Ponymation: Learning Articulated 3D Animal Motions from Unlabeled Online Videos

– Supplementary Material –

Keqiang Sun[1*], Dor Litvak[2,3*], Yunzhi Zhang[2], Hongsheng Li[1], Jiajun Wu[2†], and Shangzhe Wu[2†]

[1]CUHK MMLab [2]Stanford University [3]UT Austin
https://keqiangsun.github.io/projects/ponymation

1 Additional Qualitative Results

1.1 Additional Motion Generation Results

Additional generated 3D motion sequences for are shown in Figures 3 and 4. Please refer to the video[1] for more 3D animation visualizations. As shown in the video, by sampling the learned motion latent VAE, we can generate diverse motion patterns, such as `eating` with the head bending towards the ground, `walking` with the legs moving alternately, and `jumping` with the front legs lifted up.

We trained our VAE model with a sequence length of 10 frames. To produce longer motion sequences as demonstrated in the video, we first sample 2 latent codes to generate 2 motion sequences, each comprising 10 frames. We then optimize 1 additional transition motion latents by encouraging the poses of the first frame and the last frame to be consistent with the last frame and the first frame of two consecutive sequences previously generated.

1.2 Qualitative Comparison of Video Reconstruction Results

Figure 1 compares the 3D reconstruction results on video sequences obtained from the MagicPony [6] model and our proposed method. Although MagicPony predicts a plausible 3D shape in most cases, it tends to produce temporally inconsistent poses, including both the rigid pose $\hat{\xi}_{t,1}$ and bone rotations $\hat{\xi}_{t,2:B}$, as highlighted in Figure 1. In contrast, our method leverages the temporal signals in training videos, and produces temporally coherent reconstruction results.

2 Additional Ablation Studies

2.1 Spatio-Temporal Transformer Architecture

We conduct an ablation study to verify the effectiveness of the proposed spatio-temporal transformer architecture. In particular, we remove each individual component from the final model or replace it with a default option, train the model

[*] Equal contribution. [†]Equal advising.

[1] https://youtu.be/poc7c-9hCvQ?si=3k874zHackOre94R

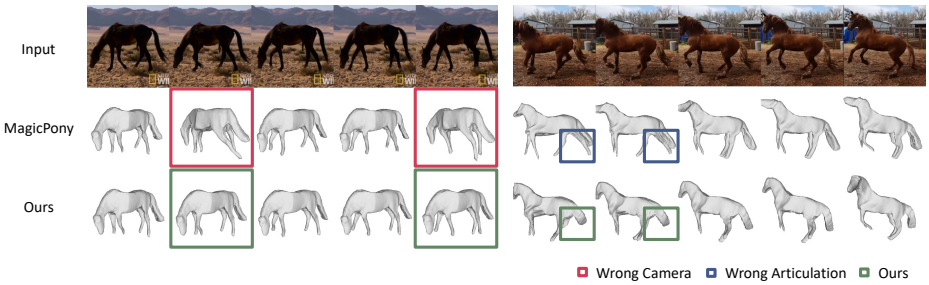

Fig. 1: Comparison of 3D Reconstruction Results with MagicPony [6]. With the video training framework, our method produces temporally coherent and more accurate pose predictions. In comparison, the baseline model of MagicPony often predicts incorrect rigid poses $\hat{\xi}_{t,1}$ (red boxes), and incorrect bone articulation $\hat{\xi}_{t,2:B}$ (blue boxes), resulting in inaccurate 3D reconstruction.

Table 1: Ablation study on the architecture of the motion VAE model.

Row	Method	PCK@0.1	Mask IoU
1	Final (with ST-Transformer)	37.6%	62.0%
2	without spatial Transformers E_s, D_s	33.4%	58.9%
3	without Teacher Loss $\mathcal{L}_\mathrm{teacher}$	32.4%	57.9%
4	without motion VAE	44.3%	66.7%

on the same dataset, and evaluate its performance on 3D reconstruction with the same protocol described in Section 4.3 of the main paper.

First, we remove the spatial transformer encoder and decoder, E_s and D_s, and report the results in row 2 of Table 1. In this variant, specifically, instead of using the spatial transformer encoder E_s to fuse bone-specific local image features before passing them to the temporal transformer encoder E_t, we directly feed the global image features $\{\phi_1, \cdots, \phi_T\}$ into the temporal encoder. Similarly, we also remove the spatial decoder D_s, and directly decode a fixed set of bone rotations from the temporal transformer decoder D_t.

Compared to the final model with spatio-temporal transformer architectures in row 1 of Table 1, the variant without spatial transformer results in less accurate reconstructions, and hence lower scores on the metrics. This confirms the effectiveness of the proposed spatial transformer in extracting motion-specific spatial information from the images.

2.2 Teacher Loss

We also demonstrate the effect of the Teacher Loss $\mathcal{L}_\mathrm{teacher}$ introduced in Section 3.3 of the main paper. We train a variant motion VAE model without this loss, and report its reconstruction performance in Row 3 of Table 1. Without $\mathcal{L}_\mathrm{teacher}$, the model fails to learn accurate poses effectively, leading to degraded reconstruction results. This is mainly because that training the motion VAE

Table 2: Ablation study with different sequence lengths for motion generation evaluated using Motion Chamfer Distance (MCD) on APT-36K [7].

Sequence Length	$K = 10$	$K = 20$	$K = 50$
MCD ↓	**38.03**	38.25	39.25

Table 3: Ablation study on the weight of the KL divergence loss λL_{KL}.

	PCK@0.1	Mask IoU
$\lambda_{\text{KL}} = 0.01$	33.58%	59.85%
$\lambda_{\text{KL}} = 0.001$	**37.63%**	**62.03%**
$\lambda_{\text{KL}} = 0.0001$	35.75%	61.11%

from scratch is computationally inefficient with an expensive rendering step in the loop, and the Teacher Loss can significantly improve training efficiency.

2.3 Sequence Length.

We conducted experiments to understand the effect of different sequence lengths during training ($K = 10, 20, 50$ frames). For a fair comparison, to evaluate the longer motion sequences generated by these variants ($K = 20, 50$), we divide them into consecutive sub-sequences of 10 frames, and average the MCD metric across the subsequences. We use the same metric as introduced in Section 4.2 of the main paper, the Motion Chamfer Distance (MCD) calculated between generated sequences and the annotated sequences in the APT-36K dataset [7]. The results are presented in Table 2.

Upon analyzing the results, we observed that the generated sequences still look plausible as the sequence length increases from 10 to 20. However, a notable degradation in quality is observed as the sequence length increases to 50. This could potentially be attributed to the limited capacity of the motion VAE model as well as the limited size of the training dataset. For our final model, we set the sequence length to 10, which tends to yield the most satisfactory results with a reasonable training efficiency.

2.4 KL Loss Weight.

To train the motion VAE, in addition to the reconstruction losses, we also use the Kullback–Leibler (KL) divergence loss $\mathcal{L}_{\text{KL}}$ in Equation (6) in the main paper. We conducted an ablation study on its weight λ_{KL} to assess its impact on the overall 3D reconstruction accuracy. As shown in Table 3, $\lambda_{\text{KL}} = 0.001$ achieves the best reconstruction results, and is used in all experiments in the main paper.

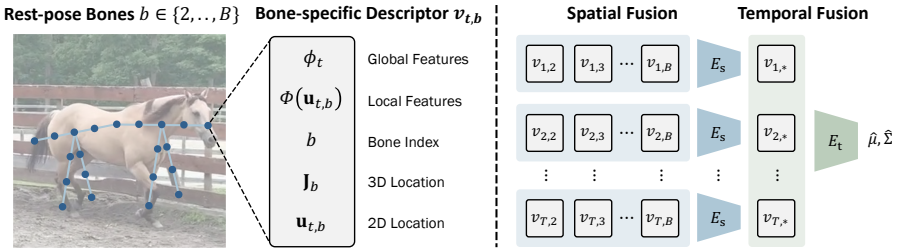

Fig. 2: Illustration of the Spatio-temporal Transformer-based Motion Encoder. For each frame, the bone-specific features $\{\nu_{t,b}\}_{b=2}^{B}$ are first extracted from image features and fused by a spatial encoder E_s to obtain a single feature vector $\nu_{t,*}$. A temporal encoder E_t then further fuses the feature vectors of all frames $\{\nu_{t,*}\}_{t=1}^{T}$ and produces the motion VAE distribution parameters $\hat{\mu}$ and $\hat{\Sigma}$. Please refer to the Section 3.2 in the main paper for detail.

3 Additional Technical Details

3.1 Architecture Details

As explained in the paper, we adopt a spatio-temporal transformer architecture for sequence feature encoding and motion decoding. For better illustrating the architecture, we depict the framework of the spatial and temporal transformer encoders in Figure 2. Also, as presented in Table 4, we use the 4-layer transformer to implement the spatial and temporal transformer encoders E_s, E_t and decoders D_s, D_t. Given the DINO features of the input image, we first concatenate the bone position as Positional Encoding to obtain the bone-specific feature descriptors $\nu_{t,b}$ with shape (BoneNum, FrameNum, FeatureDim) $= (20 \times 10 \times 640)$. Then we map the feature dimension to 256 with a simple Linear layer, and concatenate an additional BoneFeatureQuery token. We use the 4-layer transformer E_s to aggregate all the bone-specific feature descriptors into a per-frame pose feature $\nu_{t,*}$, and subsequently E_t to aggregate all frame-specific features into the VAE distribution parameters, including the mean $\hat{\mu}$ and variance $\hat{\Sigma}$. Using the reparametrization trick, we then sample a latent code z from the Gaussian distribution $z \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma})$, which is first decoded by the temporal decoder D_t and the spatial decoder D_s into a final sequence of bone rotation angles $\hat{\xi}_{*,2:B} \in \mathbb{R}^{20\times10\times3}$.

3.2 Articulation Model Specifications

The configuration of bone topology and skinning weights was established following Magicpony [6]. Here, we give a brief recap of the model.

Posed Shape. The blend skinning model for posing [3, 4, 6] was utilized to articulate the skeleton into a specific pose. This model is parameterised by $B-1$

Table 4: Architecture of the proposed spatio-temporal transformer VAE.

Operation	Output Size
Positional Encoding	$20 \times 10 \times 640$
Linear(640, 256)	$20 \times 10 \times 256$
Concat BoneFeatQuery	$21 \times 10 \times 256$
TransformerLayer $\times$ 4	$1 \times 10 \times 256$
Reshape	$10 \times 1 \times 256$
Concat muQuery and sigmaQuery	$12 \times 1 \times 256$
Positional Encoding	$12 \times 1 \times 256$
TransformerLayer $\times$ 4	$2 \times 1 \times 256$
Reparameterizion	$1 \times 1 \times 256$
TransformerLayer $\times$ 4	$10 \times 1 \times 256$
Reshape	$1 \times 10 \times 256$
TransformerLayer $\times$ 4	$20 \times 10 \times 256$
Linear(256, 3)	$20 \times 10 \times 3$

bone rotations $\xi_b \in SO(3), b = 2, \ldots, B$, and the viewpoint $\xi_1 \in SE(3)$. A set of rest-pose joint locations $\mathbf{J}_b$ was initialized on the instance mesh using straightforward heuristics. Each bone b, excluding the root, has a single parent $\pi(b)$, thereby forming a tree structure.

Each vertex V_i is linked to the bones via the skinning weights w_{ib}, determined based on their relative proximity to each bone. The vertices are then posed using the linear blend *skinning equation*:

$$\begin{aligned} V_i(\xi) &= \left(\sum_{b=1}^{B} w_{ib} G_b(\xi) G_b(\xi^*)^{-1} \right) V_{\text{ins},i}, \\ G_1 = g_1, \quad G_b = G_{\pi(b)} \circ g_b, \quad g_b(\xi) &= \begin{bmatrix} R_{\xi_b} & \mathbf{J}_b \\ 0 & 1 \end{bmatrix}, \end{aligned} \tag{1}$$

where ξ^* denotes the bone rotations at the rest pose.

Bone Topology For all quadrupedal animals examined in this paper, a chain of 8 bones of equal lengths was estimated. These bones lie on two line segments that extend from the centre (root) of the rest-pose mesh to the two most extreme vertices along the z-axis (4 bones on each side), thereby forming a "spine". Then the root joint was slightly elevated, and 4 sets of bones were added to model the legs. The foot joints were first identified as the lowest points of the mesh (in the y-axis) in each of the four xz-quadrants. Subsequently, 4 line segments were drawn from the foot joints to their nearest spine joints, and a chain of 3 bones of equal lengths was defined on each of the segments, representing each leg.

Skinning Weight The skinning weight $w_{i,b}$, which associates each vertex $V_{\text{ins},i}$ with the bones, was defined as follows:

$$\begin{aligned} w_{i,b} &= \frac{e^{-d_{i,b}/\tau_{\text{s}}}}{\sum_{k=1}^{B} e^{-d_{i,k}/\tau_{\text{s}}}}, \\ \text{where} \quad d_{i,b} &= \min_{r\in[0,1]} \|V_{\text{ins},i} - r\tilde{\mathbf{J}}_b - (1-r)\tilde{\mathbf{J}}_{\pi(b)}\|_2^2 \end{aligned} \tag{2}$$

In this context, $d_{i,b}$ is the minimal distance from the vertex $V_{\text{ins},i}$ to each bone b, defined by the rest-pose joint locations $\tilde{\mathbf{J}}_b$ and $\tilde{\mathbf{J}}_{\pi(b)}$ in world coordinates. $\tilde{\mathbf{J}}_{\pi(b)}$ denotes the parent joint of $\tilde{\mathbf{J}}_b$. The temperature parameter τ_{s} is set to 0.5.

3.3 Text Prompts for 4D-fy Evaluation

We provide the 4D-fy [1] model with a list of text prompts, which are enriched by ChatGPT [5] from a list of basic prompts describing horse motions. The complete list is enumerated in the following:

- A horse is running.
- A horse is running.
- A majestic horse galloping swiftly across the verdant meadow.
- An energetic steed dashing with unbridled enthusiasm under the azure sky.
- A spirited horse racing with the wind, its mane flowing like waves.
- A horse is walking.
- A horse is walking.
- A serene horse ambling gently through a misty forest at dawn.
- An elegant steed strolling leisurely along a cobblestone path.
- A calm equine sauntering with grace across a blooming meadow.
- A horse is eating.
- A horse is eating.
- A serene horse gently nibbling on the lush green grass of a tranquil meadow.
- An elegant equine gracefully bending to graze on the dew-kissed clover.
- A peaceful steed leisurely munching on hay in the golden light of dawn.
- A horse is jumping.
- A horse is jumping.
- A majestic horse soaring effortlessly over a rustic wooden fence, its muscles rippling with power.
- An agile steed leaping gracefully, silhouetted against the vibrant hues of the setting sun.
- A spirited equine vaulting energetically over an obstacle, mane flowing like a river in the wind.

4 Limitations and Future Directions

While the model demonstrates promising results, there are several areas where further improvements can be made.

A significant limitation is that the articulated motions are learned on top of a fixed bone topology, which is pre-defined using strong heuristics, such as the number of legs. This approach may not effectively generalize across diverse animal species. A potential avenue for future research could involve the joint discovery of the articulation structure in conjunction with video training.

Additionally, the current model does not distinguish between different legs due to the nature of the DINO features. This can result in a "curious legs" problem, where the model confuses left and right legs of an animal seen from the side. This can be observed in the reconstruction results and subsequently in the generated motion sequences, and is also a common issue even with the most powerful video generation models [2]. Accurately capturing the leg ordering and precise motion is an intriguing challenge for future research in motion generation.

5 Societal Impact

The task of generating 3D motion from unlabeled videos represents a fundamental challenge in the fields of computer vision and computer graphics, in order to extend our current models to the long tail distribution of all kinds of objects in the real world. As an initial exploration in this area, our aim is to stimulate increasing interest and research in this direction. The continued advancement in this field holds great potential of significantly improving the diversity and quality of 3D and 4D models of real-world objects, thereby supporting numerous following applications in virtual reality, robotics and scientific discovery.

References

1. Bahmani, S., Skorokhodov, I., Rong, V., Wetzstein, G., Guibas, L., Wonka, P., Tulyakov, S., Park, J.J., Tagliasacchi, A., Lindell, D.B.: 4D-fy: Text-to-4d generation using hybrid score distillation sampling. In: CVPR (2024) 6
2. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators 7
3. Chadwick, J.E., Haumann, D.R., Parent, R.E.: Layered construction for deformable animated characters. ACM SIGGRAPH Computer Graphics (1989) 4
4. Magnenat-Thalmann, N., Primeau, E., Thalmann, D.: Abstract muscle action procedures for human face animation. The Visual Computer (1988) 4
5. OpenAI: ChatGPT (2023), https://chat.openai.com/ 6
6. Wu, S., Li, R., Jakab, T., Rupprecht, C., Vedaldi, A.: MagicPony: Learning articulated 3d animals in the wild. In: CVPR (2023) 1, 2, 4
7. Yang, Y., Yang, J., Xu, Y., Zhang, J., Lan, L., Tao, D.: APT-36K: A large-scale benchmark for animal pose estimation and tracking. In: NeurIPS Dataset and Benchmark Track (2022) 3

Fig. 3: Additional Motion Generation Results on Horses. Conditioned on an input image, which can be either a real photo or a painting of a horse, our model can generate realistic 4D animations of the instance. See the supplementary video for better visualizations.

Fig. 4: Additional Motion Generation Results for Other Categories. Our model can also be trained on other categories besides horses, and generates realistic motion sequences.