RigGS: Rigging of 3D Gaussians for Modeling Articulated Objects in Videos (Supplementary Material)

Yuxin Yao[1] Zhi Deng[2] Junhui Hou[1]*
[1]City University of Hong Kong [2]University of Science and Technology of China
yuxinyao@cityu.edu.hk zhideng@mail.ustc.edu.cn jh.hou@cityu.edu.hk

1. More results

More Results for Editing. SC-GS [3] introduces sparse control points and ARAP regularization, allowing them to complete editing tasks by fixing some control points and moving other points. As shown in Fig. 1, using SC-GS to edit the object is not easily controllable. For example, we want only to edit the arm, but the leg is also moving. Furthermore, even by adding more fixed points, achieving reasonable edits of movements is still challenging. As demonstrated in Figs. 2 and 3, we also compared the extracted skeletons, skinning weights, and editing performance of our RigGS with those of AP-NeRF [7] on more examples. Due to the different canonical shapes established by the AP-NeRF and ours, the poses of the skeletons / skinning weights are different. We aligned the skeletons generated by two methods to the same pose to create new poses for the objects. It can be observed that our method generally yields more reasonable results with higher clarity. More edited animations are shown in the submitted **Video Demo**.

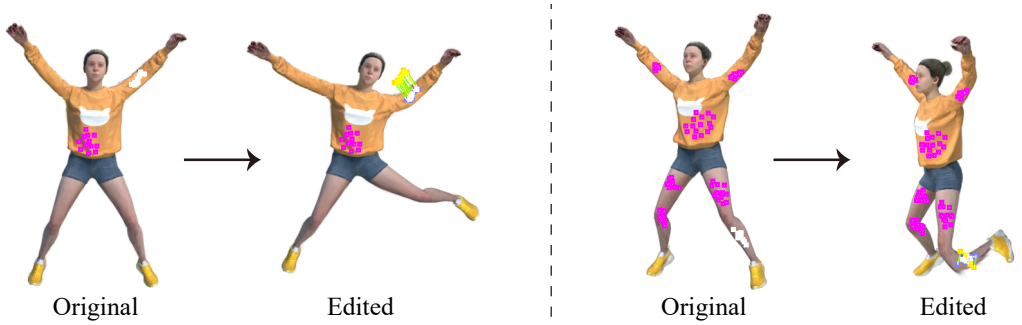

Figure 1. Editing by SC-GS [3]. The pink dots mark the fixed locations, while the white dots indicate the positions to be edited. The green lines represent the trajectories of the edits.

More Results for Novel View Synthesis. We present the numerical results for each sequence in Table 1 and Table 2. Except for SC-GS [3], we can see our rendering quality is significantly better than that

*Corresponding author.

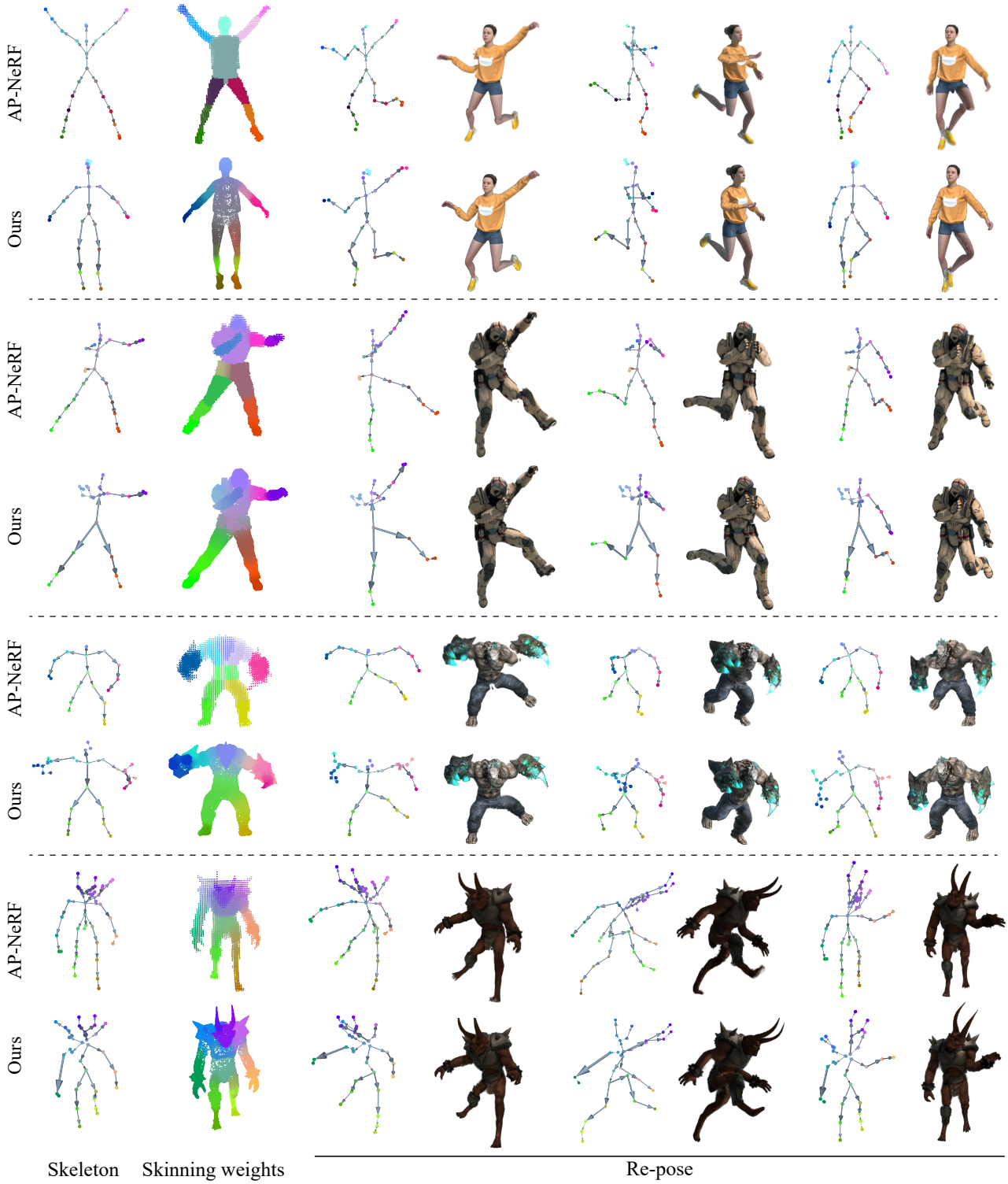

Figure 2. Editing on the D-NeRF dataset.

of other methods. Additionally, we showcase more visual results in Fig. 4 and the complete motion sequences in the **Video Demo.** Despite SC-GS having higher numerical accuracy, from Fig. 4, we can

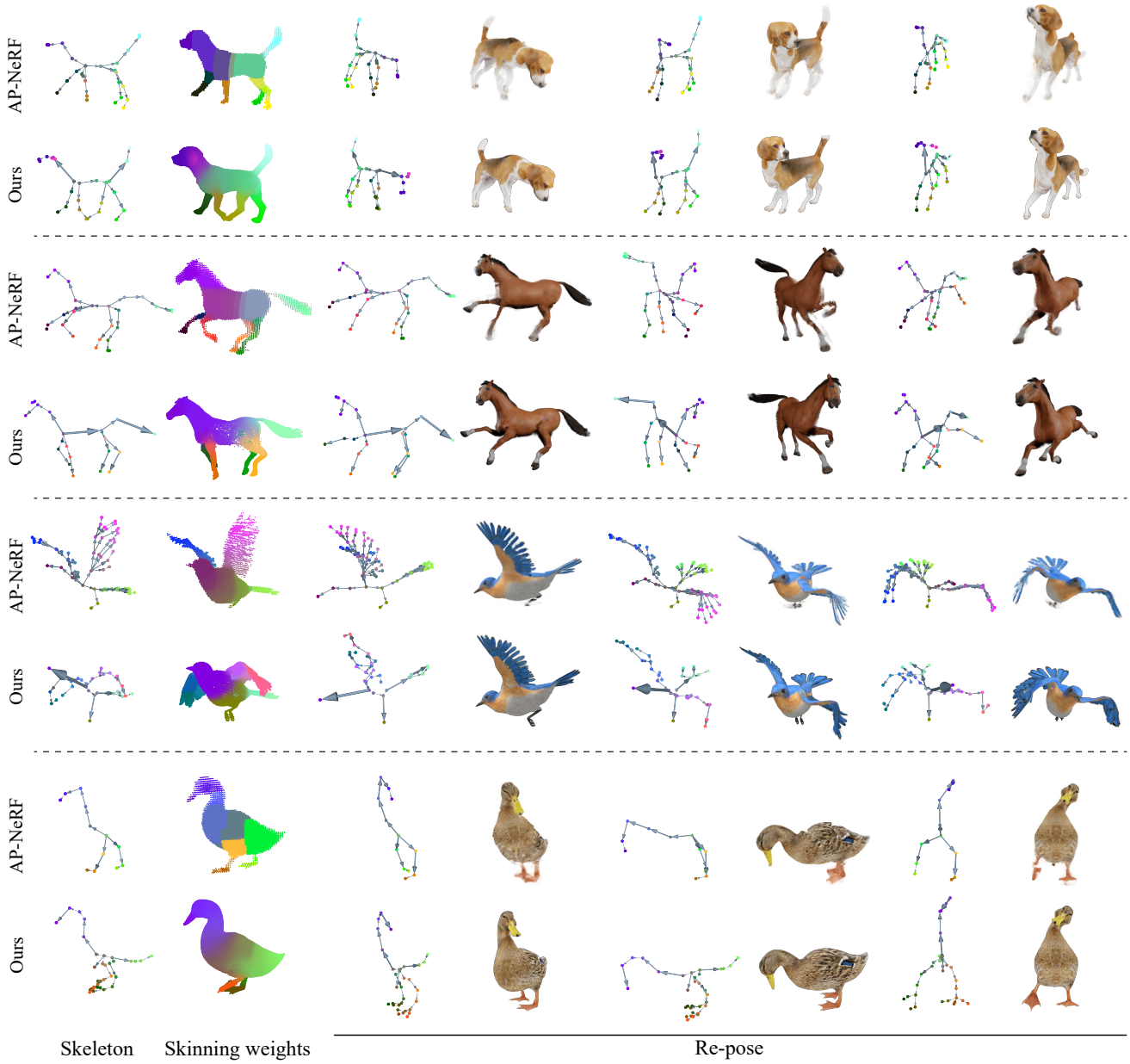

Figure 3. Editing on the DG-Mesh dataset.

see that it exhibits more artifacts compared to our method, such as in the wings of the "Bird" and the legs of the "Horse". Furthermore, when dealing with real data, the performance of SC-GS is notably inferior to our method, as shown in the **Video Demo**.

Figure 4. Novel view rendering on the DG-Mesh dataset.

Table 1. Comparisons with the state-of-the-art methods on the D-NeRF dataset [6]. The best and second-best results are highlighted in bold and underlined.

Method	Skeleton	Hook			Trex			JumpingJacks		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-NeRF [6]	No	29.25	0.968	0.1120	31.75	0.974	0.0367	32.80	0.981	0.0381
TiNeuVox [2]	No	31.45	0.971	0.0569	32.70	0.987	0.0340	34.23	0.986	0.0383
4D-GS [8]	No	30.99	0.990	0.0248	32.16	0.988	0.0216	33.59	0.990	0.0242
SC-GS [3]	No	**39.87**	**0.997**	**0.0076**	**41.24**	**0.998**	**0.0046**	**41.13**	**0.998**	**0.0067**
AP-NeRF [7]	Yes	30.24	0.970	0.0500	32.85	0.980	0.0200	34.50	0.980	0.0300
Ours	Yes	37.49	0.994	0.0136	38.40	**0.998**	0.0063	40.70	0.997	0.0069
Method	Skeleton	Hellwarrior			Mutant			Standup		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-NeRF [6]	No	25.02	0.955	0.0633	31.29	0.978	0.0212	32.79	0.983	0.0241
TiNeuVox [2]	No	28.17	0.978	0.0706	33.61	0.982	0.0388	35.43	0.991	0.0230
4D-GS [8]	No	31.39	0.974	0.0436	35.98	0.996	0.0120	35.37	0.994	0.0136
SC-GS [3]	No	**42.93**	**0.994**	**0.0155**	**45.19**	**0.999**	**0.0028**	**47.89**	**0.999**	**0.0023**
AP-NeRF [7]	Yes	27.53	0.960	0.0600	28.56	0.960	0.0300	31.93	0.970	0.0200
Ours	Yes	41.21	0.989	0.0301	42.72	0.998	0.0057	44.37	0.998	0.0047

Table 2. Comparisons with the state-of-the-art methods on the DG-Mesh dataset [4]. The best and second-best results are highlighted in bold and underlined.

Method	Skeleton	Beagle			Bird			Duck		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-NeRF [6]	No	–	–	–	21.05	0.884	0.1890	32.71	0.982	0.0312
TiNeuVox [2]	No	38.86	0.983	0.0287	25.69	0.934	0.0841	34.38	0.973	0.0291
4D-GS [8]	No	**42.15**	0.990	0.0222	26.75	0.958	0.0443	36.69	0.984	0.0193
SC-GS [3]	No	41.20	**0.998**	**0.0054**	**32.55**	**0.980**	0.0269	**40.41**	**0.998**	**0.0047**
AP-NeRF [7]	Yes	38.70	0.984	0.0281	25.08	0.933	0.0827	34.17	0.973	0.0287
Ours	Yes	39.74	**0.998**	0.0077	31.82	**0.980**	**0.0263**	39.84	0.997	0.0061
Method	Skeleton	Girlwalk			Horse			Average		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
D-NeRF [6]	No	31.15	0.989	0.0336	27.78	0.971	0.0573	28.17	0.957	0.0778
TiNeuVox [2]	No	32.62	0.984	0.0341	28.18	0.960	0.0623	31.95	0.967	0.0477
4D-GS [8]	No	34.15	0.989	0.0145	30.07	0.974	0.0357	33.96	0.979	0.0272
SC-GS [3]	No	**42.33**	**0.998**	**0.0084**	**38.29**	**0.990**	**0.0227**	**38.96**	**0.993**	**0.0136**
AP-NeRF [7]	Yes	32.63	0.984	0.0344	28.58	0.963	0.0561	31.83	0.967	0.0460
Ours	Yes	40.98	0.997	0.0107	35.87	0.984	0.0337	37.65	0.991	0.0169

More Details on the ZJU-MoCap Dataset. We conduct more experiments on the real-captured dataset, ZJU-MoCap dataset [5], and show the comparisons with AP-NeRF. Since our template-free method performs reconstruction and rigging simultaneously, it faces challenges with videos captured by a fixed camera. Improved results can be achieved when the camera is allowed to move. Therefore, we used 6 cameras (1, 5, 9, 13, 17, 21) to simulate monocular videos with camera movement. At each time, we only select one image captured by one of these cameras. Additionally, to more accurately capture the motion of the human body, we use frame-by-frame corresponding SMPL vertices to initialize the 3D Gaussians and deformation fields. We show the comparisons on the other 17 cameras in Table 3 and Fig. 5. We can see that the performance of our method is significantly better than AP-NeRF.

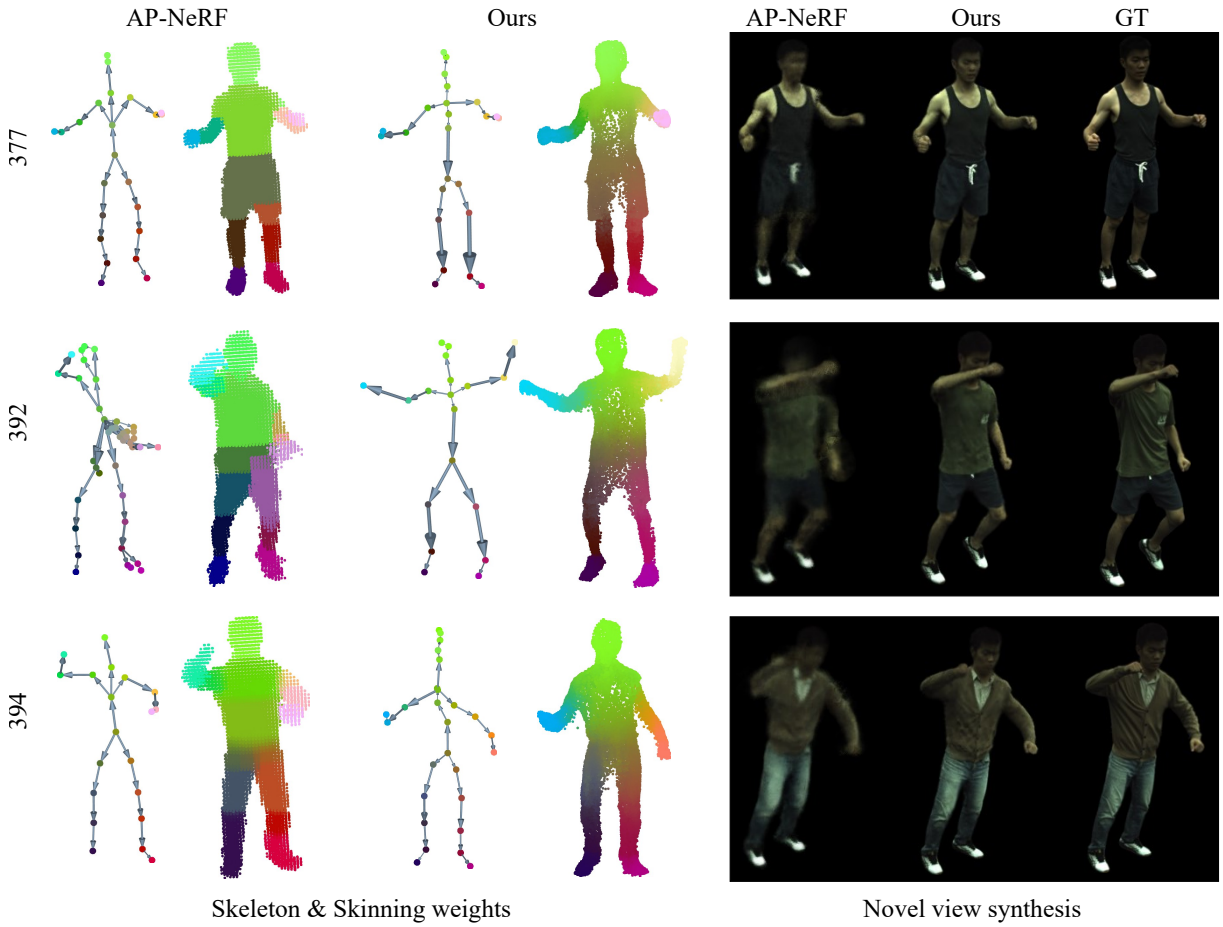

Figure 5. Comparisons of skeleton, skinning weights and novel view synthesis with AP-NeRF on the ZJU-MoCap dataset [5].

2. More Ablation Studies

Table 4 lists the numerical results of our ablation experiments. Additionally, we validated the effectiveness of our skeleton-driven deformation module. We compared our method without MLP $F_\Phi(\gamma_t(t))$ for skinning weight or without pose-dependent detail deformation. From Table 4 and Fig. 6, we can see these variants result in a slight decrease in rendering quality, indicating that these two modules are

Table 3. Comparisons of the average precision on the ZJU-MoCap dataset [5].

Method	377			386			387		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AP-NeRF [7]	24.39	0.925	0.0827	28.94	0.932	0.0841	24.30	0.917	0.0961
Ours	33.78	0.983	0.0202	36.63	0.981	0.0285	31.25	0.971	0.0395
Method	392			393			394		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
AP-NeRF [7]	26.08	0.918	0.0983	24.53	0.908	0.1050	25.49	0.915	0.0943
Ours	34.06	0.975	0.0351	31.65	0.969	0.0391	33.87	0.974	0.0339

effective in matching details without changing the main parts of the deformation.

Table 4. Ablation studies of our method on the D-NeRF dataset [6].

Variants	PSNR ↑	SSIM ↑	LPIPS ↓
w.o. 2D proj. loss L^t_{proj}	40.60	0.996	0.0108
Fixed weight w^t_{proj}	40.68	0.996	0.0117
w.o. MLP for skin.	40.58	0.996	0.0112
w.o. detail def.	39.02	0.993	0.0163
Anisotropy	41.98	0.996	0.0080
Isotropy (Ours)	40.82	0.996	0.0112

Robustness of Skeleton Construction under Large Motion. To evaluate the robustness of our method under large motion, we selected the sequence "393" from the ZJU-MoCap dataset. We tested the sequence using frames indexed as $[0, k, 2k, ..., nk]$, where a larger value of k corresponds to greater motion. We set $k = 1, 5, 10$ and $n = 65$ and show the results in Fig. 7, which demonstrates the robustness of our skeleton extraction method in handling large motions.

Failure Cases of Skeleton Construction. Our method is dependent on the quality of 2D skeleton extraction and silhouettes. Although we proposes an adaptive weighting mechanism to mitigate the impact of erroneous 2D skeletons, when a significant portion of frames contain inaccurate estimations, our method fails to produce semantically plausible skeletal structures (Fig. 8 (a)). Developing higher-quality 2D skeleton extraction methods will greatly improve the quality of the resulting skeleton tree. Secondly, since our automated rigging is inherently related to motions, when two adjacent regions exhibit no relative transformations over the input sequence, our method struggles to distinguish them, resulting in their inability to be properly separated (Fig. 8 (b)). Integrating semantic segmentation or similar techniques to model the skeleton and skinning weights represents a promising direction for future research.

3. Details of Skeleton Construction

We provide a detailed description of the skeleton construction process. Alg. 1 demonstrates the overall process.

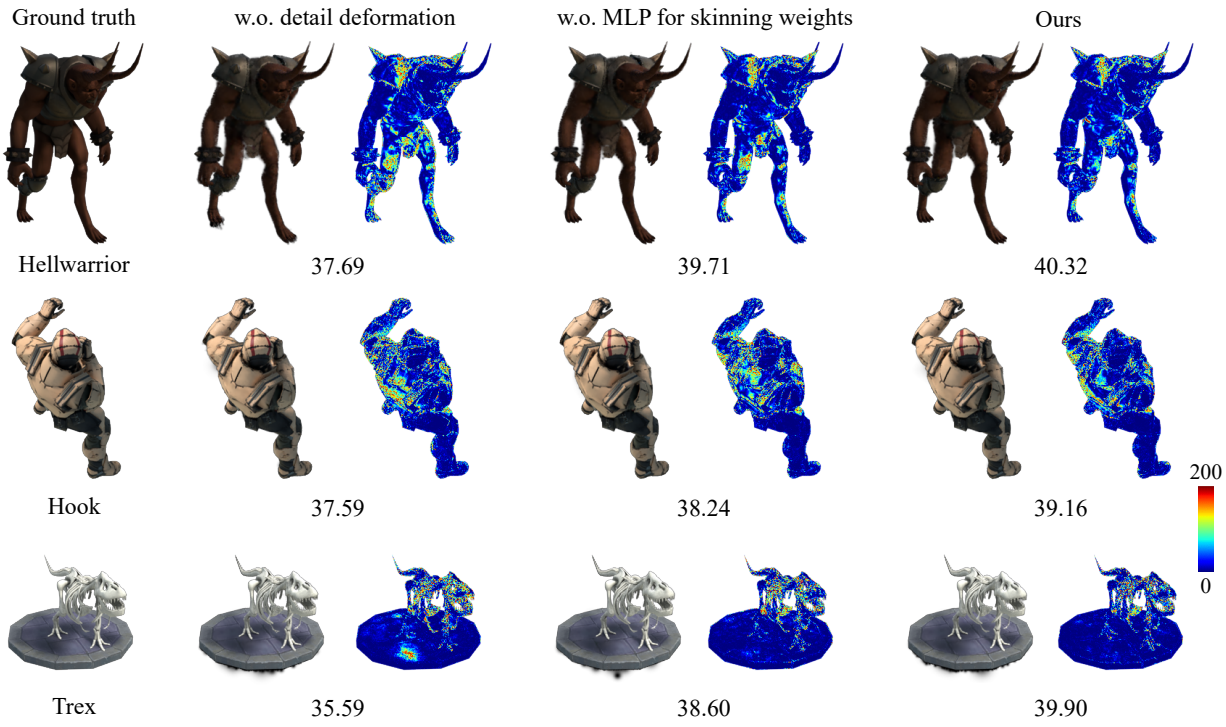

Figure 6. Visual comparisons of our method, without MLP for skinning weights or pose-dependent detail deformation, with annotated PSNR values.

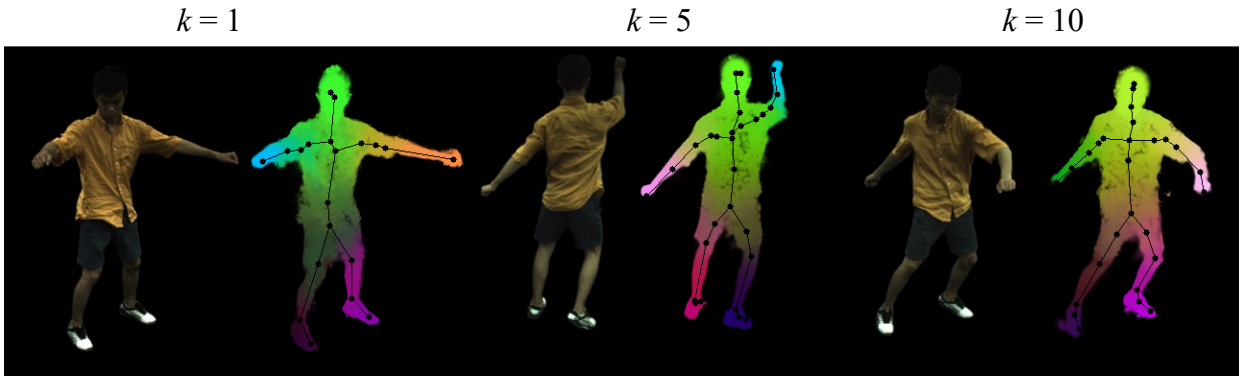

Figure 7. The input image (left) and the visualized skeleton and skinning weights (right).

Prune or Merge for Dense Skeleton Construction. After constructing the minimum spanning tree, due to the presence of noise, we need to remove redundant branches and merge closely located joints. Specifically, if an endpoint passes through fewer than r connected points on its way to the nearest junction, we consider it as an unnecessary point and remove this endpoint along with these connection points. In addition, we will merge two junctions if the number of connection points between them is less than r, and remove these connected points if they exist. Since the distribution of these nodes is uniform, such operations generally do not remove important feature points with long neighboring edges. By default, we set $r = 3$.

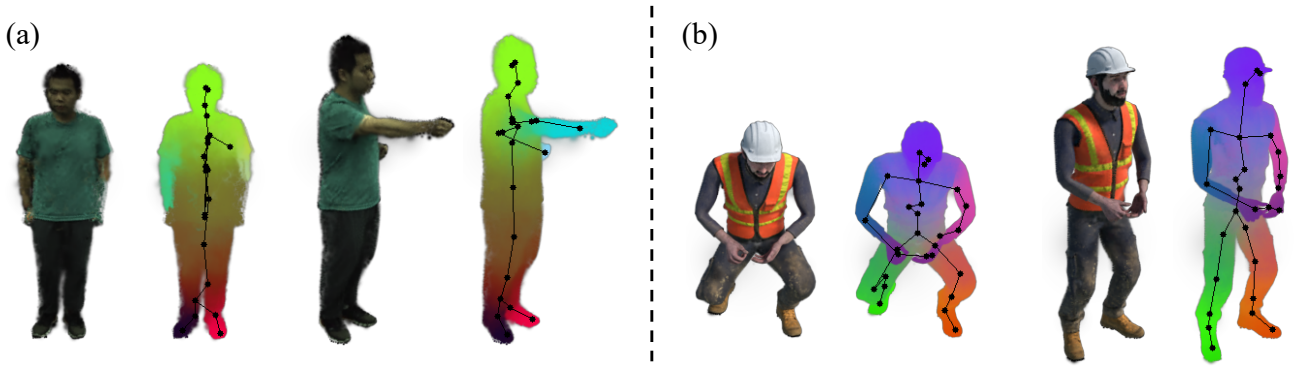

Figure 8. Two frames of rendered images (left), skeleton and skinning weights (right) from (a) "386" on the ZJU-MoCap dataset, and (b) "Standup" on the D-NeRF dataset.

Construction of $\mathcal{J}$ for Skeleton Simplification. The following steps are performed to obtain the joint set $\mathcal{J}$ of the sparse skeleton:

Step 1: Initialize $\mathcal{J}$ and define initial paths. We regard all junctions and endpoints as key points of $\mathcal{J}_d$ and add them into $\mathcal{J}$. For any two connected key points $(\mathbf{J}_a, \mathbf{J}_b)$, i.e. when moving from $\mathbf{J}_a$ to $\mathbf{J}_b$, there are no other key points, we define a path $\mathtt{P}_{ab} = \{\mathbf{J}_a, \mathbf{p}_1^{ab}, ..., \mathbf{p}_m^{ab}, \mathbf{J}_b\}$ (Fig. 9 (a)) where $\mathbf{p}_i^{ab} \in \mathcal{J}_d$ are the passing points from $\mathbf{J}_a$ to $\mathbf{J}_b$.

Step 2: Select a candidate joint from each path. From these passing points $\{\mathbf{p}_i^{ab}\}$, we select a point that is more likely based on geometric position and motion information as the candidate points of $\mathcal{J}$. We assume that geometric turning points are more likely to be joint points. So we first connect $\mathbf{J}_a$ and $\mathbf{J}_b$ with a straight line segment l_{ab} and calculate the distance $D(\mathbf{p}_i^{ab}, l_{ab})$ from $\mathbf{p}_i^{ab}$ to l_{ab}. See Fig. 9 (b), the point with the maximum distance is more likely to be a geometric turning point. However, due to noise in $\mathcal{J}_d$, directly selecting points based on this distance may lead to selecting points very close to $\mathbf{J}_a$ or $\mathbf{J}_b$ (such as red points in Fig. 9 (c)), which are undesirable joints. Therefore, we define the following score

$$\mathtt{s}_i^{ab} = D(\mathbf{p}_i^{ab}, l_{ab}) - 0.1 \min(D(\mathbf{p}_i^{ab}, \mathbf{J}_a), D(\mathbf{p}_i^{ab}, \mathbf{J}_b)), \tag{1}$$

where $D(\mathbf{p}_i^{ab}, \mathbf{J}_a)$ and $D(\mathbf{p}_i^{ab}, \mathbf{J}_b)$ are the distance between $\mathbf{p}_i^{ab}$ with $\mathbf{J}_a$ and $\mathbf{J}_b$, separately.

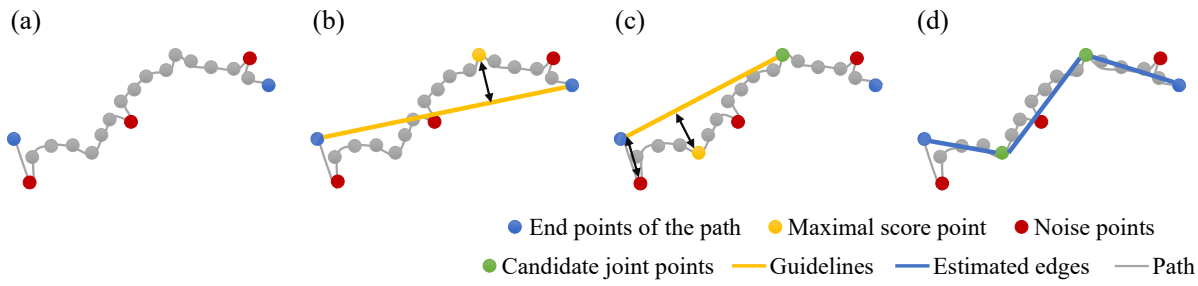

Figure 9. Selection of candidate joints.

To consider motion information, at each time t, we transform $\mathbf{p}_i^{ab}, \mathbf{J}_a$, and $\mathbf{J}_b$ to the position of the $\mathbf{p}_i^{ab^t}, \mathbf{J}_a^t$, and $\mathbf{J}_b^t$ according to the initial deformation field in Sec. 3.1 and compute $\mathtt{s}_i^{ab^t}$ according to Eq. (1). The point $\mathbf{p}_*^{ab}$ with the maximal $\sum_t \mathtt{s}_i^{ab^t}$ is selected as the candidate joint.

Algorithm 1: Construction of skeleton.

Input: Skeleton candidate nodes $\{\mathbf{c}^{t^*}\}$ and the corresponding semantic labels;
Result: Skeleton including joints $\mathcal{J} = \{\mathbf{J}_j\}$ and parent indices $\{\mathbf{J}_{A(j)}\}$.

```
// Construct dense skeleton.
```
Perform farthest point sampling to obtain uniformly distributed control points $\{\mathbf{c}_s^{t^*}\}_{s\in\mathcal{S}}$;
Construct the edge set $\mathcal{E}$ for $\{\mathbf{c}_s^{t^*}\}$ using Prim's algorithm;
Prune redundant endpoints and merge close junctions;
```
// Skeleton simplification.
```
$\mathcal{J} = \emptyset$;
Add all junctions and ends to $\mathcal{J}$;
Establish paths $\{\mathtt{P}\}$ between two connected nodes in $\mathcal{J}$;
for $\forall(J_a, J_b) \in \mathcal{J}$ **do**
    Compute the candidate joints set h_{ab} according to Alg. 2;
end for
```
// Symmetry enhancement based on semantic labels.
```
for $\forall(\mathtt{P}_{ab}, \mathtt{P}_{ef})$ **do**
    if $||\mathtt{P}_{ab}| - |\mathtt{P}_{ef}|| < \epsilon_1 \cdot \max(|\mathtt{P}_{ab}|, |\mathtt{P}_{ef}|)$ *and* $|\mathtt{sem}(\mathtt{P}_{ab}) \cap \mathtt{sem}(\mathtt{P}_{ef})| > \epsilon_2 \cdot |\mathtt{sem}(\mathtt{P}_{ab}) \cup \mathtt{sem}(\mathtt{P}_{ef})|$ **then**
        Select a path with more moderate number of candidate joints;
        Reselect the candidate joints for other path according to the selected path;
    end if
    Add these candidate joints into $\mathcal{J}$;
end for

***Step 3: Add paths and continue to select candidate points*.** The new candidate point $\mathbf{p}_*^{ab}$ and endpoints $\mathbf{J}_a$ and $\mathbf{J}_b$ form two new paths. Repeat the *step 2* to find new candidate points. Fig. 9 shows this process.

***Step 4: Enhance the symmetry of candidate joints using precomputed DINO features* [1].** Considering that many objects exhibit symmetry, we further utilize semantic information to enhance the symmetry of candidate joints selected by *step 2-3*. For two paths $\mathtt{P}_{ab}$ and $\mathtt{P}_{ef}$ in *step 1*, we consider they are symmetric if their lengths and semantic labels are similar. By projecting $\mathbf{J}_a^t, \mathbf{J}_b^t$ and $\mathbf{p}_i^{ab^t}$ onto the image according to the camera perspective, we can obtain semantic classification based on image segmentation using DINO features. We set the semantic label of $\mathbf{J}_a, \mathbf{J}_b$ and $\mathbf{p}_i^{ab}$ as the median of these labels at all times to represent the semantic category of the majority of frames. Since these semantic labels are also not precise, we count the number of different kinds $\mathtt{sem}(\mathtt{P}_{ab})$ and $\mathtt{sem}(\mathtt{P}_{ef})$ of semantics that appear on $\mathtt{P}_{ab}$ and $\mathtt{P}_{ef}$ respectively. If $|\mathtt{sem}(\mathtt{P}_{ab}) \cap \mathtt{sem}(\mathtt{P}_{ef})| > \epsilon_2 \cdot |\mathtt{sem}(\mathtt{P}_{ab}) \cup \mathtt{sem}(\mathtt{P}_{ef})|$, then we consider them to be similar in semantics.

If $\mathtt{P}_{ab}$ and $\mathtt{P}_{ef}$ are similar in length and semantics, then we will perform symmetrical correction on them so that the number and distribution of joints along the two paths are similar. We tend to choose the path with a moderate number of candidate points as templates (closer to ϵ_3) and adjust the other path to be similar. Without loss of generality, let's assume $\mathtt{P}_{ab}$ is the template. we will discard the candidate joints in $\mathtt{P}_{ef}$ and reselect points from $\mathtt{P}_{ef} = \{\mathbf{J}_e, \mathbf{p}_{d_1}, ..., \mathbf{p}_{d_{n_{ab}}}, \mathbf{J}_f\}$, where $\mathbf{p}_{d_i}$ is selected by

Algorithm 2: Selection of candidate joints on a path.

Input: $\mathtt{P}_{ab} = [\mathbf{J}_a, \mathbf{p}_1, ..., \mathbf{p}_m, \mathbf{J}_b]$, current candidate joints set h_{ab};
Result: Updated candidate joints set h_{ab}.

Set the candidate joints set $h_{ab} = \emptyset$ for $\mathtt{P}_{ab}$;
Compute $\mathtt{s}_i^{ab^t}$ according to Eq. (1);
$\mathbf{p}_*^{ab} = \arg\max_{\mathbf{p}_i} \sum_t \mathtt{s}_i^{ab^t} / |\mathcal{I}|$;
if $\sum_t D_t(\mathbf{p}_*^{ab}, l_{ab}) / |\mathcal{I}| < \bar{l}$ **then**
    return h_{ab};
else
    $h_{ab} = h_{ab} + \{\mathbf{p}_*^{ab}\}$;
    Perform Alg. 2 with $[\mathbf{J}_a, \mathbf{p}_1, ..., \mathbf{p}_*^{ab}]$ and h_{ab} as input;
    Perform Alg. 2 with $[\mathbf{p}_*^{ab}, ..., \mathbf{J}_b]$ and h_{ab} as input;
    return h_{ab};
end if

$\frac{\mathtt{len}([\mathbf{J}_e, \mathbf{p}_{d_i}])}{\mathtt{len}([\mathbf{J}_e, \mathbf{J}_f])} \approx \frac{\mathtt{len}([\mathbf{J}_a, \mathbf{p}_i^{ab}])}{\mathtt{len}([\mathbf{J}_a, \mathbf{J}_b])}$. Here $\mathtt{len}(x, y)$ denotes the length of the path between x and y. Based on our experience, setting $\epsilon_1 = 30\%, \epsilon_2 = 60\%, \epsilon_3 = 3$ is a robust choice. After processing each path pair, we obtain the final sparse joint set $\mathcal{J}$.

4. Implementation Details

During the training process, we first trained the initialization stage for 80,000 iterations, optimizing 3D Gaussians $\mathcal{G}$, the positions of skeleton-aware nodes and the parameters of MLP F_Θ for the corresponding time-related rotations and translations. Then we obtained the initialized deformation field and candidate nodes of skeleton significance. We set the weight $w_{\mathtt{proj}} = 10^{-3}$, and dynamically decreased $w_{\mathtt{arap}}$ from 10^{-4} to 0 during iterations. After completing the initialization training, we obtained a new canonical shape and constructed the skeleton. The new canonical shape will serve as the initial 3D Gaussian representation for the skeleton-driven dynamic model. We obtained the initial values of skeleton-driven deformation by pointwise supervision with the skeleton-aware node-controlled deformation field. Finally, we performed the training process for 100,000 iterations, optimizing 3D Gaussians $\mathcal{G}$, the parameters of MLP F_Φ for time-related skeleton poses and translations, scaling factors $\{\eta_{i,j}\}_{j=2}^{\mathcal{J}}$ and the parameters of F_Ψ for learnable skinning weights, as well as the parameters of F_Π for pose-dependent detail deformation. At this stage, the positions of joints and parent indices of the skeleton are fixed, as treating them as variables can lead to instability. During the inference stage, we solely executed the skeleton-driven deformation model.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conf. Comput. Vis.*, pages 9650–9660, 2021. 10

[2] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 2022. 5

[3] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4220–4230, 2024. 1, 5

[4] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from monocular videos. In *Int. Conf. Learn. Represent.*, 2025. 5

[5] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 6, 7

[6] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10318–10327, 2021. 5, 7

[7] Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. Template-free articulated neural point clouds for reposable view synthesis. *Adv. Neural Inform. Process. Syst.*, 36, 2024. 1, 5, 7

[8] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 20310–20320, 2024. 5