Supplementary Material for ARTIC3D: Learning Robust Articulated 3D Shapes from Noisy Web Image Collections

Chun-Han Yao[1*] **Amit Raj**[2] **Wei-Chih Hung**[3] **Yuanzhen Li**[2] **Michael Rubinstein**[2]
Ming-Hsuan Yang[124] **Varun Jampani**[2]

[1]UC Merced [2]Google Research [3]Waymo [4]Yonsei University

In this supplementary document, we present the implementation details (Sec. 1), ablative analyses (Sec. 2), additional results including user study (Sec. 3), and dataset details (Sec. 4). We also provide a short video to explain our framework with illustrations and visual results. Given the 3D nature of our results, we encourage readers to see the supplementary video for better visualization of our 3D reconstructions.

1 Implementation Details

We implement the ARTIC3D framework using PyTorch [8], and optimize all parameters using an Adam optimizer [5]. For an image ensemble containing 30 images, the overall optimization takes roughly 45 minutes on a single GTX 1080 GPU.

1.1 Input preprocessing

To enhance the quality of noisy web images, we perform input preprocessing via DASS with the number of accumulation steps $n = 20$, noise timestep $t = 0.3$, and guidance weight $w_g = 15$. Similar to [1, 12, 11], we obtain 2D semantic features and silhouette estimates from a trained DINO-ViT [2] network. Specifically, we extract the 384 dimensional *key* from the last layer of ViT-S8. Likewise, we estimate a rough foreground mask via the average attention map of *class* tokens. Considering that the features and silhouettes tend to be quite noisy in the presence of occlusions or truncation, we apply our DASS module as image enhancement. Finally, we cluster the foreground features using an off-the-shelf K-means algorithm (8 clusters) and obtain pseudo ground-truth masks via dense CRF filtering [6]. Fig. 1 shows some sample results of image enhancement and DINO feature clustering.

1.2 3D skeleton and part MLPs

We adopt Hi-LASSIE [11] to automatically discover a 3D skeleton given a reference image in the collection. The skeleton specifies the initial joint coordinates, bone connection, primitive part shapes, and 3D symmetry plane. For each 3D part, we then construct a neural surface corresponding to individual skeleton bones. Similar to Hi-LASSIE [11], we adopt frequency-decomposed MLPs to regularize shared and instance-specific surface deformation. That is, the part MLPs are composed of multiplicative layers as in BACON [7]. Each layer encodes a surface point x via positional encoding (PE) as: $\text{PE}_i(x) = \sin(\omega_i x + \phi_i)$, where $i = 0, ..., L$ ($L = 9$ in our experiments). The frequencies are pre-defined as $\omega_i = 0.25\pi \cdot i$ and fixed during training. The decomposition allows us to constrain low-frequency base part shapes during per-instance optimization by fixing the early layers of part MLPs. Please refer to the Hi-LASSIE [11] papers for more details. All animal skeletons in our experiments contain roughly 8-13 parts, and we densely sample 2562 vertices per part for rendering.

[*]Work done as a student researcher at Google.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

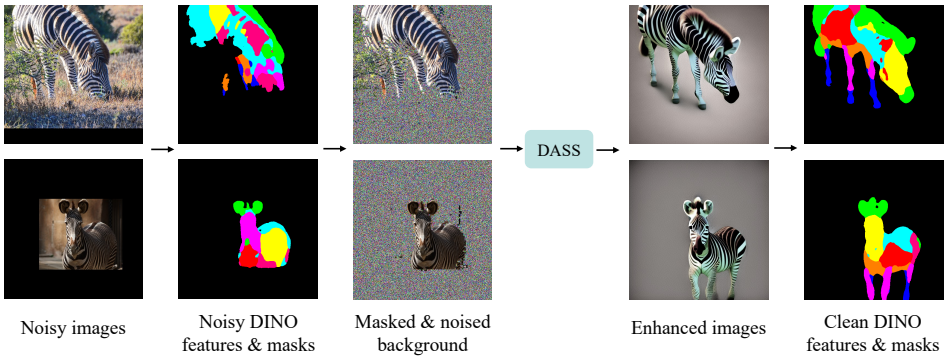

Figure 1: **Image enhancement via DASS.** Our input preprocessing approach can effectively deal with occlusions or truncation by producing images with complete shapes and detailed texture.

1.3 3D shape and texture optimization

Given the preprocessed images and initial 3D skeleton, we optimize the shared and instance-specific parameters: camera viewpoints π^j, part scaling $\{s_i\}$, resting part rotation $\{\bar{R}_i\}$, part rotation $\{R_i\}^j$, part MLPs $\{\mathcal{F}_i\}^j$, and part texture $\{T_i\}^j$ (i: part index, j: instance index). In particular, we first assume all instance share the same part shapes $\{\mathcal{F}_i\}$ and jointly optimize all parameters for 1000 epochs with learning rate 0.01. Then, we fine-tune per-instance shapes $\{\mathcal{F}_i\}^j$ and texture $\{T_i\}^j$ for 500 epochs. To ensure training stability with $\mathcal{L}_{dass}$, we set the number of accumulation steps $n = 5$, noise timestep $t = 0.5$, and guidance weight $w_g = 15$. Fig. 2 illustrates the optimization process.

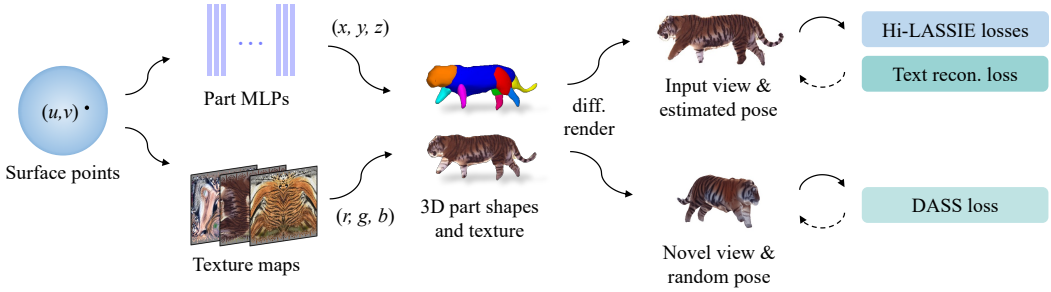

Figure 2: **3D shape and texture optimization via DASS.** We apply the Hi-LASSIE [11] and texture reconstruction losses on the input-view renders, and DASS loss on the novel views.

1.4 Animation fine-tuning

In Fig. 3, we illustrate the animation fine-tuning step with our T-DASS module. First, we obtain K (typically 30) video frames by rendering a sequence of rigid part transformations from our 3D articulated shapes. For each pair of neighboring frames, we also compute the 2D surface flow via mesh rasterization for forward/backward temporal warpping. Then, we optimize the latent codes of each frame with $\mathcal{L}_{recon}$ and $\mathcal{L}_{temp}$ for 300 iterations. The reconstruct targets are obtained from DASS with $n = 5$, $t = 0.2$, and $w_g = 15$. Finally, we can obtain temporally consistent animation by decoding the updated latent codes. Although one can alternatively perform temporal warpping and calculate $\mathcal{L}_{temp}$ in the pixel space, we observe that it leads to blurrier results compared to the high-quality decoded outputs.

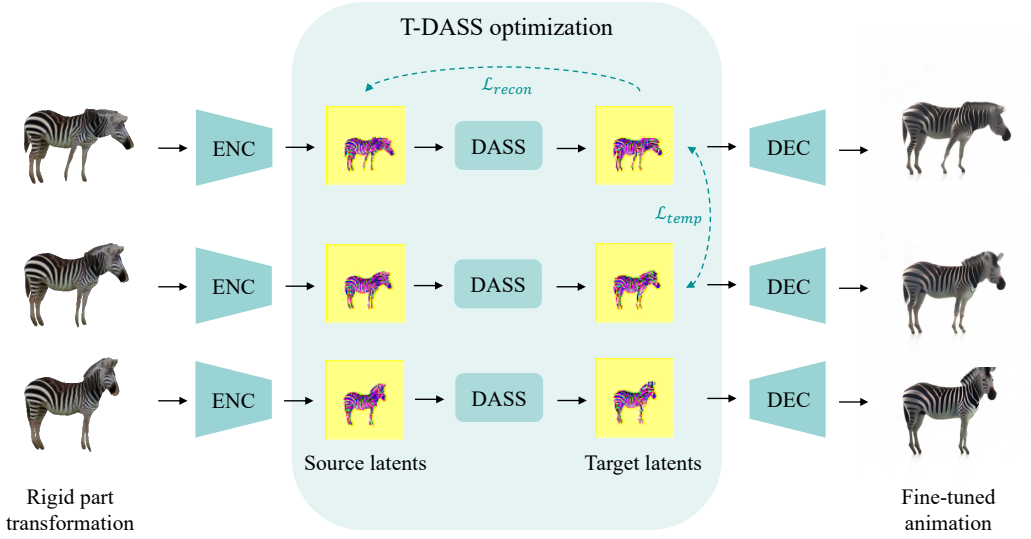

Figure 3: **Animation fine-tuning via T-DASS.**

2 Ablative Analyses

To justify the design of individual components in ARTIC3D, we show the ablative results of keypoint transfer in Table 1. Compared to Hi-LASSIE+ which adopts the common SDS loss, our DASS loss achieves 0.5-1.3% accuracy gain. The proposed input enhancement can further improve the PCK by 0.8-1.9%. The results demonstrate that our DASS module can effectively improve 3D shape quality via both the input enhancement and shape optimization stages.

Table 1: **Keypoint transfer evaluations on the E-LASSIE image sets.** We report the average PCK@0.05 (↑) on all pairs of images.

Method	Input enhance.	$\mathcal{L}_{dass}$	Elephant	Giraffe	Kangaroo	Penguin	Tiger	Zebra
Hi-LASSIE [11]			37.6	54.3	31.9	41.7	57.4	60.1
Hi-LASSIE+			38.3	54.8	32.8	41.8	57.7	61.3
ARTIC3D		✓	38.8	56.1	34.0	42.7	58.5	61.9
ARTIC3D	✓		39.0	57.3	34.6	43.4	58.5	62.4
ARTIC3D (full)	✓	✓	**39.8**	**58.0**	**35.3**	**43.8**	**59.3**	**63.0**

2.1 Input preprocessing

In Fig. 4, we compare the results with and without DASS image enhancement as input preprocessing. The qualitative results show that DASS can effectively produce images with complete animal body shapes and enhanced texture, resulting in more robust 3D outputs.

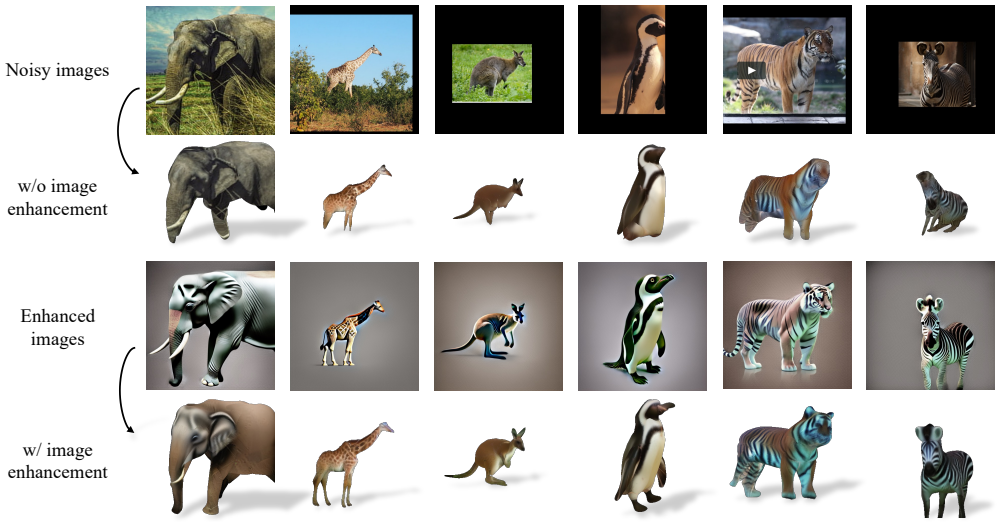

Figure 4: **Ablative results of input preprocessing.** Our image enhancement via DASS can complete the occluded or truncated parts, and thus lead to more accurate 3D reconstruction from noisy images.

2.2 Shape and texture optimization

We show the shape and texture optimization results using different losses in Fig. 5. Hi-LASSIE [11] naively samples surface texture from input images, which leads to unrealistic outputs in novel views. Hi-LASSIE+ adopts the common SDS loss to improve shape and texture details. However, it often produces noisy texture or irregular shapes due to the noisy gradients. ARTIC3D, on the other hand, can effectively reconstruct 3D articulated shapes that look realistic in input and novel views.

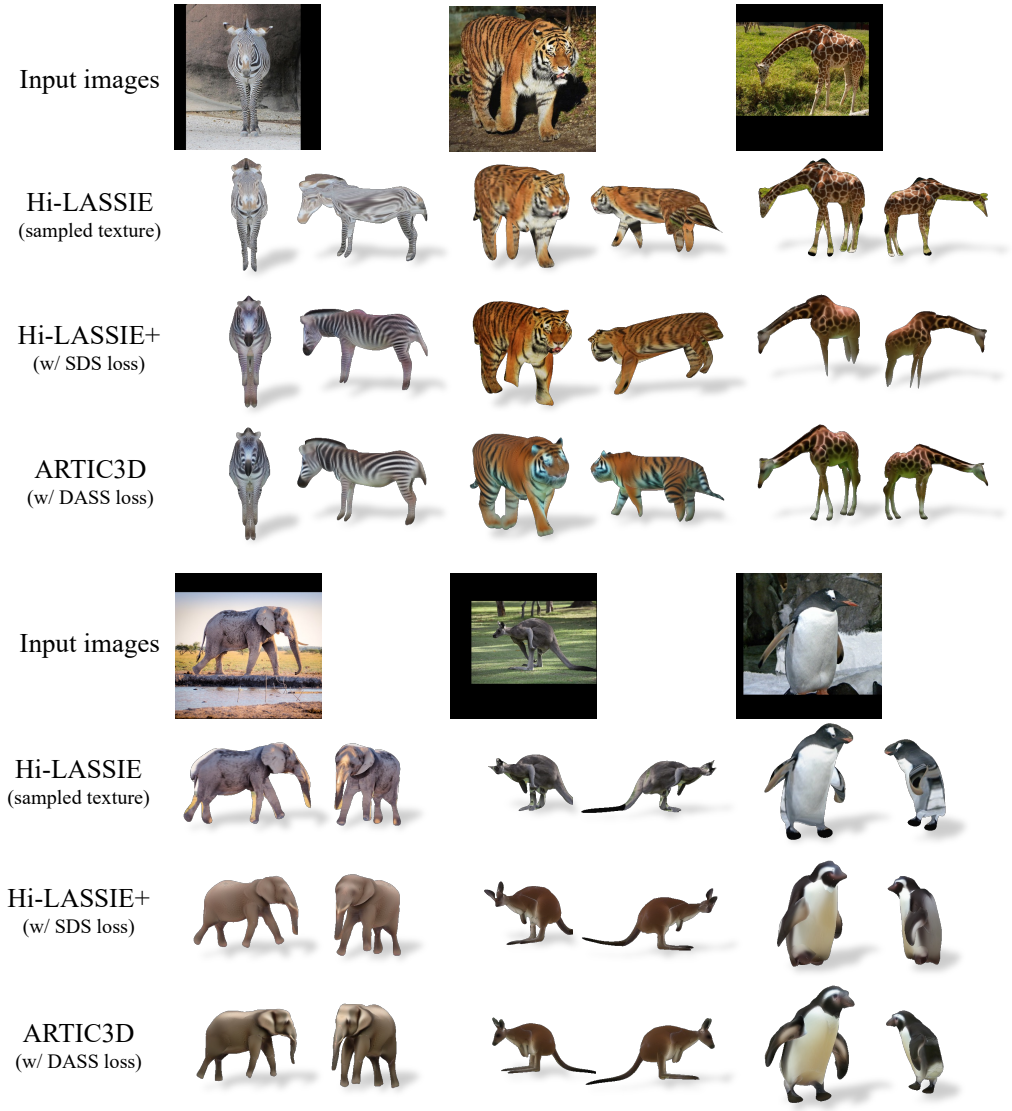

Figure 5: **Ablative results of shape and texture optimization.** We show the sample outputs of Hi-LASSIE [11], Hi-LASSIE+, and ARTIC3D on the E-LASSIE images, which demonstrate that the proposed DASS loss can effectively improve 3D shape and texture details compared to our baselines.

2.3 Animation fine-tuning

In Fig. 6, we compare the animation results via rigid part transformation, per-frame DASS, and the proposed T-DASS. As shown in the sample frames, rigid transformation leads to static texture during motion and sometimes creates disconnected shapes and texture. Applying DASS to each frame individually can improve shape and texture details, which, however, are temporally inconsistent. We propose T-DASS as an alternative to find a better tradeoff between high-fidelity details and temporal smoothness. Please see our supplemental video for better visualization of animations.

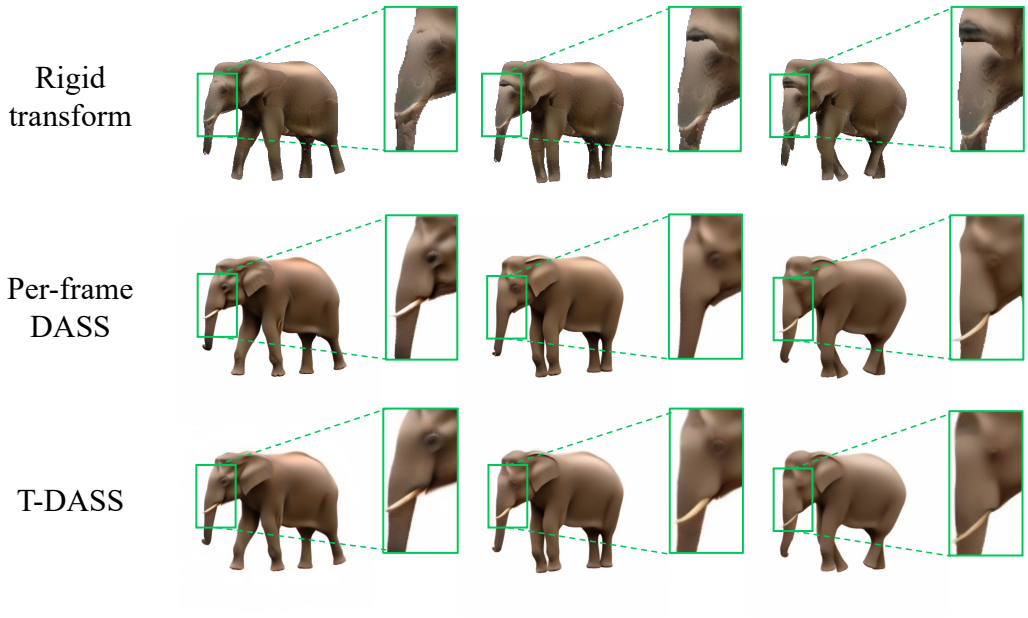

Figure 6: **Ablative results of animation fine-tuning.** The zoomed-in parts show that rigid part transformation creates shape and texture gaps between parts (forehead), per-frame DASS leads to temporal inconsistency (ivory), whereas T-DASS can produce more realistic and smooth animations.

3 Additional Results

3.1 User studies on animation

As mentioned in the manuscript, we conduct user studies to evaluate the quality of our animations. In our user study, we randomly select 3 examples per animal class from the E-LASSIE and Pascal-Part datasets. Specifically, we ask users to perform two pair-wise comparisons: 1) rigid transform v.s. T-DASS and 2) per-frame DASS v.s. T-DASS. As shown in Figs. 7 and 8, each user is presented with two videos from different methods at a time and asked to select the more photorealistic and smooth one while considering video smoothness, flickering, sharpness, and texture consistency. 100 users participated in the study and each were given up to 40 minutes to complete the study. As a result, T-DASS is preferred 55.3% of the time over rigid transform. In the second study, T-DASS is preferred 60.8% of the time over per-frame DASS. Both user studies indicate that T-DASS outputs are more realistic and temporally consistent than the baseline methods.

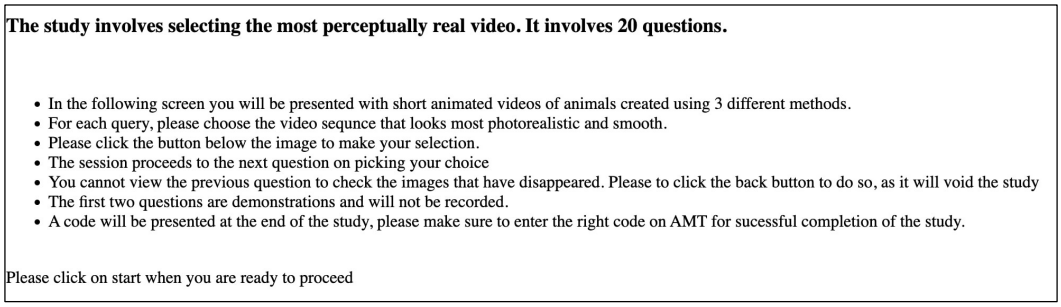

Figure 7: **Text prompt of our user study form.** The users are asked to select the most photorealistic and smooth video from each presented pair.

Figure 8: **Example question in our user study.** We present two videos from different methods at a time and ask users to select the preferred one.

3.2 Qualitative results on Pascal-Part

We show several qualitative results on the Pascal-Part [4] images (horse, cow, sheep) in Fig. 9. Compared to Hi-LASSIE [11], ARTIC3D produces more detailed shapes and realistic texture in both input and novel views.

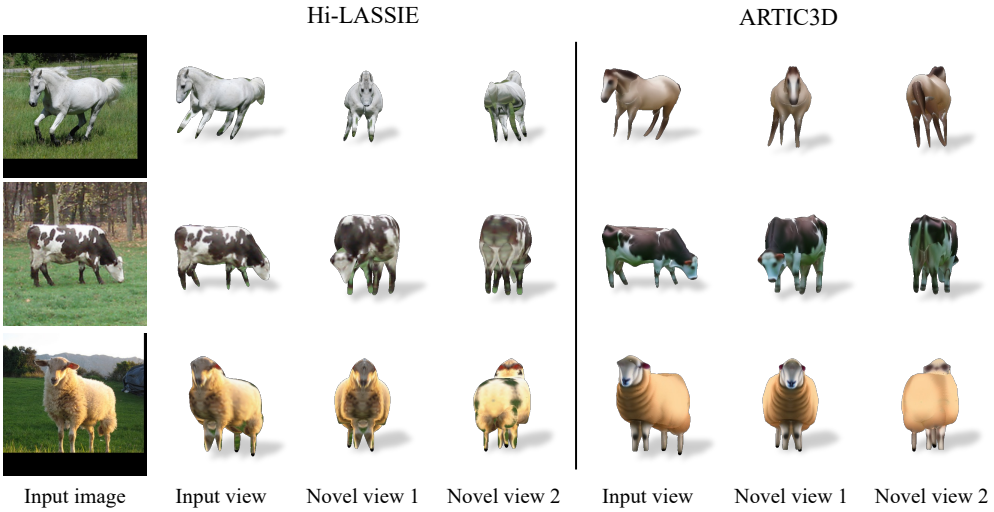

Figure 9: **Qualitative results on Pascal-Part.** ARTIC3D outputs are more detailed and realistic compared to Hi-LASSIE [11], especially in novel views.

3.3 Failure cases

The common failure cases of ARTIC3D include the famous multi-face issue in diffusion-guided 3D reconstruction and inaccurate estimations from heavy occlusions. We visualize two examples of both cases in Fig. 10 and aim to address them in the future.

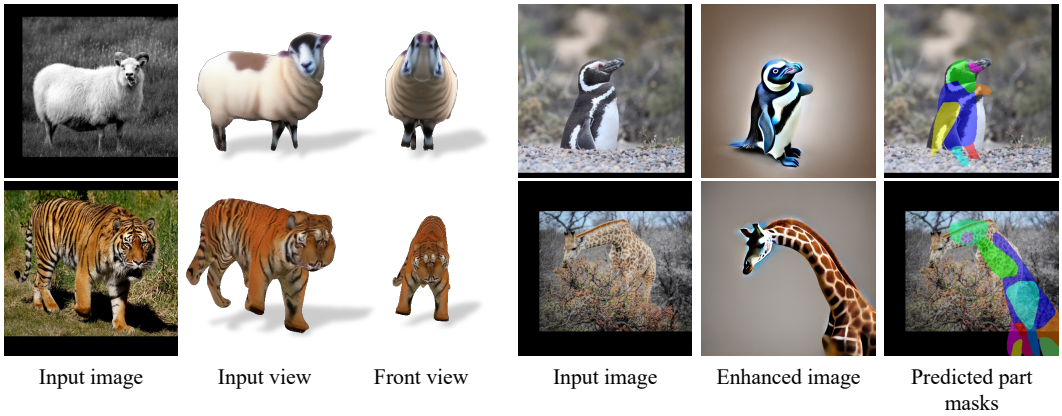

Figure 10: **Sample failure cases.** ARTIC3D sometimes produces multiple faces (left) due to the structural ambiguity during diffusion-guided optimization. In addition, it may lead to inaccurate pose or shape if the animal body is heavily occluded or truncated (right).

4 Datasets and Code

4.1 Dataset details

We conduct experiments on the publically available Pascal-part [3] (http://roozbehm.info/pascal-parts/pascal-parts.html) and LASSIE [12] (https://github.com/google/lassie/blob/main/LICENSE) datasets. Each Pascal-part or LASSIE image collection contains $n = 30$ images ($n = 16$ for Pascal-Part sheep). In addition, we introduce E-LASSIE, an extended LASSIE image set with 15 additional online images (CC-licensed) with occlusions and truncation, totaling 45 images per class. The number of source-target pairs for keypoint transfer evaluation is $n \times (n - 1)$ since we use every image as source or target.

4.2 Code licenses

For ARTIC3D implementation and evaluation, we also make use of the released source code or models of the following methods:

- LASSIE [12]: https://github.com/google/lassie/blob/main/LICENSE (Apache License 2.0)
- NeRS [13]: https://github.com/jasonyzhang/ners/blob/main/LICENSE (BSD 3-Clause License)
- DINO-ViT [2]: https://github.com/facebookresearch/dino/blob/main/LICENSE (Apache License 2.0)
- DINO clustering [1]: https://github.com/ShirAmir/dino-vit-features/blob/main/LICENSE (MIT-License)
- Stable Diffusion [9]: https://github.com/CompVis/stable-diffusion/blob/main/LICENSE (CreativeML Open RAIL-M License)
- Stable DreamFusion [10]: https://github.com/ashawkey/stable-dreamfusion/blob/main/LICENSE (Apache License 2.0)

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021. 1, 9

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 9

[3] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 9

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 8

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[6] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. *NeurIPS*, 24, 2011. 1

[7] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, pages 16252–16262, 2022. 1

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 1

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 9

[10] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 9

[11] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Hi-lassie: High-fidelity articulated shape and skeleton discovery from sparse image ensemble. *arXiv preprint arXiv:2212.11042*, 2022. 1, 2, 3, 5, 8

[12] Chun-Han Yao, Wei-Chih Hung, Yuanzhen Li, Michael Rubinstein, Ming-Hsuan Yang, and Varun Jampani. Lassie: Learning articulated shapes from sparse image ensemble via 3d part discovery. *arXiv preprint arXiv:2207.03434*, 2022. 1, 9

[13] Jason Zhang, Gengshan Yang, Shubham Tulsiani, and Deva Ramanan. NeRS: Neural reflectance surfaces for sparse-view 3D reconstruction in the wild. *NeurIPS*, 34, 2021. 9