Semantically Multi-modal Image Synthesis

Zhen Zhu[1*], Zhiliang Xu[1*], Ansheng You[2], Xiang Bai[1†]

[1]*Huazhong University of Science and Technology*, [2]*Peking University*

{zzhu, zhiliangxu1, xbai}@hust.edu.cn, youansheng@pku.edu.cn

Abstract

*In this paper, we focus on **semantically multi-modal image synthesis** (SMIS) task, namely, generating multi-modal images at the semantic level. Previous work seeks to use multiple class-specific generators, constraining its usage in datasets with a small number of classes. We instead propose a novel Group Decreasing Network (GroupDNet) that leverages group convolutions in the generator and progressively decreases the group numbers of the convolutions in the decoder. Consequently, GroupDNet is armed with much more controllability on translating semantic labels to natural images and has plausible high-quality yields for datasets with many classes. Experiments on several challenging datasets demonstrate the superiority of GroupDNet on performing the SMIS task. We also show that GroupDNet is capable of performing a wide range of interesting synthesis applications. Codes and models are available at: https://github.com/Seanseattle/SMIS.*

1. Introduction

Semantic image synthesis, namely translating semantic labels to natural images, has many real-world applications and draws much attention from the community. It is innately a one-to-many mapping problem. Countless possible natural images correspond to one single semantic label. Previous works utilized different strategies for the task: adopting the idea of variational auto-encoder [36, 56, 2, 11], introducing noise while training [19], building multiple subnetworks [10] and including instance-level feature embeddings [41], *etc*. While these methods made exceptional achievements in improving image quality and extending more applications, we take a step further to particularly focus on a specific multi-modal image synthesis task that adds more flexibility to control the generated results.

Just imagine a content creation scenario from a human parsing map. With the help of semantics-to-image translation models, the parsing map can be converted to a real per-
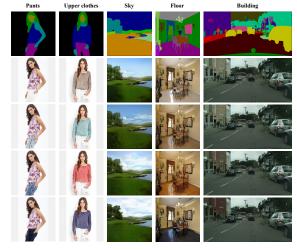
---

* Equal contribution
† Corresponding author

Figure 1: Demonstration of the semantically multi-modal image synthesis (SMIS) task. Text of each column above the images indicates the semantic area that is changing across the whole column. The first row represents the input labels and the rest rows are generated images by our method.

son image. It looks good in general, but the upper clothes do not suit your taste. Then comes the problem—either these models do not support multi-modal synthesis, or when these models change the upper clothes, other parts vary accordingly. Neither of these fulfills your intention. To conclude, this user controllable content creation scenario can be interpreted as performing a task that produces multi-modal results at the semantic level with other semantic parts untouched. We summarize this task as: **Semantically Multi-modal Image Synthesis** (SMIS). As exemplified in Fig. 1, for each semantics, we have its specific controller. By adjusting the controller of a specific class, only the corresponding areas are changed accordingly.

An intuitive solution for the task is to build different generative networks for different semantics and then produce the final image by fusing the outputs of different networks. It is quite similar to the overall scheme of [10], which focused on portrait editing. However, this type of methods soon face degradation in performance, a linear increase of training time and computational resource consumption under a growing number of classes.

To make the network more elegant, we unify the gen-

eration process in only one model by creatively replacing all regular convolutions in the generator with group convolutions. Our strategy is mathematically and functionally equivalent to [10] when the group numbers of the convolutions are equal to that of the classes. Another strategy we adopt, however, set our paths differently—we decrease the number of groups in the decoder convolutions in the forwarding progress. We observe that different classes have internal correlations among each other, *e.g.*, the color of the grass and the tree leaves should be highly similar. In this case, progressively merging the groups gives the model enough capacity to build inter-correlations among different classes, and consequently improves the overall image quality. Besides, this strategy also considerably mitigates the computation consumption problem when the class number of the dataset is substantial (*e.g.*, ADE20K [53]). We call the generator equipped with these two strategies as **Group Decreasing Network** (GroupDNet). To evaluate GroupDNet's performance on the SMIS task, we propose two new metrics called mean Class-Specific Diversity (mCSD) and mean Other-Class Diversity (mOCD). The mCSD metric tends to hold high value and the mOCD tends to be low when some semantic parts vary drastically but other parts stay unchanged.

We conduct experiments on several challenging datasets: DeepFashion [30], Cityscapes [5], and ADE20K [53]. The results show that our GroupDNet introduces more controllability over the generation process, and thus produces semantically multi-modal images. Moreover, GroupDNet maintains to be competitive with previous state-of-the-art methods in terms of image quality, exhibiting the superiority of GroupDNet. Furthermore, GroupDNet introduces much controllability over the generation process and has a variety of interesting applications such as appearance mixture, semantic manipulation, and style morphing.

2. Related work

Generative models. Generative Adversarial Networks (GANs) [9], comprised of a generator and a discriminator, have the amazing ability to generate sharp images even for very challenging datasets. [9, 21, 3, 22]. Variational autoencoder [24] contains an encoder and a decoder, and requires the latent code yield by the encoder to conform to Gaussian distribution. Its results usually exhibit large diversity. Some methods [2] combine VAE and GAN in their models, producing realistic while diverse images.

Conditional image synthesis. Conditional Generative Adversarial Networks [34] inspire a wide range of conditional image synthesis applications, such as image-to-image translation [19, 17, 41, 55, 27, 29, 15], super resolution [26, 20], domain adaption [14, 52], single model image synthesis [54, 37, 33, 38], style transfer [16, 7, 20], person

image generation [31, 57, 11] and image synthesis from text [48, 49] *etc*. We focus on transforming conditional semantic labels to natural images while adding more diversity and controllability to this task at the semantic level.

Multi-modal label-to-image synthesis. There has been a number of works [1, 28, 40] in multi-modal label-to-image synthesis task. Chen *et. al.* [4] avoided using GAN and leveraged cascaded refinement network to generate high-resolution images. Wang *et. al.* [41] added additional instance-level feature channels to the output of the encoder that allows object-level control on the generated results. Wang *et. al.* [40] used another source of images as stylish examples to guide the generation process. Park *et. al.* [36] incorporated VAE into their network that enables the generator to yield multi-modal images. Li *et. al.* [28] adopted an implicit maximum likelihood estimation framework to alleviate the mode collapse issue of GAN, thus encouraged diverse outputs. Bansal *et. al.* [1] used classic tools to match the shape, context and parts from a gallery with the semantic label input in exponential ways, producing diverse results. Different from these works, we focus on semantically multi-modal image synthesis, which requires fine-grained controllability at the semantic level instead of the global level. Gu *et. al.* [10] built several auto-encoders for each face component to extract different component representations which are then merged into the next foreground generator in the task of portrait editing. Our work is highly related to this work since both methods devise to cope with the SMIS task by treating different classes with different parameters. However, our unique design of progressively decreasing the group numbers in the decoder enables our network to deal with datasets of many classes where their method is possibly incapable of.

Group convolution. Previous works [25, 44, 51, 32, 43] indicate that group convolutions are advantageous for reducing the computational complexity and model parameters, thus they have been widely used in light-weight networks. Ma *et. al.* [32] mentioned that excessive use of group convolutions results in large Memory Access Cost (MAC). Although it is ideal to use group convolutions with small groups or even no group convolutions in the network, we show in our experiments that completely avoiding group convolutions in the decoder is problematic for the performance on the SMIS task. Moreover, our decreasing group numbers strategy considerably alleviates the huge MAC problem so that it is applicable to real-world applications.

3. Semantically multi-modal image synthesis

3.1. Problem definition.

Let M denote a semantic segmentation mask. Suppose there are $\mathcal{C}$ semantic classes in the dataset. H and W represent the image height and width, respectively. As a very

straightforward manner of conducting label-to-image translation, the generator G requires M as conditional input to generate images. However, in order to support multi-modal generation, we need another input source to control generation diversity. Normally, we employ an encoder to extract a latent code Z as the controller inspired by VAE [24]. Upon receiving these two inputs, the image output O can be yield through $O = G(Z, M)$. However, in the semantically multi-modal image synthesis (SMIS) task, we aim to produce semantically diverse images by perturbing the class-specific latent code which independently controls the diversity of its corresponding class.

3.2. Challenge

For the SMIS task, the key is to divide the latent code into a series of class-specific latent codes each of which controls only a specific semantic class generation. The traditional convolutional encoder is not an optimal choice because the feature representations of all classes are internally entangled inside the latent code. Even if we have class-specific latent code, it is still problematic on how to utilize the code. As we will illustrate in the experiment part, simply replacing the original latent code in SPADE [36] with class-specific codes has limited capability to deal with the SMIS task. This phenomenon inspires us that we need to make some architecture modifications in both the encoder and decoder to accomplish the task more effectively.

3.3. GroupDNet

Based on the above analysis, we now give more details about our solution for this task—**Group Decreasing Network** (GroupDNet). The main architecture of GroupDNet takes design inspirations from SPADE [36], considering its superior performance in the label-to-image generation task. A major modification of GroupDNet is the replacement of typical convolutions to group convolutions [25] to achieve class-specific controllability. In the following, we will first present a brief overview of the architecture of our network and then describe the modifications that we made in different components of the network.

Overview. As can be seen from Fig. 2, GroupDNet contains one encoder and one decoder. Inspired by the idea of VAE [24] and SPADE [36], the encoder E produces a latent code Z that is supposed to follow a Gaussian distribution $\mathcal{N}(0, 1)$ during training. While testing, the encoder E is discarded. A randomly sampled code from the Gaussian distribution substitutes for Z. To fulfill this, we use the re-parameterization trick [24] to enable a differentiable loss function during training. Specifically, the encoder predicts a mean vector and a variance vector through two fully connected layers to represent the encoded distribution. The gap between the encoded distribution and Gaussian distribution

can be minimized by imposing a KL-divergence loss:

$$\mathcal{L}_{\mathrm{KL}} = \mathcal{D}_{\mathrm{KL}}(E(I)||\mathcal{N}(0, 1)), \tag{1}$$

where $\mathcal{D}_{\mathrm{KL}}$ represents the KL divergence.

Encoder. Let M_c denote the binary mask for class c and $X \in \mathbb{R}^{H \times W}$ be the input image. By splitting X to different images of different semantic classes, we have

$$X_c = M_c \cdot X. \tag{2}$$

This operation reduces the dependence on E to process feature disentanglement, saving more capacity to precisely encode the feature. The input to the encoder is the concatenation of these images: $S = \underset{c}{\mathrm{cat}} X_c$. All convolutions inside E have the same number of groups, that is, the total number of classes $\mathcal{C}$. From both the input and the architecture side, we decouple different classes to be independent on each other. As a result, the encoded latent code Z is comprised of the class-specific latent code Z_c (a discrete part of Z) of all classes. And Z_c serves as the controller of class c in the forthcoming decoding phase. Different from the general scheme of producing two vectors as the mean and variance prediction of the Gaussian distribution, our encoder produces a mean map and a variance map through convolutional layers to massively retain structural information in the latent code Z.

Decoder. Upon receiving the latent code Z, the decoder transforms it to natural images with the guidance of semantic labels. The question is how to leverage the semantic labels to guide the decoding phase properly. Several ways can serve this purpose, such as concatenating the semantic labels to the input or conditioning on every stage of the decoder. The former one is not suitable for our case because the decoder input has a very limited spatial size that will acutely lose many structural information of the semantic labels. We opt to the latter one and choose a typical advanced model—SPADE generator [36] as the backbone of our network. As mentioned in [36], SPADE is a more general form of some conditional normalization layers [6, 16], and shows superior ability to produce pixel-wise guidance in semantic image synthesis. Following the general idea of using all group convolutions in the generator, we replace the convolutional layers in SPADE module with group convolutions and call this new conditional module as *Conditional Group Normalization* (CG-Norm), as depicted in Fig 2. We then compose a network block called *Conditional Group Block* (CG-Block) by dynamically merging CG-Norm and group convolutions. The architecture of CG-Block is also demonstrated in Fig 2.

Likewise, let $\mathbf{F}^i \in \mathbb{R}^{H^i \times W^i}$ denote the feature maps of the i-th layer of the decoder network and $\mathcal{G}^i$ represent the number of groups of the i-th layer. Moreover, N, D^i, H^i
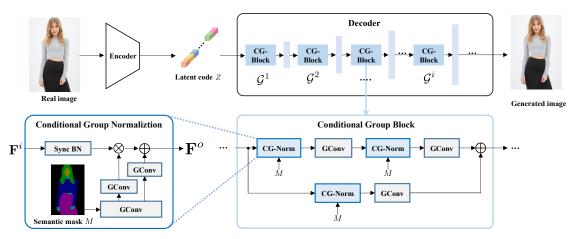
Figure 2: Architecture of our generator (GroupDNet). "GConv" means group convolution and "Sync BN" represents synchronized batch normalization. $\mathcal{G}^i$ is the group number of i-th layer. Note normally $\mathcal{G}^i \geq \mathcal{G}^{i+1}$ for $i \geq 1$ for GroupDNet.

and W^i are the batch size, number of channels, height and width of the feature map, respectively. As demonstrated in Fig. 2, the group convolutions inside CG-Norm will transform the semantic label input to pixel-wise modulation parameters $\gamma \in \mathbb{R}^{D^i \times H^i \times W^i}$ and $\beta \in \mathbb{R}^{D^i \times H^i \times W^i}$. The feature input $\mathbf{F}^i$ will first go through a batch normalization layer [18] that normalizes $\mathbf{F}^i$:

$$\mathrm{BN}(\mathbf{F}^i) = \gamma_{\mathrm{BN}} \left(\frac{\mathbf{F}^i - \mu(\mathbf{F}^i)}{\sigma(\mathbf{F}^i)} \right) + \beta_{\mathrm{BN}}, \qquad (3)$$

where here $\gamma_{\mathrm{BN}}, \beta_{\mathrm{BN}} \in \mathbb{R}^D$ are affine parameters learned from data. μ_d and σ_d are computed across batch size and spatial dimensions for each feature channel:

$$\mu_d(\mathbf{F}^i) = \frac{1}{NH^iW^i} \sum_{n=1}^{N} \sum_{h=1}^{H^i} \sum_{w=1}^{W^i} \mathbf{F}^i_{ndhw}$$

$$\sigma_d(\mathbf{F}^i) = \sqrt{\frac{1}{NH^iW^i} \sum_{n=1}^{N} \sum_{h=1}^{H^i} \sum_{w=1}^{W^i} (\mathbf{F}^i_{ndhw})^2 - (\mu_d(\mathbf{F}^i))^2}$$

$$(4)$$

Afterwards, the output $\mathrm{BN}(\mathbf{F}^i)$ interacts with previously predicted pixel-wise γ and β, yielding a new feature map $\mathbf{F}^o$ with semantic information inserted. Taking Eq. 3 into account,

$$\begin{aligned} \mathbf{F}^o &= \gamma \cdot \mathrm{BN}(\mathbf{F}^i) + \beta \\ &= \gamma \cdot \gamma_{\mathrm{BN}} \left(\frac{\mathbf{F}^i - \mu(\mathbf{F}^i)}{\sigma(\mathbf{F}^i)} \right) + (\gamma \cdot \beta_{\mathrm{BN}} + \beta). \end{aligned} \qquad (5)$$

When i becomes larger, the group number is finally reduced to 1. After a regular convolution, the feature is mapped to a three-channel RGB image O.

3.4. Other solutions

Aside from GroupDNet, a simple solution to perform the SMIS task is to build a set of encoders and decoders, each of which focus on a specific semantic class, as demonstrated in Fig. 3 (a). The underlying idea is to treat each class independently and then fuse the results of different sub-networks. For simplicity, we call such networks as *Multiple Networks* (MulNet). Another alternative with a similar idea is to use group convolution [25] throughout the network. As depicted in Fig. 3 (b), replacing all the convolutions in the encoder and the decoder with group convolutions [25] and setting the group number equal to the class number present the *Group Network* (GroupNet). It is theoretically equivalent to MulNet if the channel number in every group is equal to those of the corresponding layer in a single network of MulNet. Fig. 3 (c) illustrates our GroupDNet. The primary difference between GroupDNet and GroupNet is the monotonically decreasing number of groups in the decoder. Although this modification seems to be simple, it brings several noticeable benefits, mainly in the following three aspects:

Class balance. It is worth noticing that different classes have a different number of instances [30, 5, 53] and require different network capacity to model these classes. It is difficult for MulNet and GroupNet to find a suitable network design to balance all the classes. More importantly, not all the classes appear in one image. In this case, MulNet and GroupNet inevitably waste a lot of computational resources because they have to activate all the sub-networks or sub-groups for all the classes during training or testing. However, in GroupDNet, unbalanced classes share parameters with their neighbor classes, hugely alleviating the class imbalance problem.

Class correlation. In natural worlds, a semantic class usually has relationships with other classes, *e.g.*, the color of grass and the color of tree leaves are similar, and buildings influence the sunshine on the roads in their vicinity, *etc*. To generate plausible results, both of MulNet and GroupNet have a fusion module (several regular convolutions in our
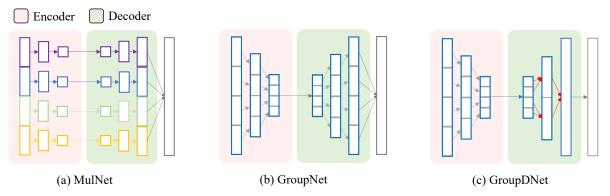
Figure 3: An illustration of MulNet (a), GroupNet (b) and GroupDNet (c). Note the last layers of MulNet and GroupNet are fusing modules that are comprised of several normal convolutional layers to fuse results of different classes.

case) at the end of the decoder to merge features of different classes into one image output. In general, the fusion module roughly considers the correlations of different classes. However, we argue it is not sufficient because the correlation of different classes is too complex to be fully explored by using such a simple component with restricted receptive fields. An alternative is to use some network modules like self-attention block to capture long-range dependencies of the image, but its prohibitive computation hinders its usage in such scenarios [47]. GroupDNet, however, carves these relationships throughout the decoder; hence, it exploits the correlations more accurately and thoroughly. As a result, the generated images of GroupDNet are better and more realistic than those generated by the other two methods.

GPU memory. In order to guarantee that every single network of MulNet, or the grouped parameters for each class in GroupNet have sufficient capacity, the channel numbers in total will increase significantly with the increase of class number. Up to a limit, the maximum GPU memory of a graphics card would no longer be able to hold even one sample. As we roughly estimate on the ADE20K dataset [53], one Tesla V100 graphics card cannot hold the model with sufficient capacity even when batch size is set to 1. However, the problem is less severe in GroupDNet because different classes share parameters, thus it is unnecessary to set so many channels for each class.

3.5. Loss function

We adopt the same loss function as SPADE [36]:

$$\mathcal{L}_{\text{full}} = \arg \min_G \max_D \mathcal{L}_{\text{GAN}} + \lambda_1 \mathcal{L}_{\text{FM}} + \lambda_2 \mathcal{L}_{\text{P}} + \lambda_3 \mathcal{L}_{\text{KL}}. \tag{6}$$

The $\mathcal{L}_{\text{GAN}}$ is the hinge version of GAN loss, and $\mathcal{L}_{\text{FM}}$ is the feature matching loss between the real and synthesized images. Specifically, we use a multiple-layer discriminator to extract features from real and synthesized images. Then, we calculate the L_1 distance between these paired features. Likewise, $\mathcal{L}_{\text{P}}$ is the perceptual loss proposed for style transfer [20]. A pre-trained VGG network [39] is used

to get paired intermediate feature maps, and then we calculate the L_1 distance between these paired maps. $\mathcal{L}_{\text{KL}}$ is the KL-divergence loss term as Eq. 1. We set $\lambda_1 = 10, \lambda_2 = 10, \lambda_3 = 0.05$, the same as SPADE [36].

4. Experiments

4.1. Implementation details

We apply Spectral Normalization [35] to all the layers in both the generator and discriminator. The learning rates for the generator and discriminator are set to 0.0001 and 0.0004, respectively, following the two time-scale update rule [12]. We use the Adam optimizer [23] and set $\beta_1 = 0, \beta_2 = 0.9$. All the experiments are conducted on at least 4 P40 GPUs. Besides, we use synchronized batch normalization to synchronize the mean and variance statistics across multiple GPUs. More details, such as the detailed network design and more hyper-parameters, are given in the supplementary materials.

4.2. Datasets

We conduct experiments on three very challenging datasets, including DeepFashion [30], Cityscapes [5], and ADE20K [53]. We choose DeepFashion because this dataset shows lots of diversities among all semantic classes, which is naturally suitable for assessing the model's ability to conduct multi-modal synthesis. Consequently, we compare with several baseline models on this dataset to evaluate the superior power of our model on the SMIS task. The size of the images in Cityscapes are quite large, so it is proper to test the model's ability to produce high-resolution images on this dataset. ADE20K is extremely challenging for its massive number of classes, and we find it hard to train MulNet and GroupNet on ADE20K with our limited GPUs. More details can be found in the supplementary materials.

4.3. Metrics

Mean SMIS Diversity. In order to evaluate the performance of a model designed for the SMIS task, we intro-

duce two new metrics named: *mean Class-Specific Diversity* (mCSD) and *mean Other-Classes Diversity* (mOCD). We design the new metrics based on the LPIPS metric [50], which is used to assess the generation diversity of a model by computing the weighted $\mathcal{L}_2$ distance between deep features of image pairs. For the same semantic label input, we generate n images for each semantic class by only modulating the latent code Z_c for the semantic class c. Therefore, we have a set of images $S = \{I_1^1, ..., I_1^n, ..., I_{\mathcal{C}}^1, ..., I_{\mathcal{C}}^n\}$. Finally, mCSD and mOCD are calculated by

$$\text{mCSD} = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} L_c, \ \text{mOCD} = \frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} L_{\neq c}. \quad (7)$$

where L_c is the average LPIPS distance [50] of the semantic area of class c between sampled m pairs and $L_{\neq c}$ represents the average LPIPS distance [50] in the areas of all other classes between the same pairs. In our settings, we set $n = 100, m = 19$ following [56, 17]. ImageNet pre-trained AlexNet [25] is served as the deep feature extractor. Higher performance on the SMIS task demands a high diversity of the specific semantic areas (high mCSD) as well as a low diversity of all other areas (low mOCD). Besides, we also report the overall LPIPS distance by producing globally diverse results for the same semantic labels.

Human Evaluation Metrics. We further introduce human evaluation to evaluate whether the generative model performs well in the SMIS task. We recruit 20 volunteers that have research experience in generation tasks. We show them an input mask along with two images generated from only one model. The two images are multi-modal results which only vary in the areas of one random semantic class. The volunteers judge whether the given two images only vary in one semantic class. The percentage of pairs that are judged to be semantically different in only one semantic class represents the human evaluation of a model's performance on the SMIS task. We abbreviate this metric as **SHE** (**S**MIS **H**uman **E**valuation). For each phase, the volunteers are given 50 questions of unlimited answering time.

Fréchet Inception Distance. We use Fréchet Inception Distance (FID) [13] to calculate the distance between the distributions of synthesized results and the distribution of real images. Lower FID generally hints better fidelity of the generated images.

Segmentation Performance. It is reasonable that the predicted labels of the generated images are highly similar to those of the original images if they look realistic. Therefore, we adopt the evaluation protocol from previous work [4, 41, 36] to measure the segmentation accuracy of the generated images. We report results on the mean Intersection-over-Union (mIoU) and pixel accuracy (Acc) metrics without considering the classes that can be ignored. Images

Figure 4: Qualitative comparison between GroupDNet and other baseline models. The first two rows represent the results of different models by changing their upper-clothes latent code while the last two rows represent their results of changing the pants latent code. Note, for those models which have no class-specific controller such as VSPADE, we alter their overall latent codes to generate different images.

are evaluated using well-trained segmentation models UperNet101 [42] for ADE20K, DRN-D-105 [46] for Cityscapes, off-the-shelf human parser CIHP [8] for DeepFashion.

4.4. Results

Besides the following sections, we have more justification of our model design in the supplementary materials for the reference of interested readers.

4.4.1 Comparison on SMIS

A basic requirement for models that potentially could be modified for the SMIS task is that they should possess the ability to conduct multi-modal image synthesis. We compare with several methods that support multi-modal image synthesis to demonstrate the superiority of GroupDNet:

- **Variational SPADE [36]** (**VSPADE**) has an image encoder processing a real image to a mean and a variance vector where a KL divergence loss is applied to support multi-modal image synthesis. Detailed description can be found in their paper;

- **BicycleGAN [56]** maps the given image input into a latent code, which is later combined with a label input to produce outputs. Since the latent code is constrained by a KL divergence loss, it could be substituted by a random sample from the Gaussian distribution;

- **DSCGAN [45]** is complementary to BicycleGAN by introducing an explicit regularization upon the generator, trying to alleviate the mode collapse issue of previous models.

Aside from MulNet and GroupNet described in Sec. 3.4, we also conduct two further experiments by replacing the convolutions in the encoder/decoder of the VSPADE model to group convolutions with group numbers set equal to the dataset class number, denoted as **GroupEnc/GroupDec**, respectively. Note MulNet, GroupNet, GroupEnc, GroupDec

and VSPADE are trained with the same kind of multi-scale discriminator [41] and training settings as GroupDNet. To fairly compare the performances, we balance the number of parameters of these models to mitigate the suspicion that the performance improvements are brought by using more parameters. For BicycleGAN and DSCGAN, we adopt their original training and testing protocols.

Quantitative and qualitative results are given in Tab. 1 and Fig. 4, respectively. The quantitative results demonstrate the overall superiority of GroupDNet. Generally, GroupDNet exhibits the best image quality (lowest FID) and overall diversity (highest LPIPS). In terms of the performance on the SMIS task, MulNet and GroupNet are slightly better than GroupDNet, given evidence that they have either larger mCSD or lower mOCD. However, the image quality of MulNet and GroupNet is not satisfactory (high FID) and MulNet shows much lower FPS than GroupDNet. In terms of the SHE metric, GroupDNet is also very competitive to MulNet and GroupNet. Although VSPADE, has rather large mCSD, its mOCD is also very large, indicating that it performs unsatisfactorily on the SMIS task. The same phenomenon is also observed in BicycleGAN and DSCGAN and their FIDs are relatively much higher than VSPADE, showing the advantage of the SPADE architecture. From the high mOCD values of VSPADE and GroupDec, whose encoders are composed from regular convolutions, we conclude that group encoder serves as a key to the high performance of the SMIS task. However, the exceptional performance of GroupDNet suggests that the group decreasing modification in the decoder is also effective and brings further performance boost when compared to GroupEnc. Gathering these information, GroupDNet is a good trade-off model considering the speed, visual quality and the performance on the SMIS task.

According to the qualitative results, it is obvious that MulNet, GroupNet, GroupEnc and GroupDNet are able to generate semantically multi-modal images while others cannot. However, the image quality of MulNet, GroupNet, BicycleGAN and DSCGAN is far from satisfaction because their images are visually implausible. GroupEnc is better in image quality but it degrades in the SMIS task. It can be seen from the first two rows in Fig. 4 that, when the upper clothes are changed to another style, GroupEnc slightly changes the color of the short jeans pants as well.

4.4.2 Comparison on label-to-image translation

In this section, we mainly assess the generated image quality of our method by comparing with some label-to-image methods on the FID, mIoU and Accuracy metrics. We choose four very recent state-of-the-art methods: BicycleGAN [56], DSCGAN [45], pix2pixHD [41] and SPADE [36], as the comparison methods. Compar-

Models	FID↓	mCSD↑	mOCD↓	LPIPS↑	SHE↑	Speed↑	# Param↓
MulNet	12.07	0.0244	0.0019	0.202	79.2	6.3	105.1
GroupNet	12.58	0.0276	**0.0017**	0.203	**83.7**	8.2	97.7
Group Enc	10.83	0.0232	0.0065	0.217	69.3	19.6	105.5
Group Dec	9.84	0.0003	0.0257	0.206	26.4	12.1	111.3
VSPADE [36]	10.02	0.0304	0.1843	0.207	23.6	20.4	106.8
BicycleGAN [56]	40.07	**0.0316**	0.2147	**0.228**	24.8	66.9	**58.4**
DSCGAN [45]	38.40	0.0245	0.1560	0.163	27.6	**67.2**	**58.4**
GroupDNet	**9.50**	0.0264	0.0033	**0.228**	81.2	12.2	109.1

Table 1: Quantitative comparison results with baseline models. "SHE" means human evaluation of a model's performance on SMIS task. We use Frame Per Second (FPS) to represent the "Speed" of the model. "# Param" means the number of parameters, whose unit is "M", denoting million. For mCSD, the higher, the better. For mOCD, the lower, the better.
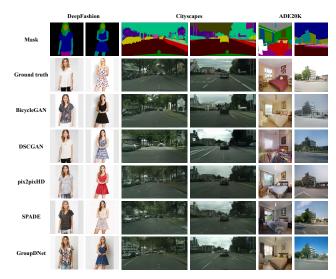

Figure 5: Qualitative comparison with the SOTA label-to-image methods. From top to bottom, the images represent the experiments on DeepFashion, Cityscapes and ADE20K, respectively.

isons are performed across the DeepFashion, Cityscapes and ADE20K datasets. We evaluate the performance of their well-trained models downloaded from their official GitHub repositories if they have. For those experiments not included in their original papers, we follow their codes and run the experiments with similar settings to GroupDNet. Quantitative results are shown in Tab. 2. In general, as our network is built based on SPADE, it maintains nearly the same performance as SPADE on the DeepFashion and Cityscapes datasets. While on the ADE20K dataset, our method is inferior to SPADE but still outperform other methods. This phenomenon on the one hand shows the superiority of the SPADE architecture and on the other hand also exposes even GroupDNet still struggles to handle datasets with a huge number of semantic classes.

Qualitative comparisons on DeepFashion, Cityscapes and ADE20K are shown in Fig. 5. In general, the images generated by GroupDNet are more realistic and plausible than others. These visual results consistently show the high image quality of GroupDNet's generated images, verifying its efficacy on various datasets.

(a) Semantically multi-modal image synthesis

(b) Appearance mixture

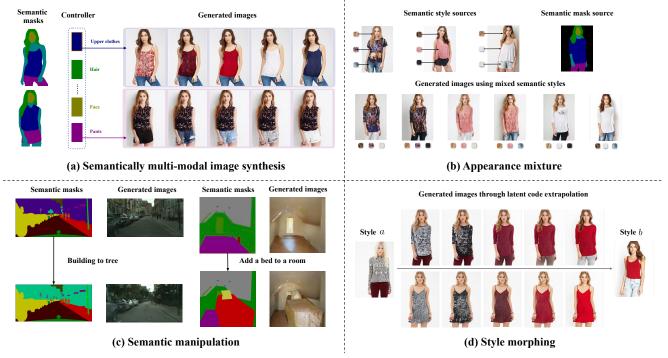**(c) Semantic manipulation**

(d) Style morphing

Figure 6: Exemplar applications of the proposed method. (a) Demonstration of the semantically multi-modal image synthesis (SMIS) task. (b) Application of our SMIS model in appearance mixture. Our model extracts styles of different semantic classes from different sources and generates a mixed image by combining these semantic styles with the given semantic mask. (c) Application of our SMIS model in semantic manipulation. (d) Application of our SMIS model in image extrapolation. **Zoom in** for better details.

Method	DeepFashion			Cityscapes			ADE20K		
	mIoU↑	Acc↑	FID↓	mIoU↑	Acc↑	FID↓	mIoU↑	Acc↑	FID↓
BicycleGAN [56]	76.8	97.8	40.07	23.3	75.4	87.74	4.78	29.6	87.85
DSCGAN [45]	81.0	98.3	38.40	37.8	86.7	67.77	10.2	58.8	83.98
pix2pixHD [41]	85.2	98.8	17.76	58.3	92.5	78.24	27.6	75.7	55.9
SPADE [36]	87.1	**98.9**	10.02	**62.3**	93.5	58.10	**42.0**	**81.4**	**33.49**
GroupDNet	**87.3**	**98.9**	**9.50**	**62.3**	**93.7**	49.81	30.4	77.1	42.17

Table 2: Quantitative comparison with label-to-image models. The numbers of pix2pixHD and SPADE are collected by running the evaluation on our machine instead of their papers.

4.4.3 Applications

Since GroupDNet contributes more user controllability to the generation process, it can also be applied to lots of exciting applications in addition to the SMIS task, which are demonstrated as follows. More results are available in the supplementary materials.

Appearance mixture. By utilizing the encoder in GroupDNet during inference, we can gather the distinct styles of a person's different body parts. Every combination of these styles presents a distinct person image, given a human parsing mask. In this way, we can create thousands of diverse and realistic person images given a person image gallery. This application is demonstrated in Fig. 6(b). A demo video can be found in our code repository.

Semantic manipulation. Similar to most label-to-image methods [41, 36, 28], our network also supports semantic manipulation. As exemplified in Fig. 6 (c), we can insert a bed in the room or replace the building with trees, *etc*.

Style morphing. Feeding two real images to the encoder generates two style codes of these images. By extrapolating between these two codes, we can generate a sequence of images that progressively vary from image a to image b,

depicted in Fig. 6 (d).

5. Conclusion and future work

In this paper, we propose a novel network for semantically multi-modal synthesis task, called GroupDNet. Our network unconventionally adopts all group convolutions and modifies the group numbers of the convolutions to decrease in the decoder, considerably improving the training efficiency over other possible solutions like multiple generators.

Although GroupDNet performs well on semantically multi-modal synthesis task and generates results with relatively high quality, there are still some problems remained to be solved. First, it requires more computational resources to train compared to pix2pixHD and SPADE though it is nearly 2 times faster than multiple generators networks. Second, GroupDNet still has difficulty in modeling different layouts of a specific semantic class for datasets with limited diversity, even though it demonstrates some low-level variations like illumination, color, and texture, *etc*.

Acknowledgment

References

[1] Aayush Bansal, Yaser Sheikh, and Deva Ramanan. Shapes and context: In-the-wild image synthesis & manipulation. In *Proc. CVPR*, pages 2317–2326, 2019.

[2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. CVAE-GAN: fine-grained image generation through asymmetric training. In *Proc. ICCV*, pages 2764–2773, 2017.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. ICLR*, 2019.

[4] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. ICCV*, pages 1520–1529, 2017.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, pages 3213–3223, 2016.

[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *Proc. ICLR*, 2017.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, June 2016.

[8] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *Proc. ECCV*, pages 805–822, 2018.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, pages 2672–2680, 2014.

[10] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *Proc. CVPR*, pages 3436–3445, 2019.

[11] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R. Scott, and Larry S. Davis. Compatible and diverse fashion image inpainting. In *Proc. ICCV*, 2019.

[12] 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 *Proc. NeurIPS*, pages 6626–6637, 2017.

[13] 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 *Proc. NeurIPS*, pages 6626–6637, 2017.

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proc.*, pages 1994–2003, 2018.

[15] Seunghoon Hong, Xinchen Yan, Thomas S. Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Proc. NeurIPS*, pages 2713–2723, 2018.

[16] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. ICCV*, pages 1510–1519, 2017.

[17] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proc. ECCV*, pages 179–196, 2018.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, pages 448–456, 2015.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, pages 5967–5976, 2017.

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711, 2016.

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. ICLR*, 2018.

[22] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, pages 4401–4410, 2019.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

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

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NeurIPS*, pages 1106–1114, 2012.

[26] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. CVPR*, pages 105–114, 2017.

[27] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proc. ECCV*, pages 36–52, 2018.

[28] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional IMLE. *CoRR*, abs/1811.12373, 2018.

[29] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proc. NeurIPS*, pages 700–708, 2017.

[30] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proc. CVPR*, 2016.

[31] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Proc. NIPS*, pages 405–415, 2017.

[32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In *Proc. ECCV*, pages 122–138, 2018.

[33] Jiayuan Mao, Xiuming Zhang, Yikai Li, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Program-guided image manipulators. In *Proc. ICCV*, 2019.

[34] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

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

[36] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. CVPR*, pages 2337–2346, 2019.

[37] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proc. ICCV*, 2019.

[38] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural image. In *Proc. ICCV*, 2019.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[40] Miao Wang, Guo-Ye Yang, Ruilong Li, Runze Liang, Song-Hai Zhang, Peter M. Hall, and Shi-Min Hu. Example-guided style-consistent image synthesis from semantic labeling. In *Proc. CVPR*, pages 1495–1504, 2019.

[41] 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 *Proc. CVPR*, pages 8798–8807, 2018.

[42] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proc. ECCV*, pages 432–448, 2018.

[43] Guotian Xie, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. Interleaved structured sparse convolutional neural networks. In *Proc. CVPR*, pages 8847–8856, 2018.

[44] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, pages 5987–5995, 2017.

[45] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *Proc. ICLR*, 2019.

[46] Fisher Yu, Vladlen Koltun, and Thomas A. Funkhouser. Dilated residual networks. In *Proc. CVPR*, pages 636–644, 2017.

[47] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proc. ICML*, pages 7354–7363, 2019.

[48] Han Zhang, Tao Xu, and Hongsheng Li. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. ICCV*, pages 5908–5916, 2017.

[49] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1947–1962, 2019.

[50] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, pages 586–595, 2018.

[51] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proc. CVPR*, pages 6848–6856, 2018.

[52] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proc. ICCV*, pages 2039–2049, 2017.

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proc. CVPR*, pages 5122–5130, 2017.

[54] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Trans. Graph.*, 37(4):49:1–49:13, 2018.

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, pages 2242–2251, 2017.

[56] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proc. NeurIPS*, pages 465–476, 2017.

[57] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proc. CVPR*, pages 2347–2356, 2019.