arXiv:2012.05903v1 [cs.CV] 10 Dec 2020

Portrait Neural Radiance Fields from a Single Image

Chen Gao
Virginia Tech

Yichang Shih
Google

Wei-Sheng Lai
Google

Chia-Kai Liang
Google

Jia-Bin Huang
Virginia Tech

https://portrait-nerf.github.io

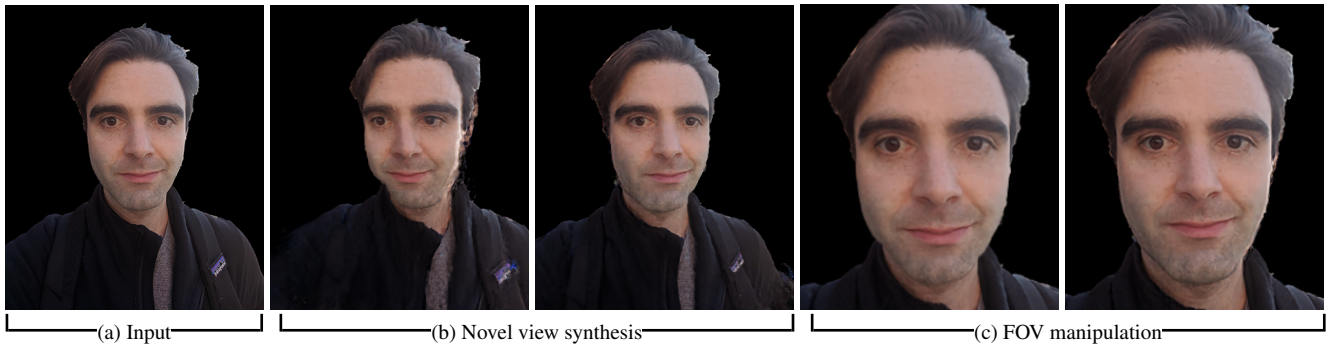


Figure 1. **Applications of the proposed method.** Given a single portrait image (a) as input, our method produces a *portrait neural radiance field* that facilitates photorealistic face editing tasks such as (b) novel view synthesis and (c) field-of-view (FOV) manipulation, where the left and right are rendered with wider and narrower camera FOVs, respectively.

Abstract

We present a method for estimating Neural Radiance Fields (NeRF) from a single headshot portrait. While NeRF has demonstrated high-quality view synthesis, it requires multiple images of static scenes and thus impractical for casual captures and moving subjects. In this work, we propose to pretrain the weights of a multilayer perceptron (MLP), which implicitly models the volumetric density and colors, with a meta-learning framework using a light stage portrait dataset. To improve the generalization to unseen faces, we train the MLP in the canonical coordinate space approximated by 3D face morphable models. We quantitatively evaluate the method using controlled captures and demonstrate the generalization to real portrait images, showing favorable results against state-of-the-arts.

1. Introduction

Portrait view synthesis enables various post-capture edits and computer vision applications, such as pose manipulation [9], selfie perspective distortion (foreshortening) correction [68, 14, 39], improving face recognition accuracy by view normalization [69], and greatly enhancing the 3D viewing experiences. Compared to 3D reconstruction and view synthesis for generic scenes, portrait view synthesis requires a higher quality result to avoid the uncanny valley, as human eyes are more sensitive to artifacts on faces or inaccuracy of facial appearances.

To achieve high-quality view synthesis, the filmmaking production industry densely samples lighting conditions and camera poses synchronously around a subject using a light stage [10]. The process, however, requires an expensive hardware setup and is unsuitable for casual users. Reconstructing the facial geometry from a *single capture* requires face mesh templates [6] or a 3D morphable model [4, 7, 5, 29]. While the quality of these 3D model-based methods has been improved dramatically via deep networks [16, 62], a common limitation is that the model only covers the center of the face and excludes the upper head, hairs, and torso, due to their high variability. These excluded regions, however, are critical for natural portrait view synthesis.

Recently, neural implicit representations emerge as a promising way to model the appearance and geometry of 3D scenes and objects [50, 38, 32]. For example, Neural Radiance Fields (NeRF) demonstrates high-quality view synthesis by implicitly modeling the volumetric density and color using the weights of a multilayer perceptron (MLP). However, training the MLP requires capturing images of *static subjects* from *multiple viewpoints* (in the order of 10-100 images) [38, 35]. It is thus impractical for portrait view synthesis because a slight subject movement or inaccurate camera pose estimation degrades the reconstruction quality.

In this paper, we propose to train an MLP for modeling the radiance field using a *single headshot portrait* illustrated in Figure 1. Unlike NeRF [38], training the MLP with a sin-

gle image from scratch is fundamentally ill-posed, because there are infinite solutions where the renderings match the input image. Our key idea is to pretrain the MLP and finetune it using the available input image to adapt the model to an unseen subject's appearance and shape.

To pretrain the MLP, we use densely sampled portrait images in a light stage capture. However, using a naïve pretraining process that optimizes the reconstruction error between the synthesized views (using the MLP) and the rendering (using the light stage data) over the subjects in the dataset performs poorly for *unseen* subjects due to the diverse appearance and shape variations among humans. We address the challenges in two novel ways. First, we leverage gradient-based meta-learning techniques [12] to train the MLP in a way so that it can quickly adapt to an unseen subject. Second, we propose to train the MLP in a canonical coordinate by exploiting domain-specific knowledge about the face shape. Our experiments show favorable quantitative results against the state-of-the-art 3D face reconstruction and synthesis algorithms on the dataset of controlled captures. We further show that our method performs well for real input images captured in the wild and demonstrate foreshortening distortion correction as an application.

In this work, we make the following contributions:

- We present a single-image view synthesis algorithm for portrait photos by leveraging meta-learning. Our method produces a *full reconstruction*, covering not only the facial area but also the upper head, hairs, torso, and accessories such as eyeglasses.
- We propose an algorithm to pretrain NeRF in a canonical face space using a rigid transform from the world coordinate. We show that compensating the shape variations among the training data substantially improves the model generalization to unseen subjects.
- We provide a multi-view portrait dataset consisting of controlled captures in a light stage. Our data provide a way of quantitatively evaluating portrait view synthesis algorithms.

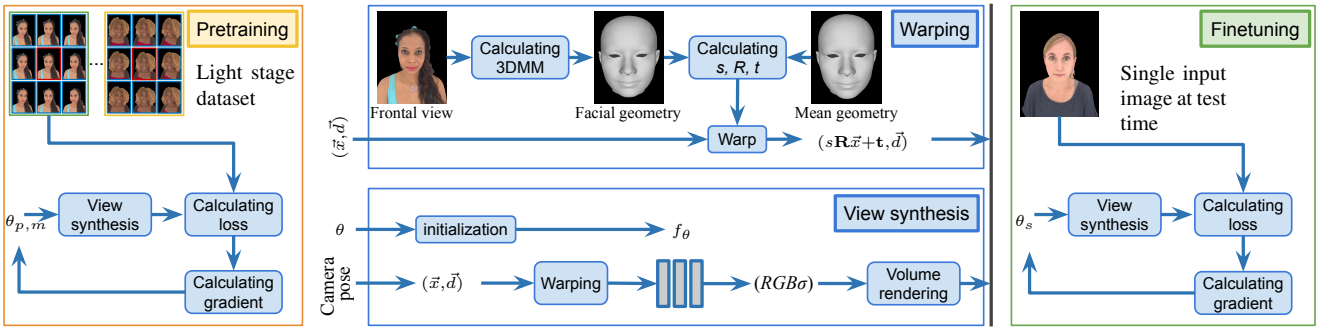
2. Related Work

View synthesis with neural implicit representations. Our method builds on recent work of neural implicit representations [50, 38, 33, 65, 3, 35, 61] for view synthesis. NeRF [38] represents the scene as a mapping $\mathcal{F}$ from the world coordinate and viewing direction to the color and occupancy using a compact MLP. Given a camera pose, one can synthesize the corresponding view by aggregating the radiance over the light ray cast from the camera pose using standard volume rendering. The MLP is trained by minimizing the reconstruction loss between synthesized views and the corresponding ground truth input images. Existing methods require tens to hundreds of photos to train a scene-specific NeRF network. In contrast, our method requires only *one single image* as input. We finetune the pretrained weights learned from light stage training data [10, 36] for unseen inputs. Compared to the unstructured light field [37, 13, 46, 43], volumetric rendering [34], and image-based rendering [20, 19], our single-image method does not require estimating camera pose [47]. In our experiments, the pose estimation is challenging at the complex structures and view-dependent properties, like hairs and subtle movement of the subjects between captures. Similarly to the neural volume method [34], our method improves the rendering quality by sampling the warped coordinate from the world coordinates.

Existing single-image view synthesis methods model the scene with point cloud [41, 58], multi-plane image [56, 22], or layered depth image [48, 27]. Our method focuses on headshot portraits and uses an implicit function as the neural representation. We demonstrate foreshortening correction as applications [68, 14, 39]. Instead of training the warping effect between a set of *pre-defined* focal lengths [68, 39], our method achieves the perspective effect at *arbitrary* camera distances and focal lengths.

Face pose manipulation. Conditioned on the input portrait, generative methods learn a face-specific Generative Adversarial Network (GAN) [18, 24, 25] to synthesize the target face pose driven by exemplar images [60, 44, 42, 54, 26, 64], rig-like control over face attributes via face model [52, 15, 17, 28], or learned latent code [11, 1]. While the outputs are photorealistic, these approaches have common artifacts that the generated images often exhibit inconsistent facial features, identity, hairs, and geometries across the results and the input image. As a strength, we preserve the texture and geometry information of the subject across camera poses by using the 3D neural representation invariant to camera poses [53, 40] and taking advantage of pose-supervised training [63].

3D face modeling. Reconstructing face geometry and texture enables view synthesis using graphics rendering pipelines. Existing single-image methods use the symmetric cues [59], morphable model [4, 7, 5, 29], mesh template deformation [6], and regression with deep networks [23]. However, these *model-based* methods only reconstruct the regions where the model is defined, and therefore do not handle hairs and torsos, or require a separate explicit hair modeling as post-processing [62, 21, 30]. Our method takes the benefits from both face-specific modeling and view synthesis on generic scenes. To leverage the domain-specific knowledge about faces, we train on a portrait dataset and propose the canonical face coordinates using the 3D face proxy derived by a morphable model. To model the portrait subject, instead of using face meshes consisting only the facial landmarks, we use the finetuned NeRF at the test time to include hairs and torsos.



(a) Pretrain NeRF (b) Warp to canonical coordinate (c) Finetune

Figure 2. **Overview.** Our method builds on the MLP network f_θ in NeRF [38], and requires only a single image as input at the test time. To learn the geometry and shape priors for single-image view synthesis, we (a) pretrain the model parameter θ_p using the light stage dataset (Section 3.1) consisting of multiple training views and subjects indexed by m (Section 3.2). To improve the generalization, we (b) warp the world coordinate $\vec{x}$ to the canonical coordinate derived from the 3D model through a rigid transform $(s, \mathbf{R}, \mathbf{t})$, and feed forward the warped coordinate and viewing direction $\vec{d}$ to f_θ (Section 3.3). In test time, (c) we finetune the model parameter against the input view to output θ_s, and use f_{θ_s} and volume rendering from color and occupancy ($RGB\sigma$) for view synthesis (Section 3.4).

Meta-learning. Our work is closely related to meta-learning and few-shot learning [45, 2, 12, 8, 51, 55]. To explain the analogy, we consider view synthesis from a camera pose as a *query*, captures associated with the known camera poses from the light stage dataset as *labels*, and training a subject-specific NeRF as a *task*. Training NeRFs for different subjects is analogous to training classifiers for various tasks. At the test time, given a single label from the frontal capture, our goal is to optimize the *testing task*, which learns the NeRF to answer the queries of camera poses. We leverage gradient-based meta-learning algorithms [12, 49] to learn the weight initialization for the MLP in NeRF from the *meta-training tasks*, *i.e.*, learning a single NeRF for different subjects in the light stage dataset.

3. Algorithm

Figure 2 illustrates the overview of our method, which consists of the pretraining and testing stages. In the pretraining stage, we train a coordinate-based MLP (same in NeRF) f_θ on diverse subjects captured from the light stage and obtain the pretrained model parameter optimized for generalization, denoted as θ_p^* (Section 3.2). The high diversities among the real-world subjects in identities, facial expressions, and face geometries are challenging for training. We address the variation by normalizing the world coordinate to the canonical face coordinate using a rigid transform and train a shape-invariant model representation (Section 3.3). At the test time, we initialize the NeRF with the pretrained model parameter θ_p^* and then finetune it on the frontal view for the input subject s. We use the finetuned model parameter (denoted by θ_s^*) for view synthesis (Section 3.4).

3.1. Training data

Our training data consists of light stage captures over multiple subjects. Each subject is lit uniformly under controlled lighting conditions. We process the raw data to reconstruct the depth, 3D mesh, UV texture map, photometric normals, UV glossy map, and visibility map for the subject [67, 36]. For each subject, we render a sequence of 5-by-5 training views by uniformly sampling the camera locations over a solid angle centered at the subject's face at a fixed distance between the camera and subject. We span the solid angle by 25° field-of-view vertically and 15° horizontally. We set the camera viewing directions to look straight to the subject. Figure 3 and supplemental materials show examples of 3-by-3 training views. The center view corresponds to the front view expected at the test time, referred to as the *support* set $\mathcal{D}_s$, and the remaining views are the target for view synthesis, referred to as the *query* set $\mathcal{D}_q$. We refer to the process training a NeRF model parameter for subject m from the support set as a *task*, denoted by $\mathcal{T}_m$.

3.2. Pretraing NeRF

Our goal is to pretrain a NeRF model parameter θ_p^* that can easily adapt to capturing the appearance and geometry of an unseen subject. We loop through K subjects in the dataset, indexed by $m = \{0, ..., K-1\}$, and denote the model parameter pretrained on the subject m as $\theta_{p,m}$. We sequentially train on subjects in the dataset and update the pretrained model as $\{\theta_{p,0}, \theta_{p,1}..., \theta_{p,K-1}\}$, where the last parameter is outputted as the final pretrained model, *i.e.*, $\theta_p^* = \theta_{p,K-1}$. For each task $\mathcal{T}_m$, we train the model on $\mathcal{D}_s$ and $\mathcal{D}_q$ alternatively in an inner loop, as illustrated in Figure 3. Since $\mathcal{D}_s$ is available at the test time, we only need to propagate the gradients learned from $\mathcal{D}_q$ to the pretrained model θ_p^*, which transfers the common representations un-

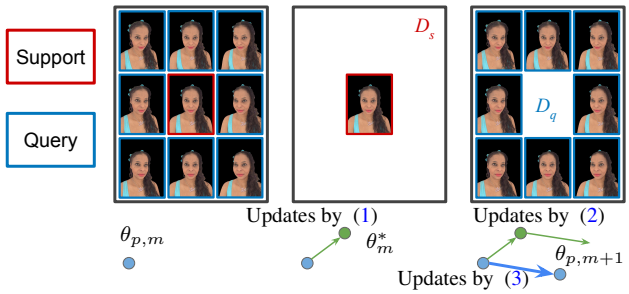


Figure 3. **Pretraining with meta-learning framework.** The training views of each subject m consists of the frontal view, called support set D_s, and the rest views, called query set D_q. In each iteration, we update the pretrained parameter from $\theta_{p,m}$ to $\theta_{p,m+1}$ using two steps. We first update the parameter to θ^*_m by finetuning on the D_s using (1). Then we continue to update θ^*_m using (2) on D_q, and feedback the update to the pretrained parameter $\theta_{p,m+1}$ using (3).

seen from the front view $\mathcal{D}_s$ alone, such as the priors on head geometry and occlusion.

Pretraining on $\mathcal{D}_s$. For the subject m in the training data, we initialize the model parameter from the pretrained parameter learned in the previous subject $\theta_{p,m-1}$, and set $\theta_{p,-1}$ to random weights for the first subject in the training loop. We train a model θ^*_m optimized for the front view of subject m using the L_2 loss between the front view predicted by f_{θ_m} and $\mathcal{D}_s$, denoted as $\mathcal{L}_{\mathcal{D}_s}(f_{\theta_m})$. The optimization iteratively updates the θ^t_m for N_s iterations as the following:

$$\theta^{t+1}_m = \theta^t_m - \alpha \nabla_\theta \mathcal{L}_{\mathcal{D}_s}(f_{\theta^t_m}), \tag{1}$$

where $\theta^0_m = \theta_{p,m-1}$, $\theta^*_m = \theta^{N_s-1}_m$, and α is the learning rate.

Pretraining on $\mathcal{D}_q$. We proceed the update using the loss between the prediction from the known camera pose and the query dataset $\mathcal{D}_q$. Since $\mathcal{D}_q$ is unseen during the test time, we feedback the gradients to the pretrained parameter $\theta_{p,m}$ to improve generalization. The update is iterated N_q times as described in the following:

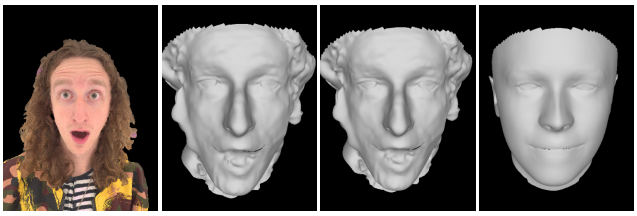
$$\theta^{t+1}_m = \theta^t_m - \beta \nabla_\theta \mathcal{L}_{\mathcal{D}_q}(f_{\theta^t_m}), \tag{2}$$

$$\theta^{t+1}_{p,m} = \theta^t_{p,m} - \beta \nabla_\theta \mathcal{L}_{\mathcal{D}_q}(f_{\theta^t_m}), \tag{3}$$

where $\theta^0_m = \theta^*_m$ learned from D_s in (1), $\theta^0_{p,m} = \theta_{p,m-1}$ from the pretrained model on the previous subject, and β is the learning rate for the pretraining on $\mathcal{D}_q$. After N_q iterations, we update the pretrained parameter by the following:

$$\theta_{p,m} = \theta^{N_q-1}_{p,m} \tag{4}$$

Note that (3) does not affect the update of the current subject m, *i.e.*, (2), but the gradients are carried over to the subjects in the subsequent iterations through the pretrained model parameter update in (4). The training is terminated after visiting the entire dataset over K subjects. The pseudo code of the algorithm is described in the supplemental material.



(a) Subject (b) World coordinate (c) Canonical face coordinate (d) Mean face

Figure 4. **Rigid transform between the world and canonical face coordinate.** For every subject (a), we detect the face mesh (b), and fit a rigid transform (c) using the vertices correspondence between (b) and the mean face of the dataset (d). We warp from the world (b) to the canonical (c) coordinate, and feed the warped coordinate to NeRF to predict color and occupancy. The meshes (b-d) are for visualization and not used in view synthesis. Only the rigid transform is used in our work.

Discussion. We assume that the order of applying the gradients learned from $\mathcal{D}_q$ and $\mathcal{D}_s$ are interchangeable, similarly to the first-order approximation in MAML algorithm [12]. We transfer the gradients from $\mathcal{D}_q$ independently of $\mathcal{D}_s$. For better generalization, the gradients of $\mathcal{D}_s$ will be adapted from the input subject at the test time by finetuning, instead of transferred from the training data. In our experiments, applying the meta-learning algorithm designed for image classification [55] performs poorly for view synthesis. This is because each update in view synthesis requires gradients gathered from *millions of samples* across the scene coordinates and viewing directions, which do not fit into a single batch in modern GPU. Our method takes a lot more steps in a single meta-training task for better convergence.

3.3. Canonical face space

To address the face shape variations in the training dataset and real-world inputs, we normalize the world coordinate to the canonical space using a rigid transform and apply f_θ on the warped coordinate. Specifically, for each subject m in the training data, we compute an approximate facial geometry F_m from the frontal image using a 3D morphable model and image-based landmark fitting [7]. We average all the facial geometries in the dataset to obtain the mean geometry $\overline{F}$. During the training, we use the vertex correspondences between F_m and $\overline{F}$ to optimize a rigid transform by the SVD decomposition (details in the supplemental documents). The transform is used to map a point $\mathbf{x}$ in the subject's world coordinate to $\mathbf{x}'$ in the face canonical space: $\mathbf{x}' = s_m \mathbf{R}_m \mathbf{x} + \mathbf{t}_m$, where $s_m, \mathbf{R}_m$ and $\mathbf{t}_m$ are the optimized scale, rotation, and translation.

During the prediction, we first warp the input coordi-

(a) Input (front view) (b) Top-left (c) Bottom-left (d) Bottom (e) Top-Right (f) Right (g) Disparity on (a)

Figure 5. **View synthesis from a single front view**. Our method finetunes the pretrained model on (a), and synthesizes the new views using the controlled camera poses (c-g) relative to (a). The results in (c-g) look realistic and natural. To validate the face geometry learned in the finetuned model, we render the (g) disparity map for the front view (a). The videos are accompanied in the supplementary materials.

nate from the world coordinate to the face canonical space through $(s_m, \mathbf{R}_m, \mathbf{t}_m)$. We then feed the warped coordinate to the MLP network f_θ to retrieve color and occlusion (Figure 4). The warp makes our method robust to the variation in face geometry and pose in the training and testing inputs, as shown in Table 3 and Figure 10. In our method, the 3D model is used to obtain the rigid transform $(s_m, \mathbf{R}_m, \mathbf{t}_m)$. We do not require the mesh details and priors as in other model-based face view synthesis [62, 7].

3.4. Finetuning and rendering

At the test time, only a single frontal view of the subject s is available. We first compute the rigid transform described in Section 3.3 to map between the world and canonical coordinate. Then, we finetune the pretrained model parameter θ_p^* by repeating the iteration in (1) for the input subject and outputs the optimized model parameter θ_s^*. To render novel views, we sample the camera ray in the 3D space, warp to the canonical space, and feed to $f_{\theta_s^*}$ to retrieve the radiance and occlusion for volume rendering.

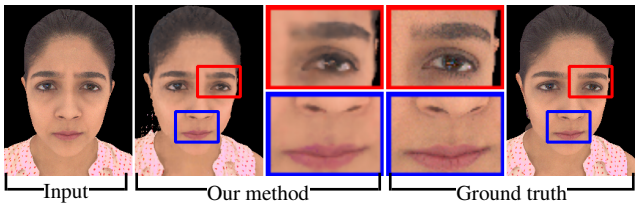


Figure 6. **Comparisons to the ground truth.** Our results faithfully preserve the details like skin textures, personal identity, and facial expressions from the input.

4. Experimental Results

Our dataset consists of 70 different individuals with diverse gender, races, ages, skin colors, hairstyles, accessories, and costumes. For each subject, we capture 2-10 different expressions, poses, and accessories on a light stage under fixed lighting conditions. We include challenging cases where subjects wear glasses, are partially occluded on faces, and show extreme facial expressions and curly

Input Jackson *et al*. [23] Xu *et al*. [62] Our method Ground truth Jackson *et al*. [23] Xu *et al*. [62] Our method Ground truth

Figure 7. **Comparison to the state-of-the-art portrait view synthesis on the light stage dataset.** In each row, we show the input frontal view and two synthesized views using [23, 62] and our method. The method by Jackson *et al*. [23] recovers the geometry only in the center of the face. The deep 3D portrait [62] is capable of reconstructing the full head (including hair and ear), but fails to preserve eye gaze, expression, face shape, identity, and hairstyle when comparing to the ground truth. Our method produces photorealistic results closest to the ground truth, as shown in the numerical evaluations in Table 1.

hairstyles. In total, our dataset consists of 230 captures. We hold out six captures for testing. We render the support $\mathcal{D}_s$ and query $\mathcal{D}_q$ by setting the camera field-of-view to 84°, a popular setting on commercial phone cameras, and sets the distance to 30cm to mimic selfies and headshot portraits taken on phone cameras.

Figure 5 shows our results on the diverse subjects taken in the wild. The subjects cover different genders, skin colors, races, hairstyles, and accessories. We stress-test the challenging cases like the glasses (the top two rows) and curly hairs (the third row). Our results look realistic, preserve the facial expressions, geometry, identity from the input, handle well on the occluded area, and successfully synthesize the clothes and hairs for the subject. Our method generalizes well due to the finetuning and canonical face coordinate, closing the gap between the unseen subjects and the pretrained model weights learned from the light stage dataset. When the face pose in the inputs are slightly rotated away from the frontal view, *e.g*., the bottom three rows of Figure 5, our method still works well. In the supplemental video, we hover the camera in the spiral path to demonstrate the 3D effect. Our method preserves temporal coherence in challenging areas like hairs and occlusion, such as the nose and ears.

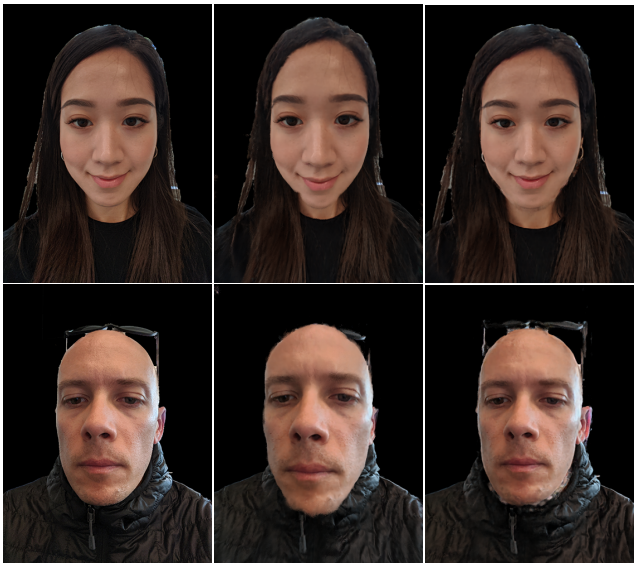
Figure 6 compares our results to the ground truth using the subject in the test hold-out set. Our method precisely controls the camera pose, and faithfully reconstructs the details from the subject, as shown in the insets.

Table 1. **View synthesis metrics.** We report the average PSNR, SSIM and LPIPS metrics against the existing methods over the testing set from our light stage dataset.

	PSNR ↑	SSIM ↑	LPIPS ↓
Jackson *et al*. [23]	10.23	0.4356	0.485
Xu *et al*. [62]	18.91	0.5609	0.276
Our method	**23.92**	**0.7688**	**0.161**

Comparisons. Figure 7 compares our method to the state-of-the-art face pose manipulation methods [62, 23] on six testing subjects held out from the training. The subjects cover various ages, gender, races, and skin colors. We obtain the results of Jackson *et al*. [23] using the official implementation[1]. The results from [62] were kindly provided by the authors. The work by Jackson *et al*. [23] only covers the face area. The learning-based head reconstruction method from Xu *et al*. [62] generates plausible results but fails to preserve the gaze direction, facial expressions, face shape, and the hairstyles (the bottom row) when comparing to the ground truth. Our method is visually similar to the ground truth, synthesizing the entire subject, including hairs and body, and faithfully preserving the texture, lighting, and expressions. We report the quantitative evaluation using PSNR, SSIM, and LPIPS [66] against the ground truth in Table 1.

[1] http://aaronsplace.co.uk/papers/jackson2017recon



(a) Input (b) Closer to subject (c) Further

Figure 8. **Perspective effect manipulation.** Given an input (a), we virtually move the camera closer (b) and further (c) to the subject, while adjusting the focal length to match the face size. We manipulate the perspective effects such as dolly zoom in the supplementary materials.

Table 2. **Ablation study on initialization methods.**

Initialization	PSNR ↑	SSIM ↑	LPIPS ↓
Random	14.99	0.5763	0.491
Pretrain by (5)	22.87	0.7824	0.215
Our method	**23.70**	**0.8051**	**0.178**

Table 3. **Ablation study on canonical face coordinate.**

Coordinate	PSNR ↑	SSIM ↑	LPIPS ↓
World	24.80	0.8167	0.172
Canonical (our method)	**24.98**	**0.8178**	**0.156**

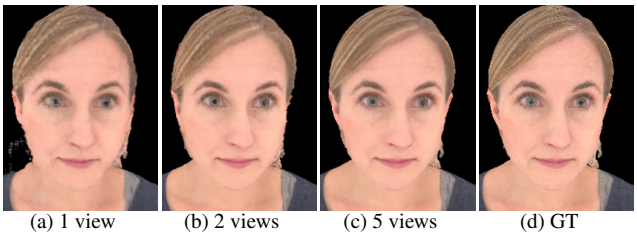
Perspective manipulation. Portraits taken by wide-angle cameras exhibit undesired foreshortening distortion due to the perspective projection [14, 68]. By virtually moving the camera closer or further from the subject and adjusting the focal length correspondingly to preserve the face area, we demonstrate perspective effect manipulation using portrait NeRF in Figure 8 and the supplemental video. When the camera sets a longer focal length, the nose looks smaller, and the portrait looks more natural. Since our training views are taken from a single camera distance, the vanilla NeRF rendering [38] requires inference on the world coordinates outside the training coordinates and leads to the artifacts when the camera is too far or too close, as shown in the supplemental materials. We address the artifacts by re-parameterizing the NeRF coordinates to infer on the training coordinates.

(a) Random (b) Pretrain by (5) (c) Our method (d) GT

Figure 9. **Ablation study on different weight initialization.** (a) Training the MLP from scratch using random weight initialization fails to converge and leads to poor results. (b) Pretraining using (5) suffers from blurry rendering and artifacts. (c) Our method produces the best quality compared to the ground truth (d).

(a) Input (b) World coordinate (c) Our method

Figure 10. **Ablation study on face canonical coordinates.** Our method using (c) canonical face coordinate shows better quality than using (b) world coordinate on chin and eyes.



(a) 1 view PSNR = 24.98 (b) 2 views PSNR = 27.70 (c) 5 views PSNR = 32.45 (d) GT

Figure 11. **Ablation study on the number of input views during testing.** Our results improve when more views are available.

Table 4. **Ablation study on training sizes.**

Training size	PSNR ↑	SSIM ↑	LPIPS ↓
18	24.61	0.8110	0.163
27	24.98	**0.8178**	**0.156**
59	**25.04**	0.8129	0.164
100	24.65	0.8120	0.165

4.1. Ablation study

Initialization. Figure 9 compares the results finetuned from different initialization methods. Without any pretrained prior, the random initialization [38] in Figure 9(a) fails to learn the geometry from a single image and leads to poor view synthesis quality. Next, we pretrain the model parameter by minimizing the L_2 loss between the prediction and the training views across all the subjects in the dataset as

Table 5. **Ablation study on number of input views.**

Number of input views	Random initialization			Pretrain with (5)			Our method		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
1	17.34	0.6988	0.347	24.47	0.8109	0.175	**24.98**	**0.8178**	**0.156**
2	23.21	0.7936	0.214	27.37	0.8445	0.130	**27.70**	**0.8647**	**0.115**
5	31.12	0.8958	0.094	31.57	0.8826	0.107	**32.45**	**0.9045**	**0.090**

the following:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_m \mathcal{L}_{\mathcal{D}_s}(f_\theta) + \mathcal{L}_{\mathcal{D}_q}(f_\theta), \tag{5}$$

where m indexes the subject in the dataset. Figure 9(b) shows that such a pretraining approach can also learn geometry prior from the dataset but shows artifacts in view synthesis. The synthesized face looks blurry and misses facial details. Our pretraining in Figure 9(c) outputs the best results against the ground truth. The quantitative evaluations are shown in Table 2.

Canonical face coordinate. Figure 10 and Table 3 compare the view synthesis using the face canonical coordinate (Section 3.3) to the world coordinate. Without warping to the canonical face coordinate, the results using the world coordinate in Figure 10(b) show artifacts on the eyes and chins. Our method outputs a more natural look on face in Figure 10(c), and performs better on quality metrics against ground truth across the testing subjects, as shown in Table 3.

Training task size. Our method does not require a large number of training tasks consisting of many subjects. In Table 4, we show that the validation performance saturates after visiting 59 training tasks. To balance the training size and visual quality, we use 27 subjects for the results shown in this paper. Compared to the majority of deep learning face synthesis works, *e.g.*, [62], which require *thousands of individuals* as the training data, the capability to generalize portrait view synthesis from a smaller subject pool makes our method more practical to comply with the privacy requirement on personally identifiable information.

Input views in test time. Our method can incorporate multi-view inputs associated with known camera poses to improve the view synthesis quality. At the finetuning stage, we compute the reconstruction loss between each input view and the corresponding prediction. We show the evaluations on different number of input views against the ground truth in Figure 11 and comparisons to different initialization in Table 5. Compared to the vanilla NeRF using random initialization [38], our pretraining method is highly beneficial when very few (1 or 2) inputs are available. The margin decreases when the number of input views increases and is less significant when 5+ input views are available.

(a) Background (b) Non-frontal view

Figure 12. **Limitations.** Left and right in (a) and (b): input and output of our method. (a) When the background is not removed, our method cannot distinguish the background from the foreground and leads to severe artifacts. (b) When the input is not a frontal view, the result shows artifacts on the hairs.

5. Conclusions

We presented a method for portrait view synthesis using a single headshot photo. Our method builds upon the recent advances of neural implicit representation and addresses the limitation of generalizing to an unseen subject when only one single image is available. Specifically, we leverage gradient-based meta-learning for pretraining a NeRF model so that it can quickly adapt using light stage captures as our meta-training dataset. We also address the shape variations among subjects by learning the NeRF model in canonical face space. We validate the design choices via ablation study and show that our method enables natural portrait view synthesis compared with state of the arts.

Limitations. As illustrated in Figure 12(a), our method cannot handle the subject background, which is diverse and difficult to collect on the light stage. Users can use off-the-shelf subject segmentation [57] to separate the foreground, inpaint the background [31], and composite the synthesized views to address the limitation. Our method requires the input subject to be roughly in frontal view and does not work well with the profile view, as shown in Figure 12(b). Extrapolating the camera pose to the unseen poses from the training data is challenging and leads to artifacts.

Future work. We are interested in generalizing our method to class-specific view synthesis, such as cars or human bodies. Extending NeRF to portrait video inputs and addressing temporal coherence are exciting future directions. Our work is a first step toward the goal that makes NeRF practical with casual captures on hand-held devices. Addressing the finetuning speed and leveraging the stereo cues in dual camera popular on modern phones can be beneficial to this goal.

References

[1] Yazeed Alharbi and Peter Wonka. Disentangled image generation through structured noise injection. In *CVPR*, 2020. 2

[2] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. 3

[3] Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. X-fields: Implicit neural view-, light- and time-image interpolation. *ACM TOG*, 2020. 2

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *CGIT*, 1999. 1, 2

[5] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniah, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *CVPR*, 2016. 1, 2

[6] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM TOG*, 2013. 1, 2

[7] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *TVCG*, 2013. 1, 2, 4, 5

[8] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 3

[9] Antonio Criminisi, Jamie Shotton, Andrew Blake, and Philip HS Torr. Gaze manipulation for one-to-one teleconferencing. In *ICCV*, 2003. 1

[10] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *CGIT*, 2000. 1, 2

[11] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *CVPR*, 2020. 2

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017. 2, 3, 4

[13] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *CVPR*, 2019. 2

[14] Ohad Fried, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM TOG*, 2016. 1, 2, 7

[15] Baris Gecer, Binod Bhattarai, Josef Kittler, and Tae-Kyun Kim. Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3d morphable model. In *ECCV*, 2018. 2

[16] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *CVPR*, 2018. 1

[17] Partha Ghosh, Pravir Singh Gupta, Roy Uziel, Anurag Ranjan, Michael Black, and Timo Bolkart. Gif: Generative interpretable faces. *arXiv:2009.00149*, 2020. 2

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[19] Peter Hedman and Johannes Kopf. Instant 3d photography. *ACM TOG*, 2018. 2

[20] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM TOG*, 2018. 2

[21] Liwen Hu, Chongyang Ma, Linjie Luo, and Hao Li. Single-view hair modeling using a hairstyle database. *ACM TOG*, 2015. 2

[22] Hsin-Ping Huang, Hung-Yu Tseng, Hsin-Ying Lee, and Jia-Bin Huang. Semantic view synthesis. In *ECCV*, 2020. 2

[23] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. In *ICCV*, 2017. 2, 6

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2

[26] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM TOG*, 2018. 2

[27] Johannes Kopf, Kevin Matzen, Suhib Alsisan, Ocean Quigley, Francis Ge, Yangming Chong, Josh Patterson, Jan-Michael Frahm, Shu Wu, Matthew Yu, et al. One shot 3d photography. *ACM TOG*, 2020. 2

[28] Marek Kowalski, Stephan J Garbin, Virginia Estellers, Tadas Baltrušaitis, Matthew Johnson, and Jamie Shotton. Config: Controllable neural face image generation. *ECCV*, 2020. 2

[29] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM TOG*, 2017. 1, 2

[30] Shu Liang, Xiufeng Huang, Xianyu Meng, Kunyao Chen, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. Video to fully automatic 3d hair model. *ACM TOG*, 2018. 2

[31] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. 8

[32] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 1

[33] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[34] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM TOG*, 2019. 2

[35] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv:2008.02268*, 2020. 1, 2

[36] Abhimitra Meka, Rohit Pandey, Christian Haene, Sergio Orts-Escolano, Peter Barnum, Philip Davidson, Daniel Erickson, Yinda Zhang, Jonathan Taylor, Sofien Bouaziz, Chloe Legendre, Wan-Chun Ma, Ryan Overbeck, Thabo Beeler, Paul Debevec, Shahram Izadi, Christian Theobalt, Christoph Rhemann, and Sean Fanello. Deep relightable textures - volumetric performance capture with neural rendering. *ACM TOG*, 2020. 2, 3

[37] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM TOG*, 2019. 2

[38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 1, 2, 3, 7, 8

[39] Koki Nagano, Huiwen Luo, Zejian Wang, Jaewoo Seo, Jun Xing, Liwen Hu, Lingyu Wei, and Hao Li. Deep face normalization. *ACM TOG*, 2019. 1, 2

[40] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 2

[41] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D Ken Burns effect from a single image. *ACM TOG*, 38(6):1–15, 2019. 2

[42] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject agnostic face swapping and reenactment. In *ICCV*, 2019. 2

[43] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM TOG*, 2017. 2

[44] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a face: Towards arbitrary high fidelity face manipulation. In *ICCV*, 2019. 2

[45] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2017. 3

[46] Gernot Riegler and Vladlen Koltun. Free view synthesis. *ECCV*, 2020. 2

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[48] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2

[49] Vincent Sitzmann, Eric R Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *NeurIPS*, 2020. 3

[50] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 1, 2

[51] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 3

[52] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, 2020. 2

[53] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM TOG*, 2019. 2

[54] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 2

[55] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *ICLR*, 2020. 3, 4

[56] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020. 2

[57] Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM TOG*, 2018. 8

[58] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020. 2

[59] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, 2020. 2

[60] Wayne Wu, Yunxuan Zhang, Cheng Li, Chen Qian, and Chen Change Loy. Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV*, 2018. 2

[61] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *arXiv preprint arXiv:2011.12950*, 2020. 2

[62] Sicheng Xu, Jiaolong Yang, Dong Chen, Fang Wen, Yu Deng, Yunde Jia, and Xin Tong. Deep 3d portrait from a single image. In *CVPR*, 2020. 1, 2, 5, 6, 8

[63] Xiaogang Xu, Ying-Cong Chen, and Jiaya Jia. View independent generative adversarial network for novel view synthesis. In *ICCV*, 2019. 2

[64] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, 2019. 2

[65] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 2

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[67] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM TOG*, 2020. 3

[68] Yajie Zhao, Zeng Huang, Tianye Li, Weikai Chen, Chloe LeGendre, Xinglei Ren, Ari Shapiro, and Hao Li. Learning perspective undistortion of portraits. In *ICCV*, 2019. 1, 2, 7

[69] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *CVPR*, 2015. 1