4D-Animal: Freely Reconstructing Animatable 3D Animals from Videos

Shanshan Zhong[1,2,*], **Jiawei Peng**[1], **Zehan Zheng**[1], **Zhongzhan Huang**[2], **Wufei Ma**[1], **Guofeng Zhang**[1], **Qihao Liu**[1], **Alan Yuille**[1], **Jieneng Chen**[1,†]
[1]Johns Hopkins University [2]Sun Yat-sen University

Contents

[*] Work done while visiting at JHU.

[†] Corresponding author.

1. The novelty of 4D-Animal

The primary motivation behind 4D-Animal arises from the limitations of existing model-based methods for animatable 3D animal reconstruction, which still rely on sparse semantic keypoints. Annotating such keypoints is labor-intensive, and keypoint detectors, often trained on limited animal datasets, suffer from unreliability. Even the most recent model-based methods [18] that incorporate CSE [14] still require some degree of keypoint supervision, which inherently limits their robustness, generalizability, and scalability.

In contrast, 4D-Animal departs from this paradigm by leveraging well-trained 2D vision models to eliminate the need for sparse keypoint annotations, enabling robust and animatable 3D reconstruction from video. Our contributions are twofold:

- **Efficient feature mapping for SMAL fitting.** We propose a simple-yet-effective feature network that maps 2D features from pre-trained vision models to SMAL parameters. This design ensures high-quality feature input, improving both the efficiency and accuracy of the SMAL fitting process. Unlike prior work [18], which often relies on generic feature representation, our feature mapping is specifically optimized for animatable 3D animal reconstruction, balancing simplicity and performance.
- **Hierarchical alignment for robust reconstruction.** To further enhance the robustness of video-based reconstruction, we propose a hierarchical alignment strategy that operates across pixel-to-vertex correspondences, body part segmentations, and point trajectories. This strategy enables joint optimization of shape, pose, and motion in the absence of annotated keypoints, providing strong geometric and temporal cues derived solely from visual evidence.

Extensive experiments demonstrate that 4D-Animal outperforms both model-based and model-free baselines, achieving superior reconstruction quality without requiring sparse keypoint annotations. Additionally, the 3D assets generated by our method are beneficial for downstream tasks, highlighting its potential for scalable and generalizable animal modeling.

Finally, we emphasize that while 2D cues such as pixel-to-vertex correspondence [6, 19, 21], part segmentation [1, 11, 12], and point tracking [9, 23, 27], are well-established tools in computer vision and have seen widespread application in 3D/4D reconstruction, their systematic and task-driven integration into a keypoint-free 4D animal reconstruction pipeline is novel. Our method is distinguished by its careful integration of these components, specifically tailored to address the unique challenges of animatable animal shape and motion reconstruction.

2. The differences between 4D-Animal and Avatar

While both 4D-Animal and Animal Avatars [18] aim to reconstruct animatable 3D animals, they differ fundamentally in their reliance on sparse keypoints, input features, and hierarchical supervision. Avatars integrate Continuous Surface Embeddings and still require sparse keypoints to guide SMAL fitting, whereas 4D-Animal eliminates the need for keypoints entirely by leveraging dense 2D visual features and hierarchical alignment cues. Especially, 4D-Animal incorporates part-level and temporal alignment to improve motion coherence and robustness, features that are absent in Avatars. The two methods also differ in how camera initialization and mesh fitting are handled, with 4D-Animal using part masks in addition to CSE for improved alignment. Table **??** summarizes these differences in representation, camera initialization, and learning formulation.

Table 1. Comparison between 4D-Animal and Animal Avatars [18]. 4D-Animal eliminates the need for sparse keypoints, leverages dense 2D features, and integrates hierarchical alignment cues, including part-level and temporal supervision, while Avatars still rely on keypoints and CSE for alignment.

Category		4D-Animal (ours)	Avatars [18]
Representation	3D shape representation	Based on DINO features	Based on learnable parameters
	w/o Continuous embeddings	✓	×
	Texture reconstruction	✓	✓
	Double-mesh rendering	✓	✓
Camera initialization	PnP with CSE	✓	✓
	PnP with Part Mask	✓	×
Learning formulation	w/o Sparse-keypoint loss	✓	×
	Part mask loss	✓	×
	Temporal loss	✓	×
	Object mask loss	✓	✓
	Pixel loss.	✓	✓
	Photometric loss	✓	✓

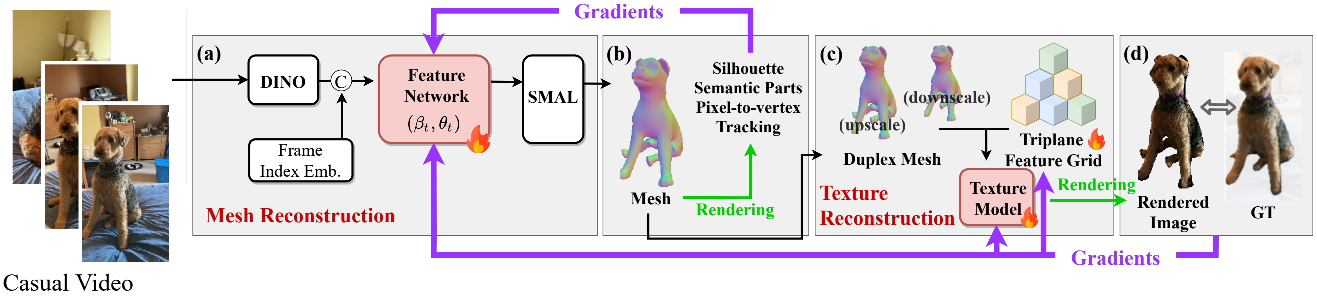

Figure 1. The overview of proposed 4D-Animal. (a) Mesh reconstruction. “C" stands for concatenation. “Frame Index Emb." refers to the embedding of the index of a video frame. (β_t, θ_t) are inputs to SMAL for t-th frame. The feature network is learnable. (b) Hierarchical geometric alignment. The alignment loss is constructed using silhouette, semantic part, pixel, and tracking information generated by a series of 2D vision pre-trained models. (c) Texture reconstruction. The RGB texture is reconstructed through a learnable texture model and triplane feature grid based on the duplex mesh. (d) Texture loss.

3. Additional details of 4D-Animal

3.1. The details of texture model

High-quality texture requires an exact supporting 3D shape. However, mesh m_t reconstructed by SMAL has only 3,889 vertices and 7,774 triangular faces, so directly modeling vertex texture lacks expressivity. Therefore, following recent advances in new-view synthesis of humans [3, 4] and the design of animals [18], we leverage m_t as a scaffold supporting a more accurate implicit radiance field.

We first extend SMAL mesh into a 3D volume by interpolating arbitrary face attributes between a canonical SMAL mesh and its deformed instances at multiple scales. Specifically, as shown in Fig. 1 (c), given a posed SMAL mesh m_t parameterized by shape and pose, we construct both an upscaled and a downscaled version, ensuring a local geometric neighborhood around the canonical surface. By rasterizing both meshes under a differentiable renderer, we identify per-pixel surface intersections and interpolate their precise 3D locations using precomputed canonical vertex structures. This results in a structured mapping, where each pixel is associated with two 3D points, providing a robust foundation for texture reconstruction.

Then, the rendering process implemented by Lightplane [3] follows a volumetric Emission-Absorption model inspired by NeRF, integrating a hybrid representation to efficiently reconstruct appearance. Given a set of rays, each sampled at multiple points along its path, the model estimates per-point features by querying a learned representation (triplane feature grid as shown in Fig. 1 (c)) and refines them via MLP-based decoding. The transmittance along each ray is computed to model light attenuation, ensuring physically consistent color composition at the image plane. By leveraging a structured decomposition of feature extraction and transformation, the approach maintains compatibility with powerful hybrid representations while significantly reducing memory overhead. This enables high-fidelity color reconstruction with improved computational efficiency.

3.2. The details of feature network

The architecture of the feature network shown in Fig. 1 (a) is highly streamlined. We use a single-layer linear network as the reduction layer to project the 2D representation and 8-dimensional positional encoding into 64-dimensional vectors. This is followed by a single-layer linear network with the same dimensionality, serving as the intermediate feature layer. Finally, another single-layer linear network acts as the output layer, mapping the 2D representation to the SMAL parameters.

3.3. The details of learning formulation

3.3.1. Object-level alignment

For in-the-wild videos, object masks can be easily generated by a well-trained segmentation model, i.e., SAM [15]. These masks provide coarse supervision by encouraging the projected mesh silhouette to match the animal’s outline in the image. Specifically, we use instance masks u_t to enforce alignment between the projected mesh and the animal silhouette via the Chamfer distance: $\mathcal{L}^{\text{obj}} = \sum_t D(u_t, P_t v_t)$, where P_t is the projection matrix, and v_t the mesh vertices. This ensures global shape coverage.

However, instance masks ignore the semantic consistency, for example, neglecting the difference between the head and tail of the dog as shown in Fig. 3 (b). This ambiguity can lead to incorrect pose estimation and unrealistic deformations. To address this limitation, we introduce additional coarse-to-fine guidance to enhance semantic alignment.

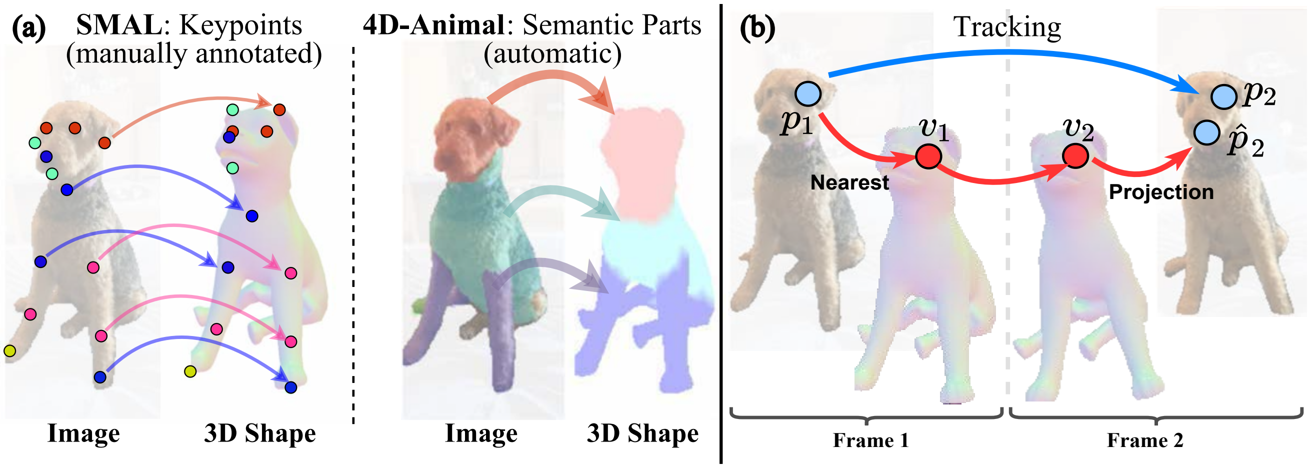

Figure 2. (a) **Left**: Keypoint alignment (manually annotated) . **Right**: Part-level alignment (automatic) using semantic masks divided into head, body, feet, and tail, which are mapped to corresponding SMAL mesh regions. (b) Temporal tracking alignment. A tracked 2D point p_1 in Frame 1 is matched to p_2 in Frame 2. The nearest vertex v_1 to p_1 is found, and its corresponding vertex v_2 is located via SMAL topology. The loss is computed between p_2 and the projection of v_2.

3.3.2. Points sampling for part-level alignment

At the *part level*, we utilize semantic part masks s_t predicted by PartGLEE [10], which segment the animals into head, body, feet, and tail as shown in Fig. 2 (a). In the COP3D videos, most frames contain clear head and body regions, but the feet and tail may be missing or occluded. To reduce the impact of inaccurate predictions, we retain feet and tail masks only when their confidence scores from PartGLEE exceed 0.3. Additionally, for blurry frames where no parts are detected, we reuse the part masks from the previous frame.

For each semantic mask s_t, we first compute the area of each part (head, body, feet, tail), and allocate the number of sampled points N_s proportionally. For example, the body may receive more samples than the head if it occupies a larger area. We then sample 2D points p_t^s uniformly within each part mask by randomly selecting pixel locations with equal probability. Each sampled point is assigned a semantic region label, and we identify the corresponding SMAL mesh vertices v_t^s using the part annotations defined on the SMAL model. Formally, the part-level loss is formulated as: $\mathcal{L}^{\text{part}} = \sum_t \|p_t^s - P_t v_t^s\|^2$. In our experiments, N_s is set as 200.

3.3.3. Pixel-to-vertex alignment via CSE to SMAL

At the *pixel level*, we use CSE [14] to obtain dense pixel-to-vertex correspondences. CSE produces a per-pixel coordinate map aligned to a predefined 3D template. Since this template differs from SMAL, we employ Zoom-Out [13] to construct a functional map between the two surfaces. Once the mapping is obtained, we transfer CSE outputs into the SMAL vertex space. Formally, we supervise the dense foreground pixel coordinates p_t^c by enforcing consistency with the corresponding projected mesh vertices v_t^c via the following loss: $\mathcal{L}^{\text{pix}} = \sum_t \|p_t^c - P_t v_t^c\|^2$, where P_t is the projection matrix at frame t.

During training, foreground pixels are selected using the instance mask, and only those with valid CSE predictions and confidence scores above 0.5 are retained. For blurry frames with no valid CSE output, we reuse predictions from the previous frame.

3.3.4. Temporal tracking via SMAL topology

At the *temporal level*, we use 2D tracking from BootsTAP [7]. BootsTAP is a model that can track any point on solid surfaces in a video. Specifically, We sample N_t points uniformly within instance mask of Frame t by randomly selecting pixel locations with equal probability, and track them in all frames of the video using BootsTAP. To ensure reliability, we filter any point whose tracked positions fall outside the instance mask in any frame, as this may indicate tracking drift or failure.

Furthermore, since COP3D videos consist of 200 frames captured by a fly-around camera to ensure full-body coverage, we initialize point tracking from four key frames: 1, 51, 101, and 151. For each key frame, we sample $N_t = 500$ points and track them across the entire sequence. This results in a total of 2000 tracked points per video. After filtering out unreliable trajectories, approximately 1000 valid ones are typically retained.

Next, as shown in Fig. 2(b), for each tracked 2D point $p_1 \rightarrow p_2$ obtained from BootsTAP, we first project all mesh vertices from Frame 1 to the screen and identify the vertex v_1 closest to p_1. Given that the meshes m_1 and m_2 from Frames 1 and 2 share the same topology, we locate the corresponding vertex v_2 in m_2 by matching the index of v_1. Projecting v_2 to the screen yields the estimated location $\hat{p}_2$. This allows us to enforce temporal consistency by penalizing the discrepancy between the tracked point p_2 and the projected vertex $\hat{p}_2$. Formally, for a set of 2D trajectories $p_t \rightarrow p'_t$ between frames t and t', we project the corresponding mesh vertices v'_t to the screen using the projection matrix $P_{t'}$, and define the temporal consistency loss as: $\mathcal{L}^{\text{time}} = \sum_t \|p'_t - P'_t v'_t\|^2$.

4. Additional details and results of experiments

4.1. Training procedure

As shown in Fig.3 (a), the initial 3D pose is often inaccurate. Applying all loss terms uniformly from the beginning can lead to unstable optimization. For instance, as shown in Fig.3 (b), the model may confuse semantic parts and prioritize silhouette coverage over precise alignment. To address this, we adopt a multi-stage fitting strategy.

In the early stage of training, we focus on stabilizing the global pose. We reduce the weight of the object-level alignment loss (λ^{obj}) and the temporal level loss (λ^{time}), and disable vertex offsets in the SMAL template. This encourages coarse alignment of the overall shape without introducing unstable local deformations or enforcing premature temporal constraints.

In the later stage, once the global pose has stabilized, we begin refining local geometry. Vertex offsets are enabled to allow detailed shape adjustment, especially along the silhouette. Meanwhile, the weights of pixel-level and part-level losses (λ^{pixel}, λ^{part}) are gradually reduced to avoid overfitting to noisy or ambiguous regions. In contrast, the temporal consistency loss (λ^{time}) is progressively increased to enforce smooth motion and coherent alignment across frames. As shown in Fig. 3 (c), this leads to more accurate and temporally consistent reconstructions.

In our experiments, training runs for 10,000 epochs. The loss weights are scheduled as follows:

- λ^{obj}: 1, 100, 500, 800 at milestones 300, 1000, 6000.
- λ^{part}: 5e-4, 5e-8 at milestones 300.
- λ^{pixel}: 5, 1e-1, 1e-2 at milestones 1000, 2000.
- λ^{time}: 5e-4, 5e-2, 5, 50, 100, 300 at milestones 300, 1000, 2000, 5000, 8000.

As shown in Fig. 4, although our performance is initially lower than that of Avatars during the early fitting stage, it quickly surpasses Avatars and ultimately achieves the target metric values with significantly higher efficiency.

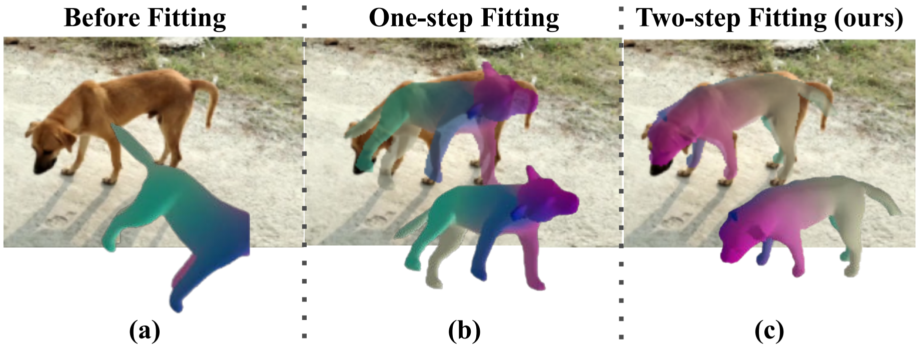

Figure 3. (a) The initial pose of the mesh is often inaccurate. (b) In one-step fitting, the model tends to confuse the head and tail, focusing primarily on silhouette coverage. (c) Our two-step fitting strategy enables a more precise alignment, resulting in higher-quality reconstruction.

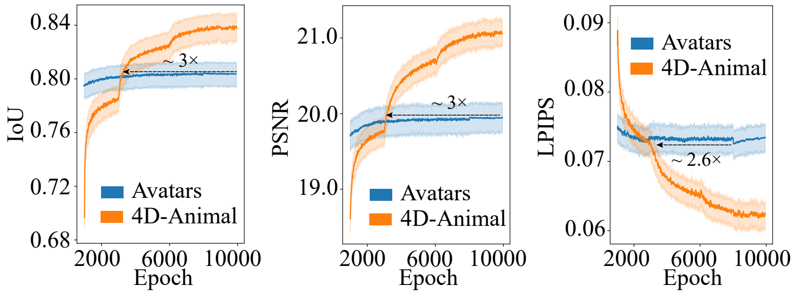

Figure 4. Comparison of training efficiency between 4D-Animal and Avatars.

4.2. Implementation details

We evaluate all models on COP3D [20], a publicly available dataset featuring fly-around videos of pets, with annotated camera parameters and object masks. Following [18], we select a subset of 50 dog videos capturing diverse poses, motions, and textures. Each test video contains 200 frames, which we split into the training and test sets by considering contiguous blocks of 15 frames as train, interleaved by blocks of 5 frames as test. The videos we use are the same as those used in Avatars [18]. We also use casual videos with depth maps of dogs from TracksTo4D [8] to evaluate the reconstructed structure.

We employ the Adam optimizer and apply a learning rate decay to each parameter group by a factor of γ when the epoch count reaches predefined milestones. For texture reconstruction, γ is set to 0.5 with milestones at [9000, 9500]. The number of epochs is set to 10000. The batch size is set as 32. The training and evaluation of all models is conducted using the RTX 6000 GPU (48G).

4.3. Introduction of baselines

We compare 4D-Animal against the state-of-the-art model-based counterparts, including Avatars [18], BARC [16], and BITE [17], as well as the model-free approach RAC [28].

- BARC [16] recovers the 3D shape and pose of dogs from a single image by leveraging breed information. It modifies the SMAL animal model to better represent dog shapes and addresses the challenge of limited 3D training data by incorporating breed-specific losses during training.
- BITE [17] improves 3D dog pose estimation by introducing a dog-specific model (D-SMAL) and leveraging ground contact constraints to refine poses. A neural network enhances initial predictions by integrating contact information, enabling realistic reconstructions of complex postures.
- RAC [28] reconstructs animatable 3D models of object categories like humans, cats, and dogs from monocular videos by disentangling morphology (shape variations) and articulation (motion over time). It achieves this by learning a category-level skeleton, using latent space regularization to maintain instance-specific details, and leveraging 3D background models for better segmentation.
- Avatars [18] reconstructs animatable 3D dog models from monocular videos by jointly optimizing shape, pose, and texture in a canonical space. It improves model-based reconstruction using Continuous Surface Embeddings for dense keypoint supervision and introduces a duplex-mesh implicit texture model for realistic appearance.

4.4. Quantitative results of Avatars

To ensure a fair comparison, we re-run the official codebase of Avatars [17] using the same experimental setup as our method. The numerical results reported in Table 1 are based on these re-runs. We observe that our re-run results are slightly lower than those reported in the original paper. This gap may come from several factors, including:

- **Incomplete configuration details**: Some hyperparameters, such as loss weights or learning rate schedules, may not be fully specified in the released codebase.
- **Differences in environment**: Variations in hardware, software versions, or random seeds can lead to slight performance fluctuations.

Despite the discrepancy, we use our re-run results for Avatars to maintain fairness, since all other re-run baselines are evaluated under the same experimental setup.

4.5. Qualitative comparison of hierarchical alignment

The quantitative impact of hierarchical alignment is presented in Section 6 of the main text. Here, we provide a qualitative analysis based on the ablation study to further understand the contribution of each alignment component.

First, as shown in Fig. 6, removing the part-level loss L_{part} leads to failures in cases involving fast camera or animal motion. In such scenarios, temporal tracking and pixel-to-vertex correspondences are often unreliable, making semantic part masks an essential source of guidance.

Second, Fig. 7 shows that while the 4D-Animal without pixel-to-vertex supervision (L_{pix}) can still recover the overall pose, it struggles to capture finer details such as the accurate articulation of the head and legs.

Finally, as shown in Fig. 5, the temporal consistency loss L_{tracking} helps maintain coherent leg behavior over time. Without L_{tracking}, 4D-Animal may reconstruct inconsistent poses across frames. For instance, reconstructing crossed legs in later frames that contradict the pose observed earlier. In contrast, incorporating L_{tracking} ensures temporal consistency, preventing such contradictions and leading to accurate and stable reconstructions.

In summary, each component of the hierarchical alignment plays a crucial role in handling different challenges, and their combination improves the robustness of 4D-Animal in complex video scenarios.

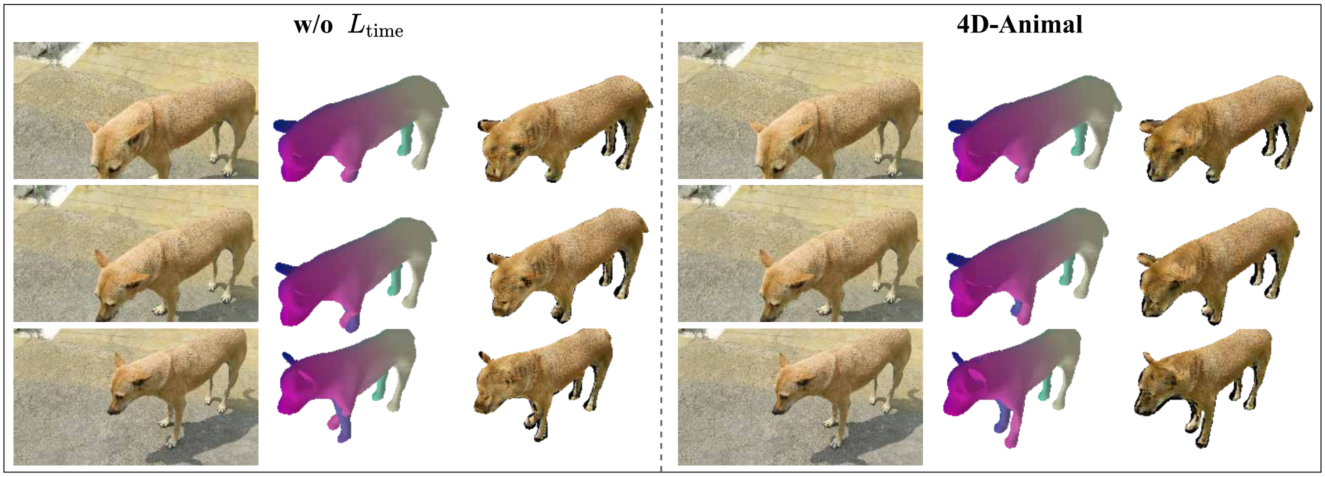

Figure 5. Qualitative ablation of temporal-level alignment L_{time}.

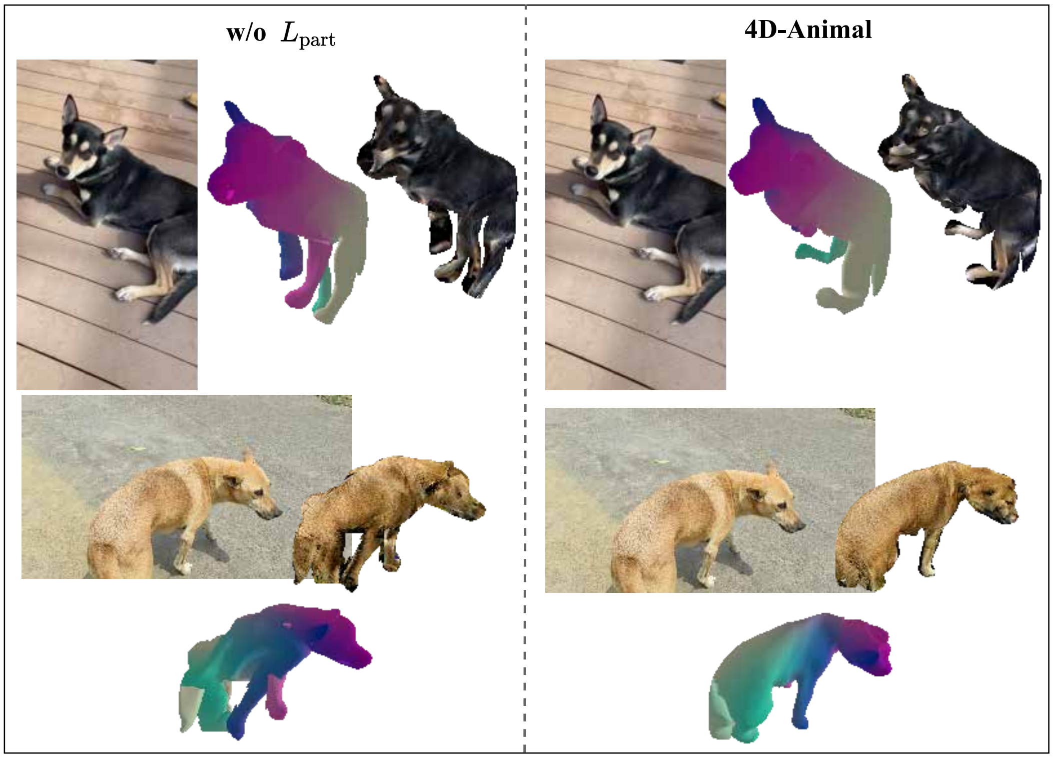

Figure 6. Qualitative ablation of part-level alignment L_{part}.

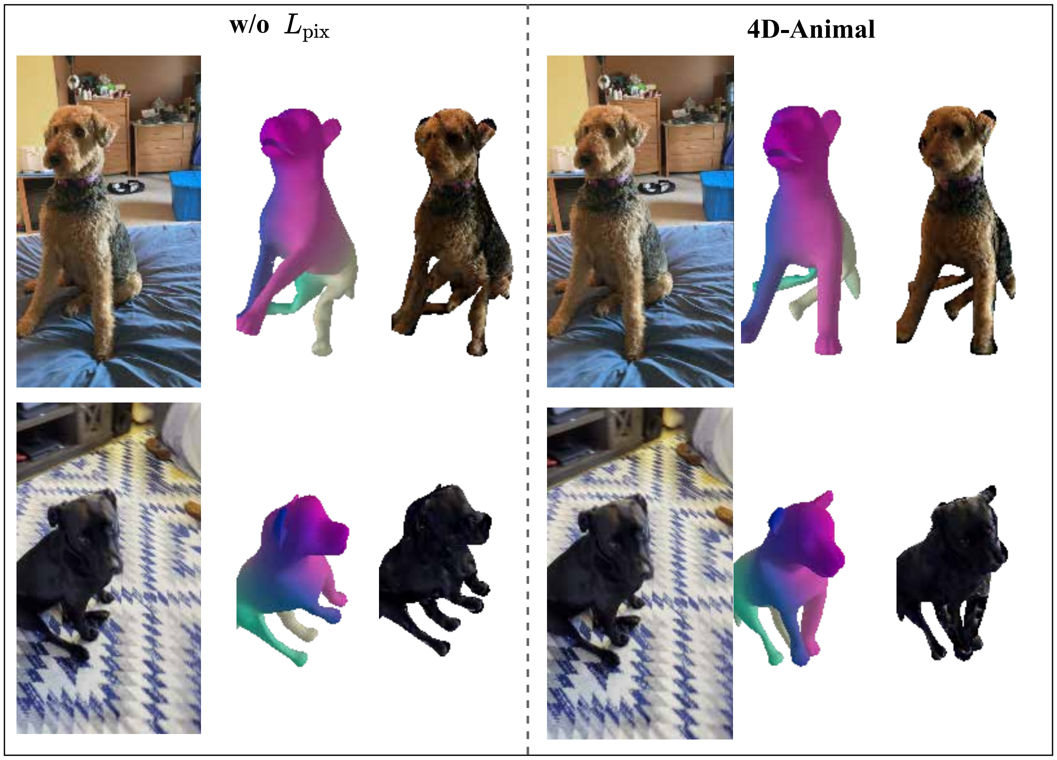

Figure 7. Qualitative ablation of pixel-level alignment L_{pix}.

4.6. Novel view of animatable 3D assets

We provide additional materials showcasing novel views of the animatable 3D assets reconstructed using our 4D-Animal, as shown in Fig. 8, to offer a comprehensive evaluation of 4D-Animal's performance. These views highlight 4D-Animal's ability to generate consistent reconstructions from previously unseen perspectives, further demonstrating the robustness and generalization capability of 4D-Animal.

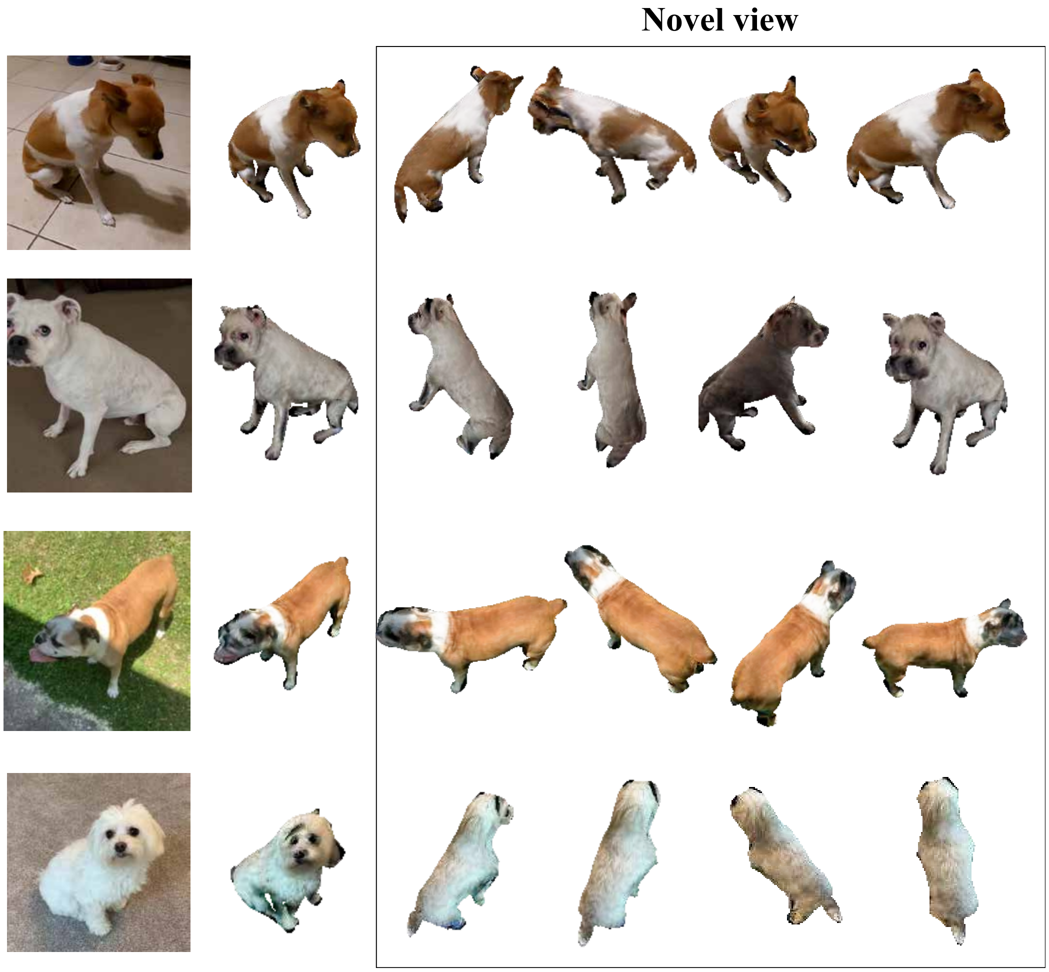

Figure 8. Novel views of animatable 3D assets reconstructed using 4D-Animal.

4.7. Reconstruction on videos of other animal types

Why use a dog-specific SMAL model? In our experiments, we implement 4D-Animal using the state-of-the-art dog-specific SMAL template [17] to ensure a focused and comprehensive evaluation, minimizing external factors such as template limitations. However, we acknowledge the importance of generalizing our method to other animal categories and species, and we clarify that the core of our method is indeed designed to be broadly applicable.

Why can our method be generalized to other animal categories? The key components of our pipeline, including hierarchical alignment cues such as object masks, semantic part masks, pixel-to-vertex correspondences, and motion tracking, are not tied to any specific animal species. These cues can be applied across different quadruped species without significant modification. Our method relies on dense feature networks that align 2D representations to SMAL parameters, and since these features are general and not category-dependent, they are expected to work similarly across other quadrupeds.

How does our method perform on other animal categories? We conduct preliminary tests of our method on cat videos from the COP3D dataset, as shown in Fig.9. Despite using a dog-specific template, 4D-Animal still reconstructs reasonable mesh fittings to the 2D images. Although some details such as the head are less accurate, the overall results remain promising.

Although our experiments focus on dogs as a proof-of-concept, the proposed approach is inherently generalizable. We believe it holds strong potential for extension to a wide range of quadruped animals.

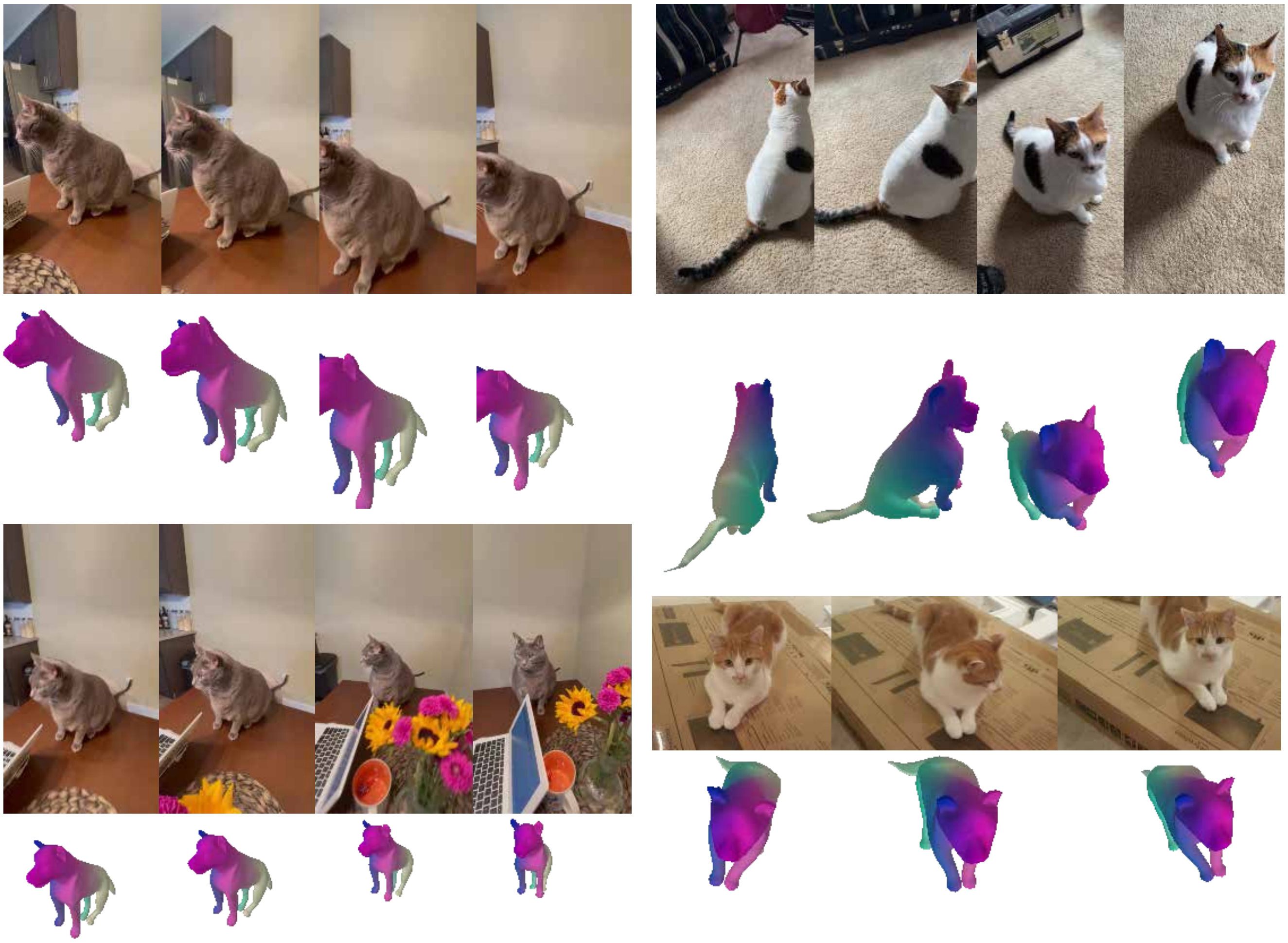

Figure 9. Qualitative results of 4D-Animal on cat videos from COP3D. Despite using a dog-specific template, the method achieves reasonable mesh fittings, demonstrating its potential generalization to other animal categories.

4.8. Additional comparison of reconstruction quality and temporal consistency

We provide further qualitative comparisons to evaluate the reconstruction quality of our 4D-Animal method against existing methods as shown in Fig. 10, 11, 12, 13, 14. For a more detailed visualization and an evaluation of temporal consistency, please refer to the **supplementary videos**.

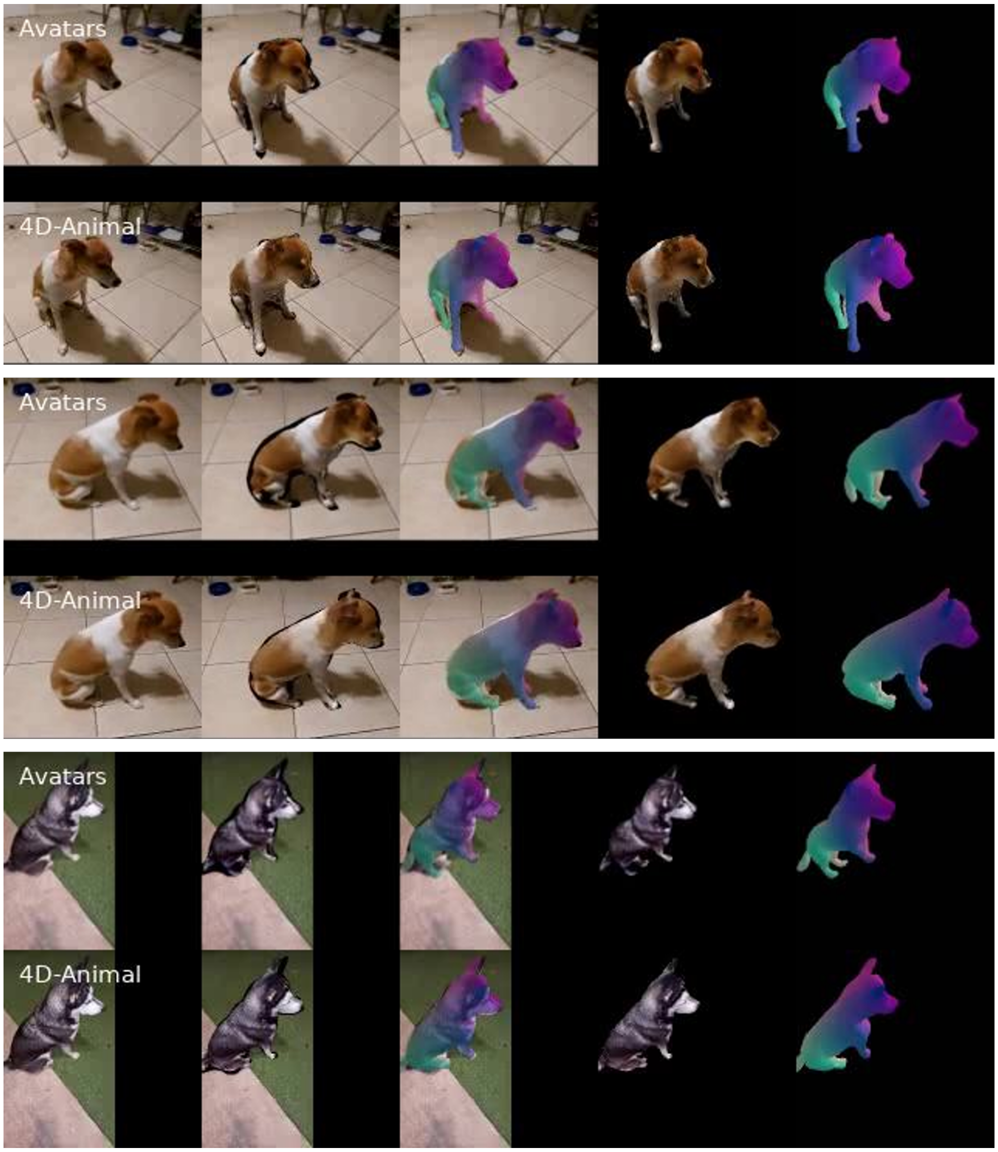

Figure 10. Qualitative comparison. We compare our 4D-Animal with the state-of-the-art model Avatars [18] by selecting images from videos. The **first column** shows the original images, the **second column** presents the reconstructed appearance overlaid on the original image, and the **third column** displays the reconstructed 3D shape overlaid on the original image. The **fourth** and **fifth columns** show the standalone rendered appearance and 3D shape, respectively.

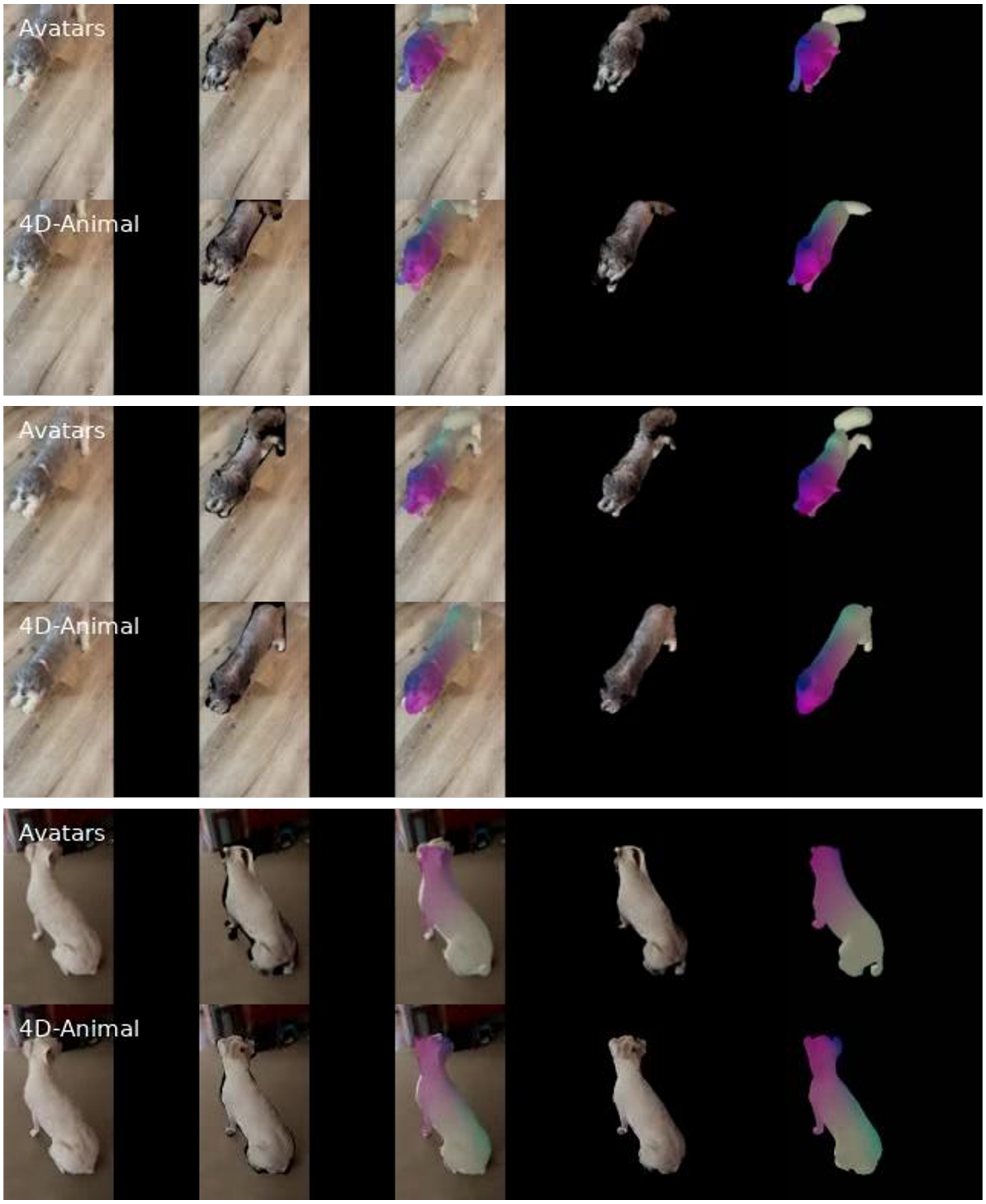

Figure 11. Qualitative comparison. We compare our 4D-Animal with the state-of-the-art model Avatars [18] by selecting images from videos. The **first column** shows the original images, the **second column** presents the reconstructed appearance overlaid on the original image, and the **third column** displays the reconstructed 3D shape overlaid on the original image. The **fourth** and **fifth columns** show the standalone rendered appearance and 3D shape, respectively.

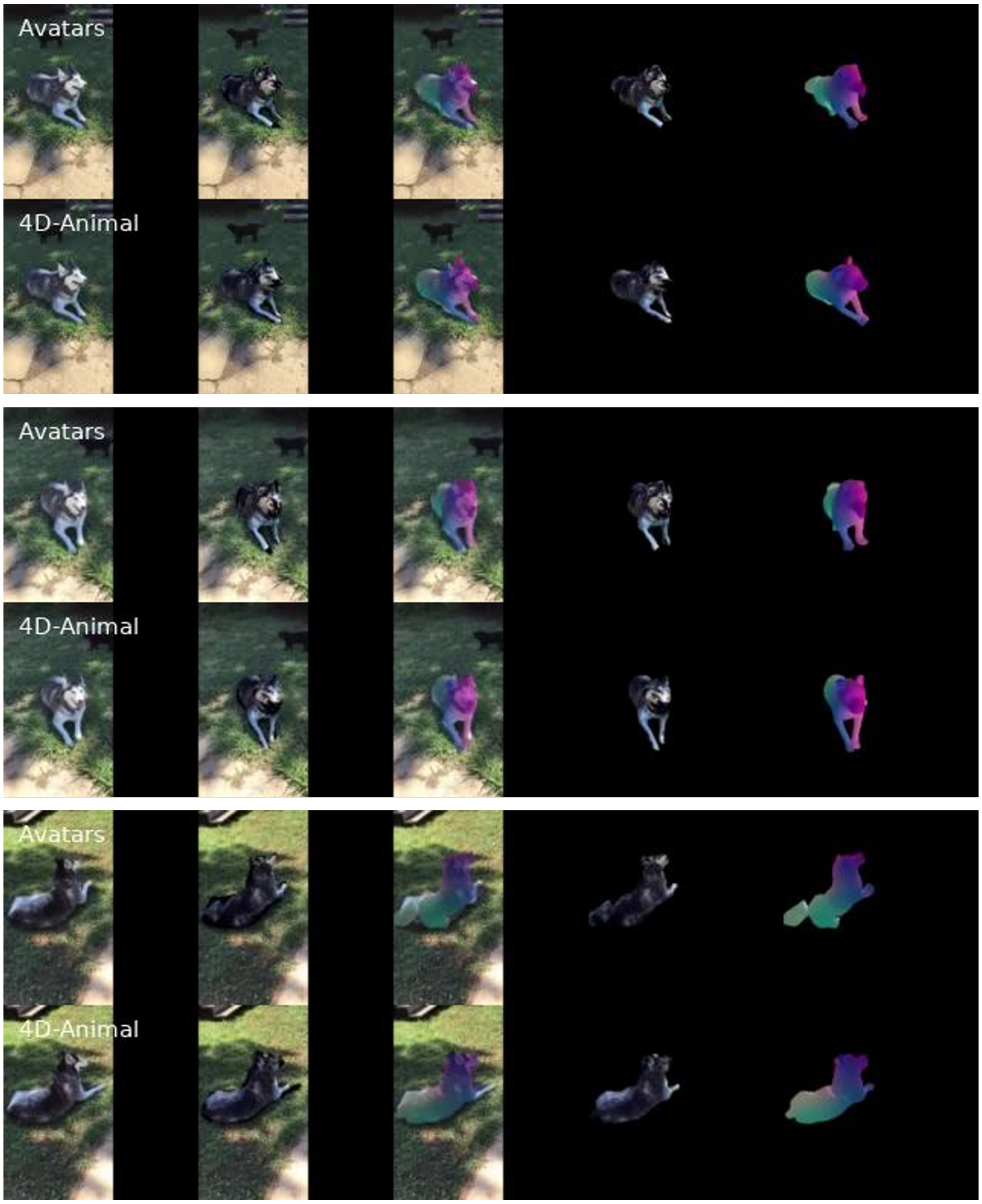

Figure 12. Qualitative comparison. We compare our 4D-Animal with the state-of-the-art model Avatars [18] by selecting images from videos. The **first column** shows the original images, the **second column** presents the reconstructed appearance overlaid on the original image, and the **third column** displays the reconstructed 3D shape overlaid on the original image. The **fourth** and **fifth columns** show the standalone rendered appearance and 3D shape, respectively.

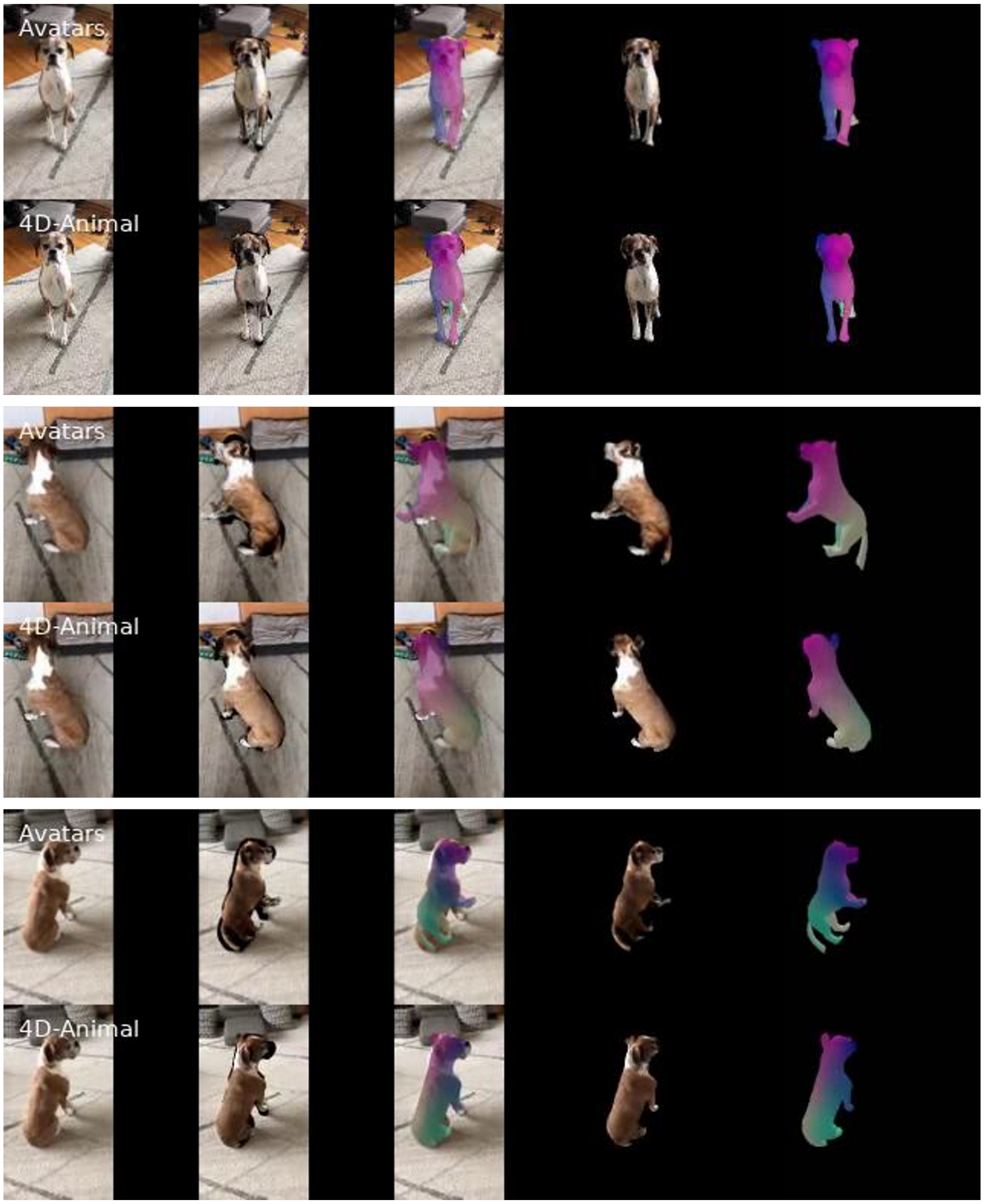

Figure 13. Qualitative comparison. We compare our 4D-Animal with the state-of-the-art model Avatars [18] by selecting images from videos. The **first column** shows the original images, the **second column** presents the reconstructed appearance overlaid on the original image, and the **third column** displays the reconstructed 3D shape overlaid on the original image. The **fourth** and **fifth columns** show the standalone rendered appearance and 3D shape, respectively.

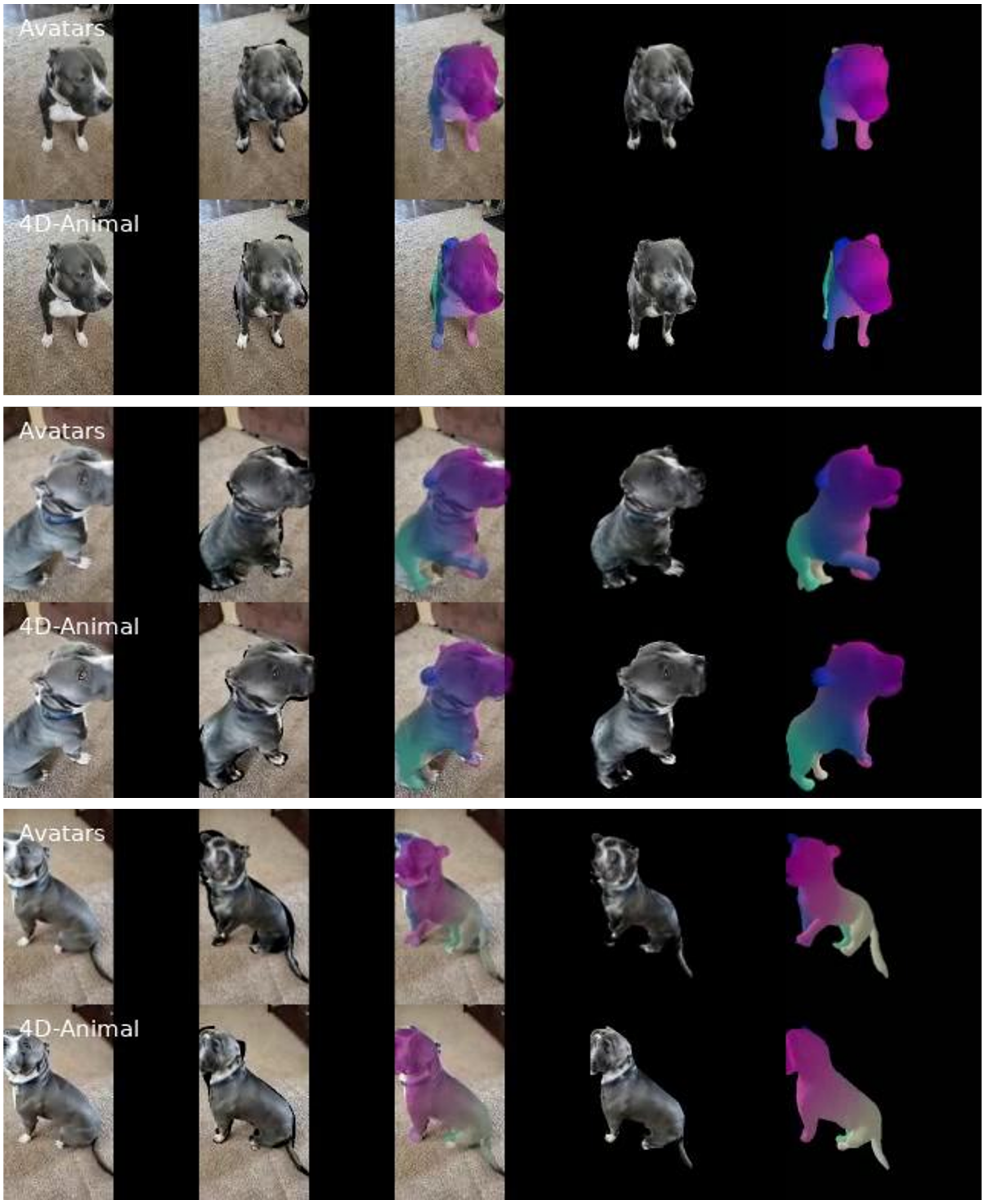

Figure 14. Qualitative comparison. We compare our 4D-Animal with the state-of-the-art model Avatars [18] by selecting images from videos. The **first column** shows the original images, the **second column** presents the reconstructed appearance overlaid on the original image, and the **third column** displays the reconstructed 3D shape overlaid on the original image. The **fourth** and **fifth columns** show the standalone rendered appearance and 3D shape, respectively.

5. Analysis of state-of-the-art image-to-3D models

As we have mentioned in Sec. 6 (main text), we observe that current 3D generative models trained on synthetic datasets often fail when processing real-world animal images. Here, we assess the performance of a wider variety of 3D models beyond LGM [22] as shown in Fig 15, such as SF3D [2], TriplaneGaussian [29], CRM [24], 3DTopoa-XL [5], InstantMesh [26] and TRELLIS [25]. There is still a gap between such models and model-based methods, but we can take full advantage of both. Fine-tuning existing large models using 4D-Animal reconstruction results of real-world casual videos could be very efficient and does not limit the representation of the models. This highlights that our work could form an essential contribution to the image-to-3D community, complementing existing efforts.

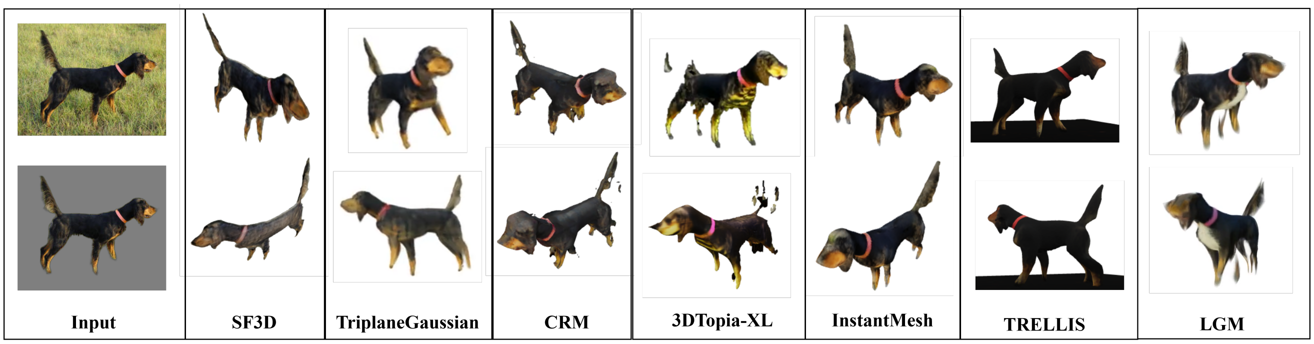

Figure 15. The performance of image-to-3D models SF3D [2], TriplaneGaussian [29], CRM [24], 3DTopoa-XL [5], InstantMesh [26], TRELLIS [25], LGM [22] on animal images.

References

[1] Mehmet Aygun and Oisin Mac Aodha. Saor: Single-view articulated object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10382–10391, 2024. 2

[2] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint arXiv:2408.00653*, 2024. 16

[3] Ang Cao, Justin Johnson, Andrea Vedaldi, and David Novotny. Lightplane: Highly-scalable components for neural 3d fields. *arXiv preprint arXiv:2404.19760*, 2024. 3

[4] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 3

[5] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: High-quality 3d pbr asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957*, 8, 2024. 16

[6] Sergio M de Paco and Antonio Agudo. 4dpv: 4d pet from videos by coarse-to-fine non-rigid radiance fields. In *Proceedings of the Asian Conference on Computer Vision*, pages 2596–2612, 2024. 2

[7] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, João Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pages 3257–3274, 2024. 4

[8] Yoni Kasten, Wuyue Lu, and Haggai Maron. Fast encoder-based 3d from casual videos via point track processing. *arXiv preprint arXiv:2404.07097*, 2024. 6

[9] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 2

[10] Junyi Li, Junfeng Wu, Weizhi Zhao, Song Bai, and Xiang Bai. Partglee: A foundation model for recognizing and parsing any objects. In *European Conference on Computer Vision*, pages 475–494. Springer, 2024. 4

[11] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 677–693. Springer, 2020. 2

[12] Di Liu, Anastasis Stathopoulos, Qilong Zhangli, Yunhe Gao, and Dimitris Metaxas. Lepard: Learning explicit part discovery for 3d articulated shape reconstruction. *Advances in Neural Information Processing Systems*, 36:54187–54198, 2023. 2

[13] Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. 4

[14] Natalia Neverova, David Novotny, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. 2020. 2, 4

[15] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3

[16] Nadine Rueegg, Silvia Zuffi, Konrad Schindler, and Michael J Black. Barc: Learning to regress 3d dog shape from images by exploiting breed information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3876–3884, 2022. 6

[17] Nadine Rüegg, Shashank Tripathi, Konrad Schindler, Michael J Black, and Silvia Zuffi. Bite: Beyond priors for improved three-d dog pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8867–8876, 2023. 6, 10

[18] Remy Sabathier, Niloy J Mitra, and David Novotny. Animal avatars: Reconstructing animatable 3d animals from casual videos. In *European Conference on Computer Vision*, pages 270–287. Springer, 2024. 2, 3, 6, 11, 12, 13, 14, 15

[19] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. Shic: Shape-image correspondences with no keypoint supervision. In *European Conference on Computer Vision*, pages 129–145. Springer, 2024. 2

[20] Samarth Sinha, Roman Shapovalov, Jeremy Reizenstein, Ignacio Rocco, Natalia Neverova, Andrea Vedaldi, and David Novotny. Common pets in 3d: Dynamic new-view synthesis of real-life deformable categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4881–4891, 2023. 6

[21] Leonhard Sommer, Artur Jesslen, Eddy Ilg, and Adam Kortylewski. Unsupervised learning of category-level 3d pose from object-centric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22787–22796, 2024. 2

[22] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 16

[23] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 2

[24] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision*, pages 57–74. Springer, 2024. 16

[25] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 16

[26] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 16

[27] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. *Advances in Neural Information Processing Systems*, 34:19326–19338, 2021. 2

[28] Gengshan Yang, Chaoyang Wang, N Dinesh Reddy, and Deva Ramanan. Reconstructing animatable categories from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16995–17005, 2023. 6

[29] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10324–10335, 2024. 16