Interpreting Class Conditional GANs with Channel Awareness

Yingqing He† Zhiyi Zhang† Jiapeng Zhu† Yujun Shen† Qifeng Chen†
†HKUST ‡ByteDance Inc.

Fig. 1. Novel applications enabled by interpreting class conditional GANs. Given a conditional generator in (a), we propose channel awareness to quantify the contribution of each feature channel to the output image, as shown in (b), which reveals how the categorical information is handled by different channels. Red, green, and blue channels are primarily responsible for the synthesis of a particular category, while yellow ones are shared by all classes. (c) Such an interpretation facilitates a range of applications, including single-channel image editing, category hybridization, fine-grained semantic segmentation, and category-wise synthesis performance evaluation.

Abstract

Understanding the mechanism of generative adversarial networks (GANs) helps us better use GANs for downstream applications. Existing efforts mainly target interpreting unconditional models, leaving it less explored how a conditional GAN learns to render images regarding various categories. This work fills in this gap by investigating how a class conditional generator unifies the synthesis of multiple classes. For this purpose, we dive into the widely used class-conditional batch normalization (CCBN), and observe that each feature channel is activated at varying degrees given different categorical embeddings. To describe such a phenomenon, we propose channel awareness, which quantitatively characterizes how a single channel contributes to the final synthesis. Extensive evaluations and analyses on the BigGAN model pre-trained on ImageNet reveal that only a subset of channels is primarily responsible for the generation of a particular category, similar categories (e.g., cat and dog) usually get related to some same channels, and some channels turn out to share information across all classes. For good measure, our algorithm enables several novel applications with conditional GANs. Concretely, we achieve (1) versatile image editing via simply altering a single channel and manage to (2) harmoniously hybridize two different classes. We further verify that the proposed channel awareness shows promising potential in (3) segmenting the synthesized image and (4) evaluating the category-wise synthesis performance.¹

¹Project page can be found here.
1 Introduction

The past few years have witnessed the rapid advancement of generative adversarial networks (GANs) in image synthesis [6,11,22–26]. Despite the wide range of applications powered by GANs, like image-to-image translation [19,41,69], super-resolution [7,31,35,57], and image editing [4,30,33,42,43], it typically requires learning a separate model for a new task, which can be time and resources consuming. Some recent studies have confirmed that a well-trained GAN model naturally supports various downstream applications [7,12,14,40,60,61,65], benefiting from the rich knowledge learned in the training process [3,5,51]. Therefore, to make sufficient use of a GAN, it becomes crucial to explore and further exploit its internal knowledge.

Many attempts have been made to understand the generation mechanism of GANs [5,20,50,51,58,60,62]. It is revealed that, to produce a fair synthesis, the generator is required to render multi-level semantics, such as the overall attributes (e.g., the gender of a face image) [20,51], the objects inside (e.g., the bed in a bedroom image) [5,62], the part-whole organization (e.g., the segmentation of the synthesis) [60,65], etc. However, existing efforts mainly focus on interpreting unconditional GANs, leaving conditional generation as a black box.

Compared with unconditional models, a class conditional model is more informative and efficient in that it unifies the synthesis of multiple categories, like animals, vehicles, and scenes [6]. Figuring out how it manages the class information owns much great potential yet rarely explored. To fill in this gap, we take a close look at the popular class-conditional batch normalization (CCBN) [6,21,36,37,64], which is one of the core modules distinguishing conditional generators from unconditional ones. Concretely, CCBN learns category-specific parameters to scale and shift the input features, such that the output features developed with different class embeddings can be easily told apart from each other, eventually resulting in the synthesis of various categories. We notice from such a process that, under the perspective of the ReLU activation [38] following CCBN, different feature channels present varying behaviors given different embeddings.

To quantify the aforementioned channel effect, we propose channel awareness that characterizes how a single channel contributes to the final synthesis. Through in-depth analyses on the BigGAN [6] model pre-trained on ImageNet [9], we have the following key findings, which are also illustrated in Fig. 1b. First, only a portion of channels are active in rendering images for a particular class while the remaining channels barely affect the generation. Second, more similar categories tend to share more relevant channels. For instance, channels regarding dog synthesis intersect with those of cats but disjoint from those of buses. Third, some channels highly response to the latent code instead of the class embedding and hence appear to deliver knowledge to all classes.

Beyond model interpretation, our proposed channel awareness facilitates a range of novel applications with class conditional GANs, as shown in Fig. 1c. First, after identifying the relevant channels through awareness ranking, we realize versatile image editing by simply altering a single feature channel (Sec. 4.1). Second, through mixing the channels that are related to two classes respectively, we achieve harmonious category hybridization (Sec. 4.2). Third, we verify that intermediate feature maps from the generator, after weighted by our channel awareness, can be convincingly used for fine-grained semantic segmentation (Sec. 4.3). Fourth, we empirically demonstrate the potential of our channel awareness in evaluating the category-wise synthesis performance (Sec. 4.4).

1.1 Related Work

Among various types of generative models, such as variational auto-encoder (VAEs) [16,28,46,55], flow-based model [27,47], diffusion model [10,17], etc., GAN [11] has received wide attention due to its impressive performance on both unconditional synthesis [22,24–26] and conditional synthesis [6,36,49,64]. Early studies on interpreting GANs [5,8,51,62] suggest that, a well-learned GAN generator has encoded rich knowledge that can be promising applied to various downstream tasks, including attribute editing [2,3,20,33,62,63,67], image processing [12,18,40,48,68], super-resolution [7,35], image classification [61], semantic segmentation [1,32,54,60,65], and visual alignment [44]. Existing interpretation approaches usually focus on the relationship between the latent space and the image space [51,58,62,66], hence commonly evaluated on unconditional models. Some attempts are made to also analyze class conditional models [13,20,45,52,56], but they still target the latent space, leaving it unclear how the generator leverages the categorical information. This work clearly differs from prior arts from the following aspects. (1) We inspect the conditional generator from the channel perspective, which aggregates the messages from both the latent code and the class embedding. To our knowledge, this is the first attempt on understanding the function of embedding space in conditional generation. (2) We demonstrate the editability of altering a single channel of the conditional generator. Different from the single-channel editing in unconditional GANs [58], our approach identifies different relevant channels with respect to different categories in an unsupervised manner. (3) We achieve fine-grained semantic segmentation by paying more attention to some “important” channels. Unlike the existing efforts on single-object generation [60,65], our method does not require data-driven learning and can be robustly generalized to all classes. (4) We also enable some applications that are peculiar to conditional models, including the category hybridization and the category-wise synthesis performance evaluation.
input feature maps, as CCBN learns class-specific parameters to scale and shift the batch normalization (CCBN) \([6, 21, 36, 37, 64]\). In particular, where the most popular one is to adopt the class-conditional images for that category.

Given a different embedding, the generator is able to produce are the output image and the generator, respectively. That way, the process can be formulated as

\[
\mathbf{y}^c = \gamma^c \mathbf{x}^c + \beta^c,
\]

where \(\mathbf{x}^c\) and \(\mathbf{y}^c\), both with shape \(1 \times H \times W\), denote the normalized input (i.e., subtracting mean and dividing by standard deviation) and the output. \(\gamma^c\) and \(\beta^c\) are scalars for the \(c\)-th channel. Here, \(e\) stands for the embedding index, carrying the categorical information, and the effect of \(z\) is omitted for simplicity.

CCBN is usually followed by ReLU \([6, 64]\). Such an activation controls the information flow in that negative values in \(\mathbf{y}^c\) are cut off to zero. According to Eq. (2), it is equivalent to cutting off the values in \(\mathbf{x}^c\) that are smaller than

\[
t_c = -\frac{\beta^c}{\gamma^c}.
\]

In this way, we manage to directly relate the output feature channel to its corresponding input channel, as shown in Fig. 2. The value \(t_c\) acts like a channel probe since it measures the channel-wise activation of a particular synthesis. Generally, for a certain channel, it presents different behaviours (i.e., with different \(t_c\) values) for different samples and different categories.

**Category-oriented Channel Awareness.** As discussed above, given a well-trained generator with CCBN, \(t_c\) is strictly determined by the latent code, \(z\), and the class embedding, \(e\). Recall that the channel probe \(t_c\) is instance-aware, which

![Fig. 2. Concept diagram of channel probe, from whose statistics we develop the channel awareness. For a particular feature channel, it is first transformed by class-conditional batch normalization (CCBN) with category-specific scale \(\gamma^c\), and bias \(\beta^c\), then activated by ReLU to filter out negative neurons. Such a process is reformulated as a combined operation (as shown at the bottom), where we define the channel probe as \(t = -\frac{\beta^c}{\gamma^c}\). This value reflects the integrated action of CCBN and ReLU.](image)

### 2 Methodology

In this section, we introduce the proposed channel awareness. Specifically, we re-examine the class-conditional batch normalization (CCBN), which is widely used in class conditional generation \([6, 21, 36, 37, 64]\), and investigate how it helps the generator with the categorical information provided by the class embedding. It is noteworthy that our approach is fully unsupervised, without relying on any segmentation masks or annotations.

#### 2.1 Preliminaries

Unlike unconditional GANs \([11, 22, 25]\) where the generator takes the latent code \(z\), as the only input, a class conditional generator employs an additional embedding vector, \(e\), to provide the categorical information. Accordingly, the generation process can be formulated as \(\mathbf{I} = G(z, e)\), where \(\mathbf{I}\) and \(G(\cdot, \cdot)\) are the output image and the generator, respectively. That way, given a different embedding, the generator is able to produce images for that category.

There are many ways of integrating \(e\) into \(G(\cdot, \cdot)\) \([6, 25]\), where the most popular one is to adopt the class-conditional batch normalization (CCBN) \([6, 21, 36, 37, 64]\). In particular, CCBN learns class-specific parameters to scale and shift the input feature maps, as

\[
\mathbf{y} = \gamma(\text{concat}(z, e)) \odot \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})} + \beta(\text{concat}(z, e)),
\]

where \(\mathbf{x}\) and \(\mathbf{y}\), both with shape \(C \times H \times W\), denote the input and output features. \(\mu(\cdot)\) and \(\sigma(\cdot)\) compute the mean and variance of a tensor along the spatial dimensions (i.e., \(H\) and \(W\)), \(\text{concat}(\cdot, \cdot)\) stands for the concatenation operation. \(\gamma(\cdot)\) and \(\beta(\cdot)\) outputs the \(C\)-dimensional scale and bias by learning from both \(z\) and \(e\). \(\odot\) represents the element-wise multiplication with broadcasting.

#### 2.2 Channel Awareness

**Channel Probe.** From Eq. (1), we can tell that both the latent code and the class embedding act on the generation through CCBN. In other words, their messages are delivered to the feature channels through the learning of \(\gamma(\cdot)\) and \(\beta(\cdot)\). Now, we take a look at how each single channel contributes to the synthesis. For a particular channel with index \(c\), Eq. (1) can be simplified as

\[
\mathbf{y}^c = \gamma^c \mathbf{x}^c + \beta^c,
\]

where \(\mathbf{x}^c\) and \(\mathbf{y}^c\), both with shape \(1 \times H \times W\), denote the normalized input (i.e., subtracting mean and dividing by standard deviation) and the output. \(\gamma^c\) and \(\beta^c\) are scalars for the \(c\)-th channel. Here, \(e\) stands for the embedding index, carrying the categorical information, and the effect of \(z\) is omitted for simplicity.

In this way, we manage to directly relate the output feature channel to its corresponding input channel, as shown in Fig. 2. The value \(t_c\) acts like a channel probe since it measures the channel-wise activation of a particular synthesis. Generally, for a certain channel, it presents different behaviours (i.e., with different \(t_c\) values) for different samples and different categories.

**Category-oriented Channel Awareness.** As discussed above, given a well-trained generator with CCBN, \(t_c\) is strictly determined by the latent code, \(z\), and the class embedding, \(e\). Recall that the channel probe \(t_c\) is instance-aware, which
fluctuates along with the synthesis varying. To get a more reliable understanding of the function of a single feature channel, we derive channel awareness from the statistics of $t^e_c$. Oriented to the class embedding, $e$, we would like to eliminate the impacts caused by the randomness of $z$. To this end, we first sample a number of latent codes with the embedding fixed, then calculate the $t^e_c$ value for each synthesis and perform averaging. Recall that a lower $t^e_c$ suggests that more information will be preserved along this channel. Therefore, channels with lower mean values are more likely to be used for generating the target class. We define category-oriented channel awareness as $-\mathbb{E}_z[t^e_c]$.

Latent-oriented Channel Awareness. We further study the contribution of the latent code, $z$, to each channel. Similarly, we sample a collection of latent codes with a fixed embedding and calculate $t^e_c$. Differently, this time we are interested in the variance instead of the mean, which gives us the latent-oriented channel awareness. A larger variance indicates that the randomness has a stronger influence on the synthesis regarding this channel. Hence, the latent-oriented channel awareness, $\text{Var}(t^e_c)$, reflects how sensitively a channel reacts to the latent code with the given embedding.

3 Evaluation and Analysis

In this section, we evaluate and analyze the proposed channel awareness of category and latent in Sec. 3.1 and Sec. 3.2, respectively. We exploit the BigGAN [6] conditionally trained on the large-scale ImageNet [9] as the target model, which can generate realistic images of 1,000 various categories at 256 × 256 resolution.

3.1 Category-oriented Channel Awareness

Qualitative Evaluation. We measure the causal channel effect on the image generation via forcing the target channel to zero during the forward process [5]. Then we compare the results after zeroing out of three groups of channels with different awareness scores in Fig. 3.

We noticeably observe that: Zeroing out channels with the highest class awareness removes essential information of that class (e.g., fur type of Japanese Spaniel, stem of mushroom, and structure of castle), while masking channels with the lowest class awareness only causes negligible changes, and randomly selected channels have the in-between effect. This phenomenon not only shows the effectiveness of our method but also demonstrates our key finding that channels are not equally contributing to the generation of a particular class: Only partial channels (i.e., with highest class awareness) primarily contribute to it, and some channels barely affect its generation (i.e., with lowest class awareness).

Quantitative Evaluation. We further quantitatively verify the category-oriented channel awareness regarding every single channel in all 1,000 classes. We utilize a pre-trained classifier as an channel effect evaluator by assessing generated class images with and without a certain channel. Specifically, we measure the classification score drop between generated image with zeroing out a channel and the original one, which we refer to as channel-classifier response. We report the correlation coefficient between the channel-classifier response and the proposed category-oriented channel awareness. High correlation indicates the effectiveness of our method.

Implementation details. We use the Inception-v3 [53] model trained on ImageNet [9] as the evaluation classifier, a widely used classifier for assessing generated images [15,29].
For each channel, we calculate its channel-classifier response among 1,000 generated sample pairs (a sample pair means the generated images before and after the intervention of single-channel masking). To reduce data noise, we smooth the raw data of channel-classifier response via the commonly-used Savitzky-Golay filter [34] with a window length 51 (It is a tiny length of the filter since the full number of data points is 1,536).

We show four classes with the highest correlation in Fig. 4. The trend of channel-classifier response is highly aligned with that of category-oriented channel awareness. We also calculate the average correlation for all 1,000 classes, even including categories that BigGAN does not synthesize well, and obtain 0.503. Furthermore, we evaluate our method on BigGAN-deep [6] regarding 1,000 classes, and obtain an average correlation of 0.603, even better than our base model. Details can be found in Appendix A. This result indicates the effectiveness of our metric regarding all classes as well as different models.

**Analysis via channel overlap estimation.** From above, we verify that channels with high awareness of a given class are primarily responsible for generating that particular category. Next, we wonder how distinct those channels are regarding different categories, and do similar categories share several channels? Answering this question could help us deeply understand how a conditional GAN organizes characteristics from numerous distinct categories. Thus, we further analyze the channel overlap among them. We estimate the channel overlap of two classes as \(|S_i \cap S_j|_k\), where \(S_i\) and \(S_j\) represent the set of channel indices with top k highest awareness of class \(i\) and \(j\), \(\cap\) denotes the intersection operation, and \(k = |S_i| = |S_j|\), which is the number of selected channels. For the given two classes, higher channel overlap indicates that more channels are responsible for generating both of them, and zero channel overlap means they own two separate sets of channels inside the network.

The results of channel overlap of different categories are shown in Fig. 5a. We can observe that: 1) Channels with high class awareness are different between different classes since their overlaps are less than 1. 2) Several channels are overlapped among similar classes (see the three \(3 \times 3\) blocks along the diagonal). 3) For two largely distinct classes, such as cars with dogs and birds, their channel overlaps are close to zero, meaning that their corresponding channels are almost disjoint. These observations indicate that different classes own different sets of relevant channels, and similar classes own several mutual relevant channels.

**Analysis via single-channel manipulation.** Furthermore, we discover that altering a single channel of the highest awareness can manipulate the class-relevant semantics which results are shown in Fig. 5b (left two columns), such as the pointed ear of class Chihuahua and the long fur of class Japanese Spaniel. Meanwhile, when applying the semantic found by one source class to other target classes, we found that this can only manipulate similar classes (red box), or other classes that exhibit the corresponding semantic (blue box), while does not affect other irrelevant classes. For example, altering the semantic of water area (last row) found by the class of speedboat can manipulate the promontory class yet not affect animal classes, same as other semantics.

### 3.2 Latent-oriented Channel Awareness

**Discovery of class-shared channels.** From above, we show that channels with high category-oriented channel awareness are class-distinct thus responsible for class-distinct semantics. We also aim to inspect if there exist channels that took charge of common characteristics for all 1,000 classes, which we refer to as class-shared channels. To answer this question, we perform channel intersection among all classes to check whether there has left channels. Specifically, we select class-specific channels of 1) highest awareness to the class, 2) lowest awareness to the class, and 3) highest awareness to latent, and then perform intersection successively. The channel intersection results of these three settings are shown in Tab. 1. We can discover that after intersecting among 1,000 classes, there is no channel left in the first two settings. However, we noticeably identified that channels that highly respond to latent exist in all 1,000 classes. This implies that a class-conditional generator can learn unified representations provided to all classes. Besides, these class-shared channels
Fig. 5. Analysis on category-oriented channel awareness. (a) Channel overlap between different categories (including dogs, birds, and vehicles), where a higher overlap indicates two classes sharing more relevant channels. The overlap is computed using 100 channels with the highest awareness for each category. (b) The semantic discovered from the source class is applied to other target classes. We observe that semantics are shared between similar categories, like cat and dog.

Tab. 1. Cross-category analysis. Regarding the 1,000 classes in ImageNet [9], we select only 10 channels for each class according to different criteria, and find the common channels through intersection. Results on all layers suggest that our latent-oriented channel awareness helps identify channels that can deliver information to all classes.

| Criterion                          | L0 | L1 | L2 | L3 | L4 | L5 | L6 | L7 | L8 | L9 | L10 | L11 |
|-----------------------------------|----|----|----|----|----|----|----|----|----|----|-----|-----|
| highest awareness w.r.t. category | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   |
| lowest awareness w.r.t. category  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   |
| highest awareness w.r.t. latent   | 3  | 3  | 3  | 1  | 3  | 3  | 0  | 4  | 2  | 1  | 5   | 4   |

can further enable semantic manipulation with highly similar effect on disparate classes. Visual editing results with respect to these channels are provided in Fig. 7.

4 Applications with Class Conditional GANs

In this section, we demonstrate four novel applications enabled by our channel awareness. (1) Versatile image editing for category-oriented attributes and latent-oriented attributes (Sec. 4.1). (2) Category hybridization that smoothly blends attributes from two different categories (Sec. 4.2). (3) Fine-grained Segmentation on the synthesized image (Sec. 4.3). (4) Category-wise synthesis evaluation from the perspective of internal parameters of GANs (Sec. 4.4).

4.1 Image Editing via Single-channel Manipulation

With our method, we can identify meaningful channels through awareness ranking, from which we discover that simply manipulate a single channel can achieve versatile effects for image attribute editing. Such a manipulation can be either multiplying or adding a constant of manipulation magnitude. Therefore, with channels that are relevant to a particular class, we can discover and then manipulate various attributes of this class, such as fur, mouth, and nose for dogs and cats, explosion for volcanic. Results of image editing about class-oriented attribute are shown in Fig. 6. Meanwhile, with channels that respond to latent, we can edit meaningful class-shared semantics controlled by the latent vector, which can therefore manipulate all kinds of classes in BigGAN. Results of image editing about latent-oriented attributes are shown in Fig. 7.

Discussions and comparisons. Compared with prior works which present manipulation results on generated images of BigGAN via manipulating attribute vectors in the latent space [13, 20, 52, 56], our method demonstrates that simply altering only one feature channel could achieve this. Besides, our method can retrieve semantics controlled by both latents and class embeddings, rather than latent only. Compared with StyleSpace [58], and GAN dissection [5], which edit on a single channel, our method has two obvious differences: 1) Our method does not require any semantic segmentation masks and data collection for annotation. Instead, we discover
channels in a fully unsupervised manner. 2) Our method targets class conditional GAN, which can synthesize 1,000 categories rather than unconditional GANs, limited to single class modeling.

4.2 Category Hybridization via Channel Mixing

In this section, we define a novel image editing task called category hybridization, which aims to synthesize realistic images while preserving meaningful attributes from more
than one categories. This task could benefit novel content creation and produce unnatural yet plausible combinations of two categories which do not exist in the training set.

Our method intrinsically fits the category hybridization task since we can locate class-distinct channel representations for different given classes. After that, we transplant the representations of the reference class to the corresponding channel positions in the feature of the input class, which we refer to as channel mixing. To activate the transplanting channels, we exploit the class embedding of the reference class for a continued generation. Results of category hybridization are shown in Fig. 8. We observed that with channel mixing, we could harmoniously fuse attributes from two categories into one realistic image which cannot achieve by style mixing (i.e., simply mixes layer-wise class embeddings). These editing effects also validate the effectiveness of our channel awareness that we can discover class-oriented attributes.

### 4.3. Segmentation via Channel Weighted Clustering

Existing works [1, 32, 33, 54, 60, 65] revealed that internal features in GAN can enable object part segmentation of generated images, facilitating semantic annotation synthesis and local semantic editing. Here, we present a simple prototype that utilizes category-oriented channel awareness to do unsupervised semantic segmentation on generated images. Specifically, we collect the internal features from multiple ReLU layers of the generator, then upsample each feature to be the same size as the output image $H' \times W'$. After upsampling, features from multiple layers could be concatenated along the channel dimension into a total feature volume as $C' \times H' \times W'$, where $C'$ is the total number of channels.

A pixel-wise feature vector in the volume, in size of $C' \times 1 \times 1$, is considered as a data sample that contains sufficient semantic information. Then, we perform $K$-Means clustering on pixel-wise feature vectors to segment the image.

Recall that category-oriented channel awareness quantifies the importance of each channel to a target category. Hence, we employ normalized channel awareness to weight those pixel-wise feature vectors before clustering. Segmentation results without and with channel awareness are shown in Fig. 9. We show the results of two feature collections (i.e., from all layers and the second half of layers) and two different numbers of clusters (i.e., three and five). We can observe that our channel awareness can split very fine regions such as the beak and legs in the first image of the bird in Fig. 9. This is due to our channel awareness could highlight important channel representations and therefore could obtain a more precise and finer segmentation map.

### 4.4. Synthesis Evaluation via Total Channel Awareness

Reliable evaluation metrics can help researchers better assess and improve the overall performance of generative models. In this section, we describe why and how our channel awareness is capable of evaluating the category-wise synthesis performance.

**Assumption and method.** Our key assumption is that if one class has the awareness of consistently activating certain channels for producing category-oriented attributes while deactivating certain channels for suppressing irrelevant attributes, then the category tend to provide faithful generation results. For example, generating a flower needs to suppress those channels responsible for eyes and nose (mainly used by other classes like animals) and activate those related to petals. Thus, we measure the overall selection and suppression awareness as total channel awareness, by summing the absolute values of category-oriented channel awareness over all layers, which can be denoted as follows: $a_c = \sum_{e \in C'} |E_c[t'_e]|$. Here $C'$ means the number of total channels among all layers and $e$ is the given class.

**Results.** With the total channel awareness, we empirically find that classes with the higher total channel awareness exhibit better quality yet low diversity of generated samples.

| Input | Reference | Output w/o Channel Mixing | Output w/ Channel Mixing | Input | Reference | Output w/o Channel Mixing | Output w/ Channel Mixing |
|-------|-----------|---------------------------|--------------------------|-------|-----------|---------------------------|--------------------------|
| ![Input Image](image1.png) | ![Reference Image](image2.png) | ![Output Image w/o Channel Mixing](image3.png) | ![Output Image w/ Channel Mixing](image4.png) | ![Input Image](image5.png) | ![Reference Image](image6.png) | ![Output Image w/o Channel Mixing](image7.png) | ![Output Image w/ Channel Mixing](image8.png) |

![Fig. 8. Category hybridization by mixing the channels that are relevant to two different categories. Different from the conventional style mixing [25] (i.e., the third column), our approach (i.e., the fourth column) better fuses the characteristics (including both shape and appearance) of both classes.](image9.png)
Fine-grained semantic segmentation with channel weighted clustering. Concretely, we collect the multi-level features produced by the generator, weight them with the proposed channel awareness, and perform clustering on all pixels. It turns out that our approach helps the clustering to focus more on the relevant channels, and hence suggests a better performance.

5 classes with highest total channel awareness

Manhole Cover

| Total Channel Awareness: 568; | Precision: 0.98; | Recall: 0.03; |
|-------------------------------|-----------------|---------------|

Rapeseed

| Total Channel Awareness: 527; | Precision: 0.85; | Recall: 0.14; |
|-------------------------------|-----------------|---------------|

Odometer

| Total Channel Awareness: 512; | Precision: 0.93; | Recall: 0.03; |
|-------------------------------|-----------------|---------------|

Website

| Total Channel Awareness: 510; | Precision: 0.98; | Recall: 0.04; |
|-------------------------------|-----------------|---------------|

Cypripedium parviflorum

| Total Channel Awareness: 502; | Precision: 0.84; | Recall: 0.05; |
|-------------------------------|-----------------|---------------|

5 classes with lowest total channel awareness

Chainsaw

| Total Channel Awareness: 205; | Precision: 0.65; | Recall: 0.20; |
|-------------------------------|-----------------|---------------|

Stretcher

| Total Channel Awareness: 199; | Precision: 0.70; | Recall: 0.20; |
|-------------------------------|-----------------|---------------|

Reel

| Total Channel Awareness: 198; | Precision: 0.73; | Recall: 0.13; |
|-------------------------------|-----------------|---------------|

Plastic Bag

| Total Channel Awareness: 197; | Precision: 0.71; | Recall: 0.20; |
|-------------------------------|-----------------|---------------|

Barrow

| Total Channel Awareness: 194; | Precision: 0.73; | Recall: 0.15; |
|-------------------------------|-----------------|---------------|

Fig. 10. Qualitative results of evaluating category-wise synthesis performance, which is enabled by our total channel awareness. We observe that classes with high total awareness tend to have high synthesis quality yet low diversity.

Qualitative results of generated samples of classes with the highest and lowest total channel awareness are shown in Fig. 10. For quantitative verification, we calculate the correlation between total channel awareness and other widely used metrics for GANs among all 1k classes. Specifically, we estimate the quality via precision [29], diversity via recall [29] and MS-SSIM [39], and a general metric FID [15] for each class. Implementation details can be found in Appendix B. We plot the results in Fig. 11, with raw data (dots in light colors) smoothed by a sliding window of size 20 and a scaled y-axis for better view. Note that for metrics taking diversity into major consideration (i.e., Recall and MS-SSIM), higher total channel awareness relates to lower diversity, which is aligned with the observation presented in Fig. 10.

Discussions. Unlike FID, precision and recall, which are estimated via features extracted from a wide range of generated and real images (e.g., usually 50k generated images for FID), and MS-SSIM, which is calculated via similarity between sample pairs, our total channel awareness is calculated by the statistics of internal parameters in GANs (statistics...
of the proposed channel probe). This difference makes our method own several obvious advantages as follows: 1) Our algorithm runs extremely faster than aforementioned metrics. Instead of sampling numerous images and then extracting features via an extra classifier, channel awareness can be easily estimated via forwarding only the FC layer in the CCBN module. 2) Additionally, we provide a novel and complementary perspective for evaluating class conditional GANs based on the internal parameters inside a GAN. We provide insight that if a class has high channel awareness of activating and deactivating certain channels, then the class tends to generate well, based on which a better regularization for improving synthesis performance can be developed in future works. 3) Furthermore, as we illustrate that high-quality classes exhibit low intra-class diversity, this implies that the class generation tends to sacrifice one aspect to meet another, showing the quality-diversity trade-off the model exhibits. This observation may provide future work with insights on how to better balance the this trade-off during learning.

5 Conclusion and Discussion

This work takes the first step towards understanding the generation mechanism of conditional GANs from the channel perspective. Concretely, we propose simple yet effective channel awareness, by which we successfully identify that channels that response to the latent code and to different class embedding. Extensive analyses shed light on how a conditional GAN manages the categorical information with different channels. More importantly, our channel awareness enables four novel applications with class conditional generators, which are rarely explored by prior work. In particular, we achieve single-channel attribute editing as well as harmonious category hybridization in a fully unsupervised manner. We also demonstrate the promising potential of our channel awareness in fine-grained semantic segmentation and category-wise synthesis performance evaluation.

Despite the appealing results, there are still some directions worth exploring. Recall that this work primarily targets the BigGAN generator [6], which learns the categorical information through CCBN with ReLU activation. Investigating other architectures, like the style modulation layer [25] with Leaky ReLU activation [59], can be one of the future works. Besides, our approach mainly focuses on interpreting and utilizing a well-learned model for downstream tasks. How the insights provided in this work can inspire the design of a more powerful GAN model would be of more significance.

References

[1] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka. Labels4free: Unsupervised segmentation using StyleGAN. In IEEE Conf. Comput. Vis. Pattern Recog., 2021.
[2] D. Bau, A. Andonian, A. Cui, Y. Park, A. Jahanian, A. Oliva, and A. Torralba. Paint by word. arXiv preprint arXiv:2103.10951, 2021.
[3] D. Bau, S. Liu, T. Wang, J.-Y. Zhu, and A. Torralba. Rewriting a deep generative model. In Eur. Conf. Comput. Vis., 2020.
[4] D. Bau, H. Strobelt, W. Peebles, J. Walff, B. Zhou, J.-Y. Zhu, and A. Torralba. Semantic photo manipulation with a generative image prior. arXiv preprint arXiv:2005.07727, 2019.
[5] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In Int. Conf. Learn. Represent., 2019.
[6] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In Int. Conf. Learn. Represent., 2019.
[7] K. C. Chan, X. Wang, X. Xu, J. Gu, and C. C. Loy. GLEAN: Generative latent bank for large-factor image super-resolution. In IEEE Conf. Comput. Vis. Pattern Recog., pages 14245–14254, 2021.
[8] E. Collins, R. Bala, B. Price, and S. Süssstrunk. Editing in style: Uncovering the local semantics of GANs. In IEEE Conf. Comput. Vis. Pattern Recog., 2020.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, 2009.

[10] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. *Adv. Neural Inform. Process. Syst.*, 2021.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Adv. Neural Inform. Process. Syst.*, 2014.

[12] J. Gu, Y. Shen, and B. Zhou. Image processing using multicode GAN prior. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 3012–3021, 2020.

[13] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. GANSpace: Discovering interpretable GAN controls. In *Adv. Neural Inform. Process. Syst.*, 2020.

[14] Y. He, Y. Xing, T. Zhang, and Q. Chen. Unsupervised portrait shadow removal via generative priors. In *ACM Int. Conf. Multimedia*, pages 236–244, 2021.

[15] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.*, 2017.

[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Int. Conf. Learn. Represent.*, 2017.

[17] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, pages 6840–6851, 2020.

[18] M. Huh, R. Zhang, J.-Y. Zhu, S. Paris, and A. Hertzmann. Transforming and projecting images to class-conditional generative networks. In *Eur. Conf. Comput. Vis.*, 2020.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 1125–1134, 2017.

[20] A. Jahanian, L. Chai, and P. Isola. On the “steerability” of generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2020.

[21] M. Kang and J. Park. ContraGAN: Contrastive learning for conditional image generation. *Adv. Neural Inform. Process. Syst.*, pages 21357–21369, 2020.

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Int. Conf. Learn. Represent.*, 2018.

[23] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila. Training generative adversarial networks with limited data. In *Adv. Neural Inform. Process. Syst.*, 2020.

[24] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. In *Adv. Neural Inform. Process. Syst.*, 2021.

[25] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, 2019.

[26] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, 2020.

[27] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Adv. Neural Inform. Process. Syst.*, 2018.

[28] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[29] T. Kynkänneni, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved precision and recall metric for assessing generative models. *Adv. Neural Inform. Process. Syst.*, 2019.

[30] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato. Fader networks: Manipulating images by sliding attributes. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

[31] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 4681–4690, 2017.

[32] D. Li, H. Ling, S. W. Kim, K. Kreis, A. Barriuso, S. Fidler, and A. Torralba. BigDatasetGAN: Synthesizing imagenet with pixel-wise annotations. *arXiv preprint arXiv:2201.04684*, 2022.

[33] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler. EditGAN: High-precision semantic image editing. In *Adv. Neural Inform. Process. Syst.*, 2021.

[34] J. Luo, K. Ying, and J. Bai. Savitzky–golay smoothing and differentiation filter for even number data. In *Signal processing*, pages 1429–1434, 2005.

[35] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin. PLUSE: Self-supervised photo upsampling via latent space exploration of generative models. In *IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 2437–2445, 2020.

[36] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[37] T. Miyato and M. Koyama. cGANs with projection discriminator. *Int. Conf. Learn. Represent.*, 2018.

[38] V. Nair and G. E. Hinton. Rectified linear units improve neural network performance. *Adv. Neural Inform. Process. Syst.*, 2010.

[39] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Int. Conf. Mach. Learn.*, 2017.

[40] X. Pan, X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *Eur. Conf. Comput. Vis.*, 2020.

[41] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu. Contrastive learning for unpaired image-to-image translation. In *Eur. Conf. Comput. Vis.*, pages 319–345, 2020.

[42] T. Park, J.-Y. Zhu, O. Wang, J. Lu, E. Shechtman, A. Efros, and R. Zhang. Swapping autoencoder for deep image manipulation. *Adv. Neural Inform. Process. Syst.*, pages 7198–7211, 2020.
[43] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski. StyleClip: Text-driven manipulation of StyleGAN imagery. In Int. Conf. Comput. Vis., pages 2085–2094, 2021.

[44] W. Peebles, J.-Y. Zhu, R. Zhang, A. Torralba, A. Efros, and E. Shechtman. GAN-supervised dense visual alignment. In IEEE Conf. Comput. Vis. Pattern Recogn., 2022.

[45] A. Plumerault, H. L. Borgne, and C. Hudelot. Controlling generative models with continuous factors of variations. In Int. Conf. Learn. Represent., 2020.

[46] A. Razavi, A. Van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In Adv. Neural Inform. Process. Syst., 2019.

[47] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In Int. Conf. Mach. Learn., pages 1530–1538, 2015.

[48] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. Encoding in style: a StyleGAN encoder for image-to-image translation. In IEEE Conf. Comput. Vis. Pattern Recogn., pages 2287–2296, 2021.

[49] A. Sauer, K. Schwarz, and A. Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. arXiv preprint arXiv:2202.00273, 2022.

[50] Y. Shen, J. Gu, X. Tang, and B. Zhou. Interpreting the latent space of GANs for semantic face editing. In IEEE Conf. Comput. Vis. Pattern Recogn., 2020.

[51] Y. Shen, C. Yang, X. Tang, and B. Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. IEEE Trans. Pattern Anal. Mach. Intell., 2020.

[52] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in GANs. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021.

[53] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In IEEE Conf. Comput. Vis. Pattern Recogn., 2016.

[54] N. Tritrong, P. Rewatbowornwong, and S. Suwajanakorn. Repurposing GANs for one-shot semantic part segmentation. In IEEE Conf. Comput. Vis. Pattern Recogn., 2020.

[55] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. Adv. Neural Inform. Process. Syst., 2017.

[56] A. Vovlovich and A. Babenko. Unsupervised discovery of interpretable directions in the GAN latent space. In Int. Conf. Mach. Learn., 2020.

[57] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In Eur. Conf. Comput. Vis. Worksh., pages 0–0, 2018.

[58] Z. Wu, D. Lischinski, and E. Shechtman. StyleSpace analysis: Disentangled controls for StyleGAN image generation. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021.

[59] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853, 2015.

[60] J. Xu and C. Zheng. Linear semantics in generative adversarial networks. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021.

[61] Y. Xu, Y. Shen, J. Zhu, C. Yang, and B. Zhou. Generative hierarchical features from synthesizing images. In IEEE Conf. Comput. Vis. Pattern Recogn., pages 4432–4442, 2021.

[62] C. Yuan, Y. Shen, and B. Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. Int. J. Comput. Vis., 2020.

[63] C. Zhang, Y. Xu, and Y. Shen. Decorating your own bedroom: Locally controlling image generation with generative adversarial networks. In IEEE Conf. Comput. Vis. Pattern Recogn. Worksh., 2021.

[64] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In Int. Conf. Mach. Learn., pages 7354–7363, 2019.

[65] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barrusso, A. Torralba, and S. Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In IEEE Conf. Comput. Vis. Pattern Recogn., 2021.

[66] J. Zhu, R. Feng, Y. Shen, D. Zhao, Z. Zha, J. Zhou, and Q. Chen. Low-rank subspaces in GANs. In Adv. Neural Inform. Process. Syst., 2021.

[67] J. Zhu, Y. Shen, Y. Xu, D. Zhao, and Q. Chen. Region-based semantic factorization in GANs. arXiv preprint arXiv:2202.09649, 2022.

[68] J. Zhu, Y. Shen, D. Zhao, and B. Zhou. In-domain GAN inversion for real image editing. In Eur. Conf. Comput. Vis., 2020.

[69] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Int. Conf. Comput. Vis., pages 2223–2232, 2017.

Appendix

A Evaluation on BigGAN-Deep

This section provides implementation details and results for evaluating the category-oriented channel awareness on the BigGAN-Deep model [6]. Unlike BigGAN, BigGAN-Deep has twice the number of channels (e.g., 2,048 channels in layer 0), making the single-channel evaluation more time-consuming. Besides, zeroing out a single channel makes an invisible change on the output image. Thus in this experiment, we zero out five channels for each intervention, and every five channels are selected along with the order of sorted channel indices by the category-oriented channel awareness. Results of top four results of four classes are shown in Fig. 12. Since zeroing out five channels makes more changes than zeroing out one channel, here we need to scale the two curves into the range of 0-1 for a better view. The average correlation of all 1,000 classes is 0.603.
Fig. 12. Quantitative evaluation of category-oriented channel awareness on BigGAN-Deep with five-channel modulation. The correlation between the awareness score (blue) and channel-classifier response (red) appears on top of each figure. X-axis: intervention order following the sorted awareness score. Y-axis: Scaled values of both lines. The average correlation of all 1,000 classes is 0.603.

B Synthesis Performance Evaluation

This section gives the implementation details and results for category-wise synthesis performance evaluation via total channel awareness.

Data preparation. For precision, recall, and FID, which need to compare fake images with real images during estimation, we firstly pre-process real images from ImageNet [9] following the procedure in BigGAN training which is cropping and resizing to 256 × 256. The number of real images in ImageNet is around 1,300 for each class. We randomly sample all generated images without truncation for fairly measuring the synthesis performance.

Precision and Recall. Precision [29] is used for measuring sample quality by retrieving fake images and checking whether it locates within the manifold of real images. Recall [29] is used to measure sample diversity by retrieving real images and checking whether it is within the manifold formed by fake images. The ranges of these two metrics are both between zero to one. A higher value indicates a better performance. We calculate precision and recall with the number of fake images equal to the number of real images of each class followed the suggested setting [29]. The feature extraction network is the pre-trained VGG-16. We set the neighborhood size $k$ equal to 3, which is a more robust choice [29].

MS-SSIM. We calculate the similarity between fake image pairs via the image similarity metrics MS-SSIM following [39]. Lower image similarity indicates better diversity. For each class, we randomly sample 100 fake images to construct 10,000 image pairs for calculating similarity, then an averaged similarity score for the class can be obtained as the final result for the intra-class diversity.

Fréchet Inception Distance (FID). FID is a general metric that considers both diversity and quality by directly estimating the distance between the feature distribution of fake and real images. We calculate the FID with 50,000 generated images for each class. Then we exploit a pre-trained Inception-v3 as the feature extractor for estimating the feature distance between real and fake images.

C Efficiency of Computing Channel Awareness

Given a pre-trained class conditional model, our proposed channel awareness, including both category-oriented channel awareness and latent-oriented channel awareness, can be computed unsupervisedly and efficiently. As shown in Fig. 13, with one NVIDIA Tesla V100 GPU, it costs 0.0017s to interpret all channels in the first layer of BigGAN regarding a certain class. The average running time for all layers is 0.0012s. Here, for each layer, the running time is averaged using 200 classes.

D More Results

Given a target class, our category-oriented channel awareness can unsupervisedly identify channels responsible for the class-oriented attributes, based on which image editing can be achieved via altering a single channel. We provide results of more classes for category-oriented attribute editing in Fig. 14. Given one input class and one reference class, mixing channels relevant for the two classes can harmoniously mix attributes from two classes and obtain the realistic output. Fig. 15 presents more results for category hybridization via channel mixing.
Fig. 14. **Category-oriented attribute editing.** For each group of images, the middle one is input, while the left and the right one are manipulating the single channel along the positive and the negative direction. Corresponding class names are annotated on the left.
Fig. 15. **Category hybridization** by mixing the channels that are relevant to two different categories. For each group, the first two columns present the original syntheses, while the third column shows the hybridization result, which successfully fuses the characteristics (including both shape and appearance) of both classes.