西北工业大学

数字图像处理–论文翻译

原论文标题：3D Gaussian Blendshapes for Head Avatar Animation

魏梦奇

计算机学院

计算机科学与技术

2024 年 11 月

学号：2022302723

Northwestern Polytechnical University

3D Gaussian Blendshapes for Head Avatar Animation

Shengjie Ma
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, China
qtdysjj@gmail.com

Yanlin Weng
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, China
weng@cad.zju.edu.cn

Tianjia Shao
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, China
tjshao@zju.edu.cn

Kun Zhou*
State Key Lab of CAD&CG, Zhejiang University
Hangzhou, China
kunzhou@acm.org

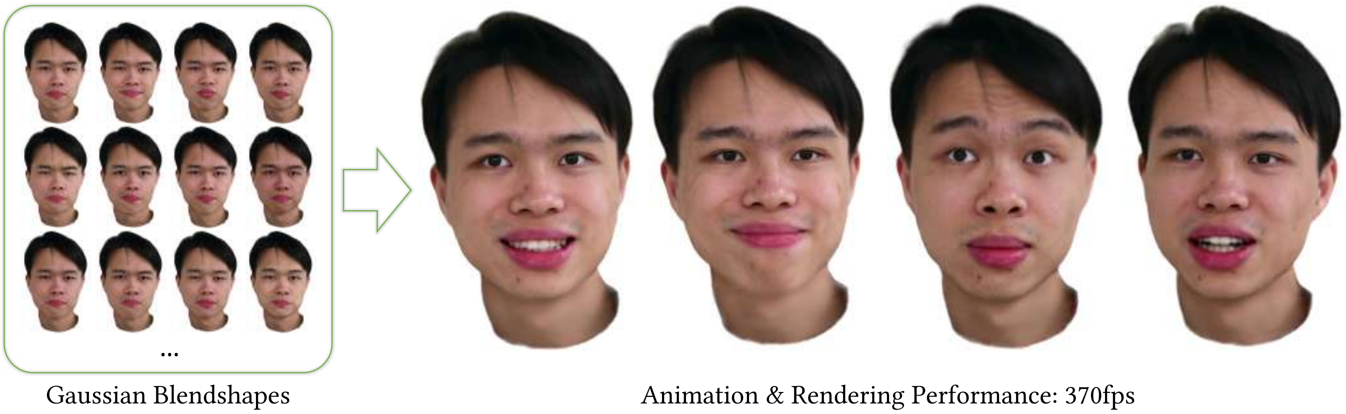


Figure 1: Our 3D Gaussian blendshapes are analogous to mesh blendshapes in classical parametric face models, which can be linearly blended with expressions coefficients to synthesize photo-realistic avatar animations in real time (370fps).

ABSTRACT

We introduce 3D Gaussian blendshapes for modeling photorealistic head avatars. Taking a monocular video as input, we learn a base head model of neutral expression, along with a group of expression blendshapes, each of which corresponds to a basis expression in classical parametric face models. Both the neutral model and expression blendshapes are represented as 3D Gaussians, which contain a few properties to depict the avatar appearance. The avatar model of an arbitrary expression can be effectively generated by combining the neutral model and expression blendshapes through linear blending of Gaussians with the expression coefficients. High-fidelity head avatar animations can be synthesized in real time using Gaussian splatting. Compared to state-of-the-art methods, our Gaussian blendshape representation better captures high-frequency details exhibited in input video, and achieves superior rendering performance.

CCS CONCEPTS

• **Computing methodologies → Reconstruction**; **Point-based models**.

KEYWORDS

Parametric face models, facial animation, facial tracking, facial reenactment

ACM Reference Format:
Shengjie Ma, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2024. 3D Gaussian Blendshapes for Head Avatar Animation. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24 (SIGGRAPH Conference Papers '24), July 27-August 1, 2024, Denver, CO, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641519.3657462

1 INTRODUCTION

Reconstructing and animating 3D human heads has been a long studied problem in computer graphics and computer vision, which is the key technology in a variety of applications such as telepresence, VR/AR and movies. Most recently, head avatars based on neural radiance fields (NeRF) [Mildenhall et al. 2020] demonstrate great potential in synthesizing photorealistic images. These techniques achieve dynamic avatar control typically by conditioning

*Corresponding author

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA

ACM ISBN 979-8-4007-0525-0/24/07
https://doi.org/10.1145/3641519.3657462

NeRFs on a parametric head model [Zielonka et al. 2023] or expression codes [Gafni et al. 2021]. Gao et al. [2022] and Zheng et al. [2022] instead propose to construct a set of NeRF blendshapes and linearly blend them to animate the avatar.

The blendshape model is a classic representation for avatar animation. It consists of a group of 3D meshes, each of which corresponds to a basis expression. The face shape of an arbitrary expression can be efficiently computed by linearly blending the basis meshes with corresponding expression coefficients. The advantages of easy-to-control and high efficiency make blendshape models the most popular representation in professional animation production [Lewis et al. 2014] as well as consumer avatar applications (e.g., iPhone Memoji) [Weng et al. 2014].

In this paper, we introduce a 3D Gaussian blendshape representation for constructing and animating head avatars. We build the representation upon 3D Gaussian splatting (3DGS) [Kerbl et al. 2023], which represents the radiance field of a static scene as 3D Gaussians and provides compelling quality and speed in novel view synthesis. Our representation consists of a base model of neutral expression and a group of expression blendshapes, all represented as 3D Gaussians. Each Gaussian contains a few properties (e.g., position, rotation and colors) as in 3DGS and depicts the appearance of the head avatar. Each Gaussian blendshape corresponds to a mesh blendshape of traditional parametric face models [Cao et al. 2014b; Li et al. 2017] and has the same semantic meaning. A Gaussian head model of an arbitrary expression can be generated by blending the Gaussian blendshapes with the expression coefficients, which is rendered to high-fidelity images in real time using Gaussian splatting. The motion parameters tracked by previous face tracking algorithms (e.g., [Cao et al. 2014a; Zielonka et al. 2022]) can be used to drive the Gaussian blendshapes to produce head avatar animations.

We propose to learn the Gaussian blendshape representation from a monocular video. We use previous methods to construct the mesh blendshapes from the input video, and distribute a number of Gaussians on the mesh surfaces as an initialization. We then jointly optimize all Gaussian properties. As Gaussian blendshapes are driven by the same expression coefficients for mesh blendshapes, each Gaussian blendshape must be semantically consistent with its corresponding mesh blendshape, i.e., the differences between the Gaussian blendshape and neutral model should be consistent with the differences between the corresponding mesh blendshape and neutral mesh. Directly optimizing Gaussian properties without considering blendshape consistency causes overfitting and artifacts for novel expressions unseen in training. To this end, we present an effective strategy to guide the Gaussian optimization to follow the consistency requirement. Specifically, we introduce an intermediate variable to formulate the Gaussian difference as terms proportional to the mesh difference. By optimizing this intermediate variable directly during training, we produce Gaussian blendshapes differing from the neutral model in a consistent way that mesh blendshapes differ from the neutral mesh.

Extensive experiments demonstrate that our Gaussian blendshape method outperforms state-of-the-art methods [Gao et al. 2022; Zheng et al. 2023; Zielonka et al. 2023] in synthesizing high-fidelity head avatar animations that best capture high-frequency details observed in input video, and achieving significantly faster speeds in avatar animation and rendering (see Fig. 1).

2 RELATED WORK

Researchers have proposed various representations for head avatars. Early works employ explicit 3D mesh representation to reconstruct the 3D shape and appearance from images. The seminal work [Blanz and Vetter 1999] proposes the 3D Morphable Model (3DMM) to model the face shape and texture on a low-dimensional linear subspace. There are many follow-up works along this direction such as full-head models [Ploumpis et al. 2021], and deep non-linear models [Tran and Liu 2018]. The 3D mesh representation is also used to build riggable heads for head animation [Bai et al. 2021; Chaudhuri et al. 2020; Hu et al. 2017]. To generate detailed animations, researchers further propose image-based dynamic avatars controlling the full head with hair and headwear [Cao et al. 2016], or additionally reconstruct fine-level correctives [Feng et al. 2021; Garrido et al. 2016; Ichim et al. 2015; Yang et al. 2020].

In order to achieve high realism rendering, recent approaches utilize neural radiance fields (NeRF) [Mildenhall et al. 2020] to implicitly represent head avatars and have achieved impressive results [Gafni et al. 2021; Grassal et al. 2022; Jiang et al. 2022; Lombardi et al. 2021; Xu et al. 2022, 2023b,c; Zheng et al. 2022]. For instance, i3DMM [Yenamandra et al. 2021] presents the first neural implicit function based on the 3D morphable model of full heads. HeadNerf [Hong et al. 2022] introduces a NeRF-based parametric head model that integrates the neural radiance field to the parametric representation of the head. The state-of-the-art work INSTA [Zielonka et al. 2023] models a dynamic neural radiance field based on Instant-NGP [Müller et al. 2022] embedded around a parametric face model. It is able to reconstruct a head avatar in less than 10 minutes. PointAvatar [Zheng et al. 2023] presents a point-based representation and learns a deformation field based on FLAME's expression vectors to drive the points. NeRFBlendshape [Gao et al. 2022] constructs NeRF-based blendshape models for semantic animation control and photorealistic rendering by combining multi-level voxel fields with expression coefficients.

Many concurrent works have been proposed to apply the 3D Gaussian representation introduced by [Kerbl et al. 2023] to construct head avatars (e.g., [Chen et al. 2023; Dhamo et al. 2023; Qian et al. 2023; Saito et al. 2023; Wang et al. 2023; Xiang et al. 2023; Xu et al. 2023a]). Most of them use the 3D Gaussian representation together with neural networks. For example, GaussianHead [Wang et al. 2023] uses Multi-layer Perceptrons (MLPs) to decode the dynamic geometry and radiance parameters of Gaussians. FlashAvatar [Xiang et al. 2023] attaches Gaussians on a mesh with learnable offsets, which are represented as MLPs. Saito et al. [2023] construct relightable head avatars by using networks to decode the parameters of 3D Gaussians and learnable radiance transfer functions. To our knowledge, none of concurrent works introduce the idea of Gaussian blendshapes as in our paper. A unique advantage of our method is that it only requires linear blending of Gaussian blendshapes to construct a head avatar of arbitrary expressions, which brings significant benefits in both training and runtime performance. The method closest to our work in terms of performance is FlashAvatar [Xiang et al. 2023], which achieves 300fps for 10k

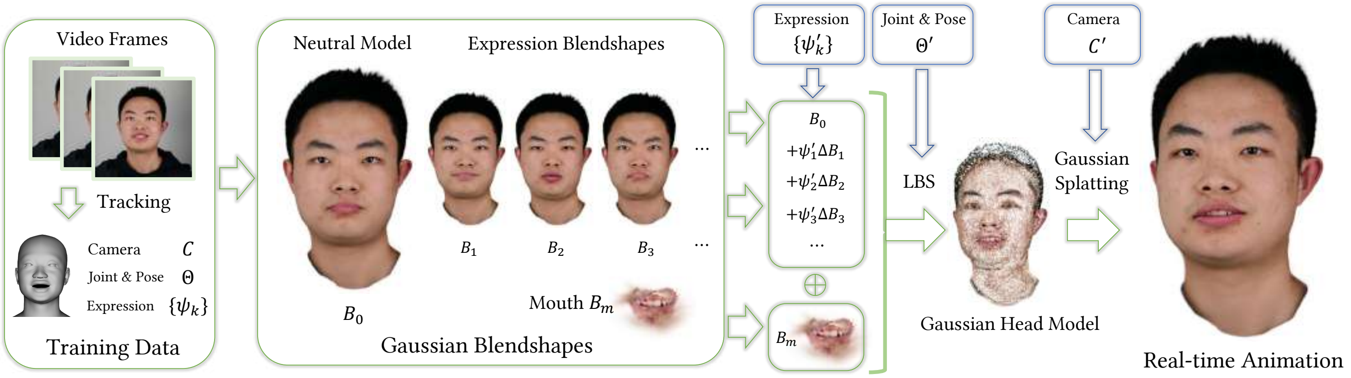


Figure 2: Overview of our method. Taking a monocular video as input, our method learns a Gaussian blendshape representation of a head avatar, which consists of a neutral model B_0, a group of expression blendshapes $\{B_1, B_2, ..., B_K\}$, and the mouth interior model B_m, all represented as 3D Gaussians. Avatar models of arbitrary expressions and poses can be generated by linear blending with expression coefficients $\{\psi'_k\}$ and linear blend skinning with joint and pose parameters Θ', from which we render high-fidelity images in real time using Gaussian splatting.

Gaussians and degrades to ~100fps for 50k Gaussians, while we achieve 370fps for 70k Gaussians.

3 METHOD

3.1 3D Gaussian BlendShapes

Our Gaussian blendshape representation consists of a neutral base model B_0 and a group of expression blendshapes $\{B_1, B_2, ..., B_K\}$, all represented as a set of 3D Gaussians, each of which has a few basic properties (i.e., position $\mathbf{x}$, opacity α, rotation $\mathbf{q}$, scale $\mathbf{s}$ and spherical harmonics coefficients SH) as in 3DGS [Kerbl et al. 2023]. Each Gaussian of B_0 also has a set of blend weights $\mathbf{w}$ for joint and pose control. There is a one-to-one correspondence between the Gaussians of B_0 and each blendshape B_k. The deviation of B_k from B_0 can be defined as the difference between their Gaussian properties, $\Delta B_k = B_k - B_0$. The head avatar model of an arbitrary expression is computed as:

$$B^{\psi} = B_0 + \sum_{k=1}^{K} \psi_k \Delta B_k, \tag{1}$$

where $\{\psi_k\}$ are the expression coefficients.

Currently we use the Principal Component Analysis (PCA) based blendshape model FLAME [Li et al. 2017], although other muscle-inspired blendshapes such as the Facial Action Coding System (FACS) based model FaceWarehouse [Cao et al. 2014b] can be also employed. Besides facial expression control, FLAME also provides joint and pose parameters, Θ, for controlling the motions of head, jaw, eyeballs and eyelids, which are used with linear blend skinning (LBS) to transform the head avatar model (i.e., its Gaussians): $B^{\psi *} = LBS(B^{\psi}, \Theta)$, where the blend weights associated with Gaussians of B_0 are used.

Mouth Interior Gaussians. The motions of mouth interior and hair are usually not affected by facial expressions, and thus not covered in the FLAME mesh, neither the blendshape model described above. Hair can move with the head rigidly, while the motion of teeth is controlled by the jaw joint in FLAME. We find that in practice the blendshape Gaussians generated in our training are able to model hair well, but the mouth interior results are not good enough. We thus define a separate set of Gaussians for mouth interior, B_m, which move with the jaw joint in FLAME. The properties of these mouth Gaussians do not change with expressions, but are only transformed with the jaw joint, i.e., $B^*_m = LBS(B_m, \Theta)$.

Finally, the transformed Gaussian model ($B^{\psi *}$, B^*_m) can be rendered to high-fidelity images I_r in real time using Gaussian splatting. Fig. 2 shows the overview of our method.

3.2 Training

Data Preparation. Following [Zielonka et al. 2023], we use the face tracker of [Zielonka et al. 2022] to compute the FLAME meshes of neutral expression and $K = 50$ basis expressions, as well as the camera parameters, joint and pose parameters, and expression coefficients, for each video frame. We also extract the foreground head mask for each input frame.

Initialization. We first initialize the neutral model B_0, expression blendshapes $\{B_k\}$, as well as the mouth interior Gaussians B_m. For B_0, we distribute a number of points on the neutral FLAME mesh M_0 using Poisson disk sampling [Bowers et al. 2010], and use them as the initialization of Gaussian positions. Other Gaussian properties are initialized as in 3DGS [Kerbl et al. 2023]. For each Gaussian, we also find its closest triangle on M_0, and compute its LBS blend weights as the linear interpolation of blend weights of the triangle vertices. To initialize the mouth interior Gaussians B_m, we use two pre-defined billboards to represent the upper and lower teeth, which are sampled to Gaussians using Poisson disk sampling. The upper teeth Gaussians are rigidly bound to the back of the head, while lower teeth Gaussians are bound to the vertex having the largest skinning weight for the jaw joint.

To initialize the expression blendshape B_k, we transform each Gaussian of B_0 using the deformation gradients [Sumner and Popović 2004] from M_0 to the expression FLAME mesh M_k. Specifically, for each neutral Gaussian G^i_0, we compute the affine transformation from its closest triangle on M_0 to the corresponding triangle on M_k, and extract the rotation component [Shoemake and Duff 1992], which is applied to the position, rotation and spherical harmonics (SH) coefficients of G_0^i to yield the corresponding Gaussian G_k^i of expression blendshape B_k. Note that we omit the scale component as we find the transformation is very close to rigid. The scale and opacity properties of G_k^i are kept the same as those of G_0^i. In this way, we can construct each expression blendshape B_k from B_0, as well as their difference $\Delta B_k = B_k - B_0$.

Optimization. After initialization, we jointly optimize B_0, $\{\Delta B_k\}$, and B_m. For each video frame, we reconstruct the Gaussian head model B_ψ by linearly blending B_0 and $\{\Delta B_k\}$ with the tracked expression coefficients according to Eq. (1), and then transform B_ψ and B_m using LBS with the tracked joint and pose parameters: $B^{\psi *} = LBS(B^\psi, \Theta)$, $B_m^* = LBS(B_m, \Theta)$. Finally, we get the rendered image from $B^{\psi *}$ and B_m^* using Gaussian splatting. The optimization process is similar to 3DGS [Kerbl et al. 2023], which also involves adaptive density control steps of adding and removing Gaussians.

During optimization, a crucial thing to avoid overfitting is to preserve the semantic consistency between each Gaussian blendshape B_k and its corresponding mesh blendshape M_k. As aforementioned, the Gaussian blendshapes are blended using the same tracked expression coefficients based on the parametric mesh model of FLAME, in both training and runtime computations. To ensure the semantic validity of such blending calculation, the difference between B_k and B_0 (i.e., ΔB_k) must be consistent with the difference between M_k and M_0 (i.e., ΔM_k), which means in head regions having large vertex position differences between M_k and M_0, the Gaussian differences between B_k and B_0 should also be large, and small otherwise. Directly optimizing $\{\Delta B_k\}$ without such consistency consideration will lead to overfitting, where apparent artifacts easily occur on novel expression coefficients unseen in the training images (see Fig. 4 for examples).

However, unlike ΔM_k only containing vertex position displacements, ΔB_k contains different kinds of properties, such as position, rotation, and color. It is thus difficult to design a loss function term to explicitly enforce consistency between ΔB_k and ΔM_k, while not sacrificing the image loss. Instead, we propose a simple yet effective strategy to guide the Gaussian optimization to implicitly follow the consistency requirement. Specifically, for each Gaussian G_i, let $\Delta G_{i,k}$ be the difference between its properties in B_k and B_0. We introduce an intermediate variable, $\Delta \widehat{G}_{i,k}$, to formulate $\Delta G_{i,k}$ as:

$$\Delta G_{i,k} = \Delta G_{i,k}^{init} + max(f(d_{i,k}), 0)\Delta \widehat{G}_{i,k}, \tag{2}$$

where $\Delta G_{i,k}^{init}$ is the initial value of $\Delta G_{i,k}$ calculated in the aforementioned initialization stage and regarded as a constant during optimization, and $d_{i,k}$ is the magnitude of position displacement of the surface point closest to G_i, from M_0 to M_k. The linear function $f(x) = (x - \epsilon)/(\tilde{d} - \epsilon)$ scales the maximum magnitude of position difference $\tilde{d}$ between M_k and M_0 to 1, and a threshold magnitude $\epsilon = 0.00001$ to 0. The *max* function is necessary to avoid negative scaling values for positional displacement magnitudes below ϵ.

Eq. (2) essentially represents the actual Gaussian difference $\Delta G_{i,k}$ as the sum of its initial value and the scaled value of $\Delta \widehat{G}_{i,k}$ according to its corresponding positional displacement in mesh blendshapes, which effectively correlates Gaussian differences with position displacements. The initial Gaussian difference $\Delta G_{i,k}^{init}$ is proportional to the position displacement of mesh blendshapes, as it is computed using the deformation gradients from M_0 to M_k. Scaling $\Delta \widehat{G}_{i,k}$ according to positional displacement ensures that the Gaussian difference $\Delta G_{i,k}$ is updated at a rate proportional to the position displacement. Please note for Gaussians with positional displacement magnitudes below ϵ, the second term in Eq. (2) vanishes to

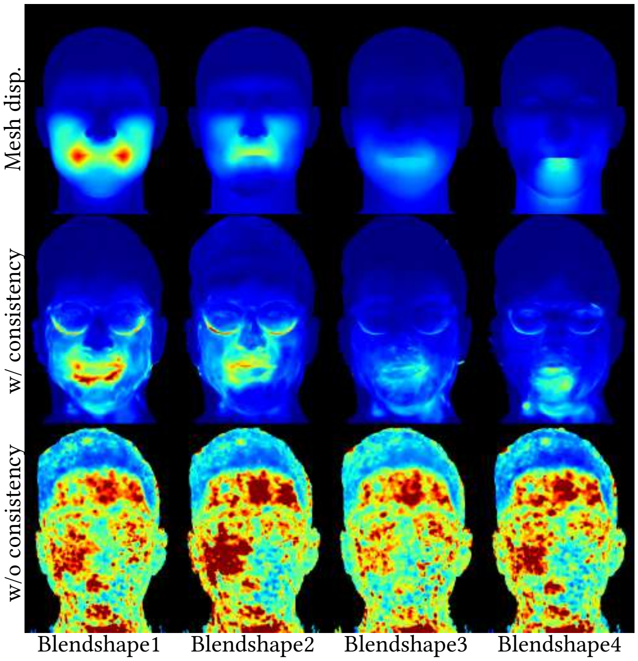


Figure 3: The impact of the blendshape consistency on the optimization of expression blendshapes. The first row shows the displacement magnitude between M_k and M_0. The second and the third rows show the magnitude of optimized ΔB_k with or without blendshape consistency.

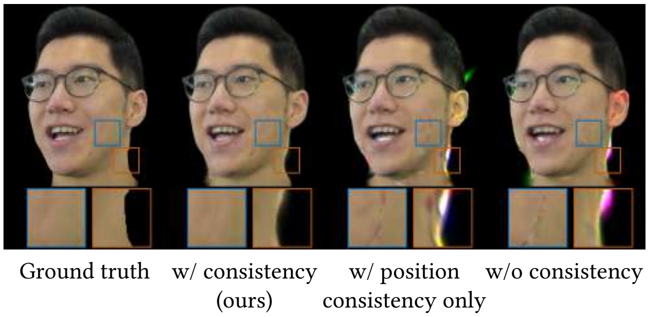


Figure 4: Ablation study on blendshape consistency. The optimization without blendshape consistency leads to apparent artifacts like dirty color and glitch in both interior and boundary areas. Enforcing blendshape consistency only on Gaussian positions also leads to poor results.

Table 1: Quantitative comparisons between INSTA [Zielonka et al. 2023], PointAvatar [Zheng et al. 2023], and our method.

Datasets		INSTA dataset								Our dataset			
		bala	biden	justin	malte_1	marcel	nf_01	nf_03	wojtek_1	subject1	subject2	subject3	subject4
PSNR ↑	INSTA	28.66	28.38	29.74	26.27	23.75	25.89	26.10	29.84	28.88	28.16	28.60	30.83
	PointAvatar	29.60	31.72	32.31	27.46	24.60	28.34	29.82	31.94	31.43	32.57	30.95	32.57
	Ours	33.34	32.48	32.49	28.56	26.61	27.92	28.62	32.39	32.87	32.55	31.54	34.03
SSIM ↑	INSTA	0.9130	0.9484	0.9530	0.9262	0.9133	0.9246	0.9129	0.9457	0.9195	0.9443	0.9078	0.9445
	PointAvatar	0.9099	0.9565	0.9595	0.9225	0.9121	0.9278	0.9208	0.9502	0.9219	0.9463	0.9062	0.9433
	Ours	0.9490	0.9672	0.9696	0.9461	0.9348	0.9448	0.9381	0.9645	0.9384	0.9577	0.9268	0.9646
LPIPS ↓	INSTA	0.0817	0.0545	0.0614	0.0751	0.1540	0.1285	0.1137	0.0588	0.1536	0.1208	0.1733	0.1144
	PointAvatar	0.0821	0.0535	0.0649	0.0718	0.1574	0.1350	0.1221	0.0661	0.1568	0.1190	0.1715	0.1285
	Ours	0.0772	0.0522	0.0631	0.0703	0.1391	0.1174	0.0965	0.0595	0.1520	0.1197	0.1689	0.1075

Table 2: Quantitative comparisons between NeRFBlendShape [Gao et al. 2022] and our method.

Datasets		NeRFBlendShape dataset					
		id1	id2	id3	id4	id5	id6
PSNR ↑	NeRFBlendShape	32.12	32.25	37.25	36.86	34.26	35.74
	Ours(w/o LPIPS)	33.13	33.23	39.83	38.34	35.58	36.59
	Ours(w/ LPIPS)	33.09	33.15	39.64	38.17	35.46	36.46
SSIM ↑	NeRFBlendShape	0.9412	0.9369	0.9750	0.9791	0.9541	0.9786
	Ours(w/o LPIPS)	0.9532	0.9473	0.9836	0.9846	0.9635	0.9814
	Ours(w/ LPIPS)	0.9522	0.9457	0.9828	0.9837	0.9625	0.9806
LPIPS ↓	NeRFBlendShape	0.0715	0.0756	0.0436	0.0460	0.0448	0.0366
	Ours(w/o LPIPS)	0.0862	0.0937	0.0486	0.0495	0.0538	0.0414
	Ours(w/ LPIPS)	0.0550	0.0620	0.0359	0.0334	0.0381	0.0303

0 and $\Delta G_{i,k}$ is always equal to $\Delta G_{i,k}^{init}$, which is inherently proportional to the positional displacement.

Instead of optimizing $\Delta G_{i,k}$, we directly optimize $\Delta \widehat{G}_{i,k}$ using the loss functions described in the following section. Specifically, $\Delta \widehat{G}_{i,k}$ is initialized to 0. Each time $\Delta \widehat{G}_{i,k}$ is updated, we calculate $\Delta G_{i,k}$ according to Eq. (2), from which the avatar model is constructed. The avatar model is then rendered to an image through Gaussian splatting, which is used in loss function computation.

In this way, we effectively guide the Gaussian differences to change consistently with positional displacements, leading to optimized Gaussian blendshapes with strong semantic consistency with mesh blendshapes (see Fig. 3).

3.3 Loss Functions

The optimization goal is to minimize the image loss between the rendering and input, under some regularization constraints. The first loss is the image loss as in 3DGS [Kerbl et al. 2023], consisting of the L_1 differences between the rendered images and the video frames and a D-SSIM term:

$$L_{rgb} = (1 - \lambda)L_1 + \lambda L_{D-SSIM} \tag{3}$$

with $\lambda = 0.2$.

We also design an alpha loss to constrain the Gaussians to stay within the head region. We perform Gaussian splatting to get the accumulated opacity image I_α, and compare it with the foreground head mask $Mask_h$. The alpha loss is defined as:

$$L_\alpha = \frac{1}{F}\sum_{i=1}^{F}(||(I_\alpha^i - Mask_h^i)||_2), \tag{4}$$

where F is the frame number.

We further introduce a regularization loss to constrain the mouth interior Gaussians to stay within a pre-defined volume of the mouth. Specifically, we compute the signed distance for each Gaussian to the volume boundary and apply an L_2 loss to retract it when it goes out of the volume. The regularization loss is defined as:

$$L_{reg} = \frac{1}{N}\sum_{i=1}^{N}(||max(SDF(\mathbf{x}_i, V), 0)||_2^2), \tag{5}$$

where V is the pre-defined cylindrical volume, $\mathbf{x}_i$ is the Gaussian position and N is the number of mouth interior Gaussians. The overall loss function is defined as:

$$L = \lambda_1 L_{rgb} + \lambda_2 L_\alpha + \lambda_3 L_{reg}. \tag{6}$$

We set $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 100$ by default.

3.4 Implementation Details

We implement our method using Pytorch. The Adam solver [Kingma and Ba 2015] is employed for parameter optimization. The learning rates are 3.2×10^{-7}, 5×10^{-5}, 5×10^{-4}, 1×10^{-4}, 1.25×10^{-3} respectively for the Gaussian properties $\{\mathbf{x}_k, \alpha_k, \mathbf{s}_k, \mathbf{q}_k, SH_k\}$. The initially sampled Gaussian number is 50k for the neutral model, and 14k for the mouth interior Gaussians.

The training is conducted on an A800 GPU and testing is conducted on an RTX 4090 GPU. We also build a C++/CUDA interactive viewer following 3DGS [Kerbl et al. 2023] and use it to measure our runtime frame rates.

As Gaussian positions frequently change during optimization, we need to efficiently update their LBS blend weights $\mathbf{w}$ and the positional displacements of nearest points $\{d_{i,k}\}$. We precompute and store these values on a 3D grid of $256 \times 256 \times 256$ surrounding the neutral mesh M_0. The values of an arbitrary Gaussian can be effectively computed as the linear blending of the values of eight grid points nearest to the Gaussian center.

4 RESULTS

4.1 Baselines and Datasets

We compare our method with state-of-the-art methods, NeRF-based INSTA [Zielonka et al. 2023] and point-based PointAvatar [Zheng

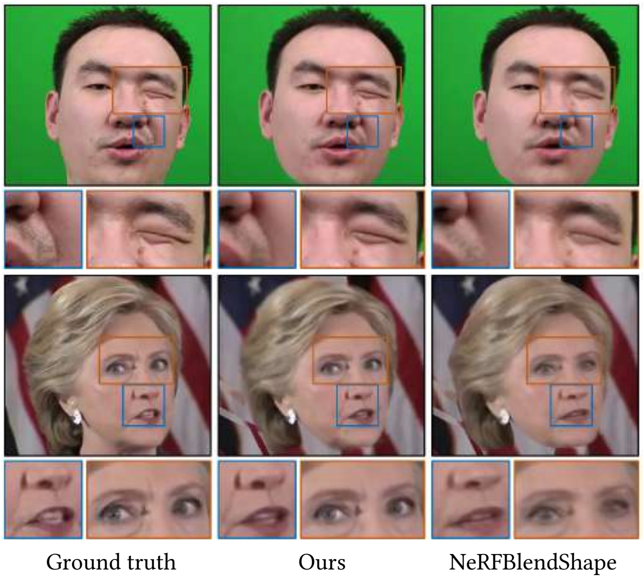


Figure 5: Qualitative comparisons with NeRFBlendShape [Gao et al. 2022]. Our method more faithfully captures fine facial details (e.g., wrinkles around the eyes and nose), and better recovers the eyeball movement. YouTube video ID is -yHgE9W699w for Hillary Clinton.

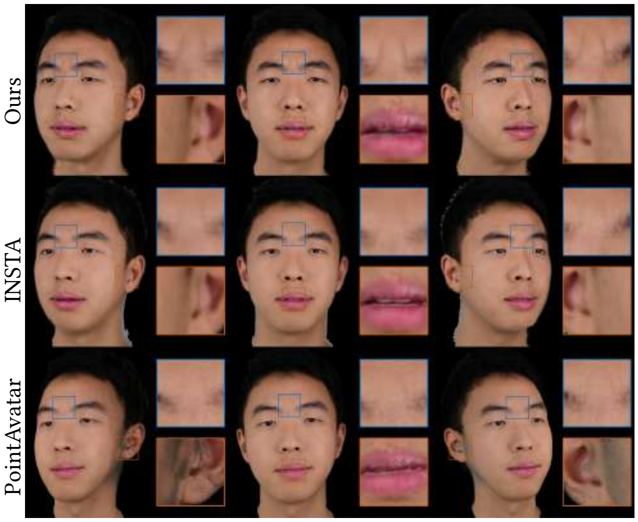


Figure 6: Qualitative comparisons for novel view extrapolation. Our method produces better results with fine details under novel views.

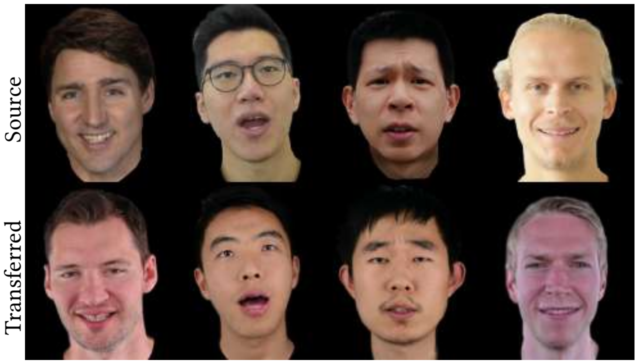


Figure 7: Results of cross-identity reenactment. YouTube video ID is mKHgXHKbJUE for Justin Trudeau.

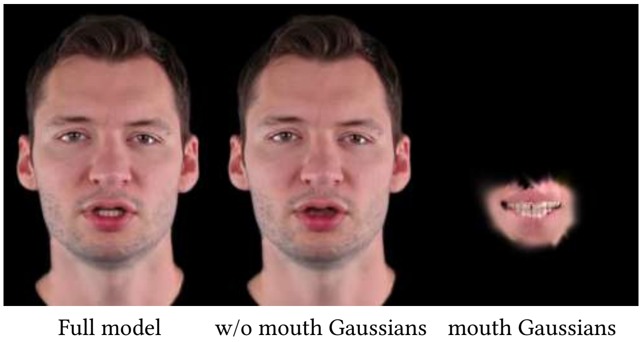


Figure 8: Demonstration of mouth interior Gaussians.

Table 3: Performance comparisons. We perform training for all methods on an A800 GPU. Testing is done on a RTX 4090 GPU for INSTA, NeRFBlendshape, and our method, but on an A800 GPU for PointAvatar due to out-of-memory errors on the RTX 4090 GPU. The rendering resolution is 512×512. Note our running time includes both animation (i.e., linear blending and LBS transformation) and rendering, and our performance is insensitive to the rendering resolution. We also report the peak memory consumption during training and runtime computation.

	Training	Runtime	Mem. (train)	Mem. (runtime)
INSTA	10min	70fps	16G	4G
PointAvatar	3.5h	5fps*	40G	32G*
NeRFBlendShape	20min	26fps	7G	2G
Ours	25min	370fps	14G	2G

et al. 2023], on the INSTA dataset and our own dataset. We hold the last 350 frames of each video as the test set for self-reenactment similar to INSTA. The training data preparation time is about 12 hours for 4500 frames as in INSTA. We also compare our method with NeRFBlendShape [Gao et al. 2022] on their public dataset consisting of eight videos. The last 500 frames of each video are reserved for test. We reduce the alpha weight λ_2 to 1 for the entire dataset due to the relatively inaccurate binary foreground mask provided, which we find slightly sharpens the hair around the contour.

Our own dataset consists of four subjects, captured in an indoor environment using a Nikon D850 camera. For each subject, we collected a 3 minute video in 1080p, which was then cropped and resized to 1024×1024 resolution.

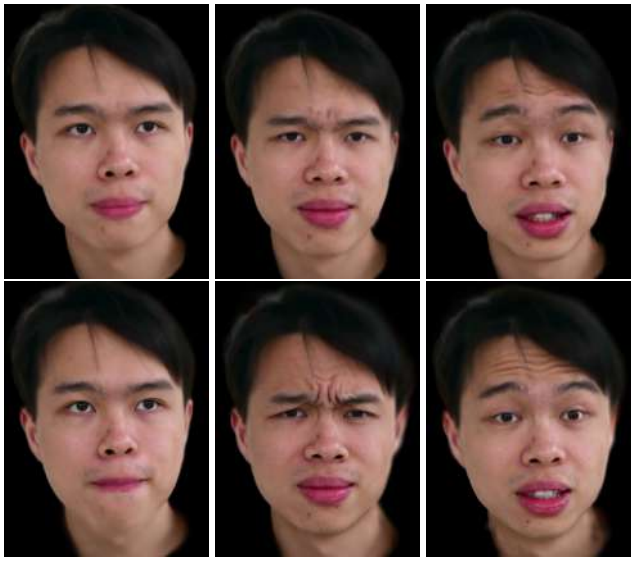


Figure 9: Ablation study on blendshape optimization. The first row shows the results with the initial values of $\{\Delta B_k\}$ kept unchanged during optimization. The second row shows our results with joint optimization of B_0, $\{\Delta B_k\}$, and B_m, which better capture intricate details of facial animations.

4.2 Comparisons

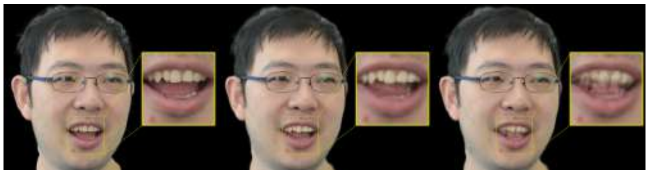
We evaluate the results using standard metrics including PSNR, SSIM and LPIPS [Zhang et al. 2018]. As show in the quantitative results Table 1, in most cases, our method outperforms INSTA and pointAvatar in terms of PSNR and LPIPS, while the SSIM of our method is consistently better. As shown in Table 2, our method also surpasses NeRFBlendShape in terms of PSNR and SSIM. Note that NeRFBlendShape utilizes the LPIPS loss during training, leading to better LPIPS. When we add the LPIPS loss with a weight of 0.05 in training, our method also performs better.

The qualitative comparisons are shown in Fig. 13 and Fig. 5. Compared with INSTA and PointAvatar, our method is better at capturing high-frequency details observed in the training video, such as wrinkles, teeth, and specular highlights of glasses and noses. Compared with NeRFBlendShape, our method also synthesizes images of higher quality with sharper details. Moreover, our method better recovers the eyeball movement than NeRFBlendShape, thanks to the eyeball motion control provided in FLAME.

Our method also performs better in novel view extrapolation (Fig. 6), while PointAvatar [Zheng et al. 2023] suffers from artifacts around the ear region, and both INSTA [Zielonka et al. 2023] and PointAvatar tend to lose high-frequency details.

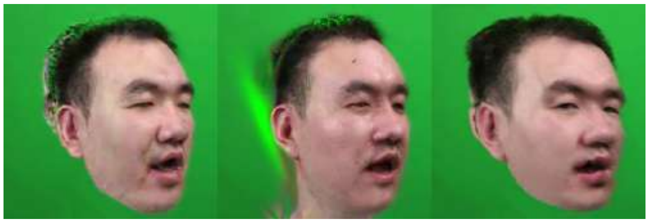
We show qualitative results on cross-identity reenactment in Fig. 7. Our method faithfully transfers expressions while maintaining the personal attributes of the target subject.

The training and runtime performance comparison is shown in Table 3. Our method is able to synthesize facial animations at 370fps, over 5× faster than INSTA and about 14× faster than NeRF-Blendshape. Our training time is comparable to NeRFBlendshape.



Ground truth Ours w/o mouth Gaussians

Figure 10: Ablation study on mouth interior Gaussians. We find that without the mouth interior Gaussians, the teeth may not be well modeled, leading to blurry or ghosting artifacts.



Ours FlashAvatar NeRFBlendShape

Figure 11: Failure cases of side view rendering.

4.3 Gaussian Blendshape Visualization

Fig. 12 demonstrates eight Gaussian blendshapes of a subject and their corresponding mesh blendshapes. Please refer to the supplementary video for live demonstration. The effect of our mouth Gaussians is show in Fig. 8.

4.4 Ablation Studies

Blendshape consistency. Fig. 3 visualizes the magnitudes of ΔM_k and ΔB_k. As you can see, imposing blendshape consistency during optimization does produce Gaussian blenshapes $\{B_k\}$ differing from the base model B_0 in a consistent way that mesh blendshapes $\{M_k\}$ differ from the base mesh M_0. Fig. 4 demonstrates the importance of blendshape consistency between Gaussian and mesh blendshapes. The optimization without considering blendshape consistency on all Gaussian properties results in apparent artifacts on the face and head boundary under novel expressions. Note that the magnitude of ΔB_k visualized in Fig. 3 only represents the difference between each individual blendshape B_k and the base model B_0, and thus does not necessarily correspond to errors in rendered images of the avatar model, which is the linear blending of all blendshapes.

Optimization of $\{\Delta B_k\}$. The initialization stage of our training constructs Gaussian blendshapes $\{B_k\}$ by transforming Gaussians of B_0 using the deformation gradients from M_0 to $\{M_k\}$, resulting in Gaussian differences $\{\Delta B_k\}$ consistent with mesh differences $\{\Delta M_k\}$. Keeping the initial values of $\{\Delta B_k\}$ unchanged during optimization and only optimizing B_0 and B_m can produce reasonable results, but fail to capture the fine details of facial animations, as shown in Fig. 9.

Mouth interior Gaussians. We evaluate the effect of mouth interior Gaussians by comparing our full result with the one using only the neutral model and expression blendshapes to represent the whole head (Fig. 10). We can see that apparent artifacts and blurring occur around the mouth region, demonstrating the necessity of mouth interior Gaussians.

5 CONCLUSION

We present a novel 3D Gaussian blendshape representation for animating photorealistic head avatars. We also introduce an optimization process to learn the Gaussian blendshapes from a monocular video, which are semantically consistent with their corresponding mesh blendshapes. Our method outperforms state-of-the-art NeRF and point based methods in producing avatar animations of superior quality at significantly faster speeds, while the training and memory cost is moderate.

Limitation and Discussion. Our constructed avatar models can exhibit apparent artifacts in side-view rendering if the training data does not contain side views. As shown in Fig. 11, this is also a limitation in previous NeRF-based methods and concurrent Gaussian-based methods. Improving the generalization capability to handle dramatically novel views is an open problem for further research. The extrapolation capabilities of our model are also restricted by its linear blending nature of the model, leading to potential failures when processing exaggerated expressions unseen in the training set. Another limitation is that the model cannot represent deformable hair, which is an interesting direction for future investigation. It is worth noting that there is a risk of misuse of our method (e.g., the so-called DeepFakes). We strongly oppose applying our work to produce fake images or videos of individuals with the intention of spreading false information or damaging their reputations.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (No. 2022YFF0902302), NSF China (No. 62172357 & 62322209), and the XPLORER PRIZE. The source code is available at https://gapszju.github.io/GaussianBlendshape.

REFERENCES

Ziqian Bai, Zhaopeng Cui, Xiaoming Liu, and Ping Tan. 2021. Riggable 3D Face Reconstruction via In-Network Optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021.* Computer Vision Foundation / IEEE, 6216–6225. https://doi.org/10.1109/CVPR46437.2021.00615

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, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, Warren N. Waggenspack (Ed.). ACM, 187–194. https://dl.acm.org/citation.cfm?id=311556

John C. Bowers, Rui Wang, Li-Yi Wei, and David Maletz. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Trans. Graph.* 29, 6 (2010), 166. https://doi.org/10.1145/1882261.1866188

Chen Cao, Qiming Hou, and Kun Zhou. 2014a. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.* 33, 4, Article 43 (jul 2014), 10 pages. https://doi.org/10.1145/2601097.2601204

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014b. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE Trans. Vis. Comput. Graph.* 20, 3 (2014), 413–425. https://doi.org/10.1109/TVCG.2013.249

Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2016. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.* 35, 4 (2016), 126:1–126:12. https://doi.org/10.1145/2897824.2925873

Bindita Chaudhuri, Noranart Vesdapunt, Linda G. Shapiro, and Baoyuan Wang. 2020. Personalized Face Modeling for Improved Face Reconstruction and Motion Retargeting. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 12350)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 142–160. https://doi.org/10.1007/978-3-030-58558-7_9

Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. 2023. MonoGaussianAvatar: Monocular Gaussian Point-based Head Avatar. arXiv:2312.04558 [cs.CV]

Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. 2023. HeadGaS: Real-Time Animatable Head Avatars via 3D Gaussian Splatting. arXiv:2312.02902 [cs.CV]

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Trans. Graph.* 40, 4 (2021), 88:1–88:13. https://doi.org/10.1145/3450626.3459936

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 8649–8658.

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. 2022. Reconstructing personalized semantic facial nerf models from monocular video. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–12.

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. *ACM Trans. Graph.* 35, 3 (2016), 28:1–28:15. https://doi.org/10.1145/2890493

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. 2022. Neural head avatars from monocular rgb videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 18653–18664.

Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022. Headnerf: A real-time nerf-based parametric head model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 20374–20384.

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017. Avatar digitization from a single image for real-time rendering. *ACM Trans. Graph.* 36, 6 (2017), 195:1–195:14. https://doi.org/10.1145/3130800.31310887

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. *ACM Trans. Graph.* 34, 4 (2015), 45:1–45:14. https://doi.org/10.1145/2766974

Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. 2022. SelfRecon: Self Reconstruction Your Digital Avatar from Monocular Video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022.* IEEE, 5595–5605. https://doi.org/10.1109/CVPR52688.2022.00552

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*, Sylvain Lefebvre and Michela Spagnuolo (Eds.). The Eurographics Association. https://doi.org/10.2312/egst.20141042

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph.* 36, 6 (2017), 194:1–194:17. https://doi.org/10.1145/3130800.3130813

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)* 40, 4 (2021), 1–13.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 405–421. https://doi.org/10.1007/978-3-030-58452-8_24

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–15.

Stylianos Ploumpis, Evangelos Ververas, Eimear O' Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick E. Pears, William A. P. Smith, Baris Gecer, and Stefanos Zafeiriou. 2021. Towards a Complete 3D Morphable Model of the Human Head. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 11 (2021), 4142–4160. https://doi.org/10.1109/TPAMI.2020.2991150

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. 2023. GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians. arXiv:2312.02069 [cs.CV]

Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. 2023. Relightable Gaussian Codec Avatars. arXiv:2312.03704 [cs.GR]

Ken Shoemake and Tom Duff. 1992. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface*, Vol. 92. 258–264.

Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)* 23, 3 (2004), 399–405.

Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Model. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 7346–7355. https://doi.org/10.1109/CVPR.2018.00767

Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. 2023. GaussianHead: High-fidelity Head Avatars with Learnable Gaussian Derivation. arXiv:arXiv:2312.01632 [cs.CV]

Yanlin Weng, Chen Cao, Qiming Hou, and Kun Zhou. 2014. Real-time facial animation on mobile devices. *Graphical Models* 76, 3 (2014), 172–179. https://doi.org/10.1016/j.gmod.2013.10.002 Computational Visual Media Conference 2013.

Jun Xiang, Xuan Gao, Yudong Guo, and Juyong Zhang. 2023. FlashAvatar: High-Fidelity Digital Avatar Rendering at 300FPS. arXiv:2312.02214 [cs.CV]

Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. 2022. Surface-aligned neural radiance fields for controllable 3d human synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15883–15892.

Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. 2023a. Gaussian Head Avatar: Ultra High-fidelity Head Avatar via Dynamic Gaussians. arXiv:2312.03029 [cs.CV]

Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. 2023b. Avatarmav: Fast 3d head avatar reconstruction using motion-aware neural voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.

Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Huang Han, Qi Guojun, and Yebin Liu. 2023c. LatentAvatar: Learning Latent Expression Code for Expressive Neural Head Avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 598–607. https://doi.org/10.1109/CVPR42600.2020.00068

Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. 2021. i3DMM: Deep Implicit 3D Morphable Model of Human Heads. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 12803–12813. https://doi.org/10.1109/CVPR46437.2021.01261

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

Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C Bühler, Xu Chen, Michael J Black, and Otmar Hilliges. 2022. Im avatar: Implicit morphable head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13545–13555.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. 2023. Pointavatar: Deformable point-based head avatars from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 21057–21067.

Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2022. Towards Metrical Reconstruction of Human Faces. In *European Conference on Computer Vision*. https://api.semanticscholar.org/CorpusID:248177832

Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant volumetric head avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4574–4584.

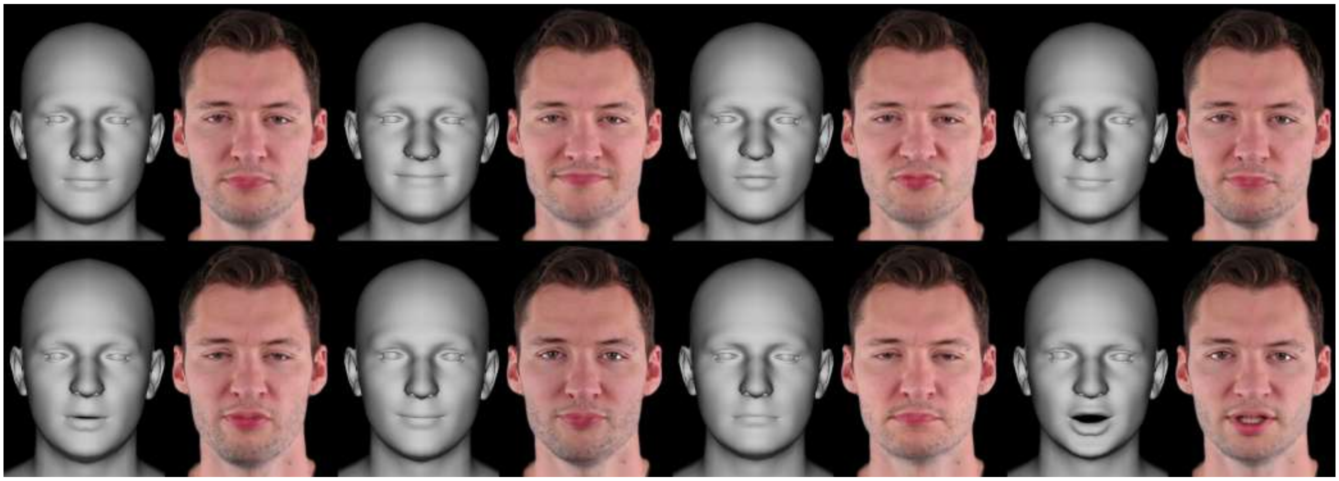


Figure 12: Visualization of our Gaussian blendshapes. Each Gaussian blendshape resembles its corresponding FLAME mesh blendshape, and captures photo-realistic appearance.

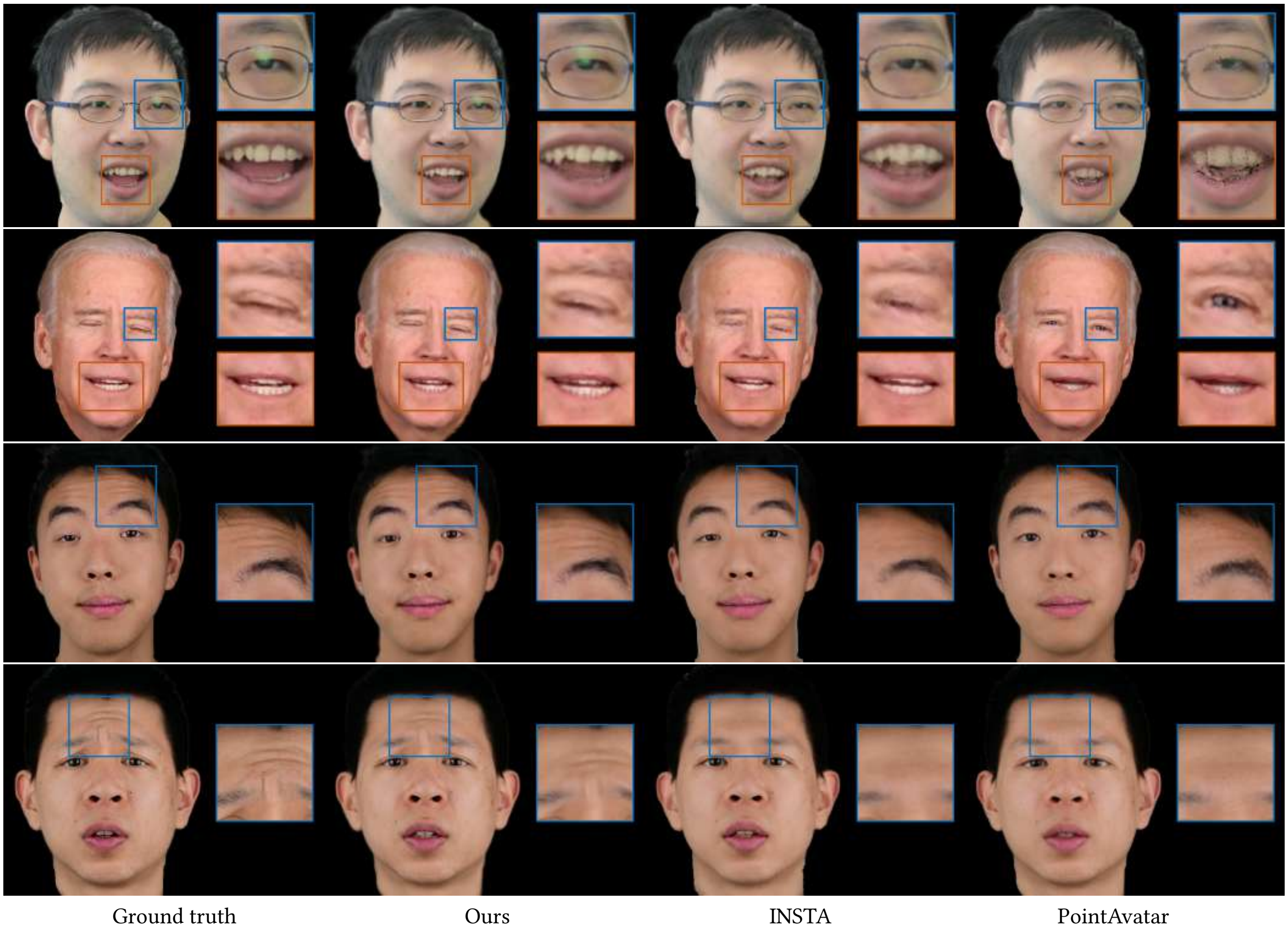


Figure 13: Qualitative comparisons between INSTA [Zielonka et al. 2023], PointAvatar [Zheng et al. 2023], and our method. Our method better captures high-frequency details and specular highlights. YouTube video ID is smghyezLW5o for Joe Biden.

头部虚拟形象动画的 3D 高斯混合形

马圣杰
浙江大学国家 CAD&CG 重点实验室
杭州，中国
qtdysjj@gmail.com
翁艳琳
浙江大学国家 CAD&CG 重点实验室
杭州，中国
weng@cad.zju.edu.cn
沙天佳
浙江大学国家 CAD&CG 重点实验室
杭州，中国
tjshao@zju.edu.cn
周军 *
国家重点实验室，浙江大学 CAD&CG
杭州，中国
kunzhou@acm.org

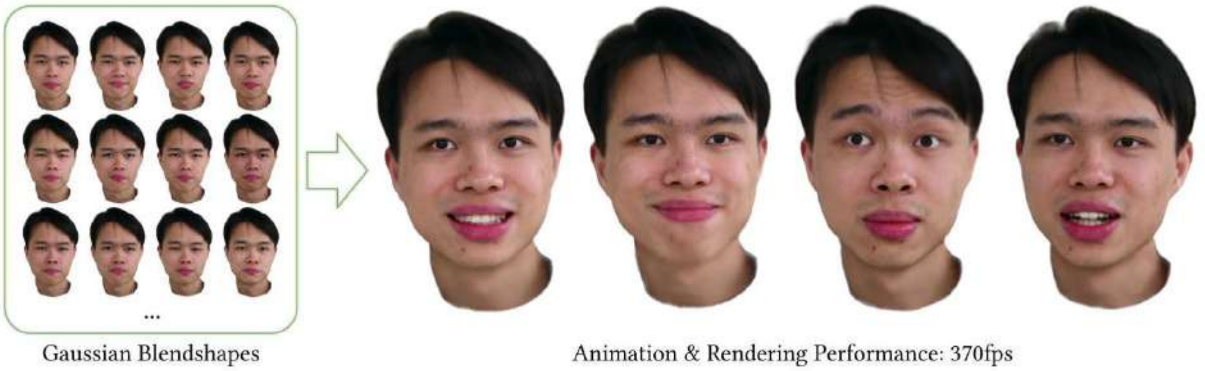


图 1: 我们的 3D 高斯混合形状类似于经典参数化人脸模型中的网格混合形状，可以通过表达式系数线性混合来实时 (370fps) 合成逼真的虚拟形象动画。

摘要

我们介绍了用于建模逼真头部虚拟形象的 3D 高斯混合形状。以单目视频作为输入，我们学习了一个中性表情的基础头部模型，以及一组表情混合形状，每个表情混合形状对应于经典参数化人脸模型中的一个基础表情。中性模型和表情混合形状都表示为 3D 高斯分布，包含几个描述虚拟形象外观的属性。通过将中性模型和表情混合形状通过高斯线性混合与表情系数相结合，可以有效地生成任意表情的虚拟形象模型。使用高斯散点法可以在实时中合成高保真的头部虚拟形象动画。与最先进的方法相比，我们的高斯混合形状表示更好地捕捉了输入视频中的高频细节，并实现了卓越的渲染性能。

CCS 概念

- 计算方法 → 重建；基于点的模型。

关键词

参数化人脸模型，面部动画，面部追踪，面部再现

ACM 参考格式:

马盛杰，翁艳琳，邵天家，周军。2024。3D 高斯混合形状用于头部虚拟形象动画。在计算机图形与交互技术特别兴趣小组会议论文集’24(SIGGRAPH 会议论文’24)，2024 年 7 月 27 日至 8 月 1 日，丹佛，科罗拉多州，美国。ACM，纽约州纽约市，美国，10 页。https://doi.org/10.1145/3641519.3657462

1 引言

重建和动画化三维人类头部一直是计算机图形学和计算机视觉领域长期研究的问题，它是远程呈现、虚拟现实/增强现实和电影等多种应用的关键技术。最近，基于神经辐射场 (NeRF)[Mildenhall 等人 2020] 的头像模型展示了在生成逼真图像方面的巨大潜力。这些技术通常通过将 NeRFs 与参数化头部模型 [Zielonka 等人 2023] 或表情码 [Gafni 等人 2021] 相结合来实现动态头像控制。Gao 等人 [2022] 和 Zheng 等人 [2022] 则提出构建一组 NeRF 混合形状，并通过线性混合它们来动画化头像。

混合形状模型是头像动画的经典表示方法。它由一组 3D 网格组成，每个网格对应一个基础表情。任意表情的面部形状可以通过线性混合基础网格与相应的表情系数高效计算。易于控制和高效的优势使得混合形状模型在专业动画制作 [Lewis 等人 2014] 以及消费者头像应用 (例如，iPhone Memoji)[Weng 等人 2014] 中成为最受欢迎的表示方法。

在本文中，我们介绍了一种用于构建和动画化头部虚拟形象的 3D 高斯混合形状表示法。我们在 3D 高斯散布 (3DGS)[Kerbl 等人 2023] 的基础上构建这种表示法，该方法将静态场景的辐射场表示为 3D 高斯分布，并在新视角合成中提供了引人注目的质量和速度。我们的表示法包括一个中性表情的基础模型和一组表情混合形状，所有这些都表示为 3D 高斯分布。每个高斯分布包含几个属性 (例如，位置、旋转和颜色)，与 3DGS 中的属性相同，并描绘了头部虚拟形象的外观。每个高斯混合形状对应于传统参数化面部模型 [曹等人 2014b；李等人 2017] 中的一个网格混合形状，并具有相同的语义含义。通过将高斯混合形状与表情系数混合，可以生成任意表情的高斯头部模型，使用高斯散布可以实时将其渲染为高保真图像。先前面部追踪算法 (例如，[曹等人 2014a；Zielonka 等人 2022]) 跟踪的运动参数可以用来驱动高斯混合形状，生成头部虚拟形象动画。

我们提议从单目视频中学习高斯混合形状表示。我们使用先前的方法从输入视频中构建网格混合形状，并在网格表面分布多个高斯函数作为初始化。然后我们共同优化所有高斯属性。由于高斯混合形状由与网格混合形状相同的表达系数驱动，因此每个高斯混合形状必须与其对应的网格混合形状在语义上保持一致，即高斯混合形状与中性模型之间的差异应与相应的网格混合形状与中性网格之间的差异保持一致。直接优化高斯属性而不考虑混合形状的一致性会导致对新训练中未见过的表情产生过拟合和伪影。为此，我们提出了一种有效的策略，指导高斯优化以满足一致性要求。具体来说，我们引入了一个中间变量，将高斯差异表示为与网格差异成比例的项。通过在训练过程中直接优化这个中间变量，我们产生了与中性模型不同的高斯混合形状，其差异方式与网格混合形状与中性网格的差异方式一致。

大量的实验表明，我们的高斯混合形状方法在合成高保真度头部虚拟角色动画方面优于现有最佳方法 [Gao et al. 2022; Zheng et al. 2023; Zielonka et al. 2023]，能够最好地捕捉输入视频中观察到的高频细节，并在虚拟角色动画和渲染速度上取得了显著提升 (见图 1)。

2 相关工作

研究人员提出了各种头部虚拟形象的表示方法。早期工作采用显式的 3D 网格表示，从图像中重建 3D 形状和外观。开创性工作 [Blanz 和 Vetter 1999] 提出了 3D Morphable Model (3DMM)，用于在低维线性子空间中建模面部形状和纹理。沿着这个方向有许多后续工作，例如全身头部模型 [Ploumpis 等人 2021]，以及深度非线性模型 [Tran 和 Liu 2018]。3D 网格表示也用于构建可绑定的头部，用于头部动画 [Bai 等人 2021; Chaudhuri 等人 2020; Hu 等人 2017]。为了生成详细的动画，研究人员进一步提出了基于图像的动态虚拟形象，控制整个头部包括头发和头饰 [Cao 等人 2016]，或者额外重建细粒度修正 [Feng 等人 2021; Garrido 等人 2016; Ichim 等人 2015; Yang 等人 2020]。

为了实现高度逼真的渲染效果，近期方法使用神经辐射场 (NeRF)[Mildenhall et al. 2020] 隐式地表示头部虚拟形象，并取得了令人印象深刻的结果 [Gafni et al. 2021; Grassal et al. 2022; Jiang et al. 2022; Lombardi et al. 2021; Xu et al. 2022, 2023b, c; Zheng et al. 2022]。例如，i3DMM [Yenamandra et al. 2021] 提出了基于完整头部三维形变模型的第一个神经隐式函数。Head-Nerf [Hong et al. 2022] 引入了一种基于 NeRF 的参数化头部模型，将神经辐射场整合到头部的参数化表示中。最先进的工作 INSTA [Zielonka et al. 2023] 基于 Instant-NGP [Müller et al. 2022] 构建了一个围绕参数化面部模型的动态神经辐射场。它能在不到 10 分钟内重建一个头部虚拟形象。PointA-vatar [Zheng et al. 2023] 提出了一个基于点的表示方法，并基于

* 通讯作者

FLAME 的表达向量学习一个形变场来驱动点。NeRFBlendshape [Gao et al. 2022] 通过结合多层次体素场和表达式系数构建了基于 NeRF 的混合形变模型，用于语义动画控制和高保真渲染。

许多并发工作已被提出，以应用 [Kerbl et al. 2023] 引入的 3D 高斯表示来构建头像模型 (例如，[Chen et al. 2023; Dhamo et al. 2023; Qian et al. 2023; Saito et al. 2023; Wang et al. 2023; Xiang et al. 2023; Xu et al. 2023a])。它们中的大多数将 3D 高斯表示与神经网络结合使用。例如，GaussianHead [Wang et al. 2023] 使用多层感知器 (MLPs) 解码高斯动态几何和辐射参数。FlashAvatar [Xiang et al. 2023] 在具有可学习偏移的网格上附加高斯，这些偏移以 MLPs 表示。Saito et al. [2023] 通过使用网络解码 3D 高斯参数和学习辐射传输函数来构建可重光照的头像模型。据我们所知，目前没有并发工作引入如我们论文中的高斯混合形 (Gaussian blendshapes) 概念。我们方法的独特优势在于，它只需要线性混合高斯混合形来构建具有任意表情的头像模型，这在训练和运行时性能上带来了显著的好处。在性能方面与我们的工作最接近的方法是 FlashAvatar [Xiang et al. 2023]，它使用 10k 高斯实现了 300fps，而对于 $\sim$ 100fps 高斯降低到 50k fps，而我们的方法对于 370fps 高斯实现了 70k fps。

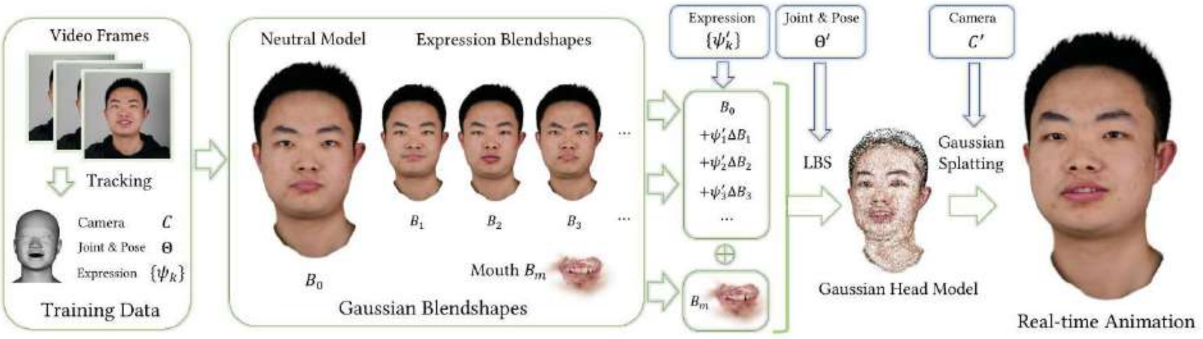


图 2: 我们的方法概述。输入为单目视频，我们的方法学习了一个头部虚拟形象的高斯混合形状表示，该表示包括一个中性模型 B_0，一组表情混合形状 $\{B_1, B_2, \ldots, B_K\}$，以及口腔内部模型 B_m，所有这些都表示为 3D 高斯分布。通过使用表情系数 $\{\psi'_k\}$ 进行线性混合以及使用关节和姿态参数 Θ' 进行线性混合蒙皮，可以生成任意表情和姿态的虚拟形象模型，我们使用高斯散布在实时中渲染高保真图像。

3 方法

3.1 3D 高斯混合形状

我们的高斯混合形状表示包括一个中性基础模型 B_0 和一组表情混合形状 $\{B_1, B_2, \ldots, B_K\}$，所有这些都表示为一组 3D 高斯分布，每一个都具有几个基本属性 (即，位置 $\mathbf{x}$，不透明度 α，旋转 $\mathbf{q}$，缩放 $\mathbf{s}$ 和球谐系数 SH) 如 3DGS [Kerbl 等人 2023] 中所述。每个 B_0 的高斯分布还拥有一组用于关节和姿态控制的混合权重 $\mathbf{w}$。B_0 的每个高斯分布与每个混合形状 B_k 之间都有一一对应关系。B_k 与 B_0 的偏差可以定义为它们高斯属性之间的差异 $\Delta B_k = B_k - B_0$。任意表情的头部虚拟形象模型计算如下:

$$B^{\psi} = B_0 + \sum_{k=1}^{K} \psi_k \Delta B_k \tag{1}$$

其中 $\{\psi_k\}$ 是表情系数。

目前我们使用基于主成分分析 (PCA) 的混合形状模型 FLAME [Li et al. 2017]，尽管也可以使用其他肌肉启发的混合形状，例如基于面部动作编码系统 (FACS) 的模型 FaceWarehouse [Cao et al. 2014b]。除了面部表情控制，FLAME 还提供了关节和姿态参数 Θ，用于控制头部、下巴、眼球和眼睑的运动，这些参数与线性混合蒙皮 (LBS) 一起使用，以转换头像模型 (即其高斯分布): $B^{\psi *} = LBS\left(B^{\psi}, \Theta\right)$，其中与 B_0 的高斯分布相关的混合权重被使用。

口腔内部高斯分布。口腔内部和头发的运动通常不受面部表情的影响，因此不在 FLAME 网格的覆盖范围内，上述混合形状模型也未涉及。头发可以随头部刚体移动，而牙齿的运动则由 FLAME 中的下巴关节控制。我们发现，在实际操作中，我们在训练中生成的混合形状高斯分布能够很好地模拟头发，但口腔内部的结果不够理想。因此，我们为口腔内部定义了一套单独的高斯分布 B_m，这些高斯分布随 FLAME 中的下巴关节移动。这些口腔高斯分布的性质不会随表情改变，而只随下巴关节转换，即 $B_m^* = \mathrm{LBS}\left(B_m, \Theta\right)$。

最终，转换后的高斯模型 $\left(B^{\psi *}, B_m^*\right)$ 可以使用高斯散布技术在实时中渲染成高保真图像 I_r。图 2 展示了我们方法的概述。

3.2 训练

数据准备。遵循 [Zielonka 等人 2023], 我们使用 [Zielonka 等人 2022] 的面部追踪器计算中性表情和 $K = 50$ 基础表情的 FLAME 网格，以及每个视频帧的相机参数、关节和姿态参数、表情系数。我们还提取了每个输入帧的前景头部遮罩。

初始化。我们首先初始化中性模型 B_0 、表情混合形状 $\{B_k\}$ 以及口内高斯分布 B_m 。对于 B_0 ，我们在中性 FLAME 网格 M_0 上使用泊松盘采样 [Bowers 等人 2010] 分布若干点，并将它们作为高斯位置的初始化。其他高斯属性则按照 3DGS [Kerbl 等人 2023] 的方式进行初始化。对于每个高斯，我们还找到它最近的三角形 M_0 ，并计算其线性混合权重 (LBS blend weights) 作为三角形顶点混合权重的线性插值。为了初始化口内高斯 B_m ，我们使用两个预定义的布告板来表示上牙和下牙，这些布告板使用泊松盘采样转换为高斯分布。上牙高斯被刚性地绑定到头部的后部，而下牙高斯则绑定到颚关节具有最大蒙皮权重的顶点。

为了初始化表达式混合形状 B_k ，我们使用变形梯度 [Sumner and Popović 2004] 从 M_0 转换每个高斯分布 B_0 到表达式 FLAME 网格 M_k 。具体来说，对于每个中性高斯分布 G_0^i ，我们计算从 M_0 上最近的三角形到 M_k 上相应三角形的仿射变换，并提取旋转分量 [Shoemake and Duff 1992]，该旋转分量应用于 G_0^i 的位置、旋转和球谐函数 (SH) 系数，以产生表达式混合形状 B_k 的相应高斯分布 G_k^i 。注意，我们省略了缩放分量，因为发现该变换非常接近刚体变换。G_k^i 的缩放和透明度属性保持与 G_0^i 相同。这样，我们可以从 B_0 构建每个表达式混合形状 B_k ，以及它们的差异 $\Delta B_k = B_k - B_0$ 。

优化。初始化后，我们联合优化 $B_0, \{\Delta B_k\}$ 和 B_m 。对于每个视频帧，我们通过线性混合 B_0 和 $\{\Delta B_k\}$ 并根据方程 (1) 中跟踪到的表达式系数重建高斯人头模型 B_ψ ，然后使用 LBS 和跟踪到的关节和姿态参数变换 B_ψ 和 B_m : $B^{\psi *} = \text{LBS}\left(B^\psi, \Theta\right), B_m^* = \text{LBS}\left(B_m, \Theta\right)$ 。最后，我们使用高斯散点法从 $B^{\psi *}$ 和 B_m^* 获得渲染图像。优化过程与 3DGS [Kerbl et al. 2023] 类似，也涉及添加和删除高斯分布的适应性密度控制步骤。

在优化过程中，避免过拟合的一个关键点是保持每个高斯混合形状 B_k 与其对应的网格混合形状 M_k 之间的语义一致性。如前所述，高斯混合形状使用基于 FLAME 参数网格模型的相同跟踪表情系数在训练和运行时计算中进行混合。为了确保这种混合计算的语义有效性，B_k 与 B_0 之间的差异 (即 ΔB_k) 必须与 M_k 与 M_0 之间的差异 (即 ΔM_k) 一致，这意味着在头部区域，M_k 与 M_0 之间的顶点位置差异较大时，B_k 与 B_0 之间的高斯差异也应较大，反之则较小。如果不考虑这种一致性直接优化 $\{\Delta B_k\}$ ，将导致过拟合，在训练图像中未见的新的表情系数上容易出现明显的伪影 (见图 4 的示例)。

然而，与仅包含顶点位置位移的 ΔM_k 不同，ΔB_k 包含不同类型的属性，例如位置、旋转和颜色。因此，很难设计一个损失函数项来明确强制 ΔB_k 与 ΔM_k 之间的一致性，同时不牺牲图像损失。相反，我们提出了一种简单而有效的策略来指导高斯优化隐式地遵循一致性要求。具体来说，对于每个高斯 G_i ，设 $\Delta G_{i,k}$ 为其在 B_k 和 B_0 中属性的差异。我们引入一个中间变量 $\Delta \widehat{G}_{i,k}$ ，将 $\Delta G_{i,k}$ 表示为:

$$\Delta G_{i,k} = \Delta G_{i,k}^{\text{init}} + \max\left(f\left(d_{i,k}\right), 0\right) \Delta \widehat{G}_{i,k}, \tag{2}$$

其中 $\Delta G_{i,k}^{\text{init}}$ 是在上述初始化阶段计算得到的 $\Delta G_{i,k}$ 的初始值，并在优化过程中被视为常数，而 $d_{i,k}$ 是距离 G_i 最近的表面点的位置位移量，从 M_0 到 M_k 。线性函数 $f(x) = (x - \epsilon) / \left(\widetilde{d} - \epsilon\right)$ 将 M_k 和 M_0 之间的最大位置差异 $\widetilde{d}$ 缩放到 1，并将阈值量 $\epsilon = 0.00001$ 缩放到 0。max 函数是必要的，以避免位置位移量低于 ϵ 时的负缩放值。

公式 (2) 本质上表示实际的高斯差 $\Delta G_{i,k}$ 作为其初始值与根据网格混合形变中相应位置位移缩放后的 $\Delta \widehat{G}_{i,k}$ 之和，这有效地将高斯差与位置位移关联起来。初始高斯差 $\Delta G_{i,k}^{\text{init}}$ 与网格混合形变的位置位移成比例，因为它是通过从 M_0 到 M_k 的变形梯度计算得到的。根据位置位移缩放 $\Delta \widehat{G}_{i,k}$ 确保高斯差 $\Delta G_{i,k}$ 以与位置位移成比例的速率更新。请注意，对于位置位移量低于 ϵ 的高斯函数，公式 (2) 中的第二项将消失为 0，$\Delta G_{i,k}$ 总是等于 $\Delta G_{i,k}^{\text{init}}$ ，这本质上与位置位移成比例。

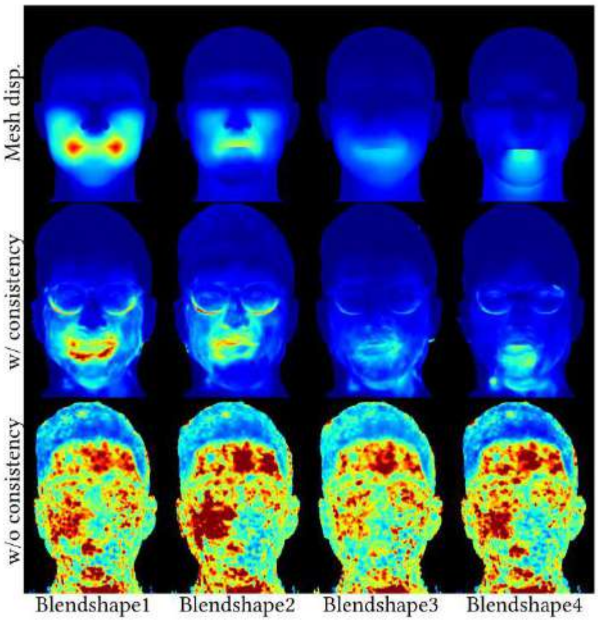


图 3: 混合形变一致性对表情混合形变优化的影响。第一行显示了 M_k 和 M_0 之间的位移量。第二行和第三行分别显示了有无混合形变一致性时优化的 ΔB_k 的量级。

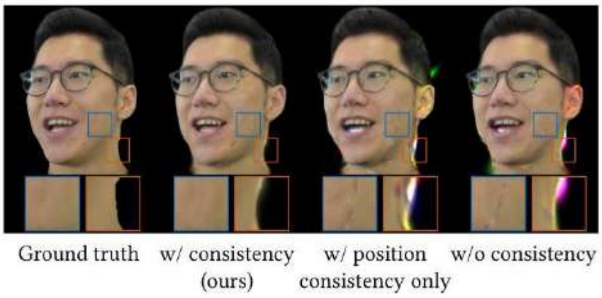


图 4: 关于混合形状一致性的消融研究。在不考虑混合形状一致性的优化过程中，会导致明显的伪影，如内部和边界区域的颜色污点和故障。仅在高斯位置上强制混合形状一致性也会导致不良结果。

表 1:INSTA [Zielonka 等人 2023]、PointAvatar [Zheng 等人 2023] 与我们方法之间的定量比较。

数据集		INSTA 数据集								我们的数据集			
		bala	biden	justin	malte_1	marcel	nf 01	nf 03	wojtek_1	subject1	subject2	subject3	subject4
峰值信噪比↑	INSTA	28.66	28.38	29.74	26.27	23.75	25.89	26.10	29.84	28.88	28.16	28.60	30.83
	PointAvatar	29.60	31.72	32.31	27.46	24.60	28.34	29.82	31.94	31.43	32.57	30.95	32.57
	我们的	33.34	32.48	32.49	28.56	26.61	27.92	28.62	32.39	32.87	32.55	31.54	34.03
SSIM↑	INSTA	0.9130	0.9484	0.9530	0.9262	0.9133	0.9246	0.9129	0.9457	0.9195	0.9443	0.9078	0.9445
	PointAvatar	0.9099	0.9565	0.9595	0.9225	0.9121	0.9278	0.9208	0.9502	0.9219	0.9463	0.9062	0.9433
	我们的	0.9490	0.9672	0.9696	0.9461	0.9348	0.9448	0.9381	0.9645	0.9384	0.9577	0.9268	0.9646
LPIPS J	INSTA	0.0817	0.0545	0.0614	0.0751	0.1540	0.1285	0.1137	0.0588	0.1536	0.1208	0.1733	0.1144
	PointAvatar	0.0821	0.0535	0.0649	0.0718	0.1574	0.1350	0.1221	0.0661	0.1568	0.1190	0.1715	0.1285
	我们的	0.0772	0.0522	0.0631	0.0703	0.1391	0.1174	0.0965	0.0595	0.1520	0.1197	0.1689	0.1075

我们不是优化 $\Delta G_{i,k}$，而是直接使用以下章节描述的损失函数优化 $\Delta \widehat{G}_{i,k}$。具体来说，$\Delta \widehat{G}_{i,k}$ 初始化为 0。每次更新 $\Delta \widehat{G}_{i,k}$ 时，我们根据公式 (2) 计算 $\Delta G_{i,k}$，从而构建虚拟形象模型。然后通过高斯散点绘制将虚拟形象模型渲染成图像，该图像用于损失函数的计算。

这样，我们有效地引导高斯差异随位置位移一致变化，从而得到与网格混合形状具有强烈语义一致性的优化高斯混合形状 (见图 3)。

3.3 损失函数

优化目标是减小渲染图像与输入之间的图像损失，同时满足一些正则化约束。第一个损失是图像损失，与 3DGS [Kerbl 等人 2023] 中的损失类似，包括渲染图像与视频帧之间的 L_1 差异和 D-SSIM 项:

$$L_{rgb} = (1-\lambda)\, L_1 + \lambda L_{D-SSIM} \tag{3}$$

其中 $\lambda = 0.2$ 。

我们还设计了一个 alpha 损失，以限制高斯分布保持在头部区域内。我们执行高斯散点绘制以获得累积不透明度图像 I_α ，并将其与前景头部掩码 Mask_h 进行比较。alpha 损失定义为:

$$L_\alpha = \frac{1}{F}\sum_{i=1}^{F}\left(\left\|\left(I_\alpha^i - \text{Mask}_h^i\right)\right\|_2\right) \tag{4}$$

其中 F 是帧编号。

我们进一步引入了一个正则化损失，用以约束口内高斯分布保持在预先定义的口腔体积内。具体来说，我们计算了每个高斯分布到体积边界的符号距离，并在其超出体积时应用 L_2 损失以将其回缩。正则化损失定义为:

$$L_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\| \max\left(SDF\left(\mathbf{x}_i, V\right), 0\right)\right\|_2^2\right), \tag{5}$$

其中 V 是预先定义的圆柱体积，$\mathbf{x}_i$ 是高斯分布的位置，N 是口内高斯分布的数量。整体损失函数定义为:

$$L = \lambda_1 L_{rgb} + \lambda_2 L_\alpha + \lambda_3 L_{reg}. \tag{6}$$

表 2:NeRFBlend-Shape [Gao et al. 2022] 与我们方法之间的定量比较。

数据集		NeRFBlendShape 数据集					
		id1	id2	id3	id4	id5	id6
PSNR 上箭头	NeRFBlendShape	32.12	32.25	37.25	36.86	34.26	35.74
	我们的 (不含 LPIPS)	33.13	33.23	39.83	38.34	35.58	36.59
	我们的 (含 LPIPS)	33.09	33.15	39.64	38.17	35.46	36.46
SSIM 上箭头	NeRFBlendShape	0.9412	0.9369	0.9750	0.9791	0.9541	0.9786
	我们的 (不含 LPIPS)	0.9532	0.9473	0.9836	0.9846	0.9635	0.9814
	我们的 (含 LPIPS)	0.9522	0.9457	0.9828	0.9837	0.9625	0.9806
LPIPS ↓	NeRFBlendShape	0.0715	0.0756	0.0436	0.0460	0.0448	0.0366
	我们的 (不含 LPIPS)	0.0862	0.0937	0.0486	0.0495	0.0538	0.0414
	我们的 (含 LPIPS)	0.0550	0.0620	0.0359	0.0334	0.0381	0.0303

我们默认设置 $\lambda_1 = 1, \lambda_2 = 10, \lambda_3 = 100$ 。

3.4 实施细节

我们使用 Pytorch 实现了我们的方法。采用 Adam 优化器 [Kingma and Ba 2015] 进行参数优化。学习率分别为 $3.2 \times 10^{-7}, 5 \times 10^{-5}, 5 \times 10^{-4}, 1 \times 10^{-4}, 1.25 \times 10^{-3}$ 用于高斯属性 $\{\mathbf{x}_k, \alpha_k, \mathbf{s}_k, \mathbf{q}_k, SH_k\}$ 。初始采样的高斯数量为 50k 用于中性模型，14k 用于口内高斯分布。

训练在 A800 GPU 上进行，测试在 RTX 4090 GPU 上进行。我们还根据 3DGS [Kerbl et al. 2023] 构建了一个 C++/CUDA 交互式查看器，并使用它来测量我们的运行时帧率。

由于高斯位置在优化过程中频繁变化，我们需要有效地更新它们的 LBS 混合权重 $\mathbf{w}$ 以及最近点的位置位移 $\{d_{i,k}\}$ 。我们在中性网格 $256 \times 256 \times 256$ 周围的 3D 网格上预计算并存储这些值。任意高斯值可以有效地计算为距离高斯中心最近的八个网格点值的线性混合。

4 结果

4.1 基线和数据集

我们将我们的方法与最先进的方法进行了比较，包括基于 NeRF 的 INSTA [Zielonka 等人 2023] 和基于点的 PointAvatar [Zheng 等人 2023]，在 INSTA 数据集和我们自己的数据集上。我们像 INSTA 一样，将每个视频的最后 350 帧作为自我再现的测试集。对于 INSTA 中的 4500 帧，训练数据的准备时间大约是 12 小时。我们还在他们的公共数据集上与 NeRFBlendShape [Gao 等人 2022] 进行了比较，该数据集由八个视频组成。每个视频的最后 500 帧被保留用于测试。我们将 alpha 权重 λ_2 降低到整个过程中的 1

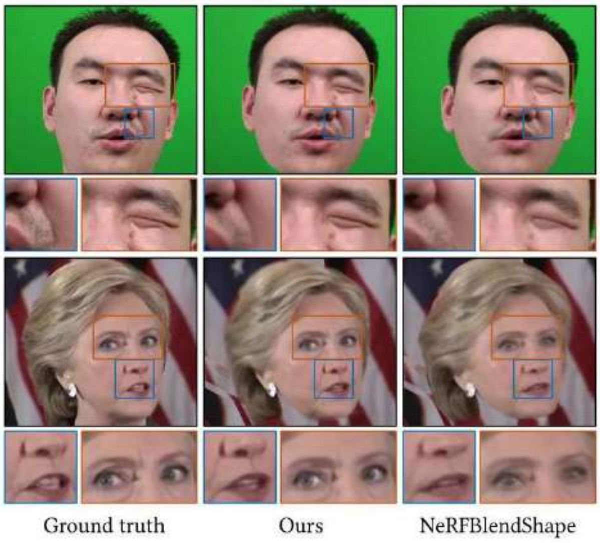


图 5: 与 NeRFBlend-Shape [Gao 等人 2022] 的定性比较。我们的方法更忠实地捕捉到了细微的面部细节 (例如，眼睛和鼻子周围的皱纹)，并更好地恢复了眼球运动。YouTube 视频 ID 为-yHgE9W699w，是 Hillary Clinton 的视频。

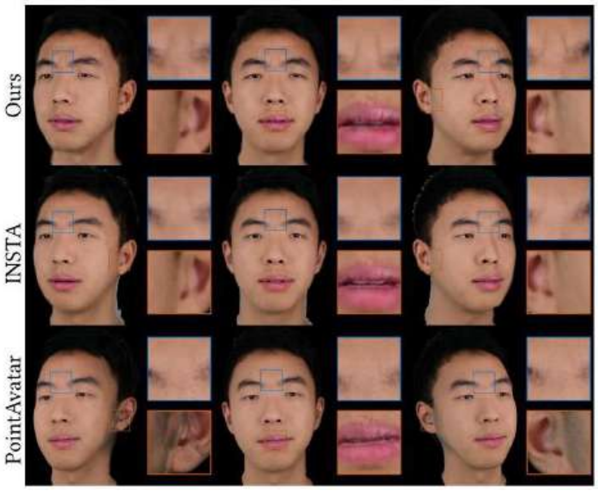


图 6: 新颖视角外推的定性比较。我们的方法在新视角下产生了具有更细微细节的更好结果。

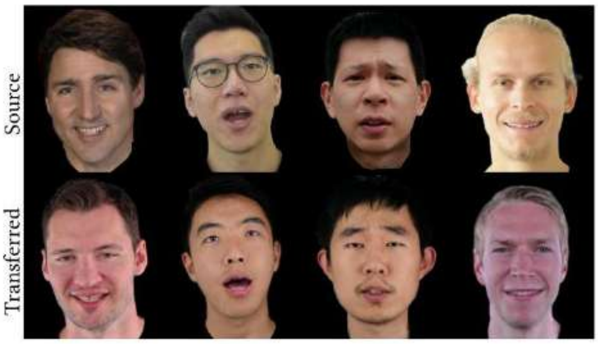


图 7: 跨身份再现的结果。YouTube 视频 ID 为 mKHgXHKbJUE，是 Justin Trudeau 的视频。

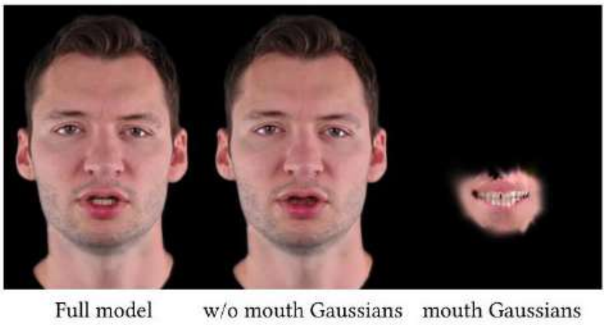


图 8: 展示嘴部内部高斯分布。

表 3: 性能比较。我们所有的算法都在 A800 GPU 上进行训练。INSTA、NeRFBlendshape 和我们的方法在 RTX 4090 GPU 上进行测试，但由于 RTX 4090 GPU 内存不足，PointAvatar 则在 A800 GPU 上进行测试。渲染分辨率为 512×512 。请注意，我们的运行时间包括动画 (即线性混合和 LBS 转换) 和渲染，我们的性能对渲染分辨率不敏感。我们还报告了训练和运行时计算期间的最大内存消耗。由于提供的二值前景掩码相对不准确，这导致数据集上的性能略有下降，我们发现这略微锐化了轮廓周围的头发。

	训练	运行时	内存 (训练)	记忆 (运行时)
INSTA	10 分钟	70 帧/秒	16G	4G
PointAvatar	3.5 h	5 帧/秒 *	40G	32G*
NeRFBlendShape	20 分钟	26 帧/秒	7G	2G
我们的	25 分钟	370 帧/秒	14G	2G

我们自己的数据集由四个主题组成，在室内环境中使用尼康 D850 相机捕获。对于每个主题，我们收集了一段 3 分钟的视频 1080p ，然后将其裁剪并调整大小到 1024×1024 分辨率。

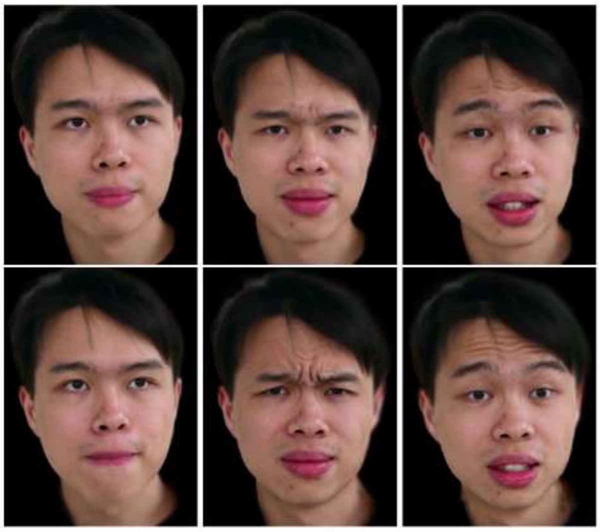


图 9: 关于混合形状优化的消融研究。第一行显示了在优化过程中保持 $\{\Delta B_k\}$ 初始值不变的结果。第二行显示了我们对 $B_0, \{\Delta B_k\}$ 和 B_m 进行联合优化的结果，这更好地捕捉到了面部动画的复杂细节。

4.2 比较

我们使用标准指标, 包括 PSNR、SSIM 和 LPIPS [Zhang et al. 2018] 来评估结果。如表 1 的定量结果所示, 在大多数情况下，我们的方法在 PSNR 和 LPIPS 方面优于 INSTA 和 pointAvatar，而我们的方法的 SSIM 始终更佳。如表 2 所示，我们的方法在 PSNR 和 SSIM 方面也超过了 NeRFBlendShape。注意，NeRFBlendShape 在训练过程中使用了 LPIPS 损失, 因此具有更好的 LPIPS 表现。当我们在训练中加入权重为 0.05 的 LPIPS 损失时，我们的方法也表现更佳。

定性比较结果展示在图 13 和图 5 中。与 INSTA 和 PointAvatar 相比，我们的方法在捕捉训练视频中观察到的高频细节方面表现更好，如皱纹、牙齿以及眼镜和鼻子的镜面高光。与 NeRFBlendShape 相比，我

们的方法也能合成具有更清晰细节的高质量图像。此外，由于 FLAME 中提供的眼球运动控制，我们的方法在恢复眼球运动方面比 NeRFBlendShape 做得更好。

我们的方法在新型视角外推方面也表现更好 (图 6)，而 PointAvatar [Zheng et al. 2023] 在耳朵区域周围存在伪影，INSTA [Zielonka et al. 2023] 和 PointAvatar 都倾向于丢失高频细节。

我们在图 7 中展示了跨身份再现的定性结果。我们的方法在保持目标主体个人特征的同时，忠实地转移表情。

训练和运行时性能比较如表 3 所示。我们的方法能够合成面部动画 370fps，比 INSTA 快 $5\times$，比 NeRF-Blendshape 快约 $14\times$。我们的训练时间与 NeRFBlendshape 相当。

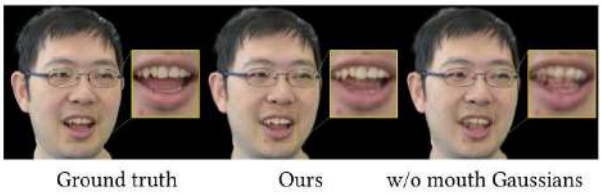


图 10: 关于口内高斯分布的消融研究。我们发现，如果没有口内高斯分布，牙齿可能无法很好地建模，导致模糊或重影伪迹。

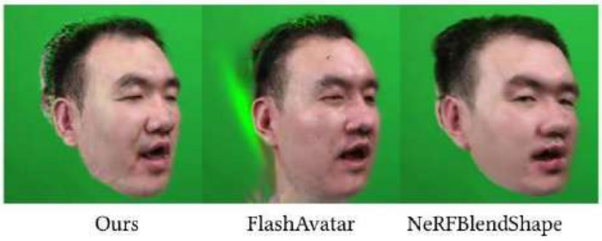


图 11: 侧面渲染的失败案例。

4.3 高斯融合形状可视化

图 12 展示了某个体八个高斯融合形状及其对应的网格融合形状。请参考补充视频以获得实时演示。我们的口高斯效果在图 8 中展示。

4.4 消融研究

融合形状一致性。图 3 可视化了 ΔM_k 和 ΔB_k 的大小。如你所见，在优化过程中强加融合形状一致性确实会产生与基础模型 B_0 以一致方式不同的高斯融合形状 $\{B_k\}$，就像网格融合形状 $\{M_k\}$ 与基础网格 M_0 的差异一样。图 4 展示了高斯融合形状与网格融合形状之间一致性的重要性。在所有高斯属性上不考虑融合形状一致性的优化会在新颖表情下的面部和头部边界产生明显的伪迹。注意，图 3 中可视化的 ΔB_k 的大小仅代表每个单独融合形状 B_k 与基础模型 B_0 之间的差异，因此不一定对应于虚拟人模型渲染图像中的错误，这是所有融合形状的线性混合。

优化 $\{\Delta B_k\}$。我们的训练初始化阶段通过使用来自 M_0 到 $\{M_k\}$ 的变形梯度转换 B_0 的正态分布来构造正态混合形状 $\{B_k\}$，产生与网格差异 $\{\Delta M_k\}$ 一致的正态差异 $\{\Delta B_k\}$。在优化过程中保持 $\{\Delta B_k\}$ 的初始值不变，仅优化 B_0 和 B_m 可以产生合理的结果，但无法捕捉面部动画的细微细节，如图 9 所示。

口腔内部正态分布。我们通过比较使用完整结果和使用仅中性模型以及表情混合形状来表示整个头部的结果 (图 10)，评估口腔内部正态分布的效果。我们可以看到口腔区域周围出现了明显的伪影和模糊，证明了口腔内部正态分布的必要性。

5 结论

我们提出了一种新颖的 3D 正态混合形状表示方法，用于动画化逼真的头部虚拟形象。我们还引入了一种优化过程，从单目视频中学习正态混合形状，这些混合形状在语义上与其对应的网格混合形状一致。我

们的方法在生成高质量头像动画方面优于最先进的 NeRF 和基于点的技术，速度显著更快，同时训练和内存成本适中。

限制与讨论。我们构建的虚拟形象模型如果在训练数据中不包含侧视图，则在侧视渲染中会表现出明显的伪影。如图 11 所示，这也是之前基于 NeRF 的方法和并发的高斯方法的一个限制。提高处理剧新型视角的泛化能力是一个有待进一步研究的开放性问题。我们的模型的推断能力也受到其模型线性混合特性的限制，当处理训练集中未见的夸张表情时可能会导致潜在的失败。另一个限制是模型无法表示可变形的头发，这是未来研究的一个有趣方向。值得注意的是，我们的方法存在被滥用的风险 (例如所谓的深度伪造)。我们强烈反对将我们的工作应用于制作意图传播虚假信息或损害个人声誉的虚假图像或视频。

致谢

本工作得到中国国家重点研发计划 (编号 2022YFF0902302)、中国国家自然科学基金 (编号 62172357 & 62322209) 以及 XPLORER PRIZE 的支持。源代码可在 https://gapszju.github.io/GaussianBlendshape 获取。

参考文献

Ziqian Bai, Zhaopeng Cui, Xiaoming Liu, and Ping Tan. 2021. Riggable 3D Face Reconstruction via In-Network Optimization. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 6216-6225. https://doi.org/10.1109/CVPR46437.2021.00615

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, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999, Warren N. Waggenspack (Ed.). ACM, 187-194. https://dl.acm.org/citation.cfm?id=311556

John C. Bowers, Rui Wang, Li-Yi Wei, and David Maletz. 2010. Parallel Poisson disk sampling with spectrum analysis on surfaces. ACM Trans. Graph. 29, 6 (2010), 166.

Chen Cao, Qiming Hou, and Kun Zhou. 2014a. Displaced dynamic expression regression for real-time facial tracking and animation. ACM Trans. Graph. 33, 4, Article 43 (jul 2014), 10 pages. https://doi.org/10.1145/2601097.2601204

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014b. FaceWare-house: A 3D Facial Expression Database for Visual Computing. IEEE Trans. Vis. Comput. Graph. 20, 3 (2014), 413-425. https://doi.org/10.1109/TVCG.2013.249

Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. 2016. Real-time facial animation with image-based dynamic avatars. ACM Trans. Graph. 35, 4 (2016),

Bindita Chaudhuri, Noranart Vesdapunt, Linda G. Shapiro, and Baoyuan Wang. 2020. Personalized Face Modeling for Improved Face Reconstruction and Motion Re-targeting. In Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V (Lecture Notes in Computer Science, Vol. 12350), Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 142-160. https://doi.org/10.1007/978-3-030-58558-7_9

Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun

Helisa Dhamo, Yinyu Nie, Arthur Moreau, Jifei Song, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. 2023. HeadGaS: Real-Time Animatable Head Avatars via 3D Gaussian Splatting. arXiv:2312.02902 [cs.CV]

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. ACM Trans. Graph. 40, 4 (2021), 88:1-88:13. https://doi.org/10.1145/3450626.3459936

Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 8649-8658.

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. 2022. Reconstructing personalized semantic facial nerf models from monocular video. ACM Transactions on Graphics (TOG) 41, 6 (2022), 1-12.

Pablo Garrido, Michael Zollhöfer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Pérez, and Christian Theobalt. 2016. Reconstruction of Personalized 3D Face Rigs from Monocular Video. ACM Trans. Graph. 35, 3 (2016), 28:1-28:15. https: //doi.org/10.1145/2890493 and Justus Thies. 2022. Neural head avatars from monocular rgb videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 18653-18664.

Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022. Headnerf: A real-time nerf-based parametric head model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 20374-20384.

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017. Avatar digitization from a https://doi.org/10.1145/3130800.31310887

Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. 2015. Dynamic 3D avatar creation from hand-held video input. ACM Trans. Graph. 34, 4 (2015), 45:1-45:14. https://doi.org/10.1145/2766974

Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. 2022. SelfRecon: Self Reconstruction Your Digital Avatar from Monocular Video. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. IEEE, 5595-5605. https://doi.org/10.1109/CVPR52688.2022.00552

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics 42, 4 (July 2023). https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980 J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In Eurographics 2014 - State of the Art Reports, Sylvain Lefebvre and Michela Spagnuolo (Eds.). The Eurographics Association. https://doi.org/10.2312/egst.20141042

Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. ACM Trans. Graph. 36, 6 (2017), 194:1-194:17. https://doi.org/10.1145/3130800.3130813

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. ACM Transactions on Graphics (ToG) 40, 4 (2021), 1-13.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ra-mamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346), Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 405-421. https://doi.org/10.1007/978-3-030-58452-8_24

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) 41, 4 (2022), 1-15.

Stylianos Ploumpis, Evangelos Ververas, Eimear O' Sullivan, Stylianos Moschoglou, Haoyang Wang, Nick E. Pears, William A. P. Smith, Baris Gecer, and Stefanos Zafeiriou. 2021. Towards a Complete 3D Morphable Model of the Human Head. IEEE Trans. Pattern Anal. Mach. Intell. 43, 11 (2021), 4142-4160. https://doi.org/10.1109/TPAMI.2020.299115

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. 2023. GaussianAvatars: Photorealistic Head Avatars with

Shunsuke Saito, Gabriel Schwartz, Tomas Simon, Junxuan Li, and Giljoo Nam. 2023. Relightable Gaussian Codec Avatars. arXiv:2312.03704 [cs.GR]

Ken Shoemake and Tom Duff. 1992. Matrix animation and polar decomposition. In Proceedings of the conference on Graphics interface, Vol. 92. 258-264.

Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes.

ACM Transactions on Graphics (TOG) 23, 3 (2004), 399-405。

Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Model. In 2018 IEEE 7346-7355. https://doi.org/10.1109/CVPR.2018.00767

Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. 2023. GaussianHead: High-fidelity Head Avatars with Learnable Gaussian Derivation. arXiv:arXiv:2312.01632 [cs.CV]

Yanlin Weng, Chen Cao, Qiming Hou, and Kun Zhou. 2014. Real-time facial animation on mobile devices. Graphical Models 76, 3 (2014), 172-179. https://doi.org/10.1016/ j.gmod.2013.10.002 Computational Visual Media Conference 2013.

Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. 2022. Surface-aligned neural radiance fields for controllable 3 d human synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 15883-15892.

Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. 2023a. Gaussian Head Avatar: Ultra High-fidelity Head Avatar via Dynamic Gaussians. arXiv:2312.03029 [cs.CV]

Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. 2023b. In ACM SIGGRAPH 2023 Conference Proceedings. 1-10.

Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Huang Han, Qi Guojun, and Yebin Liu. 2023c. LatentAvatar: Learning Latent Expression Code for Expressive Neural Head Avatar. In ACM SIGGRAPH 2023 Conference Proceedings.

Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. 2020. FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed

Riggable 3D Face Prediction. In 2020 IEEE/CVF Conference on Computer Vision and

Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. 2021. i3DMM: Deep Implicit 3D Morphable Model of Human Heads. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 12803-12813. https://doi.org/10.1109/CVPR46437.2021.01261

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings Black, and Otmar Hilliges. 2022. Im avatar: Implicit morphable head avatars from videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 13545-13555.

Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J Black, and Otmar Hilliges. 2023. Pointavatar: Deformable point-based head avatars from videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 21057- struction of Human Faces. In European Conference on Computer Vision. https: //api.semanticscholar.org/CorpusID:248177832

Wojciech Zielonka, Timo Bolkart, and Justus Thies. 2023. Instant volumetric head avatars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 4574-4584.

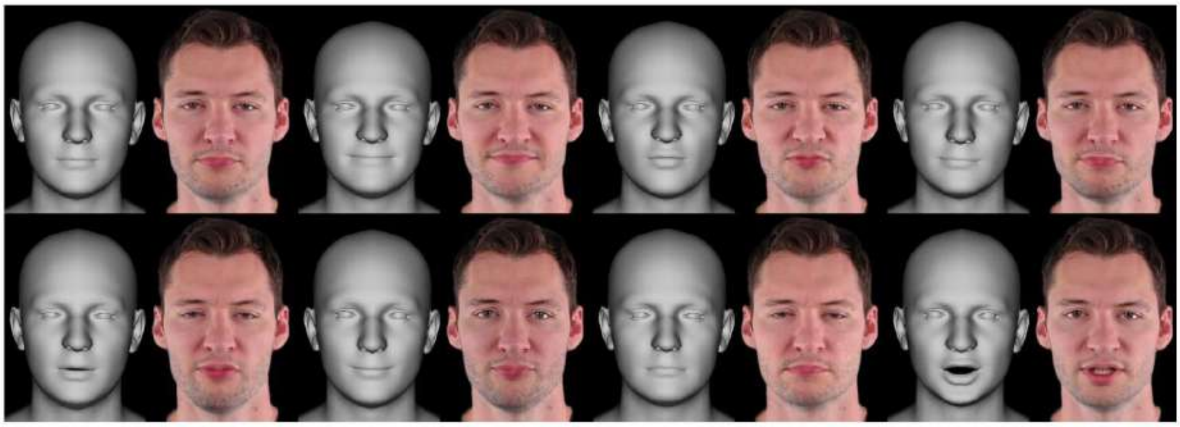


Figure 12: Visualization of our Gaussian blendshapes. Each Gaussian blendshape resembles its corresponding FLAME mesh blendshape, and captures photo-realistic appearance.

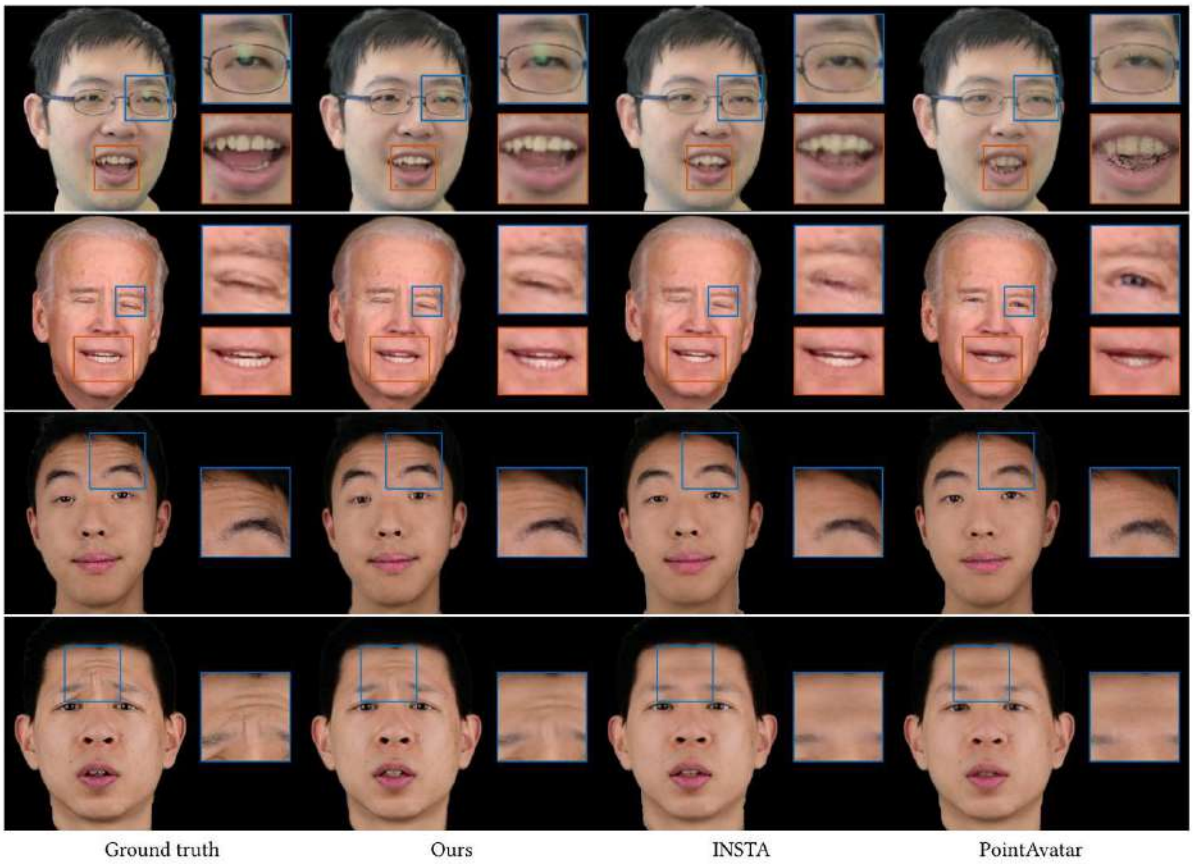


图 13:INSTA [Zielonka et al. 2023]、PointAvatar [Zheng et al. 2023] 和我们方法之间的定性比较。我们的方法更好地捕捉了高频细节和镜面高光。YouTube 视频 ID 为 smghyezLW5o 的 Joe Biden。