VR Facial Animation via Multiview Image Translation

SHIH-EN WEI, JASON SARAGIH, TOMAS SIMON, ADAM W. HARLEY*, STEPHEN LOMBARDI, MICHAL PERDOCH, ALEXANDER HYPES, DAWEI WANG, HERNAN BADINO, and YASER SHEIKH
Facebook Reality Labs

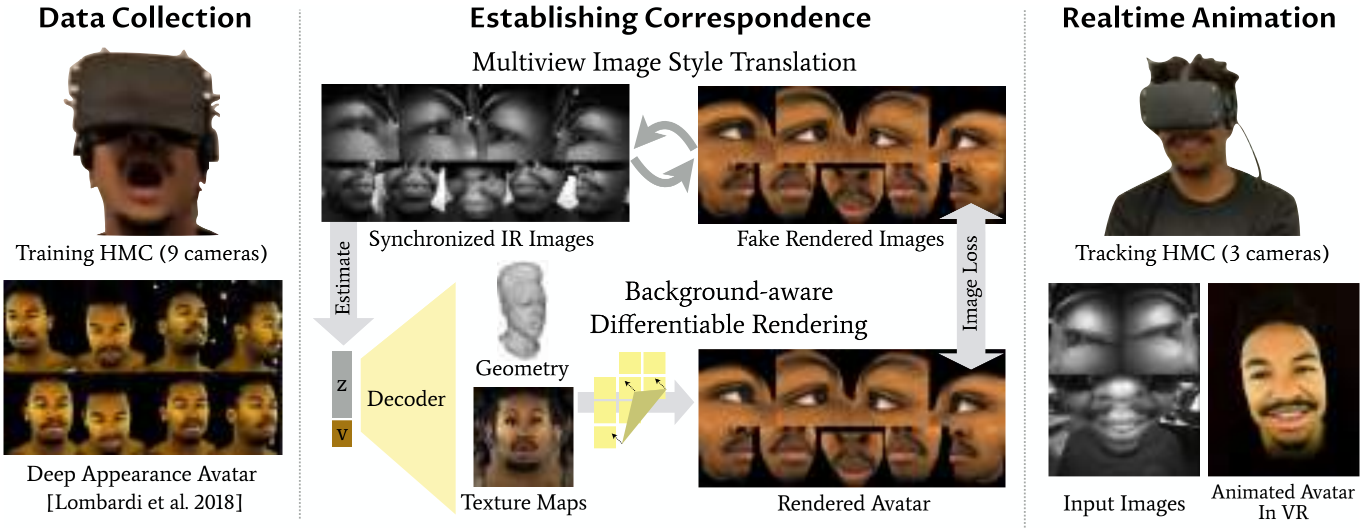

Fig. 1. We present a **VR realtime facial animation** system with headset mounted cameras (HMC) which augment a standard head mounted display (HMD). Our method establishes precise correspondence between 9 camera images from a *training* HMC and the parameters of a photorealistic avatar. Finally, we use a common subset of 3 cameras on a *tracking* HMC to animate the avatar in realtime.

A key promise of Virtual Reality (VR) is the possibility of remote social interaction that is more immersive than any prior telecommunication media. However, existing social VR experiences are mediated by inauthentic digital representations of the user (i.e., stylized avatars). These stylized representations have limited the adoption of social VR applications in precisely those cases where immersion is most necessary (e.g., professional interactions and intimate conversations). In this work, we present a bidirectional system that can animate avatar heads of both users' full likeness using consumer-friendly headset mounted cameras (HMC). There are two main challenges in doing this: unaccommodating camera views and the image-to-avatar domain gap. We address both challenges by leveraging constraints imposed by multiview geometry to establish precise image-to-avatar correspondence, which are then used to learn an end-to-end model for real-time tracking. We present designs for a *training* HMC, aimed at data-collection and model building, and a *tracking* HMC for use during interactions in VR. Correspondence between the avatar and the HMC-acquired images are automatically found through self-supervised multiview image translation, which does not require manual annotation or one-to-one correspondence between domains. We evaluate the system on a variety of users and demonstrate significant improvements over prior work.

CCS Concepts: • **Human-centered computing** → **Virtual reality**; • **Computing methodologies** → **Computer vision**; **Unsupervised learning**; **Animation**.

Additional Key Words and Phrases: Face Tracking, Unsupervised Image Style Transfer, Differentiable Rendering

ACM Reference Format:
Shih-En Wei, Jason Saragih, Tomas Simon, Adam W. Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. 2019. VR Facial Animation via Multiview Image Translation. *ACM Trans. Graph.* 38, 4, Article 67 (July 2019), 16 pages. https://doi.org/10.1145/3306346.3323030

1 INTRODUCTION

Virtual Reality (VR) has seen increased ubiquity in recent years. This has opened up the possibility for remote collaboration and interaction that is more engaging and immersive than achievable through other media. Concurrently, there has been great progress in generating accurate digital doubles and avatars. Driven by the gaming and movie industries, a number of compelling demonstrations of state of the art systems have recently attracted interest in the community [Epic Games 2017; Hellblade 2018; Magic Leap 2018; Seymour et al. 2017; Unreal Engine 4 2018]. These systems show

*Currently at Carnegie Mellon University, work done while at Facebook Reality Labs.

Authors' address: Shih-En Wei, shih-en.wei@fb.com; Jason Saragih, jason.saragih@fb.com; Tomas Simon, tomas.simon@fb.com; Adam W. Harley, aharley@cmu.edu; Stephen Lombardi, stephen.lombardi@fb.com; Michal Perdoch, michal.perdoch@oculus.com; Alexander Hypes, alexander.hypes@oculus.com; Dawei Wang, dawei.wang@oculus.com; Hernan Badino, hernan.badino@oculus.com; Yaser Sheikh, yasers@fb.com, Facebook Reality Labs, Pittsburgh, PA.

0730-0301/2019/7-ART67
https://doi.org/10.1145/3306346.3323030

highly photo-realistic avatars driven and rendered in real-time. Although impressive results are achieved, they are all designed for one-way interactions, where the actor is equipped with sensors optimally placed to capture facial expression. Unfortunately, these sensor placements are not compatible with existing VR-headset designs, which largely occlude the face. Thus, these systems are better suited to live performances than interaction.

If we consider instead works that are aimed at bidirectional communication in VR, we discover that existing systems mostly use non-photorealistic/stylized avatars [BinaryVR 2019; Li et al. 2015; Olszewski et al. 2016]. These representations tend to have a more limited range of expressivity, which renders errors from facial expression tracking less perceptible than with photo-real avatars. In this work, we argue that this is no coincidence, and that precise face tracking from consumer-friendly headset-mounted camera configurations is significantly harder than in conventional settings. There are two main reasons for this. First, instead of capturing a complete and unobscured view of the face, headset-mounted camera placements tend to provide only partial and non-overlapping views of the face at extreme and oblique views. Minimizing reconstruction errors in these viewpoints often does not translate to correct results when viewed frontally. Secondly, these cameras often operate in the infrared (IR) spectrum, which is not directly comparable to the avatar's RGB appearance and makes analysis-by-synthesis techniques less effective. To partially alleviate these difficulties, existing systems are designed to work using structurally and mechanically sub-optimal sensor designs that are more accommodating to the computer-vision tracking problem. Despite this, their performance is still only suitable for stylized avatar representations.

Although the difficult sensor configurations of headset-mounted cameras (HMC) can prove challenging for classical face alignment approaches like [Saragih et al. 2009; Xiong and la Torre 2013], in this work we show that end-to-end deep neural networks are capable of learning the complex mapping from sensor measurements to avatar parameters, given the availability of sufficient high-precision training examples relating the two domains. We demonstrate compelling results where fully expressive and accurate performances can be tracked in real time, at a precision matching the representation capacity of modern photo-realistic avatars. The challenge, then, is how to acquire the correspondences required to pose the problem as supervised learning.

Our solution for acquiring correspondence is to leverage multiview geometry in addressing both the problem of oblique viewpoints as well as the sensor-avatar domain gap. Specifically, we propose the use of a *training* HMC design that shares a sensor configuration with a consumer-friendly design (the *tracking* HMC), but has additional cameras to support better coverage and viewing angles while minimally disturbing the quality of data acquired from the shared cameras. With these additional viewpoints, classical analysis-by-synthesis constraints become more meaningful, and results generalize better to common vantage points (i.e., frontal view of the face). These additional views also provide more signal to improve the fidelity at which we can perform domain translation to address the sensor-avatar domain gap. With data collected from these cameras along with a pre-trained personalized avatar, our system learns to discover correspondences through self-supervision, without requiring any manual input or semantically defined labels. The results are highly precise estimates of the avatar's parameters that match the user's performance, which are suitable for learning an end-to-end mapping relating them to the tracking HMC.

In the following, we discuss prior work in §2, and present our method for finding image-to-avatar correspondence in §3, including hardware design, image style transfer and the core optimization problem. Real-time facial animation is then covered in §4. We present results of our approach in §5, and conclude in §6 with a discussion and directions of future work.

2 RELATED WORK

2.1 Face Tracking in VR

Tracking faces in VR is a unique and challenging problem because the object we want to track is largely occluded by the VR headset. In the literature, solutions vary in how hardware designs are used to circumvent sensing challenges, as well as in methods to bridge the domain gap between sensor data and the face representation in order to find their correspondence. Specifically, sensors used to build the face model and later drive it are typically comprised of different sets of camera configurations. Modeling sensors (i.e., cameras for building a shape and appearance model of the face) are typically multiview, high-fidelity RGB cameras with an unobstructed view of the face [Beeler et al. 2011; Dimensional Imaging 2016; Fyffe et al. 2014; Lombardi et al. 2018]. Tracking sensors (i.e., cameras for driving the face model to match a user's expressions) are typically mounted on the HMD, resulting in a patch-work of oblique and partially overlapping views of different facial parts with narrow depth of field, and typically operate in the infra-red (IR) spectrum [Li et al. 2015; Olszewski et al. 2016]. Moreover, since the HMDs obscure the face, and modeling sensors require an unobscured view, data from modeling and tracking sensors can not be captured concurrently.

As the eyebox in a VR headset is enclosed, the face is typically divided into an occluded upper face and a visible lower face. As such, some works use specialized strategies to obtain the facial state for each part separately, followed by a compositing step to get the full face state in realtime. In [Li et al. 2015] and [Thies et al. 2018], an RGBD sensor is used, which allows direct registration of the lower face with the model's geometry. To have a better viewpoint, Li et al. [2015] attach the sensor with a protruding mount, placing it slightly below the mouth. Similar to [BinaryVR 2019], this frontal viewpoint is optimal for modeling lower mouth expressions, but is not ideal from a hardware design standpoint. In [Thies et al. 2018], the sensor is placed in the environment which limits the range of a user's head pose. For the upper face, Li et al. [2015] use strain gauges to sense voltage changes that accompany facial expressions. By building a *skeleton* headset without a display unit that otherwise occludes the upper face, they can acquire face model parameters corresponding to strain gauge measurements through depth registration. They use this to train a real-time upper face blendshape regressor. However, this input signal has low SNR, exhibits drift, and contains only limited information about facial expressions. In comparison, Thies et al. [2018] use infrared (IR) cameras pointing at the eye region to avoid interfering with VR usage. They use a calibration process where subjects are instructed to gaze at known positions to obtain correspondence for training. Although effective in estimating gaze direction and eyelid blinks, it does not capture other upper-face expressions such as eyebrow motion and the temporal pattern of wrinkles in the forehead, nose and areas surrounding the eyes.

If the parametric facial model exhibits appearance statistics matching those from the sensors used to drive the model, then "analysis-by-synthesis" or "vision as inverse graphics" approaches [Kulkarni et al. 2015; Nair et al. 2008; Yildirim et al. 2015] can be used to find correspondences. Here, the challenge is to find the parameters of the models that, when rendered from the corresponding camera view, match the images obtained from the driving sensors. This is typically achieved though variants of the gradient-descent algorithm [Blanz and Vetter 1999; Cootes et al. 1998; Thies et al. 2016]. Unfortunately, for VR applications, the domain gap between imaging sensors and modeling sensors makes it unlikely that minimizing the difference between the rendered model and the sensor's image will result in an expression matching that of the user. Although explicit landmark detectors can be used to define sparse geometric correspondence between the model and tracking images [Cao et al. 2014], landmarks alone lack expressiveness, and the oblique viewpoints from HMD mounted cameras make reprojection errors less effective.

Other approaches correspond sensor data and face parameters using non-visual information such as manual semantic annotations of facial expressions and audio signals. For example, in [Olszewski et al. 2016], to model the entire face using a single RGB sensor for the lower face and IR cameras for the eyes, the subjects were instructed to performed a predefined set of expressions and sentences that were used as correspondence with a blendshape model face animation of the same content. To generate more correspondence for training a neural network with a small temporal window, dynamic time warping of the audio signal was used to align the sentences with the animation. Although the approach demonstrated realistic results, the animation tends to exhibit tokenized expressions which look plausible but are not faithful reconstructions of the user's facial motion. This approach is also limited by the granularity at which facial expressions can be expressed, as well as their repeatability.

Unpaired learning-based methods have also been investigated. In [Lombardi et al. 2018], synthetic renderings of the avatar are used together with real tracking sensor images to build a domain-conditioned variational autoencoder (VAE) while simultaneously learning a mapping from the latent space to face model parameters, using correspondences from the rendered domain exclusively. In that work, correspondences are found by leveraging parsimony in the VAE network, and the model is never trained directly for the target setting (i.e., input is headset images, output is avatar parameters). Although compelling results were demonstrated for speech sequences, its performance deteriorates with expressive content.

While our method also leverages unpaired learning methods, differently, we explicitly transfer multiview IR images to avatar-like rendered images with Generative Adversarial Networks (GANs), which can generate high quality modality-transferred images while better preserving facial expression. We also leverage the additional views provided by the training HMC, and infer more precise correspondences through differentiable rendering.

2.2 Image Style Transfer

Image style transfer is the task of transforming one image to match the appearance of another while retaining its semantic and structural content [Gatys et al. 2016]. Recent works [Isola et al. 2017] have tackled this task using GANs [Goodfellow et al. 2014], which can generate images with high realism. CycleGAN [Zhu et al. 2017] introduced the concept of cycle-consistency which improves on the mode-collapse problem with GANs. Although architectural choices, such as the U-net architecture, in CycleGAN encourages structural consistency during transfer, it can still suffer from semantic drift, especially when the distributions between the domains are not balanced. In [Fu et al. 2018], a geometric loss was added to encourage the preservation of spatial structure, and in [Mueller et al. 2018], a silhouette matching term was used instead. To further encourage the preservation of semantic information during transfer, [Bansal et al. 2018] extend the idea of cycle-consistency by utilizing temporal structures of data. Instead of adding additional terms, in [Harley et al. 2019], an "uncooperative" optimization strategy is employed instead to prevent semantic drift caused by the forward and backwards transformations colluding to complement errors produced by each other. In our method, the preservation of facial expression is particularly important, because we rely on generated (fake) images as supervision for optimizing face parameters. Besides matching distribution carefully and the use of uncooperative optimization, we present cross-view cycle consistency to further enforce semantic preservation, utilizing synchronized multiview data from our designed HMC.

2.3 Differentiable rendering

In analysis-by-synthesis approaches [Tewari et al. 2017; Thies et al. 2018], differentiable rendering a textured mesh is necessary to allow the error signal to flow from pixel errors to parameters of the graphics engine in a system trained end-to-end. However, it involves a discrete rasterization step to assign triangles to every pixel, which has a non-differentiable property causing the gradients from pixel errors hard to be propagated to mesh geometry. Tewari et al. [2017] circumvent this issue by formulating the loss over the vertices instead of the image pixels. Kato et al. [2018] address the problem by approximating gradients according to the geometric relationship between a triangle and a pixel. In our application, this problem manifests as failures in matching face silhouettes, and we address it through a formulation whereby gradients of background pixels that are not covered by any rasterized triangle can still be backpropagated to affect mesh geometry.

3 ESTABLISHING CORRESPONDENCE

In recent years, end-to-end regression from images has proven effective for high-quality and real-time facial motion estimation [Laine et al. 2017; Tewari et al. 2017]. These works leverage the representation capacity of deep neural networks to model the complex mapping between raw image pixels and the parameters of a parametric face model. With input-output pairs, the problem can be posed within a supervised learning framework, where a precise mapping that generalizes well can be found (see §4). The main challenge, therefore, is to acquire a high quality training set: a sufficient

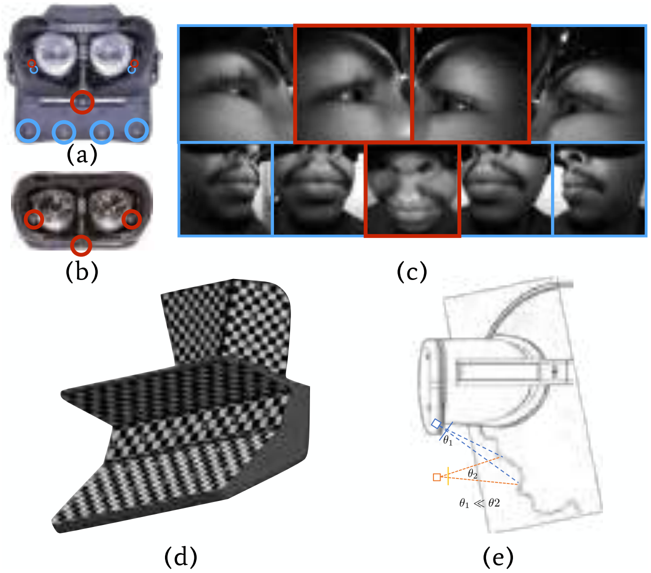

Fig. 2. **Headset mounted cameras (HMC)**; (a) Training HMC, with standard cameras (circled in red) and additional cameras (circled in blue). It is used for collecting data to help establish better correspondence between HMC images and avatar parameters. (b) Tracking HMC, used for realtime face animation, with minimal camera configuration. (c) Example of captured images, with colored frames indicating standard views (red) and additional views (blue). (d) Multi-plane calibration pattern used to geometrically calibrate all cameras in the training and tracking HMCs. (e) An illustration of the challenges of ergonomic camera placement (blue), where large motions such as mouth opening project to small changes in the captured image, in comparison to more accommodating camera placements (orange).

number of precise correspondence between input HMC images and avatar parameters spanning diverse facial expressions. This is particularly challenging for VR applications, where existing methods have significant limitations as described in §2.1.

In this work, we establish high-quality correspondence using domain-transferred multi-view analysis-by-synthesis. We describe our training- and tracking-HMC designs in §3.1. Our approach for multiview consistent correspondence estimation is then described in §3.2, with a detailed treatment of domain transfer in §3.3, and background-aware differentiable rendering in §3.4.

3.1 Data Capture

Consumer-grade VR headset designs need to be structurally and mechanically robust, ergonomic, and aesthetically pleasing. HMC designs that fit this criteria typically have partial and oblique views of the face which are challenging to use with image-space reconstruction losses typically employed in registration algorithms. Fig. 2(e) illustrates this problem. For this reason, most published [Li et al. 2015; Olszewski et al. 2016] and commercial [BinaryVR 2019] VR face tracking systems are designed to operate with more accommodating designs, at the expense of size, weight, and cost considerations. The core idea of our work is that the challenging images acquired from optimal camera mounting configurations do, in fact, contain sufficient information for precise expression estimation, as long as there are sufficient number of high-quality samples relating those images to the face model's expression space. In support of this idea, we built two versions of the same headset; one with a consumer-friendly design with a minimally intrusive camera configuration (i.e., the tracking HMC), and another with an augmented camera set with more accommodating viewpoints to support correspondence finding (i.e., the training HMC). Shown in Fig. 2, the training HMC is used to collect data and build a mapping between the minimal headset camera configuration and the user's facial expressions. Specifically, the minimal set consists of 3 IR VGA cameras for the mouth, left-eye and right-eye, respectively. The training HMC has 6 additional cameras: an additional view of each eye, and 4 additional views of the mouth, strategically placed lower to capture lip-touching and vertical mouth motion, and on either side to capture lip protrusion. All cameras are synchronized and capture at 90Hz. They were geometrically calibrated using a custom 3D printed calibration pattern that ensures that parts of the pattern are within the depth of field of each camera, as shown in Fig. 2(d).

To build the dataset, we captured each subject twice using the same stimuli; once using the training HMC, and again using the tracking HMC. The content included 73 expressions, 50 sentences, a range of motion, a range of gaze directions and 10 minutes of free conversation. This set was designed to cover the range of natural expressions. Collecting the same content in both devices ensures a roughly balanced distribution of facial expressions between the two domains, which is important for unpaired domain transfer algorithms to work well (see §3.3).

3.2 Overall Algorithm

We illustrate our overall algorithm to establish correspondences in Fig. 3. It assumes the availability of a pre-trained personalized parametric face model. For the experiments in this paper, we use the deep appearance model [Lombardi et al. 2018] which generates geometry and view-conditioned texture from an l-dimensional latent code $z \in \mathbb{R}^l$ and a 6-DOF rigid transform $v \in \mathbb{R}^6$ from the avatar's reference frame to the headset (represented by a reference camera) using a deep deconvolutional neural network $\mathbf{D}$:

$$M, T \leftarrow \mathbf{D}(z, v). \tag{1}$$

Here, $M \in \mathbb{R}^{n\times 3}$ is the facial shape comprising n-vertices, and $T \in \mathbb{R}^{w\times h}$ is the generated texture. A rendered image R can be generated from this shape and texture through rasterization:

$$R \leftarrow \mathbf{R}(M, T, A(v)), \tag{2}$$

where A denotes the camera's projection function.

Given multiview images $\mathcal{H} = \{H_i\}_{i\in C}$ acquired from a set of headset cameras C, our goal is to estimate the user's facial expression as seen in these views. We solve this by estimating the latent parameters z and headset's pose v that best aligns the rendered face model to the acquired images. However, instead of performing this task separately for each frame in a recording, similar to Tewari et al. [2017], we simultaneously estimate these attributes over the entire dataset comprising thousands of multiview frames. Specifically, we estimate the parameters θ of a predictor $\mathbf{E}_\theta$ that extracts

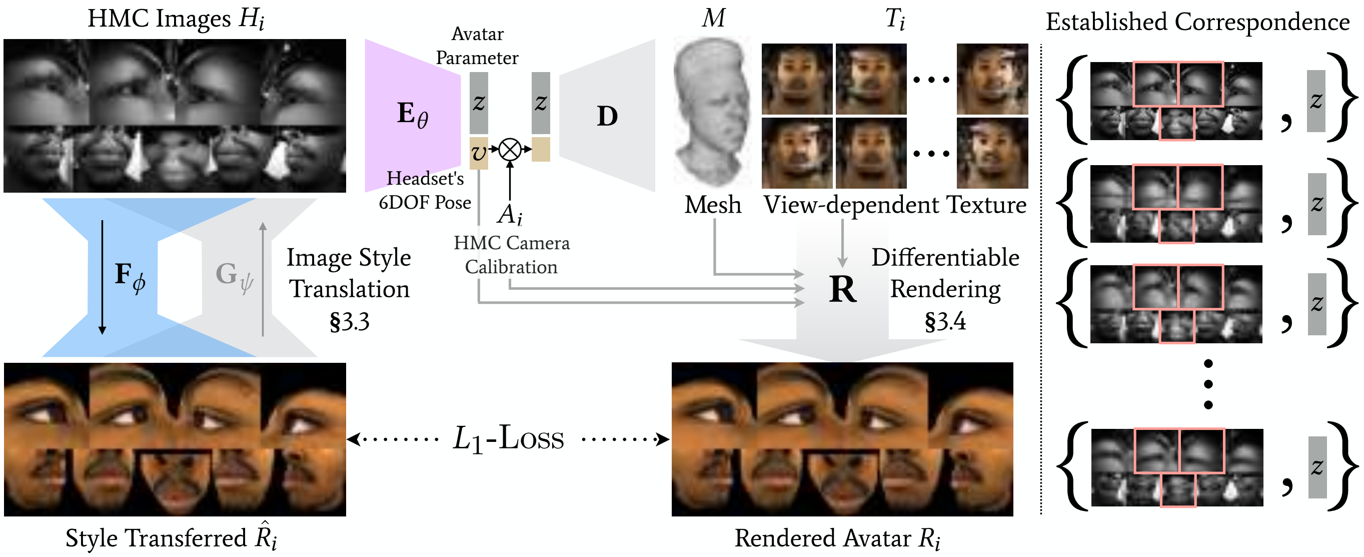

Fig. 3. **Overall optimization for establishing correspondence.** We train a network $\mathbf{E}_\theta$ to jointly estimate avatar parameters z and headset position v from 9-view images H_i from a training HMC. These estimated values are later used by $\mathbf{D}$ and $\mathbf{R}$ to render multiview images of the avatar R_i. In order to apply an L_1 reconstruction loss, we bridge the domain gap between HMC images and rendered images by also training an image style transformer $\mathbf{F}_\phi$ (along with $\mathbf{G}_\psi$) to convert H_i to the rendered domain $\hat{R}_i$, with preserved semantics. Colored modules ($\mathbf{E}_\theta$ and $\mathbf{F}_\phi$) have trainable parameters, while parameters of $\mathbf{D}$ are frozen and $\mathbf{R}$ is parameter free. On the right, the found correspondences between H and z can be used as pairs of 3-view images for a tracking HMC (shown in pink-boxes) and avatar parameters for later regression (see §4).

$\{z^t, v^t\}$, the latent code and headset's pose for frame $t \in \mathcal{T}$, by jointly considering data from all cameras at that time instant:

$$z^t, v^t \leftarrow \mathbf{E}_\theta(\mathcal{H}^t), \ \forall t \in \mathcal{T}. \tag{3}$$

Note that the same predictor is used for all frames in the dataset. Analogous to non-rigid structure from motion [Xiao et al. 2006], this has the benefit that regularity in facial expression across time can further constrain the optimization process, making the estimation proces more resistant to terminating in poor local minima.

Due to the domain-gap between the rendered image R and the camera images H, they are not directly comparable. To address this, we also learn the parameters ϕ of a view-dependent domain transfer network:

$$\hat{R}_i = \mathbf{F}_\phi(H_i; i), \ \forall i \in C. \tag{4}$$

In its simplest form, this function is comprised of independent networks for each camera i. With this, we can formulate the analysis-by-synthesis reconstruction loss:

$$\mathbf{L}(\theta, \phi) = \sum_{t \in \mathcal{T}} \left(\sum_{i \in C} \left\| \hat{R}_i^t - R_i^t \right\|_1 + \lambda \delta(z^t) \right), \tag{5}$$

where R_i^t is the rendered face model from Eq. (2), rasterized using the known projection function A_i whose parameters are obtained from the calibrated camera i, as mentioned in § 3.1. Here, δ is a regularization term over the latent codes z, and λ weights its contribution against the L_1-norm reconstruction of the domain-transferred image.

Although at first glance this formulation appears to be reasonable, with high capacity encoder $\mathbf{E}_\theta$ and domain-transfer $\mathbf{F}_\phi$ functions, there is a space of solutions where one network can compensate for the semantic error incurred by the other, leading to low reconstruction errors but incorrect estimation of expression z and headset pose v. Without additional constraints, we observe that this phenomenon often occurs in practice, which we refer to as collaborative self-supervision. When the domain gap comprises primarily of appearance differences, we conjecture that collaborative self-supervision tends to be more prominent in architectures that do not retain spatial structure. This is the case in our system, where the latent code z is a vectorized encoding of the image. This problem has been observed in the style-transfer community, where a common solution is to use fully convolutional architectures with skip connections [Isola et al. 2017; Zhu et al. 2017]. As spatial structure is propagated through the entire network, it is easier to retain structure, and thus, it tends to remain unchanged when differences in style can be well explained by appearance alone.

Motivated by the compelling results achieved by methods such as [Zhu et al. 2017], we decouple Eq. (5) into two stages. First, we learn the domain transfer $\mathbf{F}_\phi$ that converts headset images H_i into *fake* rendered images R_i without changing the apparent expression (i.e., semantics). In the second stage, we fix $\mathbf{F}_\phi$ and optimize Eq. (5) with respect to $\mathbf{E}_\theta$.

3.3 Expression-Preserving Domain Transfer

Our expression-preserving transfer is based on unpaired image translation networks [Zhu et al. 2017]. This architecture learns a bidirectional domain mapping ($\mathbf{F}_\phi$ and $\mathbf{G}_\psi$) by enforcing cyclic consistency between the domains and an adversarial loss for each of the

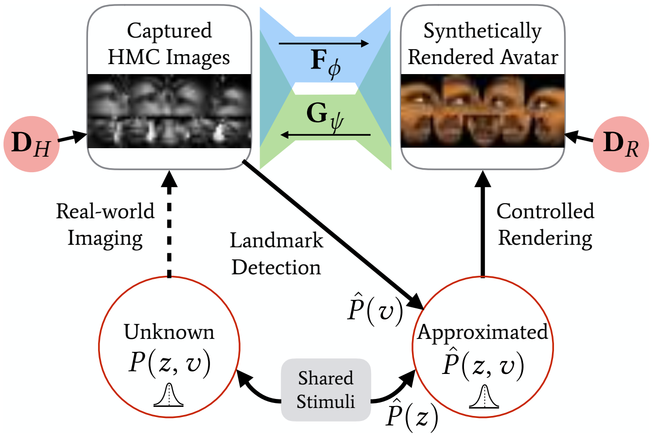

Fig. 4. **Matching the distribution of spatial structures** in images across domains is important for training expression-preserving image style transfer networks $\mathbf{F}_\phi$ and $\mathbf{G}_\psi$. We use landmark detection to estimate a distribution over headset pose, and use the same expression stimuli during data capture (for both avatar building and HMC capture) to ensure the distribution of expressions is comparable.

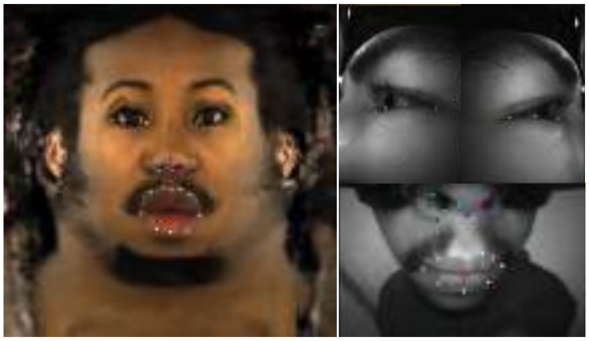

Fig. 5. **Landmarks on the texture map of the avatar (left) and HMC images (right).** Colors of landmarks indicate the point-to-point correspondence across domains. We detect landmarks on all 9 views (only 3 views are shown) from detectors trained from manual annotations. The uv-coordinates of these landmarks are then jointly solved with z^t and v^t across multiple frames. Note that we do not minimize projected distance of landmarks in Eq. (5) to find image-to-avatar correspondence.

two. To achieve preservation of expression, we need to eliminate the tendency of generators to modify spatial structure on the images. With a fully convolutional architecture where random initialization already leads to retained image structures, this tendency mainly comes from the pressure to prevent opposing discriminators from spotting fake images from their spatial structure. In other words, if the distribution of spatial structure, which in our case is jointly determined by headset positions v and facial expressions z, is balanced, the generators then have no pressure to begin modifying them. In the following, we begin by describing how we balance the distribution when preparing real datasets before training, followed by the learning problem itself.

3.3.1 Matching Distribution of Image Spatial Structure. While we don't have control over the underlying expression z^t and headset position v^t in captured headset data $\{H_i^t\}_{t\in\mathcal{T}}$, we can generate a set of rendered images $\{R_i^s\}_{s\in\mathcal{S}}$ with the desired statistics if we have an estimate $\hat{P}(z, v)$ of the joint distribution $P(z, v)$. However, since estimating individual z^t and v^t for headset data is our original problem, we need to find a proxy to approximate $\hat{P} \approx P$.

Fig. 4 summarizes our strategy. Here, we assume independent distribution between z and v, or $\hat{P}(z, v) = \hat{P}(z)\hat{P}(v)$, and estimate $\hat{P}(z)$ and $\hat{P}(v)$ individually. For $\hat{P}(z)$, we rely on the data capture process, where the subject is captured twice with the same stimuli as mentioned in § 3.1. Even though this does not lead to a frame-to-frame mapping between the captures, we can assume the distribution of facial expression is comparable. Therefore, we can use the set of expression codes from the modeling capture as approximate samples from $P(z)$.

For the distribution over headset pose $\hat{P}(v)$, we resort to fitting the face model's 3D geometry to detected landmarks on headset images by collecting 2D landmark annotations and training landmark detectors [Wei et al. 2016]. Although landmark fitting alone does not produce precise enough estimates of expression (because landmarks are not able to describe complete and subtle facial expressions), it can often give reasonable estimates for v owing to its low dimensionality and limited range of variation. One of the challenges in fitting a 3D mesh to 2D detections is defining correspondence between mesh vertices and detected landmarks. Typically, the landmark set for which annotations are available does not match exactly to any vertex in a particular mesh topology. Manually assigning a single vertex to each landmark can lead to suboptimal fitting results for coarse mesh topologies. To address this problem, while fitting individual meshes, we simultaneously solve for each landmark's mesh correspondence (used across all frames) in the texture's uv-space $\{u_j \in \mathbb{R}^2\}_{j=1}^m$, where m is the number of available landmarks. To project each landmark j on rendered images of every view, we calculate a row vector of the barycentric-coordinates $\mathbf{b}_j \in \mathbb{R}^{1\times 3}$ of the current u_j in its enclosing triangle, with vertices indexed by $\mathbf{a}_j \in \mathbb{N}^3$, and then linearly interpolate projections of the enclosing triangle's 3D vertices, $M_{\mathbf{a}_j} \in \mathbb{R}^{3\times 3}$, where M is the mesh from Eq. (1). Specifically, we solve the following optimization problem:

$$\min_{u_j, v^t, z^t} \sum_{t\in\mathcal{T}} \sum_{i\in C} \sum_{j=1}^{m} w_{ij}^t \left\| \mathbf{p}_{ij}\left(H_i^t\right) - \mathbf{b}_j \mathbf{P}(M_{\mathbf{a}_j}^t, A_i v^t) \right\|^2, \qquad (6)$$

where $\mathbf{p}_{ij} \in \mathbb{R}^2$ is the 2D detection of landmark j in HMC camera i, $\mathbf{P}$ is a camera projection generating 2D points in $\mathbb{R}^{3\times 2}$, and w_{ij}^t is the landmark's detection confidence in $[0, 1]$. Note that for a landmark j not observable by view i, w_{ij}^t is zero. In practice, we initialize the u_j's to a predefined set of vertices in the template mesh to prevent divergence. We also avoid using landmarks in regions where the avatar doesn't have mesh vertices, such as the pupils and the mouth interior. Fig. 5 shows an example of the u_j's at convergence.

By solving Eq. (6), we obtain a set of HMC pose $\{v^t\}_{t\in\mathcal{T}}$ from each frame $\mathcal{H}^t$. We can now render the required dataset $\{R_i^s\}_{s\in\mathcal{S}}$ by randomly sampling a HMC pose $|\mathcal{S}|$ times from $\{v^t\}_{t\in\mathcal{T}}$, and an expression code also $|\mathcal{S}|$ times from the set of encoded values

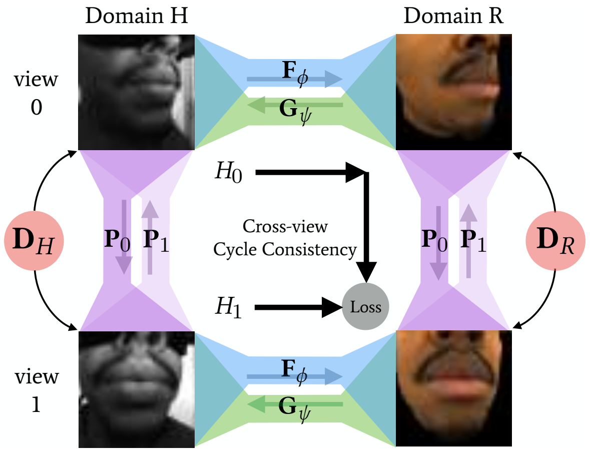

Fig. 6. **Cross-view cycle consistency in multiview image style translation.** In additional to original cycle consistency within each domain, we further constrain style transformers given paired multiview data in each domain, to encourage preserving spatial structure of the face images. Only one loss term (out of all 4 directions) is shown here.

of the face modeling capture [Lombardi et al. 2018], independently. Together with $\{H_i^t\}$, we now have training data for unpaired style transfer. We discard the estimated z^t solved by Eq. (6) since they tend to be inaccurate when relying solely on landmarks.

3.3.2 Multiview Image Style Translation. Given images from the two domains $\{H_i^t\}$ and $\{R_i^s\}$, one can utilize a method such as [Zhu et al. 2017] to learn the view-specific mappings $\mathbf{F}_{\phi,i}$ and $\mathbf{G}_{\psi,i}$ that translate images back and forth between the domains. Since we are careful to encourage a balanced distribution $P(z, v)$ between the domains, this straight-forward approach already produces reasonable results. The main failure cases occur due to limited rendering fidelity of the parametric face model. This is most noticeable in the eye images, where eyelashes and glints are almost completely absent due to poor generalization of the view-conditioned rendering method of [Lombardi et al. 2018] to very close viewpoints. Specifically, the style transferred image often exhibits a modified gaze direction compared to the source. In addition, this modification is often inconsistent across different camera views. This last effect is also observed for the rest of the face, though not to the same extent as that for the eyes. When solving for (z, v) using constraints from all camera views in Eq. (5), these inconsistent and independent errors have an averaging effect, which manifests as dampened facial expressions.

To overcome this problem, we propose a method that exploits the spatial relationship between cameras during image style translation. Inspired by [Bansal et al. 2018], where additional cyclic-consistency constraints are obtained through temporal prediction, we enforce cyclic-consistency through cross-view prediction. Specifically, for a pairs of views, denoted 0 and 1 for simplicity, we train "spatial-predictors" $\mathbf{P}_0$ and $\mathbf{P}_1$ to transform images in view 0 to view 1 and vice versa. These pairs are chosen in such a way that they observe similar parts of the face to ensure that their contents are mutually predictable, for example the stereo eye-camera pair, or lower-face cameras on the same side. Together with the standard terms of CycleGAN [Zhu et al. 2017], our loss function takes the form:

$$\mathbf{L} = \mathbf{L}_C + \lambda_G \mathbf{L}_G + \lambda_P \mathbf{L}_P + \lambda_V \mathbf{L}_V, \tag{7}$$

where $\mathbf{L}_C = \mathbf{L}_{CH} + \mathbf{L}_{CR}$ is the cycle-consistency loss for each domain and for each view,

$$\mathbf{L}_{CH} = \sum_{i \in \{0,1\}} \sum_t \left\| \mathbf{G}_\psi \circ \mathbf{F}_\phi \left(H_i^t\right) - H_i^t \right\|_1, \tag{8}$$

$\mathbf{L}_G = \mathbf{L}_{GH} + \mathbf{L}_{GR}$ is the GAN-loss (for both the generator and discriminator) for each domain and for each view,

$$\mathbf{L}_{GH} = \sum_{i \in \{0,1\}} \sum_t \left[\log \left(\mathbf{D}_H \left(H_i^t\right)\right) + \log \left(1 - \mathbf{D}_R \circ \mathbf{F}_\phi \left(H_i^t\right)\right) \right], \tag{9}$$

$\mathbf{L}_P$ is the loss for the view predictor,

$$\mathbf{L}_P = \sum_{i \in \{0,1\}} \left(\sum_t \left\| \mathbf{P}_i(H_i^t) - H_{1-i}^t \right\|_1 + \sum_s \left\| \mathbf{P}_i(R_i^s) - R_{1-i}^s \right\|_1 \right), \tag{10}$$

and finally the cross-view cycle consistency $\mathbf{L}_V = \mathbf{L}_{VH} + \mathbf{L}_{VR}$,

$$\mathbf{L}_{VH} = \sum_{i \in \{0,1\}} \sum_t \left\| \mathbf{P}_i \circ \mathbf{F}_\phi \left(H_i^t\right) - \mathbf{F}_\phi \left(H_{1-i}^t\right) \right\|_1 \tag{11}$$

where $\mathbf{L}_{CR}$, $\mathbf{L}_{GR}$, and $\mathbf{L}_{VR}$ are defined symmetrically, and $\mathbf{D}_H$ and $\mathbf{D}_R$ are discriminators in both domains, respectively. Note that while we do not have paired H_i^t and R_i^s, we do have paired H_i^t and H_{1-i}^t, as does R_i^s. An illustration of these components is shown in Fig. 6. Different than [Bansal et al. 2018], in our formulation, we share $\mathbf{P}_0$ and $\mathbf{P}_1$ across domains, since the relative structural difference between the views should be the same in both domains.

Our problem takes the form of a minimax optimization problem:

$$\min_{\mathbf{P}_0, \mathbf{P}_1, \mathbf{F}_\phi, \mathbf{G}_\psi} \max_{\mathbf{D}_H, \mathbf{D}_R} \mathbf{L} \left(\mathbf{P}_0, \mathbf{P}_1, \mathbf{F}_\phi, \mathbf{G}_\psi, \mathbf{D}_H, \mathbf{D}_R \right), \tag{12}$$

which we repeat for every pair of views. Compared to standard CycleGAN, our problem involves additional $\mathbf{P}_i$ modules. If we simply alternately train parameters in $\{\mathbf{P}_0, \mathbf{P}_1, \mathbf{F}_\phi, \mathbf{G}_\psi\}$ and parameters in $\{\mathbf{D}_H, \mathbf{D}_R\}$ like [Bansal et al. 2018], we find that collusion between $\mathbf{P}$ and $\mathbf{F}_\phi$ (or $\mathbf{G}_\psi$) can minimize the loss function without preserving expression across the domains; effectively learning different behaviors on real and face data to compensate errors made by each other. As a result, the semantics that we want to keep unchanged get lost during the style transformation. To address this problem, we apply "uncooperative training", recently proposed by Harley et al. [2019], which prevents this "cheating" by breaking the optimization into more steps. At each step, the loss function is readjusted so that only terms that operate on real data remain, and only modules that take real data as input are updated. An outline of the algorithm is presented in Algorithm 1. In this way, modules have no chance to learn to compensate for errors made by previous modules. As a result, expressions are better preserved through domain transfer and the cross-view predictions ensure multiview consistency.

ALGORITHM 1: Uncooperative training for multiview image style translation

```
Input: Unpaired real HMC images $\{H_i^t\}$ and real rendered images
       $\{R_i^s\}$, in two common views $i \in \{0, 1\}$.
Output: Converged $\mathbf{F}_\phi$ and $\mathbf{G}_\psi$
Initialize parameters in all modules;
repeat
    Sample $(t, s)$ to get $H_i^t$, $R_i^s$ for $i \in \{0, 1\}$ (total 4 images) ;
    Update $\phi$ for $\mathbf{F}$ using gradients minimizing $\mathbf{L}_{CH} + \mathbf{L}_{GH} + \mathbf{L}_{VH}$ ;
    Update $\psi$ for $\mathbf{G}$ using gradients minimizing $\mathbf{L}_{CR} + \mathbf{L}_{GR} + \mathbf{L}_{VR}$ ;
    Update $\mathbf{P}$ using gradients minimizing $\mathbf{L}_P$ ;
    Update $\mathbf{D}_H$ and $\mathbf{D}_R$ using gradients maximizing $\mathbf{L}_G$
until converged;
```

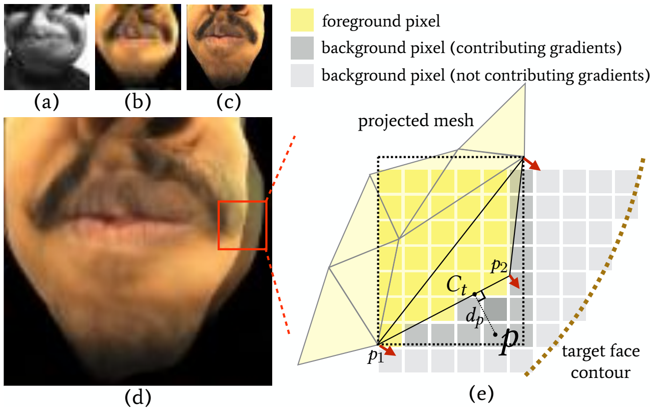

Fig. 7. **Background-aware differentiable rendering. (a)** Input HMC image with a puffed cheek **(b)** Style transferred result (target image) **(c)** Current rendered avatar **(d)** Overlaying (b) and (c). Many pixels in the red box are currently rendered as background pixels, but should be in foreground. **(e)** A closer look at pixels around the face contour. For a background pixel p within a bounding box of any projected triangle (dashed rectangle), its color is blended from color C_t at the closest point on the closest edge $\overline{p_1p_2}$ and background color C_b, with a weighting related to distance d_p. The red arrows indicate that the gradient generated from dark gray pixels can be back-propagated to geometry of the face.

3.4 Background-aware Differentiable Rendering

A core component of our system is a differentiable renderer $\mathbf{R}$ for the parametric face model. It is used to generate synthetic samples for the domain transfer described above, but also for evaluating the reconstruction accuracy given the estimated expression and pose parameters $q = (z, v)$ in Eq. (5). Our rendering function blends a rasterization of the face model's shape and background, so that the pixel color $C(p)$ at image position p is defined as:

$$C(p) = W(p)\, C_t(p) + (1 - W(p))\, C_b \qquad (13)$$

where $C_t(p)$ is the rasterized color from texture at position p, and C_b is a constant background color. If we simply define W as a binary mask of the rasterization's pixel coverage, where $W(p) = 1$ if p is within a triangle and otherwise $W(p) = 0$, then $\frac{dW(p)}{dq}$ would be zero for all p because of the discreteness of rasterization. In this case, for a foreground pixel (i.e., $W(p) = 1$) the gradient of $C(p)$ still can be calculated from $\frac{dC_t(p)}{dq}$, by parameterizing the coordinates in the texture (from which the pixel color is sampled) by the barycentric coordinates of that pixel in its currently enclosing triangle. Although this way of formulating the rendering function and its derivative can already produce good results, in practice, in the presence of multiview constraints, it exhibits failure cases that result from zero gradients from $W(p)$. Specifically, if p is rendered as background (i.e., $W(p) = 0$) but the target for that pixel is a foreground value, there are no gradients propagated to the parameters q. Similarly, a foreground pixel at the boundary of the rasterization has no pressure to expand. In practice, this can lead to terminating in poor local minima with large reconstruction errors. For example, in a puffed-cheek expression, where the target image's foreground image tends to occupy larger area of the image, the estimated expression often fails to match the contour of the cheek well (see Fig. 7).

Intuitively, the force expanding the foreground area should come from a soft blending around the boundary between foreground and background. Therefore, instead of a binary assignment of pixels around the boundary to either a color sampled from the texture map or the background color, we employ soft blending similar to anti-aliasing. It is important to parameterize the blending weight as a function of the face model's projected geometry so that reconstruction errors along the rasterization's boundary can be back-propagated to the parameters q. To this end, we use a decaying blend-function away from the boundary:

$$W(p) = \exp\left\{-\frac{d_p^2}{\sigma^2}\right\}, \qquad (14)$$

where d_p is the perpendicular 2D distance from a background pixel p to its closest edge of any projected triangle for pixels outside the rasterization coverage, and σ controls the rate of decay. The value of C_t used in Eq. (13) for $W(p)$ is set to the color in the texture of the triangle at the closest edge. For pixels within the coverage, $d_p = 0$. In practice, we set $\sigma = 1$ and evaluate W only for pixels within enclosing rectangles of each projected triangle for efficiency. With this background-aware rendering, even though only a small portion of background pixels contribute gradients to expand or contract the boundary at each iteration, it is enough to prevent optimization from terminating in poor local minima.

3.5 Implementation Details

For the style transformation described in §3.3, we use (256 × 256)-sized images for both domains, and adapt the architecture design from [Zhu et al. 2017]. For $\mathbf{F}_\phi$, $\mathbf{G}_\psi$, and $\mathbf{P}_i$, we use a ResNet with 4× downsampling followed by 3 ResNet modules and another 4× upsampling. For discriminators $\mathbf{D}_H$ and $\mathbf{D}_R$, we apply spectral normalization [Miyato et al. 2018] for better quality of generated images and more stable training. For $\mathbf{E}_\theta$ in the overall training described in 3.2, we build separate convolutional networks to convert individual H_i^t to $|C|$ vectors, which are concatenated and then converted into both z^t and v^t using separate MLPs. For the prior $\delta(z^t)$ in Eq.(5), we use an L_2-loss $\delta(z^t) = \left\|z^t\right\|_2^2$ because the latent space associated with $\mathbf{D}$ is learned with a KL divergence against a standard normal distribution [Lombardi et al. 2018].

4 REALTIME FACE ANIMATION

After minimizing the loss in Eq. (5), we can apply a converged $\mathbf{E}_\theta$ to all $\mathcal{H}^t$ to obtain per-frame correspondences $\{(\mathcal{H}^t, z^t)\}_{t\in\mathcal{T}}$. We can now drop all the auxiliary views in $\mathcal{H}^t$ and only retain the views available in a tracking HMC $\tilde{\mathcal{H}}^t = \{H_i^t\}_{i\in C'}$ where $|C'| = 3$. This forms the training data $\{(\tilde{\mathcal{H}}^t, z^t)\}_{t\in\mathcal{T}}$ for training the regressor that will be used during realtime animation.

Rather than simply minimizing L_2-loss in the latent space of z^t, we find it is important to measure loss in a way that encourages the network to spend capacity on the most visually sensitive parts, such as subtle lip shape and gaze direction. We additionally minimize the error in geometry and texture map particularly in eye and mouth regions, because the avatar does not have detailed geometry and relies on view-dependent texture to be photorealistic in these regions. Specifically, we build regressor $\tilde{\mathbf{E}}_{\tilde\theta}$ to convert $\tilde{\mathcal{H}}^t$ to target z^t:

$$\min_{\tilde\theta} \sum_t \left\|z^t - \tilde z^t\right\|^2 + \lambda_1 \left\|M^t - \tilde M^t\right\|^2 + \lambda_2 \left\|\kappa(T_0^t) - \kappa(\tilde T_0^t)\right\|^2, \quad (15)$$

where

$$\tilde z^t = \tilde{\mathbf{E}}_{\tilde\theta}(\tilde{\mathcal{H}}^t) \quad (16)$$

$$\tilde M^t, \tilde T_0^t \leftarrow \mathbf{D}(\tilde z^t, v_0) \quad (17)$$

$$M^t, T_0^t \leftarrow \mathbf{D}(z^t, v_0), \quad (18)$$

where κ is a crop on texture maps focusing on eye and mouth area, and v_0 is a fixed frontal view of the avatar.

The architectural design of $\tilde{\mathbf{E}}_{\tilde\theta}$ should allow good fitting to the target z^t, be robust against real-world variations such as surrounding illumination and headset wearing positions, and at the same time, achieve realtime inference speed. These requirements are different from $\mathbf{E}_\theta$, whose function is solely to minimize Eq. (5), or to overfit. Therefore, we use smaller input images (192 × 192) and a smaller number of convolutional filters and layers for $\tilde{\mathbf{E}}_{\tilde\theta}$, compared to $\mathbf{E}_\theta$. The architectural design is similar: we build 3 separated branches of convolutional networks to convert $\tilde H_i^t$ to 3 1D vectors, because the input images are observing different parts of the face and hence don't share spatial structure. Finally, these 1D vectors are concatenated and converted to $\tilde z^t$ through an MLP. During training, we augment input images with a random small angle homography to simulate camera rotation to account for manufacturing variance in camera mounts, as well as directional image intensity histogram perturbation to account for lighting variation. Our architecture design balances speed with almost no obvious quality drop (from 9-view input to 3-view input) on both training data and validation data, at an inference speed of 73 fps on a NVIDIA Titan 1080Ti.

Given that $\tilde{\mathbf{E}}$ and $\mathbf{D}$ can both be evaluated in realtime [Lombardi et al. 2018], we can build a two-way social VR system, where both users can see high-fidelity animation of each other's personalized avatar while wearing an HMC. On each side, the computing node runs $\tilde{\mathbf{E}}$ of the user on one GPU and sends encoded z^t over the network. At the same time, the node receives z^t from the other side, runs the decoder $\mathbf{D}$ of the other person, and renders stereo images (for left and right eye) of the other user's avatar on a second GPU.

5 RESULTS AND EXPERIMENTS

We first show qualitative results of both the found correspondences using the training HMC (§ 3), and realtime prediction using the tracking HMC (§4). Then we characterize the importance of various components in our system design, including multiview style transfer, distribution matching, cross-view consistency, and background-aware differentiable rendering, in an ablation study. Finally, we compare our animation results with [Olszewski et al. 2016] and [Lombardi et al. 2018]. For more results, such as speech and dynamics which are better shown as consecutive frames with audio, please refer to our supplementary video. Our method achieves natural speech dynamics with only frame-by-frame inference without temporal constraints.

5.1 Qualitative Results

5.1.1 Established Correspondence. Fig. 8 shows three examples of corresponding HMC images and rendered avatars for three different subjects. In the last row of each example, we show the alignment between HMC images and the rendered avatar for every view of the training HMC, where good alignment demonstrates the high-fidelity of the obtained correspondence. We present more results in Fig. 9, showing that our method robustly produces high-fidelity results for a large range of facial expressions, including extreme expressions with occluded face parts, such as biting lips, puffed cheeks, wrinkles on the forehead and nose, and tightly closed eyes. Note that the inside of the eyes and mouth are not modeled with detailed geometry [Lombardi et al. 2018], so the mapping for nuanced expressions, like gaze directions and visibility of teeth and shape of tongue, rely completely on minimizing the pixel loss of the rendered avatar against style transferred images.

5.1.2 Realtime Animation. Fig. 10 shows example outputs from a trained regressor $\tilde{\mathbf{E}}$ that runs in realtime. We captured each subject with a training HMC in 4 different environments, with different lighting and different HMC placement relative to the head (the subjects had to take off and put on the HMC between captures). We use our method described in §3 to generate "ground-truth" correspondences for all data, but only train $\tilde{\mathbf{E}}$ on 3 of the captures and use the remaining capture as the testing set. For most of the facial expressions, the regressor using only the 3 tracking views perform very well and matches the established correspondence using the 9 training views, indicating that the 3 tracking views contain sufficient information despite obliqueness. For a few cases highlighted in the red boxes, the tracking views have poor coverage of the mouth interior (illustrated in Fig. 2(e)) and gaze direction because the eyelids are almost closed (the additional eye cameras in the training HMC are lower), resulting in slightly different results. While the use of region-focused texture and geometry terms in Eq. (15) already helps concentrate the capacity of the network on perceptually important parts, the error for extreme gaze directions is still observable.

5.2 Ablation Study

We first validate the importance of having the additional camera views of the training HMC, style transfer, and distribution matching for finding correspondences. We then compare independent per-view style transfer with our cross-view cycle consistency. We also

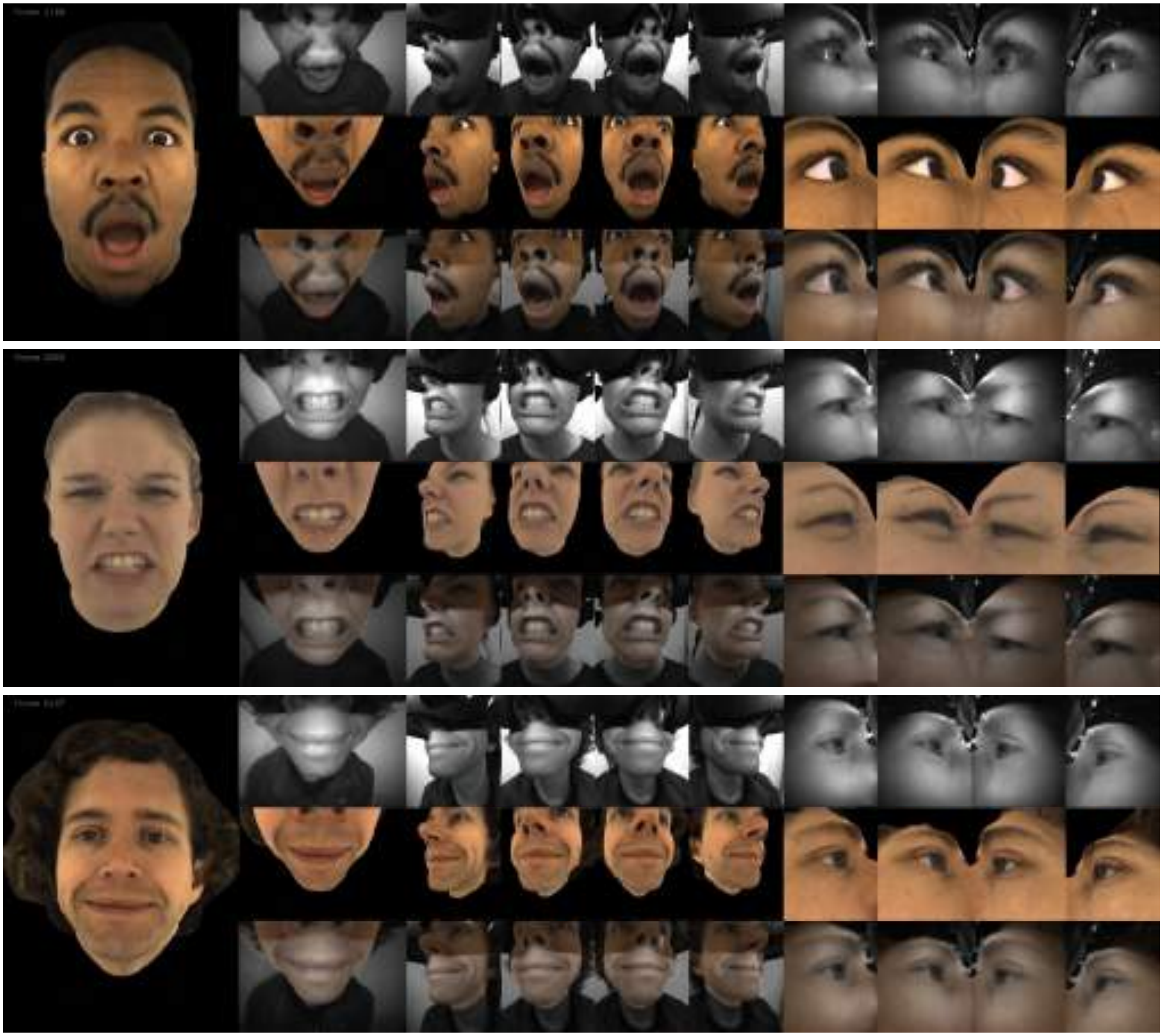

Fig. 8. **Qualitative results of established correspondence using the training HMC.** For each subject, the first row shows input images from the training HMC and detected landmarks. The second row shows the avatar rendered with found corresponding facial expressions and headset position (coupled with camera calibration at each view), with projected landmarks whose uv-coordinates are optimized by Eq. (6). The leftmost image is rendered from a fixed frontal view. The last row overlays the previous two, where good alignment shows high fidelity correspondence.

show evidence that background-aware rendering leads to better convergence. Finally, we discuss the effects of using different loss functions in the optimization Eq. (5) and Eq. (15).

5.2.1 Effect of in-domain pixel matching, additional viewpoints, and distribution matching. To show the importance of using multiview style transfer in Eq. (5), we compare our results with **(a) only using domain-invariant features**: minimizing 2D distance between detected landmarks and projected landmarks of the avatar on all 9 views, plus matching image gradients (edges) **(b) only using 3 tracking views**: training independent per-view style transfer and solving Eq. 5 with only the tracking views, and **(c) not matching the distribution of HMC pose**: simply using a mean pose to render $\{R_i^s\}$ for style transfer, instead of solving Eq. (6) to collect a set of HMC poses to sample from.

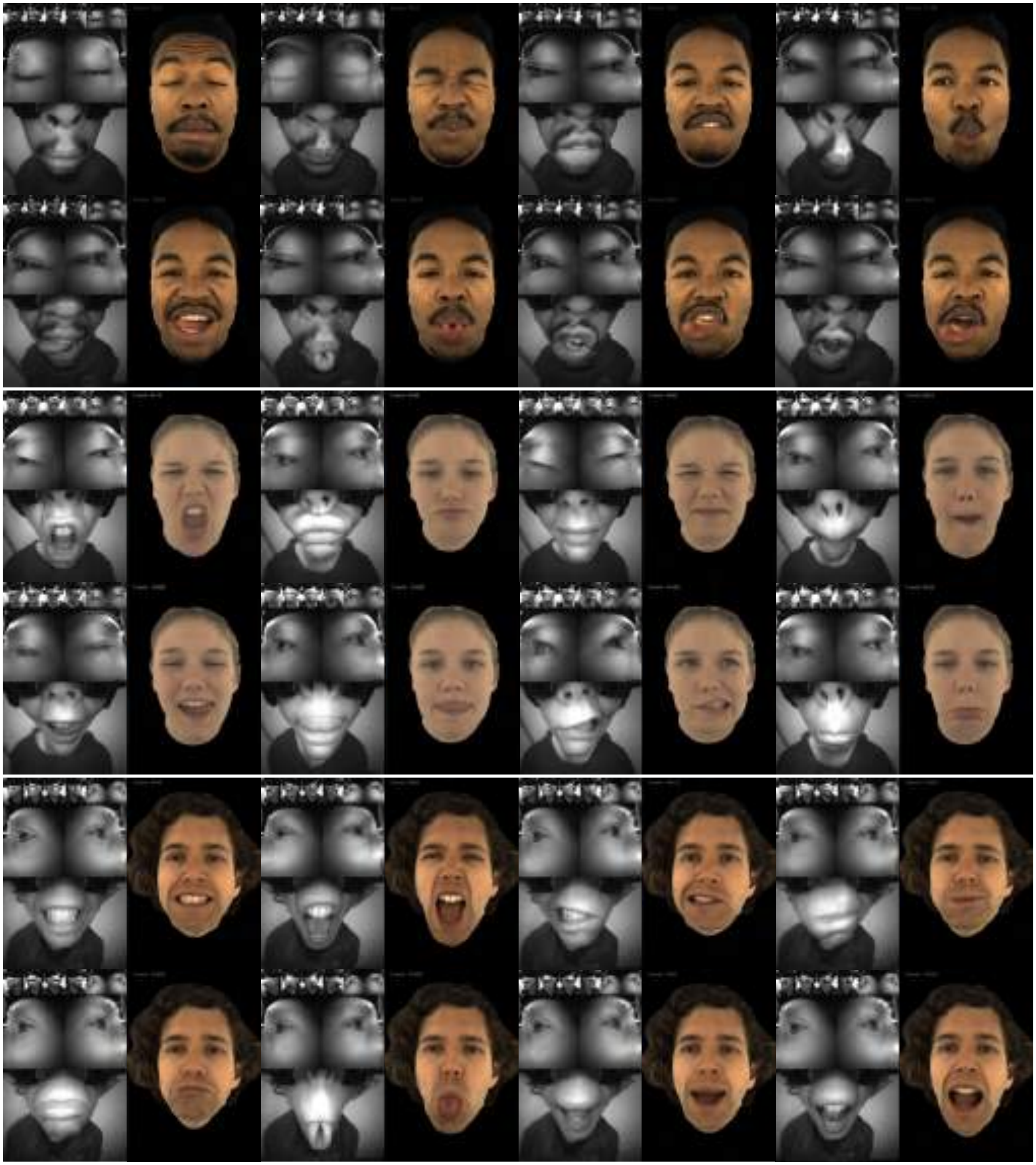

Fig. 9. **More examples of established correspondence**. Each subfigure shows 8 facial expressions for each identity. The 9 grayscale images, with 3 tracking views (large images) and 6 additional training views (small images), are input HMC images, and in color on the right is the rendered 3D avatar. Our method can find high quality mappings even for subtle expressions on the upper face, where the camera angle is oblique and close to the subject. Our method can also capture subtle differences in tongues, teeth, and eyes where the avatar does not have detailed geometry.

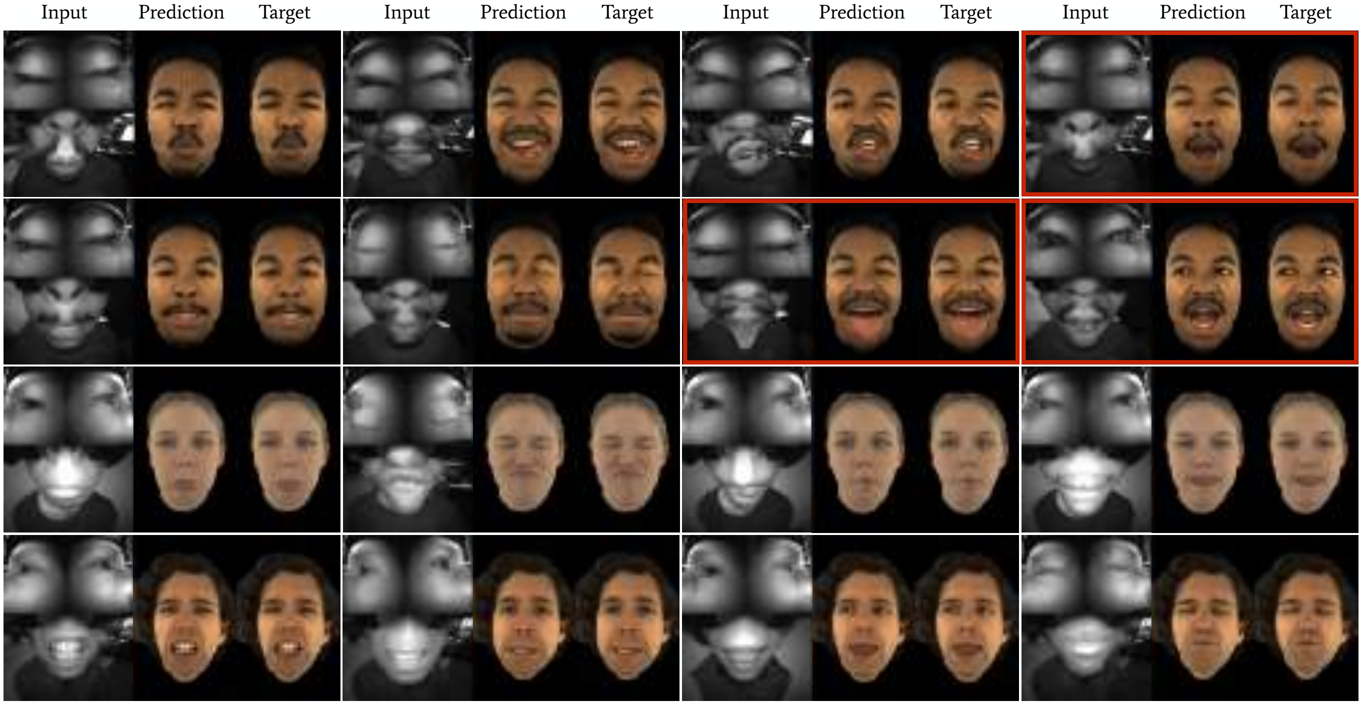

Fig. 10. **Predictions on held out data for a tracking HMC**. In each subfigure, the target image is "ground-truth" from our established correspondences, and the prediction image is the output from $\hat{\mathbf{E}}$. While most of the time we can map from only 3-view inputs to targets successfully, there are some failure cases highlighted in the red boxes, where the nuances are hard to observe without the additional cameras on the training HMC.

In Fig. 11, we show 8 expressions to compare. Results that only use landmark and edge constraints can roughly match the shape of mouth, but completely fail to match parts such as teeth, tongue, and gaze direction, as landmarks cannot describe these details on a photorealistic avatar. Matching image gradients does not improve the results because the domain gap between HMC images and rendered avatars is too large for naive edge matching to be useful. Using only 3 tracking views with view-independent style transfer and Eq. (5) already generates much better results than 9-view results in (a), showing the importance of bridging the domain gap. But because of the obliqueness of viewpoints, it sometimes fails to generate precise results when compared to using the full 9 views, such as estimating the amount of mouth openness (1st, 5th expressions), mouth interior (6th), and lip shape (7th). Finally, the results from (c) are in general worse than the full method, showing the importance of matching distribution of spatial structure. The quality of style transferred results degrades, leading to uncanny faces (1st, 4th, 8th) and semantics are sometimes modified such as gaze (3rd).

5.2.2 Effect of view-consistent image style transfer. When training style transformers $\mathbf{F}_\phi$ and $\mathbf{G}_\psi$ for all 9 views, we found that training them independently for each view already gives good results despite artifacts on the fake rendered image such as checkerboard patterns. By jointly considering multiview and a robust L_1 loss in Eq.(5), these uncorrelated artifacts are often averaged out. However, other artifacts such as wrong color of mouth interior or changes in gaze direction, do impact the final result, as illustrated in Fig. 12. To show the effect of the cross-view training described in §3.3, we group 9 camera views into 4 pairs and run Algorithm 1 for each pair (and still train the remaining 1 view individually). The resulting style transferred images have fewer artifacts on the face in general. We highlight the cases where the per-view CycleGAN baseline fails to capture visibility of the tongue and a consistent gaze direction. Using the cross-view consistency loss in our method, in contrast, gives consistent images across views and therefore the final optimized avatar better matches the inputs.

5.2.3 Effect of background-aware differentiable rendering. We compare our background-aware differentiable rendering with a baseline where background pixels are simply assigned a constant color and therefore have no path to back-propagate gradients from errors in pixel color to mesh geometry. In particular, we show the image loss curve during convergence in Fig. 13 to a target puffed cheek. Even though there are constraints from 9 views, which already greatly alleviates the background problem, the baseline still plateaus at a higher loss, and the cheek does not fully expand to match the inputs. In contrast, our method allows gradients from the image loss to affect the geometry through Eq. 13 and therefore shows lower loss and better face alignment.

5.2.4 Is image loss alone enough to make Eq. (5) *converge?* While style transferred images provide supervision with great details, it is a common observation that the convergence basin when using a

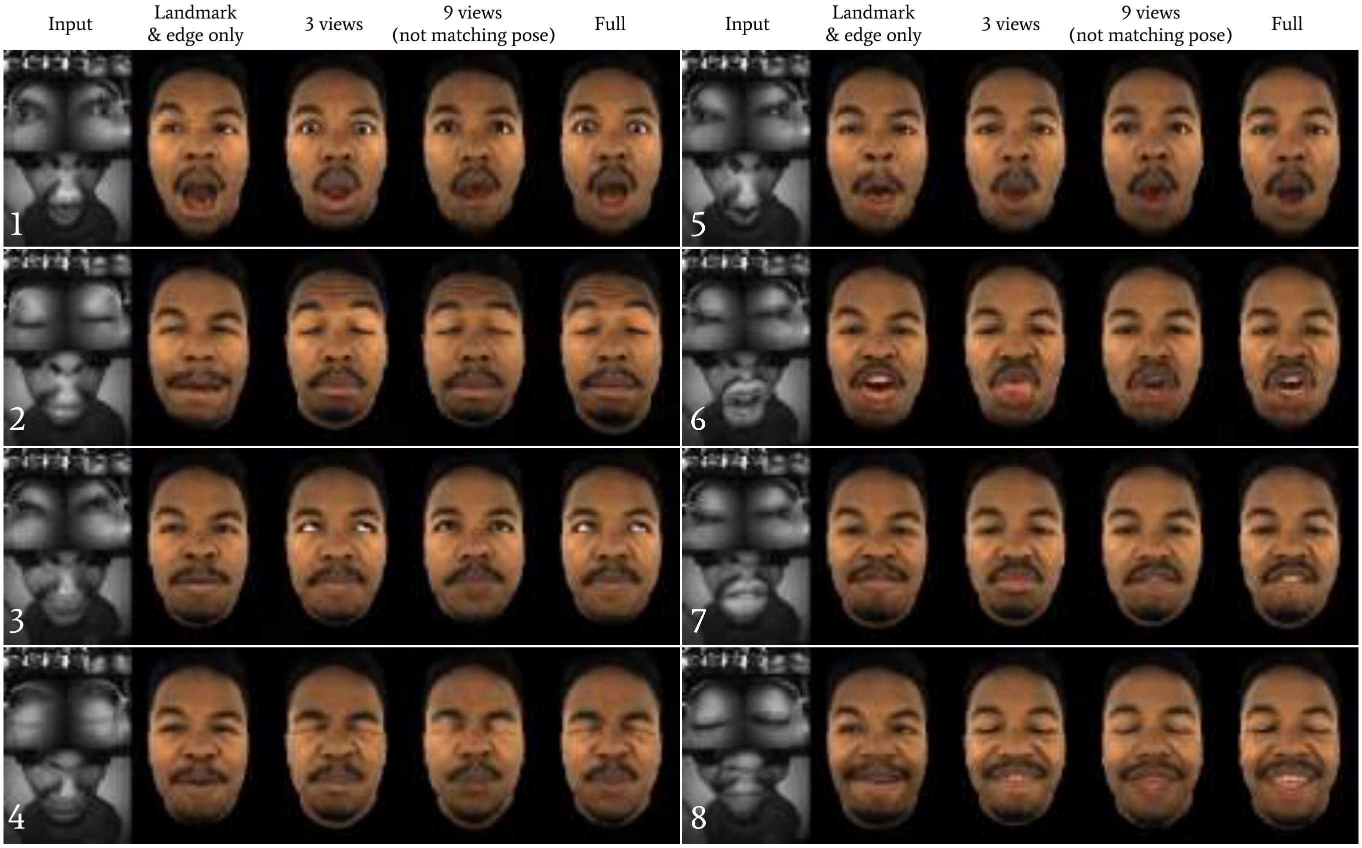

Fig. 11. **Ablation on system design**. We compare our method with simpler settings such as (a) without style transformation, only using landmarks and edge matching, (b) without additional cameras, only using 3 tracking views, and (c) without matching distribution of headset positions, only using mean headset pose to render data for style transfer. These settings failed to obtain good correspondences in different characteristics.

differentiable renderer can be small and therefore requires careful initialization. However, we found that the convergence basin of our method is large enough that no sample-specific initialization is required. In Fig. 14, we compare two training schemes with the same initialization for θ: (1) use landmark constraints (L_2-loss of 2D distances on images) for the first 2000 iterations to bring the geometry into roughly the correct position, and then remove it to leave only the image loss to fine-tune the result, and (2) only use an image loss, as presented in our method. The results show that these two schemes achieve the same level of image loss at convergence, showing that the convergence basin of (2) is good enough to avoid getting stuck in local minima. Fig. 14 also shows minimizing a landmark loss can lead to gradients that contradict the image loss (red solid curve goes up after iteration 2000). This is because of the imprecision of landmark detections in HMC images and uv-coordinate estimation on the avatar's texture map. While the landmark loss can be used to find good headset positions, for facial expression, which requires high precision, it can only provide a relatively rough signal that ultimately becomes unnecessary given the good convergence basin from the background-aware renderer.

5.2.5 Why do we need geometry and texture terms in Eq. (15)*?* To fit established correspondences in § 4, we did not simply apply loss on the predicted z^t but also geometry M and texture T. Ideally, minimizing differences in z^t should also lead to reduced differences in the appearance of the avatar. However, Fig. 15 shows that if we don't add geometry and texture terms, while the geometry error is still well minimized, the texture suffers from much higher error, which leads to mismatched expressions such as changes in gaze direction or mouth interior (i.e., regions that do not have detailed geometry). It is also interesting to see that the decrease of distance in z^t latent space becomes much slower if we add these terms, showing that z^t and appearance (at least in texture space) are not strongly coupled through the decoder,

5.3 Comparison with prior art

We compare our animation results with our reimplementations of [Olszewski et al. 2016] and [Lombardi et al. 2018]. To compare with [Olszewski et al. 2016], we select a short temporal window of HMC images labeled as a certain peak expression, and map them to the encoded z^t of the corresponding expression captured with the modeling sensors during avatar building. We also performed dynamic time warping to determine correspondence labels for speech

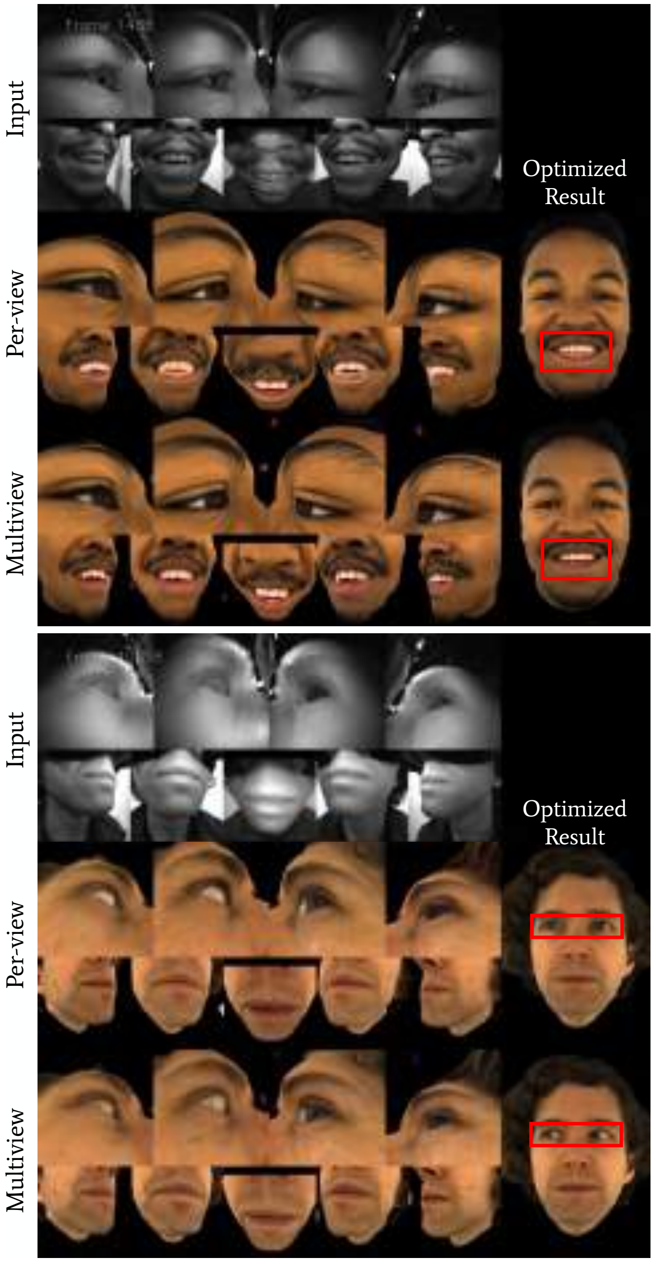

Fig. 12. **Comparison between per-view style transfer and cross-view style transfer**. In each subfigure, we show the results of style transferred multiview images and optimized result from Eq. (5) using these target images respectively. The cross-view style transferred images have better quality and better preserve semantics. Here we show the visibility of tongue and gaze direction are better mapped to input with cross-view style transfer.

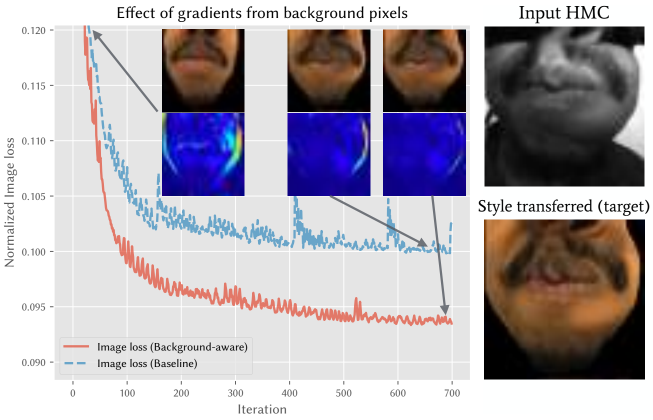

Fig. 13. **Effect of background-aware rendering**. We compare the image loss curves of a baseline (background pixels contribute no gradient) with our method, with a target "puffed cheeks" on all 9 views (only 1 is shown). While the baseline gets stuck at a higher loss, our method fully expands the cheek.

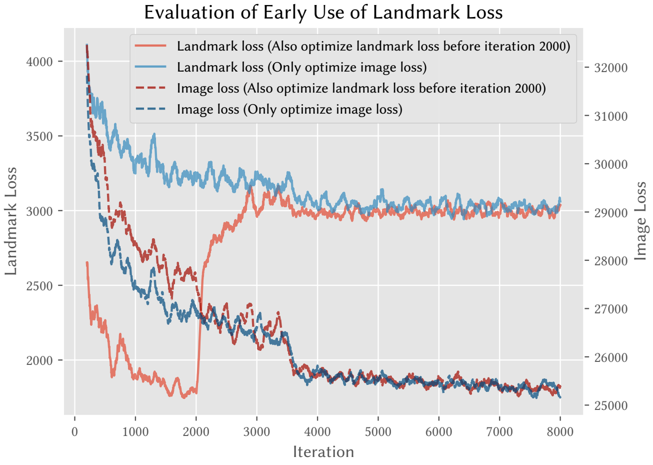

Fig. 14. **Evaluating landmark loss as initialization**. We compare two training schemes for Eq. (5): (1) use both landmark loss and image loss during the first 2000 iterations, and only use image loss after that, in red curves, and (2) only use image loss, in blue curves. The result shows the convergence basin is large enough that landmark initialization is not necessary. We also observe the contradiction between the two losses, as the red landmark loss curve goes back to a similar level as blue after iteration 2000. The learning rate is decreased by 10× at iteration 3500 and 7000 and hence the loss drop in both methods.

data. Fig. 16 (left) clearly shows the limitations. It not only suffers from inconsistency due to subjects not repeating the same expression twice, but also cannot generalize well to unlabeled expressions, such as those in a range of motion clip (a short period of free-form facial movements) or in transitions from peak expressions to neutral. Moreover, even though speech data is densely labeled through dynamic time warping based on audio, the expression, especially in the upper face, can be quite different across captures.

[Lombardi et al. 2018] only demonstrated their animation results on speech data. When applied to extreme expressions, it often generates far worse results as show in Fig. 16 (right). However, their method only uses 3 views and does not measure headset position.

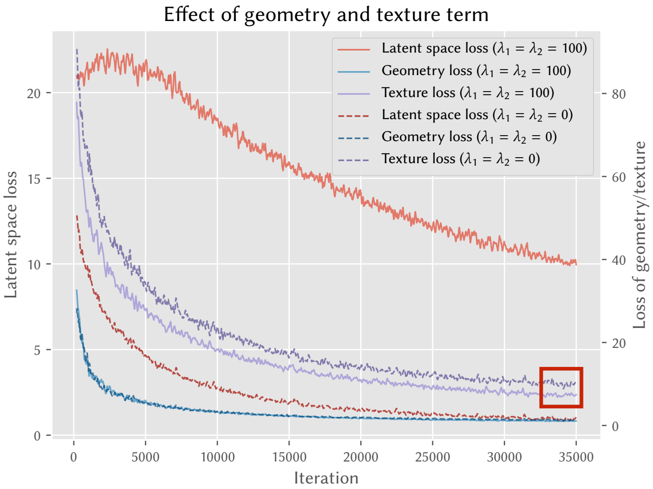

Fig. 15. **Effect of geometry and texture terms in Eq.** (15). We compare errors in the latent space of z, geometry, and texture when training with and without additional geometry and texture terms. The curves show the significant tradeoff between distributing network capacity to match the latent space loss or texture loss, showing the necessity of non-zero λ_1 and λ_2. The loss values are normalized so only relative values are meaningful.

For a better comparison, we augment their algorithm with our 9-view training HMC, as well as additionally matching the distribution of headset poses. We find that both of these design choices improve their results, showing their benefit even on other approaches. Additionally, our method still captures subtleties much more precisely, such as the tongue sticking out, lip shapes, visibility of the teeth, and the amount of mouth openness. We argue that since their method involves converting images from both domains to 1D vectors in a VAE, it is hard to ensure pixel-level alignments as well as our method, which compares the images directly.

6 DISCUSSION

In this work, we present a system that enables high-fidelity bidirectional communication in VR using photorealistic avatars. By leveraging a training headset with augmented cameras, we formulate a self-supervised learning problem that can generate sensor-to-avatar correspondences. These correspondences are then used to train a deep network that directly produces highly precise face parameters in realtime from images captured by a tracking headset with a minimal sensor design.

Although the application of this work has focused on VR headset designs, the same difficulties identified in this work apply also to AR-headsets, where camera viewpoint difficulties are exacerbated due to stricter requirements on minimal headset design. One direction of future work is to investigate the applicability of our method to these more extreme cases.

REFERENCES

Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-GAN: Unsupervised Video Retargeting. In *IEEE European Conference on Computer Vision (ECCV)*.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-quality Passive Facial Performance Capture Using Anchor Frames. *ACM Transactions on Graphics (TOG)* 30, 4, Article 75 (July 2011), 10 pages.

BinaryVR. 2019. Real-time Facial Tracking. https://www.binaryvr.com/vr.

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 187–194.

Chen Cao, Qiming Hou, and Kun Zhou. 2014. Displaced Dynamic Expression Regression for Real-time Facial Tracking and Animation. *ACM Transactions on Graphics (TOG)* 33, 4 (July 2014), 43:1–43:10.

Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. 1998. Active Appearance Models. In *IEEE European Conference on Computer Vision (ECCV)*.

Dimensional Imaging. 2016. DI4D PRO System. http://www.di4d.com/systems/di4d-pro-system/.

Epic Games. 2017. Epic Games. https://www.epicgames.com.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. 2018. Geometry-Consistent Generative Adversarial Networks for One-Sided Unsupervised Domain Mapping. *arXiv preprint arXiv:1706.00826* (2018).

Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. 2014. Driving High-Resolution Facial Scans with Video Performance Capture. *ACM Transactions on Graphics (TOG)* 34, 1, Article 8 (Dec. 2014), 8:1–8:14 pages.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NIPS)*.

Adam W. Harley, Shih-En Wei, Jason Saragih, and Katerina Fragkiadaki. 2019. Image Disentanglement and Uncooperative Re-Entanglement for High-Fidelity Image-to-Image Translation. *arXiv preprint arXiv:1901.03628* (2019).

Hellblade. 2018. Hellblade. https://www.hellblade.com/.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image Translation with Conditional Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. 2015. Deep Convolutional Inverse Graphics Network. In *Advances in Neural Information Processing Systems (NIPS)*. 2539–2547.

Samuli Laine, Tero Karras, Timo Aila, Antti Herva, Shunsuke Saito, Ronald Yu, Hao Li, and Jaakko Lehtinen. 2017. Production-level Facial Performance Capture Using Deep Convolutional Neural Networks. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. Article 10, 10 pages.

Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. 2015. Facial Performance Sensing Head-mounted Display. *ACM Transactions on Graphics (TOG)* 34, 4, Article 47 (July 2015), 47:1–47:9 pages.

Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep Appearance Models for Face Rendering. *ACM Transactions on Graphics (TOG)* 37, 4, Article 68 (July 2018), 13 pages.

Magic Leap. 2018. Magic Leap. https://www.magicleap.com/.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated Hands for Real-time 3D Hand Tracking from Monocular RGB. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vinod Nair, Josh Susskind, and Geoffrey E. Hinton. 2008. Analysis-by-Synthesis by Learning to Invert Generative Black Boxes. In *Proceedings of the 18th International Conference on Artificial Neural Networks (ICANN), Part I*. 971–981.

Kyle Olszewski, Joseph J. Lim, Shunsuke Saito, and Hao Li. 2016. High-fidelity Facial and Speech Animation for VR HMDs. *ACM Transactions on Graphics (TOG)* 35, 6, Article 221 (Nov. 2016), 14 pages.

Jason M. Saragih, Simon Lucey, and Jeffrey F. Cohn. 2009. Face Alignment through Subspace Constrained Mean-shifts. In *IEEE International Conference on Computer Vision (ICCV)*.

Mike Seymour, Chris Evans, and Kim Libreri. 2017. Meet Mike: Epic Avatars. In *ACM SIGGRAPH 2017 VR Village*. Article 12, 2 pages.

Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. 2017. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*.

Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. 2016. Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

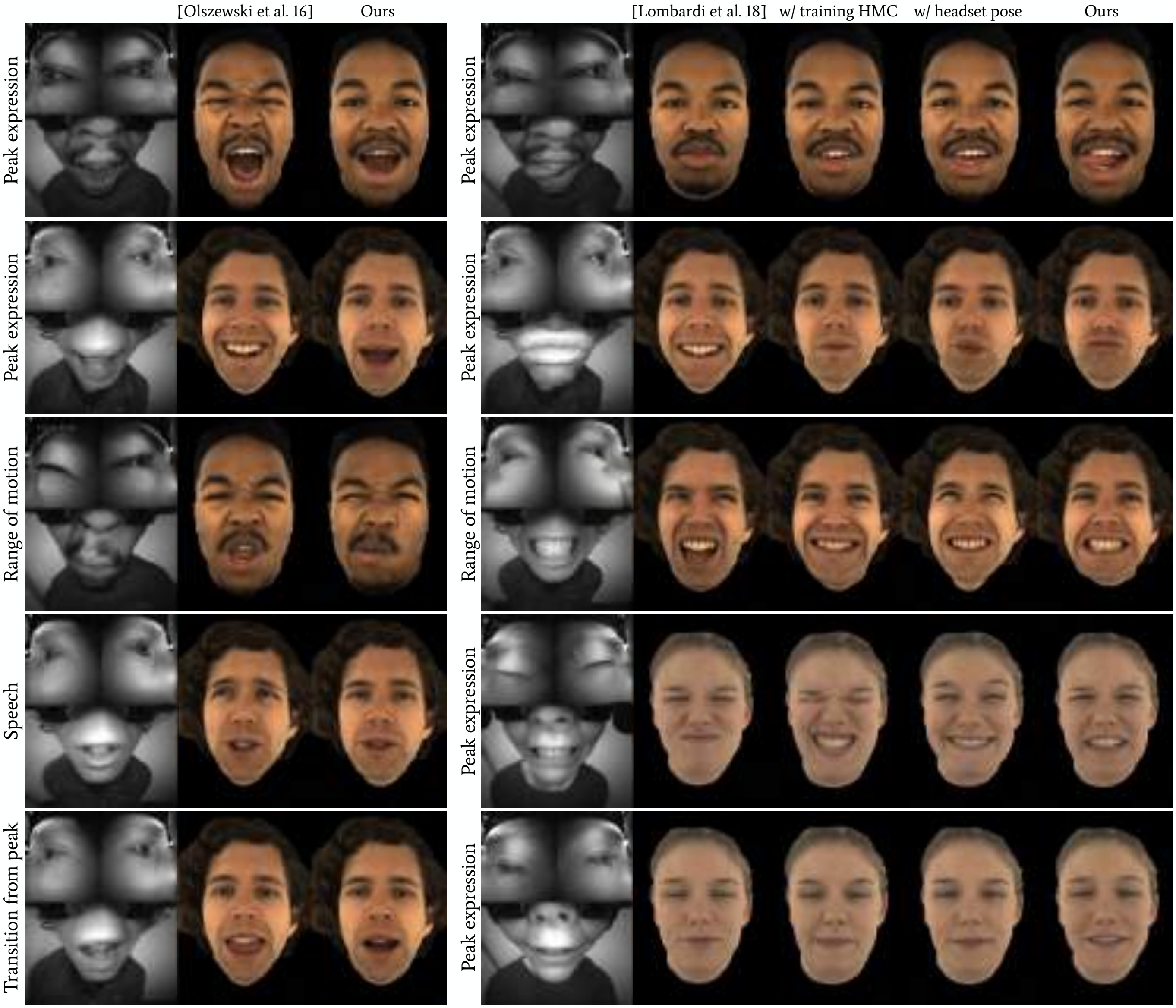

Fig. 16. **Comparison with prior arts**. On the left, we compare our results with [Olszewski et al. 2016], which suffers from the subject's inconsistency in peak expressions and poor generalization to data they cannot acquire semantic labels. On the right, we compare with [Lombardi et al. 2018] and also with their method augmented with our design, including more input views ("w/ training HMC"), and matching distribution of headset position ("w/ headset pose"). Their results improve when augmented with our designs, but our methods still significantly better capture nuances on the face.

Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Niessner. 2018. FaceVR: Real-Time Gaze-Aware Facial Reenactment in Virtual Reality. *ACM Transactions on Graphics (TOG)* 37, 2, Article 25 (June 2018), 25:1–25:15 pages.

Unreal Engine 4. 2018. Unreal Engine 4. https://www.unrealengine.com/.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Jing Xiao, Jinxiang Chai, and Takeo Kanade. 2006. A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. *International Journal of Computer Vision (IJCV)* 67, 2 (April 2006), 233–246.

Xuehan Xiong and Fernando De la Torre. 2013. Supervised Descent Method and Its Applications to Face Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Ilker Yildirim, Winrich Freiwald, Tejas Kulkarni, and Joshua B. Tenenbaum. 2015. Efficient Analysis-by-synthesis in Vision: A Computational Framework, Behavioral Tests, and Comparison with Neural Representations. In *Proceedings of 37th Annual Conference of the Cognitive Science Society.*

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision (ICCV).*