4DEquine: Disentangling Motion and Appearance for 4D Equine Reconstruction from Monocular Video

Supplementary Material

1. More Qualitative Results

We present additional qualitative results in Fig. S1. The video sequences were collected from the Internet. For these visualizations, the input to 4DEquine is the first frame of the video; Therefore, the results shown in this figure are the novel-pose animation.

2. More Details about 4DEquine

2.1. Overview of 4DEquine

2.1.1. Training Stage

Due to the scarcity of high-quality 4D equine data, both components of 4DEquine are trained exclusively on synthetic datasets. Motion: AniMoFormer is trained on VarenPoser, our large-scale synthetic video dataset, to regress the VAREN model's pose and shape parameters. Appearance: EquineGS is trained on VarenTex, a synthetic multi-view image dataset, to predict the attributes of canonical 3D Gaussian primitives.

2.1.2. Inference Stage

During inference, 4DEquine processes a real-world monocular video of arbitrary length. The pipeline operates as follows: (1) Motion Recovery: The video is processed by AniMoFormer using a sliding-window strategy to extract temporally smooth VAREN parameters for each frame. (2) Post-Optimization: Because the feed-forward motion predictions may not perfectly align with the image evidence, we apply a brief post-optimization step using 2D keypoints and masks to ensure pixel-aligned equine geometry. (2) Appearance Generation: Concurrently, EquineGS takes a single representative keyframe (e.g., the first frame of the video) and processes it in a feed-forward manner to output a high-fidelity, canonical 3D Gaussian avatar. (3) 4D Reconstruction: Finally, the canonical Gaussian points are deformed into the per-frame pose space using Linear Blend Skinning (LBS) driven by the optimized VAREN parameters, yielding the final 4D reconstruction.

2.2. More Details about VarenPoser Dataset

We collected 500 horse images from the Internet, spanning diverse categories, to serve as appearance guidance for MV-Adapter [2]. To ensure appearance diversity within the VarenPoser dataset, we randomly assigned a generated texture to each of the 1,171 video clips. To maximize this diversity, we enforced a constraint limiting the reuse of any unique texture to a maximum of three clips ($1171/500 = 2.342$). The categorical breakdown and image counts for this 500-image guidance set are detailed in Tab. S1.

Table S1. **The number of different categories of horse images.**

Category	Number	Category	Number
Black Shire	22	Perlino Akhal Teke	29
Black Tobiano Saddlebred	40	Palomino Quarter Horse	40
Skewbald Shetland Pony	15	White Arabian	28
Pinto Paint Horse	50	Chestnut Morgan	18
Black Friesian	23	Buckskin Tennessee Walker	18
Bay Thoroughbred	41	Dapple Gray Andalusian	48
Gray Percheron	5	Grullo Dun Mustang	14
Leopard Appaloosa	58	Silver Dapple Icelandic Horse	16
Gray Lipizzan	35	Total	500

Furthermore, to ensure viewpoint diversity, the initial camera pitch and azimuth angles were randomly sampled within the ranges of $(-15°, 15°)$ and $(-180°, 180°)$, respectively. The camera trajectory settings—fix, orbit, and dolly—were sampled with probabilities of 0.4, 0.3, and 0.3, respectively. Additional visualizations of these camera settings are provided in Fig. S2 and background images were sourced from the COCO dataset [5].

Consistent with the PFERD [4] dataset, VarenPoser maintains a frame rate of 60 frames per second (fps). The video resolution is set to 512×512. Each frame is annotated with VAREN parameters, 21 3D keypoints, 21 2D keypoints (including visibility flags), and a segmentation mask.

2.3. More Details about AniMoFormer

2.3.1. Implementation Details

Spatio-Temporal Transformer. Our spatial transformer employs a Vision Transformer-Huge (ViT-H) [1] backbone, with weights initialized from the pre-trained AniMer [7]. The temporal transformer is constructed using a standard transformer encoder architecture [9]. We train the model using the AdamW [6] optimizer with an initial learning rate of 1×10^{-5}. To enhance robustness to occlusion, we employ a copy-paste data augmentation, randomly overlaying object masks from [8] with a probability of 0.3. Additionally, we simulate varying video frame rates during training by randomly sampling frames at different temporal strides. The model is trained for 100,000 steps on a single NVIDIA RTX 4090 GPU, requiring approximately 10 hours. The final loss components are balanced with the following weights: $\lambda_{\text{varen}} = 0.01$, $\lambda_{\text{smooth}} = 0.001$, $\lambda_{\text{2D}} = 0.05$ and $\lambda_{\text{3D}} = 0.05$.

Post-Optimization. We employ a two-stage post-optimization process to refine the results, with each stage

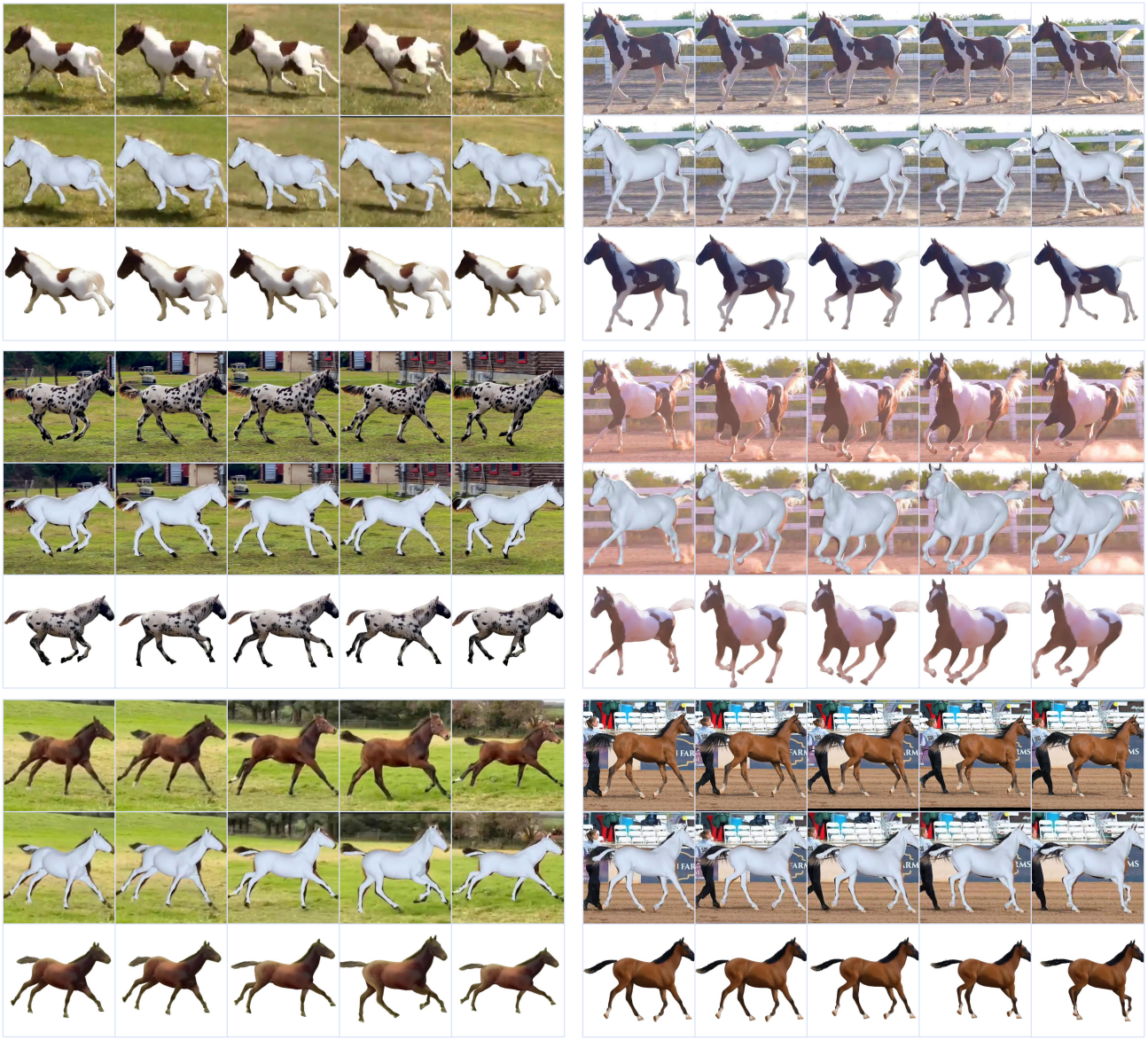

Figure S1. **More Qualitative Results of 4DEquine.** In each example, the first row displays the reference video frames at various time steps, the second row visualizes the output of AniMoFormer, and the third row presents the final reconstruction from EquineGS.

consisting of 100 iterations. Both stages utilize the AdamW optimizer with a learning rate of 5×10^{-3}. The two stages are designed to sequentially refine the fit, first by aligning to keypoints and then by fitting the silhouette. In the first stage, we optimize all VAREN parameters and the camera. The loss weights are set to heavily prioritize keypoint accuracy: $\lambda_{\text{2D}} = 10000$, $\lambda_{\text{smooth}} = 100$, $\lambda_{\text{mask}} = 100$, and $\lambda_{reg} = 1000$. In the second stage, we freeze the pose and camera parameters, allowing the optimization to focus on refining the shape. The loss weights are adjusted to prioritize mask alignment: $\lambda_{\text{2D}} = 100$, $\lambda_{\text{smooth}} = 100$, $\lambda_{\text{mask}} = 10000$, $\lambda_{reg} = 300$.

2.4. More Details about EquineGS

Throughout the training process, the DINOv3 backbone remains frozen. For each training instance, we randomly select a single view as input and utilize four additional views for supervision. The model is optimized using the AdamW optimizer with an initial learning rate of 2×10^{-4}, which includes a linear warmup phase over the first 3,000 steps. We train the model for 100,000 steps using a gradient accumulation of 4 and an image resolution of 448×448. The entire training process is distributed across eight NVIDIA RTX 4090 GPUs and requires approximately 3 days to complete. To balance

Figure S2. **Visualization of camera trajectory settings in the VarenPoser dataset.** We illustrate the three camera behaviors used during data generation: Top: Fixed camera; Middle: Orbit camera; Bottom: Dolly camera (The camera moves farther away from or closer to the object). Background images are sampled from the COCO dataset.

the objective function, the loss hyperparameters are set as follows: $\lambda_{\text{image}} = 1.0$, $\lambda_{\text{mask}} = 0.5$, and $\lambda_{\text{reg}} = 0.1$.

2.5. Visualization for VarenTex

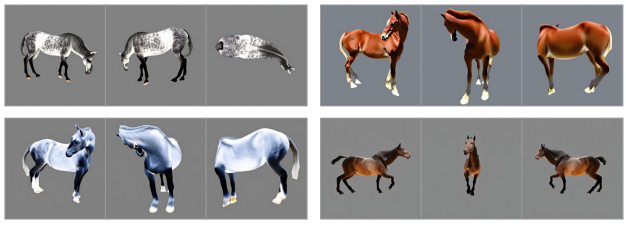

Figure S3. **VarenTex visual samples.**

We present representative visual samples from the VarenTex dataset in Fig. S3.

3. More Experiments

3.1. Comparison with GART

We further evaluate pose estimation performance by comparing our method with GART [3] on the AiM dataset, because GART optimizes underlying mesh pose parameters together with its Gaussian appearance using a photometric loss. As detailed in Tab. S2, AniMoFormer consistently outperforms GART across all metrics. It is important to note that for fair comparison, GART utilizes the estimation results from AniMoFormer as its initialization. We observe that this optimization process inadvertently degrades pose estimation performance because the photometric loss prioritizes pixel-level color alignment (i.e., rendered image quality) over geometric fidelity. This result further proves the validity and importance of our decomposition design.

Table S2. **Quantitative comparisons on the AiM dataset.** GART*: Few-shot GART.

Method	Horse			Zebra		
	PCK@0.05↑	PCK@0.1↑	Accel↓	PCK@0.05↑	PCK@0.1↑	Accel↓
GART*	79.0	97.1	30.7	79.5	95.8	27.8
GART	81.8	97.5	29.1	82.5	95.7	27.3
4DEquine	**87.9**	**98.6**	**22.4**	**89.0**	**96.8**	**19.9**

3.2. Comparison with AniMer + PO

To highlight the effectiveness of our model design, we compare AniMoFormer against AniMer [7] with Post-Optimization. AniMoFormer outperforms "AniMer + PO" (Tab. S3) due to our temporal attention and VAREN's superior expressiveness over SMAL.

Table S3. **Compared to AniMer + PO.**

Variant	APT36K		AiM	
	PCK@0.05↑	Accel↓	PCK@0.05↑	Accel↓
AniMer + PO	61.1	129.5	78.4	24.4
AniMoFormer	**61.8**	**128.6**	**84.2**	**21.8**

3.3. Ablation on the Number of Input Frames

We perform an ablation for window size (N) on the AiM dataset. The results demonstrate consistent performance gains as N increases from 4 to 8 and 16: PCK@0.05 rises from 46.8 to 47.5 and 47.8, while acceleration error decreases from 27.0 to 25.8 and 25.7. However, setting $N = 32$ leads to out-of-memory error. To accommodate videos longer than 16 frames, we employ a sliding-window approach to enable inference over sequences of arbitrary lengths.

3.4. Qualitative Results of Challenging Pose

We provide more challenging poses in Fig. S4 to demonstrate the robustness of 4DEquine.

Figure S4. **Challenging poses. Left: Rearing. Right: Sitting.**

4. Failure Cases and Discussion

We illustrate a failure case in Fig. S5. Although AniMoFormer is capable of reconstructing geometry when input images suffer from severe truncation or occlusion (as it incorporates data augmentation for occlusion and truncation during training), EquineGS struggles to reconstruct a consistent appearance. Therefore, when selecting keyframes as

input for EquineGS, it is necessary to ensure that the horse in the image is not significantly occluded or truncated. In future work, we plan to address these challenges by developing a method for the efficient fusion of multiple keyframes, allowing the model to aggregate appearance information from unoccluded views.

Figure S5. **Analysis of failure cases.** We illustrate a scenario where severe occlusion impacts performance. The first image: The truncated/occluded input image. The second image: The output of AniMoFormer. The third image: The output of EquineGS, showing appearance degradation at unseen areas. The fourth image: Reference frame from a different time step showing the horse's true appearance.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1

[2] Zehuan Huang, Yuanchen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024. 1

[3] Jiahui Lei, Yufu Wang, Georgios Pavlakos, Lingjie Liu, and Kostas Daniilidis. Gart: Gaussian articulated template models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19876–19887, 2024. 3

[4] Ci Li, Ylva Mellbin, Johanna Krogager, Senya Polikovsky, Martin Holmberg, Nima Ghorbani, Michael J Black, Hedvig Kjellström, Silvia Zuffi, and Elin Hernlund. The poses for equine research dataset (pferd). *Scientific Data*, 11(1):497, 2024. 1

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019. 1

[7] Jin Lyu, Tianyi Zhu, Yi Gu, Li Lin, Pujin Cheng, Yebin Liu, Xiaoying Tang, and Liang An. Animer: Animal pose and shape estimation using family aware transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17486–17496, 2025. 1, 3

[8] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1

[9] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017. 1