Generative Hierarchical Features from Synthesizing Images

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

Abstract

Generative Adversarial Networks (GANs) have recently advanced image synthesis by learning the underlying distribution of the observed data. However, how the features learned from solving the task of image generation are applicable to other vision tasks remains seldom explored. In this work, we show that learning to synthesize images can bring remarkable hierarchical visual features that are generalizable across a wide range of applications. Specifically, we consider the pre-trained StyleGAN generator as a learned loss function and utilize its layer-wise representation to train a novel hierarchical encoder. The visual feature produced by our encoder, termed as Generative Hierarchical Feature (GH-Feat), has strong transferability to both generative and discriminative tasks, including image editing, image harmonization, image classification, face verification, landmark detection, and layout prediction. Extensive qualitative and quantitative experimental results demonstrate the appealing performance of GH-Feat.[1]

1. Introduction

Representation learning plays an essential role in the rise of deep learning. The learned representation is able to express the variation factors of the complex visual world. 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 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. Third, it should have strong transferability, not only from datasets to datasets but also from tasks to tasks.

Deep neural networks supervisedly trained for image classification on large-scale datasets (*e.g.*, ImageNet [8] and Places [66]) have resulted in expressive and discriminative visual features [46]. However, the developed features are heavily dependent on the training objective. For example, prior work has shown that deep features trained for the object recognition task may mainly focus on the shapes and parts of the objects while remain invariant to rotation [1, 41], 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) [65]. Thus the discriminative features learned from solving high-level image classification tasks might not be necessarily good for other mid-level and low-level tasks, limiting their transferability [57, 64]. Besides, it remains unknown how the discriminative features can be used in generative applications like image editing.

Generative Adversarial Network (GAN) [17] has recently made great process in synthesizing photo-realistic images. It considers the image generation task as the training supervision to learn the underlying distribution of real data. Through adversarial training, the generator can capture the multi-level variations underlying the input data to the most extent, otherwise, the discrepancy between the real and synthesized data would be spotted by the discriminator. The recent state-of-the-art StyleGAN [31] has been shown to encode rich hierarchical semantics in its layer-wise representations [31, 55, 47]. However, the generator is primarily designed for image generation and hence lacks the inference ability of taking an image as the input and extracting its visual feature, which greatly limits the applications of GANs to real images.

To solve this problem, a common practice is to introduce an additional encoder into the two-player game described in GANs [12, 15, 13, 44]. Nevertheless, existing encoders typically choose the initial latent space (*i.e.*, the most abstract level feature) as the target representation space, omitting the pre-layer information learned by the generator. On the other hand, the transferability of the representation from GAN models is not fully verified in the literature. Most prior work focuses on learning discriminative features for the high-level image classification task [12, 15, 13] yet put little effort on other mid-level and low-level downstream tasks, such as landmark detection and layout prediction.

*denotes equal contribution.
[1]Project page is at https://genforce.github.io/ghfeat/.

In this work, we show that the pre-trained GAN generator can be considered as a learned loss function. Training with it can bring highly competitive hierarchical visual features which are generalizable to 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 generator takes the feature hierarchy produced by the encoder as the per-layer inputs and supervises the encoder via reconstructing the input image. We evaluate such visual features, termed as *Generative Hierarchical Features (GH-Feat)*, on both generative and discriminative tasks, including image editing, image harmonization, image classification, face verification, landmark detection, layout prediction, *etc*. Extensive experiments validate that the generative feature learned from solving the image synthesis task has compelling hierarchical and transferable properties, facilitating many downstream applications.

2. Related Work

Visual Features. Visual Feature plays a fundamental role in the computer vision field. Traditional methods used manually designed features [40, 3, 7] for pattern matching and object detection. These features are significantly improved by deep models [34, 49, 20], which automatically learn the feature extraction from large-scale datasets. However, the features supervisedly learned for a particular task could be biased to the training task and hence become difficult to transfer to other tasks, especially when the target task is too far away from the base task [57, 64]. Unsupervised representation learning is widely explored to learn a more general and transferable feature [10, 61, 53, 16, 24, 68, 19, 42, 21, 51]. However, most of existing unsupervised feature learning methods focus on evaluating their features on the tasks of image recognition, yet seldom evaluate them on other mid-level or low-level tasks, let alone generative tasks. Shocher *et al.* [48] discover the potential of discriminative features in image generation, but the transferability of these features are still not fully verified.

Generative Adversarial Networks. GANs [17] are able to produce photo-realistic images via learning the underlying data distribution. The recent advance of GANs [45, 30, 5] has significantly improved the synthesis quality. StyleGAN [31] proposes a style-based generator with multi-level style codes and achieves the start-of-the-art generation performance. However, little work explores the representation learned by GANs as well as how to apply such representation for other applications. Some recent work interprets the semantics encoded in the internal representation of GANs and applies them for image editing [27, 47, 2, 18, 55, 67]. But it remains much less explored whether the learned GAN representations are transferable to 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 [12, 15]. Donahue and Simonyan [13] and Pidhorskyi *et al.* [44] extend this idea to the state-of-the-art BigGAN [5] and StyleGAN [31] models respectively. In this paper, we also study the representation learning using GANs, with following **improvements** compared to existing methods. First, we propose to treat the well-trained StyleGAN generator as *a learned loss function*. Second, instead of mapping the images to the initial GAN latent space, like most algorithms [12, 15, 13, 44] have done, we design a novel encoder to produce *hierarchical* features that well align with the layer-wise representation learned by StyleGAN. Third, besides the image classification task that is mainly targeted at by prior work [12, 15, 13, 44], we validate the *transferability* of our proposed GH-Feat on a range of generative and discriminative tasks, demonstrating its generalization ability.

3. Methodology

We design a novel encoder to extract hierarchical visual features from the input images. This encoder is trained in an unsupervised learning manner from the image reconstruction loss based on a prepared StyleGAN generator. Sec. 3.1 describes how we abstract the multi-level representation from StyleGAN. Sec. 3.2 presents the structure of the novel hierarchical encoder. Sec. 3.3 introduces the idea of using pre-trained StyleGAN generator as a learned loss function for representation learning.

3.1. Layer-wise Representation from StyleGAN

The generator $G(\cdot)$ of GANs typically takes a latent code $\mathbf{z} \in \mathcal{Z}$ as the input and is trained to synthesize a photo-realistic image $\mathbf{x} = G(\mathbf{z})$. The recent state-of-the-art StyleGAN [31] proposes to first map $\mathbf{z}$ to a disentangled space $\mathcal{W}$ with $\mathbf{w} = f(\mathbf{z})$. Here, $f(\cdot)$ denotes the mapping implemented by multi-layer perceptron (MLP). The $\mathbf{w}$ code is then projected to layer-wise style codes $\{\mathbf{y}^{(\ell)}\}_{\ell=1}^{L} \triangleq \{(\mathbf{y}_s^{(\ell)}, \mathbf{y}_b^{(\ell)})\}_{\ell=1}^{L}$ with affine transformations, where L is the number of convolutional layers. $\mathbf{y}_s^{(\ell)}$ and $\mathbf{y}_b^{(\ell)}$ correspond to the scale and weight parameters in Adaptive Instance Normalization (AdaIN) [26]. These style codes are used to modulate the output feature maps of each convolutional layer with

$$\texttt{AdaIN}(\mathbf{x}_i^{(\ell)}, \mathbf{y}^{(\ell)}) = \mathbf{y}_{s,i}^{(\ell)} \frac{\mathbf{x}_i^{(\ell)} - \mu(\mathbf{x}_i^{(\ell)})}{\sigma(\mathbf{x}_i^{(\ell)})} + \mathbf{y}_{b,i}^{(\ell)}, \quad (1)$$

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

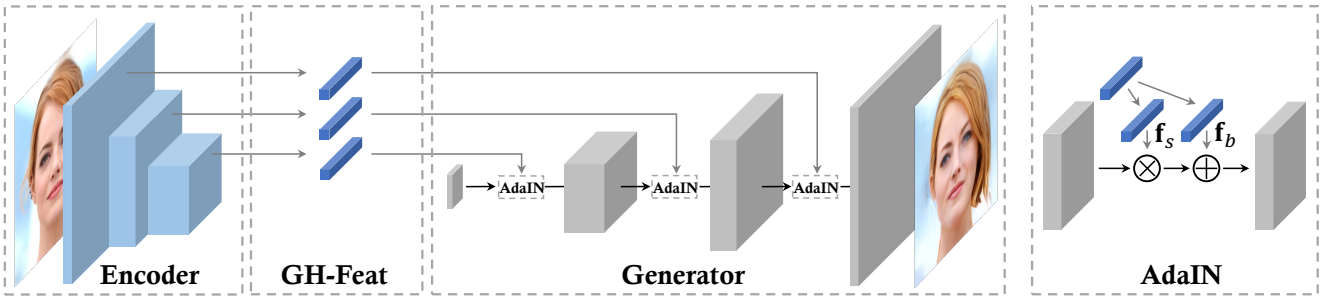


Figure 1. Framework of the proposed encoder, which is able to extract *Generative Hierarchical Features (GH-Feat)* from input image. This feature hierarchy highly aligns with the layer-wise representation (*i.e.*, style codes of per-layer AdaIN) learned by the StyleGAN generator. Parameters in **blue** blocks are trainable.

Here, we treat the layer-wise style codes, $\{\mathbf{y}^{(\ell)}\}_{\ell=1}^{L}$, as the generative visual features that we would like to extract from the input image. There are two major advantages. First, the synthesized image can be completely determined by these style codes without any other variations, making them suitable to express the information contained in the input data from the generative perspective. Second, these style codes are organized as a hierarchy where codes at different layers correspond to semantics at different levels [31, 55]. To the best of our knowledge, this is the first work that adopts the style codes for the per-layer AdaIN module as the learned representations of StyleGAN.

3.2. Hierarchical Encoder

Based on the layer-wise representation described in Sec. 3.1, we propose a novel encoder $E(\cdot)$ with a hierarchical structure to extract multi-level visual features from a given image. As shown in Fig. 1, the encoder is designed to best align with the StyleGAN generator. In particular, the Generative Hierarchical Features (GH-Feat) produced by the encoder, $\{\mathbf{f}^{(\ell)}\}_{\ell=1}^{L} \triangleq \{(\mathbf{f}_s^{(\ell)}, \mathbf{f}_b^{(\ell)})\}_{\ell=1}^{L}$, are fed into the per-layer AdaIN module of the generator by replacing the style code $\mathbf{y}^{(L-\ell+1)}$ in Eq. (1).

We adopt ResNet [20] architecture as the encoder backbone and add an extra residual block to get an additional feature map with lower resolution.[2] Besides, we introduce a feature pyramid network [36] to learn the features from multiple levels. The output feature maps from the last three stages, $\{R_4, R_5, R_6\}$, are used to produce GH-Feat. Taking a 14-layer StyleGAN generator as an instance, R_4 aligns with layer 9-14, R_5 with 5-8, while R_6 with 1-4. Here, to bridge the feature map with each style code, we first downsample it to 4×4 resolution and then map it to a vector of the target dimension using a fully-connect (FC) layer. In addition, we introduce a lightweight Spatial Alignment Module (SAM) [56, 38] into the encoder structure to better capture the spatial information from the input image. SAM works in a simple yet efficient way:

$$R_i = W_i \texttt{down}(R_i) + W_6 R_6 \quad i \in \{4, 5\},$$

where W_4, W_5, and W_6 (all are implemented with an 1×1 convolutional layer) are used to project the feature maps R_4, R_5, and R_6 to have the same number of feature channels respectively. R_4 and R_5 are downsampled to the same resolution of R_6 before fusion. The detailed structure of the encoder can be found in **Supplementary Material**.

3.3. StyleGAN Generator as Learned Loss

We consider the pre-trained StyleGAN generator as a leaned loss function. Specifically, we employ a StyleGAN generator to supervise the encoder training with the objective of image reconstruction. We also introduce a discriminator to compete with the encoder, following the formulation of GANs [17], to ensure the reconstruction quality. To summarize, the encoder $E(\cdot)$ and the discriminator $D(\cdot)$ are jointly trained with

$$\begin{aligned} \min_{\Theta_E} \mathcal{L}_E = &||\mathbf{x} - G(E(\mathbf{x}))||_2 - \lambda_1 \mathbb{E}_{\mathbf{x}}[D(G(E(\mathbf{x})))] \\ &+ \lambda_2 ||F(\mathbf{x}) - F(G(E(\mathbf{x})))||_2, \end{aligned} \tag{2}$$

$$\begin{aligned} \min_{\Theta_D} \mathcal{L}_D = &\mathbb{E}_{\mathbf{x}}[D(G(E(\mathbf{x})))] - \mathbb{E}_{\mathbf{x}}[D(\mathbf{x})] \\ &+ \lambda_3 \mathbb{E}_{\mathbf{x}}[||\nabla_{\mathbf{x}} D(\mathbf{x})||_2^2], \end{aligned} \tag{3}$$

where $||\cdot||_2$ denotes the ℓ_2 norm and $\lambda_1, \lambda_2, \lambda_3$ are loss weights to balance different loss terms. The last term in Eq. (2) represents the perceptual loss [29] and $F(\cdot)$ denotes the `conv4_3` output from a pre-trained VGG [49] model.

4. Experiments

We evaluate Generative Hierarchical Features (GH-Feat) on a wide range of downstream applications. Sec. 4.1 introduces the experimental settings, such as implementation details, datasets, and tasks. Sec. 4.2 conducts ablation study on the proposed hierarchical encoder. Sec. 4.3 and Sec. 4.4 evaluate the applicability of GH-Feat on generative and discriminative tasks respectively.

[2] In fact, there are totally six stages in our encoder, where the first one is a convolutional layer (followed by a pooling layer) and each of the others consists of several residual blocks.

4.1. Experimental Settings

Implementation Details. The loss weights are set as $\lambda_1 = 0.1$, $\lambda_2 = 5e^{-5}$, and $\lambda_3 = 5$. We use Adam [32] optimizer, with $\beta_1 = 0$ and $\beta_2 = 0.99$, to train both the encoder and the discriminator. The learning rate is initially set as $1e^{-4}$ and exponentially decayed with the factor of 0.8.

Datasets and Models. We conduct experiments on four StyleGAN [31] models, pre-trained on MNIST [35], FF-HQ [31], LSUN bedrooms [58], and ImageNet [8] respectively. The MNIST model is with 32×32 resolution and the remaining models are with 256×256 resolution.

Tasks and Metrics. Unlike existing adversarial feature learning methods [15, 12, 44, 13] that are mainly evaluated on the high-level image classification task, we benchmark GH-Feat on a range of both generative and discriminative tasks from multiple levels, including ***(1)*** *Image editing.* It focuses on manipulating the image content or style, *e.g.*, style mixing, global editing, and local editing. ***(2)*** *Image harmonization.* This task harmonizes a discontinuous image to produce a realistic output. ***(3)*** *MNIST digit recognition.* It is a long-standing image classification task. We report the Top-1 accuracy on the test set following [35]. ***(4)*** *Face verification.* It aims at distinguishing whether the given pair of faces come from the same identity. We validates on the LFW dataset [25] following the standard protocol [25]. ***(5)*** *ImageNet classification.* This is a large-scale image classification dataset [8], consisting of over $1M$ training samples across 1,000 classes and $50K$ validation samples. We use Top-1 accuracy as the evaluation metric following existing work [12, 13]. ***(6)*** *Pose estimation.* This task targets at estimating the yaw pose of the input face. $70K$ real faces on FF-HQ [31] are split to $60K$ training samples and $10K$ test samples. The ℓ_1 regression error is used as the evaluation metric. ***(7)*** *Landmark detection.* This task learns a set of semantic points with visual meaning. We use FF-HQ [31] dataset and follow the standard MSE metric [63] to report performances in inter-ocular distance (IOD). ***(8)*** *Layout prediction.* We extract the corner points of the layout line and convert the task to a landmark regression task. The annotations of the collected $90K$ bedroom images ($70K$ for training and $20K$ for validation) are obtained with [62]. Following [69], we report the corner distance as the metric. ***(9)*** *Face luminance regression.* It focuses on regressing the luminance of face images. We use it as a low-level task on the FF-HQ [31] dataset.

4.2. Ablation Study

We make ablation studies on the training of encoder from two perspectives. (1) We choose the layer-wise style codes $\mathbf{y}$ over the $\mathbf{w}$ codes as the representation from StyleGAN. (2) We introduce Spatial Alignment Module (SAM) into the encoder to better handle the spatial information.

Table 1. Quantitative results on ablation study.

Space	SAM	MSE↓	SSIM↑	FID↓
$\mathcal{W}$	✓	0.0601	0.540	22.24
$\mathcal{Y}$		0.0502	0.550	19.06
$\mathcal{Y}$	✓	0.0464	0.558	18.48

Table 2. Quantitative comparison with ALAE [44] on reconstructing images from FF-HQ faces [31] and LSUN bedrooms [58].

	Face			Bedroom		
Method	MSE↓	SSIM↑	FID↓	MSE↓	SSIM↑	FID↓
ALAE [44]	0.182	0.398	24.86	0.275	0.315	21.01
GH-Feat (Ours)	0.046	0.558	18.42	0.068	0.507	16.01

Figure 2. Qualitative comparison on reconstructing real images. From left to right: Inputs, ALAE [44], and our GH-Feat.

Since the encoder is trained with the objective of image reconstruction, we use mean square error (MSE), SSIM [52], and FID [22] to evaluate the encoder performance. Tab. 1 shows the results where we can tell that our encoder benefits from the effective SAM module and that choosing an adequate representation space (*i.e.*, the comparison between the first row and the last row) results in a better reconstruction. More discussion on the differences between $\mathcal{W}$ space and $\mathcal{Y}$ space can be found in Sec. 4.4.1.

4.3. Evaluation on Generative Tasks

Thanks to using the StyleGAN as a learned loss function, a huge advantage of GH-Feat over existing unsupervised feature learning approaches [24, 68, 42, 51, 19], which mainly focus on the image classification task, is its generative capability. In this section, we conduct a number of generative experiments 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 [44] also employs StyleGAN for representation learning. We have following differences from ALAE: (1) We use the $\mathcal{Y}$ space instead of the $\mathcal{W}$

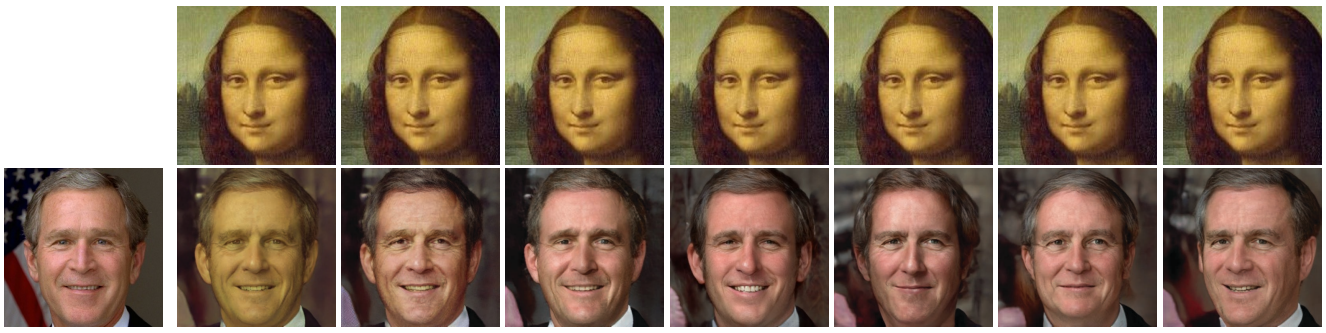


Figure 3. **Style mixing** results by exchanging the GH-Feat extracted from the content image and the style image (first row) at different levels. Higher level corresponds to more abstract feature.

Figure 4. **Global image editing** achieved by GH-Feat. On the left is the input image, while the others are generated by randomly sampling the visual feature at some particular level.

space of StyleGAN as the representation space. (2) We learn *hierarchical* features that highly align with the per-layer style codes in StyleGAN. (3) Our encoder can be *efficiently* trained with a well-learned generator by treating StyleGAN as a loss function. Tab. 2 and Fig. 2 show the quantitative and qualitative comparison between GH-Feat and ALAE [44] on FF-HQ faces [31] and LSUN bedrooms [58]. We can tell that GH-Feat better reconstructs the input by preserving more information, resulting a more expressiveness representation.

4.3.2 Image Editing

In this part, we evaluate GH-Feat on a number of image editing tasks. Different from the features learned from discriminative tasks [20, 19], our GH-Feat naturally supports sampling and enables creating new data.

Style Mixing. To achieve style mixing, we use the encoder to extract 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. 3. We can observe the compelling hierarchical property of the learned GH-Feat. 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, high-level features correspond to the face shape and pose information (last two columns).

Global Editing. The style mixing results have suggested the potential of GH-Feat in multi-level image stylization. Sometime, however, we may not have a target style image to use as the reference. Thanks to the design of the latent space in GANs [17], the generative representation naturally supports sampling, resulting in a strong creativity. In other words, based on GH-Feat, we can arbitrarily sample meaningful visual features and use them for image editing. Fig. 4 presents some high-fidelity editing results at multiple levels. This benefits from the matching between the learned GH-Feat and the internal representation of StyleGAN.

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 certain level. In this way, only a local patch in the output image will be modified while other parts remain untouched. As shown in Fig. 5, we can successfully manipulate the input face with different eyes, noses, and mouths.

4.3.3 Image Harmonization

Our hierarchical encoder is robust such that it can extract reasonable visual features even from discontinuous 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. 6. 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 robust generative visual features.

4.4. Evaluation on Discriminative Tasks

In this part, we verify that even the proposed GH-Feat is learned from generative models, it can be applicable

Figure 5. **Local image editing** achieved by GH-Feat. On the left is the input image, while the others are generated by randomly sampling the visual feature and replacing the spatial feature map (for different regions) at some particular level. Zoom in for details.

Figure 6. **Image harmonization** with GH-Feat. On the top left corner is the original image. Pasting a target image patch onto the original image then feeding it as the input (top row), our hierarchical encoder is able to smooth the image content and produce a photo-realistic image (bottom row).

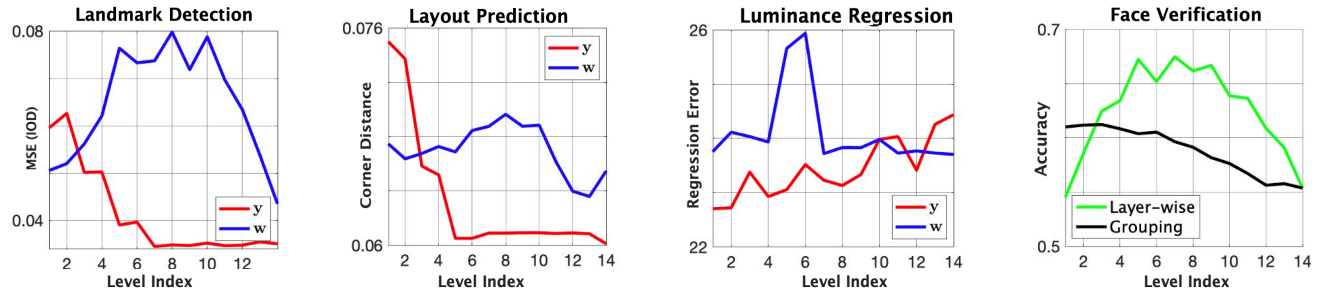


Figure 7. Performance on different discriminative tasks using GH-Feat. Left three columns enclose the comparisons between using different spaces of StyleGAN as the representation space, where $\mathcal{Y}$ space (in **red** color) shows stronger discriminative and hierarchical property than $\mathcal{W}$ space (in **blue** color). This is discussed in Sec. 4.4.1. The last column compares the two different strategies used in the face verification task, which is explained in Sec. 4.4.2. Higher level corresponds to more abstract feature.

to a wide range of discriminative tasks with competitive performances. Here, we do not fine-tune the encoder for any certain task. In particular, we choose multi-level downstream applications, including image classification, face verification, pose estimation, layout prediction, landmark detection, and luminance regression. For each task, we use our encoder to extract visual features from both the training and the test set. A linear regression model (*i.e.*, a fully-connected layer) is learned on the training set with ground-truth and then evaluated on the test set.

4.4.1 Discriminative and Hierarchical Property

Recall that GH-Feat is a multi-scale representation learned by using StyleGAN as a loss function. As a results, it consists of features from multiple levels, each of which correspond to a certain layer in the StyleGAN generator. Here, we would to explore how this feature hierarchy is organized as well as how they can facilitate multi-level discriminative tasks, including face pose estimation, indoor scene layout prediction, and luminance[3] regression from face images. In particular, we evaluate GH-Feat on each task level by level. As a comparison, we also train encoders by treating the $\mathbf{w}$ code, instead of the style code $\mathbf{y}$, as the representation. From Fig. 7, we have three observations: (1) GH-Feat is discriminative. (2) Features at lower level are more suitable for low-level tasks (*e.g.*, luminance regression) and those at higher level better aid high-level tasks (*e.g.*, pose estimation). (3) $\mathcal{Y}$ space demonstrates a more obvious hierarchical property than $\mathcal{W}$ space.

[3]We convert images from RGB space to YUV space and use the mean value from Y space as the luminance.

Figure 8. **Image reconstruction** results on LFW [25]. For each pair of images, left is the low-resolution input while right is reconstructed by GH-Feat. All samples are with the same identity.

4.4.2 Digit Recognition & Face Verification

Image classification is widely used to evaluate the performance of learned representations [24, 68, 19, 42, 13]. In this section, we first compare our proposed GH-Feat with other alternatives on a toy dataset, *i.e.*, MNIST [35]. Then, we use a more challenging task, *i.e.*, face verification, to evaluate the discriminative property of GH-Feat.

MNIST Digit Recognition. We first show a toy example on MNIST following prior work [12, 44]. We make a little modification to ResNet-18 like [37] which is widely used in literatures to handle samples from MNIST [35] in lower resolution. The Top-1 accuracy is reported in Tab. 3 (a). Our GH-feat outperforms ALAE [44] and BiGAN [12] with 1.45% and 1.92%, suggesting a stronger discriminative power. Here, ResNet-18 [20] is employed as the backbone structure for both MoCo [19] and GH-Feat.

LFW Face Verification. We directly use the proposed encoder, which is trained on FF-HQ [31], to extract GH-Feat from face images in LFW [25] and tries three different strategies on exploiting GH-Feat for face verification: (1) using a single level feature; (2) grouping multi-level features (starting from the highest level) together; (3) voting by choosing the largest face similarity across all levels. Fig. 7 (last column) shows the results from the first two strategies. Obviously, GH-Feat from the 5-th to the 9-th levels best preserve the identity information. Tab. 3 (b) compares GH-Feat with other unsupervised feature learning methods, including VAE [33], MoCo [19], and ALAE [44]. All these competitors are also trained on FF-HQ dataset [31] with optimally chosen hyper-parameters. ResNet-50 [20] is employed as the backbone for MoCo and GH-Feat. 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. 8, where our GH-Feat well handles the domain gap (*e.g.*, image resolution) and preserves the identity information.

4.4.3 Large-Scale Image Classification

We further evaluate GH-Feat on the high-level image classification task using ImageNet [8]. Before the training of encoder, we first train a StyleGAN model, with 256×256 resolution, on the ImageNet training collection. After that, we learn the hierarchical encoder by using the pre-trained generator as the supervision. No labels are involved in the above training process.[4] For the image classification problem, we train a linear model on top of the features extracted from the training set with the softmax loss. Then, this linear model is evaluated on the validation set.[5] Tab. 4 shows the comparison between GH-Feat and other unsupervised representation learning approaches [54, 42, 19, 12, 6, 13], where we beat most of the competitors. The state-of-the-art MoCo [19] gives the most compelling performance. But different from the representations learned with contrastive learning, GH-Feat has huge advantages in generative tasks, as already discussed in Sec. 4.3. Among adversarial representation learning approaches, BigBiGAN [13] achieves

Table 3. Quantitative comparison between our proposed GH-Feat and other alternatives on MNIST [35] and LFW [25].

(a) Digit recognition on MNIST.

Methods	Acc.
AE(ℓ_1) [23]	97.43
AE(ℓ_2) [23]	97.37
BiGAN [12]	97.14
ALAE [44]	97.61
MoCo-R18 [19]	95.89
GH-Feat (Ours)	99.06

(b) Face verification on LFW.

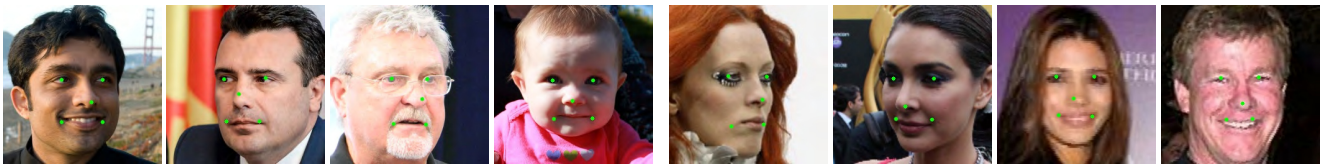
Methods	Acc.
VAE [33]	49.3
MoCo-R50 [19]	48.9
ALAE [44]	55.7
GH-Feat (Grouping)	60.1
GH-Feat (Layer-wise)	67.5
GH-Feat (Voting)	69.7

Table 4. Quantitative comparison on the ImageNet [8] classification task.

Method	Architecture	Top-1 Acc.
Motion Seg (MS) [43, 11]	ResNet-101	27.6
Exemplar (Ex) [14, 11]	ResNet-101	31.5
Relative Po (RP) [9, 11]	ResNet-101	36.2
Colorization (Col) [60, 11]	ResNet-101	39.6
Contrastive Learning		
InstDisc [54]	ResNet-50	42.5
CPC [42]	ResNet-101	48.7
MoCo [19]	ResNet-50	60.6
Generative Modeling		
BiGAN [12]	AlexNet	31.0
SS-GAN [6]	ResNet-19	38.3
BigBiGAN [13]	ResNet-50	55.4
GH-Feat (Ours)	ResNet-50	51.1

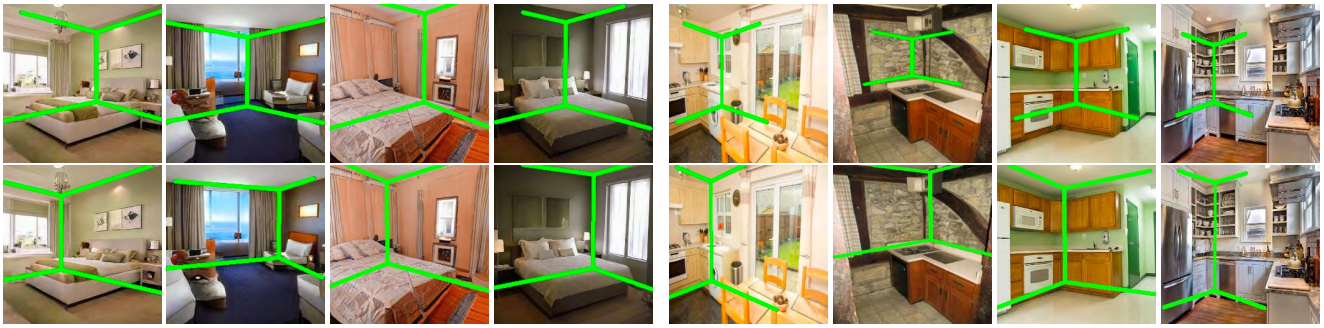
[4] Our encoder can be trained very efficiently, usually $3\times$ faster than the GAN training.

[5] During testing, we adopt the fully convolutional form as in [50] and average the scores at multiple scales.



FF-HQ CelebA

Figure 9. **Landmark detection** results. GH-Feat is trained on FF-HQ [31] dataset but can successfully handle the hard cases (large pose and low image quality) in MAFL dataset [63], a subset of CelebA [39].



Bedroom Kitchen

Figure 10. **Layout prediction** results using feature learned by MoCo [19] (top row) and our GH-Feat (bottom row). Both methods are trained on LSUN bedrooms [58] and then transferred to LSUN kitchens.

the best performance, benefiting from the incredible large-scale training. However, GH-Feat presents a stronger generative ability, suggested by the comparison results on image reconstruction shown in Tab. 5. More discussion can be found in **Supplementary Material**.

4.4.4 Transfer Learning

In this part, we explore how GH-Feat can be transferred from one dataset to another.

Landmark Detection. We train a linear regression model using GH-Feat on FF-HQ [31] and test it on MAFL [63], which is a subset of CelebA [39]. This two datasets have a large domain gap, *e.g.*, faces in MAFL have larger poses yet lower image quality. As shown in Fig. 9, GH-Feat shows a strong transferability across these two datasets. We compare our approach with some supervised and unsupervised alternatives [63, 59, 28, 19]. For a fair comparison, we try the multi-scale representations from MoCo [19] (*i.e.*, Res2, Res3, Res4, and Res5 feature maps) and report the best results. Tab. 6 demonstrates the strong generalization ability of GH-Feat. In particular, it achieves on-par or better performance than the methods that are particular designed for this task [63, 59, 28]. Also, it outperforms MoCo [19] on this mid-level discriminative task.

Layout Prediction. We train the layout predictor on LSUN [58] bedrooms and test it on kitchens to validate how GH-Feat can be transferred from one scene category to another. Feature learned by MoCo [19] on the bedroom dataset is used for comparison. We can tell from Fig. 10 that GH-Feat shows better predictions than MoCo, especially on the target set (*i.e.*, kitchens), suggesting a stronger transferability. Like landmark detection, we also conduct experiments with the 4-level representations from MoCo [19] and select the best.

Table 5. Qualitative comparison between BigBiGAN [13] and GH-Feat on reconstructing images from ImageNet [8].

	MSE↓	SSIM↑	FID↓
BigBiGAN [13]	0.363	0.236	33.42
GH-Feat (Ours)	0.078	0.431	22.70

Table 6. Landmark detection results on MAFL [63].

Method	Supervision	MSE↓
TCDCN [63]	✓	7.95
MTCNN [59]	✓	5.39
Cond. ImGen [28]		4.95
ALAE [44].		10.13
MoCo-R50 [19]		9.07
GH-Feat (Ours)		5.12

5. Conclusion

In this work, we consider the well-trained GAN generator as a learned loss function for learning multi-scale features. The resulting Generative Hierarchical Features are shown to be generalizable to a wide range of vision tasks.

Acknowledgements: This work is supported in part by the Early Career Scheme (ECS) through the Research Grants Council (RGC) of Hong Kong under Grant No.24206219, CUHK FoE RSFS Grant, SenseTime Collaborative Grant, and Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Fund.

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 1

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2019. 2

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Eur. Conf. Comput. Vis.*, 2006. 2

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013. 1

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Int. Conf. Learn. Represent.*, 2019. 2

[6] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 7

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2005. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009. 1, 4, 7, 8

[9] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Int. Conf. Comput. Vis.*, 2015. 7

[10] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Int. Conf. Comput. Vis.*, 2017. 2

[11] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Int. Conf. Comput. Vis.*, 2017. 7

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *Int. Conf. Learn. Represent.*, 2017. 1, 2, 4, 7

[13] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Adv. Neural Inform. Process. Syst.*, 2019. 1, 2, 4, 7, 8

[14] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2014. 7

[15] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. In *Int. Conf. Learn. Represent.*, 2017. 1, 2, 4

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Int. Conf. Learn. Represent.*, 2018. 2

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, 2014. 1, 2, 3, 5

[18] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2, 4, 5, 7, 8

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2, 3, 5, 7

[21] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Adv. Neural Inform. Process. Syst.*, 2017. 4

[23] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006. 7

[24] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Int. Conf. Learn. Represent.*, 2019. 2, 4, 7

[25] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik 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. 4, 7

[26] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, 2017. 2

[27] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2020. 2

[28] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Adv. Neural Inform. Process. Syst.*, 2018. 8

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Eur. Conf. Comput. Vis.*, 2016. 3

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Int. Conf. Learn. Represent.*, 2018. 2

[31] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2, 3, 4, 5, 7, 8

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2015. 4

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent.*, 2014. 7

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2012. 2

[35] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 4, 7

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 3

[37] Kuang Liu. Pyotrch cifar10. https://github.com/kuangliu/pytorch-cifar.git, 2019. 7

[38] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 3

[39] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Int. Conf. Comput. Vis.*, 2015. 8

[40] David G Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 2004. 2

[41] DZ Matthew and R Fergus. Visualizing and understanding convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014. 1

[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 4, 7

[43] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 7

[44] Stanislav Pidhorskyi, Donald Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1, 2, 4, 5, 7, 8

[45] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Int. Conf. Learn. Represent.*, 2016. 2

[46] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2014. 1

[47] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 1, 2

[48] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T Freeman, and Tali Dekel. Semantic pyramid for image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015. 2, 3

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 7

[51] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2, 4

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 2004. 4

[53] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2

[54] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 7

[55] Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *Int. J. Comput. Vis.*, 2020. 1, 2, 3

[56] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 3

[57] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Adv. Neural Inform. Process. Syst.*, 2014. 1, 2

[58] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 4, 5, 8

[59] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016. 8

[60] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Eur. Conf. Comput. Vis.*, 2016. 7

[61] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2

[62] Weidong Zhang, Wei Zhang, and Jason Gu. Edge-semantic learning strategy for layout estimation in indoor environment. *Transactions On Cybernetics*, 2019. 4

[63] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *Eur. Conf. Comput. Vis.*, 2014. 4, 8

[64] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 1, 2

[65] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *Int. Conf. Learn. Represent.*, 2015. 1

[66] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017. 1

[67] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *Eur. Conf. Comput. Vis.*, 2020. 2

[68] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Int. Conf. Comput. Vis.*, 2019. 2, 4, 7

[69] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 4