(Supplementary Material) Hierarchically Structured Neural Bones for Reconstructing Animatable Objects from Casual Videos

In this supplementary material, we provide additional details, comparisons, and results of our method:

- Manipulation UI and comparison in Section A.
- Manipulation user study in Section B
- Descriptions of the datasets in Section C.
- Dynamic addition and deletion of bones in Section D
- Details of our method in Section E.
- Additional ablation studies in Section F.
- Additional s reconstruction results in Section G.
- Additional manipulation results in Section H.
- Discussion on the societal impacts of our method in Section I.

A Manipulation Comparison

We showcase the easier and more comprehensible manipulation process achieved by our method through the supplementary videos and manipulated results. During the manipulation process, the animator utilizes our manipulation UI and manually adjusts the bone parameters to achieve the desired poses of the objects. We provide a description of our manipulation UI in Fig. 8. The supplementary video (named **"manipulation-UI-and-comparison.mp4"**) demonstrates the actual manipulation process of our method and BANMo [6]. As shown in the video, the manipulation process of our method is much easier and interpretable compared to BANMo, achieving the desired poses in about 4× shorter time.

The manipulated objects are demonstrated in Fig. 9. It is worth mentioning that users need to take significantly fewer actions for manipulating our structured deformation model. For instance, to manipulate Eagle, users can obtain the target pose by manipulating just 5 bones. In contrast, at least 18 bones are need to be adjusted when manipulating the result of BANMo, as its bones are unstructured, and just scattered throughout the surfaces without considering the basis of motions. In the manipulation process of Cat, coarse and large motions like standing are achieved by moving coarse-level bones using our method. On the other hand, the result of BANMo requires adjustments of almost all bones (20 out of 25 bones) to make such manipulations, leading to intricate adjustments and a challenging manipulation process. Thanks to the hierarchically structured deformation model, the proposed method provides much more intuitive and convenient manipulation process to users.

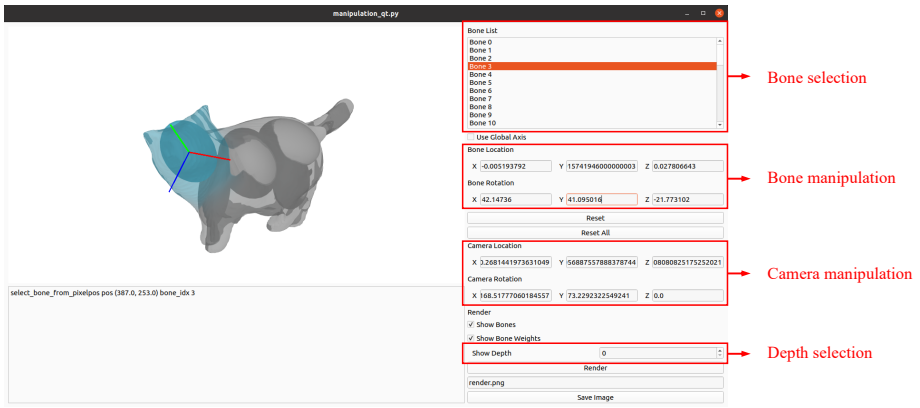

Fig. 8: Description of our manipulation UI. Users can manipulate cameras and bone parameters with mouse actions. To select the designated bone, users can see the entire bones or bones at a specific depth, and select the target bone by clicking it in the left side, or choosing it from the bone list on the top-right side. The right side shows the manipulation and camera parameters, in which users can directly manipulate these parameters. We refer to the provided supplementary video for more descriptions the actual manipulation process.

Table 5: User study results on manipulation.

	Whale		Eagle		Cat		Swing		Avg.	
	Time	Pref	Time	Pref	Time	Pref	Time	Pref	Time	Pref
BANMo	2m 11s	3.2	3m 48s	2.9	5m 17s	2.6	9m 6s	1.5	5m 5s	2.55
Ours	**1m 29s**	**4.6**	**3m 15s**	**3.7**	**3m 31s**	**3.9**	**7m 4s**	**2.5**	**3m 50s**	**3.68**

B Manipulation User Study

We compare our method with BANMo in terms of manipulation capabilities by conducting a user study. For the user study, we recruited 12 participants with no prior experience using 3D tools. Each participant was instructed to manipulate 3D models to match given target poses. The test was conducted on four different objects, including Whale, Eagle, Cat, and Swing. Fig. 10 shows the target poses used in the user study. We measured both the time taken to achieve the desired poses and the preference ratings, rated on a scale from 1 (Difficult) to 5 (Easy). For each object, we calculated the average of the 10 responses, excluding the shortest and longest times among the 12 responses. As shown in Table 5, our method achieve higher preference ratings and shorter completion times across all objects. The results demonstrate that our structured bone representation improves manipulation capability in terms of time taken and interpretability of learned control points.

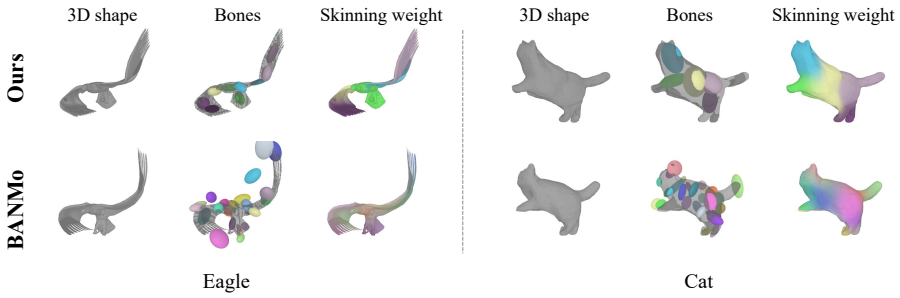

Fig. 9: Manipulation comparison with BANMo [6]. Users can firstly achieve manipulations of coarse and larger motions using our method, whereas BANMo requires adjustments of almost all bones to make such manipulations. Notably, the manipulation of Eagle is achieved only using 5 bones with our method, while at least 18 bones are adjusted in BANMo.

Fig. 10: The problem presented in the user study. In the user study, we instructed the users to manipulate to achieve the following target pose.

C Dataset

We conduct additional experiments on a more diverse range of animals, including a dog, a bat, and a whale:

- **AMA human dataset** [4] includes multi-view videos capturing actor performances from 8 synchronized cameras and ground-truth mesh. We select two sets of videos, Swing (1200 frames) and Samba (1400 frames). We omit time synchronization and camera extrinsic parameters during training, treating the videos as monocular.
- **Animated objects dataset** [6] offers Eagle videos, that are rendered with an animated 3D eagle model and varying camera trajectories. Each video comprises 150 frames, and a total of 5 videos are utilized as input.
- **Casual video dataset** [6] includes multiple videos featuring a Cat and a Shiba Inu dog, respectively. These videos are captured casually using monocular cameras, with no control over camera movements. We utilize a total of 11 videos (900 frames) for Cat and 14 videos (1407 frames) for Dog. Specifically, objects exhibit unrestricted movement within individual videos, and the background undergoes changes across the different video sequences.

Table 6: Dataset license description.

Dataset	Instance	Human	Synthetic	Paper	License
AMA human dataset	Swing, Samba	✓		[4]	License not specified
Animated object dataset	Eagle		✓	[6]	Turbosquid license
Casual video dataset	Cat, Dog			[6]	CC0
Dynamic Object dataset	Bat, Whale		✓	[2]	SketchFab Standard License

- **Dynamic object dataset** [2] presents videos of a whale and a bat, which are rendered using animated 3D objects. The animals are depicted from 15 different viewing angles, and for optimization purposes, we utilize videos from 12 of these angles. Each video consists of 46 frames, with a total of 552 frames used for both Bat and Whale.

Dataset license. Additionally, we provide the dataset license, the research paper introducing the dataset, and information on whether it includes human subjects in Table 6.

Human subject. We adhere to ethical principles outlined in ECCV ethics guidelines. When utilizing human-derived data, particularly in the case of the AMA human dataset, we exercise careful consideration. The dataset is collected with consent and is made publicly available. We utilize the data with proper citation to acknowledge its source. The dataset is intended for editing purposes, and we ensure its usage aligns with our purpose. If concerns arise regarding the potential presence of personally identifiable information in facial regions, we pledge to blur or mask the facial area.

D Dynamic Addition and Deletion of Bones

Thanks to the flexible structure of hierarchically structured bones, users have the capability to add additional control points where needed or remove unnecessary ones. Specifically, users select the designated parent bones to add more bones, and then the child bones are appended to the selected segments accordingly. With further optimization of the appended bones, users finally obtain the 3D models with more control points for finer manipulation. For the removal of redundant bones, users select the target bones, and the corresponding child bones can be eliminated by removing them from our tree structures. This process can be easily implemented by modifying the leaf bones. We would like to note that prior template-free methods [5, 6] lack the capability of dynamically adding or removing control points in designated areas, as their Gaussian ellipsoids are unstructured. Skeleton-based approaches [1, 7] have insufficient capability of modifying predefined templates, and they offer limited transformations that are restricted to a given skeleton. Fig. 11 illustrates the examples of the dynamic addition and deletion of the bones on Cat.

E Method Detail

E.1 Losses

Our method follows reconstruction losses L_{recon} and cycle loss L_{cycle} that are proposed in BANMo [6], as follows:

$$\mathcal{L}_{recon} = \mathcal{L}_{rgb} + \mathcal{L}_{sil} + \mathcal{L}_{OF} + \mathcal{L}_{feat}, \tag{15}$$

$$\mathcal{L}_{cycle} = \mathcal{L}_{2D-cyc} + \mathcal{L}_{3D-cyc}. \tag{16}$$

- **RGB reconstruction loss** L_{rgb} compares rgb values C_{GT} of given frames to the composited values $\hat{C}(r)$, as

$$L_{rgb} = \sum_r ||\hat{C}(r) - C_{GT}||^2. \tag{17}$$

- **Silhouette reconstruction loss** L_{sil} compares mask values M_{GT} extracted from given frames and the composited density values $\hat{M}(r)$ through differentiable volume rendering:

$$L_{sil} = \sum_r ||\hat{M}(r) - M_{GT}||^2. \tag{18}$$

- **Flow reconstruction loss** L_{OF} compares 2D optical flow values F_{GT} extracted from the off-the-shelf flow network and the predicted flow values. In detail, given two frames of time t and t', we compute flows by firstly backward warping rays r^t to the rays in the canonical space r^c, then forward warping the rays r^c to the $r^{t'}$ in the t' frame. The predicted pixel locations at time t' are compared to the pixel location at time t to compute 2D optical flows $\hat{F}$. The flow reconstruction loss is computed as

$$L_{OF} = \sum_{r,(t,t')} ||\hat{F}\big(r,(t,t')\big) - F_{GT}||^2. \tag{19}$$

- **Feature rendering loss** L_{feat} compares 2D Dense-CSE feature D_{GT} from Dense-CSE [3] to the composited predicted Dense-CSE feature values $\hat{D}$. For each 3D point sampled from rays r, the 3D Dense-CSE feature is queried from the feature MLP, and composited to the 2D rendered value.

$$L_{feat} = \sum_r ||\hat{D}(r) - D_{GT}||^2. \tag{20}$$

- **2D cycle loss** L_{2D-cyc} computes cycle consistency between original pixel locations r and the re-projected pixel locations $\hat{r}_{reproj}$. Per each pixel, a 3D point is predicted via canonical embedding in the canonical space. The point is warped to time t space (forward warping), and then projected to image space using a predicted camera projection matrix.

$$L_{2D-cyc} = \sum_r ||\hat{r}_{reproj} - r||^2. \tag{21}$$

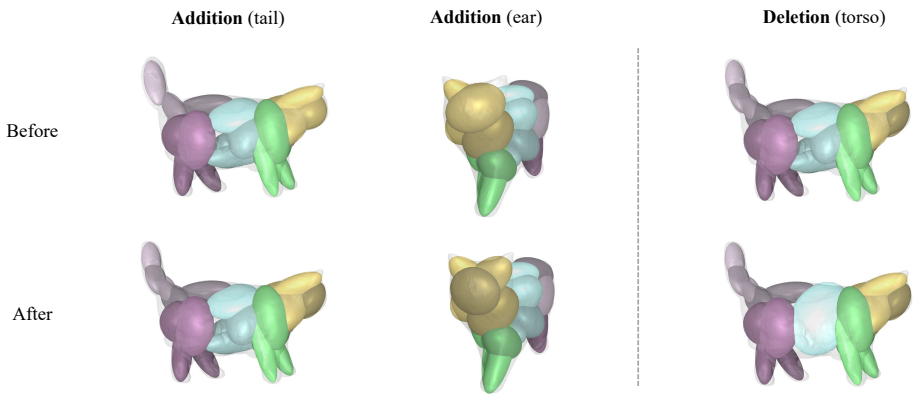

Fig. 11: Examples of dynamic addition and deletion of neural bones on the 3D model of Cat. We add extra bones to the tail and head, allowing for manipulation of finer regions. Conversely, the torso, which requires fewer bones, can be merged.

– **3D cycle loss** L_{3D-cyc} computes cycle consistency of 3D points $\mathbf{x}^t$ by forward warping the canonical points in the canonical space, which was given by backward warping in the time t space as

$$L_{3D-cyc} = \sum_i \tau ||\mathcal{W}^{c\to t} \cdot \mathcal{W}^{t\to c}\mathbf{x^t} - \mathbf{x^t}||^2, \tag{22}$$

where τ is the opacity of the point $\mathbf{x^t}$.

E.2 Child Bone Initialization

When increasing the depths of our bone hierarchy, child bones are initialized using properties inherited from their parent bone. Specifically, a canonical mesh is extracted from the canonical model. Skinning weights of previous depths are computed based on the vertices of the canonical mesh. We identify vertices with the highest skinning weights on the parent bone and cluster them into groups corresponding to the number of child bones based on euclidean distance. The centers of these clusters serve as the initial center positions for the child bones. As for the orients of the child bones, we set them to the identity rotation matrix. For scales, we initialize them with constant values for all bones, regardless of depth. Using these initial values, the deformation MLP f^d for the new depth d is optimized with a small number of iterations. Since this procedure relies solely on the canonical poses of bones, we discovered that a large number of iterations can lead f^d to overfit to these poses. Therefore, additional optimization of f^d using video data containing various poses is necessary.

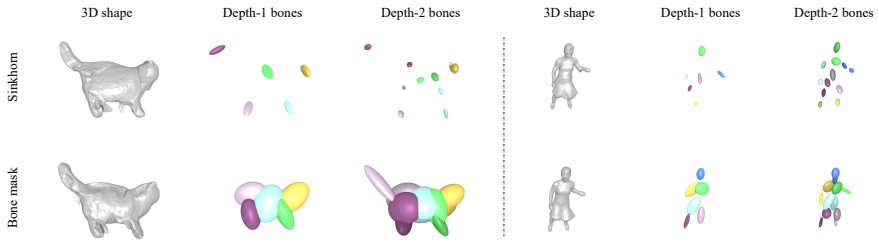

Fig. 12: Ablation results on the bone regularization terms within the framework of bone hierarchy. When bone mask regularization is utilized, it ensures that the scales of bones correspond to the actual scale of the shape, thereby enabling the subdivision of depth-1 bones into depth-2 bones.

Table 7: Ablation on the bone regularization with bone hierarchy. The combination of bone mask regularization with our hierarchical deformation model achieves the best scores.

Bone Reg	#depths	#bones	Samba		Swing	
			CD	F2	CD	F2
Sinkhorn	1	6	8.56	57.23	9.60	52.91
	2	12	7.84	60.79	8.88	56.22
Bone mask	1	6	7.65	61.93	9.27	54.74
	2	12	6.87	66.76	7.74	61.64

E.3 Additional Optimization Detail

We optimize our overall system jointly, including the canonical model g_c and the hierarchical deformation model f, through the previously mentioned losses. Specifically, we sample 6 pixels for each image and 128 points are sampled for each ray. All frames are cropped around the object and resized to the size of 512×512, and we use 512 images for one iteration. We use loss weight 1 for L_{OF}, L_{match}, weight 0.1 for $L_{rgb}, L_{sil}, L_{bone}$, and weight 0.001 for $L_{overlap}$, L_{cover}. As described in the manuscript, we optimize overall system in a coarse-to-fine manner according to the depth of hierarchical neural deformation model. After parent bones are sufficiently optimized and child bones are appended, we freeze the parent bones and concentrate on optimizing the newly added child bones. We use two NVIDIA GeForce RTX 3090 GPUs for the optimization, and each stage takes less than 3 hours in our environment.

F Additional Ablation Study

F.1 Bone Regularization

We further present the ablation results on the effects of combining the bone mask loss with our hierarchical deformation model. We compare the results at depth-1 and depth-2, with our framework using Sinkhorn divergence regularization

Table 8: Progressive optimization ablation on Eagle, Samba, and Swing. BANMo+ extends BANMo by gradually increasing the number of bones during optimization, while BANMo maintains a constant number of bones throughout. Our method also gradually increases the number of bones but utilizes bone hierarchy when adding bones.

data	Eagle		Samba		Swing	
	CD	F2	CD	F2	CD	F2
BANMo	4.66	81.44	7.22	64.99	7.33	64.88
BANMo+	5.52	71.61	6.82	67.17	7.13	64.56
Ours	**4.64**	**81.59**	**6.15**	**72.07**	**7.11**	**65.88**

as in the prior work [6]. The reconstructed 3D shapes and their corresponding bones at each depth are reported in Fig. 12. The most notable difference is that the scale of neural bones align with the scale of the shapes when using bone mask loss. This effect arises from the fact that Sinkhorn regularization only encourages the center of the bones to be placed near the surfaces, while bone mask loss regularizes all properties of the bones, scales, orients, and centers, by encouraging the bones to fit the foreground masks of the objects. Combining the bone mask loss with our hierarchical deformation model results in improved interpretability. Users can better understand the corresponding parts assigned to each bone, while semantic correlations between the bones emerge through the tree structures. The combination of bone mask loss with our hierarchical deformation model also leads to more notable improvement in reconstruction quality, as can be observed in Table 7.

F.2 Progressive Optimization

In our framework, the number of bones increases gradually as depth grows and is further optimized. To analyze whether the improvement arises from hierarchical modeling or the gradual increase in the number of elements, we conduct additional ablation studies on the optimization process. For the analysis, we introduce BANMo+, in which a small number of bones are initialized and optimized in the initial stage. Subsequently, additional bones are progressively added and then re-optimized. We begin with 6 bones in the first stage, doubling their quantity over 3 stages, resulting in a total of 24 bones. We employ identical settings for progressive optimization as in our hierarchical bones. As shown in Table 8, BANMo+ does not bring meaningful improvement, sometimes showing degraded results compared to BANMo. The results suggest that the advancement of our framework is primarily due to the structured modeling of foundational elements, which facilitates the disentanglement of coarse and fine motions.

G Additional Reconstruction Result

Reconstruction results for a wider range of object categories are illustrated in Fig. 13 and Fig. 14. We also present the learned bones, where the bones with the

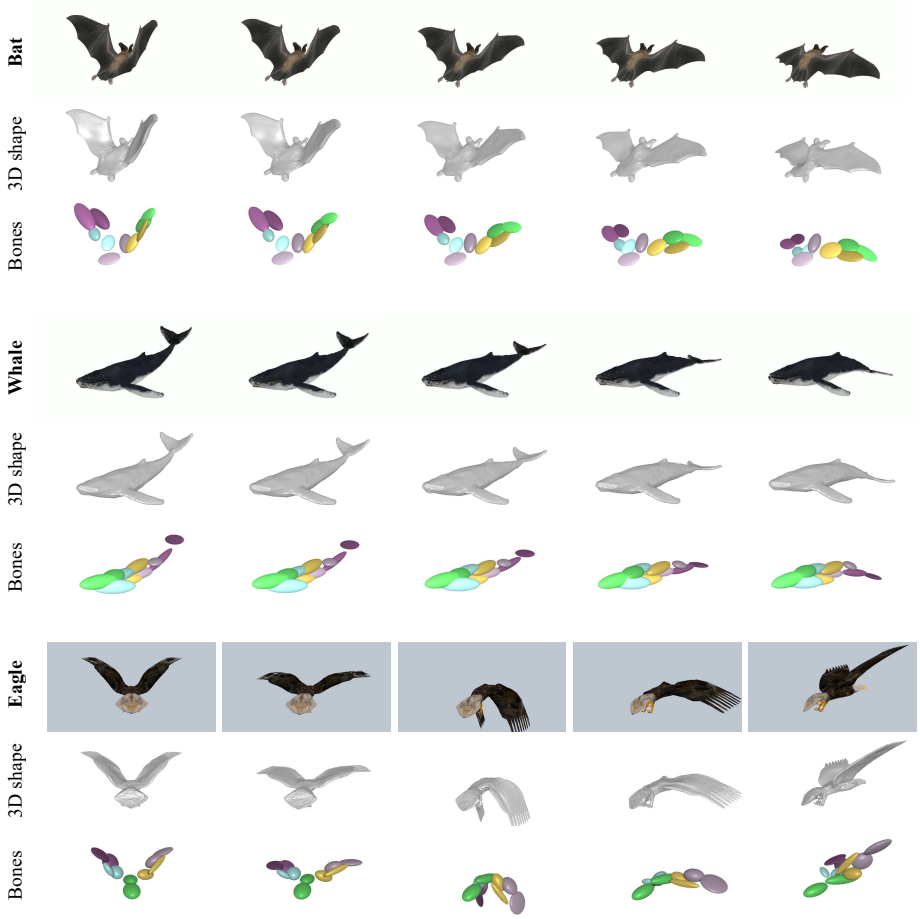

Fig. 13: Reconstruction results on synthetic animals (Bat, Whale, and Eagle). Reconstructed 3D shapes and their corresponding leaf bones are described.

same color indicate the bones assigned to the same parent. Our method demonstrates generalizability across diverse types of animals with distinct motion properties. More results of Samba and Swing are depicted in Fig. 15. Template-free methods excel in reconstructing regions where templates are not provided, such as the skirts of humans. We emphasize and showcase such cases in Fig. 16. Additionally, we present reconstruction results along depths in Fig. 17. As the depth increases, the detailed motion *e.g.* legs of the cat, and arms of human, is captured. For more results and comparisons, please refer to our supplementary video.

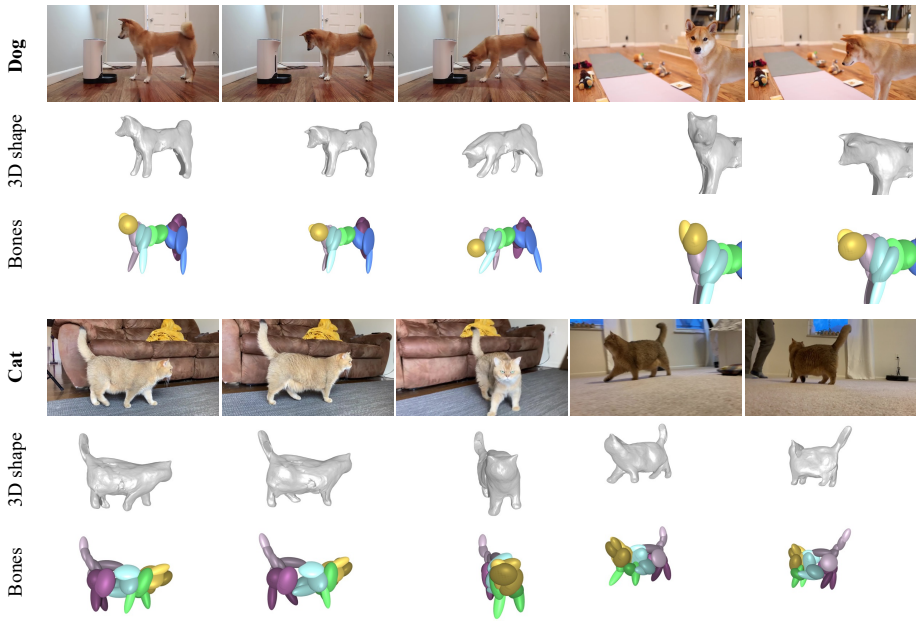

Fig. 14: Reconstruction results of animals from casually captured videos (Dog, Cat). Reconstructed 3D shapes and their corresponding leaf bones are described.

H Additional Manipulation Result

Coarse-to-fine manipulation. Fig. 18 outlines the process of coarse-to-fine manipulation employing our hierarchical deformation models. In the coarse manipulation, all child bones are adjusted simultaneously, *e.g.* the head of cat and the left leg of the human. In the fine manipulation, child bones are manipulated in the local coordinate of their parents, enabling the fine adjustments of the motions, as shown in the left ear of the cat, and the foot of the human.
Coarse-only manipulation. The decomposition of coarse and fine motions allows coarse-level manipulation of the provided videos while preserving fine-level motions. Fig. 19 illustrates the results of manipulation. Specifically, adjusting the parent motions of the arms (colored in blue), which are responsible for controlling both arms, results in the lifting of both arms. The detailed motions of all child bones are brought from the given sequence, preserving the detailed motions of upper arms, lower arms, and hands. The decomposition property of our hierarchical deformation model provides an easier and novel way to manipulate 3D models, which is difficult to be achieved in previous approaches.
Manipulation results. Lastly, we present manipulation results using both coarse and fine-level manipulations in Fig. 20. Such results demonstrate the capability to manipulate 3D models in detail and showcase the ability of our framework to create 3D models with novel poses. For video results, please refer to the supplementary video.

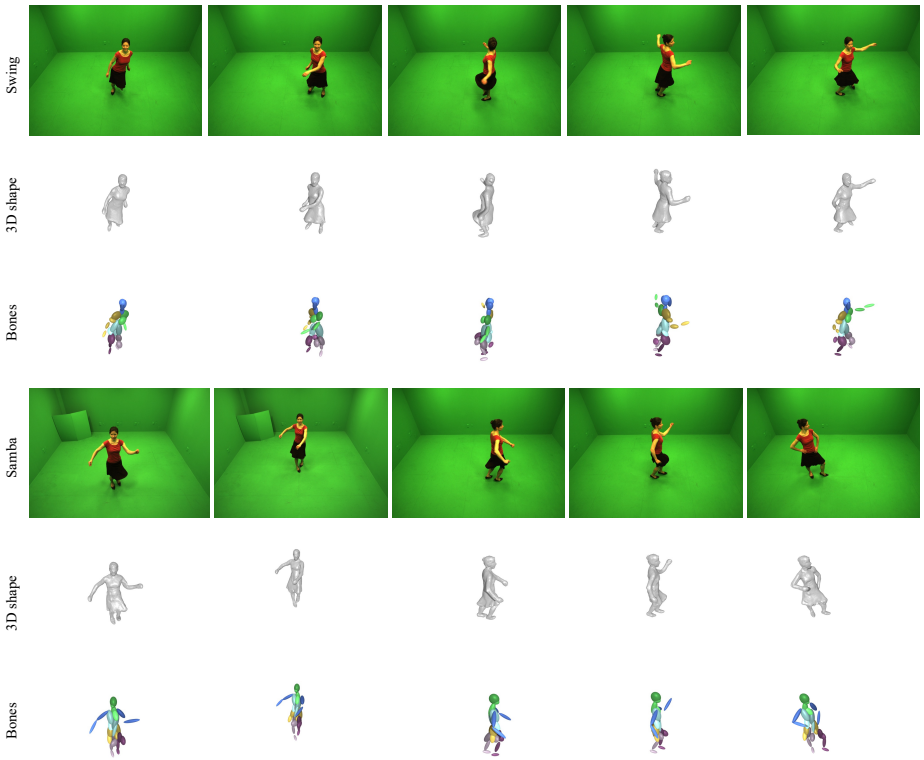

Fig. 15: Reconstruction results on AMA human datasets (Swing, Samba). Reconstructed 3D shapes and their corresponding leaf bones are described.

I Societal Impact

Our framework presents a range of societal impacts, both positive and negative. Positively, it revolutionizes 3D modeling by leveraging casually captured videos, democratizing access to these tools and empowering individuals and small businesses to produce animatable models. Additionally, its simplification of the modeling process enhances accessibility, particularly for users with limited technical skills or resources. However, there are notable concerns regarding potential job displacement, particularly within industries heavily reliant on traditional 3D modeling techniques, as automation may reduce demand for skilled modelers. Furthermore, the use of casually captured videos raises privacy concerns, with unauthorized utilization posing risks such as identity theft. Additionally, the ease of manipulation facilitated by our framework may exacerbate issues of digital manipulation and misinformation, potentially leading to the spread of false representations and harmful societal consequences.

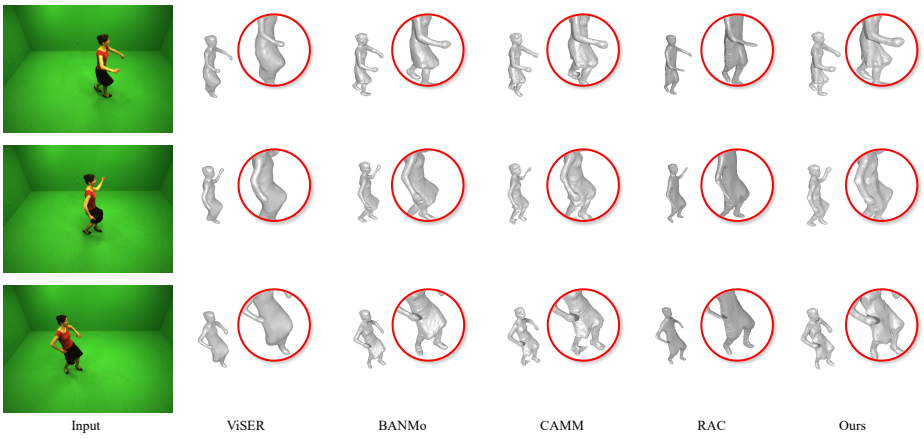

Fig. 16: Skirt reconstruction of the Samba dataset. Template-free methods excel in reconstructing regions where templates are not provided.

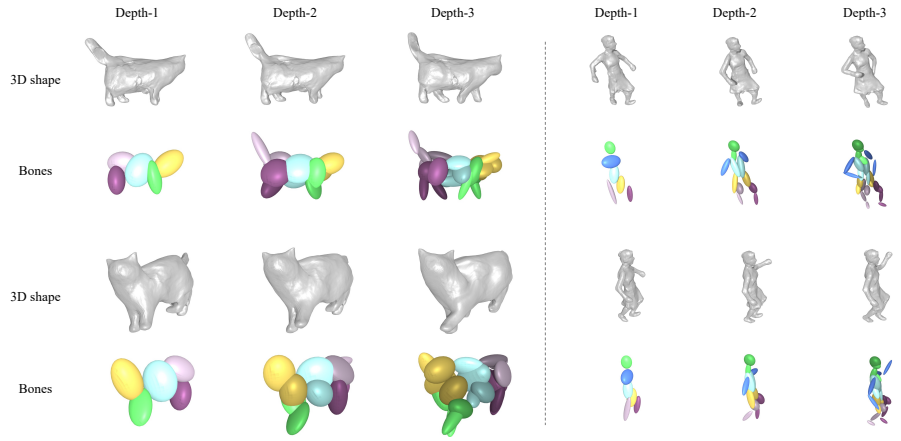

Fig. 17: Motion specification along depths.

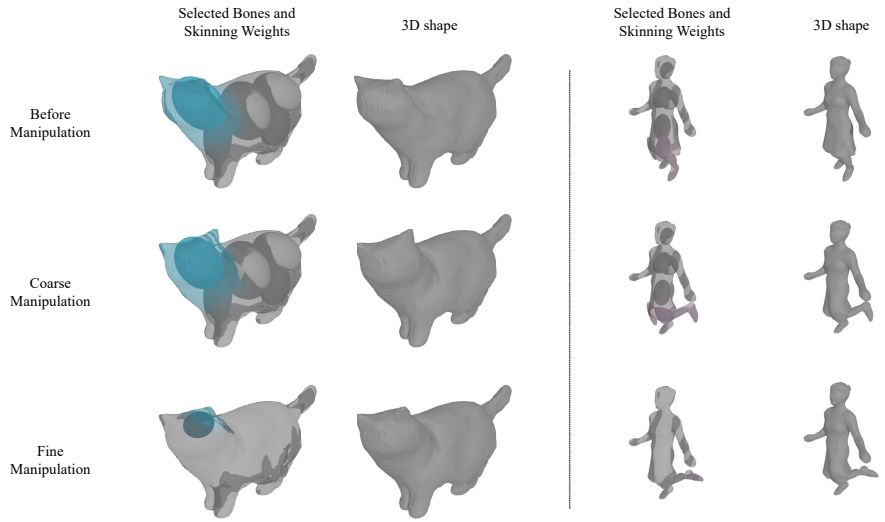

Fig. 18: Results of coarse-to-fine manipulation.

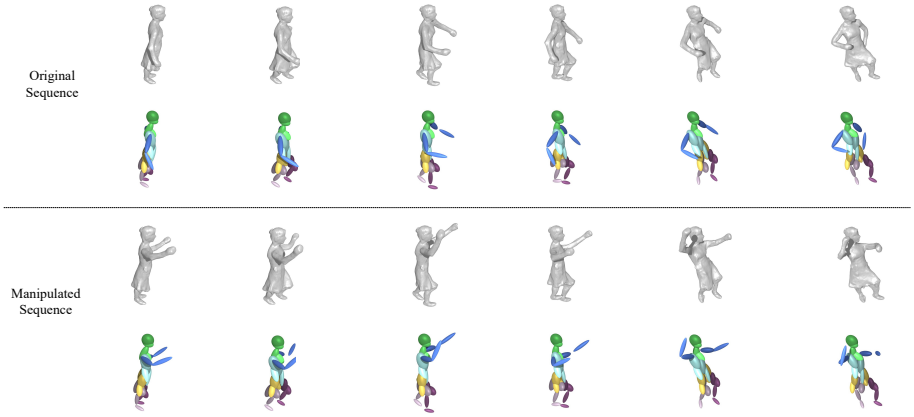

Fig. 19: Coarse-only manipulation results.

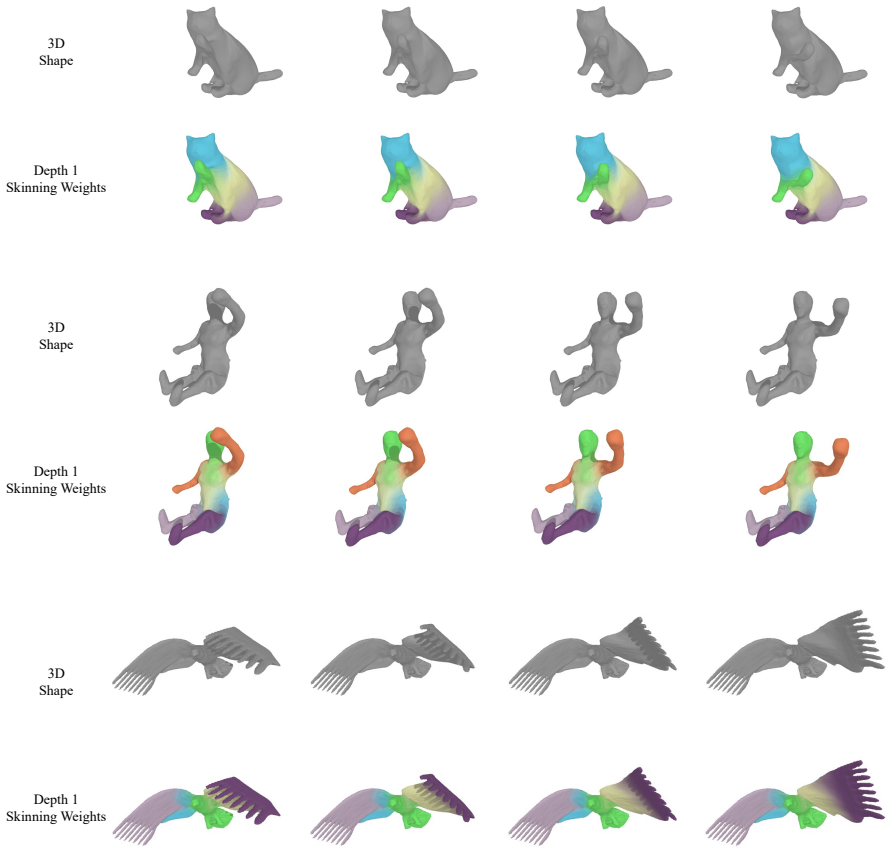

Fig. 20: Manipulation results on the diverse categories of objects.

References

1. Kuai, T., Karthikeyan, A., Kant, Y., Mirzaei, A., Gilitschenski, I.: Camm: Building category-agnostic and animatable 3d models from monocular videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6586–6596 (2023)
2. Li, J., Song, Z., Yang, B.: Nvfi: Neural velocity fields for 3d physics learning from dynamic videos. Advances in Neural Information Processing Systems **36** (2024)
3. Neverova, N., Novotny, D., Szafraniec, M., Khalidov, V., Labatut, P., Vedaldi, A.: Continuous surface embeddings. Advances in Neural Information Processing Systems **33**, 17258–17270 (2020)
4. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. In: Acm Siggraph 2008 papers, pp. 1–9 (2008)
5. Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Liu, C., Ramanan, D.: Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. Advances in Neural Information Processing Systems **34**, 19326–19338 (2021)
6. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., Joo, H.: Banmo: Building animatable 3d neural models from many casual videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2863–2873 (2022)
7. Yang, G., Wang, C., Reddy, N.D., Ramanan, D.: Reconstructing animatable categories from videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16995–17005 (2023)