Forecasting Motion in the Wild

Neerja Thakkar[1,2] Shiry Ginosar[3]* Jacob Walker[2] Jitendra Malik[1]* Joao Carreira[2]* Carl Doersch[2]*
[1]UC Berkeley, [2]Google DeepMind, [3]Toyota Technical Institute at Chicago

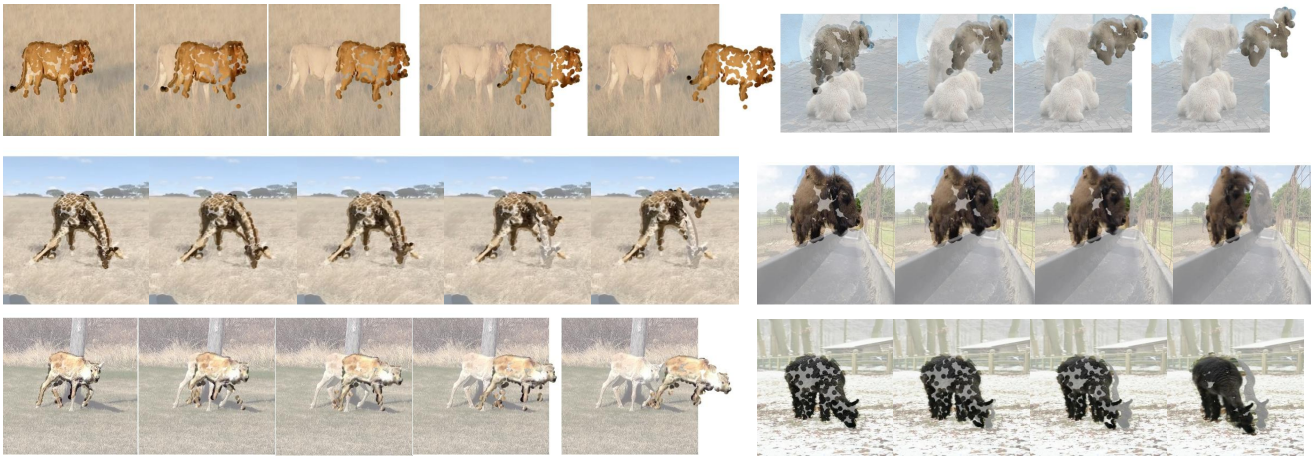

Figure 1. **Dense point trajectories act as visual tokens for behavior, enabling scalable prediction of complex motion across diverse species.** Our method takes as input a single RGB image and a short history of motion, and forecasts future animal motion in the form of point trajectories. For each predicted point trajectory, we translate a small circular patch of the input image along the motion trajectory and superimpose it on the input image (**no pixels are generated!**). Leftmost shows the start locations on the input frame; the rest is forecast by our model. Our method is capable of forecasting many different species and behaviors, even long-tail ones; the polar bear (top right) is only present in 0.31% of the training data, the caribou (bottom left) in 0.025%, and the alpaca (bottom right) in 0.50%. Results video here.

Abstract

Visual intelligence requires anticipating the future behavior of agents, yet vision systems lack a general representation for motion and behavior. We propose dense point trajectories as visual tokens for behavior, a structured mid-level representation that disentangles motion from appearance and generalizes across diverse non-rigid agents, such as animals in-the-wild. Building on this abstraction, we design a diffusion transformer that models unordered sets of trajectories and explicitly reasons about occlusion, enabling coherent forecasts of complex motion patterns. To evaluate at scale, we curate MammalMotion, 300 hours of unconstrained animal video with robust shot detection and camera-motion compensation. Experiments show that forecasting trajectory tokens achieves category-agnostic, data-efficient prediction, outperforms state-of-the-art baselines, and generalizes to rare species and morphologies, providing a foundation for predictive visual intelligence in the wild.

1. Introduction

Predicting the future motion of objects and agents is a fundamental capability of visual intelligence. In dynamic environments, agents, from animals in the wild to humans in social settings, must anticipate the behavior of others in order to act effectively or survive. Despite major advances in visual recognition and generation, predicting behavior remains one of the least understood capabilities of modern vision systems.

A key reason for this gap is the lack of an appropriate representation for behavior. In language, prediction is enabled by discrete tokens that structure the modeling problem. Vision systems lack an analogous token for motion and behavior. In this work, we show that dense point trajectories can serve as such tokens, enabling scalable prediction of behavior across diverse agents. To understand why such a representation is needed, consider the limitations of existing approaches. Forecasting directly in pixel space is universal

but poorly structured: while recent video diffusion models can generate realistic short clips, forecasting behavior directly in pixel space entangles appearance, lighting, and camera motion with object dynamics, making the learning problem unnecessarily complex and data inefficient. At the opposite extreme, parameterized 3D models provide compact and physically valid representations for forecasting, but rely on strong object-specific priors and therefore apply only to a small number of carefully modeled categories, such as humans [43] and a handful of animals [61, 69, 78, 87–90]. Even in these settings they often miss fine-grained deformation and shape variation. Without an intermediate representation that captures motion structure while remaining general, scalable behavior prediction remains difficult. We therefore seek a representation that introduces structure without sacrificing generality.

We therefore propose dense point trajectories as visual tokens for behavior, providing a structured representation for forecasting motion across diverse agents. While sparse points carry little semantic meaning when static, their motion reveals rich information about 3D structure and intent, as demonstrated in Johansson's classical biological motion studies [29] and subsequent work [2, 13, 17, 21, 35]. Representing behavior as evolving 2D point tracks focuses prediction directly on motion dynamics while remaining agnostic to appearance and scene variation. This formulation is significantly more data efficient than forecasting pixels directly [4, 5] and naturally applies to arbitrary non-rigid agents without requiring category-specific models. Point trajectories therefore occupy a principled middle ground between raw pixels and full 3D parameterizations: structured enough to constrain prediction, yet general enough to scale across species, morphologies, and environments.

Building on this abstraction, we introduce a diffusion transformer that forecasts behavior from short motion histories. Unlike prior trajectory-based approaches designed for robotics or rigid scenes [4, 11, 77], our formulation models motion for non-rigid agents in the wild. The model predicts future behavior as an unordered set of point trajectories (Fig. 1), treating each trajectory as a token augmented with local visual context from DINOv3 features. The architecture jointly models trajectories while explicitly reasoning about occlusion and visibility, enabling coherent predictions of complex non-rigid motion. Our model learns diverse motion patterns including gait, cyclical, and linear behaviors, and forecasts future motion across a wide range of species, outperforming state-of-the-art baselines. Training on the broad diversity of motion found in nature further enables generalization to previously unseen categories and morphologies of animate agents.

To study long-tailed biological motion at scale, we focus on unconstrained video of animals in the wild. Animals provide a particularly challenging testbed for behavior prediction: they exhibit highly diverse morphologies and motion patterns, and data for many species is inherently sparse. A representation that succeeds in this regime must generalize across categories without relying on category-specific models. We develop a large-scale pipeline for isolating animal motion from raw video, including robust shot detection and camera-motion compensation, and curate over 300 hours of annotated footage for behavior forecasting which we release with this paper. Using this in-the-wild data, we demonstrate that our approach operates on tracks extracted from unconstrained video and is robust to the noise and partial observability inherent in real-world tracking. This dataset reveals previously unreported statistical structure in animal motion, and provides a foundation for studying predictive visual intelligence in natural environments.

Our contributions are:

1. **Point trajectories as visual tokens for behavior forecasting.** We introduce dense point tracks as a compact mid-level representation for modeling long-tailed natural-world behavior that disentangles motion from appearance and generalizes beyond category-specific 3D models.
2. **A diffusion transformer for trajectory forecasting.** We design a DiT-based architecture that treats trajectories as tokens and predicts diverse futures of non-rigid behavior from short histories while explicitly reasoning about occlusion in unordered track sets.
3. **MammalMotion, a large-scale dataset of animal motion.** We develop a robust pipeline for isolating animal motion in unconstrained video and release 300 hours of annotated footage for behavior forecasting in the wild.

2. Method

2.1. Diffusion-based Point Trajectory Forecasting

We model animal motion as a set of N point tracks $\mathbf{X} \in \mathbb{R}^{T\times N\times 2}$ over T timesteps. Each track $\mathbf{x}_n$ consists of normalized coordinates (x_n^t, y_n^t) and an associated visibility state $\mathbf{O}_n^t \in [0,1]$. We learn the conditional distribution $p(\mathbf{X}_{T_c+1:T}, \mathbf{O}_{T_c+1:T} | \mathbf{I}, \mathbf{X}_{1:T_c}, \mathbf{O}_{1:T_c}, \mathbf{d})$, where $\mathbf{I}$ is the first frame and $\mathbf{d}$ is an optional 2D global displacement.

To improve training dynamics and handle occlusions, we reparameterize the diffusion target as $\mathbf{Z}_0^{\text{diff}} = \{\gamma\mathbf{V}, \beta\mathbf{O}\}$, where $\mathbf{V}$ represents velocities $\dot{x}_n^t = x_n^{t+1} - x_n^t$. For occluded points, we use linear interpolation between the nearest visible frames. Following DDPM [26], our model f_θ minimizes the L_1 denoising objective:

$$\mathcal{L} = \mathbb{E}_{\mathbf{Z}_0^{\text{diff}}, \tau, \boldsymbol{\epsilon}} \left[\| \mathbf{Z}_0^{\text{diff}} - f_\theta(\mathbf{Z}_\tau^{\text{diff}}, \mathbf{Z}^{\text{cond}}, \tau) \|_1 \right] \quad (1)$$

where $\mathbf{Z}^{\text{cond}}$ encapsulates the image $\mathbf{I}$, motion history and initial spatial positions, and optionally $\mathbf{d}$. For efficient inference, we employ DDIM [67] with 100 steps.

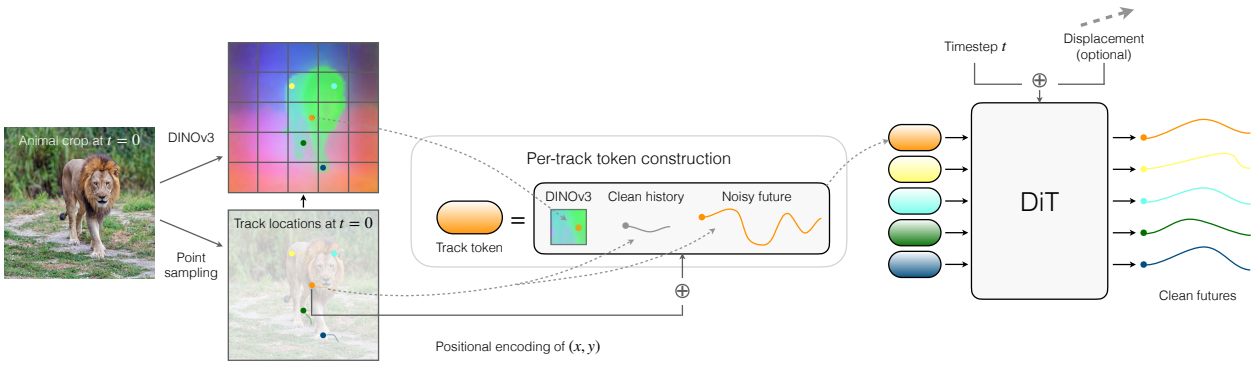

Figure 2. **Architecture.** Given an input frame and (noisy) tracks, we construct a single token for every track, which includes a DINO feature at the start location, the motion history, and the noisy track values, both with occlusion indicators. After projection, we add a position encoding for the initial point location. Tokens are stacked and fed to a transformer (DiT) to predict clean tracks (right).

2.2. Diffusion Transformer Architecture

Our denoiser f_θ is a Transformer (DiT) that treats each track as a token, ensuring permutation invariance. Each token for the n-th track is a concatenation: $\mathbf{Z}_n = [\mathbf{Z}_n^{\text{diff}}, \mathbf{f}_n^{\text{DINO}}, \mathbf{V}_{1:T_c}, \mathbf{O}_{1:T_c}]$.

- **Visual Context:** We extract local features $\mathbf{f}_n^{\text{DINO}}$ from a frozen DINOv3 backbone via bilinear interpolation at the track's initial location (x_n^1, y_n^1).
- **Motion Context:** Velocity histories are embedded using sinusoidal encodings and scaled by γ to match noise variance; occlusion histories are kept as scalar and multiplied by β.

After concatenation, we project the tokens to the transformer's hidden dimension and add a sinusoidal positional encoding of the initial coordinates (x_n^1, y_n^1) to retain explicit spatial relationships. Global conditioning variables—the diffusion timestep τ and the optional displacement $\mathbf{d}$—are integrated directly into each DiT layer via AdaLN [54].

3. Data Processing and Motion Distribution

To isolate animal motion from camera ego-motion in the wild, we developed a data processing pipeline and camera stabilization pipeline that we apply to the MammalNet dataset [10]. We utilize VideoSAM [59] to segment animals and BootsTAPIR [15] for point tracking within segments. We then use a RANSAC-based homography estimation [15] on background points (excluding segments from VideoSAM [59]) and transform points into a camera-stabilized coordinate system normalized to an initial animal bounding box. This results in MammalMotion, $\sim$ 300 hrs of animal motion, which we release.

Log-Normal Distribution of Motion. To validate our motion processing, we compute a histogram of the average displacement (i.e. average distance between start and end for points that remain visible) in Figure 4. While one might

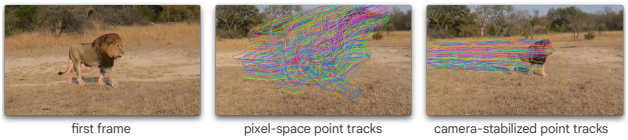

Figure 3. **Our processed data before and after camera stabilization.** Given a first frame (left), the middle image shows the point tracks in pixel space, where the motion of the animals and the camera (panning, zooming out) are entangled. On the right are our point tracks in camera-stabilized space.

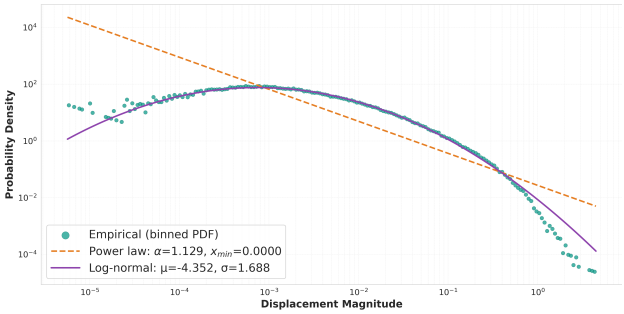

Figure 4. **Animal motion follows a log normal distribution:** We plot a histogram of animal displacement. Horizontal axis is a binned log displacement, while vertical axis is log frequency. We find that log-normal (purple) fits much better than both a power law (orange).

expect animal motion to follow a power law, we instead find that a log-normal distribution fits far better (i.e., the log displacements are normally distributed). Such distributions have been found in other datasets of animal motions, e.g. Lévy flights [8, 23, 28], foraging decisions in rats [30], and general spontaneous behavior in animals [57], and are suggested to imply that motion magnitude is the result of a *multiplicative* interaction of independent factors. We believe this is the first time such a result has arisen from such a diverse dataset of many different species, and without any painstaking manual annotation or use of tracking devices.

(a) **Samples from our model.** Sampling from our model with different random seeds (each row) and no displacement conditioning. The frame on the left is the input state after the motion history. We see different frequencies of the grooming behavior for the jaguar and the dog's head moves different directions.

(b) **Out-of-distribution.** Our model *generalizes* to humans and non-mammals.

(c) **Our Model vs. Stable Diffusion.** Our approach can model the behavior of less common animals in our dataset such as hares, while conventional video models struggle with these animals.

Figure 5. **Qualitative Forecasting Results.**

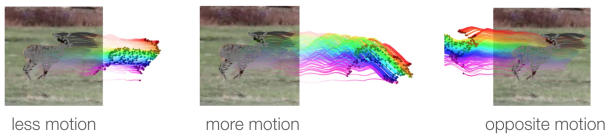

Figure 6. **Prompting our model with different levels of motion.** Grey represents motion history, colors are our predictions.

Table 1. **Quantitative comparison on MammalMotion.**

Method	ADE ↓	FDE ↓	Avg PWT ↑	FD (V) ↓	FVMD ↓
Constant Vel	0.104	0.215	41.15%	2.59	89.77
WHN	0.105	0.200	29.92%	5.34	94.70
Track2Act	0.064	0.126	43.04%	3.20	55.84
Ours	**0.046**	**0.102**	**60.01%**	**1.96**	**17.0**

4. Results

See Sec 10 for our experimental setup: how we use the MammalNet dataset, our metrics, and our baselines.

Samples from our model. Figure 1 shows qualitative results which exhibit convincing forecasts across a variety of behaviors: e.g. the lion's legs follows natural articulation, giraffe raises its neck, the alpaca grazes naturally. This also works for rare animal categories. Figure 5a shows the diversity that our model produces with only different seeds. Fig. 6 demonstrates prompting our model for more motion, less motion, or motion in a different direction, and the model produces plausible behaviors consistent with these motions. Static visualizations do not do justice to motion accuracy; we urge our readers to watch the supplementary video.

Out-of-distribution examples. Fig. 5b displays qualitative results of our model on OOD data. We note that the MammalNet dataset does have videos that contain humans and other types of animals; the ostrich on the bottom right was found in our validation set, so this type of generalization is not surprising. However, the Lego robot (bottom left) and butterfly (top right) are unlike the expected MammalNet data distribution, but still observe physically plausible motion.

Comparison with Video Generation Models. Figure 5c displays a comparison with Stable Video Diffusion [6]. While video models often struggle with physical realism due to the overhead of modeling pixels (textures, lighting), our trajectory-token approach produces more realistic biological behavior with less compute and data. This extends the findings of [7] from rigid synthetic objects to in-the-wild nonrigid motion. E.g., our model is able to forecast the foraging behavior of a hare even though hares only constitute 0.39% of the training data. The video model struggles not only to model this behavior but even to maintain basic anatomy, morphing ears into wings.

Quantitative Results See 6 for detailed quantitative results. Results suggest that our method substantially outperforms other approaches on all metrics, and that when training on our full dataset, there is transfer between species.

References

[1] David J Anderson and Pietro Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014. 3

[2] Anthony P Atkinson, Winand H Dittrich, Andrew J Gemmell, and Andrew W Young. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception*, 33(6):717–746, 2004. 2

[3] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[4] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 4, 8

[5] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In *Conference on Robot Learning*, pages 3936–3951. PMLR, 2025. 2

[6] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 4, 1

[7] Gabrijel Boduljak, Laurynas Karazija, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. What happens next? anticipating future motion by generating point trajectories. *arXiv preprint arXiv:2509.21592*, 2025. 4, 2, 8

[8] Greg A Breed, Paul M Severns, and Andrew M Edwards. Apparent power-law distributions in animal movements can arise from intraspecific interactions. *Journal of the Royal Society Interface*, 12(103), 2015. 3

[9] Hila Chefer, Uriel Singer, Amit Zohar, Yuval Kirstain, Adam Polyak, Yaniv Taigman, Lior Wolf, and Shelly Sheynin. Videojam: Joint appearance-motion representations for enhanced motion generation in video models. In *Forty-second International Conference on Machine Learning*. 2

[10] Jun Chen, Ming Hu, Darren J Coker, Michael L Berumen, Blair Costelloe, Sara Beery, Anna Rohrbach, and Mohamed Elhoseiny. Mammalnet: A large-scale video benchmark for mammal recognition and behavior understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13052–13061, 2023. 3, 5, 7

[11] Yixiang Chen, Peiyan Li, Yan Huang, Jiabing Yang, Kehan Chen, and Liang Wang. Ec-flow: Enabling versatile robotic manipulation from action-unlabeled videos via embodiment-centric flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11958–11968, 2025. 2

[12] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 2

[13] James E Cutting and Lynn T Kozlowski. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the psychonomic society*, 9(5):353–356, 1977. 2

[14] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 2, 8

[15] Carl Doersch, Pauline Luc, Yi Yang, Dilara Gokay, Skanda Koppula, Ankush Gupta, Joseph Heyward, Ignacio Rocco, Ross Goroshin, Joao Carreira, et al. Bootstap: Bootstrapped training for tracking-any-point. In *Proceedings of the Asian Conference on Computer Vision*, pages 3257–3274, 2024. 3, 2, 6, 7

[16] David C Dowson and B. V. Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982. 7

[17] Robert Fox and Cynthia McDaniel. The perception of biological motion by human infants. *Science*, 218(4571):486–487, 1982. 2

[18] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning for general-purpose manipulation tasks. *arXiv preprint arXiv:2412.08261*, 2024. 2

[19] Google DeepMind. Veo 3 technical report. Technical report, Google DeepMind, 2025. 2

[20] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 2

[21] Emily Grossman, Michael Donnelly, R Price, D Pickens, V Morgan, G Neighbor, and Randolph Blake. Brain areas involved in perception of biological motion. *Journal of cognitive neuroscience*, 12(5):711–720, 2000. 2

[22] Xianfan Gu, Chuan Wen, Weirui Ye, Jiaming Song, and Yang Gao. Seer: Language instructed video prediction with latent diffusion models. *arXiv preprint arXiv:2303.14897*, 2023. 2

[23] Richard Gunner, Rory Wilson, Miguel Lurgi, Luca Borger, James Redcliffe, Emily Shepard, Mark Holton, Margaret Crofoot, Abdulaziz Alagaili, Samantha Andrzejaczek, et al. High resolution data reveal fundamental steps and turning points in animal movements. 2024. 3

[24] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018. 3

[25] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *ECCV*, 2024. 2

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 4

[27] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022. 2

[28] Nicolas E Humphries, Nuno Queiroz, Jennifer RM Dyer, Nicolas G Pade, Michael K Musyl, Kurt M Schaefer, Daniel W Fuller, Juerg M Brunnschweiler, Thomas K Doyle, Jonathan DR Houghton, et al. Environmental context explains lévy and brownian movement patterns of marine predators. *Nature*, 465(7301):1066–1069, 2010. 3

[29] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14 (2):201–211, 1973. 2

[30] Kanghoon Jung, Hyeran Jang, Jerald D Kralik, and Jaeseung Jeong. Bursts and heavy tails in temporal and sequential dynamics of foraging decisions. *PLoS computational biology*, 10(8):e1003759, 2014. 3

[31] Bingyi Kang, Yang Yue, Rui Lu, Zhijie Lin, Yang Zhao, Kaixin Wang, Gao Huang, and Jiashi Feng. How far is video generation from world model: A physical law perspective. In *International Conference on Machine Learning*, pages 28991–29017. PMLR, 2025. 2

[32] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*, pages 18–35. Springer, 2024. 2

[33] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6013–6022, 2025. 2

[34] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European conference on computer vision*, pages 201–214. Springer, 2012. 3

[35] Lynn T Kozlowski and James E Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Perception & psychophysics*, 21(6):575–580, 1977. 2

[36] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Steffen Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, et al. Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods*, 19(4):496–504, 2022. 3

[37] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018. 2

[38] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 336–345, 2017. 3

[39] Shijie Li, Chunyu Liu, Xun Xu, Si Yong Yeo, and Xulei Yang. Future-aware interaction network for motion forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7505–7515, 2025. 3

[40] Jiahe Liu, Youran Qu, Qi Yan, Xiaohui Zeng, Lele Wang, and Renjie Liao. Fr\'echet video motion distance: A metric for evaluating motion consistency in videos. *arXiv preprint arXiv:2407.16124*, 2024. 7

[41] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 6

[42] Zhenguang Liu, Pengxiang Su, Shuang Wu, Xuanjing Shen, Haipeng Chen, Yanbin Hao, and Meng Wang. Motion prediction using trajectory cues. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13299–13308, 2021. 3

[43] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 2

[44] Konrad Lorenz. Der kumpan in der umwelt des vogels. der artgenosse als auslösendes moment sozialer verhaltungsweisen. *Journal für Ornithologie. Beiblatt.(Leipzig)*, 1935. 3

[45] Konrad Lorenz and Nikolaas Tinbergen. Taxis und instinkthandlung in der eirollbewegung der graugans. *Zeitschrift für Tierpsychologie*, 1938. 3

[46] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European conference on computer vision*, pages 759–776. Springer, 2020. 3

[47] Alexander Mathis, Pranav Mamidanna, Kevin M Cury, Taiga Abe, Venkatesh N Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21(9):1281–1289, 2018. 3

[48] Seokha Moon, Hyun Woo, Hongbeen Park, Haeji Jung, Reza Mahjourian, Hyung-gun Chi, Hyerin Lim, Sangpil Kim, and Jinkyu Kim. Visiontrap: Vision-augmented trajectory prediction guided by textual descriptions. In *European Conference on Computer Vision*, pages 361–379. Springer, 2024. 3

[49] Tanmay Nath, Alexander Mathis, An Chi Chen, Amir Patel, Matthias Bethge, and Mackenzie Weygandt Mathis. Using deeplabcut for 3d markerless pose estimation across species and behaviors. *Nature protocols*, 14(7):2152–2176, 2019. 3

[50] Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. *arXiv preprint arXiv:2502.13142*, 2025. 2

[51] Ian Noronha, Aneeq Chowdhury, Saru Bharti, and Upinder Kaur. Quadforecaster: Diffusion-based quadruped pose prediction for animal communication analysis. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems workshop: AI for non-human animal communication*. 3

[52] OpenAI. Sora, 2024. 2

[53] Sergiu Oprea, Pablo Martinez-Gonzalez, Alberto Garcia-Garcia, John Alejandro Castro-Vargas, Sergio Orts-Escolano, Jose Garcia-Rodriguez, and Antonis Argyros. A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6): 2806–2826, 2022. 2

[54] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 3, 5

[55] Talmo D Pereira, Nathaniel Tabris, Arie Matsliah, David M Turner, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Edna Normand, David S Deutsch, Z Yan Wang, et al. Sleap: A deep learning system for multi-animal pose tracking. *Nature methods*, 19(4):486–495, 2022. 3

[56] Jernej Polajnar, Elizaveta Kvinikadze, Adam W Harley, and Igor Malenovskỳ. Wing buzzing as a mechanism for generating vibrational signals in psyllids (hemiptera: Psylloidea). *Insect science*, 31(5):1466–1476, 2024. 3

[57] Alex Proekt, Jayanth R Banavar, Amos Maritan, and Donald W Pfaff. Scale invariance in the dynamics of spontaneous behavior. *Proceedings of the National Academy of Sciences*, 109(26):10564–10569, 2012. 3

[58] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014. 2

[59] 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, 6

[60] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020. 3

[61] 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. 2, 3

[62] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European conference on computer vision*, pages 683–700. Springer, 2020. 3

[63] Tim Salzmann, Hao-Tien Lewis Chiang, Markus Ryll, Dorsa Sadigh, Carolina Parada, and Alex Bewley. Robots that can see: Leveraging human pose for trajectory prediction. *IEEE Robotics and Automation Letters*, 8(11):7090–7097, 2023. 3

[64] Leandro A Scholz, Tessa Mancienne, Sarah J Stednitz, Ethan K Scott, and Conrad CY Lee. Plug-and-play automated behavioral tracking of zebrafish larvae with deeplabcut and sleap: pre-trained networks and datasets of annotated poses. *bioRxiv*, 2025. 3

[65] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. Motionlm: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8579–8590, 2023. 3

[66] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 5

[67] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 5

[68] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. 2

[69] Keqiang Sun, Dor Litvak, Yunzhi Zhang, Hongsheng Li, Jiajun Wu, and Shangzhe Wu. Ponymation: Learning articulated 3d animal motions from unlabeled online videos. In *European Conference on Computer Vision*, pages 100–119. Springer, 2024. 2, 3

[70] Neerja Thakkar, Karttikeya Mangalam, Andrea Bajcsy, and Jitendra Malik. Adaptive human trajectory prediction via latent corridors. In *European Conference on Computer Vision*, pages 297–314. Springer, 2024. 3

[71] Niko Tinbergen. On aims and methods of ethology. *Zeitschrift für tierpsychologie*, 20(4):410–433, 1963. 3

[72] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 2

[73] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pages 4601–4607. IEEE, 2018. 3

[74] Karl Von Frisch. *The dancing bees*. A Harvest, 1953. 3

[75] Jacob C Walker, Pedro Vélez, Luisa Polania Cabrera, Guangyao Zhou, Rishabh Kabra, Carl Doersch, Maks Ovsjanikov, João Carreira, and Shiry Ginosar. Generalist forecasting with frozen video models via latent diffusion. 2025. 2, 7

[76] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1160–1169, 2020. 2

[77] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2023. 2, 4, 8

[78] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Magicpony: Learning articulated 3d animals in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8792–8802, 2023. 2, 3

[79] Zhen Xing, Qi Dai, Zejia Weng, Zuxuan Wu, and Yu-Gang Jiang. Aid: Adapting image2video diffusion models for instruction-guided video prediction. *arXiv preprint arXiv:2406.06465*, 2024. 2

[80] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024. 2

[81] Jiange Yang, Haoyi Zhu, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Tra-moe: Learning trajectory prediction model from multiple domains for adaptive policy conditioning. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 6960–6970, 2025. 2

[82] Siyuan Yang, Lu Zhang, Yu Liu, Zhizhuo Jiang, and You He. Video diffusion models with local-global context guidance. *arXiv preprint arXiv:2306.02562*, 2023. 2

[83] Shaokai Ye, Anastasiia Filippova, Jessy Lauer, Steffen Schneider, Maxime Vidal, Tian Qiu, Alexander Mathis, and Mackenzie Weygandt Mathis. Superanimal pretrained pose estimation models for behavioral analysis. *Nature communications*, 15 (1):5165, 2024. 3

[84] Xi Ye and Guillaume-Alexandre Bilodeau. Stdiff: Spatio-temporal diffusion for continuous stochastic video prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6666–6674, 2024. 2

[85] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024. 2

[86] Artem Zholus, Carl Doersch, Yi Yang, Skanda Koppula, Viorica Patraucean, Xu Owen He, Ignacio Rocco, Mehdi SM Sajjadi, Sarath Chandar, and Ross Goroshin. Tapnext: Tracking any point (tap) as next token prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9693–9703, 2025. 2

[87] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 2, 3

[88] Silvia Zuffi, Angjoo Kanazawa, and Michael J Black. Lions and tigers and bears: Capturing non-rigid, 3d, articulated shape from images. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3955–3963, 2018. 3

[89] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images" in the wild". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5359–5368, 2019. 3

[90] Silvia Zuffi, Ylva Mellbin, Ci Li, Markus Hoeschle, Hedvig Kjellström, Senya Polikovsky, Elin Hernlund, and Michael J. Black. Varen: Very accurate and realistic equine network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5374–5383, 2024. 2, 3

Forecasting Motion in the Wild

Supplementary Material

5. Qualitative Results: Supplementary Video

We provide a supplementary video to further demonstrate the qualitative performance of our method. Each example first displays the ground truth video segment used to extract point tracks via our data processing pipeline, followed by our model's predicted motions. The four conditioning timesteps (sampled at 15 FPS) are indicated by a grey border, while all subsequent frames are model predictions. Points predicted as occluded by our method are not rendered.

The video is organized into the following sections:

1) **Diverse Species and Behaviors:** We showcase results across a wide range of behaviors—including walking, mating, eating, fighting, and grooming—and across various species. Notably, our model demonstrates robust performance on rare species that are significantly underrepresented in the training set (e.g., fossa at 0.038%, tapir at 0.22%, and the caribou and eskimo dog at 0.025%). For context, even the most frequent species in our dataset (squirrel, giraffe, elephant, hamster, and deer) each comprise only approximately 3% of the total data.

2) **Stochastic Motion Generation:** By varying the random seed while keeping the input image and motion history fixed, we demonstrate the model's ability to generate diverse, physically plausible motion trajectories from the same initial context.

3) **Controllable Generation via Displacement Vectors**: We illustrate the model's responsiveness to an optional 2D displacement vector. All results before these were generated without this prompting. Each set of results holds the input and random seed constant, but uses a different 2D displacement vector. The displacement vectors used, where $d = [d_x, d_y]$ is the ground truth displacement, are, from left to right, d, $-d$, $\frac{d}{2}$, and $2d$.

4) **Out-of-distribution generalization**: We evaluate our model's zero-shot capabilities by prompting it with non-mammal animals, humans, and other objects.

5) **Baseline Comparisons**: We provide side-by-side visualizations against the "Oracle Velocity" (our strongest non-learned baseline) and Track2Act trained on our full dataset. Comparisons with Track2Act use identical random seeds and motion history. Note that Track2Act and oracle velocity cannot handle occlusions, so all points are treated as visible.

6) **Comparison with Stable Video Diffusion [6]**: While SVD produces high-quality results for common species (e.g., horses), it often struggles with rare species, frequently "shape-shifting" them into more common animals or failing to capture realistic behavioral patterns. We highlight these failure modes in species such as the hare (0.39% in our training dataset), elk (1.2%), bison (0.89%), and black rhino (0.20%). We specifically use the Stable Diffusion XL model available through the interface available at https://stablediffusionweb.com/.

7) **Data Preprocessing and Camera Stabilization**: We visualize results from our data preprocessing pipeline, showcasing both raw outputs and results after camera stabilization. We observe that while many animals are detected, some are missed; furthermore, while the segmentation masks from VideoSAM are highly accurate, they are not perfect on this challenging data. Crucially, the camera stabilization of point tracks allows us to effectively disentangle animal motion from camera motion.

6. Quantitative Results

Results comparing our method with baselines can be seen in table 2 and table 3 for the Panthera genus data. Simple baselines like no-motion and constant-velocity can sometimes perform well on the combined data due to the large amount of low-motion data, but fail for higher motion, and particularly for FVMD which accurately scores motion statistics. Interestingly, WHN gives an accurate acceleration distribution despite not being trained on this data, yet fails to estimate overall velocity and other statistics well (qualitatively it gives low-motion, jittery predictions that don't match animal skeletons). ATM and Track2Act, which we retrained on Panthera data, give predictions that are somewhat closer in terms of the final endpoint error and velocity statistics, but actually perform worse in terms of acceleration and point level accuracy, suggesting they learn overall motion but miss motion details, perhaps in part because the overall losses are on displacement rather than velocity. Our method—trained exclusively on Panthera data—substantially outperforms others in prediction accuracy on every metric. Furthermore, our method can take the true velocity as conditioning to improve results even further, even though for many metrics simply using the oracle velocity provides little boost.

Tables 4 and 5 give analogous results for our model (and Track2Act) trained on all species in our dataset. Results follow similar trends overall, but our model trained on the full data is substantially better, e.g. FVMD for high motion examples falls from 84.8 to 49.3 and PWT rises from 20.6 to 26.0. This isn't because the full dataset is easier than the Panthera subset; other baselines actually perform similarly or worse on these metrics. Instead, this suggests that training on the full dataset improves performance due to transfer between species.

Table 2. Quantitative results on **Panthera Data**, distribution level. FD values are multiplied by 10^3; Variance values are multiplied by 10^5; FVMD values are divided by 10^3. Best results in **bold**, second best underlined. ↑ indicates higher is better; ↓ indicates lower is better.

Selection	Method	FD (V) ↓	FD (A) ↓	Var (V)	Var (A)	FVMD ↓
High motion	GT	-	-	29.5	10.8	-
	No motion	16.6	5.61	0	0	335.406
	Constant vel	7.49	5.61	37.3	0	149.518
	WHN	15.2	**3.27**	1.37	4.11	247.56
	ATM	6.52	6.18	10	6.99	112.71
	Track2Act	6.32	5.06	8.01	0.446	104.85
	Ours (uncond)	**3.71**	4.3	12.8	1.02	**84.79**
	Oracle vel	5.7	5.61	20.9	0	218.73
	Ours (cond)	**2.82**	**4.19**	16.6	1.18	**79.38**
Medium motion	GT	-	-	1.26	0.931	-
	No motion	0.681	0.484	0	0	91.93
	Constant vel	0.822	0.484	0.959	0	54.51
	WHN	0.726	1.16	1.45	4.35	46.47
	ATM	0.494	**0.384**	0.28	0.475	36.94
	Track2Act	0.421	0.416	0.345	0.04	43.62
	Ours (uncond)	**0.405**	0.417	0.179	0.027	**26.85**
	Oracle vel	0.614	0.484	0.0708	0	107.49
	Ours (cond)	**0.389**	**0.414**	0.184	0.0285	**28.96**
Low motion	GT	-	-	0.142	0.173	-
	No motion	0.077	0.09	0	0	46.0
	Constant vel	0.0814	0.09	0.093	0	15.20
	WHN	0.527	1.56	1.44	4.33	19.89
	ATM	0.0746	0.116	0.123	0.254	19.05
	Track2Act	0.0517	**0.0731**	0.0748	0.0294	24.36
	Ours (uncond)	**0.0444**	0.0776	0.026	0.00488	**7.54**
	Oracle vel	0.0679	0.09	0.00643	0	119.12
	Ours (cond)	**0.0431**	**0.0774**	0.0258	0.00499	**9.95**
Combined	GT	-	-	6.93	2.66	-
	No motion	3.77	1.38	0	0	149.53
	Constant vel	1.86	1.38	8.58	0	62.51
	WHN	3.37	1.12	1.43	4.29	86.89
	ATM	1.49	1.4	2.42	1.75	35.50
	Track2Act	1.43	1.21	1.89	0.121	38.44
	Ours (uncond)	**0.874**	**1.05**	2.94	0.226	**24.82**
	Oracle vel	1.43	1.38	4.73	0	118.61
	Ours (cond)	**0.679**	**1.02**	3.73	0.262	**24.90**

Table 3. Quantitative evaluation on **Panthera**, example-level metrics. Best results in **bold**, second best underlined. For non-learned baselines and ATM (single output), we report single-sample metrics; for WHN, Track2Act, and Ours we report best of $K = 5$.

Selection	Method	ADE ↓	FDE ↓	VMD ↓	Avg PWT ↑
High motion	No motion	0.211	0.393	5.51	13.22%
	Constant vel	0.193	0.413	4.91	16.73%
	WHN	0.215	0.393	5.82	10.24%
	ATM	0.143	0.262	5.95	16.31%
	Track2Act	0.135	0.245	5.04	16.72%
	Ours (uncond)	**0.107**	**0.209**	**4.77**	**20.68%**
	Oracle vel	0.082	**0.095**	6.03	17.05%
	Ours (cond)	**0.067**	0.097	**4.61**	**27.31%**
Medium motion	No motion	0.022	0.030	4.18	58.72%
	Constant vel	0.044	0.080	4.78	44.33%
	WHN	0.032	0.040	5.03	36.42%
	ATM	0.025	0.037	4.51	51.24%
	Track2Act	0.024	0.032	3.99	53.69%
	Ours (uncond)	**0.020**	**0.027**	**3.82**	**60.91%**
	Oracle vel	0.022	0.027	4.37	52.53%
	Ours (cond)	**0.016**	**0.019**	**3.75**	**63.66%**
Low motion	No motion	0.007	0.010	2.71	84.71%
	Constant vel	0.013	0.024	3.55	70.99%
	WHN	0.022	0.023	4.45	42.40%
	ATM	0.010	0.016	3.29	72.06%
	Track2Act	0.008	0.012	2.70	76.87%
	Ours (uncond)	**0.006**	**0.009**	**2.57**	**86.10%**
	Oracle vel	0.007	0.009	3.40	82.43%
	Ours (cond)	**0.005**	**0.007**	**2.56**	**87.76%**
Combined	No motion	0.076	0.138	4.02	54.55%
	Constant vel	0.079	0.164	4.33	46.16%
	WHN	0.086	0.146	5.05	30.43%
	ATM	0.057	0.101	4.48	48.39%
	Track2Act	0.053	0.092	3.81	51.13%
	Ours (uncond)	**0.042**	**0.078**	**3.62**	**58.11%**
	Oracle vel	0.035	0.042	4.51	53.14%
	Ours (cond)	**0.028**	**0.039**	**3.55**	**61.67%**

7. Related Work

Pixel Forecasting. When it comes to forecasting visual information, pixels have been the natural choice for several years. Early approaches predicted future pixels deterministically, as a regression problem [53, 58, 68], which is exceedingly challenging, since the problem is ambiguous, and leads to blurry predictions.

While GANs [12, 72, 76] and variational models [37] were once promising, many modern approaches use diffusion models [26] which produce sharp videos [22, 25, 27, 79, 82, 84] – and have brought on a creative video revolution [19, 52]. However, training models directly on video is expensive and data-inefficient and models still struggle with hallucinations and basic physical interactions [3, 9, 31].

Point Track Forecasting. Several works have pushed the frontier in high-quality point-tracking [14, 15, 32, 33, 86], with broad applications across different computer vision tasks. When it comes to forecasting point tracks, the most significant advancements have come from the robotics domain. Any-point Trajectory Modeling [77] introduced the paradigm of first training a regression model to predict point tracks from an image and language instruction, and learning a robot policy on top of the track prediction model. Several approaches have followed in this direction [4, 11, 18, 50, 80, 81, 85]. These works have explored different architectures for forecasting point tracks such as conditional diffusion transformers [4, 11] and latent diffusion models [80], all with the end-goal of learning good robotic manipulation policies. Similarly, [75] applies DiTs to forecast frozen video encodings along with future decoded point tracks. We draw inspiration from these conditional DiT architectures but focus on a different application, forecasting motion in the complex domain of in-the-wild animal data.

Most recently, [7] showed that point-track forecasting outperforms pixel generation for simple Kubric [20] object motions. Our work provides further evidence that point tracks can be a more data-efficient representation for motion, by expanding their scope to more challenging and non-rigid domain of in-the-wild animal data.

Table 4. Quantitative results on **All Data**, distribution level. FD values are multiplied by 10^3; Variance values are multiplied by 10^5; FVMD values are divided by 10^3. Best results in **bold**, second best underlined. ↑ indicates higher is better; ↓ indicates lower is better.

Selection	Method	FD (V) ↓	FD (A) ↓	Var (V)	Var (A)	FVMD ↓
High motion	GT	-	-	31.5	8.94	-
	No motion	27.1	7.51	0	0	481.99
	Constant vel	13.7	7.51	23.8	0	210.47
	WHN	25.2	**3.19**	1.1	3.34	280.77
	Track2Act	14.4	5.54	6.37	1.13	114.30
	Ours (uncond)	**8.96**	3.74	13.1	1.68	**49.30**
	Oracle vel	12.1	7.51	19.8	0	326.80
	Ours (cond)	**4.86**	**3.33**	28.3	2.14	**40.24**
Medium motion	GT	-	-	1.32	1.07	-
	No motion	1.14	0.897	0	0	139.91
	Constant vel	1.43	0.897	1.03	0	89.23
	WHN	0.559	0.679	1.21	3.65	33.86
	Track2Act	0.511	0.454	0.193	0.297	43.63
	Ours (uncond)	**0.257**	**0.298**	0.396	0.251	**12.90**
	Oracle vel	1.03	0.897	0.0825	0	163.67
	Ours (cond)	**0.197**	**0.28**	0.613	0.314	**12.13**
Low motion	GT	-	-	0.111	0.157	-
	No motion	0.0957	0.132	0	0	80.13
	Constant vel	0.124	0.132	0.0891	0	34.31
	WHN	0.46	1.39	1.29	3.89	16.68
	Track2Act	0.0652	0.115	0.128	0.254	40.10
	Ours (uncond)	**0.016**	**0.0309**	0.0382	0.0365	**4.11**
	Oracle vel	0.0886	0.132	0.00404	0	212.13
	Ours (cond)	**0.0148**	**0.0304**	0.0416	0.0383	**4.51**
Combined	GT	-	-	5.41	1.82	-
	No motion	4.66	1.53	0	0	204.14
	Constant vel	2.59	1.53	4.11	0	89.77
	WHN	5.34	**0.691**	1.23	3.7	94.7
	Track2Act	3.2	1.31	1.48	0.454	55.84
	Ours (uncond)	**1.96**	0.877	2.94	0.453	**17.0**
	Oracle vel	2.26	1.53	3.15	0	185.62
	Ours (cond)	**1.07**	**0.778**	6.26	0.57	**14.38**

Table 5. Quantitative evaluation on **All Data**, example-level metrics. Best results in **bold**, second best underlined. For non-learned baselines, we report single-sample metrics; for WHN, and ours we report best of $K = 5$.

Selection	Method	ADE ↓	FDE ↓	VMD ↓	Avg PWT ↑
High motion	No motion	0.325	0.596	6.50	12.44%
	Constant vel	0.286	0.591	5.02	11.94%
	WHN	0.262	0.538	5.74	11.62%
	Track2Act	0.157	0.332	4.62	18.11%
	Ours (uncond)	**0.119**	**0.275**	**4.33**	**26.01%**
	Oracle vel	0.110	0.156	7.04	14.70%
	Ours (cond)	**0.068**	**0.103**	**4.25**	**31.50%**
Medium motion	No motion	0.032	0.057	5.29	48.53%
	Constant vel	0.068	0.142	5.70	33.28%
	WHN	0.035	0.049	4.75	34.49%
	Track2Act	0.027	0.044	3.90	46.44%
	Ours (uncond)	**0.020**	**0.035**	**3.57**	**59.45%**
	Oracle vel	0.030	0.042	5.73	43.37%
	Ours (cond)	**0.016**	**0.021**	**3.51**	**63.05%**
Low motion	No motion	0.007	0.011	3.44	83.75%
	Constant vel	0.018	0.034	4.51	65.13%
	WHN	0.023	0.024	4.19	41.93%
	Track2Act	0.013	0.016	2.88	61.17%
	Ours (uncond)	**0.005**	**0.008**	**2.27**	**88.48%**
	Oracle vel	0.008	0.010	4.54	80.95%
	Ours (cond)	**0.004**	**0.006**	**2.26**	**90.19%**
Combined	No motion	0.099	0.180	4.82	53.94%
	Constant vel	0.104	0.215	5.02	41.15%
	WHN	0.105	0.200	4.85	29.92%
	Track2Act	0.064	0.126	3.73	43.04%
	Ours (uncond)	**0.046**	**0.102**	**3.31**	**60.01%**
	Oracle vel	0.042	0.058	5.57	51.74%
	Ours (cond)	**0.028**	**0.042**	**3.26**	**63.48%**

Behavioral Forecasting in Computer Vision.

Beyond pixels and tracks, there has also been work focusing on forecasting the behavior of intelligent entities as well as their interactions. For example, human trajectory prediction has a long history with a variety of approaches [34, 60]. For direct trajectory prediction, these range from RNN based approaches [62, 73] to VAEs [46] and GANS [24], leveraging generative modeling of future human trajectories. Many recent papers also focus on utilizing scene context [63, 70] for human trajectory forecasting. Behavioral forecasting has also been extensively explored in the context of autonomous driving [38, 39, 48, 65]. Relatively few vision papers have focused on forecasting animal motion. QuadForecaster [51] predicted the poses of animals in constrained contexts while [42] demonstrated a proof of concept of their approach on fish and mice. In contrast, our approach leverages large and diverse datasets and forecasts animal motion on a general level.

Animal Pose, Motion and Behavior. Ethology, the study of animal behavior, has a long history [44, 45, 71, 74]. Recent advances in computing and machine learning show promise in aiding discoveries – e.g. the emerging field of Computational Ethology [1] where computer vision and automated motion analysis plays a major role. For example, work such as DeepLabCut [36, 47, 49] and SLEAP [55] have accelerated annotating poses of animals in video. Video analysis has aided the ethology of a wide range of animals, from jumping plant lice [56], mice [83] and even zebrafish larvae [64]. However, as [1] notes, for most of these approaches, humans still need to manually annotate behaviors in training data – which can be subjective due to varied spatial and temporal scales, and limited by human perception and difficulties in discovering new behaviors. Our work leverages massive datasets of unlabeled videos and is a step towards automatic motion understanding of animals.

There has also been a line of work on reconstructing animal pose in 3D [87, 88], moving towards accurate reconstructions of individual species [61, 78, 89, 90], or creating species-specific models to generate 3D animal motion [69]. These works provide insights into individual animal species, but our work focuses on developing an approach that is data-efficient and can generalize to many species including long tail ones.

8. Method Details: Forecasting Point Trajectories with a Diffusion Model

We present a diffusion-based approach for generating animal motion as a sequence of point tracks. Unlike video generation models that predict RGB pixels, our method operates directly on point trajectories. Given a single observation frame and optional conditioning information like motion history or desired velocity, our model generates plausible future trajectories.

8.1. Problem Formulation

We represent the motion of a single animal as a set of N point tracks, where each track describes the 2D trajectory of a single surface point over a time horizon of T timesteps. Formally, we aim to predict a set of tracks $\mathbf{X} \in \mathbb{R}^{T\times N\times 2}$. Each point track $\mathbf{x}_n = [(x_n^1, y_n^1), (x_n^2, y_n^2), \ldots, (x_n^T, y_n^T)]$, consists of a sequence of normalized coordinates (x_n^t, y_n^t) where t indexes time. Points may become occluded, in which case we assume the location is unknown: we represent the occlusion state as $\mathbf{O} \in \mathbb{R}^{T\times N}$, where $\mathbf{O}_n^t \in [0, 1]$ indicates that it the n'th point is visible (1) or occluded (0) at time t.

Our forecasting model learns a conditional generative distribution:

$$p(\mathbf{X}_{T_c+1:T}, \mathbf{O}_{T_c+1:T} | \mathbf{I}, \mathbf{X}_{1:T_c}, \mathbf{O}_{1:T_c}, \mathbf{d})$$

Where $\mathbf{I}$ is the first frame, $\mathbf{X}_{1:T_c}$ and $\mathbf{O}_{1:T_c}$ are the observed conditioning motion history tracks and occlusion states over the first T_c timesteps, and a single optional 2D displacement vector $\mathbf{d} \in \mathbb{R}^2$ describing the average motion of tracks from the last frame: $\mathbf{d} = \sum_{n=1}^{N} \mathbf{O}_n^T[(x_n^T, y_n^T) - (x_n^1, y_n^1)] / \sum_{n=1}^{N} \mathbf{O}_n^T$. The model generates future trajectories $\mathbf{X}_{T_c+1:T}$ and occlusion states $\mathbf{O}_{T_c+1:T}$ conditioned on this observed history and the optional conditioning. Because the main challenge is to predict the track positions $\mathbf{X}_{T_c+1:T}$, which are a high-dimensional and continuous value, we draw inspiration form prior work [4] and model distribution with a diffusion process.

Parameterization of the diffusion target. Diffusion involves adding Gaussian noise to the inputs (tracks and occlusions) and training a network to denoise them. While we could directly denoise $\mathbf{X}$ and $\mathbf{O}$, there are two problems. First, $\mathbf{X}$ has missing values for occluded points (prior work, e.g. [77], assumes there are no missing points, which is untenable for longer horizons). Second, $\mathbf{X}$ values are extremely correlated, and most of the variance is due to the initial point that is tracked rather than due to the motion itself. We therefore reparameterize the tracks to improve training dynamics. Specifically, we construct the diffusion target $\mathbf{Z}_0^{\text{diff}} = \{\gamma\mathbf{V}, \beta\mathbf{O}\}$ where the n'th row of $\mathbf{V} \in \mathbb{R}^{N\times T\times 2}$ is $[(\dot{x}_n^1, \dot{y}_n^1), (\dot{x}_n^2, \dot{y}_n^2), \ldots, (\dot{x}_n^T, \dot{y}_n^T)]$, and γ and β are scaling parameters so the overall variance roughly matches the noise distribution. Here $\dot{x}_n^t = (x_n^{t+1} - x_n^t)$, and $\dot{y}_n^t = (y_n^{t+1} - y_n^t)$. We interpolate occluded values $\dot{x}_n^t = (x_n^i - x_n^j)/(i - j)$ where i and j are the next and previous visible points (for occluded points at the end of the sequence, which don't have any such j, we simply use 0). We don't do any special preprocessing for the occlusion indicator; even though it's discrete, we find that the model can still denoise to the discrete values provided that they are scaled appropriately.

8.2. Diffusion Process

Following DDPM [26], we define a forward diffusion process that gradually corrupts the diffusion targets $\mathbf{Z}_0^{\text{diff}}$ with Gaussian noise. The forward process over $\tau = 1, 2, \ldots, S$ diffusion steps is:

$$q(\mathbf{Z}_\tau^{\text{diff}} | \mathbf{Z}_0^{\text{diff}}) = \mathcal{N}(\mathbf{Z}_\tau^{\text{diff}}; \sqrt{\bar{\alpha}_\tau}\mathbf{Z}_0^{\text{diff}}, (1 - \bar{\alpha}_\tau)\mathbf{I}), \quad (2)$$

where $\bar{\alpha}_\tau = \prod_{s=1}^{\tau} \alpha_s$ with $\alpha_s = 1 - \beta_s$ and $\{\beta_s\}_{s=1}^{S}$ is a linear noise schedule from $\beta_1 = 0.0001$ to $\beta_S = 0.02$.

Our diffusion model, f_θ, learns to reverse this process by predicting the clean diffusable data $\mathbf{Z}_0^{\text{diff}}$ directly. The training objective minimizes the L1 loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{Z}_0^{\text{diff}}, \tau, \boldsymbol{\epsilon}} \left[\| \mathbf{Z}_0^{\text{diff}} - f_\theta(\mathbf{Z}_\tau^{\text{diff}}, \mathbf{Z}^{\text{cond}}, \tau) \|_1 \right], \quad (3)$$

where $\mathbf{Z}_\tau^{\text{diff}} = \sqrt{\bar{\alpha}_\tau}\mathbf{Z}_0^{\text{diff}} + \sqrt{1 - \bar{\alpha}_\tau}\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, and $\mathbf{Z}^{\text{cond}} = \{\mathbf{I}, \mathbf{X}_1, \mathbf{V}_{1:T_c}, \mathbf{O}_{1:T_c}, \mathbf{d}\}$ is conditioning information including the image $\mathbf{I}$, as well as motion and occlusion history and desired displacement, if available.

Diffusion Transformer Architecture. We now turn to the description of f_θ, which predicts the clean tracks given noisy tracks and conditioning information. We do not assume that tracks are given in any meaningful order or on any grid. However, similar to [4], we note that a transformer model, where each token corresponds to a track, can handle the permutation invariance, as long as we include relevant conditioning information within each token that encodes what the track corresponds to. This design means that the model can easily reason about the full motion forecast for a single point (since everything about a point is encoded within the same point), and yet it can also easily compare and contrast nearby points via attention. It also means that we can make our network is invariant to the input ordering of the tracks.

Figure 2 shows our overall architecture. Each input token corresponds to a full point trajectory; that is, we construct a token for each track before stacking them into a matrix to pass to the transformer. Each token contains all per-track conditioning information: image features, and clean history of conditioning velocities and occlusions $\{\mathbf{I}, \mathbf{X}_1, \mathbf{V}_{1:T_c}, \mathbf{O}_{1:T_c}\}$, as well as the noisy diffusion target for the track. We can then predict the clean data for each track via simple linear projection from the transformer's output.

We construct a token for the n'th point track in the following way. We start with a visual feature derived from I, the image frame at time $t = 1$. We extract the full bounding box around the animal plus a 50% margin, and compute image features from a frozen DINOv3 [66], which should to capture priors about animal parts. We then extract a feature for the track's initial location (x_n^1, y_n^1) using bilinear interpolation. Next, we encode the velocity and occlusion history $(\dot{x}_n^{1:T_{c-1}}, \dot{y}_n^{1:T_{c-1}}, \mathbf{O}_n^{1:T_c})$; we embed the velocities $\dot{x}_n^{1:T_{c-1}}$ and $\dot{y}_n^{1:T_{c-1}}$ using a sinusoidal embedding and scale by γ; we keep the occlusions $\mathbf{O}_n^{1:T_c}$ as scalar and multiply by β. This component of the token is set to zero in the case where the conditioning is not provided. Finally, we add the noisy velocities and occlusion values $\mathbf{Z}_\tau^{\text{diff}} = \{\hat{\mathbf{V}}, \hat{\mathbf{O}}\}$. The full token construction is the concatenation of the clean conditioning DINOv3 features, the clean conditioning velocity history embedding and the occlusion history, the noisy velocities, and the noisy occlusions, along the channel dimension: $\mathbf{Z}_n = [\mathbf{Z}_n^{\text{diff}}, \mathbf{f}_n^{\text{DINO}}, \mathbf{V}_{1:T_c}, \mathbf{O}_{1:T_c}]$.

We project each token to the transformed dimension D_T and add a position encoding. Unlike sequence models, where the added position encoding is derived from the sequence index, we derive our position encoding from the initial location. (x_n^1, y_n^1). We use a simple sinusoidal position encoding with length D_T and add it to the track token embedding. Finally we apply a standard DiT transformer [54], before linearly projecting the final layer to the dimension of each track in Z^{diff}.

The final conditioning information is global, rather than per-track: the diffusion timestep τ and optionally the desired total displacement d. We embed these values via a linear embedding, zeroing out the embedding for d in the cases where it is not given, and use adaptive layer norm [54] as input directly at each layer of the diffusion model, as is typical for encoding the diffusion timestep in a diffusion transformer.

8.3. Sampling with DDIM

For efficient inference, we use the DDIM sampling algorithm [67], which enables deterministic sampling with fewer steps than the training diffusion process. DDIM defines a non-Markovian forward process that preserves the same marginals $q(\mathbf{Z}_\tau|\mathbf{Z}_0)$ but allows skipping diffusion timesteps during sampling.

Given the model's prediction $\hat{\mathbf{Z}}_0 = f_\theta(\mathbf{Z}_\tau, \tau, \mathbf{d})$ at diffusion timestep τ, we compute the next state $\mathbf{Z}_{\tau-\Delta}$ as:

$$\boldsymbol{\epsilon}_\theta = \frac{\mathbf{Z}_\tau - \sqrt{\bar{\alpha}_\tau}\hat{\mathbf{Z}}_0}{\sqrt{1-\bar{\alpha}_\tau}}, \tag{4}$$

$$\mathbf{Z}_{\tau-\Delta} = \sqrt{\bar{\alpha}_{\tau-\Delta}}\hat{\mathbf{Z}}_0 + \sqrt{1-\bar{\alpha}_{\tau-\Delta}-\sigma_\tau^2}\boldsymbol{\epsilon}_\theta + \sigma_\tau\boldsymbol{\epsilon}, \tag{5}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and $\sigma_\tau = \eta\sqrt{(1-\bar{\alpha}_{\tau-\Delta})/(1-\bar{\alpha}_\tau)}\sqrt{1-\bar{\alpha}_\tau/\bar{\alpha}_{\tau-\Delta}}$ controls stochasticity. We use deterministic sampling ($\eta = 0$) with 100 diffusion steps instead of the full 1000 training steps, yielding $10\times$ speedup with minimal quality degradation.

After sampling in velocity space, we convert back to absolute coordinates via cumulative summation: $x_n^t = x_n^1 + \sum_{s=1}^{t-1} v_n^{x,s}$ and $y_n^t = y_n^1 + \sum_{s=1}^{t-1} v_n^{y,s}$ for each point n and trajectory time t.

8.4. Implementation Details

Architecture Our model uses a DiT-B configuration with 12 transformer blocks, hidden dimension of 768, and 12 attention heads. $\sigma_v = 12.0, \sigma_o = 0.1$.

Training. We train with Adam optimizer with learning rate 5×10^{-4}, cosine annealing schedule with 5-epoch warmup, and batch size 64 distributed across 16 GPUs. We apply gradient clipping with norm 5.0. Training runs for 140 epochs.

Exponential Moving Average (EMA). We maintain an exponential moving average of model parameters with decay $\gamma = 0.9997$, a standard technique in diffusion models that stabilizes sample quality:

$$\theta_{\text{EMA}} \leftarrow \gamma\theta_{\text{EMA}} + (1-\gamma)\theta. \tag{6}$$

The EMA weights are used for all evaluation and inference.

Data representation. Each training example consists of $N = 320$ point tracks over a trajectory horizon of $T = 32$ timesteps (sampled at 15 FPS), conditioned on the first $T_c = 4$ timesteps. Tracks longer than T timesteps are subsampled with stride 8. Tracks are normalized to $[0, 1]$ image coordinates within the animal's bounding box and stabilized via homography transformation. We handle variable numbers of valid points ($32 \leq N_{\text{valid}} \leq 320$) using attention masking to ignore padded points.

9. Data Processing details

Here we present an in-depth overview of our data processing pipeline, resulting in the MammalMotion dataset.

9.1. Data Filtering

Our pipeline begins with an initial quality filtering stage applied to the full (untrimmed) 539-hour MammalNet [10] dataset. Videos were excluded if they did not meet our minimum requirements for temporal and spatial resolution: a frame rate of at least 29.9 FPS and a total resolution of $200,000$ pixels.

We also remove all videos with a low dynamic range. The dynamic range of each video is computed by analyzing the pixel intensity distribution across all frames. For each frame, we first convert the image to grayscale and then calculate the dynamic range ratio using percentile-based thresholds to account for potential outliers. The dynamic range ratio R for a frame is defined as: $R = \frac{P_{99}-P_1}{I_{max}-I_{min}}$, where P_{99}

and P_1 are the 99th and 1st percentiles of the pixel intensity distribution respectively, and I_{max} and I_{min} are the theoretical maximum and minimum intensity values possible for the image's data type. The final dynamic range measure for a video is computed as the mean of the frame-wise ratios. This metric provides a normalized measure between 0 and 1, where values closer to 1 indicate a wider effective dynamic range in the video content. We removed videos with a dynamic range value below 0.55.

After filtering according to the above criteria, we were left with 280 hours of video data.

9.2. Shot Detection via Point Tracking

After filtering at the video level, we divided the remaining videos into shots. Seeing that popular open-source libraries such as PySceneDetect fail to detect accurate shot boundaries on the difficult animal data, we developed a novel method for detecting shots based on the same point tracker that we used for obtaining point track training data.

Our algorithm works as follows: We use point-tracking to identify temporal discontinuities in video sequences that indicate shot boundaries. Our algorithm operates by greedily dividing input videos into contiguous segments of up to 100 frames and systematically analyzing the temporal consistency of sparse point correspondences within each segment. For each video segment, the system samples 50 random query points at the first frame. These query points are then tracked forward in time using BootsTAPIR, which outputs point trajectories for the whole segment as well as visibility booleans.

The shot change detection criterion is based on monitoring the percentage of visible points across all frames within each segment—when the visibility percentage drops below 6% (less than 3 points are able to be tracked) for any frame, the algorithm identifies this as a shot change boundary, under the assumption that abrupt scene transitions cause widespread tracking failures due to the disappearance or significant transformation of visual features. When a segment contains multiple frames below the visibility threshold, only the earliest is recorded as the boundary. The segment window then restarts at that boundary frame t', where new query points are sampled, allowing subsequent boundaries to be discovered in successive passes without any post-hoc merging. When no boundary is detected, the window advances by 100 frames, ensuring complete, gap-free coverage.

Using this algorithm for shot detection has the added advantage that shots returned are ones where we will be able to track points.

9.3. Detection and Segmentation

We now get a segmentation of every animal within each shot. Our pipeline begins with an initial animal detection stage for each video shot. We employ Grounding-DINO [41] on every frame, using the text prompt "animal" and a confidence threshold of 0.35. Any shots without a single successful detection are discarded from the dataset.

We next identify frames within each shot that can be used to initialize a video segmenter on every animal in the shot. To ensure tractability, shots longer than 1000 frames are first partitioned into 1000-frame segments. We then developed a multi-stage heuristic to identify a frame where all animals are clearly visible and spatially distinct.

We first estimate the number of animals in the shot, N, by averaging the number of detections across all frames and rounding to the nearest integer. We then form a candidate pool of all frames containing exactly N detections. From this pool, we isolate the top 10% of frames with the lowest average Intersection over Union (IoU) among their bounding boxes. This step prioritizes frames where the animals exhibit minimal overlap. From this refined subset, we select the single frame with the highest mean detection confidence to serve as the definitive query frame.

Finally, we initialize VideoSAM [59] with the bounding boxes from the selected query frame. The resulting segmentation masks are then propagated bi-directionally to cover the entire shot.

9.4. Point Tracking

Once we have shots with animals segmented and tracked, we can track points within each animal. As point trackers are somewhat unreliable over long timeframes, we break each shot into sub-shots of length up to 8 seconds (240 frames). For each animal segmentation mask, we sample 500 points across each sub-shot. To sample each point, we first sample uniformly in time (random frame indices within the shot). Then, we sample from the mask.

Our sampling strategy constrains query points to lie within animal segmentation masks and employs a distance transform-based weighting scheme to allow for sampling of thinner structures such as legs, tails, and heads. Specifically, 75% of points are drawn according to an inverse distance transform distribution. Let $D(\mathbf{p})$ denote the Euclidean distance transform, i.e. the distance from pixel $\mathbf{p} \in M$ to the nearest boundary of segmentation mask M. The sampling probability is: $P(\mathbf{p}) = \frac{1/\left(D(\mathbf{p})+\epsilon\right)}{\sum_{\mathbf{q}\in M} 1/\left(D(\mathbf{q})+\epsilon\right)}$, where $\epsilon = 10^{-6}$ ensures numerical stability. This assigns higher probability to pixels closer to mask boundaries, encouraging coverage of thin structures. The remaining 25% of points are sampled uniformly within the mask to ensure coverage of interior regions.

Once query points are sampled, we track across the shot (up to 8 seconds) using BootsTAPIR [15].

9.5. Camera Stabilization

While the tracked points are faithful to the animal pixels, the motion of the tracked points in pixel space confounds the motion of animals and the camera. Therefore, we employ a stabilization algorithm to disentangle the animal and camera motion, and train models on "stabilized" point tracks that only reflect the motion of animals.

Our approach first samples approximately 300 background points from regions outside dilated animal segmentation masks, applying a 32-pixel dilation buffer to ensure adequate separation from foreground motion. These background query points are evenly distributed across video frames and tracked using BootsTAPIR to establish correspondence across the temporal sequence. The resulting background point trajectories are then used to estimate inter-frame camera transformations through a robust RANSAC-based optimization process using publicly available code [15] that estimates a full homography (8 degrees of freedom). The camera motion estimation employs a reference frame approach where transformations are computed relative to a canonical middle frame, with iterative refinement passes to improve accuracy. To ensure high-quality transformations, we require a $> 50\%$ average inlier ratio, and for the transformation matrix to be well-conditioned. We fail to stabilize 7% of the data and discard this before training.

Once the homographies for each frame in a shot relative to a reference frame are computed, we can stabilize the point tracks at each timestep relative to the start of a time horizon. This enables us to understand how an animal moves, irrespective of camera motion.

9.6. Training Example Construction

We construct training examples by selecting an animal and a particular starting frame t. We extract the input image by taking a bounding box around the segment and expanding it by 50% on each side. We transform all other points with respect to this bounding box using the homographies (i.e. multiply each point on frame t' by $H_t H_{t'}^{-1}$). We then normalize all coordinates with respect to the first bounding box, so that $(0, 0)$ corresponds to the upper-left corner and $(1, 1)$ the bottom right.

10. Experimental Setup

10.1. Experimental Dataset

Before processing the data to create MammalMotion, we filter the full 539-hour MammalNet dataset [10], cutting it down to 280 hours. Videos were excluded if they failed to meet minimum requirements for temporal and spatial resolution or displayed a low dynamic range.

We evaluate our approach on our filtered *all-species* dataset spanning the entire MammalNet taxonomy, as well as a *Panthera*-only subset comprising lions, tigers, and leopards. For each configuration, we construct evaluation sets by randomly sampling from the validation split with different levels of motion. In the all-species setting, random samples are also drawn using stratified sampling across species $\times$ behavior classes to ensure balanced representation of rare categories. In contrast, the Panthera-only setting uses uniform random sampling due to its more homogeneous taxonomy. In both cases, we draw even amounts of samples where the animal averages the following amounts of frame-to-frame absolute motion: less than half a pixel, half to 1.5 pixels, and greater than 1.5 pixels.

10.2. Metrics

We evaluate our model's performance using a suite of metrics that assess both example-level trajectory accuracy and distribution-level motion. All metrics are computed on predicted trajectories compared against our ground truth.

Distribution-Level Motion Statistics: we apply several metrics to the overall distributions of predicted trajectories.

Fréchet Distance (FD). To assess whether our model captures the statistical properties of animal motion, we compute the Fréchet distance [16] between predicted and ground truth trajectory distributions. It fits multivariate Gaussian distributions to a set of vectors and compares them. We compute FD on two representations: first-order differences (velocities), and second-order differences (accelerations), capturing motion dynamics, and motion smoothness, respectively. Following prior work [75], we restrict this analysis to individual tracks visible in all predicted frames to ensure complete motion sequences.

Trajectory Variance. We measure the temporal variance of predicted trajectories $\text{Var}_{\text{pred}} = \text{Var}(\text{flat}(\mathbf{P}^{\text{pred}}))$ where $\mathbf{P}^{\text{pred}} \in \mathbb{R}^{N_{\text{samples}} \times T \times 2}$ is the matrix of all predicted track samples. This captures the diversity and magnitude of motion in generated trajectories. We report this alongside ground truth variance Var_{gt} to assess whether the model reproduces natural motion magnitudes.

Fréchet Video Motion Distance (FVMD). To evaluate temporal coherence, we use the Fréchet Video Motion Distance (FVMD) [40]. FVMD quantifies the discrepancy between the distributions of motion feature vectors, where the features are local histograms of motion orientation and magnitude.

Example-Level Metrics: Since diffusion models are stochastic, we follow common practice and report best-of-K metrics by sampling $K = 5$ predictions with different random seeds for each test example. For metrics where lower is better (ADE, FDE, VMD), we compute $\min_k \text{metric}_k$ for each example, and average across examples. When higher is better (PWT) we use max.

Displacement Error (ADE and FDE). Following standard protocols, we evaluate trajectory accuracy using Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE is the mean squared Euclidean distance between predicted and ground truth trajectories for all visible points across the predicted timesteps. FDE measures endpoint accuracy at the terminal timestep T. **Points Within Threshold (PWT).** As established in point tracking literature [14], we report the fraction of predicted points within pixel-wise distance thresholds of the ground truth $\delta \in \{1, 2, 4, 8, 16\}$, in pixel space, where the input bounding boxes are all resized to (256,256).

Video Motion Distance (VMD). This is a straightforward extension of FVMD to an example-level metric: we compute the same feature vector used for FVMD for both the sample and ground-truth, and report the average Euclidean distance.

10.3. Baselines

We first compare our approach against three non-learned baselines. We then also compare our approach with learned baselines ATM and Track2Act. All baselines and our model use $N_{\text{cond}} = 4$ and predict 28 timesteps at 15 FPS.

No-Motion Baseline. The simplest prediction strategy, repeating the last conditioning position for all future timesteps: $\hat{\mathbf{p}}_t = \mathbf{p}_{N_{\text{cond}}-1}$ for $t \geq N_{\text{cond}}$.

Constant Velocity Baseline. We estimate a per-point-track velocity from conditioning frames as $\mathbf{v} = (\mathbf{p}_{N_{\text{cond}}-1} - \mathbf{p}_0)/(N_{\text{cond}} - 1)$ and linearly extrapolate future positions: $\hat{\mathbf{p}}_t = \mathbf{p}_0 + t \cdot \mathbf{v}$. This provides a simple physics-based predictor assuming constant motion dynamics.

Oracle Velocity Baseline. Uses ground truth average velocity computed from all of the points on the animal, giving a fair lower bound for the setting of our model that takes ground-truth displacement.

What Happens Next (WHN).WHN [7] aims for general-purpose point track forecasting, but the model architecture has a grid constraint that makes it difficult to train on our non-constrained data. Therefore we apply it zero-shot.

Any Trajectory Modeling (ATM). ATM's [77] Track Transformer is a regression-based method. Similarly to our method, it treats each point track over time as a token. It masks out future timesteps and learns to regress these coordinates. ATM does not handle visiblity, regresses on absolute xy-coordinates, and can only predict one plausible future. We train this baseline using our Panthera subset, using $N_{\text{cond}} = 4$.

Track2Act [4]. Most similar to our model, using a diffusion backbone, point track as tokens, and point conditioning setup. We use public Track2Act code and diffuse directly on absolute XY-coordinates, without any positional encoding following the original implementation. We use the[4]'s learned ResNet visual features integrated through AdaLN, and use a standard L2 loss. We omit the goal image (unavailable in our setting), condition the model solely on the initial image, and train the model using $N_{\text{cond}} = 4$.

11. Acknowledgments

We thank Noah Snavely for challenging us with general motion forecasting over a lovely Parisian lunch. We thank Andrew Zisserman, Drew Purves, Aleksander Holynski, Linyi Jin, Sander Dieleman, Mark Hamilton, Jathushan Rajasegeran, and Amanda Seed for helpful discussions and feedback. This work originated [in part] while the authors were visiting the Simons Institute for the Theory of Computing. This work was supported by ONR MURI N00014-21-1-280, and a NSF Graduate Fellowship to NT.