A APPENDIX

A.1 Geometry Losses for Head Motions

Similarly, we apply losses $\mathcal{L}_{\text{head\_ang}}$, $\mathcal{L}_{\text{head\_vel}}$, and $\mathcal{L}_{\text{head\_smooth}}$ to the head pose component $\boldsymbol{p}$ of the pose parameter $\boldsymbol{\theta}$ to constrain head movements:

$$\mathcal{L}_{\text{head\_ang}} = \left\| \boldsymbol{P}_{-T_p:T_w} - \hat{\boldsymbol{P}}_{-T_p:T_w} \right\|^2, \tag{1}$$

$$\mathcal{L}_{\text{head\_vel}} = \left\| \left(\boldsymbol{P}_{-T_p+1:T_w} - \boldsymbol{P}_{-T_p:T_w-1} \right) - \left(\hat{\boldsymbol{P}}_{-T_p+1:T_w} - \hat{\boldsymbol{P}}_{-T_p:T_w-1} \right) \right\|^2, \tag{2}$$

$$\mathcal{L}_{\text{head\_smooth}} = \left\| \hat{\boldsymbol{P}}_{-T_p+2:T_w} - 2\hat{\boldsymbol{P}}_{-T_p+1:T_w-1} + \hat{\boldsymbol{P}}_{-T_p:T_w-2} \right\|^2. \tag{3}$$

Furthermore, we discover that constraining the velocity and acceleration of head movement at the start of the current window to match those at the end of the previous window helps with generating smooth transition and prevents abrupt changes in head posture. Thus, we define the pose sequence during the transition as $\bar{\boldsymbol{P}}_{-3:3} = \{\boldsymbol{P}_{-3:0}, \hat{\boldsymbol{P}}_{0:3}\}$ and the transition loss $\mathcal{L}_{\text{trans}}$ as:

$$\begin{aligned} \mathcal{L}_{\text{head\_trans}} = &\left\| \left(\bar{\boldsymbol{P}}_{0:2} - \bar{\boldsymbol{P}}_{-1:1} \right) - \left(\bar{\boldsymbol{P}}_{-1:1} - \bar{\boldsymbol{P}}_{-2:0} \right) \right\|^2 + \\ &\left\| \left(\bar{\boldsymbol{P}}_{0:3} - 2\bar{\boldsymbol{P}}_{-1:2} + \bar{\boldsymbol{P}}_{-2:1} \right) - \left(\bar{\boldsymbol{P}}_{-1:2} - 2\bar{\boldsymbol{P}}_{-2:1} + \bar{\boldsymbol{P}}_{-3:0} \right) \right\|^2. \end{aligned} \tag{4}$$

The overall head loss is expressed as:

$$\begin{aligned} \mathcal{L}_{\text{head}} = &\lambda_{\text{head\_ang}} \mathcal{L}_{\text{head\_ang}} + \lambda_{\text{head\_vel}} \mathcal{L}_{\text{head\_vel}} \\ &+ \lambda_{\text{head\_smooth}} \mathcal{L}_{\text{head\_smooth}} + \lambda_{\text{head\_trans}} \mathcal{L}_{\text{head\_trans}}. \end{aligned} \tag{5}$$

During training, the weights are set as: $\lambda_{\text{head\_ang}} = 0.05$, $\lambda_{\text{head\_vel}} = 5$, $\lambda_{\text{head\_smooth}} = 0.5$, and $\lambda_{\text{head\_trans}} = 0.5$.

A.2 Inference Speed

When conducting inference with an Intel Xeon Gold 5218R CPU and a Nvidia 3090 GPU, DiffPoseTalk can generate the motion parameters at 30 FPS. Using the style encoder to extract a style feature from a reference segment takes only 2ms.

For live streaming scenarios, our system introduces a 7.33-second delay. This delay is primarily due to our windowing strategy, which involves an initial 4 seconds wait to fill the first window and an additional 3.33 seconds for processing. However, our inference does not introduce any further latency once this initial processing is complete.

A.3 Style and Content Disentanglement

To examine the disentanglement of our generation's style and content, we randomly selected 10 speaking styles and 20 audio clips to generate 10×20 animations. The speaking styles from these animations are then extracted using our style encoder. We employ t-SNE [Van der Maaten and Hinton 2008] for visualizing these extracted speaking styles. The result in Figure 1 demonstrates that animations with identical reference styles are clustered together, regardless of the content differences.

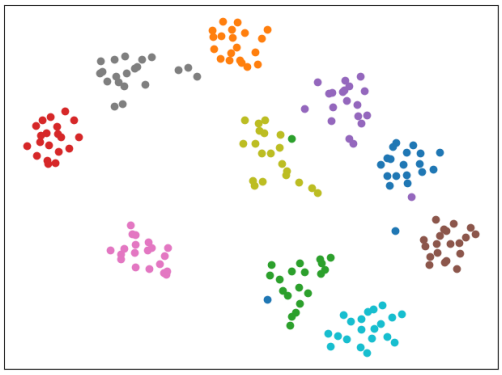


Fig. 1. Visualization of the disentanglement of style and content. Colors indicate different styles.

A.4 Integration with Rigged Avatars

We demonstrate that our method is capable of generating 3DMM parameters that effectively drive rigged avatars. Specifically, we adopt GaussianAvatars [Qian et al. 2023], which employs 3D Gaussian Splatting [Kerbl et al. 2023] to create photorealistic head avatars that can be animated using the underlying FLAME model. For this study, we select two avatars and animate them using two distinct motion sequences. To improve visual quality and minimize artifacts, we incorporate GFPGAN [Wang et al. 2021] for face restoration as a post-processing step. The results of these animations are depicted in Figure 2 and further demonstrated in the demo video.

A.5 User Study

We provide screenshots of the user study system and samples in Figure 3.

REFERENCES

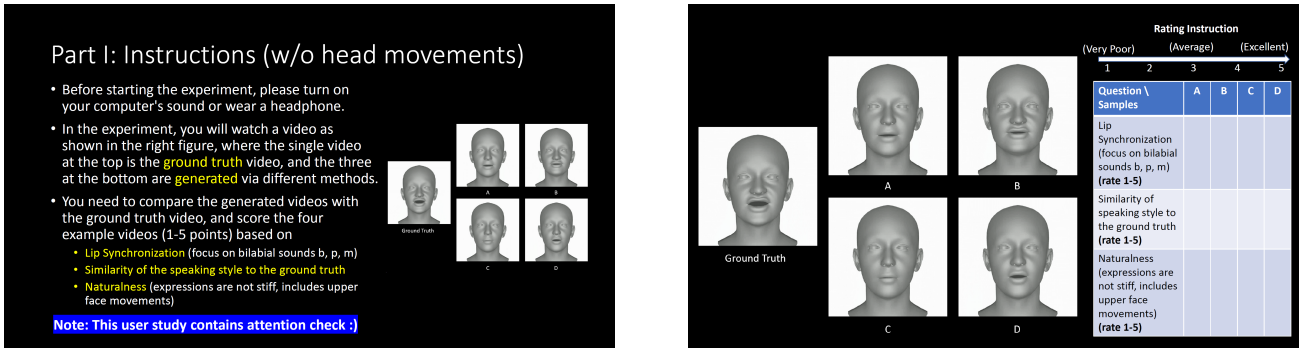
Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (2023), 139:1–139:14.

Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. 2023. GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians. *CoRR* abs/2312.02069 (2023), 13 pages.

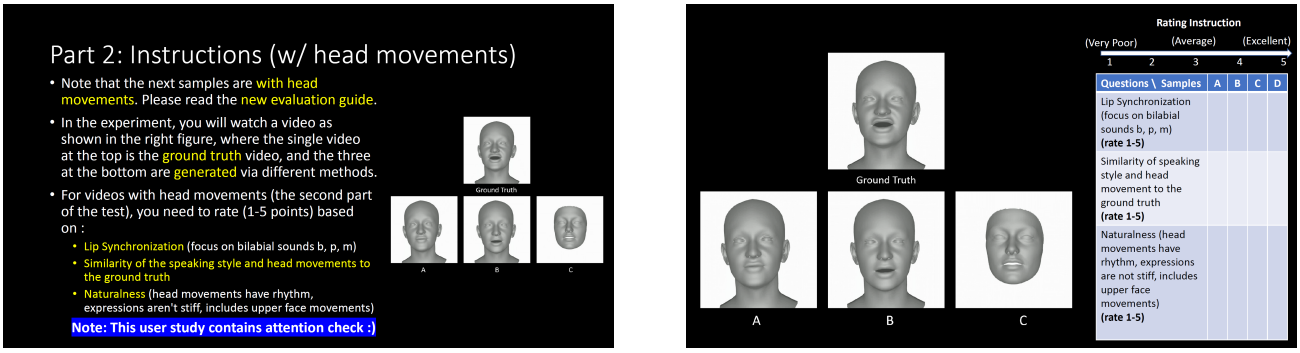
Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008), 2579–2605.

Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. 2021. Towards Real-World Blind Face Restoration With Generative Facial Prior. In *CVPR*. Computer Vision Foundation / IEEE, 9168–9178.

Fig. 2. Our method can drive rigged avatars to create photorealistic videos.



(a)



(b)

Fig. 3. Screenshots of the user study system and samples.