Generative Hierarchical Features from Synthesizing Images

Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, Bolei Zhou
The Chinese University of Hong Kong
{xy119, sy116, jzhu, yc019, bzhou}@ie.cuhk.edu.hk

Abstract

Generative Adversarial Networks (GANs) have recently advanced image synthesis by learning the underlying distribution of observed data in an unsupervised manner. However, how the features trained from solving the task of image synthesis are applicable to visual tasks remains seldom explored. In this work, we show that learning to synthesize images is able to bring remarkable hierarchical visual features that are generalizable across a wide range of visual tasks. Specifically, we consider the pre-trained StyleGAN generator as a learned loss function and utilize its layer-wise disentangled representation to train a novel hierarchical encoder. As a result, the visual feature produced by our encoder, termed as Generative Hierarchical Feature (GH-Feat), has compelling discriminative and disentangled properties, facilitating a range of both discriminative and generative tasks. Extensive experiments on face verification, landmark detection, layout prediction, transfer learning, style mixing, and image editing show the appealing performance of the GH-Feat learned from synthesizing images, outperforming existing unsupervised feature learning methods.\(^1\)

1 Introduction

Representation learning plays an essential role in the rise of deep learning. The learned deep representation is able to express the variation factors of the complex visual world around us. Accordingly, the performance of a deep learning algorithm highly depends on the features extracted from the input data. As pointed out by Bengio et al. [4], a good representation (or feature) is expected to have the following properties. First, it should be able to capture multiple configurations from the input. Second, it should organize the explanatory factors of the input data as a hierarchy, where more abstract concepts are at a higher level and less abstract concepts at a lower level. Third, it should has strong transferable ability, not only from dataset to dataset but also from task to task.

Deep neural networks supervisedly trained for image classification on large-scale datasets (e.g., ImageNet [7] and Places [53]) have resulted in expressive and discriminative visual features [46, 40]. However, the developed features are heavily dependent on the training objective. For example, some work on feature interpretation has shown that deep features trained for object recognition task may mainly focus on the shapes and parts of the objects while remain invariant to rotation [1, 35], and the deep features from a scene classification model may focus more on detecting the categorical objects (e.g., bed for bedroom and sofa for living room) [52]. Thus the discriminative features learned from solving high-level image classification tasks might not be necessarily good for other mid-level and low-level visual tasks such as landmark detection and layout prediction. Besides, it remains unknown how the discriminative features can facilitate generative tasks like image editing.

\(^{\ast}\)denotes equal contribution.

\(^{1}\)Code and models can be found at \url{https://genforce.github.io/ghfeat/}.
In recent years, generative modeling has made great progress in synthesizing photo-realistic images. Generative Adversarial Networks (GANs) [13] consider the image generation task as training supervision, which is totally different from previous discriminative tasks such as image classification on ImageNet. GAN aims at learning the underlying distribution of real data and further reproducing such distribution by synthesizing high-quality fake data. Through adversarial training, the generator is able to capture the variations underlying the input data to the most extent, otherwise, the discrepancy between the real and synthesized distributions would be spotted by the discriminator. The recent state-of-the-art StyleGAN [26] has been shown to encode rich hierarchical semantics in its layer-wise representations [26, 45, 41]. However, the generator is primarily designed for image generation and hence lacks the ability of taking an image as the input and extracting image features. To solve this problem, prior work [10, 11, 9, 37] proposed to learn an encoder together with GAN such that the generator, discriminator and encoder are jointly optimized. But the introduction of the additional encoder may affect the adversarial balance between the generator and the discriminator, and therefore affect the learned representation as well as the synthesis quality. Furthermore, existing encoders typically project the images onto the very first latent space of the generator (i.e., the most abstract level), yet omitting the hierarchical representation learned by GANs.

In this work, we aim at showing that the pre-trained GAN generator can be considered as a learned multi-scale loss. Training with it can bring highly competitive hierarchical and disentangled visual features with strong generalization ability across various tasks. Based on the StyleGAN model, we tailor a novel hierarchical encoder whose outputs align with the layer-wise representations from the generator. In particular, the fixed generator takes the feature hierarchy produced by the encoder as per-layer inputs and trains the encoder with the supervision of reconstructing the input image. We evaluate such visual features, termed as Generative Hierarchical Features (GH-Feat), on a wide range of visual tasks, including face verification, landmark detection, layout prediction, transfer learning, style mixing, and image editing. Extensive experiments validate that GH-Feat has compelling discriminative and disentangled properties and can be transferred to both generative and discriminative tasks with minor effort, surpassing existing unsupervised feature learning approaches.

2 Related Work

Visual Features. Visual Features play a fundamental role in computer vision. Traditional methods used manually designed features [34, 3, 6], for pattern matching and object detection. These features are significantly improved by deep models [29, 42, 15]. However, the features supervely learned for a particular task could be biased to the training task and hence become difficult to transfer to another task, especially when the target task is too far away from the base task [46]. Furthermore, it is not guaranteed that the features trained for high-level vision tasks such as object classification perform superior on mid-level vision tasks such as pose estimation as well. On the other hand, unsupervised representation learning is widely explored to learn a more general and transferable feature [8, 49, 44, 12, 19, 56]. The very recent MoCo [16] introduces a dynamic dictionary into the contrastive feature learning framework [36, 17, 43] and achieves state-of-the-art performance on ImageNet classification [7]. However, most of the existing unsupervised/self-supervised feature learning methods focus on evaluating their features on the tasks of image recognition (ImageNet [7]) and object detection (COCO [32]), yet seldom evaluate them on other mid-level or low-level tasks, let alone generative tasks. From this viewpoint, the transferability of these features are not fully verified.

Generative Adversarial Networks. GANs [13] are able to produce photo-realistic images via learning the underlying data distribution. The quality of synthesized images are significantly enhanced by the recent advance of GANs [38, 25, 5]. StyleGAN [26] proposes a style-based generator which employs multi-level style codes for image synthesis and achieves the start-of-the-art generation performance. However, the study of GANs remains in the stage of improving the synthesis quality. Little work explores how to apply the representation learned by GANs for other applications. Some recent work interprets the semantics encoded in the internal representation of GANs and applies them for image editing [22, 41, 2, 14, 45, 55]. But it remains much less explored whether the learned GAN representations are transferable to other discriminative tasks.

Adversarial Representation Learning. The main reason of hindering GANs from being applied to discriminative tasks comes from the lack of inference ability. To fill this gap, prior work introduces an additional encoder to the GAN structure [10, 11], similar to Autoencoder [18, 28, 30]. Donahue and Simonyan [9] and Pidhorskyi et al. [37] further extend this idea to the state-of-the-art BigGAN [5] and
3 Methodology
3.1 Layer-wise Representation from StyleGAN

The generator $G(\cdot)$ of GANs typically takes a latent code $z \in \mathcal{Z}$ as the input and is trained to synthesize a photo-realistic image $x = G(z)$. Recent state-of-the-art StyleGAN [26] first maps $z$ to a disentangled space $W$ with $w = f(z)$. Here, $f(\cdot)$ denotes the mapping implemented by Multi-Layer Perceptron. The $w$ code is then projected to layer-wise style codes $\{y^{(\ell)}\}_{\ell=1}^L \triangleq \{(y_s^{(\ell)}, y_b^{(\ell)})\}_{\ell=1}^L$ with affine transformations, where $L$ is the number of convolutional layers. $y_s^{(\ell)}$ and $y_b^{(\ell)}$ correspond to the scale and weight parameters in Adaptive Instance Normalization (AdaIN) [21]. These style codes are used to modulate the output feature maps of each convolutional layer with

$$
\text{AdaIN}(x_i^{(\ell)}, y^{(\ell)}) = y_s^{(\ell)} \frac{x_i^{(\ell)} - \mu(x_i^{(\ell)})}{\sigma(x_i^{(\ell)})} + y_b^{(\ell)},
$$

where $x_i^{(\ell)}$ indicates the $i$-th channel of the output feature map from the $\ell$-th layer. $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and variance respectively.

Prior work [26, 45] has demonstrated the disentanglement property of the layer-wise representation learned by StyleGAN such that by manipulating the $w$ code at different layers, we can change different levels of semantics occurring in the output image. Considering the simple affine transformation from $w$ to $\{y^{(\ell)}\}_{\ell=1}^L$, we treat $\{y^{(\ell)}\}_{\ell=1}^L$ as the generative visual features that we would like to extract from the input image. There are mainly two reasons in doing so. First, the synthesis $x$ is purely determined by these style codes without any other variations. Hence, these codes should be able to best express $x$ under the knowledge of the fixed generator. Second, these style codes possess an excellent property of disentangled hierarchy. To the best of our knowledge, this is the first work that proposes to treat the style codes for AdaIN as the learned generative representations of GANs.

3.2 Hierarchical Encoder Trained with StyleGAN Loss

To better reuse the layer-wise representation learned by StyleGAN, we propose to train a hierarchical encoder $E(\cdot)$ by treating the pretrained generator $G(\cdot)$ as a learned loss. As shown in Fig.1, the encoder is trained to extract Generative Hierarchical Features (GH-Feat) $\{f^{(\ell)}\}_{\ell=1}^L \triangleq \{(f_s^{(\ell)}, f_b^{(\ell)})\}_{\ell=1}^L$ from the input images. These features are then fed into the per-layer AdaIN module of the generator by replacing the style code $y^{(L-\ell)}$ in Eq.(1).
**Encoder Structure.** For the hierarchical encoder, we adopt a block-wise architecture consisting of six residual blocks. As shown in Fig.1, the output feature maps from shallow blocks are mapped to low-level features, which are fed into top layers of the fixed generator. The detailed structure of the encoder can be found in Sec.A.

**Training Objective.** The encoder is trained with the following objectives. First, we regard the well-trained generator as a learned loss and use it as the training supervision. Concretely, we hope the GH-Feat produced by the encoder can best represent the input image such that this feature can be used to reconstruct the input image when fed into the generator $G(\cdot)$. Second, to further improve the quality of the reconstructed image, we involve the discriminator to compete with the encoder. Third, we include the perceptual loss [24] as a regularizer. To summarize, the encoder $E(\cdot)$ and the discriminator $D(\cdot)$ are jointly trained with

$$
\min_{\Theta_E} \mathcal{L}_E = ||x - G(E(x))||_2 - \lambda_1 \mathbb{E}_x [D(G(E(x)))] + \lambda_2 ||F(x) - F(G(E(x)))||_2,
$$

$$
\min_{\Theta_D} \mathcal{L}_D = \mathbb{E}_x [D(G(E(x)))] - \mathbb{E}_x [D(x)] + \lambda_3 \mathbb{E}_x [||\nabla_x D(x)||_2^2],
$$

where $\lambda_1$ and $\lambda_2$ are loss weights, while $\lambda_3$ is the hyper-parameter for gradient regularization. $|| \cdot ||_2$ denotes the $\ell_2$ norm and $F(\cdot)$ stands for perceptual feature extraction. Note that different from prior work [10, 11, 9, 37] that jointly trains the generator and the encoder, we treat the well-trained generator as a learned loss function. Hence, the generator is fixed in the entire training process. Meanwhile, the discriminator is fine-tuned from the original one when the generator is trained.

4 Experiments

In this section, we evaluate the superiority of the Generative Hierarchical Features (GH-Feat) on a wide range of discriminative and generative visual tasks. Sec.4.1 introduces the experimental settings, such as implementation details, models, and tasks. Sec.4.2 studies the discriminative and disentangled properties of GH-Feat. Sec.4.3 evaluates how GH-Feat can be applied to generative tasks.

4.1 Experimental Settings

**Implementation Details.** The hyper-parameters in Eq.(2) and Eq.(3) are set as $\lambda_1 = 0.1$, $\lambda_2 = 5 \times 10^{-5}$, and $\lambda_3 = 5$. We use the layer conv4_3 of the VGG [42] model to compute the perceptual loss in Eq.(2). Adam [27] with hyper-parameters $\beta_1 = 0$, $\beta_2 = 0.99$, and $\epsilon = 10^{-8}$ is used to optimize the parameters of both the encoder and the discriminator. The learning rate is initially set as $1 \times 10^{-4}$ and exponentially decayed with the factor of 0.8.

**Datasets and Models.** We conduct experiments on three StyleGAN [26] models, pre-trained on MNIST [31], FF-HQ faces [26], and LSUN bedrooms [47] respectively. The model on MNIST is used to evaluate the image classification performance, following prior work [10, 37]. The model on FF-HQ is used for multi-level discriminative tasks, including pose (yaw) estimation, landmark detection, face verification, as well as generative tasks. The model on LSUN bedrooms is evaluated on the mid-level discriminative task (i.e., layout prediction) and the generative task (i.e., image editing).

**Tasks and Metrics.** Unlike existing adversarial feature learning methods [11, 10, 37, 9] that are mainly evaluated on the high-level image classification task, we benchmark the learned GH-Feat on both generative tasks and discriminative tasks from multiple levels. Here, we briefly introduce the discriminative tasks used in this work and the corresponding evaluation metrics. (i) **MNIST digit recognition.** It is a classical image classification task. We report the Top-1 accuracy on the test set following [31]. (ii) **Pose Estimation.** This task targets at estimating the yaw pose of the input face. 70,000 real faces on FF-HQ [26] are split to 60,000 training samples and 10,000 test samples. The $\ell_1$ regression error is used as the evaluation metric. (iii) **Landmark Detection.** This task learns a set of semantic points with visual meaning. We use FF-HQ dataset and follow the standard MSE metric [51] to report performances in inter-ocular distance (IOD). (iv) **Face Verification.** It aims at distinguishing whether the given image pair is original from the same identity. We validates on the LFW dataset [20] following the standard protocol [20]. (v) **Layout Prediction.** In this work, we extract the corner points of the layout line and convert the task to a landmark regression task, whose settings are same as the facial landmark detection task. The annotations of the collected 90,000 bedroom images are obtained by the state-of-the-art layout line estimation method [50] This image collection is then split into a training set with 70,000 images and a test set with 20,000 images. Following [57], we report the corner distance as the metric.
Figure 2: Performance on different discriminative tasks using GH-Feat. Left three columns show the comparison between using different representations of the generator as the training supervision. Using style codes $y$ (red) results in a much stronger discriminative and disentangled visual features compared to using the $w$ code in $W$ space (blue). See Sec.4.2.1 for details. The last column compares the two different strategies used in the face verification task, which is explained in Sec.4.2.2. Higher level corresponds to more abstract feature.

Table 1: Quantitative comparison between our proposed GH-Feat and other alternatives on different tasks.

| (a) Digit recognition on MNIST. | (b) Face verification on LFW. | (c) Landmark detection on MAFL. |
|--------------------------------|-----------------------------|-------------------------------|
| Methods                        | Top-1 Acc.                  | Methods                       | Acc.             | Method                        | MSE | Label |
| AE($\ell_1$) [18]              | 97.43                       | VAE [28]                      | 49.3            | TCDCN [51]                    | 7.95 | ✔    |
| AE($\ell_2$) [18]              | 97.37                       | MoCo-R50 [16]                 | 48.9            | MTCNN [48]                    | 5.39 | ✔    |
| BiGAN [10]                     | 97.14                       | Ours Grouping                 | 60.1            | Cond. ImGen [23]              | 4.95 | ✗    |
| ALAE [37]                      | 97.61                       | Ours Layer-wise               | 67.5            | MoCo-R50 [16]                 | 9.07 | ✗    |
| Ours                           | 99.06                       | Ours Voting                   | 69.7            | Ours                          | 5.12 | ✗    |

4.2 Evaluation on Discriminative Tasks

In this part, we show that even the proposed GH-Feat is learned from generative modeling, it shows the strong discriminative power as well as the compelling disentangled property. To best evaluate the GH-Feat learned by the hierarchical encoder, we do not fine-tune the encoder for any certain task. Instead, for the tasks of pose estimation, landmark detection, and layout prediction, we use our encoder to extract visual features from both the training and the test sets. Then we learn a simple linear regression model on the training set with ground-truth and evaluate the regression model on the test set.

4.2.1 Comparison between $w \in W$ and Style Codes $y$ in StyleGAN

Since the proposed hierarchical encoder is trained by employing the fixed StyleGAN generator as the learned loss function, its performance is highly dependent on the internal representation from the StyleGAN model. As mentioned in Sec.3.1, unlike prior work [37, 54, 55, 41, 45] that treats the $W$ space as the representation space, we are the first to use the style codes $y$ instead. Here, we would like to first show why we choose the style codes $y$ over the latent code $w$. Note that StyleGAN feeds the same $w$ code to all layers in the generator, but the style codes $y$ are different among layers. For a fair comparison, we relax the constraint of using same $w$ for all layers, following prior work [55]. We make comparisons from two aspects, i.e., discriminative ability and disentangled property.

**Discriminative Ability.** We conduct experiments on pose (yaw) estimation, landmark detection, and layout prediction to compare the performance by using these two sets of codes ($i.e., w$ and $y$) as the supervision. Fig.2 gives the quantitative comparison results. We can see that on both three tasks, the proposed GH-Feat trained using styles codes $y$ as the supervision (in red lines) achieves better performances than training with $w$ as the supervision (in blue lines) on all three tasks.

**Disentangled Property.** We further evaluate the different levels of GH-Feat on these discriminative tasks. Specifically, for a particular task, we train the linear regression model on the learned hierarchical visual features level by level. Quantitative results in Fig.2 suggests that the performances increase significantly with the level index increasing. It matches the conclusion of [45] where the early layers of generator ($i.e.,$ the deeper level of encoder) tend to encode structural information such as face orientation and geometric landmarks. This is because we treat the StyleGAN generator as the training supervision. In this way, the visual features extracted by our encoder can best match the native representation that has already learned by StyleGAN. The disentanglement property is further analyzed in Sec.4.3.2 on generative tasks.
4.2.2 Image Classification & Face Verification

Image classification is widely used to evaluate the performance of learned representations [19, 56, 16, 36, 9]. ImageNet [7] is typically used as the benchmark. However, recall that we reuse the well-trained StyleGAN generator as the training supervision. There are no available StyleGAN model trained on the large-scale ImageNet, hence we do not test our GH-Feat on ImageNet classification. Instead, we do evaluation on the face verification task, which has far more identities than the 1,000 categories in ImageNet. A larger StyleGAN model may bring a more ImageNet-friendly representation.

MNIST Digit Recognition. We first show a toy example on MNIST, following prior work [10, 37]. The Top-1 accuracy is reported in Tab.1(a). Our GH-feat outperforms ALAE [37] and BiGAN [10] with 1.45% and 1.92%, suggesting a better discriminative power than other competitors.

LFW Face Verification. We directly use the proposed encoder to extract GH-Feat from face images in LFW [20] and tries three different strategies in exploiting the GH-Feat for face verification: (i) using single level GH-Feat; (ii) grouping multi-level (from the bottom level) GH-Feat together; (iii) voting by choosing the largest face similarity (between two faces for verification) from all levels. Fig.2 (last column) shows the results from the first two strategies. Obviously, GH-Feat from the 4-th to 9-th levels best preserve the identity information. Tab.1(b) compares GH-Feat with other unsupervised feature learning methods, including VAE [28] and MoCo [16], which are also trained on FF-HQ dataset. ResNet-50 [15] is employed as the backbone for MoCo. Our method with voting strategy achieves 69.7% accuracy, surpassing other competitors by a large margin. We also visualize some reconstructed LFW faces in Fig.3, where our GH-Feat well handles the significant domain gap (e.g. image resolution) and preserves the identity information.

4.2.3 Transfer Learning on Multiple Vision Tasks

We explore the transferability of GH-Feat on facial landmark detection and scene layout prediction.

Landmark Detection. We train a linear regression model using GH-Feat on FF-HQ [26] and test it on MAFL [51]. This two datasets have a large domain gap such that faces in MAFL have larger poses yet lower image quality. As shown in Fig.4, the proposed GH-Feat shows a strong generalization ability between these two datasets. We also compare our approach with some supervised alternatives [51, 48], the state-of-the-art unsupervised landmark detection method [23], and the unsupervised
representation learning method (i.e., MoCo [16]) The comparison results are reported in Tab.1(c), where our GH-Feat achieves comparable performance as the models that are particularly designed for this task. Meanwhile, we significantly outperforms MoCo on this mid-level vision task.

**Layout Prediction.** We train the layout predictor on LSUN [47] bedrooms and test it on kitchens to validate how a feature can be transferred from one scene category to another. Feature learned by MoCo [16] on the bedroom dataset is used for comparison. Some qualitative examples are shown in Fig.5. We can tell that our learned GH-Feat shows much better predictions than the feature from MoCo on both the base dataset (bedrooms) and the target dataset (kitchens). That is because existing representation approaches mainly focus on the high-level image classification tasks instead of other mid-level and low-level visual tasks. By contrast, GH-Feat show stronger transferability.

### 4.3 Evaluation on Generative Tasks

Besides the compelling disentangled property, another advantage of our proposed GH-Feat over existing unsupervised feature learning approaches [19, 56, 36, 43, 16] is its generative capability. More concretely, existing methods are primarily designed for the image classification task, but we propose to use the well-trained StyleGAN generator as a learned loss. Therefore, our learned representation, i.e., GH-Feat, well supports sampling. In this section, we conduct some generative tasks to verify this point.

#### 4.3.1 Image Reconstruction

Image reconstruction is an important evaluation on whether the learned features can best represent the input image. The very recent work ALAE [37] also employs StyleGAN for representation learning. We have two main differences from ALAE: (i) We treat the StyleGAN generator as a learned loss function while ALAE optimizes the generator together with the encoder. (ii) We propose to learn hierarchical features, which can capture the multi-level variation factors from the input to the most extent. Fig.6 shows some test samples which we borrow from the ALAE paper [37]. We can tell that our GH-Feat can better reconstruct the input, demonstrating its expressiveness.

#### 4.3.2 Style Mixing

We further test the learned GH-Feat on the style mixing task. Specifically, we use the encoder to extract hierarchical visual features from both the content image and the style image and swap these two features at some particular level. The swapped features are then visualized by the generator, as shown in Fig.7 where we can observe the obvious disentangled property. For example, by exchanging low-level features, only the image color tone and the skin color are changed. Meanwhile, mid-level features controls the expression, age, or even hair styles. Finally, the highest level feature is able to transit the pose information from the style image to the content image (see last row in Fig.7). This experiments demonstrates the strong disentangled property of GH-Feat, leading to the same conclusion in Sec.4.2.1.
4.3.3 Image Editing

As mentioned above, one appealing advantage of the feature learned from synthesizing images is its generative ability. Specially, our GH-Feat can not only extract expressive visual representations from the input images, but also supports sampling, enabling a lot of image editing applications.

Global Editing. The style mixing results in Fig.7 have already suggested the potential of our GHFeat in multi-level image stylization. However, sometimes we do not have a target style image to be used as the reference. Thanks to the latent space in the generator, our GH-Feat shows a much stronger creative capability such that we can freely sample meaningful visual features and use them for image editing. Fig.8 suggests that the sampled features also lead to high-fidelity editing results from multiple levels. This benefits from the matching between the learned GH-Feat with the internal representation of the StyleGAN generator.

Local Editing. Besides global editing, our GH-Feat also facilitates editing the target image locally by deeply cooperating with the generator. In particular, instead of directly swapping features, we can exchange a certain region of the spatial feature map at some particular level. In this way, only a local patch in the output image will be modified while other parts remain untouched. As shown in Fig.9, we can successfully “revise” the input face with different eyes, noses, and mouths.

Content Harmonization. Our hierarchical encoder is robust such that it can extract reasonable visual features even from unreasonable image content. We copy some patches (e.g., bed and window) onto a bedroom image and feed the stitched image into our proposed encoder for feature extraction. The extracted features are then visualized via the generator, as in Fig.10. We can see that the copied patches well blend into the “background”. We also surprisingly find that when copying a window into the source image, the view from the original window and that from the new window highly align with each other (e.g., vegetation or ocean), benefiting from the hierarchical feature extraction.

5 Conclusion

In this work, we consider the well-trained GAN generator as a learned loss function for feature learning. Training with it brings competitive Generative Hierarchical Features that can be generalized to a wide range of vision tasks, surpassing existing unsupervised representation learning approaches.
Broader Impact

Different from most works in the community, which merely use GANs for image generation, this work shows the potential of using GANs to learn a generalizable representation by treating a well-learned GAN generator as a loss function. It has two main impacts. First, representation learning is critical to deep learning algorithms. With the discovery from this work, more GAN-based frameworks, which are specially designed for better unsupervised representation learning, can be expected. Second, this work shows the great power by using a learned GAN generator as the training supervision. This may result in a more general objective function, just like the existing commonly used perceptual loss which is computed based on discriminative models.

References

[1] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In CVPR, 2017.
[2] 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 ICLR, 2019.
[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In ECCV, 2006.
[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. TPAMI, 2013.
[5] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In ICLR, 2019.
[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009.
[8] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. In ICCV, 2017.
[9] J. Donahue and K. Simonyan. Large scale adversarial representation learning. In NeurIPS, 2019.
[10] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In ICLR, 2017.
[11] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. In ICLR, 2017.
[12] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In ICLR, 2018.
[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NeurIPS, 2014.
[14] J. Gu, Y. Shen, and B. Zhou. Image processing using multi-code gan prior. In CVPR, 2020.
[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
[16] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, 2020.
[17] O. J. Hénaff, A. Srinivas, J. De Fauw, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272, 2019.
[18] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 2006.
[19] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In ICLR, 2019.

[20] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[21] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In ICCV, 2017.

[22] A. Jahanian, L. Chai, and P. Isola. On the "steerability" of generative adversarial networks. In ICLR, 2020.

[23] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In NeurIPS, 2018.

[24] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016.

[25] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In ICLR, 2018.

[26] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In CVPR, 2019.

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

[28] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In ICLR, 2014.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NeurIPS, 2012.

[30] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In ICML, 2016.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In ICCV, 2015.

[34] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 2004.

[35] D. Matthew and R. Fergus. Visualizing and understanding convolutional neural networks. In CVPR, 2014.

[36] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[37] S. Pidhorskyi, D. Adjeroh, and G. Doretto. Adversarial latent autoencoders. In CVPR, 2020.

[38] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.

[39] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, 2015.

[40] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In CVPR Workshop, 2014.

[41] Y. Shen, C. Yang, X. Tang, and B. Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. arXiv preprint arXiv:2005.09635, 2020.
[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[43] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. arXiv preprint arXiv:1906.05849, 2019.

[44] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In CVPR, 2018.

[45] C. Yang, Y. Shen, and B. Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. arXiv preprint arXiv:1911.09267, 2019.

[46] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In NeurIPS, 2014.

[47] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365, 2015.

[48] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 2016.

[49] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In CVPR, 2017.

[50] W. Zhang, W. Zhang, and J. Gu. Edge-semantic learning strategy for layout estimation in indoor environment. Transactions On Cybernetics, 2019.

[51] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In ECCV, 2014.

[52] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In ICLR, 2015.

[53] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. TPAMI, 2017.

[54] J. Zhu, D. Zhao, and B. Zhang. Lia: Latently invertible autoencoder with adversarial learning. arXiv preprint arXiv:1906.08090, 2019.

[55] J. Zhu, Y. Shen, D. Zhao, and B. Zhou. In-domain gan inversion for real image editing. arXiv preprint arXiv:2004.00049, 2020.

[56] C. Zhuang, A. L. Zhai, and D. Yamins. Local aggregation for unsupervised learning of visual embeddings. In ICCV, 2019.

[57] C. Zou, A. Colburn, Q. Shan, and D. Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In CVPR, 2018.
Appendix

The appendix is organized as follows. Sec.A describes the detailed structure of the proposed hierarchical encoder. Sec.B conducts image retrieval experiments based on Generative Hierarchical Features (GH-Feat). Sec.C compares our approach with state-of-the-art “encoder-decoder” based representation learning method, i.e. ALAE [37], from the style mixing aspect.

A Encoder Structure

Tab.2 illustrates the architecture of our hierarchical encoder in details. Note that a StyleGAN [26] model trained to synthesize images with $256 \times 256$ resolution has 14 convolutional layers in total. Hence, our GH-Feat consists of 14 levels correspondingly. To obtain such multi-level representation, we employ a model with multiple residual blocks [15]. Different from U-Net [39], we only utilize the latter stage of the residual block to get GH-Feat. For example, the 4-th residual block is used to produce GH-Feat from Level 0 to Level 5.

| Stage | Block | Shape | GH-Feat |
|-------|-------|-------|---------|
| Input | $\times$1 | $3 \times 256^2$ |
| Residual 1 | $3 \times 3$, 64 | $3 \times 3$, 64 | $64 \times 128^2$ |
| Residual 2 | $3 \times 3$, 128 | $3 \times 3$, 128 | $128 \times 64^2$ |
| Residual 3 | $3 \times 3$, 256 | $3 \times 3$, 256 | $256 \times 32^2$ |
| Residual 4 | $3 \times 3$, 512 | $3 \times 3$, 512 | $512 \times 16^2$ |
| Residual 5 | $3 \times 3$, 1024 | $3 \times 3$, 1024 | $1024 \times 8^2$ |
| Residual 6 | $3 \times 3$, 1024 | $3 \times 3$, 1024 | $1024 \times 4^2$ |

B Image Retrieval

In this section, we verify the discriminative and disentanglement property of the proposed GH-Feat with image retrieval. Concretely, given a query image, we use encoder to extract its GH-Feat. Then, we use different levels of GH-Feat to perform retrieval from $10^K$ real images (GH-Feat of these $10^K$ images are prepared in advance). $\ell_1$ distance is used as the metric for retrieval.

Fig.11 and Fig.12 show the retrieval results on MNIST [31] and LSUN bedroom [47] respectively. From Fig.11, we observe that no matter which level of features is used, the retrieved results are with the same digit as the query image, suggesting that GH-Feat can well encode the categorical information. Compared to MNIST, LSUN bedroom dataset is much more diverse due to the semantic hierarchy [45] contained in the images. As shown in Fig.12, when we use higher level (first row) features for retrieval, all retrieved results are with the same layout as the query image, but they may have different lighting conditions. On the contrary, when using lower level (bottom row) features for retrieval, the retrieved results are with similar lighting condition as the query image. This benefits from the discriminative power of the learned hierarchical feature, i.e., GH-Feat. It also aligns with the conclusion in Sec.4.2.1 of the main submission.

C Style Mixing

In this part, we compare our approach with ALAE [37] on the task of style mixing. ALAE is recently proposed for unsupervised representation learning by also introducing an encoder to the StyleGAN [26] structure. We mainly have two differences from ALAE: (i) ALAE trains the encoder and the decoder (generator) simultaneously while we treats the well-trained StyleGAN model as a learned loss function. (ii) Our GH-Feat is able to extract multi-level features from the input images. For a fair
comparison, we use ALAE and our approach to extract features from the same images (including both style images and content images) and then use the extracted features for style mixing.

Fig.13 shows the comparison results. Note that all test images are selected following the original paper of ALAE [37]. We can see that when mixing high-level features from style images (top two rows), the pose, age, and gender of mixed results are close to those of style images. By comparing with ALAE, results using GH-Feat better preserve the identity information (high-level feature) from style images as well as the color information (low-level feature) from content images. In addition, when mixing low-level features from style images (bottom two rows), both ALAE and GH-Feat can successfully transfer the color style from style images to content images, but GH-Feat shows much stronger identity preservation. This experiment demonstrates the superiority of GH-Feat over ALAE in learning accurate and disentangled representation.
Figure 12: Retrieval results on LSUN bedroom [47]. Higher level corresponds to more abstract feature.
After extracting features from both content images and style images, we replace different levels of features from content images with those from style images.

Figure 13: Qualitative comparison between our proposed GH-Feat and ALAE [37] on the style mixing task.