \dots, L\}$ as a layer index, where L is the number of decoding layers. Our discriminator D contains an U-Net-style decoder F ($F \subset D$) and the output of each layer denoted as F^l has the same resolution with the corresponding generator feature map $G(z)^l$. Let us denote the target layer index for guidance as t . Our generator-guided discriminator regularization (GGDR) is defined as:

$$\max_D \mathcal{L}_{ggdr}(G, D) = - \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[1 - \frac{F^t(G(\mathbf{z})) \cdot G(\mathbf{z})^t}{\|F^t(G(\mathbf{z}))\|_2 \cdot \|G(\mathbf{z})^t\|_2} \right] \quad (2)$$

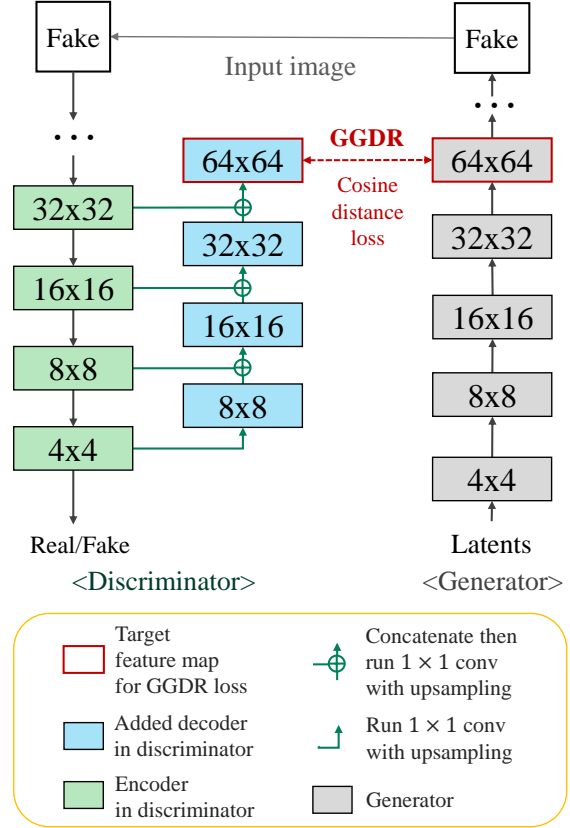


Figure 5. Visualization of our framework. Our method can be applied to a GAN model by adding a decoder and the cosine distance loss with the reference generator feature maps to a discriminator. The generator feature maps guide the discriminator to learn more semantically-aware representations.

Our full objective functions can be summarized as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_{reg} \mathcal{L}_{ggdr} \quad (3)$$

where λ_{reg} is a hyperparameter for relative strength compared to the adversarial term. Note that k -means clustering used in Section 2.2 is only for visualization purpose and we directly compare raw feature maps without any clustering. We expect the \mathcal{L}_{ggdr} term to enhance the semantic representation of D . While the generator feature maps participate the regularization loss, we do not update the generator with \mathcal{L}_{ggdr} to prevent a feature collapse which is a trivial solution making the cosine distance to zeros.

Our framework is simple and easy to apply existing GAN models, and does not require any additional annotation. Acquiring the intermediate feature maps from a generator is free because the generator already produces them in order to generate fake images. Despite its simplicity, in the next section, we show the effectiveness of our method in unconditional image generation for various datasets.

Table 1. FID scores of ours and comparison methods on FFHQ. We run three training for each data and show their means and standard deviations. The numbers are largely brought from ADA [26] and we follow their evaluation protocol.

FFHQ	2k	10k	140k
PA-GAN	56.49±7.28	27.71±2.77	3.78±0.06
WGAN-GP	79.19±6.30	35.68±1.27	6.43±0.37
zCR	71.61±9.64	23.02±2.09	3.45±0.19
AR	66.64±3.64	25.37±1.45	4.16±0.05
StyleGAN2	78.80±2.31	30.73±0.48	3.66±0.10
+GGDR	70.59±5.16	24.44±0.63	3.14±0.03
ADA	16.49±0.65	8.29±0.31	3.88±0.13
+GGDR	18.28±0.77	6.11±0.15	3.57±0.10

4. Experiments

We validate the efficacy of our GGDR on various datasets including CIFAR-10 [37], FFHQ [28], LSUN cat, horse, church [64], AFHQ [8] and Landscapes [2]. CIFAR-10 consists of 50,000 tiny color images in 10 classes. FFHQ contains 70,000 face images, and AFHQ includes approximately 5,000 images per cat, dog and wild animal faces. LSUN cat, horse and church consist of scenes with cat, horse and church respectively, and we have used 200,000 images per each dataset. Landscapes contains photographs 4,320 landscape images collected from Flickr [1]. Following StyleGAN2 and ADA [26], we have applied horizontal flips for FFHQ and small datasets. All images are resized to 256×256 except AFHQ (512×512) and CIFAR-10 (32×32). For GGDR loss, we select the 64×64 feature map of the generator as the guidance map, except CIFAR-10 where we select the 8×8 feature map. We set $\lambda_{reg} = 10$ for the weight of the proposed regularization in all experiments and let other hyperparameters unmodified. We apply R_1 regularization [42] for StyleGAN2 and ADA models. In the case of the ADA, we apply the augmentation to the generator feature maps to make them consistent with corresponding fake images. We use only geometric operations and skip color transformation for the feature map augmentation.

For evaluation metrics, we have used Fréchet Inception Distance (FID) [16] and Precision & Recall [39]. FID measures the distance between the real images and the generate samples in feature space, and Precision & Recall scores indicate sample quality and variety. We compare 50,000 generated images and all training images following previous works [26]. For CIFAR-10, we also use Inception Score (IS) [47] following the previous works [26, 54].

4.1. Comparison with baselines

We apply GGDR to StyleGAN2 [29] which is one of the standard models for unconditional image generation. Instead

Table 2. FID scores of ours and comparison methods on CIFAR-10. We run three training for mean and standard deviations. We brought the numbers of diffusion models from [54].

CIFAR-10	FID	IS
ProGAN	15.52	8.56±0.06
AutoGAN	12.42	8.55±0.10
StyleGAN2	8.32±0.09	9.21±0.09
ADA	2.92±0.05	9.83±0.04
FSMR	2.90	9.68
DDPM	3.17±0.05	9.46±0.11
NCSN++	2.2	9.89
ADA+GGDR	2.15±0.02	10.02±0.06

of the original StyleGAN2 setting, we use the baseline setting used in ADA [26] which has less parameters and shorter training iterations but shows comparable performance. For small datasets, we apply the adaptive discriminator augmentations (ADA) [26] that prevents overfitting of the discriminator and shows the superior performance in small datasets.

Table 1 and **Table 2** shows that the proposed GGDR improves the performance of the baselines in terms of FID scores which indicate the overall quality of the synthesized images. Following ADA [26], we run the experiment multiple times on FFHQ with varying numbers (2k, 10k and 140k) of training images. We largely borrowed the reported scores from ADA [26] and NCSN++ [54]. In **Table 1**, we compare our method in a varied number of training images with various regularizing methods WGAN-GP [14], PA-GAN [67], zCR [71], AR [6] and ADA [26]. In the table, StyleGAN2 with our GGDR achieves the best score on FFHQ with full training images. With a sufficient number of training images, regularization methods based on data augmentation show limited improvement or degradation, whereas in our method the discriminator learns from a more accurate semantic map provided by a better quality generator. In most cases, GGDR improves the baseline performance significantly except the FFHQ 2k setting. We conjecture the dataset is too small to learn semantically meaningful feature maps in the generator. Since the quality of generator feature maps directly affects the discriminator in our method, our method is more effective when there is sufficient number of training data. However, as shown in **Table 4**, our method shows effectiveness on the datasets with approximately 5,000 images which are not too many to collect. **Table 2** shows that our GGDR significantly improves ADA performance in terms of both FID and IS scores, and makes it superior to various models including ProGAN [25], AutoGAN [12], FSMR [33] and DDPM [17], and comparable to the NCSN++ [54].

In **Tables 3** and **4**, we conduct extensive experiments to validate the performance improvement using GGDR on vari-

Table 3. Comparison on FFHQ, LSUN Cat, LSUN Horse and LSUN Church. Our method improves StyleGAN2 [29] in large datasets in terms of FID and recall. P and R denote precision and recall. Lower FID and higher precision and recall mean better performance. The bold numbers indicate the best FID, P, R for each dataset.

Method	FFHQ			LSUN Cat			LSUN Horse			LSUN Church		
	FID↓	P↑	R↑	FID	P	R	FID	P	R	FID	P	R
UT [4]	6.11	0.73	0.48	-	-	-	-	-	-	4.07	0.71	0.45
Polarity [19]	-	-	-	6.39	0.64	0.32	-	-	-	3.92	0.61	0.39
StyleGAN2	3.71	0.69	0.44	7.98	0.60	0.27	3.62	0.63	0.36	3.97	0.59	0.39
+GGDR	3.14	0.69	0.50	5.28	0.58	0.38	2.50	0.64	0.43	3.15	0.61	0.46

Table 4. Comparison on AFHQ Cat, Dog, Wild and Landscape. Our method improves ADA [26] in small datasets in terms of FID and recall. P and R denote precision and recall. The bold numbers indicate the best FID, P, and R of the models.

Method	AFHQ Cat			AFHQ Dog			AFHQ Wild			Landscape		
	FID↓	P↑	R↑	FID	P	R	FID	P	R	FID	P	R
FastGAN [40]	4.69	0.78	0.31	13.09	0.75	0.38	3.14	0.76	0.20	16.44	0.77	0.16
ContraD [23]	3.82	-	-	7.16	-	-	2.54	-	-	-	-	-
ADA	3.55	0.77	0.41	7.40	0.76	0.48	3.05	0.76	0.13	13.87	0.72	0.20
+GGDR	2.76	0.74	0.52	4.59	0.79	0.53	2.06	0.80	0.27	10.38	0.69	0.29

ous datasets. For FFHQ and LSUN datasets, we report the scores of UT [4], and Polarity [19] which are the state-of-the-arts models that show the improvement on these datasets. For AFHQ and Landscape datasets, we report the score of ContraD [23] and FastGAN [40] that show significant improvements on image synthesis with small size datasets. We brought the numbers from their papers except FastGAN whose scores are brought from ProjGAN [48]. GGDR consistently improves the baseline in terms of FID scores with large gap. In terms of precision and recall metrics, GGDR improves the recall with significant margins compared to the baselines, which indirectly indicates where the advantages of our method come from. Better recall scores mean that our model generates more diverse images and less prone to the mode collapse. As it is known that incorporating image-level labels to the discriminator enhances coverage of classes in the data, utilizing pseudo dense semantic information could facilitate semantic diversities in the generated images.

In Fig. 9 and Fig. 8 (a), we show some selected results of our method on the evaluated datasets. We visualize StyleGAN2 with GGDR for FFHQ and LSUN, and ADA with GGDR for other datasets. More uncurated images are shown in the supplementary material. Since our method tends to improve the recall than the precision, it is hard to show visual improvement with the limited numbers of samples. Instead, we compare the worst samples in Fig. 6 (a). We follow the method of [38] to sort the samples, which uses the Inception [57] model to fit a gaussian model and sorts

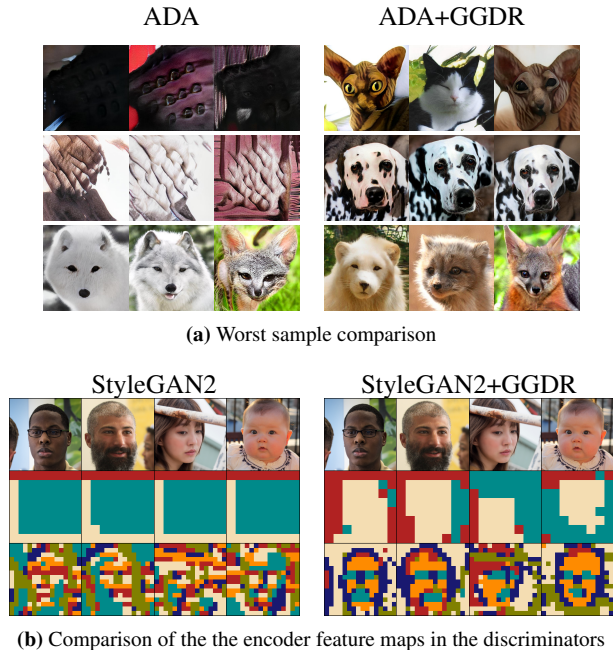
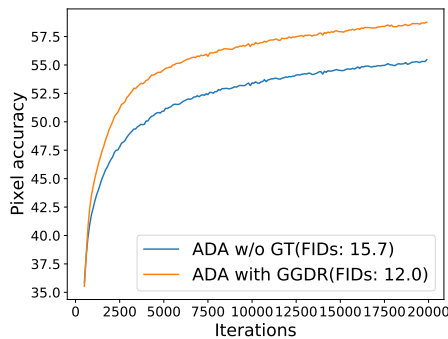


Figure 6. (a) Qualitative comparison of worst-sample images with and without GGDR on AFHQ. (b) k -means clustering of the feature maps of the encoder in the discriminator with and without GGDR loss. From top to bottom, real images, feature maps that are $8(k=3)$ and $16(k=6)$ pixels wide.



(a) FIDs in the early training phase



(b) Downstream task result

Figure 7. (a) FID score graph of multiple experiments with GGDR in the early training phase on LSUN Cat. (b) Validation pixel accuracy on the ADE20K segmentation task with the frozen discriminator trained with and without GGDR.

by the log-likelihood using it. We can see the worst samples of our method still contain objects unlike those of ADA. It is interesting that [38] reports similar improvements on the worst samples by utilizing pretrained models for the discriminator. We conjecture that the feature maps of the generator plays similar role with the pretrained models in their works [38, 48].

4.2. Analysis and ablation study

The proposed GGDR supervises a discriminator using the intermediate feature maps of a generator in a GAN model, so it depends on the quality of the generator feature maps. In Sec. 2.2, we show that the feature maps of the generator contain valid semantic information even in the early stage. In addition to the visualization, we measure the FIDs in early iterations and show in Fig. 7 (a). To check the effect of initializations, we run multiple experiments on LSUN Cat. In early iterations, GGDR can interfere the performance by using less trained feature maps and bad initialization. However, after only several thousand iterations, StyleGAN2 with GGDR starts to converge faster and shows better scores.

In Table 5, we conduct ablation studies to investigate

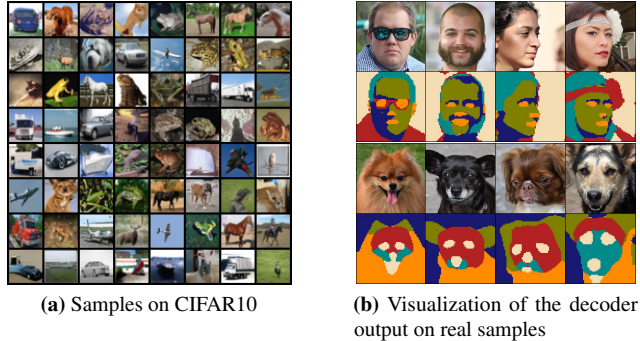


Figure 8. (a) Random samples by ADA with GGDR on CIFAR-10 dataset. (b) k -means clustering of the feature maps of the decoder in our discriminator on real images ($k = 6$).

Target	FID	Activation	FID
None	7.98	Linear	5.28
8×8	7.57	leaky ReLU	5.43
16×16	6.56	Kernel size	FID
32×32	5.98	1×1	5.28
64×64	5.28	3×3	5.25

(a) Target size

(b) Decoder design

Method	# params	time(s)
Baseline	4.87M	5.60
+ GGDR	5.05M	6.05
	(+3.7%)	(+8.0%)

(c) Calculation costs

Table 5. Ablation studies and calculation costs on LSUN Cat with eight V100 GPUs. Ablation study on (a) the target feature map size and (b) the decoder design. (c) Calculation costs with and without GGDR.

the effects of the decoder architecture and the feature map resolution. In our experiments, we select 64×64 feature maps as the guidance. One may curious the performance differences if we use different sizes of the guidance feature maps. As shown in Table 5 (a), utilizing the large and dense feature maps achieves the best FID scores. Meanwhile, we design a compact decoder with 1×1 convolutional filters and linear activation for fast training and convergence. In Table 5 (b), we show that changing decoder activation and kernel size affects only negligible performance difference.

To analyze the effects of GGDR, we visualize and compare the original discriminator part in Fig. 6 (b). We run k -means clustering as done in Sec. 2.2. With GGDR loss, the shared encoder part prefers to learn high-level features



Figure 9. Selective samples generated by our method. For FFHQ and LSUN datasets, we show the results of StyleGAN2 with GGDR. For AFHQ and Landscape, we show the results of ADA with GGDR.

which are useful for both tasks, so its feature maps reveal more semantically meaningful clusters. By guiding to learn semantic features, our approach can help discriminators to focus on salient parts of the image instead of meaningless features. To further analyze, we conduct a downstream experiment, training a shallow segmentation network using the extracted features by the discriminator with or without GGDR. As shown in **Fig. 7 (b)**, the accuracy on validation data shows the discriminator with GGDR has more representation power on semantic information. Meanwhile, since we trained on fake images only, it may be curious if the guidance by fake images is still valid for real images. In **Fig. 8 (b)**, we visualize the outputs of the decoder of our discriminator on real image. While the decoder of our method learned using fake images, we can see that its features well capture the semantically meaningful regions of the real images.

In **Table 5 (c)**, we show the additional calculation costs when use GGDR. We can see the additional costs are marginal where the parameters increase 3.7% and the time increases 8.0%. For these measurement, we run the StyleGAN2 on the 256×256 dataset with eight V100 GPUs.

5. Related Work

Conditional image synthesis utilizing semantic label map. For the controllability of the generated images, it is common to exploit semantic layout-level information for the conditional image generation [21, 41, 61]. SPADE [45] utilizes Spatially-Adaptive Denormalization which preserves semantic informations, and OASIS [56] have shown that it is able to train the conditional GANs using the discriminator that predicts pixel-level semantic labels, without incorporating semantic maps as additional conditions. Also, instead of using explicit semantic ground-truths, it is possible to use the features from the deep networks for the semantic guidance of the generator as shown in [10, 51]. Unlike these works, our method aims at unconditional image generation.

Regularization for GANs. Several works interest to stabilizing the GAN trainings, especially by regularizing the discriminators [33, 42, 43]. Recently, utilizing augmentations for the discriminator gained a lot of interests, which was proven successful in general vision tasks [66, 68]. In consistency regularization (CR) [69, 71], in addition to using regular GAN losses, a discriminator is penalized by the differences in

the outputs between augmented and non-augmented images. APA [24] regularizes the discriminator by utilizing fake images as pseudo-real data adaptively. DiffAugment [70] and ADA [26] use non-leaking augmentations for both generator and discriminator losses. Recently, several papers use pretrained models to help discriminator for fast and stable training. ProjGAN [48] uses EfficientNet [58] as a feature extractor for the discriminator, and Vision-aided GAN [38] provides automatic selection from model bank of pretrained networks to get optimal features for real and fake discrimination.

Utilization of generator features. Recent studies have shown that the generators contain rich and disentangled semantic structures in the features. Collins et al. [9] show that by applying k -means clustering on feature activations of the generator is possible to extract semantic objects and object parts in the generated images. Xu et al. [63] have trained linear mapping between feature maps in the generator and semantic maps, and Endo et al. [11] used the nearest neighbor matching between feature maps and representative vectors by averaging the feature vectors corresponding to the ground-truth semantic labels of inverted images. StyleMapGAN [30] has used spatial dimensions in the latent codes and grouping of the channels to further disentangle spatial semantic features.

6. Conclusion and limitation

In this paper, we present the efficacy of the dense semantic label maps for unconditional image generation. Inspired by this observation, we propose a new regularization method to leverage the feature maps of the generator instead of human annotating ground-truth semantic annotations to allow the discriminator to learn richer semantic representation. With negligible additional parameters and no ground-truth semantic segmentation map, the proposed GGDR consistently outperforms strong baselines. Since our method depends on the performance of the generator, if the generator cannot learn meaningful representations due to the extremely limited number of data or initial training collapse, GGDR will fail to improve the performance. However, thanks to modern GANs and training techniques, we believe our method can be easily applied in various situations.

Acknowledgement.

The experiments in the paper were conducted on NAVER Smart Machine Learning (NSML) platform [32, 55]. We thank to Jun-Yan Zhu, Jaehoon Yoo, NAVER AI LAB researchers and the reviewers for their helpful comments and discussion.

References

[1] Flickr. <https://www.flickr.com/>. 5

[2] Kaggle landscapes dataset. <https://www.kaggle.com/arnaud58/landscape-pictures>, 2019. 5, 11

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *ICLR*, 2019. 1

[4] Sam Bond-Taylor, Peter Hesse, Hiroshi Sasaki, Toby P Breckon, and Chris G Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. *arXiv preprint arXiv:2111.12701*, 2021. 6

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ICLR*, 2019. 1

[6] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *CVPR*, 2019. 5

[7] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 1

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 5, 11

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *CVPR*, 2020. 3, 9

[10] Luke Ditria, Benjamin J. Meyer, and Tom Drummond. Open-gan: Open set generative adversarial networks. In *ACCV*, 2020. 8

[11] Yuki Endo and Yoshihiro Kanamori. Few-shot semantic image synthesis using stylegan prior. *CoRR*, abs/2103.14877, 2021. 3, 9

[12] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *ICCV*, 2019. 5

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 4

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 5

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *NeurIPS*, 2020. 1

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 5

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 5

[18] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *ECCV*, 2018. 1

- [19] Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Polarity sampling: Quality and diversity control of pre-trained generative networks via singular values. In *CVPR*, 2022. 6
- [20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1
- [21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 8
- [22] Ali Jahanian, Lucy Chai, and Phillip Isola. On the “steerability” of generative adversarial networks. In *ICLR*, 2020. 1
- [23] Jongheon Jeong and Jinwoo Shin. Training gans with stronger augmentations via contrastive discriminator. In *ICLR*, 2021. 6
- [24] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Deceive d: Adaptive pseudo augmentation for gan training with limited data. In *NeurIPS*, 2021. 9
- [25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018. 1, 5
- [26] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 1, 2, 5, 6, 9
- [27] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *NeurIPS*, 2021. 1
- [28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 2, 5, 11
- [29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 2, 5, 6
- [30] Hyunsu Kim, Yunjey Choi, Junho Kim, Sungjoo Yoo, and Youngjung Uh. Exploiting spatial dimensions of latent in gan for real-time image editing. In *CVPR*, 2021. 1, 9
- [31] Hyunsu Kim, Ho Young Jhoo, Eunhyeok Park, and Sungjoo Yoo. Tag2pix: Line art colorization using text tag with secat and changing loss. In *ICCV*, 2019. 1
- [32] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, et al. Nsm1: Meet the mlaas platform with a real-world case study. *arXiv preprint arXiv:1810.09957*, 2018. 9
- [33] Junho Kim, Yunjey Choi, and Youngjung Uh. Feature statistics mixing regularization for generative adversarial networks. In *CVPR*, 2022. 5, 8
- [34] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*, 2020. 1
- [35] Taeksoo Kim, Moonsoo Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 1
- [36] Yunji Kim and Jung-Woo Ha. Contrastive fine-grained class clustering via generative adversarial networks. *ICLR*, 2022. 1
- [37] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009. 5, 11
- [38] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for gan training. In *CVPR*, 2022. 6, 7, 9
- [39] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 2019. 5
- [40] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed El-gammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *ICLR*, 2020. 6
- [41] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, 2019. 1, 2, 8
- [42] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Which training methods for gans do actually converge? In *ICML*, 2018. 1, 5, 8
- [43] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 1, 8
- [44] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 1
- [45] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 2, 8
- [46] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. *NeurIPS*, 2017. 1
- [47] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 5
- [48] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *NeurIPS*, 2021. 6, 7, 9
- [49] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *CVPR*, 2020. 1
- [50] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, 2020. 1
- [51] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T. Freeman, and Tali Dekel. Semantic pyramid for image generation. In *CVPR*, 2020. 8
- [52] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018. 1
- [53] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*, 2022. 1
- [54] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

- generative modeling through stochastic differential equations. In *ICLR*, 2021. 5
- [55] Nako Sung, Minkyu Kim, Hyunwoo Jo, Youngil Yang, Jingwoong Kim, Leonard Lausen, Youngkwan Kim, Gayoung Lee, Donghyun Kwak, Jung-Woo Ha, et al. Nsml: A machine learning platform that enables you to focus on your models. *arXiv preprint arXiv:1712.05902*, 2017. 9
- [56] Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis. In *ICLR*, 2021. 1, 2, 3, 4, 8
- [57] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 6
- [58] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 9
- [59] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 1
- [60] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 1
- [61] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 8
- [62] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 1
- [63] Jianjin Xu and Changxi Zheng. Linear semantics in generative adversarial networks. In *CVPR*, 2021. 3, 9
- [64] 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. 5, 11
- [65] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *ICLR*, 2022. 1
- [66] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 8
- [67] Dan Zhang and Anna Khoreva. Pa-gan: Improving gan training by progressive augmentation. In *NeurIPS*, 2019. 5
- [68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 8
- [69] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. *ICLR*, 2020. 1, 8
- [70] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *NeurIPS*, 2020. 1, 9
- [71] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. *AAAI*, 2021. 1, 5, 8
- [72] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2
- [73] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1

A. Additional results and visualization

We additionally provide results and their corresponding feature visualization for the datasets used in the main paper including CIFAR-10 [37], FFHQ [28], LSUN cat, horse, church [64], AFHQ [8] and Landscapes [2]. From **Fig. 10** to **Fig. 14**, we show uncurated random samples generated by our method. For FFHQ and LSUN dataset, we sample from StyleGAN2 trained with GGDR, and we sample from ADA trained with GGDR for the others. From **Figure 15** to **Figure 17**, we visualize the guidance feature maps from the generator and their corresponding fake images. For feature map visualization, we run k -means($k = 6$) clustering on 64 samples. All visualized feature map size is 64×64 except CIFAR-10(8×8).



Figure 10. Uncurated results on (top) FFHQ and (bottom) LSUN Cat.



Figure 11. Uncurated results on (top) LSUN Horse and (bottom) LSUN Church.



Figure 12. Uncurated results on (top) AFHQ Cat and (bottom) AFHQ Dog



Figure 13. Uncurated results on (top) AFHQ Wild and (bottom) Landscape.

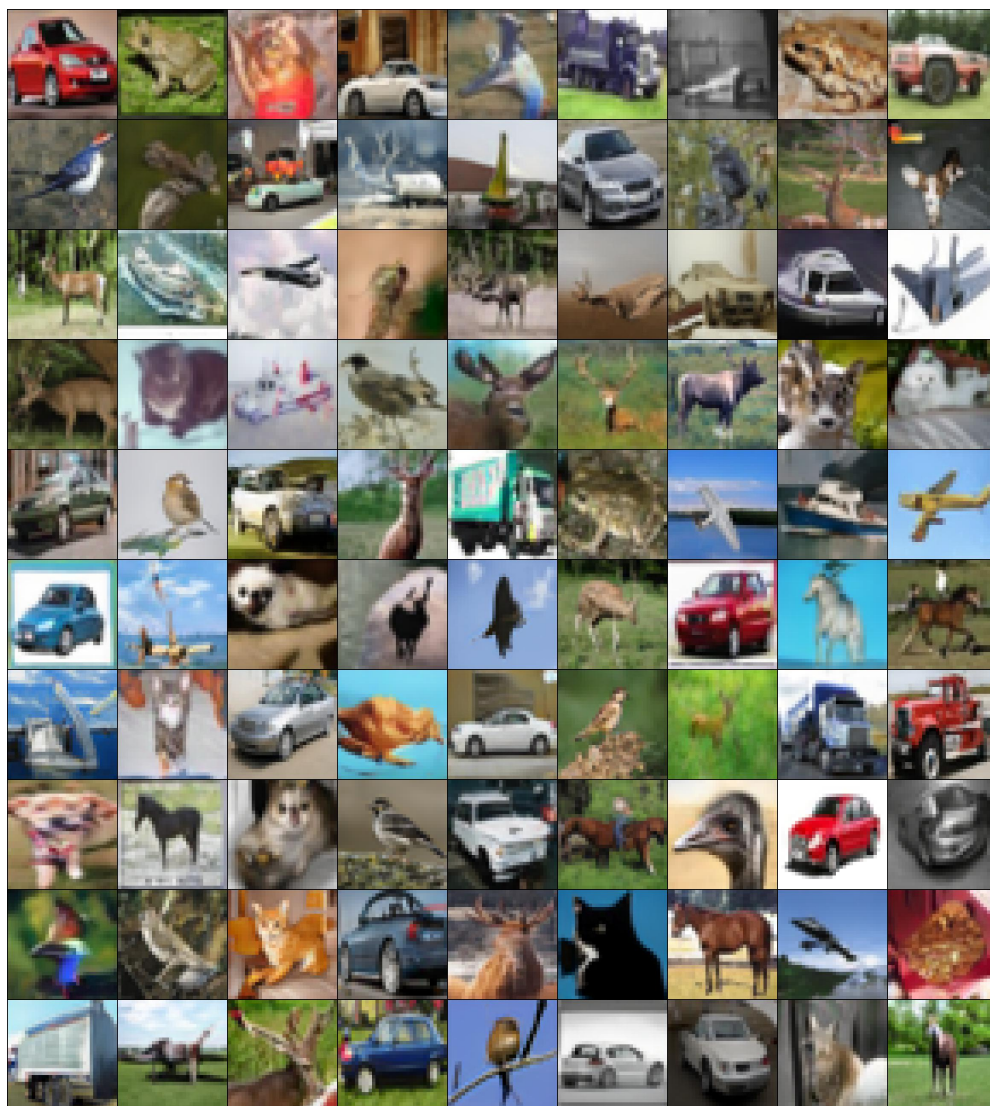
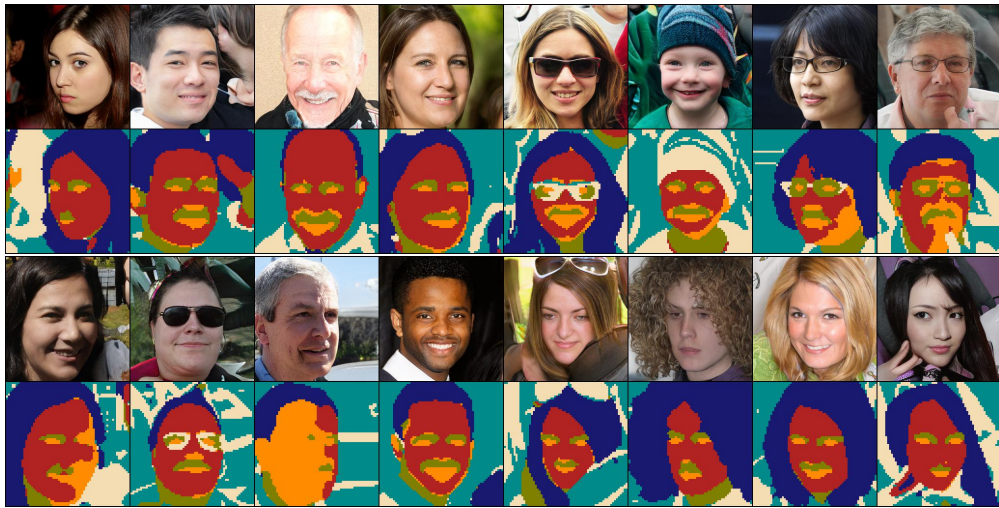


Figure 14. Uncurated results on CIFAR-10.



(a) Visualization on FFHQ.



(b) Visualization on LSUN Cat.



(c) Visualization on LSUN Horse.

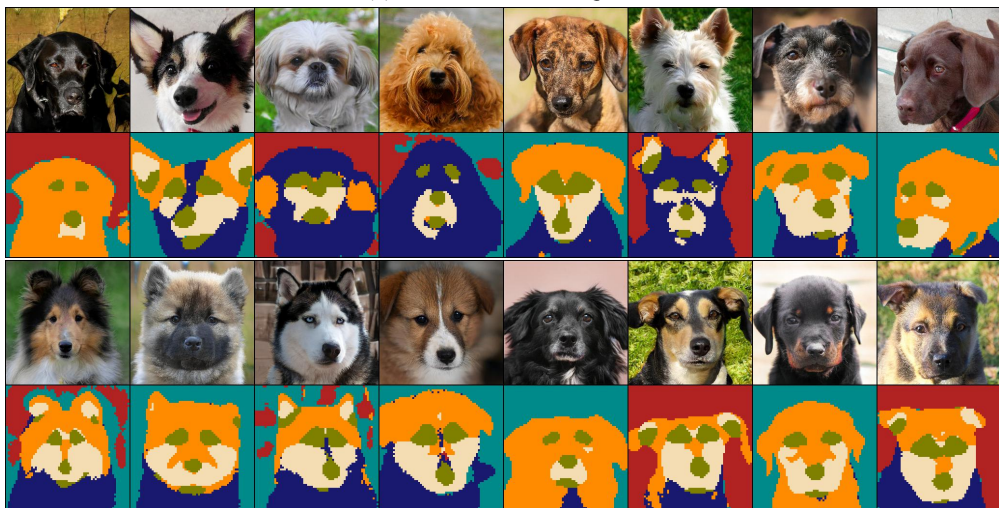
Figure 15. Visualization of fake images and their corresponding generator feature maps on FFHQ, LSUN Cat and LSUN Horse.



(a) Visualization on LSUN Church.

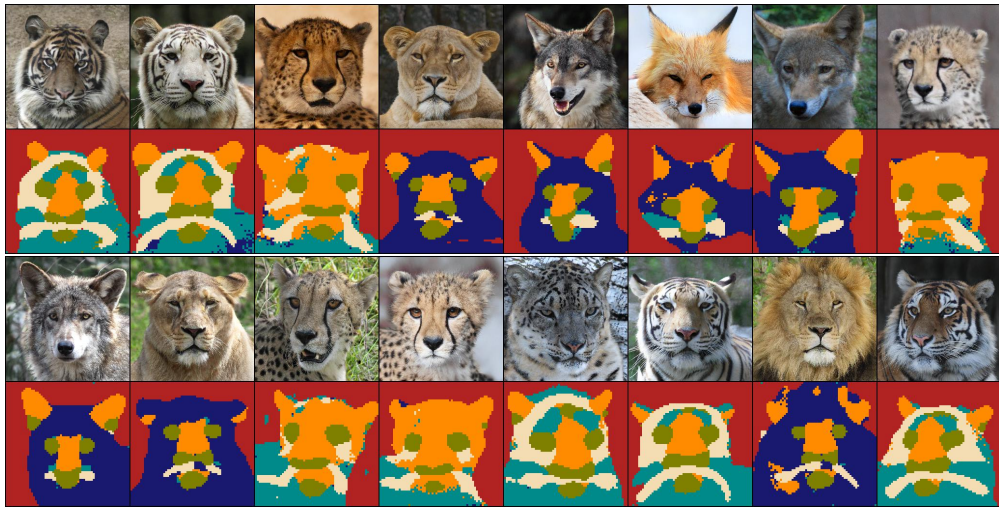


(b) Visualization on AFHQ Cat.

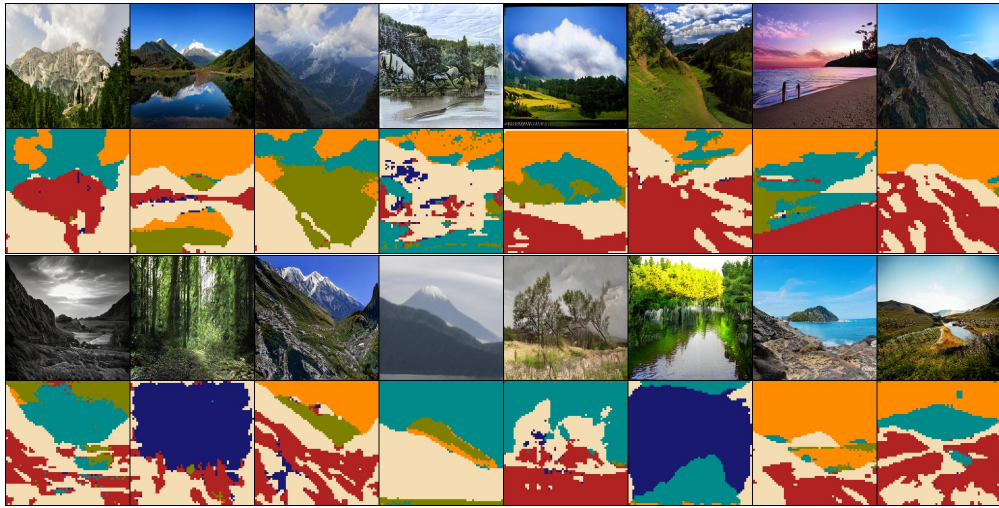


(c) Visualization on AFHQ Dog.

Figure 16. Visualization of fake images and their corresponding generator feature maps on LSUN Church, AFHQ Cat and AFHQ Dog.



(a) Visualization on AFHQ Wild.



(b) Visualization on Landscapes.



(c) Visualization on CIFAR-10.

Figure 17. Visualization of fake images and their corresponding generator feature maps on AFHQ Wild, Landscapes and CIFAR-10.