Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Neural 3D Face Rendering conditioned on 2D appearance via GAN Disentanglement Method

Ruizhao Chen[a,b], Ran Yi[a,*], Tuanfeng Yang Wang[c], Lizhuang Ma[a,*]

[a]*Shanghai Jiao Tong University, Shanghai, 200240, China*
[b]*Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai, 201112, China*
[c]*Adobe Research, North America*
[d]*Shanghai Jiao Tong University, Shanghai, 200240, China*

ARTICLE INFO

Article history:
Received July 14, 2023

Keywords: Neural Rendering, GAN Disentanglement, Conditional Generation

ABSTRACT

Previewing the shaded output of 3D models has been a long-standing requirement in the field. Typically, this is achieved by applying common materials; however, this approach is often labor-intensive and can yield only rough results in the trial stage. Conventional 2D style transfer methods are unsuitable for 3D-to-2D cross-domain conversion, and they cannot accurately reflect the mesh's geometry. Inspired by StyleGAN2's related research, we propose a method for rendering 2D images of 3D face meshes directly controlled by a single 2D reference image, using GAN disentanglement. Our approach involves an input of a 3D mesh and a reference image, where encoders extract geometric features from the mesh and appearance features from the reference image. These features control the StyleGAN2 generator to obtain a generated image that preserves the 3D mesh's geometry and the reference image's appearance. Our experiments demonstrate that this method performs well in generating images while maintaining geometric consistency.

1. Introduction

The assignment of materials and production of textures for 3D models are crucial in achieving high-quality and realistic rendering results with detail. If the user can automatically color the 3D mesh or use a picture to control the process and preview the image, it could significantly enhance the production of 3D and related digital content. However, this task requires 3D-to-2D cross-domain supervision, and conventional 2D supervision methods, such as LPIPS loss or L_1 loss on images, are not appropriate for supervising geometric consistency between 3D meshes and 2D images.

Since there are many related works in the domain of human facial recognition, we aim to achieve our objectives for this task before expanding to other domains. Our objectives involve producing a facial image that matches the geometry of the human head 3D mesh, specified camera parameters during capture, and a given 2D reference human face image, that displays the appearance features of the reference image. Figure 1 shows the definition of our task.

It is essential to disentangle the geometric and appearance information to solve this problem. There exist various definitions for this problem. Among them, Tewari *et al.* 's decoupling of shape and texture definition [1] is more aligned with our objectives. Changes in the geometric domain should not affect the texture domain. Nonetheless, previous studies have not yet achieved complete disentanglement, as demonstrated in Figure 2 [2, 1].

Achieving high-quality disentanglement poses several challenges. Firstly, we need to establish a geometric relationship

*Corresponding authors.
e-mail: ranyi@cs.sjtu.edu.cn (Ran Yi), ma-lz@cs.sjtu.edu.cn (Lizhuang Ma)

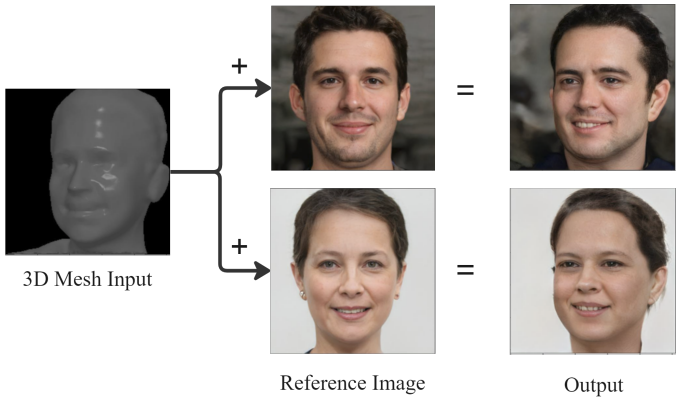


Fig. 1: Demonstration of rendering a 3D mesh under the guidance of the reference image using our method. Images and meshes are randomly sampled in our synthesized dataset.

between the 3D mesh input and the resulting 2D image to ensure proper geometric supervision. However, conventional 2D supervision methods are inadequate and cannot be applied directly [3]. To address this issue, we designed an encoder structure with PointNet++ [4] to extract geometric features from the 3D mesh. These features were then projected into image space to establish mapping correspondences. Additionally, to enable full network supervision during training, we obtained a large dataset of one-to-one image and mesh pairs from StyleSDF [5]. Due to the consistency of geometric information in the image and mesh pairs, we can impose effective geometric feature constraints in 2D image space by establishing geometric constraints between the corresponding 2D image and output.

Second, we must exclude geometric information from the input image as color guidance. When using an image as a reference input, it must contain specific geometric details that should not leak into the resulting features. To prevent the encoding of geometric information by the appearance encoder, we restrict it to produce consistent encoding results for appearance reference images with distorted geometric information.

In conclusion, we put forth a new end-to-end network architecture and training method that plausibly disentangles geometry and appearance information. In the experimental section, we demonstrate in-depth the superiority of our method for preserving geometric information, both quantitatively and qualitatively.

2. Related Work

2.1. Example-based Colorization

2D-to-2D Colorization.. Recoloring media data using an existing image as reference input is a promising research area. Currently, several successful approaches based on neural networks exist for 2D-to-2D colorization, there are currently several successful approaches[6, 7, 8] based on neural networks, whose attempts are highly dependent on the similarity between the reference image and the original image. Luo *et al.* proposed Time Travel Rephotography [9] based on GAN latent space mapping to overcome this limitation. Their work is based on finding a vector with consistent geometric information of the input grayscale image in the W_+ space of the pretrained StyleGAN generator to obtain high-quality recolorization results for a single image.

Researches on the colorization of other media like depth maps [10]and dense correspondence map[2] provide a method to make the network understand spatial structure information. However, the current research is on the compressed spatial structure information in 2D image space. There is still room for further improvement in establishing a reliable mapping from 3D to 2D and providing highly controllable input.

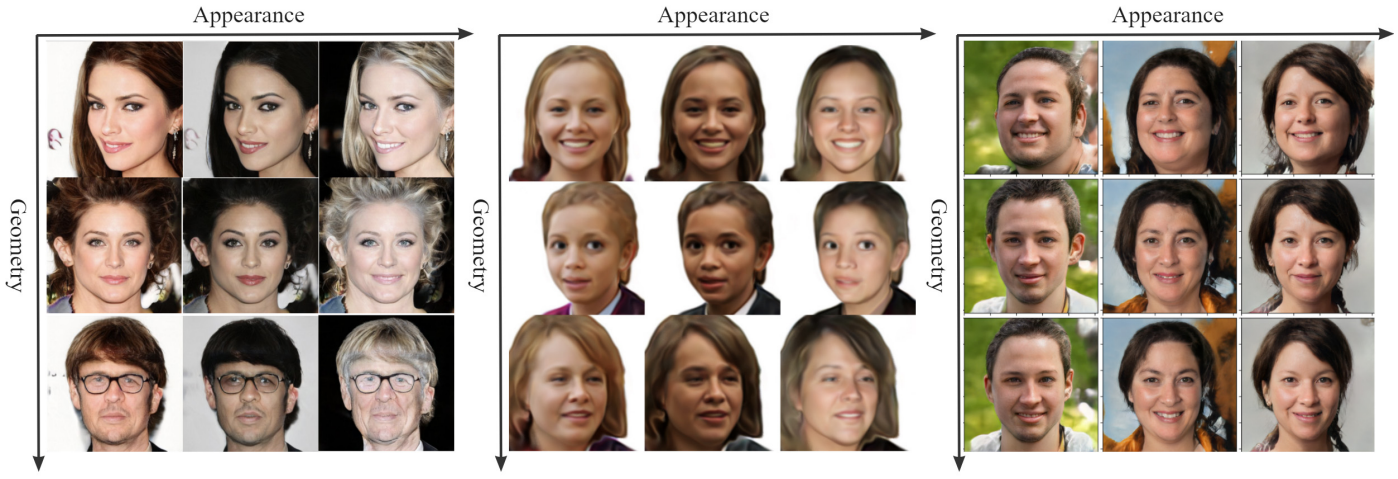
Encoder Controlled Diffusion. Several recent works have explored the combination of diffusion models and external encoders for facial image generation. Zhang et al. [11] proposed a novel framework called ControlNet for controlling the diffusion-based image generation process, while Mou and Wang [12] proposed T2I-Adapters to align internal knowledge in large-scale text-to-image models with external control signals for more granular generation control in color and structure. However, facial geometry-controlled generation based on depth maps under these frameworks have problems with the consistency of facial feature geometric shapes. The geometric shape of the generated image will be severely affected by text prompts, making it difficult to maintain the original geometric structure in its entirety.

2.2. Geometry & Appearance Disentanglement

Our work involves the extraction and retention of geometric image information. In this regard, many excellent previous works contributed in modifying other properties of the picture while preserving the geometric information in the picture.

3D-aware GAN Methods.. Disentanglement based on 3D-aware GAN achieved good performance in geometry & appearance Disentanglement due to their natual 3D encoding structure. In GIRAFFE [13], CodeNeRF[14],GARF[15], piGAN[16] and EG3D [17], the 3D representation and color information are implicitly preserved in the high-dimensional feature space. While in Disentangled3D [1],Tewari *et al.* explicitly model the forward and inverse deformation fields for the calculation of dense correspondences between images generated. These methods disentangle the geometric information from the appearance information before the synthesis process. The shape and appearance vectors are directly embedded in different gaussian distributed spaces. However, existing images or shapes cannot be used to calculate embedding vectors for synthesis guidance in these models. AniFaceGAN [18] and 3D-FM GAN [19] are other 3D-aware GANs that have achieved success in using the parametric 3D face model's latent space to represent implicit facial identity information. These methods based on existing parameterized facial models haven't achieve further disentanglement of identity, especially complete separation of geometric and appearance features.

The 3D-aware GAN approach faces another challenge: obtaining 3D models, particularly those created by non-expert or amateur users, often yields models with incomplete or significantly inconsistent human body structures, as demonstrated by the mesh in the StyleSDF generated dataset in section 4.1 of



(a) CoordGAN (b) Disentangled3D (c) Ours

Fig. 2: Comparison between the effects of how geometry and appearance information is disentangled. CoordGAN decouples the shape and texture of the image, but a geometric change can still cause a texture change in the final result. For example, in the third row and the third column of Figure 2, the texture does not match the features in the same column. Disentangled3D is implemented based on the NeRF method, but the resulting image has a lower resolution. In our method, geometric changes can be seen as deformations of the 3D mesh corresponding to the face, ensuring that texture details remain stable.

this study. Mapping these models reasonably to the generated latent variable domain is a challenge. This is similar to the problem faced by Sun *et al*. [20] since they try to provide a 3D-semantics-aware generative model. In our case, we need to create a network that can uniformly map such flawed geometric structures to the same latent variable vicinity to produce geometric results that are within a reasonable range and consistent with the original image.

2D GAN Methods.. It is research with a rich background to realize disentanglement in the generation space of existing 2D GAN generators and meaningful control of the attributes [21, 22, 23, 24, 25, 26, 2]. Zhu *et al*. [26] achieved disentanglement of shape and texture codes with basic GAN structure. Their work includes an example-based texture transfer task for encoding a real input image into texture code. Recently, in the field of face synthesis, the StyleGAN structure by Karras *et al*. [27, 28] have achieved quite a success, and much decoupling research is based on this network structure. Research of Abdal *et al*. [21], Mu *et al*. [2] and Nitzan *et al*. [29] focus on the task of disentangling human face attributes. These researches aim at different attribute control ranges. However, the desired inputs in our task are across data domains, and the geometric attributes should come from the 3D mesh of the human head. It is necessary to design a 3D-to-2D encoder to extract expressive geometric features from 3D mesh.

In finding a suitable rendering method for synthesizing geometric details, we applied the idea based on the extension of the StyleGAN2 generator similar to StylePoseGAN in the image generation step. StylePoseGAN [30] of Sarkar *et al*. and Pose with Style [31] of Albahar *et al*. apply this method to extend the original StyleGAN generator to accept the conditioning of pose and appearance separately. Both realize the disentanglement of pose and appearance attributes, which has great application value, making virtual try-on possible.

3. Method

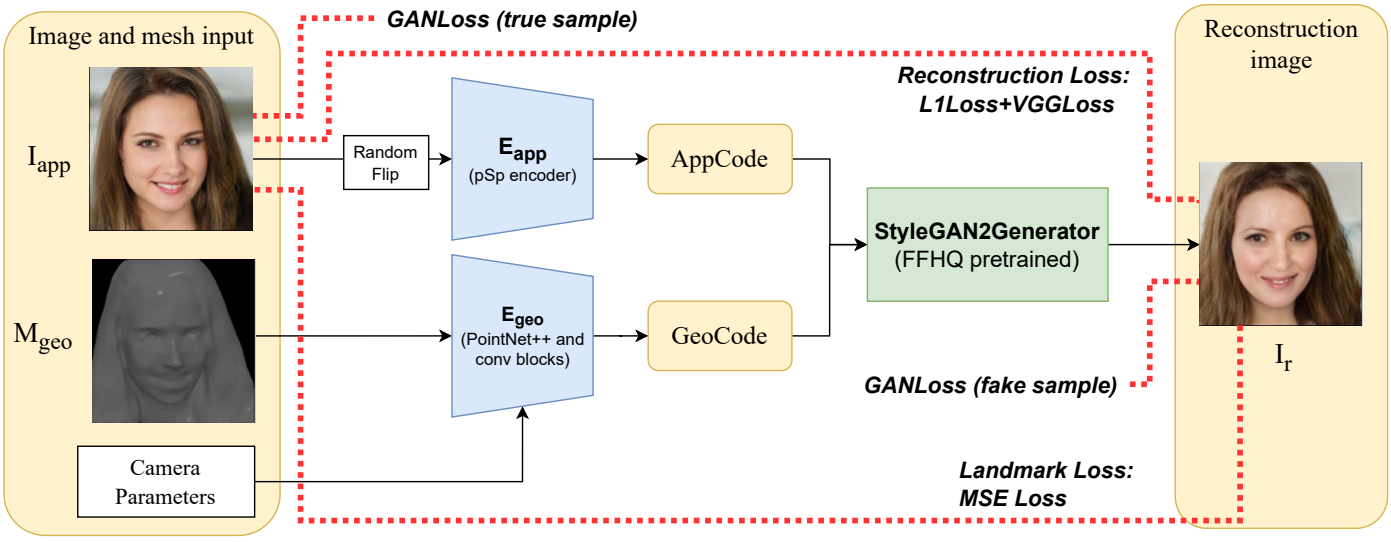
3.1. Architecture Overview

The proposed method takes a triangle mesh M_{geo} and a 2D RGB image I_{app} as input. Camera parameters P_{cam} need to be provided. The output is an image I_t that represents the shading outcome of the 3D mesh in the 2D appearance. Our network consists of four modules: geometry encoder E_{geo}, appearance encoder E_{app}, generator G, and landmark detector E_{lnd}. During training, G and E_{lnd} remain fixed while E_{geo} and E_{app} are fine-tuned. An additional image I_{geo} with the same geometric information as I_{app} will also be used as input.

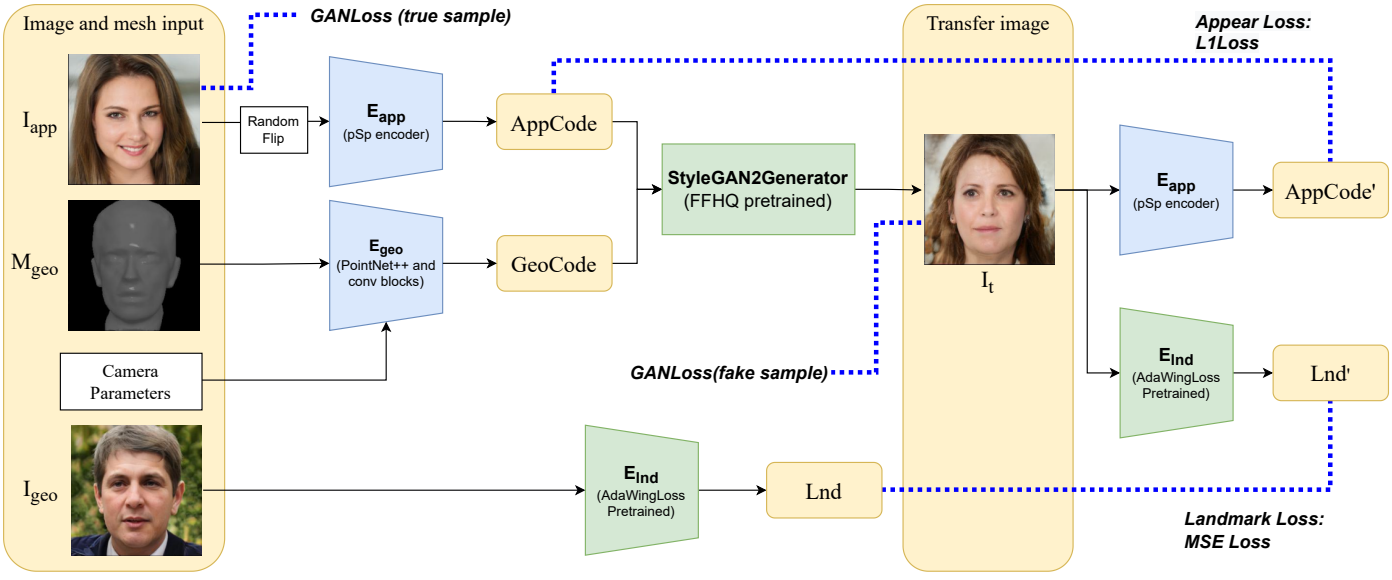
Geometry Encoder. We employed PointNet++[4] by Qi et al. as the structural foundation for the 3D encoding section of E_{geo}. The official version of the PointNet++ framework designed for segmentation tasks was utilized to obtain the feature vector, which every vertex should have. While preserving the fundamental concept of PointNet, we align the output of the latent variables with our input range by employing 1D convolution, batch normalization, and tanh activation function. A custom differentiable rendering pipeline was implemented in PyTorch3D[32] for the rasterization step. A pseudo-image that is of size 128×128 and exhibits 24-channels, carrying geometric data for each pixel, is generated by the rendering pipeline. To adapt the resulting pseudo-image into a form feasible for StyleGAN2 Generator, we used a straightforward 2D convolutional encoder, consisting of five simple-to-implement convolutional blocks following the StylePoseGAN approach[30]. The resultant *GeoCode* vector is a 2D encoding with a shape of $(N, H = 4, W = 4, C = 512)$.

3.2. Training and Losses

Appearance Encoder. To design a 2D encoder (E_{app}) for appearance attributes, challenges include handling inputs beyond StyleGAN2's training domain and aligning with its latent space



(a) Reconstruction branch of the network.



(b) Transfer branch of the network.

Fig. 3: An overview of our network structure. E_{lnd} and StyleGAN2 Generator do not participate in training and directly load pretrained weights. The network handles two tasks during training: reconstruction and transfer. The two tasks are carried out in a joint training manner, with different loss weights.

distribution. Richardson *et al.* (2021) introduced an improved pixel2style2pixel encoder framework with a hierarchical structure that can directly map images to a W_+ vector, better suited for StyleGAN2 generator training. We adopt the encoder structure outlined in this framework as our E_{app}. This allows us to utilize the pre-trained weights as the starting point for network training.

Generator. The proposed method uses a pre-trained StyleGAN2 generator (G)[28] on the FFHQ dataset to generate 256*256 resolution images. *GeoCode* replaces the initial random constant input, while $AppCode$ is directly fed into G as W_+.The background information and lighting details also come from the appearance image and are implicitly encoded within $AppCode$. The completion of the background is automatically achieved by the generator according to $AppCode$.

Landmark Detector. In order to establish adequate geometric supervision, we apply a facial landmark detector E_{lnd} based on AdaptiveWingLoss of Wang *et al.* [33] and intercept 98 channels of landmark heatmap of the network output as the final output of E_{lnd}.

Joint Training. During training, the network will perform two tasks: reconstruction and attribute transfer as seen in 3a and 3b. For the reconstruction task, the input Mesh M_{geo} contains the same geometrical information as reference image I_{app}. Through experiments, we observed that adding in cycle consistency loss of $AppCode$ in the reconstruction branch creates unnecessary complexity and could harm reconstruction performance without proper fine-tuning of hyperparameters. Therefore, the network only utilizes pixel-by-pixel reconstruction loss, landmark loss, and adversarial loss to compare the generated image I_r and the original image I_{app}.

In the attribute transfer task, we will not use reconstruction loss but instead use cycle consistency losses for constraints to supervise that the generated image I_t has geometric information that is consistent with M_{geo} and appearance information that is consistent with I_{app}. Both training tasks are described by the following function, differing only in the choice of inputs and losses:

$$I_{output} = G(E_{geo}(M_{geo}), E_{app}(I_{app})). \tag{1}$$

Reconstruction Branch. Within this task branch, we construct the reconstruction loss using the learned perceptual image patch similarity (LPIPS) metric and L_1 loss, where LPIPS is based on the AlexNet pretrained network implementation of Zhang *et al.* [34]. LPIPS enables a more precise evaluation of visual similarity between images and aids in the optimization of models and algorithms based on human perception. The training loss of the recons truction task is the weighted sum of pixel-wise reconstruction loss and adversarial loss. The adversarial loss L_{adv-r} implementation used in our paper, including the losses of generator and discriminator, is implemented as in the original StyleGAN2[28]. For details, please refer to the relevant introduction in the original paper. The LPIPS metric applied as loss is formulated as:

$$L_{lpips} = LPIPS(I_r, I_{app}). \tag{2}$$

Landmark loss a L_2 cycle consistency loss which is based on landmarks extracted by E_{lnd}:

$$L_{lnd-r} = \|E_{lnd}(I_r) - E_{lnd}(I_{geo})\|_2. \tag{3}$$

The final reconstruction loss L_{rec} can be formulated as:

$$\begin{aligned} L_{rec} = \lambda_{lpips}L_{lpips} + \lambda_{L1}\|I_r - I_{app}\|_1 + \lambda_{adv-r}L_{adv-r} \\ +\lambda_{lnd-r}L_{lnd-r}, \end{aligned} \tag{4}$$

while λ_{lpips}, λ_{L1}, λ_{adv-r} and λ_{lnd-r} are weight hyperparameters for each loss.

Transfer Branch. The transfer branch loss is the weighted sum of the loss on the appearance L_{app} and the landmark L_{lnd-t}, and the adversarial loss L_{adv-t}. Appearance loss L_{app} is also based on cycle consistency, which requires $AppCode$ of the image to be consistent before and after generation, even when random flip F_{flip} is applied:

$$L_{app} = \|E_{app}(F_{flip}(I_t)) - E_{app}(F_{flip}(I_{app}))\|_1. \tag{5}$$

According to the design of our dataset, it can be considered that there is exact geometric information between M_{geo} and I_{geo}, so direct landmark supervision for I_{geo} and I_t can obtain the same 3D supervision performance:

$$L_{lnd-t} = \|E_{lnd}(I_t) - E_{lnd}(I_{geo})\|_2. \tag{6}$$

The final transfer loss can be formulated as:

$$L_t = \lambda_{app}L_{app} + \lambda_{adv-t}L_{adv-t} + \lambda_{lnd-t}L_{lnd-t}, \tag{7}$$

while λ_{app}, λ_{adv-t} and λ_{lnd-t} are weight hyperparameters for each part of the final loss.

Random Flipping. To achieve better disentanglement, the output of E_{app} should not depend on the geometric information in the image. We achieve this by manipulating the geometric information in the reference picture input. The appearance encoder randomly flips the picture input, and a loss constraint requires the appearance encoder to output the same encoding result for different flips of the same picture.

Fine-level Latent Clipping. For the supervision of appearance, we refer to the practice of GAN embedding [9, 28] and only constrain the fine-level feature that most affects the appearance details of the picture. We only supervise the part of the $W+$ vector corresponding to StyleGAN2's 64×64 and following layers.

Before the training begins, we preprocess the generated dataset by reducing the image resolution to 256*256. The number of triangle faces of each 3D mesh is reduced to within 20,000 triangles using the fast quadric mesh simplification algorithm.

Fig. 4: Synthesized images obtained by our network structure. The top row is the input mesh M_{geo} rendered with Phong shader, assigned with gray material and fixed point light illumination, and the leftmost row is the reference image input I_{app}. The picture at each row and column intersection is generated with the input of I_{app} corresponding to this row and M_{geo} corresponding to this column. I_{app} and M_{geo} come from the same identity when the indexes of rows and columns are equal.

4. Evaluation

4.1. Dataset Setting and Evaluation Metrics

We need a face dataset with corresponding 3D meshes to properly supervise our network. However, it is challenging to create a training dataset with enough accurate pairs of images and meshes due to cost and technical difficulties. Therefore, we chose to apply StyleSDF[5] by Or-El *et al.* to generate 2D images and matching 3D meshes via their 3D-aware generator pretrained on FFHQ Dataset.

To generate the training dataset, we follow a process comprising of several steps. Firstly, a sample is generated by feeding a random Gaussian vector as the Z vector and a set of camera parameters into the StyleSDF generator. The camera parameter sampling method is as follows: the field of view half angle in degrees is fixed at 6, the elevation angle is uniformly sampled within the range of (-0.15,0.15), while the azimuth angle is in the range of (-0.3,0.3). The camera position is always placed on the unit sphere, and the near and far clip distances are 0.88 and 1.12, respectively.

4.2. Implementation Details

The networks were trained using a batch size of 2 for 10 epochs on a dataset containing 10000 generated samples. Two optimizers were applied to the generator ($Adam_G$) and discriminator ($Adam_D$) with varying learning rates ($lr_G = 5e^{-6}$, $lr_D = 1e^{-5}$) but with the same β and eps configuration ((β_1,β_2) = (0.9,0.99), eps=$1e^{-8}$). Additionally, the following hyperparameters were used to generate the final results: λ_{lpips}=2, λ_{L1}=2, λ_{lnd-r}=0.05, λ_{adv-r}=1, λ_{app}=0.02, λ_{lnd-t}=0.05, and λ_{adv-t}=1. Each training step, on average, requires 0.627 seconds to be completed. The average inference time for each individual image is 0.22 seconds.

4.3. Qualitative Evaluation

We randomly sampled several data not included in the training set as the network's input, and the resulting image is shown in Figure 4. The lighting direction and ambient light information of the generated face are independent relative to the input mesh. As the character's head rotates, the left half of the face becomes obscured by shadows. Referring to Figure 2, it can be observed that our generated image is more in line with the task requirements of decoupling geometry and appearance. We can achieve changes in geometry without causing changes in appearance.

4.4. Quantitative Evaluation

4.4.1. Quality and Diversity

To quantify the quality and diversity of the results we generate, we use the same experiment configuration as StyleFlow. In the experiment, we will compare our architecture with the current successful researches in the field of disentanglement such as Image2StyleGAN[22], InterfaceGAN[24], GANSpace[25] and StyleFlow[21]. We randomly generated 1k images and calculated Frechet Inception Distance (FID)[35],Kernel Inception Distance (KID)[**?**] and LPIPS metrics between the outputs and the appearance image inputs. The results are shown in Table 1.

Network	FID↓	LPIPS↓	KID×100↓
Image2StyleGAN	82.49	0.64	0.199
InterfaceGAN	67.08	0.65	0.202
GANSpace	64.69	0.62	0.180
StyleFlow	**53.15**	0.57	0.172
Ours (full)	59.80	**0.35**	**0.171**

Table 1: Use metrics to calculate how similar the images generated by our network through unpaired M_{geo} and I_{app} are to the original dataset. The results of some network configurations are from StyleFlow.

Considering that our network uses a dataset randomly sampled from the generation space of StyleSDF, the generative diversity of our network is limited by the sampling method and the generative capacity of the original generative model. However, it already outperforms other methods on the LPIPS metric of generated images. The difference in perception between our generated image and the input appearance image is smaller.

4.4.2. Multiview Identity Consistency.

Fig. 5: Identity consistency test. The number shown in the lower left corner of each image in the figure is the identity features' L2 distance between this image and the image in the leftmost column of corresponding row. The leftmost column is 0 because it is compared with itself.

An experiment was conducted to verify a network's ability to preserve identity consistency through a process of inputting the same mesh and image with varying mesh poses, and checking if the generated results maintain identity consistency. For each experiment configuration, we performed 200 sets of sampling. Each set included five randomly generated results with the same appearance but different fixed poses. From the five results, we selected one as the standard and calculated the L_2 distance between its identity features and those of the other four results. We averaged these distances for each group, and the average result of 100 groups indicated the level of identity change. To measure identity change, we used the advanced ArcFace algorithm [36] to compute identity features.

Previous research on ID-disentanglement [29] has demonstrated excellent performance in decoupling identity and attribute. Meanwhile, to compare our 2D method with 3D-aware methods, we also selected StyleNeRF[37] and EG3D[17] for comparison. Additionally, we conducted a small ablation study

Network	ArcFace$_{\times 10^{-2}}$ ↓
StyleNeRF (FFHQ)	0.2127
EG3D (FFHQ)	0.1260
ID-disentangle (our dataset)	0.1670
ID-disentangle (FFHQ)	0.1453
Ours (w/o L_{lnd-r})	0.1636
Ours (w/o jawline landmarks)	0.1542
Ours (full)	0.1326

Table 2: Identity consistency test results.

to identify aspects of our network's design that impact identity preservation. We considered two modifications – removing the landmark loss from the reconstruction branch and not applying landmark supervision to the jawline, similar to ID-disentanglement. Our fully optimized network produced better results than ID-disentanglement, highlighting the importance of landmark supervision.

While surpassing 2D GAN methods, our model also achieved similar identity preservation ability as 3D-aware GAN. This indicates that, during rotation, the features extracted from the head's 3D model remain stable and can correctly guide the generation of facial geometric features.

4.4.3. Geometry Consistency

In this experiment, we sampled and transformed 100 meshes into inputs for various networks. Each network generated 100 images with the same appearance but different geometric structures. To assess geometry consistency, we compared facial landmark positions in the generated and ground truth images using metrics such as mean average error (MAE), root mean squared error (RMSE), and the average of maximum displacement among all corresponding landmarks (MaxΔ). They are all measured in pixel distance.

The experiment revealed a significant impact arising from the decoupling between the detection and the input conditions in the original architecture system. Even subtle influence on geometric shape by other parameters during the generation process can have a noteworthy adverse impact on facial feature point metrics, despite their imperceptibility to human observation. The experiment demonstrated that our model exhibits better geometric consistency than other comparable frameworks.

Network	RMSE ↓	MAE ↓	MaxΔ ↓
CoordGAN	4.55	5.189	4.173
Disentangled3D	7.34	5.166	9.150
EG3D	3.82	5.012	4.966
Ours (full)	**3.43**	**3.450**	**4.138**

Table 3: Geometry consistency test results.

4.5. Ablation Study

We implemented various designs to enhance the network's disentanglement performance. In our study, we removed different components of the network with the same training configuration (as shown in Table 4) to obtain multiple training results with different structures. For each configuration, we performed 1,000 samplings and calculated FID scores, LPIPS metrics, and the average L_2 distance of the Arcface features to the respective reference inputs. The results are categorized in Table 4.

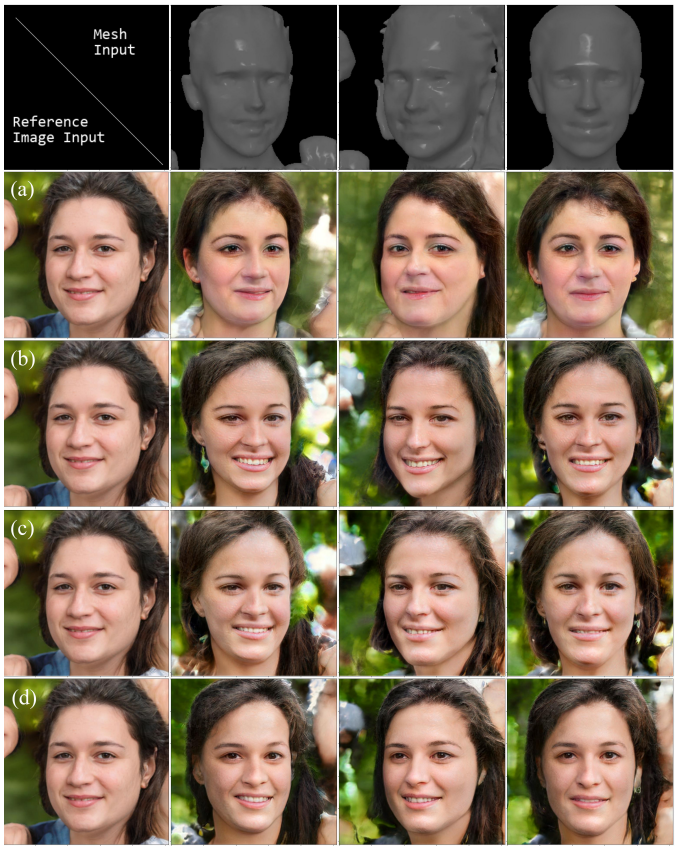


Fig. 6: Visualized results of the ablation study. We use different meshes and same reference image to generate results with different configurations. (a) Without latent clipping. (b) Without random flip. (c) Without L_{lnd-r}. (d) Full model.

Configuration	FID↓	LPIPS↓	ArcFace$_{\times 10^{-2}}$ ↓
w/o L_{app}*	-	-	-
w/o L_{lnd-t}*	-	-	-
w/o latent clipping	51.87	0.46	0.3140
w/o random flipping	63.03	0.38	0.3091
w/o L_{lnd-r}	60.21	0.36	0.2922
Full	59.80	0.35	0.2857

Table 4: Results of FID score, LPIPS metric and ArcFace features's L_2 distance with various components removed. * Removing L_{app} or L_{lnd-t} will cause the network to fail to converge to meaningful results.

Removing the latent clipping method from our network would cause a considerable performance decline. Removing supervision for the appearance code of shallow layers of the StyleGAN2 generator would lead to a drop in overall generation ability, as indicated by the FID score of the full architecture. Implementing latent clipping can considerably enhance the disengagement property of the network (as seen in Figure 6), leading to visually closer output to the reference image.

Also, in the early trained network, we did not add Landmark loss to the reconstruction branch. In this case, the images obtained from the reconstruction training have lower LPIPS Loss, but there are slight differences in facial contours. Adding Landmark loss can improve the overall network performance in terms of geometric similarity.

5. Conclusion

We propose an end-to-end method that synthesizes outputs using 3D mesh and 2D image-guided inputs. Our approach incorporates StyleGAN latent embedding, a 3D Encoder based on PointNet++, and a carefully designed dataset for supervision. By disentangling geometry and appearance, we achieve more comprehensive results than existing research. Through experiments, we demonstrate the advantages of our disentanglement objective and the success of our approach in terms of diversity and quality.

Nevertheless, our method still has limitations. It currently relies on the paired dataset generated by StyleSDF, thus introducing defects of StyleSDF itself into our network. The mesh we use currently needs to be more stable for hair representation, so the shape of hair in the generated pictures can be stably controlled (see in the last row of Figure 4). Meanwhile, our generator is a pretrained StyleGAN2 generator. The success of StyleGAN2 depends on high-quality datasets such as FFHQ. It is still difficult to apply the conversion of 3D mesh to 2D images in other domains. The method we use also faces the problem of inaccurate landmark supervision that affects performance. Additionally, our network has the potential to achieve further performance improvements by utilizing more efficient computing frameworks[38], that can expedite the convergence of model training and enhance the speed of inference. We anticipate that the network will achieve a significant level of efficiency improvement in the future when the JGAN model zoo[39] is applied.

References

[1] Tewari, A, R, MB, Pan, X, Fried, O, Agrawala, M, Theobalt, C. Disentangled3D: Learning a 3d generative model with disentangled geometry and appearance from monocular images. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, p. 1506–1515. doi:10.1109/CVPR52688.2022.00157.

[2] Mu, J, Mello, SD, Yu, Z, Vasconcelos, N, Wang, X, Kautz, J, et al. CoordGAN: Self-supervised dense correspondences emerge from GANs. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, p. 10001–10010. doi:10.1109/CVPR52688.2022.00977.

[3] Zhao, H, Gallo, O, Frosio, I, Kautz, J. Loss functions for image restoration with neural networks. IEEE Transactions on Computational Imaging 2017;3(1):47–57. doi:10.1109/TCI.2016.2644865.

[4] Qi, CR, Yi, L, Su, H, Guibas, LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17. ISBN 9781510860964; 2017, p. 5105–5114. doi:10.48550/arXiv.1706.02413.

[5] OrEl, R, Luo, X, Shan, M, Shechtman, E, Park, J, Kemelmacher-Shlizerman, I. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, p. 13493–13503. doi:10.1109/CVPR52688.2022.01314.

[6] Su, J, Chu, H, Huang, J. Instance-aware image colorization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020, p. 7965–7974. doi:10.1109/CVPR42600.2020.00799.

[7] Liu, X, Wan, L, Qu, Y, Wong, TT, Lin, S, Leung, CS, et al. Intrinsic colorization. In: ACM SIGGRAPH Asia 2008 Papers. SIGGRAPH Asia '08. ISBN 9781450318310; 2008, p. 152:1–9. doi:10.1145/1457515.1409105.

[8] Li, H, Sheng, B, Li, P, Ali, R, Chen, CLP. Globally and locally semantic colorization via exemplar-based broad-gan. IEEE Transactions on Image Processing 2021;30:8526–8539. doi:10.1109/TIP.2021.3117061.

[9] Luo, X, Zhang, XC, Yoo, P, Martin-Brualla, R, Lawrence, J, Seitz, SM. Time-travel rephotography. ACM Trans Graph 2021;40(6):213:1–12. doi:10.1145/3478513.3480485.

[10] Lai, CS, You, Z, Huang, CC, Tsai, YH, Chiu, WC. Colorization of depth map via disentanglement. In: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII. ISBN 978-3-030-58570-9; 2020, p. 450–466. doi:10.1007/978-3-030-58571-6_27.

[11] Zhang, L, Agrawala, M. Adding conditional control to text-to-image diffusion models. 2023. doi:10.48550/arXiv.2302.05543. arXiv:2302.05543.

[12] Mou, C, Wang, X, Xie, L, Wu, Y, Zhang, J, Qi, Z, et al. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. 2023. doi:10.48550/arXiv.2302.08453. arXiv:2302.08453.

[13] Niemeyer, M, Geiger, A. GIRAFFE: Representing scenes as compositional generative neural feature fields. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, p. 11448–11459. doi:10.1109/CVPR46437.2021.01129.

[14] Jang, W, Agapito, L. CodeNeRF: Disentangled neural radiance fields for object categories. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021, p. 12929–12938. doi:10.1109/ICCV48922.2021.01271.

[15] Schwarz, K, Liao, Y, Niemeyer, M, Geiger, A. GRAF: Generative radiance fields for 3d-aware image synthesis. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20. ISBN 9781713829546; 2020,doi:10.48550/arXiv.2007.02442.

[16] Chan, ER, Monteiro, M, Kellnhofer, P, Wu, J, Wetzstein, G. pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, p. 5795–5805. doi:10.1109/CVPR46437.2021.00574.

[17] Chan, ER, Lin, CZ, Chan, MA, Nagano, K, Pan, B, de Mello, S, et al. Efficient geometry-aware 3D generative adversarial networks. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, p. 16102–16112. doi:10.1109/CVPR52688.2022.01565.

[18] Wu, Y, Deng, Y, Yang, J, Wei, F, Chen, Q, Tong, X. AnifaceGAN: Animatable 3d-aware face image generation for video avatars. In: Koyejo, S, Mohamed, S, Agarwal, A, Belgrave, D, Cho, K, Oh, A, editors. Advances in Neural Information Processing Systems; vol. 35. 2022, p. 36188–36201. doi:10.48550/arXiv.2210.06465.

[19] Liu, Y, Shu, Z, Li, Y, Lin, Z, Zhang, R, Kung, S. 3d-fm GAN: towards 3d-controllable face manipulation. In: Avidan, S, Brostow, GJ, Cissé, M, Farinella, GM, Hassner, T, editors. Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XV; vol. 13675 of *Lecture Notes in Computer Science*. Springer; 2022, p. 107–125. doi:10.1007/978-3-031-19784-0_7.

[20] Sun, J, Wang, X, Shi, Y, Wang, L, Wang, J, Liu, Y. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. ACM Transactions on Graphics 2022;41(6):270:1–10. doi:10.1145/3550454.3555506.

[21] Abdal, R, Zhu, P, Mitra, NJ, Wonka, P. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Trans Graph 2021;40(3). doi:10.1145/3447648.

[22] Abdal, R, Qin, Y, Wonka, P. Image2styleGAN: How to embed images into the stylegan latent space? In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019, p. 4431–4440. doi:10.1109/ICCV.2019.00453.

[23] Abdal, R, Qin, Y, Wonka, P. Image2styleGAN++: How to edit the

embedded images? In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020, p. 8293–8302. doi:10.1109/CVPR42600.2020.00832.

[24] Shen, Y, Yang, C, Tang, X, Zhou, B. InterfaceGAN: Interpreting the disentangled face representation learned by GANs. IEEE Trans Pattern Anal Mach Intell 2022;44(4):2004–2018. doi:10.1109/TPAMI.2020.3034267.

[25] Härkönen, E, Hertzmann, A, Lehtinen, J, Paris, S. GANSpace: Discovering interpretable GAN controls. In: Larochelle, H, Ranzato, M, Hadsell, R, Balcan, M, Lin, H, editors. Advances in Neural Information Processing Systems; vol. 33. 2020, p. 9841–9850. doi:10.48550/arXiv.2004.02546.

[26] Zhu, J, Zhang, Z, Zhang, C, Wu, J, Torralba, A, Tenenbaum, J, et al. Visual object networks: Image generation with disentangled 3d representations. In: Bengio, S, Wallach, HM, Larochelle, H, Grauman, K, Cesa-Bianchi, N, Garnett, R, editors. Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018. 2018, p. 118–129. doi:10.48550/arXiv.1812.02725.

[27] Karras, T, Laine, S, Aila, T. A style-based generator architecture for generative adversarial networks. IEEE Trans Pattern Anal Mach Intell 2021;43(12):4217–4228. doi:10.1109/TPAMI.2020.2970919.

[28] Karras, T, Laine, S, Aittala, M, Hellsten, J, Lehtinen, J, Aila, T. Analyzing and improving the image quality of styleGAN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020. 2020, p. 8107–8116. doi:10.1109/CVPR42600.2020.00813.

[29] Nitzan, Y, Bermano, A, Li, Y, Cohen-Or, D. Face identity disentanglement via latent space mapping. ACM Trans Graph 2020;39(6):225:1–14. doi:10.1145/3414685.3417826.

[30] Sarkar, K, Golyanik, V, Liu, L, Theobalt, C. Style and pose control for image synthesis of humans from a single monocular view. CoRR 2021;abs/2102.11263. URL: https://arxiv.org/abs/2102.11263. arXiv:2102.11263.

[31] AlBahar, B, Lu, J, Yang, J, Shu, Z, Shechtman, E, Huang, J. Pose with Style: Detail-preserving pose-guided image synthesis with conditional styleGAN. ACM Trans Graph 2021;40(6):218:1–11. doi:10.1145/3478513.3480559.

[32] Ravi, N, Reizenstein, J, Novotný, D, Gordon, T, Lo, W, Johnson, J, et al. Accelerating 3d deep learning with pytorch3d. CoRR 2020;abs/2007.08501. URL: https://arxiv.org/abs/2007.08501. arXiv:2007.08501.

[33] Wang, X, Bo, L, Li, F. Adaptive wing loss for robust face alignment via heatmap regression. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019. IEEE; 2019, p. 6970–6980. doi:10.1109/ICCV.2019.00707.

[34] Zhang, R, Isola, P, Efros, AA, Shechtman, E, Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018. Computer Vision Foundation / IEEE Computer Society; 2018, p. 586–595. doi:10.1109/CVPR.2018.00068.

[35] Heusel, M, Ramsauer, H, Unterthiner, T, Nessler, B, Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I, von Luxburg, U, Bengio, S, Wallach, HM, Fergus, R, Vishwanathan, SVN, et al., editors. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017. 2017, p. 6626–6637. doi:10.48550/arXiv.1706.08500.

[36] Deng, J, Guo, J, Yang, J, Xue, N, Kotsia, I, Zafeiriou, S. ArcFace: Additive angular margin loss for deep face recognition. IEEE Trans Pattern Anal Mach Intell 2022;44(10):5962–5979. doi:10.1109/TPAMI.2021.3087709.

[37] Gu, J, Liu, L, Wang, P, Theobalt, C. StyleNeRF: A style-based 3d aware generator for high-resolution image synthesis. In: International Conference on Learning Representations. 2022,URL: https://openreview.net/forum?id=iUuzzTMUw9K.

[38] Hu, S, Liang, D, Yang, G, Yang, G, Zhou, W. Jittor: a novel deep learning framework with meta-operators and unified graph execution. Sci China Inf Sci 2020;63(12). doi:10.1007/s11432-020-3097-4.

[39] Zhou, W, Yang, G, Hu, S. Jittor-gan: A fast-training generative adversarial network model zoo based on jittor. Comput Vis Media 2021;7(1):153–157. doi:10.1007/s41095-021-0203-2.