

RePerformer: Immersive Human-centric Volumetric Videos from Playback to Photoreal Reperformance

Supplementary Material

Within the supplementary material, we provide:

- Implementation details in Appendix Sec. A.
- Qualitative and Quantitative comparison with Animatable Gaussians [38] in Appendix Sec. B.
- Additional ablative studies in Appendix Sec. C.

A. Implementation Details

Initialization. We initialize approximately 50,000 motion Gaussians in the canonical frame using a uniform random distribution. During subsequent optimization, aside from the photometric loss in Eq. 1, the isotropic loss and size loss are defined as follows:

$$E_{\text{iso}} = \frac{1}{N} \sum_{i=1}^N \text{ReLU}(e^{\max(s_i) - \min(s_i)} - r),$$

$$E_{\text{size}} = \sum_{i=1}^N \text{ReLU}\left(s_i - \alpha \frac{1}{N} \sum_{i=1}^N sg[s_i]\right), \quad (9)$$

where s_i represents the scaling parameters of the i -th Gaussian, N denotes the number of Gaussians, and e is the exponential activation function. The isotropic loss ensures the ratio between the major and minor axes of each Gaussian does not exceed r (set to 4 in our experiments). The sg denotes stop-gradient operator. For appearance Gaussians, we use the initialized motion Gaussians as input and further optimize it with densification and prune as the original 3DGS [30].

Network Architecture. We use three identical U-Net architectures, adjusting the output feature dimensions to accommodate different attributes. Each U-Net incorporates a self-attention layer to maintain global consistency. As illustrated in Fig. 10, the self-attention layer is applied before the final downsampling step of the U-Net.

Re-performance. During the alignment phase, we assign semantic labels to specific regions, including the head, left hand, right hand, left foot, and right foot, to align corresponding clusters between the source and target Gaussians. For objects, we use pairwise semantic prompts (e.g., “basketball” and “balloon”) and assign consistent labels during the unprojection process, ensuring accurate alignment across different objects. The re-performance stage employs the Adam optimizer, with the alignment phase trained for 15,000 iterations and the motion transfer phase for 2,000 iterations.

Our method efficiently generates Gaussian sequences for high-fidelity playback and vivid re-performance of gen-

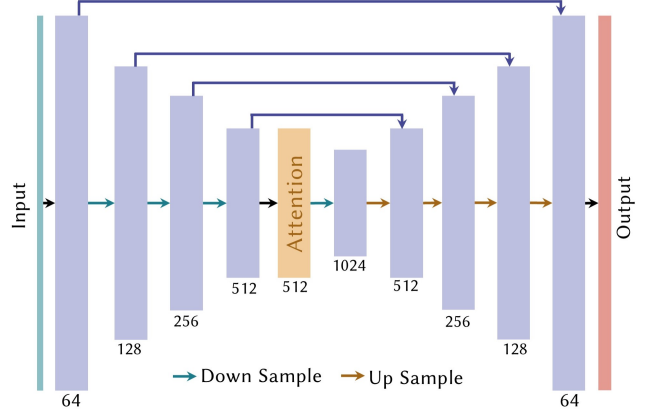


Figure 10. Network architecture.

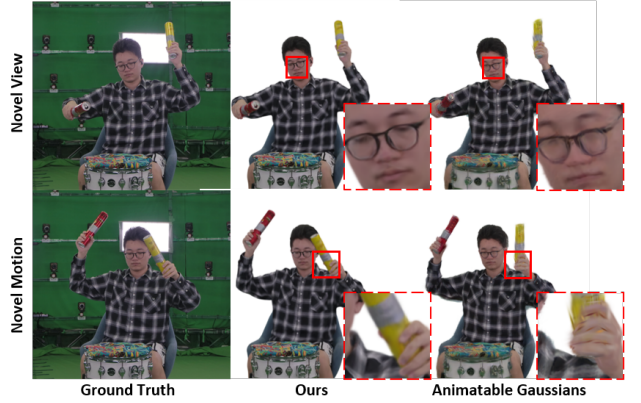


Figure 11. Qualitative comparison with Animatable Gaussians [38].

eral non-rigid scenes. We train the model using the PyTorch framework on a single NVIDIA GeForce RTX3090 GPU, achieving a rendering speed of 7 FPS. The Gaussian sequences can be further baked and compressed via DualGS [27], allowing seamless integration into low-end devices like VR headsets and iPads for intuitive, user-friendly interaction as demonstrated in Fig. 12.

B. Comparison

We further compare our method with Animatable Gaussians [38] to evaluate its rendering quality on novel views and motions. As shown in Fig. 11 and Tab. 3, this method relies heavily on the human parametric model SMPL [44], which constrains its ability to accurately estimate motion in regions far from the human body. This limitation results in severe artifacts when handling general non-rigid scenarios.

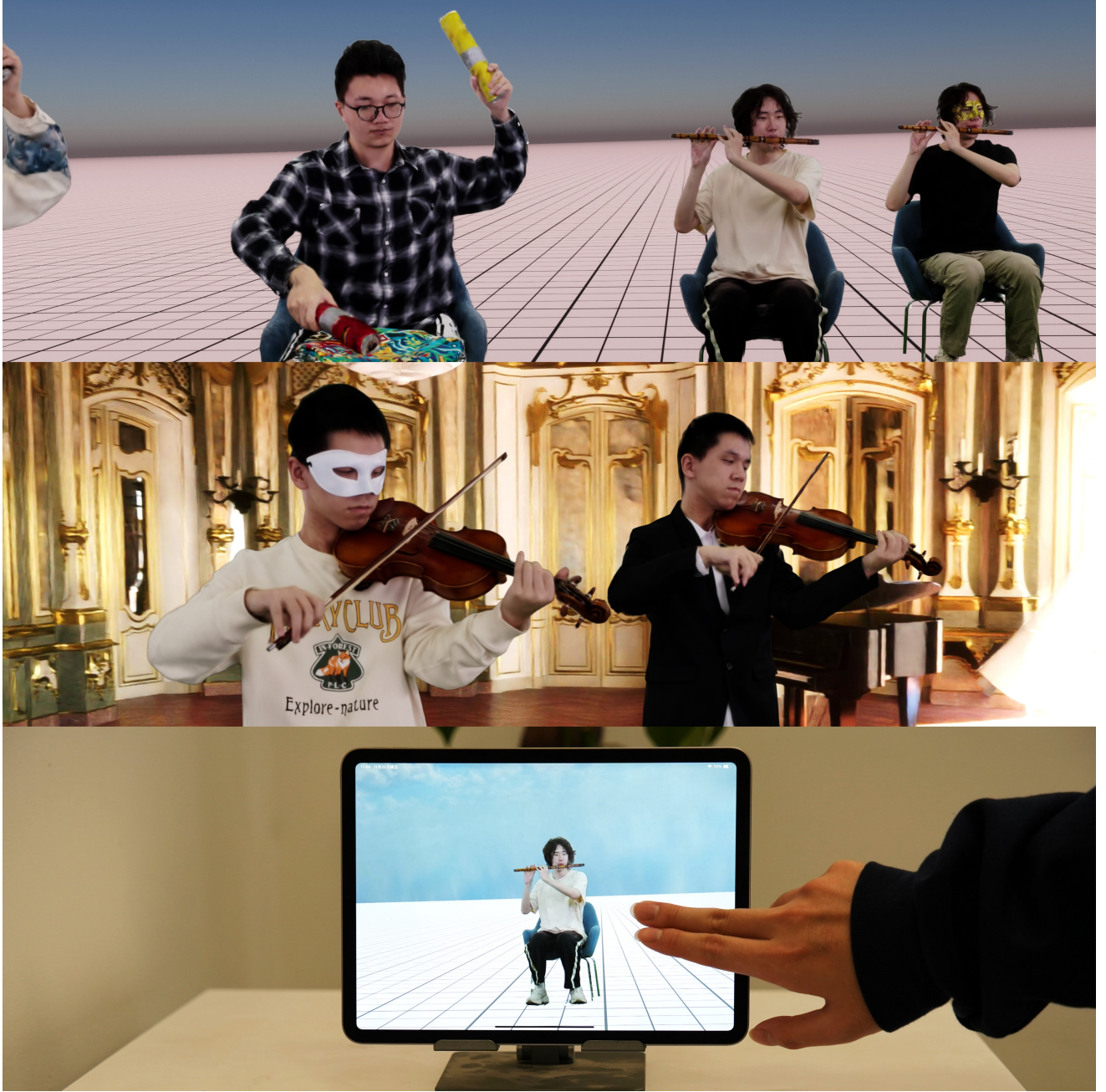


Figure 12. We further compress our Gaussian sequences using DualGS, demonstrating their adaptability in various immersive applications.

| Methods | Novel View | | | Novel Motion | | |
|----------------------|-----------------|-----------------|--------------------|-----------------|-----------------|--------------------|
| | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow |
| Animatable Gaussians | 24.26 | 0.934 | 0.0647 | 22.04 | 0.911 | 0.079 |
| Ours | 32.09 | 0.979 | 0.0310 | 30.06 | 0.976 | 0.0277 |

Table 3. Quantitative comparison with Animatable Gaussians [38].

C. Ablations

Number of Camera Views. To assess the robustness of RePerformer under sparser input views, we perform ablation experiments using uniformly selected subsets of 20, 40, and 60 camera views for training, denoted as w/20.cam, w/40.cam, and w/60.cam. As shown in Tab. 4, our method maintains satisfactory rendering quality even with sparser input views.

Position Map Resolution. As illustrated in Tab. 5, we con-

| Methods | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow |
|-------------|-----------------|-----------------|--------------------|
| w/20.cam | 30.36 | 0.971 | 0.0471 |
| w/40.cam | 30.89 | 0.975 | 0.0447 |
| w/60.cam | 31.33 | 0.977 | 0.0429 |
| Ours | 32.09 | 0.979 | 0.0310 |

Table 4. Quantitative Ablation Study on the different input view numbers.

| Methods | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow |
|-------------|-----------------|-----------------|--------------------|
| w/ 64.res | 28.85 | 0.949 | 0.0887 |
| w/ 128.res | 32.82 | 0.981 | 0.0413 |
| w/ 256.res | 33.45 | 0.986 | 0.0283 |
| w/o atten | 34.16 | 0.990 | 0.0197 |
| Ours | 34.27 | 0.989 | 0.0227 |

Table 5. Quantitative Ablation Study on the resolution of position maps.

duct ablation studies with position map resolutions of 64, 128, 256, and 512 (ours), corresponding to different numbers of Gaussians. Although larger position maps can store denser Gaussians, they exceed GPU memory limitations. We also compare results without the self-attention layer. Our full pipeline achieves high-fidelity rendering quality at a resolution of 512 with the self-attention module, as shown in Tab. 5.