Contrastive Learning for Diverse Disentangled Foreground Generation

Yuheng Li[1,2], Yijun Li[2], Jingwan Lu[2], Eli Shechtman[2], Yong Jae Lee[1], and Krishna Kumar Singh[2]

[1]University of Wisconsin-Madison [2]Adobe Research

Abstract. We introduce a new method for diverse foreground generation with explicit control over various factors. Existing image inpainting based foreground generation methods often struggle to generate diverse results and rarely allow users to explicitly control specific factors of variation (e.g., varying the facial identity or expression for face inpainting results). We leverage contrastive learning with latent codes to generate diverse foreground results for the same masked input. Specifically, we define two sets of latent codes, where one controls a pre-defined factor ("known"), and the other controls the remaining factors ("unknown"). The sampled latent codes from the two sets jointly bi-modulate the convolution kernels to guide the generator to synthesize diverse results. Experiments demonstrate the superiority of our method over state-of-the-arts in result diversity and generation controllability.

Keywords: Foreground generation, diversity, disentanglement

1 Introduction

Foreground object generation is the task of filling in the missing foreground region in a given context, such as generating human faces as shown in Figure 1. This task is useful in practice, e.g., for privacy-related applications (anonymizing a person's face by generating a new identity) or replacing/adding objects in an image (replacing a car in a photo if one does not like the original one). It is a special case of image inpainting in which the entire foreground object is masked. In inpainting, when the missing region (hole) is small, there may only be one or few "correct" completions (e.g., if only one eye is masked, then it mostly can be inferred from the other eye), but as the hole gets bigger there should be more diversity in the generated completion, especially when an entire object is masked. As there can be many different plausible solutions for filling in the missing region, this task naturally demands learning a "one-to-many" mapping between the input and outputs (e.g., Figure 1). That is, a good method should 1) synthesize foreground objects that are both *realistic* and *semantically coherent* with the surrounding unmasked context; 2) have the capability to generate *diverse* results for the same missing region and context; and 3) provide *control* over different properties of the synthesized results. While tremendous progress has been made to obtain better realism and coherence [60, 11, 40, 26], progress in diversity is

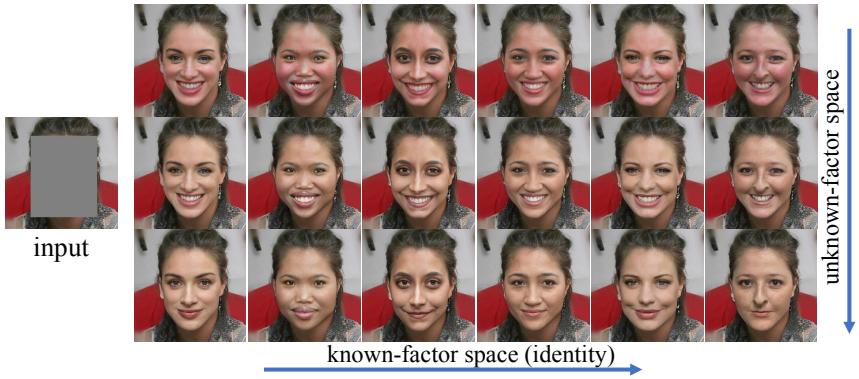

Fig. 1. Foreground generation on the same mask. We use contrastive learning to increase generation diversity. We also explicitly disentangle out an expected predefined factor (human identity here) to increase diversity and controllability.

still unsatisfactory and increasing controllability for the results is also relatively under explored.

Like the inpainting task [4, 12, 3, 32, 28, 61], foreground generation needs to consider coherence between the given context and the generated object. Existing inpaiting work can generate good quality object/foreground, but it usually lacks diversity and controllability. Although there are many inpainting methods trying to generate diverse results [61, 59, 60, 26], the results are still less satisfactory. These methods typically have an encoder-decoder architecture. To achieve diversity, different latent codes can be sampled and injected into these models. However, although the output is a function of both the masked image as well as the latent code, the spatial features from the encoder usually dominate the final results and prevent the latent codes from inducing large changes. For example, in [60], an encoder is used to extract 2D spatial features from the masked image, and skip connections are added to all levels of the encoder and decoder. The information from the latent code can be easily submerged by the large number of features from the encoder.

In this paper, we propose a novel approach for diverse and controllable foreground generation. As shown in Figure 1, our method can generate diverse results for the same input. To synthesize diverse content, we condition the generation on both the masked image and the sampled latent codes, and apply contrastive learning [5] so that the latent codes that are close/far in code space result in corresponding synthesized images that are close/far in image space.

Besides diversity, controllability is another desired property in foreground generation. Thus, we also try to explicitly disentangle a predefined factor by using a pretrained classifier on this factor. For example, as shown in Figure 1, one can disentangle human identity (rows) from other attributes (columns) for face images. We explicitly use two sets of latent codes, where one represents the predefined factor ("known"), and the other controls all the other factors ("un-

known"). This allows us to change the unknown factors while keeping the known factor fixed (e.g,, in Figure 1, changing the facial attributes which are unknown during training while keeping the identity of the face intact). To inject these two codes, we propose a bi-modulated convolution module where the convolution kernels are modulated by the two latent codes from different spaces. We design each training batch to contain a mix of instances that share the same known latent code while differing in the unknown, and instances that share the same unknown latent code while differing in the known. We use a contrastive loss to ensure that known and unknown codes control their respective factors.

Contributions. (1) We propose a novel contrastive learning based approach for diverse foreground generation; (2) An explicit disentangled latent space for controllability via a novel bi-modulated convolution module; (3) More diverse results compared to existing state-of-the-art methods on three different datasets.

2 Related Work

Image inpainting This problem has been studied for decades due to its importance. Traditional methods [2, 4, 41, 12, 3] typically rely on low-level assumptions and image statistics, leading to over smoothing and results with limited visual semantics. Recently, deep learning methods [8, 18, 33, 34, 46, 48, 50, 51, 53, 57, 28, 55, 59] dramatically boosted the quality, in terms of both visual quality and semantic coherence. [32] first uses an encoder-decoder architecture in inpainting with reconstruction loss and adversarial loss [10]. [28] and [55] proposes the use of partial and gated convolutions on irregular masks. However, these methods only generate deterministic results. Thus [61] proposes a VAE-based [23] method allowing pluralistic image completion. Recently proposed [60, 26] use StyleGAN [20, 21] architecture for inpainting. [60] combines encoded features from a masked image with a random latent code to co-modulate StyleGAN convolution kernels. [26] has a similar setting as ours as instead of traditional inpainting, they use a foreground model to synthesize high quality foreground objects conditioned on the background context. In both work, diverse images can be generated by sampling different latent codes injected into StyleGAN. But their diversity in the latent code space is restricted due to extra spatial features from the encoder which usually determine most of the aspects of the generation. [45, 56] also try to use transformer to realize diversity in image inpainting. They both use bidirectional attention to predict missing tokens. However, their image quality suffers compared to styleGAN2-based architectures. Also, none of the existing pluralistic inpainting work enables user controllability in the results via latent code disentanglement.

Contrastive learning Contrastive learning [63, 42, 13, 30, 5] has shown great potential in representation learning. Among them, [5] proposes a simple framework for contrastive learning without requiring specialized architectures or memory bank. Recently [31] proposes to use contrastive learning in image translation task. Also, there are a few work [62, 29] studying contrastive learning in the image inpainting task. Like most inpainting methods, [62, 29] use encoder-decoder

architecture. [62] encode more discriminative features using contrastive loss in different semantic sub-regions. [29] applies the contrastive loss to the output features of encoder by setting two identical images with different masks as positive pairs while different images as negative pairs. However, they are both deterministic inpainting methods which means they only produce a single result per input. Different from prior work, instead of learning a better intermediate feature using contrastive loss, we use contrastive learning to achieve disentanglement and diversity in the latent space, enabling us to produce diverse inpainting results for foreground generation in a controllable way.

Disentanglement learning For a generative model, it is desirable to disentangle the factors of variation. One way is to explicitly learn a disentangled latent space: having separate codes for different factors. A large number of work try to disentangle object shape/structure from appearance [39, 27, 37, 7, 49]. Searching for semantic directions in a pre-trained GAN latent space is another way to achieve disentanglement. This method is getting popular recently and both unsupervised methods [43, 17, 36] and supervised methods [19, 35, 52] are heavily explored. Despite the progress made in the field, few work explore it for the foreground generation task. We use contrastive learning to explicitly learn a disentangled latent space for controllable foreground generation.

3 Approach

Our goal is to propose a model (ContrasFill) which is able to generate diverse foreground objects for the same masked region while providing control over different factors of generated results. We encode spatial features corresponding to masked image and modulate them with randomly sampled latent codes to generate diverse results. However, without applying explicit training loss on diversity, different latent codes might introduce only minor changes as in [60, 26]. Thus we use contrastive learning to encourage the model to synthesize diverse results by forcing latent codes closer in the latent space to produce images closer in the image space and vice versa. To gain explicit control over certain factor, we also try to disentangle the latent space into two spaces: a known factor space which corresponds to an expected factor, and an unknown factor space which controls the rest of other factors. Section 3.1 introduces the training details for achieving diversity and disentanglement using the contrastive loss. Section 3.2 talks about how do we inject two codes into our model using the proposed bi-modulation.

3.1 Contrastive learning for diversity and disentanglement

Figure 2 shows the framework of our method. Our model takes a masked image I with context only and two latent codes as inputs: a known factor code k drawn from a distribution ϕ_k and an unknown factor code u from a distribution ϕ_u to output the synthesized image S. If we define our model as G, then we have $S_{k,u,I} = G(k, u, I)$. Since the input context is independent of the following analysis, we will omit I in $S_{k,u,I}$ for simplicity and talk about the context later.

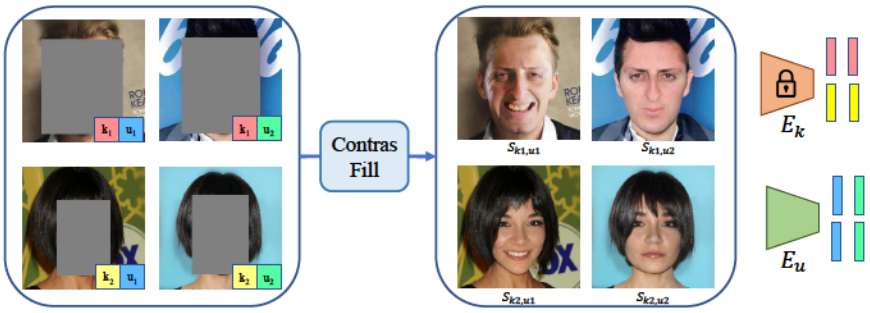

Fig. 2. ContrasFill takes as input two sets of codes (squares on the left): known-factor code (e.g., identity) and unknown-factor code (non-identity factors) to synthesize images. Two encoders (E_k and E_u) embed images into different features (bars on the right, color between code and feature refers to correspondence). A contrastive loss forces features with same/different colors closer/further in the feature spaces.

One can refer to supp. for visual examples to help understand the following analysis.

Suppose we sample N codes from each latent space, we will have N^2 combinations between the code k and u in total. To enforce a contrastive loss in the known and unknown factor latent spaces, we first define an image pair as:

$$p_{(k,u),(k',u')} = (S_{k,u}, S_{k',u'}), \tag{1}$$

Note, we do not consider images sharing the same known and unknown codes as a valid pair. In order words, $k = k'$ and $u = u'$ cannot hold at the same time. We will next define the positive and negative pairs used in our contrastive learning scheme. To simplify the explanation, we first consider the case of the known space.

Contrastive pairs in the known space. A positive image pair contains two images sharing the same known codes but different unknown codes (i.e., $k = k', u \neq u'$ in the Eq 1). We define $P_{\mathbf{k},u}$ as a set of all positive pairs *associated* with the code combination (k, u) in the known space (bold indicates the space). For example, in Figure 2 where we set the known factor as human identity, $P_{\mathbf{k1},u1} = \{p_{(k1,u1),(k1,u2)}\}$. To ease explanation, we denote $P_{\boldsymbol{K}}$ as *all* positive pairs in the known space. Here we have $P_{\boldsymbol{K}} = \{p_{(k1,u1),(k1,u2)}, p_{(k2,u1),(k2,u2)}\}$.

For the negative pair, we define two images not sharing the same known codes (i.e., $k \neq k'$). In this case, we construct two types of negative pairs. The first case is the hard negative pair where two images sharing different known codes but the same unknown codes (i.e., $k \neq k', u = u'$, images in each column in Figure 2). The reason is that these images share the same features (e.g., smile) in the unknown space which forces the learned known latent code to control different aspects of the face due to the use of contrastive loss which we will introduce later. Pairs of images sharing different known and unknown codes (i.e., $k \neq k', u \neq u'$) are easy negative pairs (diagonal image pairs in

Figure 2). Similarly, we define $N_{\mathbf{k},u}$ as all negative pairs *associated* with the code combination (k, u) in the known factor space. For example, for image $S_{k1,u1}$ in Figure 2, $N_{\mathbf{k1},u1} = \{p_{(k1,u1),(k2,u1)}, p_{(k1,u1),(k2,u2)}\}$.

Contrastive loss in the known space. The job of the known space is to control an expected factor during the generation. To push the model to learn this correspondence, we use contrastive learning. The intuition is to push images closer/further if they are positive/negative pairs. In order to measure the distance between two images, we define the similarity score f as:

$$f_{(k,u),(k',u')} = e^{\text{sim}(\boldsymbol{z}_{k,u}, \boldsymbol{z}_{k',u'})/\tau}, \tag{2}$$

where $\boldsymbol{z}_{k,u}$ is the extracted feature of the image $S_{k,u}$ from an encoder, $\text{sim}(\cdot,\cdot)$ is the cosine similarity and τ denotes a temperature parameter. To force our known space to control the expected factor, we assume having access to a pretrained and fixed classifier. The encoder E_k for the known space will output the penultimate feature of the classifier. For example, in faces, we use a pretrained ArcFace [6] to extract identity features.

For an image $S_{k,u}$, and its positive pair $S_{k,u'}$ in the known space, the contrastive loss becomes:

$$\ell_{(k,u),(k,u')} = -\log \frac{f_{(k,u),(k,u')}}{f_{(k,u),(k,u')} + FN_{\mathbf{k},u}}, \tag{3}$$

where $FN_{\mathbf{k},u}$ is the sum of similarity scores of all negative pairs with respect to the image $S_{k,u}$. In other words, it is the summation of Eq. 2 over all elements in the $N_{\mathbf{k},u}$. Finally, the total loss for the known space becomes:

$$\mathcal{L}_{known} = \frac{1}{|P_{\boldsymbol{K}}|} \sum \ell_{(k,u),(k,u')}, \tag{4}$$

where the summation is over all positive pairs $P_{\boldsymbol{K}}$ in the known space.

Contrastive learning in the unknown space. The contrastive learning idea is similarly applied to the unknown latent space and we highlight the main difference below.

In the unknown space, positive pairs share the same unknown code (each column in Figure 2) and negative pairs have different unknown codes. Similarly, we define $P_{\boldsymbol{U}}$ as *all* positive pairs in the unknown space. For example, in Figure 2, $P_{\boldsymbol{U}} = \{p_{(k1,u1),(k2,u1)}, p_{(k1,u2),(k2,u2)}\}$. For an image $S_{k,u}$, we define all negative pairs associated with it in the unknown space as $N_{k,\mathbf{u}}$ (bold indicates space). For example, $N_{k1,\mathbf{u1}} = \{P_{(k1,u1),(k1,u2)}, P_{(k1,u1),(k2,u2)}\}$ In the unknown space, the image feature $\boldsymbol{z}_{k,u}$ for calculating image pair similarity in the Eq 2 is extracted from an encoder E_u which is trained from scratch. This is because it is hard to define what factors can be controlled in the unknown space beforehand. Also, this avoids pre-training an additional feature extractor and simplifies our approach.

Then, for an image $S_{k,u}$, and its positive pair $S_{k',u}$, the counterpart of Eq 3 in the unknown space becomes

$$\ell_{(k,u),(k',u)} = -\log \frac{f_{(k,u),(k',u)}}{f_{(k,u),(k',u)} + FN_{k,\mathbf{u}}}, \tag{5}$$

where $FN_{k,\mathbf{u}}$ is the sum of similarity score of all negative pairs with respect to image $S_{k,u}$ in the unknown space. The total loss for the unknown space becomes:

$$\mathcal{L}_{unknown} = \frac{1}{|P_{\boldsymbol{U}}|} \sum \ell_{(k,u),(k',u)}, \tag{6}$$

where the summation is over all positive pairs in the $P_{\boldsymbol{U}}$ in the unknown space.

In this way the disentanglement can be learned because we use a pretrained encoder for the known-factor which only extracts expected features, thus the model will synthesize known-factors in images according to codes sampled from known space. For the unknown space, due to the existence of hard negative pair (sharing the same known factors), different unknown codes need to generate factors that are different from known factor to minimize the contrastive loss.

Overall we have the final loss $\mathcal{L}$ as

$$\mathcal{L} = \mathcal{L}_{gan} + \lambda_1 \mathcal{L}_{known} + \lambda_2 \mathcal{L}_{unknown}, \tag{7}$$

where $\mathcal{L}_{gan}$ is same as the one used in the StyleGAN2 [22]. $\mathcal{L}_{known}$ and $\mathcal{L}_{unknown}$ are two contrastive losses in known and unknown latent spaces. λ_1 and λ_2 are their weights. We sample different context (background) for different code combinations (e.g., N^2 in total) since we want to have the same context distribution for both the real and fake batches when training the discriminator. More details are presented in the supp.

3.2 Codes injection with bi-modulated convolution

Our model uses an encoder-decoder architecture (details in the supp). Inspired by StyleGAN2 that shows the effectiveness of modulation, we also use our latent codes to modulate convolution kernels. However, since we have two latent codes, we propose the bi-modulation, where the convolution kernel is modulated by two codes. We use this novel modulation scheme for all convolutions in our model.

Figure 3 shows the bi-modulation process. The two codes k and u first go to two separate fully connected layer to become scaling vectors s and t. The length of scaling vectors is the same as the number of input channels of a convolution kernel. Then the scaling vectors bi-modulate the convolution weight by: $w'_{ijk} = s_i \cdot t_i \cdot w_{ijk}$, where w and w' are the original and the bi-modulated weights. s_i and t_i are the scaling factors corresponding to the ith input feature map. j and k enumerate the output feature maps and the spatial footprint of the convolution.

4 Experiments

We perform quantitative and qualitative evaluations via comparing our proposed foreground generation model ContrasFill with prior arts.

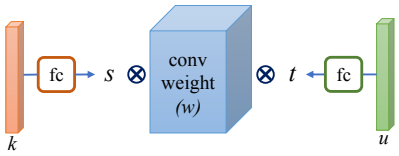

Fig. 3. The proposed bi-modulation scheme, where convolution kernels are modulated by two disentangled latent codes.

Datasets. We conduct the evaluation on three different datasets: 1) **Face**. We use CelebAMask-HD [25] that includes 30,000 face images with segmentation masks. We follow the official training/testing split. To acquire more training data, we use a publicly available face parsing model [1] on FFHQ [21] as extra training data. We use a pretrained face recognition model [6] as our known factor feature extractor. 2) **Bird**. We use the *bird* category from LSUN dataset [54]. We choose images greater than certain resolution and run the pretrained MaskR-CNN [14, 47] to remove bad images. In total, we have 34,969 images and we randomly select 10% (3,497) as test data. We train a fine-grained classification model [9] on the CUB dataset [44] as our known factor feature extractor. 3) **Car**. We use the *car* category from LSUN dataset [54] and same preprocessing steps to clean our data. In total, we have 77,840 images and we randomly select 10% (7,784) as test data. We train a shape classifier [9] on the Stanford car dataset [24] as our known factor feature extractor. To measure the extent of our ability to synthesize diverse results, we use the object bounding box as the missing region in our main study.

We train our model at 256×256 resolution on all datasets. Our unknown factor code is drawn from the normal distribution. Our known factor codes are drawn from one-hot distribution for the cars and birds; for faces, we choose to draw from a hypersphere which is the feature distribution of penultimate layer of ArcFace. We sample $N = 8$ different known and unknown codes in each training minibatch. Due to memory issue, we can not fit all 64 combinations, thus we subsample one hard negative pair for each code, resulting in a batch size of 16 during training. Please refer to supp for more dataset and implementation details.

Baselines. We mainly compare with: 1) **CollageGAN** [26], which generates foreground object conditioned on the background; 2) **CoModGAN** [60], a state-of-the-art image inpainting model. These two methods, built on top of StyleGAN2 [22], are able to generate multiple results via sampling codes in the latent space; 3) **BAT-fill** [56], a recently proposed two-stage inpainting model using transformer. It first autoregressively predicts missing tokens in a 32×32 image with bidirectional attention, and then use a convolutional network to perform upsampling to 256×256. It samples different plausible missing tokens in the 32×32 grid to achieve diversity. This work has demonstrated the benefits of adding autoregressive predication over other transformer-based inpainting work [45]. We train all baseline models with the same input mask setting as ours.

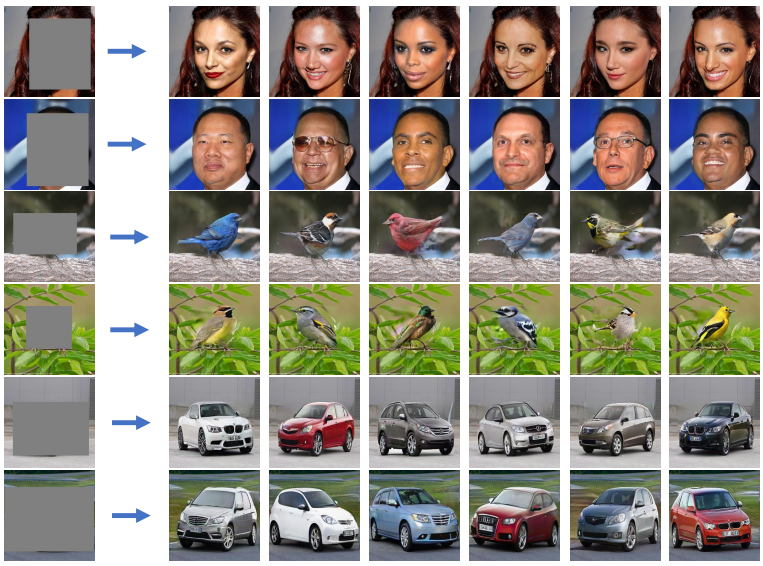

Fig. 4. We can achieve diverse samples from the same masked image by sampling in both known and unknown code spaces.

Evaluation. We use the following metrics: 1) **FID** [15] measures the quality and diversity by comparing distributions between the real and the generated images. 2) **LPIPS** [58] measures distance between two images in deep feature space. For each testing image, we compare pairs of inpainting results generated from the same input mask, we use this to measure diversity. 3) Known Factor Feature Angle (**KFFA**). To better understand how we can improve result diversity using the disentangled known factor, we sample 10 inpainting results for each input image. Then we compute deep features of these 10 results from a known factor classifier. We report average angle between all normalized feature pairs. For a fair comparison, we use feature extractors different from the one used in the training. For face, we use CurricularFace [16], and for bird and car, we train a new classifier using the VGG architecture [38]. Note that L1, SSIM and PSNR are also commonly used metrics for inpainting tasks. However, they all favor deterministic methods which aim to reproduce the single ground truth. As pointed out by [45], these metrics are more suitable for small mask cases where the synthesized contents are more likely to be similar to the ground truth. With large holes covering an entire semantic region or object, synthesized diverse contents might look plausible but different from the ground truth.

4.1 Qualitative results

Figure 4 shows random samples from our model given the same masked input. Our method can generate diverse identities, facial attributes for faces and synthesize diverse shapes, poses and object appearances for birds and cars. Figure 5

Fig. 5. Compared with the baselines, our method generates more diverse results. For faces, we have more variations in identity. For birds and cars, we have different object shapes and textures.

shows side-by-side comparisons between our method and other baselines. Our results on faces are more diverse compared to CoModGAN [60] and CollageGAN [26]. For unaligned dataset (cars and birds), these two methods tends to generate results with the same shape. BAT-fill [56] results have better diversity, but lower image quality. It sometimes generates artifacts on faces or distorted geometries for cars.

We also evaluate how disentangled our results are in Figure 6. Each column shares the same known factor and each row shares the same unknown factor. For faces, the known code controls identity and the unknown code controls facial attributes such as smile, glass and lighting condition. For cars, the known code controls car shape (e.g., sedan-like and wagon-like in the second and third column), and the unknown code changes color and orientation. For birds, the known factor changes species (color is associated with species) and the unknown factor changes pose and orientation. When one latent code changes, the image changes only along the direction it is supposed to, e.g, as the identities change, the same facial expression remains within each row in the face results.

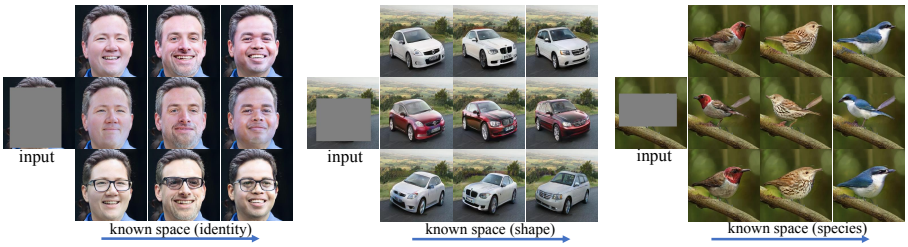

Fig. 6. Known factor code is same for each column, unknown factor code is same for each row.

	Face			Bird			Car		
	FID	LPIPS	KFFA	FID	LPIPS	KFFA	FID	LPIPS	KFFA
CoModGAN	8.88	0.045	52.41	11.35	0.090	52.45	6.59	0.183	44.34
CollageGAN	8.77	0.069	66.00	12.11	0.100	61.08	6.57	0.191	48.67
BAT-Fill	15.08	**0.102**	75.98	37.15	0.117	55.41	22.20	0.270	51.98
ContrasFill-1	8.40	0.072	74.71	**11.29**	0.151	66.06	**6.24**	0.310	63.09
ContrasFill (Ours)	**8.36**	0.075	**83.66**	11.97	**0.160**	**74.58**	6.46	**0.327**	**82.96**

Table 1. Our method has comparable image quality with the state-of-the-art, but with more diversity.

4.2 Image quality and diversity

Besides our model ContrasFill, we also evaluate one variant of our approach, where we only have one latent space using contrastive learning without explicit latent disentanglement (denoted as "ContrasFill-1", see supp for details about this variant). This entangled latent space models all factors together for generation. It is used to show the effectiveness of the contrastive loss on diversity.

Table 1 shows comparison in terms of image quality (FID) and diversity (LPIPS for overall, KFFA for known factor). Overall, our model has comparable image quality with the state-of-the-art methods, and it performs favorably against all baselines in terms of known factor diversity, especially compared with CoModGAN [60] and CollageGAN [26]. We also have the highest overall diversity on bird and car datasets. Although, BAT-fill [56] has better LPIPS distance in face dataset, but their image quality is worse (Figure 5) and they sometimes generate artifacts, which can often results in larger LPIPS difference. Our model also has better diversity, especially on the known factors, compared with our single code variant (ContrasFill-1).

We also compared with UCTGAN [59] which is designed for diverse hole filling. Due to code unavailablity, we only compare with it in CelebA-HQ dataset and use the same setting as theirs. Here we measure diversity LPIPS for full output and only mask region. We grab their numbers and notations (ours first): I_{out}: **0.036** vs 0.030; $I_{out(m)}$: **0.101** vs 0.092.

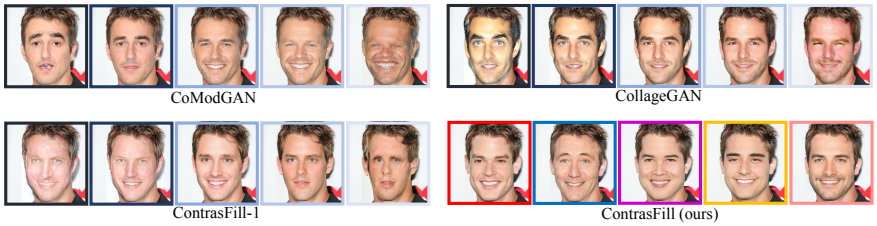

Fig. 7. Moving along the discovered identity direction causes changes in non-identity factors such as facial expressions and lighting for baselines. Our results only vary in identity.

4.3 Disentanglement study

We compare with baselines to show that having two explicit latent spaces improves the disentanglement. Recent works [43, 17, 36] show that postprocessing can be applied to find disentangled latent directions in a pretrained GAN space. Thus we use a supervised method [35] to find known factor directions for CoModGAN, CollageGAN and ContrasFill-1. We use pretrained known factor classifiers to get labels for sampled latent codes and then train a linear regressor to find latent directions [35]. We do not compare with BAT-fill since they lack controllability.

Next, we generate images with different known factors. For baselines, given a masked image, we first randomly sample a latent code, and then move along the discovered known direction to generate 10 different results. For our approach, since we have a disentangled space, thus we directly sample 10 different codes in known factor space by fixing unknown code. We calculate KFFA for 1,000 different contexts and report the average number in the Table 2. Our model has the best KFFA scores. This demonstrates the benefit of having an explicitly disentangled latent space.

Since we has two spaces, we also conduct an experiment where we fix our known code and randomly sample 10 unknown codes. The last row of Table 2 shows the average KFFA numbers over 1,000 context images which indicates when unknown factor varies, our known factor is less influenced. Please refer to Figure 6 for qualitative results.

We also visually examine the baselines in Figure 7. By moving along the discovered known (identity) direction, those baselines not only change identity to some extent, but also alter other attributes, such as smile, skin tone, and gaze, whereas our sampled results maintain the attributes controlled by the unknown factor while the known factor changes. This means that certain factors such as human identity can not be easily disentangled during the latent direction discovery stage even with explicit supervision. We also show that this is true for vanilla unconditional StyleGAN [21, 22] in the supp. Note, we move large steps in two directions on purpose (leftmost and rightmost) to show the full effect of the discovered directions. Details about this study can be found in the supp.

	face	bird	car
CoModGAN	56.00	47.55	44.25
CollageGAN	57.78	58.15	47.49
ContrasFill-1	67.40	61.14	52.01
ContrasFill	**82.03**	**75.20**	**83.71**
ContrasFill (known fixed)	26.15	21.69	35.47

Table 2. High KFFA shows our latent space is more diverse compared with discovered latent directions in baselines. The last row is a different setting, please refer the text.

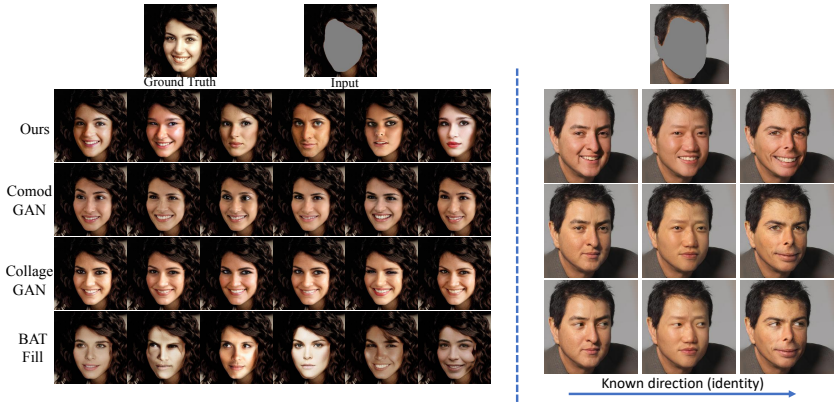

Fig. 8. Our model can generate diverse disentangled results in semantic mask case.

4.4 Ablation

Other type of masks. We also analyze our model's performance on the inpainting task where the input mask is of the shape of an object instead of a box. In this case, our model can no longer change the object shape and pose but can still generate diverse appearance in the mask. Figure 8 (left) shows that our results are more diverse than CoModGAN and CollageGAN, especially on human identity. BAT-fill has worse image quality. We can also achieve disentanglement (Figure 8 right). We compare image quality and diversity (LPIPS for overall, KFFA for known factor), and report numbers in Table 3 (left) for the face dataset. We also study the case of arbitrarily-shaped masks that cover part of the object in random places (e.g., half of face is hidden); see supp for details.
Latent codes injection method. To study the effectiveness of our bi-modulated convolutions (Figure 3), we try the following alternative approaches: (1) We concatenate s with t and pass the result to fully-connected layers to output a single scale vector to modulate the convolutions (denoted as "concat"); (2) We use each code to modulate a different set of convolution kernels. And the two sets of modulated convolutions are sequentially applied to image features. Depending on the order, we denote them as "k-u" and "u-k" (k and u stand for known and unknown codes). The first three columns of Table 3 (right) indicate that, these

	CoMod	Collage	BAT	Ours
FID	**5.73**	6.07	11.97	5.95
LPIPS	0.029	0.029	**0.050**	0.048
KFFA	51.19	58.73	72.48	**83.39**

	concat	u-k	k-u	repredict	Ours
FID	8.64	**8.41**	8.42	14.41	**8.41**
LPIPS	0.048	0.052	0.056	**0.120**	0.075
KFFA	48.99	78.19	77.58	**86.87**	83.06

Table 3. (Left) Comparison in mask case. Our method has comparable image quality but with more diversity on faces when using face masks as inpainting regions. **(Right) Ablation.** results indicate the effectiveness of bi-modulation (the first three columns) and contrastive loss (4th column).

alternative designs achieve similar image quality, but lower level of diversity. Our bi-modulation is a more direct way to inject information to the generator which makes the learning process easier compared with the "concat" alternative. The approach of applying two separate sets of convolutions results in poor diversity. We hypothesize that the later convolution set may undo what is learnt by the previous set as their objective functions are different (factors in two spaces should learn different things).

The loss choice. We use the contrastive loss to encourage diversity by forcing images with the same latent codes to have similar factors. Another way to learn this correspondence is to repredict the input code from the resulting image. For example, by sampling a code in the identity space, one can use ArcFace to repredict this code from the generated image. Table 3 (right 4th column) shows that replacing the contrastive loss with reprediction loss encourages more diversity, but at the cost of image quality. This is because a foreground generation model needs to consider the compatibility between the sampled latent code and the input context. For example, if the context contains light skin pixels on the neck, then latent codes that generate dark-skinned faces are not compatible. However, the reprediction loss forces the model to synthesize a dark-skinned face, which may not look real according to the discriminator. However if a contrastive loss is applied, which considers the *relative distance* in the feature space, then the model can adjust the input identity code based on the context information to synthesize a face that looks more plausible in the context.

5 Conclusion and Limitations

We propose ContrasFill, a novel approach for diverse and controllable foreground generation by contrastive learning. We demonstrate superior diversity and controllability over previous work. Our method has some limitations. We found that our model is sometimes sensitive to the pretrained classifier which may be biased due to the training data. For example, certain types of car are more common in certain color (e.g., van are usually white). Thus our model may also be biased.

Acknowledgement This work was supported in part by Sony Focused Research Award, NSF CAREER IIS-2150012, Wisconsin Alumni Research Foundation, and NASA 80NSSC21K0295.

References

1. https://github.com/zllrunning/face-parsing.pytorch
2. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. IEEE Transactions on Image Processing **10**(8), 1200–1211 (2001). https://doi.org/10.1109/83.935036
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. In: SIGGRAPH 2009 (2009)
4. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. pp. 417–424 (01 2000)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. ArXiv **abs/2002.05709** (2020)
6. Deng, J., Guo, J., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4685–4694 (2019)
7. Denton, E.L., Birodkar, V.: Unsupervised learning of disentangled representations from video. ArXiv **abs/1705.10915** (2017)
8. Ding, D., Ram, S., Rodríguez, J.J.: Image inpainting using nonlocal texture matching and nonlinear filtering. IEEE Transactions on Image Processing **28**, 1705–1719 (2019)
9. Du, R., Chang, D., Bhunia, A.K., Xie, J., Song, Y.Z., Ma, Z., Guo, J.: Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. In: European Conference on Computer Vision (2020)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
11. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14134–14143 (October 2021)
12. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: SIGGRAPH 2007 (2007)
13. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.B.: Momentum contrast for unsupervised visual representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9726–9735 (2020)
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask r-cnn. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**, 386–397 (2020)
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS (2017)
16. Huang, Y., Wang, Y., Tai, Y., Liu, X., Shen, P., Li, S., Jilin Li, F.H.: Curricularface: Adaptive curriculum learning loss for deep face recognition pp. 1–8 (2020)
17. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. In: Proc. NeurIPS (2020)
18. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (TOG) **36**, 1 – 14 (2017)
19. Jahanian, A., Chai, L., Isola, P.: On the "steerability" of generative adversarial networks. In: International Conference on Learning Representations (2020)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2018)

21. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)
22. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proc. CVPR (2020)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. CoRR **abs/1312.6114** (2014)
24. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia (2013)
25. Lee, C.H., Liu, Z., Wu, L., Luo, P.: Maskgan: Towards diverse and interactive facial image manipulation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
26. Li, Y., Li, Y., Lu, J., Shechtman, E., Lee, Y.J., Singh, K.K.: Collaging class-specific gans for semantic image synthesis. ICCV (2021)
27. Li, Y., Singh, K.K., Ojha, U., Lee, Y.J.: Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8036–8045 (2020)
28. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. ECCV **abs/1804.07723** (2018)
29. Ma, X., Zhou, X., Huang, H., Chai, Z., Wei, X., He, R.: Free-form image inpainting via contrastive attention network. 2020 25th International Conference on Pattern Recognition (ICPR) pp. 9242–9249 (2021)
30. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6706–6716 (2020)
31. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: ECCV (2020)
32. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2536–2544 (2016)
33. Ren, Y., Yu, X., Zhang, R., Li, T.H., Liu, S., Li, G.: Structureflow: Image inpainting via structure-aware appearance flow. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 181–190 (2019)
34. Sagong, M.C., Shin, Y.G., Kim, S.W., Park, S., Ko, S.: Pepsi : Fast image inpainting with parallel decoding network. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11352–11360 (2019)
35. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9240–9249 (2020)
36. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. ArXiv **abs/2007.06600** (2020)
37. Shu, Z., Sahasrabudhe, M., Güler, R.A., Samaras, D., Paragios, N., Kokkinos, I.: Deforming autoencoders: Unsupervised disentangling of shape and appearance. In: ECCV (2018)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2015)
39. Singh, K.K., Ojha, U., Lee, Y.J.: Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6483–6492 (2019)

40. Suin, M., Purohit, K., Rajagopalan, A.N.: Distillation-guided image inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2481–2490 (October 2021)
41. Telea, A.: An image inpainting technique based on the fast marching method. Journal of Graphics Tools **9** (01 2004). https://doi.org/10.1080/10867651.2004.10487596
42. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: ECCV (2020)
43. Voynov, A., Babenko, A.: Unsupervised discovery of interpretable directions in the gan latent space. ArXiv **abs/2002.03754** (2020)
44. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001 (2011)
45. Wan, Z., Zhang, J., Chen, D., Liao, J.: High-fidelity pluralistic image completion with transformers. ICCV (2021)
46. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: NeurIPS (2018)
47. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)
48. Xie, C., Liu, S., Li, C., Cheng, M.M., Zuo, W., Liu, X., Wen, S., Ding, E.: Image inpainting with learnable bidirectional attention maps. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 8857–8866 (2019)
49. Xing, X., Gao, R., Han, T., Zhu, S.C., Wu, Y.N.: Deformable generator network: Unsupervised disentanglement of appearance and geometry. IEEE transactions on pattern analysis and machine intelligence **PP** (2020)
50. Xiong, W., Yu, J., Lin, Z.L., Yang, J., Lu, X., Barnes, C., Luo, J.: Foreground-aware image inpainting. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5833–5841 (2019)
51. Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. ArXiv **abs/1801.09392** (2018)
52. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis. Int. J. Comput. Vis. **129**, 1451–1466 (2021)
53. Yang, C., Lu, X., Lin, Z.L., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4076–4084 (2017)
54. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. ArXiv **abs/1506.03365** (2015)
55. Yu, J., Lin, Z.L., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 4470–4479 (2019)
56. Yu, Y., Zhan, F., Wu, R., Pan, J., Cui, K., Lu, S., Ma, F., Xie, X., Miao, C.: Diverse image inpainting with bidirectional and autoregressive transformers. Proceedings of the 29th ACM International Conference on Multimedia (2021)
57. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1486–1494 (2019)
58. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
59. Zhao, L., Mo, Q., Lin, S., Wang, Z., Zuo, Z., Chen, H., Xing, W., Lu, D.: Uctgan: Diverse image inpainting based on unsupervised cross-space translation. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5740–5749 (2020)

60. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I.C., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. ArXiv **abs/2103.10428** (2021)
61. Zheng, C., Cham, T., Cai, J.: Pluralistic image completion. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1438–1447 (2019)
62. Zhou, X., Li, J., Wang, Z., He, R., Tan, T.: Image inpainting with contrastive relation network. 2020 25th International Conference on Pattern Recognition (ICPR) pp. 4420–4427 (2021)
63. Zhuang, C., Zhai, A., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 6001–6011 (2019)