ID-Unet: Iterative Soft and Hard Deformation for View Synthesis

Mingyu Yin[1]   Li Sun[1,2*]   Qingli Li[1]

[1] Shanghai Key Laboratory of Multidimensional Information Processing,

[2]Key Laboratory of Advanced Theory and Application in Statistics & Data Science,

East China Normal University, 200241 Shanghai, China

m18201815596@163.com   sunli@ee.ecnu.edu.cn   qlli@cs.ecnu.edu.cn

Abstract

View synthesis is usually done by an autoencoder, in which the encoder maps a source view image into a latent content code, and the decoder transforms it into a target view image according to the condition. However, the source contents are often not well kept in this setting, which leads to unnecessary changes during the view translation. Although adding skipped connections, like Unet, alleviates the problem, but it often causes the failure on the view conformity. This paper proposes a new architecture by performing the source-to-target deformation in an iterative way. Instead of simply incorporating the features from multiple layers of the encoder, we design soft and hard deformation modules, which warp the encoder features to the target view at different resolutions, and give results to the decoder to complement the details. Particularly, the current warping flow is not only used to align the feature of the same resolution, but also as an approximation to coarsely deform the high resolution feature. Then the residual flow is estimated and applied in the high resolution, so that the deformation is built up in the coarse-to-fine fashion. To better constrain the model, we synthesize a rough target view image based on the intermediate flows and their warped features. The extensive ablation studies and the final results on two different data sets show the effectiveness of the proposed model. https://github.com/MingyuY/
[Iterative-view-synthesis](https://github.com/MingyuY/Iterative-view-synthesis)

1. Introduction

Novel view synthesis, also known as view translation, facilitates the computer to render the same object under arbitrary poses, given an input object image in a source pose. This is a challenging task, since it requires the model to understand not only the image content, but also the relation
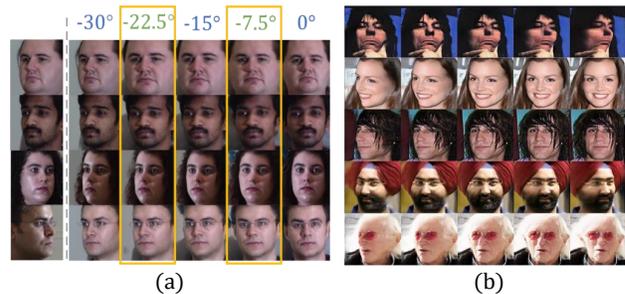

Figure 1: (a) The ID-Unet realizes the translation from the source view to the target, either existing in the MultiPIE dataset $(-30°, -15°, 0°)$, or under a new view (inside the yellow box) by the linear interpolation between two adjacent view conditions. (b) Extra results on CelebA from the existing model training on MultiPIE.

between the object poses and its appearances showing in the image. The model needs to figure out the intrinsic shape of the object and keep it stable during the translation. Meanwhile, it should be able to synthesize the appearance of the object, conforming to the target view condition.

Recently, learning-based method has been employed broadly for this task. Particularly, view synthesis is commonly regarded as a multi-domain image-to-image translation task, which is often modeled by the autoencoder (AE) [6, 42] or variational autoencoder (VAE) [4, 43]. Both consist of a pair of encoder and decoder, in which only the last layer of the encoder connects to the decoder, as shown in Figure 2 (a). However, their limitation has already been realized [20, 41]. Basically, using the latent code from the last layer is not enough to represent the content. Since the decoder can only get one latent code, the source content cannot be kept well in the translated image. A simple but effective solution is the Unet [30] structure. It utilizes several skipped connections by making the shortcuts from the encoder to the decoder, therefore the output can take more features from the source, as shown in Figure 2 (b). Such as V-Unet [9] is a VAE model with skipped connections and

used for person synthesis. Unet indeed improves the image quality. But directly using the low-level encoder features makes it difficult to satisfy the domain requirement, hence the image sometimes fails to be translated into the target domain.

Intuitively, in view translation, the encoder feature needs to be deformed before giving it to the decoder. A straightforward way is to apply the the same optical flow on the different resolutions of the feature map. The flow can be either determined by the priory knowledge [31] or learned by the model [43], and the structure is shown Figure 2 (c). However, we find that using the same flow on different resolutions limits the model's ability for synthesis. On one hand, the flow is often not accurate enough. It is estimated based on the feature of a certain resolution, therefore may be inappropriate for other sizes. On the other hand, the model can already change the view even without any intentional deformations, which implies that we should give it the flexibility to determine the deformation on different resolutions.

To properly exploit the encoder features in the view synthesis, this paper proposes an iterative way to deform them in the coarse-to-fine fashion, so that they can be aligned with the corresponding part in the decoder. The deformed features skip several intermediate layers, and are directly given to the layers in the decoder to complement the content details. Inspired by the idea of progressively estimating the optical flow for the raw pixels [3, 22], our model specifies the offset vectors for the encoder features from the low to the high resolution, and these displacements are accumulated across the multiple resolutions. Specifically, we first use offsets from the low resolution as an approximation to coarsely deform the feature, then the residual offsets are estimated by comparing the roughly deformed result to the decoder feature of the same size. The residuals refine the coarse flow and they are applied to give the additional deformation. The refined flow is further employed by the next block in a larger size. In brief, the encoder feature is first warped according to the coarse flow, and then the remaining offsets is estimated and applied, so that the result is better consistent with the target view.

To compute the initial flow and its following-up residuals, we design the Soft and Hard Conditional Deformation Modules (SCDM and HCDM) based on the features from the encoder and decoder. The view label is the extra conditional input to control the amount of displacement. The idea of the soft flow is to compute the similarity scores (also known as the attention matrix) between the encoder and decoder features like [37, 39]. Given the two of them, the spatial and channel similarities are measured, and then applied onto the encoder features to align them into the target view. However, the soft flow is not efficient enough to compute on multiple resolutions. Furthermore, if the target view is far from the source, the similarity may no longer reflect the
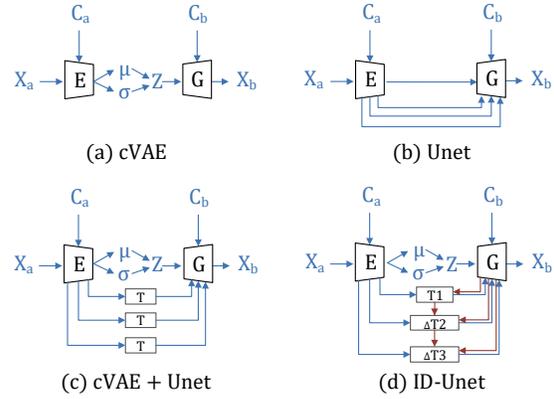

Figure 2: An illustration of several comparing frameworks. (a) and (b) are cVAE and Unet, respectively. (c) is the combination of them, and T realizes the translation from source view a to target view b based on optical flow. (d) improved from cVAE+Unet, the optical flow is estimated iteratively. The initial flow T_1 is calculated according to the low-resolution features. As the resolution increases layer by layer, the residual ΔT_n is calculated to progressively refine the previous result.

spatial deformation. Our solution is to estimate the optical flow to "hard" warp the feature before the spatial and channel attention in SCDM. Moreover, we also design the HCDM which gives the high resolution residuals onto the previous small optical flow, and it "hard" warps the current feature and further aligns it to the target view.

The contributions lie in following aspects: (1) We propose an iterative view translation framework which deforms the encoder feature from different layers and gives them to the decoder to improve the synthesis quality. (2) We design the SCDM and HCDM and use them to align the encoder feature into the target view. (3) Extensive experiments on two different datasets show the effectiveness of the proposed framework and our designed modules.

2. Related Works

GAN and its structure design. GAN [5, 10, 17, 24, 25] has shown its ability in synthesizing high dimensional structured data. The rationale behind GANs is to learn the mapping from a latent distribution $z \sim N(0, I)$ to mimic the real data through adversarial training. Because of the instability of the adversarial training, it often needs to give extra constraints on discriminator D [12,13]. Moreover, by incorporating an encoder E, GAN can be applied in a variety of I2I translation, either supervised by the groundtruth [16,38] or not [6,47]. In AE, the source image is first converted into a latent code by E, and then G takes the code and transforms it back into the image. Since there are multiple visual domains, the source and target domain labels are given to the
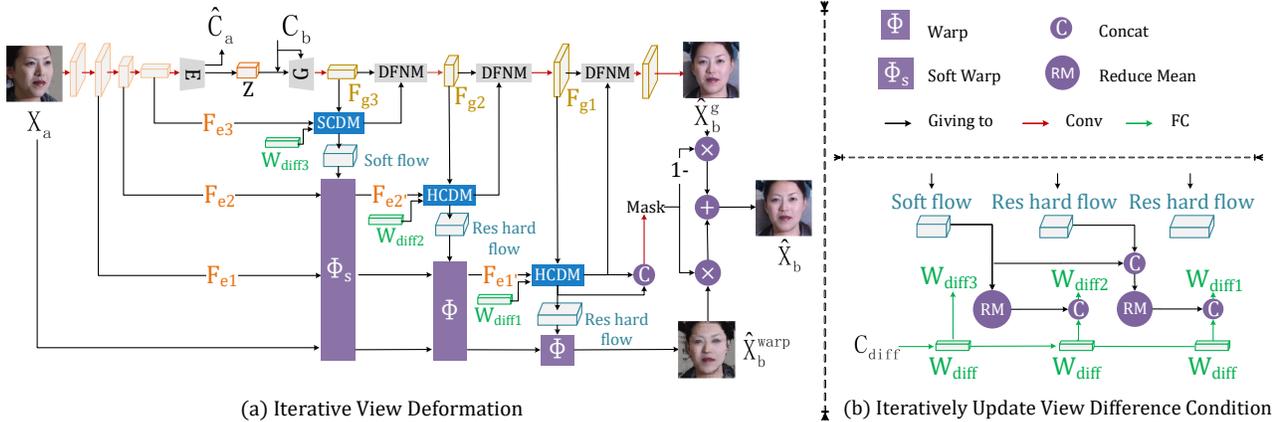
(a) Iterative View Deformation   (b) Iteratively Update View Difference Condition

Figure 3: (a) The detailed architecture of the proposed ID-Unet. X_a is mapped to the variable Z through encoder, and it is supplied to the encoder together with the target view label C_b. SCDM and HCDM warp the encoder features to the target view, and give their output to the decoder in a way of de-normalization (DFNM) [43] to complement the details. And the low resolution optical flow is as an approximation to change the high resolution feature by Φ_s and Φ, so that the flow is formed in the coarse-to-fine fashion. (b) Iteratively update of the view conditional C_{diff} to W_{diff3}, W_{diff2} and W_{diff1}.

AE as the guide. Variational autoencoder (VAE) [19] has the similar structure with AE, in which the latent code is assumed to follow the posterior distribution, and the posterior is to be close to a prior during training. Hence, VAE is not a deterministic model like AE. It can support sampling from the posterior or prior, with their corresponding synthesis looking like real images. VAE is extended to its conditional version cVAE [4, 33] as shown in Figure 2 (a), and cVAE is suitable for either synthesizing the diverse styles of images [48], or disentangling the latent code [14, 45].

In AE or VAE, E and G are only connected through the last latent code, which is not enough to guarantee the synthesis quality. AdaIN [15], SPADE [27], CIN [8] and CBIN [23] are other ways to inject the feature into the multiple decoder layers through a side branch, which adjusts the statistics of features in the main branch. The Unet [30] and its variants link E and G by setting up shortcuts between them. But it often leads to failures in I2I translation. Xiao *et al.* [41] use G's output as the residual added onto the source image to improve the quality. Li *et al.* [20] designs PONO layer in Unet, normalizing and adapting source domain features from E to G. However, these structures are not designed for view synthesis.

View synthesis. Traditional approaches [2, 18, 29] for this task are mainly based on projection geometry, which tries to recover the 3D relation between the object and its projected image plane. They directly estimate either the depth and camera pose [2], or 3D model parameters [18, 29], so that the object can be projected into the target view. Learning-based methods [7, 46] become increasingly popular nowadays. In [7], a CNN model learns to process the latent code for object shape and camera pose, and map it into an image.

In [46], the CNN predicts the optical flow to warp the source view into the target. Recently, due to the great success of GAN [26, 31, 34, 36, 42], the AE structure plus the adversarial training begins to play the key role in view synthesis. Meanwhile, VAE and its probabilistic latent vector [35, 43] can be applied in this task as well, which even better keeps the contents from the source. However, none of these works consider the coarse-to-fine iterative deformation on features to perform view synthesis.

3. Method

We intend to synthesize object in arbitrary views. Given an image X_a containing an object in the source view C_a, and an expected target view C_b as the inputs, the model outputs $\hat{X}_b$, a synthesis of the same object in the target view. The difficulty of this task lies in accurately changing the object from the original to the target view, while keeping other attributes (*e.g.* identity) unchanged during the translation.

3.1. The Framework of Iterative View Translation

A brief framework is given in Figure 2 (d). The idea is to apply multiple deformations on the shallow layer features in the encoder and give them to the decoder, which is conducive to maintain the source content irrelevant to the view. Note that in Figure 2 (c), module T also estimates the optical flow and is applied on different resolutions, but it is in the independent way. Here the key improvement is the coarse-to-fine manner to estimate the initial deformation T_1 and refine it through ΔT_i iteratively, where $i = 2, 3$ in our setting. Moreover, we find that using the deformed low-level features in the decoder causes the missing of content details in the translated image. While cVAE has a better ability
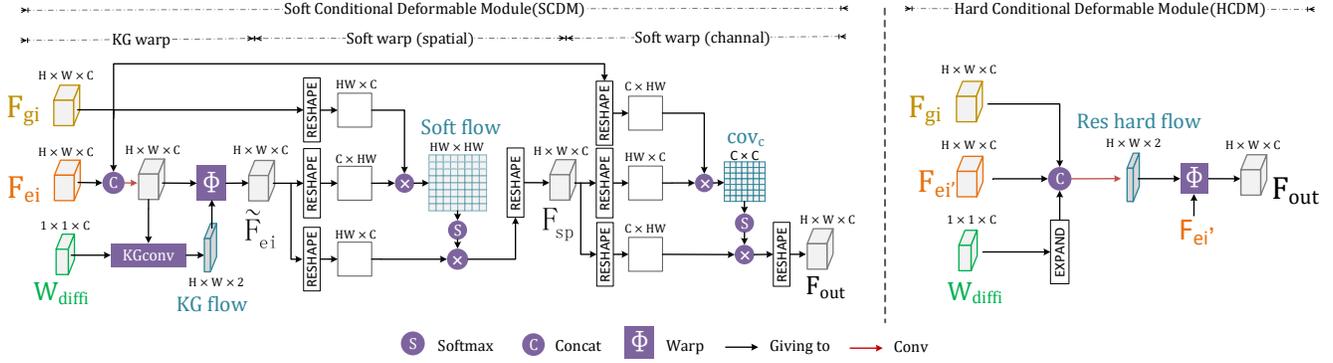
Figure 4: Illustration of Soft and Hard Conditional Deformation Module. We show the SCDM and HCDM on the left and right respectively. Both have 3 inputs, F_{gi}, F_{ei} and W_{diff} from G, E and view condition, and 1 output F_{out} given to main branch of G. SCDM consists of 3 stages, which are KG warp, spatial soft warp and channel soft warp. HCDM directly estimates the optical flow and warps the encoder feature F_{ei}.

to keep complete objects by introducing the prior distribution as a regularization. The proposed Figure 2 (d) inherits the cVAE+Unet structure in Figure 2 (c). In particular, the source view X_a is input to the encoder to provide the content, and is mapped to a posterior distribution, from which the latent Z can be sampled. The decoder takes Z and the target view condition C_b to synthesize the translation.

The proposed ID-Unet, with its specific architecture shown in Figure 3, accomplishes the iterative view translation on the features F_e in different layers of the encoder, so that they are aligned with their corresponding part F_g in the decoder. Notice that F_e have spatial dimensions and are under the original view C_a. If the same features under the target view C_b could also be obtained, it would be easy to estimate the optical flow to deform F_e. Intuitively, the decoder feature F_g can be roughly assumed in the target view C_b, since the first decoder feature F_{g3} is computed according to the latent Z and condition C_b, which has already been aligned into C_b. This can be extended to other decoder features such as F_{g2} and F_{g1}. They are closer to C_b than their counterparts F_{e2} and F_{e1}, so we employ the pair F_{e3} and F_{g3} to estimate the initial T_1, and the following pairs to predict ΔT_i.

3.2. Soft and Hard Deformation

We design two types of modules, applying the soft and hard deformations on low and high resolution feature, respectively. Both of them depend on W_{diff}, a 1×1 vector given by MLP, which reflects the view difference. We will elaborate it in the next section.

Soft Conditional Deformation Module (SCDM)

SCDM estimates the initial deformation T_1 based on a pair of features F_{e3} and F_{g3} at the lowest resolution, as

shown in the left of Figure 4. Instead of directly comparing F_{e3} and F_{g3}, a two-channel flow is first predicted through kernel given conv (KGconv) and applied onto F_{e3} by the warping operation Φ. Here, the purpose is to align F_{e3} in the target view direction to form $\tilde{F}_{e3}$, so that the soft flow can be calculated from two similar features $\tilde{F}_{e3}$ and F_{g3}, preventing from inappropriate matching two views far from each other. Note that KGconv uses W_{diff} as conv kernels to generate x and y offsets in the optical flow to assist view translations [43].

Then, to measure the similarity between source $\tilde{F}_{ei}$ and target F_{gi}, we compute the Soft flow $\in \mathbb{R}^{HW \times HW}$ by the inner product between $\hat{e}_v$ and $\hat{g}_u$: Soft flow$(u, v) = \hat{g}_u^T \hat{e}_v$, where $\hat{e}_v$ and $\hat{g}_u \in \mathbb{R}^C$ represent the channel-wise centralized feature of $\tilde{F}_{ei}$ and F_{gi} at position v and u, $\hat{e}_v = e_v - \mu(e_v)$ and $\hat{g}_u = g_u - \mu(g_u)$. Soft flow$(u) \in \mathbb{R}^{HW}$ represents the similarity between F_{g3} at position u and $\tilde{F}_{e3}$ at all position, so the weighted $\tilde{F}_{ei}$ is the output feature element $F_{sp}(u)$. The weight, Soft flow(u), is normalized by the Softmax function and multiplied on each position of $\tilde{F}_{ei}$.

$$F_{sp}(u) = \text{softmax}(\frac{1}{\tau} \cdot \text{Soft flow}(u)) \cdot \tilde{F}_{ei}. \quad (1)$$

Different from the classical flow warp (hard warp), F_{sp} in (1) is the weighted sum of the feature at multiple positions in $\tilde{F}_{ei}$. However, smooth weights may change image contents like colors or styles. In order to maintain them, we balance the soft and hard warp by incorporating a temperature $\tau < 1$ in (1), which increases the impact of the high-weight position (which is more relevant) on the output.

Finally, based on F_{sp} and F_{g3}, we obtain the similarity matrix Cov_c along the channel in the same way of spatial dimension, and "Soft warp" is also performed on F_{sp} to maintain more valid information in the channel dimension.

Hard Conditional Deformation Module (HCDM)

Basically, HCDM utilizes the results of SCDM, and refines the deformation for larger size F_{e2} and F_{e3}. Once the soft flow is obtained, the globe deformation Φ_s can be approximated. For the high-resolution features, as shown in Figure 3 (a), Φ_s also takes effect in HCDM. It first makes the coarse deformation on F_{e2} and F_{e1}. Due to the size mismatch between Soft Flow and feature F_{e2} or F_{e1}, one element in Soft Flow matrix is scaled and applied to the corresponding square area in the feature of larger size, simplifying as $F_{e2'} = \Phi_s(F_{e2})$. Then the residual optical flow at high resolution is further estimated by the deformed results $F_{e2'}$, the target view features F_{g2} and W_{diff2} together. They are concatenated to learn the residual flow. The residual (Res hard flow) can be superimposed, giving $F_{e1'} = \Phi(\Phi_s(F_{e1}))$, in which Φ denotes the hard warping operation by the optical flow. Therefore, with the increase on resolution, the optical flow for translation is gradually refined by HCDM.

3.3. Iteratively Update View Difference Condition

With the gradual refinement of optical flow, the features $F_{e2'}$ and $F_{e1'}$ have been converted to the target view to a certain extent. Then the actual view of the current features ($F_{e2'}$ or $F_{e1'}$) is no longer the same as the source, and the condition W_{diff} should also be adapted, since it no longer translates from the source to the target, but from the current view to the target. In our model, W_{diff} is updated iteratively together with the feature. Specifically, we use the current flow to measure the amount of the translation, and learn how to update W_{diff} by the model itself. In Figure 3 (b), the view label difference C_{diff} is passed through an MLP, to get W_{diff}. W_{diff3} used for the first warp is directly obtained from W_{diff} through one fc layer. During the further operation, the mean of optical flow $(\mu(\mathrm{d}x), \mu(\mathrm{d}y))$ is concatenated with W_{diff} to determine the next conditional vector (W_{diff2} or W_{diff1}) for the further deformation.

3.4. Training Details and Loss Functions

Adversarial and Reconstruction Loss

We use adversarial loss $L_{\mathrm{E,G}}^{adv}$ and L_{D}^{adv} [21] to ensure the translated image approximates the true distribution like in (2). As shown in Figure 3 (a), the final $\hat{X}_b$ is mixed by two parts. One is the $\hat{X}_b^{warp}$, obtained by the soft and hard deformation on the source X_a, and the other $\hat{X}_b^g$ is the output of the generator. The model learns a single channel mask to weight and combine the two results. The mask is computed based on the output and the optical flow in the last HCDM.

$$
\begin{aligned}
L_{\mathrm{D}}^{adv} =& \mathbb{E}_X[\max(0, 1 - \mathrm{D}(X, C_b))] \\
&+ \mathbb{E}_{\hat{X}_b}[\max(0, 1 + \mathrm{D}(\hat{X}_b, C_b))], \\
L_{\mathrm{E,G}}^{adv} =& \mathbb{E}_{\hat{X}_b}[\max(0, 1 - \mathrm{D}(\hat{X}_b, C_b))]
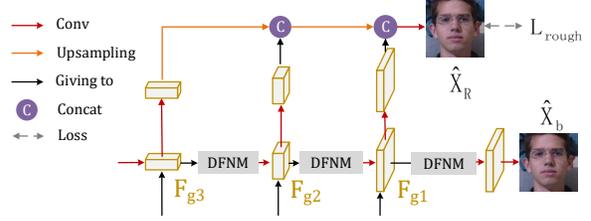\end{aligned}
\tag{2}
$$

Figure 5: Besides the normal translated image $\hat{X}_b$, image $\hat{X}_R$ is generated from F_{g3}, F_{g2} and F_{g1} for the rough loss.

Like ACGAN [25], we use classification losses L_{C}^{cls} and $L_{\mathrm{E,G}}^{cls}$ in (3). The classifier C shares a part of its weights with discriminator D.

$$
\begin{aligned}
L_{\mathrm{C}}^{cls} &= -\mathbb{E}_{X_b} \sum_c \mathbb{I}(c = C_b) \log \mathrm{C}(c|X_b), \\
L_{\mathrm{E,G}}^{cls} &= -\mathbb{E}_{\hat{X}_b} \sum_c \mathbb{I}(c = C_b) \log \mathrm{C}(c|\hat{X}_b)
\end{aligned}
\tag{3}
$$

In addition, by combining the reconstruction loss in image domain $L_{\mathrm{E,G}}^{pixel} = ||X - \hat{X}_j||_1$ and feature domain $L_{\mathrm{E,G}}^{content} = \sum_i ||\phi^i(X) - \phi^i(\hat{X}_j)||_1$, the image quality is guaranteed more faithfully. Here ϕ indicates i-th layer of a pre-trained VGG [32] network, and $j = b, a, aa$. $\hat{X}_a$ and $\hat{X}_b$ are the fake images at target view A and B. $\hat{X}_{aa}$ the cyclic translation result, which is translated back from the synthesised image in view B.

Disentangling Loss

The source image X_a is mapped to a code $Z \sim \mathrm{E}(Z|X_a)$ where $\mathrm{E}(Z|X_a)$ is a posterior depending on the source X_a. Z is fed directly into G, so it should keep the content of the object, and be irrelevant to views [42, 43]. To prevent Z from taking view relevant factors, we add two auxiliary classifier losses for E. One computes the classification loss L_E^{clsC} which tries to predict $\hat{C}_a = \mathrm{E}(c|X)$ to approximate view label C_a, as is defined in the first term in (4). Another adversarial constraint L_{E}^{cls} in (4) makes the view classification based on Z by the hidden layer classifier DAC, which is the last two terms in (4).

$$
\begin{aligned}
L_{\mathrm{E}}^{clsC} =& -\mathbb{E}_{X \sim X_a} \sum_c \mathbb{I}(c = C_a) \log \mathrm{E}(c|X), \\
L_{\mathrm{E}}^{clsZ} =& -\mathbb{E}_{Z \sim \mathrm{E}(Z|X_a)} \sum_c \frac{1}{C} \log \mathrm{DAC}(c|Z), \\
L_{\mathrm{DAC}}^{clsZ} =& -\mathbb{E}_{Z \sim \mathrm{E}(Z|X_a)} \sum_c \mathbb{I}(c = C_a) \log \mathrm{DAC}(c|Z)
\end{aligned}
\tag{4}
$$

Here L_{DAC}^{clsZ} is the penalty to train DAC, ensuring the accuracy of the view classification. L_{E}^{clsZ} is the adversarial loss applied on E to make DAC confused to predict the uniform value on each view. Furthermore, via the constraint of

Method	MultiPIE					3D chair			
	$L_1\downarrow$	SSIM↑	LPIPS↓	FID↓	id-acc↑	$L_1\downarrow$	SSIM↑	LPIPS↓	FID↓
MV [34]	15.21	0.489	0.217	29.85	0.742	13.86	0.779	0.224	104.49
Unet [30]	14.03	0.619	0.164	49.86	0.396	21.75	0.697	0.255	86.74
cVAE [4]	12.82	0.635	0.119	28.99	0.651	8.93	0.828	0.102	27.79
CRGAN [35]	14.12	0.627	0.141	26.77	0.868	13.33	0.788	0.196	28.23
VIGAN [42]	12.96	0.638	0.117	29.05	0.686	12.13	0.781	0.133	33.18
PONO [20]	13.63	0.621	0.126	**23.77**	0.862	12.74	0.780	0.148	37.85
CDVAE [43]	13.49	0.623	0.125	23.95	0.917	13.38	0.773	0.148	40.81
cVAE+Unet	12.37	0.658	0.113	28.98	0.689	11.32	0.790	0.123	32.04
A:cVAE+Unet+Iterative	12.14	0.676	0.100	27.50	0.893	10.64	0.801	0.120	**27.76**
B:A+ImageMix	12.01	0.679	0.101	26.55	**0.928**	9.30	0.819	0.104	36.18
C:B+IterativeC	11.11	0.684	0.095	24.55	0.913	9.055	0.826	0.102	29.06
D:C+rough loss	**10.72**	**0.694**	**0.093**	25.12	0.911	**7.57**	**0.847**	**0.089**	28.87

Table 1: Comparison on the MultiPIE and the 3D chair datasets.

KL loss $L_{KL} = D_{\mathrm{KL}}[\mathrm{E}(Z|X_a)||N(0,I)]$, the latent code Z from the encoder is close to the standard normal distribution and has no category-related information.

Rough Loss

We design the rough loss on the deformed features in SCDM and HCDM, to make the features conform to the target view. As is described in section 3.1 and Figure 3, the decoder features F_{g3}, F_{g2} and F_{g1} are assumed under target view C_b. To better ensure that they are in target view, F_{g3}, F_{g2} and F_{g1} are combined and fed to a layer ψ to generate an image $\hat{X}_R = \psi(F_{g3}, F_{g2}, F_{g1})$ as shown in Figure 5. The image $\hat{X}_R$ is constrained by pixel-wise L1 loss and classification loss of the classifier C, like in (5).

$$L_{\mathrm{E,G}}^{rough} = ||X_b - \hat{X}_R||_1 + \sum_c \mathbb{I}(c = C_b)\log \mathrm{C}(\hat{X}_R) \quad (5)$$

Overall Objective. The total optimization loss is a weighted sum of the above. Generators E, G, discriminator D, classifier C, and the latent classifier DAC are trained by minimizing (6).

$$
\begin{aligned}
L_{\mathrm{E,G}} =& L_{\mathrm{E,G}}^{adv} + L_{\mathrm{E,G}}^{cls} + \alpha_1 L_{\mathrm{E,G}}^{content} + \alpha_2 L_{\mathrm{E,G}}^{pixel} \\
&+ \alpha_3 L_{KL} + L_{\mathrm{E}}^{clsC} + L_{\mathrm{E}}^{clsZ} + \alpha_4 L_{\mathrm{E,G}}^{rough}, \quad (6) \\
L_{\mathrm{D}} =& L_{\mathrm{D}}^{adv}, \quad L_{\mathrm{C}} = L_{\mathrm{C}}^{cls}, \quad L_{\mathrm{DAC}} = L_{\mathrm{DAC}}^{clsZ}
\end{aligned}
$$

The loss weights $\alpha_1, \alpha_2, \alpha_3, \alpha_4 = 5, 5, 0.1, 10$.

4. Experiments

4.1. Datasets and Quantitative Metrics.

Datasets. We validate the proposed ID-Unet on face dataset MultiPIE [11] and 3D chair [1] object dataset. MultiPIE contains about 130,000 images, with 13 viewing angles, spanning $180°$. Nine of central viewing angles are used for training and testing. The 3D chair contains 86,304 images, covering a total of 62 angles. For all the datasets, 80% are used for training and the rest 20% for testing.

Quantitative Metrics. To give the evaluation on different methods, we use following metrics during the test. We calculate **L1 error** and **LPIPS** [44] to measure the difference at pixel level and feature level between the generated and ground truth image. **SSIM** [40] is calculated to compare the similarity of image structure. **FID** represents the distance between the generated image distribution and the real image distribution, so as to measure the authenticity of the generated image. At the same time, on the MultiPIE dataset [11], we use the face identity recognition network pretrained on VGGface [28] dataset to calculate the **identity accuracy** of generated image. Table 1 lists all the metrics for the ablation and comparison models. More specific training details are given in the supplementary materials.

4.2. Ablation Study

In this section, we compare the results in several different ablation settings to verify the effectiveness of every component in the proposed method.

A: cVAE+Unet+Iterative. Setting A is based on the two common models Unet and cVAE, combining them and then sending the encoder features to the corresponding decoder layer after iterative view translation. In Figure 6 and 8, the 2nd, 3rd and 4th rows are generated images from Unet, cVAE and model A, respectively. We observe that the object from Unet appears incomplete (disappeared chair part or eyes). For cVAE, the face identity and the chair color have changed to a large extent. While the setting A can ensure the integrity of the image and the invariance of the information irrelevant to the view. Meanwhile, as shown in Table 1, compared with Unet and cVAE, all results under setting A are significantly improved, especially the id-acc
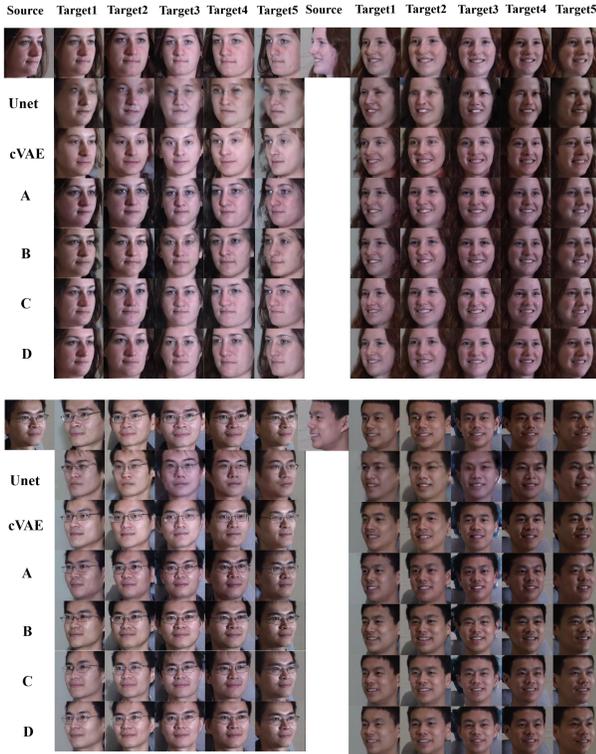
Figure 6: Ablation study on MultiPIE dataset. The source and the ground truth targets are provided in the first row. Please zoom in for details.

increases from 0.396 (Unet), 0.651 (cVAE) to 0.893.

B: A+ImageMix. Based on A, setting B combines the output of the generator $\hat{X}_b^g$ with the deformation of the original image $\hat{X}_b^{warp}$, which is conducive to maintain more valid content of the original image and generating more realistic images, as shown in the 5th row in Figure 6, with the id-acc reaching 0.928.

C: B+IterativeC. The experimental setting C further extends on B. In Figure 6 and 8, the view translation is more accurate and better handled in detail. Because the view difference condition W_{diffi}, where $i = 1, 2, 3$, is updated iteratively according to the degree of deformation of current features, the view condition is better adjusted and controlled. The result in Table 1 also verifies the conclusion.

D: C+rough loss. In setting D, the effectiveness of rough loss is validated. From the last row in Figure 8, it can be seen that the chairs are not only close to the targets on pixel, but also have stable shape at different views. It is obvious that this model can better understand the intrinsic shape of the chairs. This is also supported by Table 1.

4.3. Visualizations

Optical flow. In Figure 7, the source image is translated into 3 target views. The 3rd row is the result from soft flow,
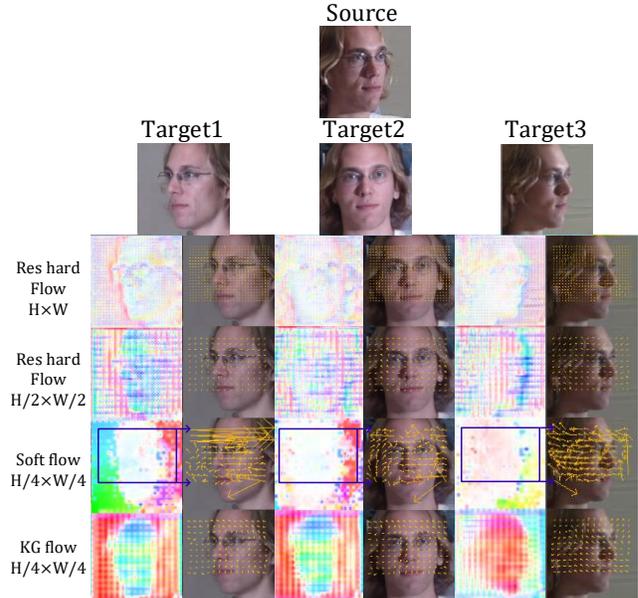

Figure 7: Visualization of optical flow on different layers. We list 4 deformation flows from the bottom to the top. The direction of the flow points from the target to source.
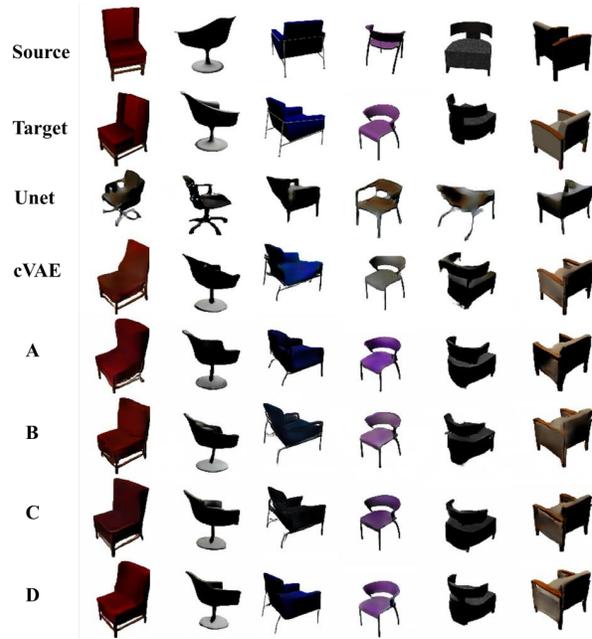

Figure 8: Ablation study on 3D chair dataset. The source and ground truth targets are given in the 1st and 2nd rows.

which is converted into a 2-channel hard flow for visualization by taking out the most relevant coordinate. We find that the absolute value of the soft flow is larger than the rest of the hard flow in the 1st, 2nd and 4th rows, which can be used to achieve overall deformation. The 4th row of KG flow in

Figure 9: Comparison on MultiPIE. For each image, the top row is the ground truth while the 2nd to 6th rows are generated by MV [34], CRGAN [35], VIGAN [42], PONO [20] and CDVAE [43] respectively. The last row is generated by our ID-Unet.
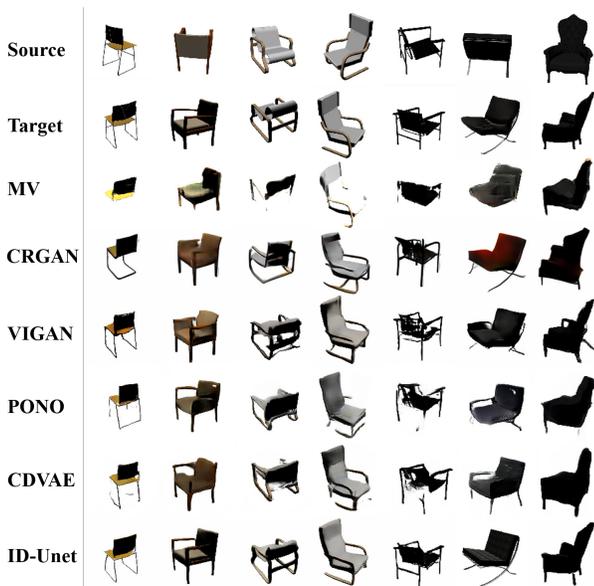

Figure 10: Comparison on 3D chair. The 1st and 2nd rows are the source and target images while the 3rd to 7th rows are generated by MV [34], CRGAN [35], VIGAN [42], PONO [20] and CDVAE [43] respectively. The last row is generated by ID-Unet.

SCDM has the clear direction. The magnitudes of residual hard flow in the 1st and 2nd row are smaller, showing that the feature progressively approaches the target view.

Continuous view synthesis by interpolation. To translate an image in an unseen view, we linearly interpolate the two conditions to get an arbitrary angle image that does not exist in the dataset. Figure 1 shows that our model is smooth enough to achieve view morphing.

Visual comparisons with previous works. As shown in Figure 9 and 10, ID-Unet can accurately achieve the view synthesis while effectively maintain the source contents, *e.g.*, the face ID and the chair style. The quantitative results in Table 1 can also confirm the effectiveness. The results from MV [34] are excessively bright, and it has problems such as ghosting for difficult samples. VIGAN [42] and CRGAN [35] have good results on simple samples, but they can not maintain the original structure for complex chairs, and synthesize the facial details like eyes in the 2nd facial image. PONO [20] and CDVAE [43] have good ability to keep the source content, but their models do not understand the structure of complex objects. For example, the 3rd face in Figure 9 cannot achieve reasonable translation.

5. Conclusion

This paper presents the ID-Unet to perform the view synthesis. It iteratively makes the deformation on the encoder features from different layers, and connects them into the decoder to complement the content details. To achieve the view translation, we design the SCDM and HCDM to align the feature from the source view to the target. Both the modules take the encoder and decoder features as well as the view condition vector as the inputs, compare the features to give either the soft or hard flow, and warp the encoder feature according to it. Since the flows are computed from features of different sizes, we accumulate them across resolutions and use the current flow to coarsely align the encoder feature first, and then estimate the residuals flow to refine it. Experiments show the effectiveness of the proposed model on two different datasets.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014. 6

[2] Shai Avidan and Amnon Shashua. Novel view synthesis in tensor space. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1034–1040. IEEE, 1997. 3

[3] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 2

[4] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017. 1, 3, 6

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 1, 2

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1538–1546, 2015. 3

[8] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 3

[9] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018. 1

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[11] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 6

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 2

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 2

[14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 3

[15] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 3

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[18] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3d object manipulation in a single photograph using stock 3d models. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 3

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[20] Boyi Li, Felix Wu, Kilian Q Weinberger, and Serge Belongie. Positional normalization. In *Advances in Neural Information Processing Systems*, pages 1622–1634, 2019. 1, 3, 6, 8

[21] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 5

[22] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 2

[23] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems*, pages 2558–2567, 2018. 3

[24] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7588–7597, 2019. 2

[25] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651, 2017. 2, 5

[26] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3500–3509, 2017. 3

[27] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 3

[28] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015. 6

[29] Konstantinos Rematas, Chuong H Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. Novel views of objects from a single image. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1576–1590, 2016. 3

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 3, 6

[31] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018. 2, 3

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[33] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015. 3

[34] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 155–171, 2018. 3, 6, 8

[35] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. Cr-gan: learning complete representations for multi-view generation. *arXiv preprint arXiv:1806.11191*, 2018. 3, 6, 8

[36] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1415–1424, 2017. 3

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[41] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 168–184, 2018. 1, 3

[42] Xiaogang Xu, Ying-Cong Chen, and Jiaya Jia. View independent generative adversarial network for novel view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7791–7800, 2019. 1, 3, 5, 6, 8

[43] Mingyu Yin, Li Sun, and Qingli Li. Novel view synthesis on unpaired data by conditional deformable variational autoencoder. *arXiv preprint arXiv:2007.10618*, 2020. 1, 2, 3, 4, 5, 6, 8

[44] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[45] Zhilin Zheng and Li Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12192–12201, 2019. 3

[46] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 3

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

[48] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017. 3