



Animal Pose Tracking: 3D Multimodal Dataset and Token-based Pose Optimization

Mahir Patel¹ · Yiwen Gu¹ · Lucas C. Carstensen¹ · Michael E. Hasselmo² · Margrit Betke^{1,2}

Received: 29 April 2022 / Accepted: 7 November 2022 / Published online: 23 November 2022
© The Author(s) 2022

Abstract

Accurate tracking of the 3D pose of animals from video recordings is critical for many behavioral studies, yet there is a dearth of publicly available datasets that the computer vision community could use for model development. We here introduce the Rodent3D dataset that records animals exploring their environment and/or interacting with each other with multiple cameras and modalities (RGB, depth, thermal infrared). Rodent3D consists of 200 min of multimodal video recordings from up to three thermal and three RGB-D synchronized cameras (approximately 4 million frames). For the task of optimizing estimates of pose sequences provided by existing pose estimation methods, we provide a baseline model called *OptiPose*. While deep-learned attention mechanisms have been used for pose estimation in the past, with *OptiPose*, we propose a different way by representing 3D poses as tokens for which deep-learned context models pay attention to both spatial and temporal keypoint patterns. Our experiments show how *OptiPose* is highly robust to noise and occlusion and can be used to optimize pose sequences provided by state-of-the-art models for animal pose estimation.

Keywords Animal video dataset · Pose estimation · Tracking · Optimization · Thermal infrared · Multimodal

1 Introduction

The precise quantification of animal behaviors based on their position, orientation, and movement of their head, body, and limbs is critical for neuroscience, ecology, and psychology studies. Computer vision has long been a facilitator of such studies, initially limited to two-dimensional analysis of single videos of rodents and flies in laboratory settings (Breslav et al., 2016; Graving et al., 2019; Liu et al., 2020; Pereira et al., 2019), then providing three-dimensional (3D) behavior analysis of wild animals in the field, for example, 3D flight paths of bats and birds (Theriault et al., 2014; Wu & Betke, 2016; Wu et al., 2011), and eventually using deep learning models to estimate 3D pose of animals, small and

large, caged and wild, e.g., (Gosztolai et al., 2021; Joska et al., 2021; Mathis et al., 2018). While DeepLabCut (Mathis et al., 2018) is likely the most widely used state-of-the-art software for estimating and tracking 3D keypoints on animals in laboratory settings, other approaches are emerging to improve accuracy of keypoint detection and tracking (Gosztolai et al., 2021; Lauer et al., 2021). Research progress, however, has been hindered by the fact that there are so few publicly available video datasets (Li & Lee, 2021; Li et al., 2020). Unlike the field of human pose estimation, where benchmarking on large, widely-acknowledged datasets is possible (Ionescu et al., 2014; Mehta et al., 2017), animal 3D pose estimation is still in its burgeoning phase. Animal video datasets, which are usually collected and curated by laboratories with an interest in behavioral or neurological motion studies of certain species are typically not shared, are very limited in scale, or lack video annotations. Moreover, the 3D groundtruth of keypoint locations of animals are difficult to obtain, because they either require laborious manual annotation and verification, or elaborate recording settings and/or infrared-reflecting markers on the animals, e.g., (Dunn et al., 2021).

In our work, we developed a multi-camera system to record rodents in a laboratory setting, using three modalities – RGB, depth, and thermal infrared, see Fig. 1, (the

Communicated by Angjoo Kanazawa.

M. Patel and Y. Gu have contributed equally to this study

✉ Mahir Patel
mahirp@bu.edu

¹ Department of Computer Science, Boston University, Boston, USA

² Center for Systems Neuroscience, Boston University, Boston, USA

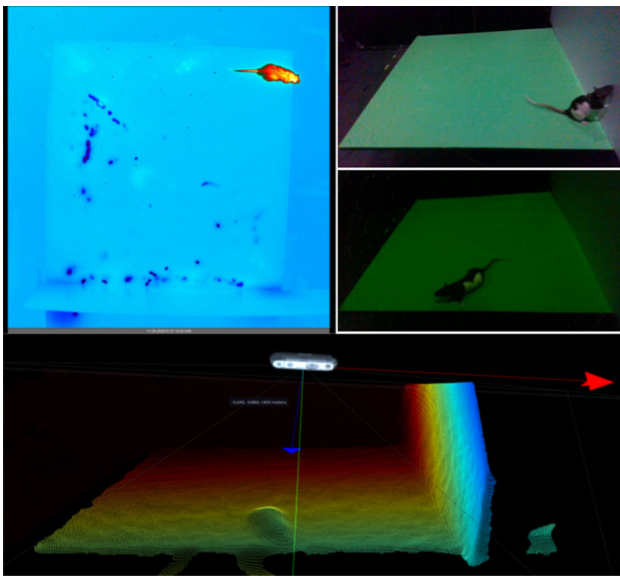


Fig. 1 The proposed publicly available Rodent3D dataset contains annotated multimodal videos, including thermal infrared (top left), regular and low-light RGB (top right), and depth (bottom)

modeling experiments described in this paper do not use the collected depth data). Thermal infrared cameras measure the infrared energy (heat) emitted by rodents in a passive way. These thermal cameras use a different technology than active night-vision cameras that illuminate their target with infrared lighting, which we do not employ. Thermal infrared cameras may be preferred over color cameras at daytime light or infrared-illuminating cameras for recording rodent behavior in neuroscience laboratories to avoid potential unnatural behavior due to light distraction by these nocturnal creatures.

We have collected and curated an animal pose estimation dataset, called Rodent3D, that we make publicly available, and provide 3D pose estimates that can serve as groundtruth labels so that benchmark experiments with single or multiple modalities (thermal, RGB, depth) can be conducted by others.

We publish our dataset together with a strong baseline model, called OptiPose. We designed OptiPose as a spatio-temporal self-attention model that outputs a sequence of 3D animal pose estimates defined by 3D keypoints located on the surface of the animal (see Fig. 2). Its input is a potentially highly noisy version of a 3D keypoint sequence. OptiPose can handle significant occlusion of keypoints (Fig. 2, left), inaccuracies introduced either by 2D keypoint extractor or human annotators (Fig. 2, middle) or 3D reconstructions (Fig. 2, right). OptiPose can handle significant occlusion of keypoints (Fig. 2, left), inaccuracies introduced either by 2D keypoint extractor or human annotators (Fig. 2, middle), and 3D reconstruction inaccuracies (Fig. 2, right). It can be employed in combination with any 3D keypoint tracker, e.g., (Mathis et

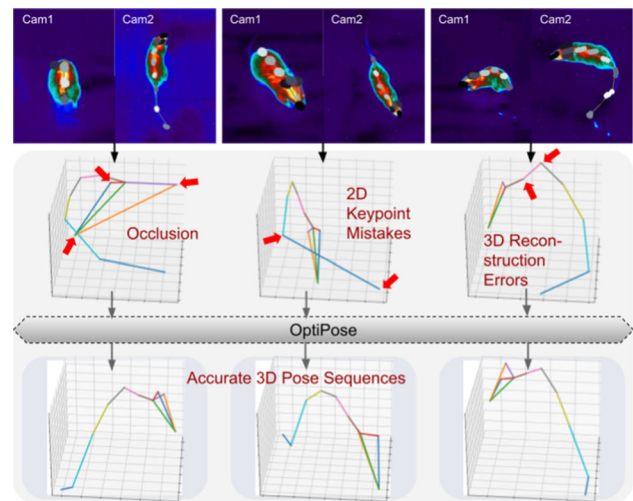


Fig. 2 OptiPose Task: Improving the tracks of 3D keypoint estimates. Red arrows indicate inaccurate keypoint estimates due to occlusion, inaccurate detection by the 2D keypoint extractor or human annotators, or errors introduced in the 3D reconstruction. Raw keypoint estimates (middle plots), computed by triangulating 2D points in the two thermal views (top row), are refined by OptiPose (bottom plots)

al., 2018), as a second stage in a two-stage 3D pose tracking system.

OptiPose is inspired by deep models used in natural language processing and the self-attention mechanism (Vaswani et al., 2017). Our model treats each pose as a token and the keypoints collectively form its embedding in a high-dimensional space. Specifically, given T consecutive sets of N keypoints on an animal of interest, its poses over a period of time, $P_{1..T}$, can be defined by an embedding vector in $\mathbb{R}^{T \times 3N}$ space. Thus, an activity, defined as a sequence $P_{1..T}$ of discrete, keypoint-defined poses can be considered on par with the embedding of a sentence in a natural language. Different permutations of pose tokens form different activities. These permutations are defined by the movement pattern of the animal, which could be interpreted as the grammar of the language. In this manner, it can be trained to learn highly representative pose embeddings and is able to predict 3D keypoints accurately, even if multiple keypoints are occluded in some or all camera views for some periods of time. These missing keypoints in the input are marked by a predefined out-of-range vector, and the model replaces them based on the deep-learned postural dynamics. Therefore, OptiPose can be considered a denoising auto-encoder.

Our contributions are summarized as follows:

- We provide a new dataset, called Rodent3D, of one or two rodents feeding and exploring their laboratory environment. Our unique dataset consists of high-speed multi-view infrared and D-RGB videos, camera calibration data, obtained with a unique thermoelectric

calibration cube that we constructed, synchronization data, hand-labeled 2D keypoints, and their 3D reconstructions. We describe experiments with baseline models on Rodent3D.

- We propose a deep-learned model of token representations of 3D pose, called OptiPose, that uses a self-attention mechanism to interpret spatial and temporal 3D keypoint patterns for the task of 3D pose sequence optimization.
- We provide tools for animal behavior analysis.

We make the dataset and code publicly available. With this new multi-modal dataset, other researchers interested in animal pose estimation, tracking, and/or multi-modal computer vision, have a much needed additional dataset to work with. Furthermore, we hope that our innovative way of using self-attention in deep learning for designing token-based 3D pose tracking will be useful for other researchers within and beyond pose estimation research.

2 Related Work

We here discuss prior work on 3D animal pose estimation and describe the relevant publicly-available datasets. We focus on approaches that exploit multi-view information for animal 3D reconstruction. For a detailed review of the work on markerless 2D animal pose estimation, see Mathis et al. (2020).

Rat 7M, Marshall et al. (2021a). The dataset includes 10.8 h of motion capture and color video data, almost 7 million frames total. The motion capture data was obtained by attaching markers to 20 sites on a rat's head, trunk, and limbs using body piercings. The RGB videos involve up to 6 synchronized camera views, operating at 30 Hz under day-time light conditions. Published by the same research group, **PAIR-R24M** (Marshall et al., 2021b) records activities from paired rats. It includes 26 h of motion capture and color video data, yielding a total of 24.3 million RGB video frames. Rat7M and PAIR-R24M are the only datasets we are aware of that provide marker-based motion capture results as 3D groundtruth for rat keypoints.

AcinoSet Cheetah Dataset, Joska et al. (2021). This dataset includes sequences of a running cheetah from six views recorded with high-speed cameras. The dataset comes with 2D estimates and calibration data for each camera view. It also includes 3D reconstructed poses for all frames, which can be used as groundtruth and which were computed by the Full Trajectory Estimation (FTE) method applied to the six-camera data (Joska et al., 2021). Each pose consists of 20 keypoints.

DeepFly3D Drosophila Dataset, Günel et al. (2019). The Drosophila dataset (Ramdya, 2019) includes sequences of

images of a tethered fly from seven views focusing on the limbs and appendages. Annotations of the dataset are provided by running the DeepFly3D model (Günel et al., 2019), which computes 3D pose consisting of 38 keypoints per frame.

Animal Pose Estimation from 2D to 3D. Several approaches exist that estimate keypoints in 3D by either computing them from extracted 2D keypoints (Hu et al., 2021; Joska et al., 2021; Kearney et al., 2020; Martinez et al., 2017; Nath et al., 2018; Zhang et al., 2021; Tome et al., 2017) or inferring them directly from 2D images or videos using volumetric convolutional networks (Dunn et al., 2021; Isakov et al., 2019; Mehta et al., 2017; Pavlakos et al., 2017). Our method falls into the first category. Most related to our work are approaches such as GIMBAL (Zhang et al., 2021), FTE (Joska et al., 2021), and Anipose (Karashchuk et al., 2021) that focus on improving 3D estimates of poses reconstructed with geometric triangulation. They use DeepLabCut (Mathis et al., 2018) as the 2D keypoint detector as we do. Anipose and our OptiPose are designed to track any animal species. FTE incorporates domain expertise in cheetah anatomy, parameterizing the roll, pitch and yaw angles of related body parts.

Physics-based Pose Estimation. The most recent trend in animal pose estimation from video has been to include information on the postural dynamics of the animal either in the model or as a post process (Joska et al., 2021; Monsees et al., 2021; Zhang et al., 2021). Including kinematics is a highly-effective and much-easier-to-accomplish approach for the task of human pose estimation, because state-of-the-art models can rely on physics-based human-motion simulators (Yuan et al., 2021) or bone models (Gong et al., 2021). In the absence of such simulators or models for a given animal species of interest, domain expertise in biomechanics and locomotion of the species is required. However, creating physics-based models is expensive in terms of human capital and typically beyond the expertise of computer vision, psychology, or neuroscience collaborators. The question that we are answering with our work is “Can we develop a generalizable architecture that can model pose-dynamics without biomechanical expertise?”

Single image versus video analysis. Studies on 3D animal pose estimation can be distinguished by the modality of data analyzed – single images or videos. Even if videos are available, some studies purposely focus on pose estimation in single images, using recursive filtering for the tracking of poses (Graving et al., 2019). We instead, with our spatio-temporal OptiPose model, show the advantage of training a deep model to interpret the input of a *sequence* of poses. However, this means we cannot test our model on any of the animal datasets with tens of thousands of single-frames, e.g., (Biggs et al., 2020; Mu et al., 2020; Wah et al., 2011), for which models have been proposed. We are therefore limited to Opti-

Pose benchmarking experiments involving Rat7M (Dunn et al., 2021), AcinoSet (Joska et al., 2021), DeepFly3D (Günel et al., 2019) and our own Rodent3D dataset.

Augmenting training data. 3D toy (Zuffi et al., 2017) and CAD (Joska et al., 2021; Li et al., 2020; Li & Lee, 2021) models of animals have been used to support the task of pose estimation. Bridging the domain gap between synthetic and real data is extremely challenging, particularly when the CAD model is not only static but is supposed to include biomechanically realistic movements of multiple limbs (Li et al., 2020). Verification of such dynamic models requires domain expertise. Our approach is to circumvent the time-consuming task of creating dynamic CAD models and instead focus on augmenting real data with our proposed masked keypoint data augmentation algorithm.

Pose Estimation From Humans to Animals. Humans are part of the animal kingdom, and there are no intrinsic reasons why work on human pose tracking could not be extended to animal pose estimation. Research on deep learning models for animal pose estimation has indeed borrowed ideas from models for human pose estimation. Attention-based approaches for temporal models have been developed concurrently. Recent advancements in 3D human pose estimation models adopt transformer architectures (Rempe et al., 2021; Li et al., 2022; Lin et al., 2021; Shuai et al., 2022; Zheng et al., 2021). Among them, PoseFormer (Zheng et al., 2021) is the first purely transformer-based network. It uses a spatial transformer to model the spatial relationships between 2D joints and a temporal transformer for temporal information in videos. It achieved state-of-the-art performance on both the Human3.6M (Ionescu et al., 2014) and MPI-INF-3DHP (Mehta et al., 2017) datasets. Such publicly-available benchmark datasets, unfortunately, are still lacking for animals, but are needed so that progress made for human pose estimation can extend to animal pose estimation.

3 The Rodent3D Dataset

The Rodent3D dataset will be made publicly available with the acceptance of this paper at <http://www.cs.bu.edu/faculty/betke/Rodent3D>. It will include 240 min of multimodal video recordings of one or two rodents exploring an arena in a laboratory. Some videos show a rodent searching for and eating food pellets, dropped into the arena by two pellet dispensers. The dataset will be published with raw video data, approximately a total of 4.5M frames, annotated data, calibration parameters, and data curation code.

3.1 Video Collection and Synchronization

The collection of Rodent3D has undergone three stages, each stage involved an increasing number of cameras and

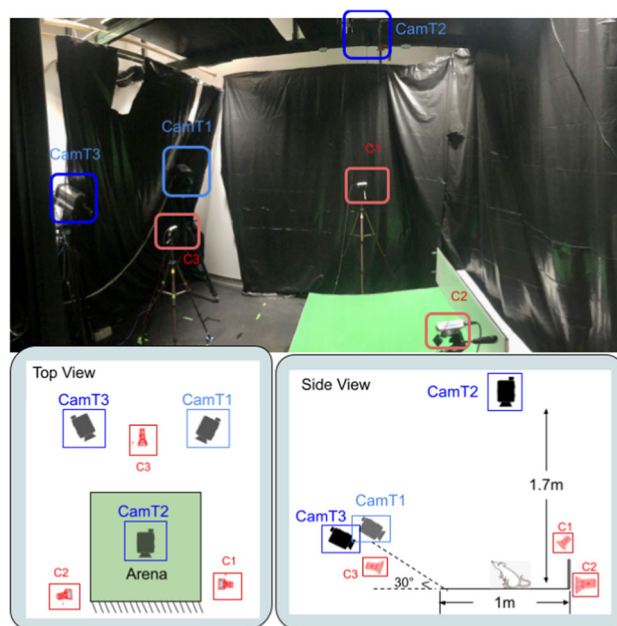


Fig. 3 Setup of video cameras for Rodent3D v3 dataset collection, including thermal cameras (blue, noted as CamT) and RGB-D cameras (red, noted as C). The animal arena (green), 1 m × 1 m, has either edges or walls on its sides

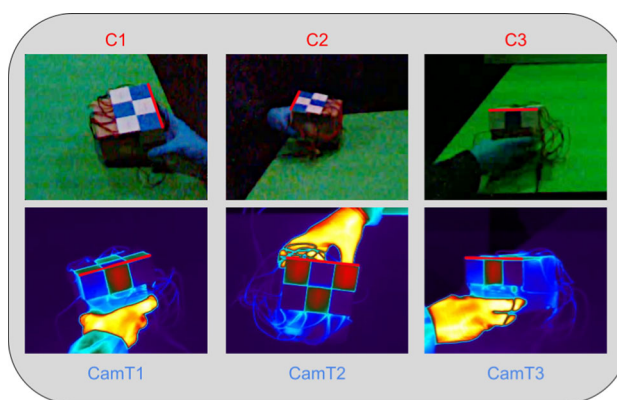


Fig. 4 Thermal calibration object from three RGB-D camera views (top) and three thermal infrared camera views (bottom)

included several recording sessions of variable length. We named the data resulting from each of the three stages Rodent3D v1, v2, v3 (v for version), respectively, and summarize their differences in Table 1. Two sets of cameras are used in collecting the multimodal video footage. One set consists of two or three thermal infrared cameras (FLIR SC8000, FLIR Systems, Inc.). We collected videos at 1024×1024 spatial resolution and 14-bit thermal resolution per pixel. The FLIR cameras were synchronized using a high resolution function generator. FLIR's High Speed Data Recorder was used to prevent frame drops. The other camera set consists of three RGB-D cameras (Intel RealSense D435, Intel, Inc). The RGB-color module is synchronized with the depth mod-

Table 1 Rodent3D Versions. Please see the supplementary material for the camera arrangement for each version

Rodent3D Version	Thermal Module		RGB-D Module		Tot. Len. (min)	Tot. Num. Frames
	Num Cams	Frame Rate	Num Cams	Frame Rate		
v1	2	120	n.a.	n.a.	108	1,550 k
v2	2	120	3	30	55	1,097 k
v3	3	60	3	60	90	1,937 k

ule. Both the color and the depth recordings were collected with a spatial resolution of 848×480 pixels.

We built a circuit to hardware-synchronize the three cameras, following the manufacturer recommendations. We further employed an additional RealSense D435 camera as the master trigger to start the recordings among the three worker-cameras simultaneously.

To synchronize the thermal camera set and the RGB-D camera set to each other, we needed a distinct signal that can be viewed by both modalities. We used a lighter to create a flash, visible in all cameras, at the beginning of each recording session. The brief appearance of the flash serves as the starting time point to align the recordings collected from the two camera sets.

The six cameras were situated as shown in Fig. 3 in Rodent3D v3. On the top is a photo of the recording room with the six cameras in it. On the bottom are illustrations of the top and side views. For camera arrangements in Rodent3D v1 and v2, we show top views in the supplementary material.

3.2 Camera Calibration

For the RGB-D cameras, we applied the on-chip self-calibration for the depth module, a feature included in the firmware provided by the manufacturer. For the RGB module, we used the traditional computer-vision method that involves a flat checkerboard. This method, however, is not applicable in the setting of thermal cameras. We therefore designed a calibration cube made with thermoelectric devices that emit heat when electricity is passed through them. The cube can thus be imaged by our thermal cameras as shown in Fig. 4.

We used the EasyWand method (Theriat et al., 2014) to process the annotated calibration data with bundle adjustment and generated the direct linear transformation (DLT) coefficients. In the case of Rodent3D v3, where 6 cameras are involved, the obtained DLT coefficients matrix is in the shape of 6 by 12. It projects all camera views (both thermal and color) to a common 3D space. In addition to the calibration accuracy, the orientation of the cameras towards the rodent arena also affects the quality of the reconstruction. To minimize the reconstruction uncertainty, we used the Easy-Camera method (Theriat et al., 2014) to optimize the design of our camera configuration.

3.3 Data Curation

We provide both the original data and curated data. The original data collected by the FLIR thermo cameras are in *.ats* format and those by RealSense RGB-D are in *.bag*. We provide scripts that process the *.bag* files, align the depth with the color, handle the occasional frame drop, and re-align timestamps. The resulted files of such processing includes *.mp4* videos for color information, pickled files for depth value for each pixel in all frames, and a *.csv* data sheet for checking timestamps.

3.4 Rodent3D Annotations

We provide human-annotated 2D keypoints which can be used to train a 3D keypoint tracker, such as DeepLabCut (Mathis et al., 2018), and the weights of the DeepLabCut model. We manually annotated about 800 frames per thermal view and over 200 frames per RGB-D view. We trained DeepLabCut models separately on each of the views. The Rodent3D dataset also includes both 3D raw and 3D refined keypoint reconstructions of 3D poses that our pipeline produced.

4 The OptiPose Model

OptiPose is a supervised model that refines the raw 3D keypoints reconstructed from the triangulation of articulated body poses, yielding refined pose estimates. See Fig. 5 for the workflow.

The central component of OptiPose is a number of “context models” (CM) that build on the idea of Self Attention encoders (Vaswani et al., 2017). We describe the architecture of the context models, the loss function used by OptiPose, and a masked keypoint data augmentation algorithm used for training.

We note here that OptiPose processes a sequence of poses simultaneously, meaning that a sequence $P_{1..T}$ of T 3D poses, i.e., sets of 3D keypoints $\{x_1, x_2, \dots, x_N\}_{t=1, \dots, T}$, depicting a certain motion, are processed by OptiPose one set at a time. Therefore, during inference, OptiPose can be operated in a sliding window manner with or without overlap. We also stress that OptiPose does not consist of RNNs (Sherstinsky,

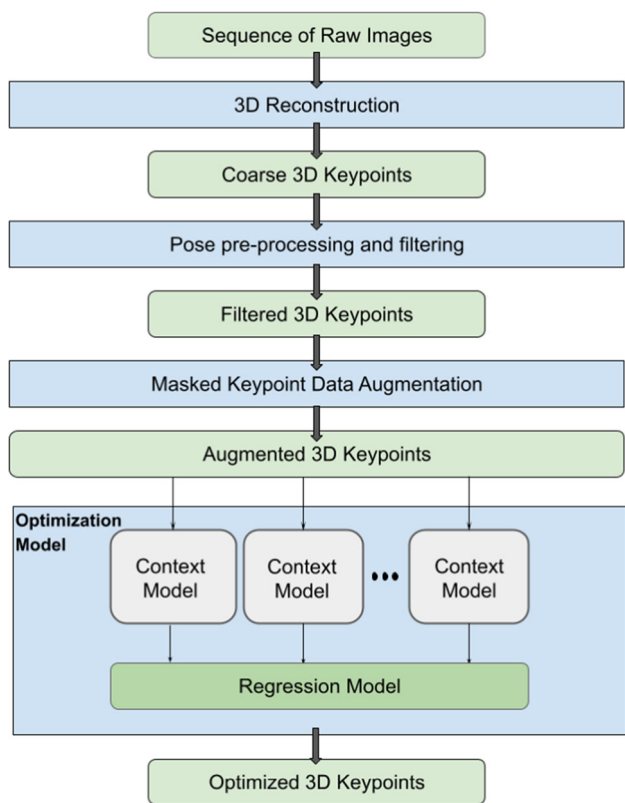


Fig. 5 Workflow for training an OptiPose model. For inference, we skip the masked keypoint data augmentation step

2020), which are commonly used to interpret temporal data but which typically take much more time to train than attention models.

4.1 OptiPose Model Architecture.

The core of OptiPose is a set of parallel context models (CMs), shown in grey boxes in Fig. 6. Each CM receives a sequence of 3D poses, i.e. tokens and computes an embedding vector. We denote the token as P and a sequence of tokens as $P_{1...T} \in \mathbb{R}^{3N}$, where the maximum length of the sequence is denoted by T and N is the number of keypoints per pose token. The output of each CM is in the dimension of $T \times E$ (e.g., we set $E = 64$ in our experiments). These outputs are combined and fed into a fully connected dense linear layer (light green box after combining the CM blocks in Fig. 6) that reshapes the dimensionality to $T \times 3N$ dimensions, generating the offsets for refining the poses. Finally, these offsets are added to the masked input to obtain the refined 3D poses.

Each CM consists of a sequence of sub-CMs, that takes a $T \times 3N$ input and generates an output of the same dimension. Each sub-CM has a standard attention block (blue) (Vaswani et al., 2017) and one fully connected layer (green). The output of the attention block is concatenated horizontally with its

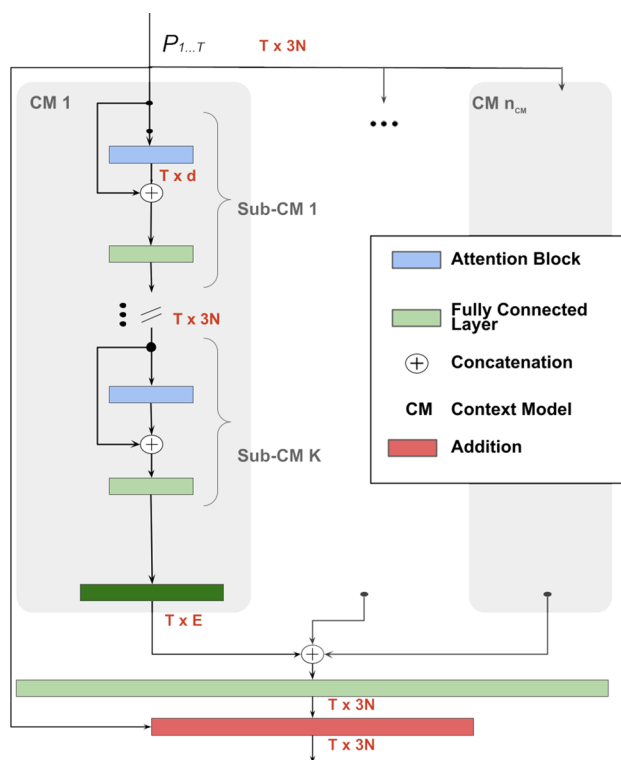


Fig. 6 Architecture of OptiPose. It consists of parallel context models CMs (grey boxes) that share the same input $P_{1...T}$ but whose weights are initialized differently during training. Each CM contains K submodels. A submodel contains an attention block and two fully connected layers

input and fed forward to the fully connected layer. Keeping the input and output dimensions of a sub-CM the same allows us to vary the number K of sub-CMs.

The attention block in OptiPose is implemented as a multi-headed self-attention model (Vaswani et al., 2017). To enable OptiPose to learn to pay attention to different spatio-temporal patterns, we employ multiple CMs in parallel, giving it an ensemble effect. OptiPose learns to pay attention to the spatial relationships between keypoints in a pose, and temporal relationships among poses. It weighs keypoint information differently in case of occlusion. The temporal attention mechanism is particularly important when sets of keypoints are missing in consecutive frames and the locations of these keypoints need to be estimated. This is further explored in the supplementary material.

All fully connected layers in the CMs have a Parametric ReLU (PReLU) as the nonlinear activation function. Using PReLU ensures that our model does not have an upper limit on the range of keypoint locations, making the prediction-range task-adaptive (i.e., laboratory scale for the rodent and wildlife-enclosure scale for the cheetah). The number of parallel CMs (n_{cm}), sub-CMs (K), heads (n_{heads}), as well as the parameters that determine the length of attention (d) and embedding encodings (E), are hyperparameters that can be

adjusted according to the number of keypoints in different animal species.

4.2 Loss with Spatiotemporal Constraints

Keypoints of interest in many animal pose estimation tasks are connected to one another and the distances between some keypoints are constrained by rigid bones. We capture such constraints using an undirected graph $G = (V, E)$ where the graph nodes set V represents a set of keypoints and the edge set E represents the distance between keypoints. We incorporate this geometric constraint and define the structure loss (Moreno-Noguer, 2017) as

$$\mathcal{L}_{st} = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N (\|x_i - x_j\|_2 - \|\hat{x}_i - \hat{x}_j\|_2)^2, \quad (1)$$

where N is the number of keypoints, and $x_i, \hat{x}_i \in \mathbb{R}^3$ are the predicted and the groundtruth coordinates of the i -th keypoint, respectively. We also introduce a temporal constraint over T consecutive poses that is designed to capture the movement pattern of the animal, called temporal loss (Cheng et al., 2020),

$$\mathcal{L}_{tp} = \frac{1}{T} \sum_{t=1}^T \sum_{i=1}^N \left\| (x_i^{(t)} - x_i^{(t-1)}) - (\hat{x}_i^{(t)} - \hat{x}_i^{(t-1)}) \right\|, \quad (2)$$

where N is the number of keypoints, $x_i^{(t)}$ is the predicted position and $\hat{x}_i^{(t)}$ the groundtruth at time t . We define our combined loss function \mathcal{L} as a weighted sum of structural and temporal losses (Zheng et al., 2021):

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \left(\frac{1}{N} \sum_{i=1}^N \|x_i - \hat{x}_i\|_2 + \alpha \mathcal{L}_{st} \right) + \beta \mathcal{L}_{tp}, \quad (3)$$

where α and β are hyperparameters weighing the structural loss \mathcal{L}_{st} and the temporal loss \mathcal{L}_{tp} .

4.3 Masked Keypoint Data Augmentation

We cannot use the standard data augmentation techniques used in computer vision (Shorten & Khoshgoftaar, 2019), given that OptiPose manipulates 3D keypoints and not images. Nonetheless, we can employ geometric transformations such as rotation and translation, noise addition, and the masking of a subset of poses for augmentation. We named this procedure the *Masked Keypoint Data Augmentation Algorithm* and show its pseudo code in Algorithm 1. In the pseudo code, the input variable T refers to the length of the pose sequence as previously defined, $size$ refers to the desired

Algorithm 1 Masked Keypoint Data Augmentation

Input $T, size, datasets, p$
Output $Dataset_{aug}$
 Initialization: $Dataset_{aug} \leftarrow$ empty array
while $len(Dataset_{aug}) < size$ **do**
 for $d \in datasets$ **do**
 $P \leftarrow$ sample $\leq T$ consecutive poses $\in d$
 $f \leftarrow$ select masking function with prob. p
 $P' \leftarrow f(P)$
 $r \leftarrow$ sample from $\{0, \dots, 360^\circ\}$ degree
 $rotate(P', Z\text{-axis}, r)$
 $v \leftarrow$ random translation vector
 $translate(P', v)$
 $add_random_noise(P')$
 $insert(Dataset_{aug}, P')$
end for
end while

size of the augmented dataset, $datasets$ refers to the original datasets, and p refers to the probabilities assigned to the masking functions. Further details are provided in the supplementary materials.

Masking enables us to train OptiPose to handle the occlusion of keypoints on the animal such as an occluded ear or foot. Such occlusions may not appear in the training set otherwise, and thus the purpose of the augmentation algorithm is to maximize the variety of pose sequences in the total training set to encourage the model to learn the structure of the animal. Relying on random decisions at every step ensures that the augmented data, while still physically valid, is substantially different from the original data, and thus enables OptiPose to generalize and correctly interpret unseen data. Figure 7 gives an example of augmenting a sequence of poses during a turning movement of a rat by geometric transformations (prior to masking).

5 Benchmarking Experiments

This section describes three sets of experiments: (1) OptiPose on Rodent3D, (2) OptiPose on other datasets and (3) ablation studies. We denote the OptiPose model with n_{cm} parallel CMs, K sub-CMs and n_{heads} attention heads as OptiPose- n_{cm} - K - n_{heads} . Different datasets, due to their differences in the number of keypoints used, require different OptiPose model variants. We report the performance of OptiPose variants in the ablation studies.

5.1 OptiPose on Rodent3D

We first report results of OptiPose on Rodent3D v3 on both color and thermal data. We trained and evaluated OptiPose separately on the different modules (Table 2). We then applied the OptiPose model trained on the Rodent3D v2 color dataset on a thermal dataset that was recorded in a differ-

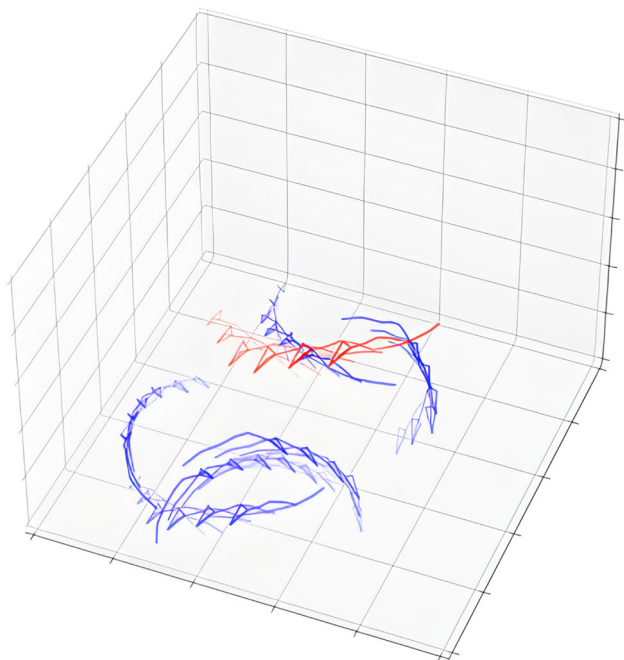


Fig. 7 An illustration of the first step of the proposed data augmentation method. The original sequence of poses (red), showing a rat turning, is augmented by translations and rotations of the sequence (blue) in the animal arena. The remaining augmentation steps, i.e., adding noise to the keypoint locations and dropping subsets of keypoints to mimic occlusion are not shown here

ent session (Table 3). Note that in v2, the color and thermal videos are recorded at different frame rates, but this does not affect the model inference. This experiment shows that OptiPose generalizes well on unseen data type and unseen animal behavior.

The third experiment in this subsection compares the performance of OptiPose with Anipose (Karashchuk et al., 2021) on Rodent3D v3 data. Anipose is a publicly available tool that shares the same overall pipeline as OptiPose. To produce 3D estimates, Anipose first uses checkerboard and bundle adjustment for camera calibration, applies 2D filters on DeepLabCut 2D predictions, triangulates, and applies 3D filters which incorporate temporal smoothing and distance constraints. Since our thermal cameras were not calibrated using checkerboards, which is required in Anipose for a robust estimation, we performed the comparison on the color data.

The 3D groundtruth based on triangulation data could be noisy while 2D labels annotated manually are verifiable. We therefore compare the two methods by looking at 2D Euclidean distance in pixels, as Joska et al. (2021) did in their work. Specifically, we reprojected the 3D keypoints predicted by the aforementioned methods to each of the 2D camera views and compared their errors with respect to hand-labeled 2D positions.

OptiPose on Rodent3D. Qualitative visualizations showed that OptiPose improves the 3D tracking and corrects the pose where the keypoints are partially occluded, mis-detected in 2D or incorrectly reconstructed by triangulation (Supplemental Videos). To quantify such improvement, we evaluate the performance of OptiPose using the standard Mean Per Joint Position Error (MPJPE) and Probability of Correct Keypoint (PCK) metric. For PCK, we report both PCK@0.05 and PCK@0.1 where the range is the maximum distance between any pair of groundtruth keypoints.

Here we define our groundtruth as a sequence of 3D poses where the locations of all 3D keypoints are given. The initial coarse 3D keypoints are reconstructed as described below. First, a DeepLabCut model per view was trained on the manual annotations to extract 2D keypoints. We then used the procedure described in Sect. 3.2 to solve for 3D coordinates of the keypoints. During the reconstruction, we only considered 2D points that DeepLabCut deemed as highly likely (>0.95). Since we have three views with the RGB-D camera set while triangulation requires only two, we dynamically excluded unlikely 2D points and use a subset of views to compute the location of the 3D points. Since filtering data based on the likelihood provided by DeepLabCut does not guarantee the quality of reconstruction, we further improved the quality of raw 3D keypoints by removing keypoints with outlier coordinates and then filling them in with interpolation where possible. An “unknown position” flag is used where interpolation does not work because the keypoint was missing in too many frames.

We generated the training dataset by augmenting and masking the 3D keypoints produced by the above steps in a way as mentioned in Sect. 4.3 (see also Alg. 1). The augmented dataset used for training had 25,800 sequences of poses generated from an original set of 9076 groundtruth poses, where pose is defined by 8 keypoints. We used TensorFlow and the Adam optimizer (Kingma & Ba, 2015) for backpropagating with adaptive gradient descent with an initial learning rate of $3.1e-4$ and a step decay. We set OptiPose hyperparameter $\alpha = 0.0001$, $\beta = 0.0001$ for all model variations. These relatively low α and β values were selected based on the observation that the model tends to converge faster if it focuses on individual keypoints before the structural and temporal relations (see Eq. 3). The need for nonzero α and β is shown below through an ablation study (Table 10). It took about 5 h to train an OptiPose model for Rodent3D on a single “RTX 3080” GPU workstation.

The testing dataset is generated from a video recorded on a different day than the training dataset. The test video contains 3713 complete poses, which we augmented using Algorithm 1 (Sect. 4.3) to generate a test dataset of 6500 sequential poses of length $\leq T$.

We report the results of OptiPose on Rodent3D v3 data when trained within modality in Table 2. On average, Opti-

Table 2 Average PCK accuracy and MPJPE of OptiPose-10-3-1 per keypoint over 6,500 sets of $n \leq T$ consecutive poses on the Rodent3D v3 dataset, compared to the baseline accuracy of noise-added keypoints (top: color camera module, $T = 30$, bottom: thermal camera module, $T = 60$). The baseline “Noisy Input” accuracy is PCK@0.1.

Rodent3D:	Snout	RightEar	LeftEar	HeadBase	Mid	TailBase	TailMid	TailTip	Avg
Noisy Input	64.50	64.52	64.73	64.86	65.00	65.68	64.97	64.90	64.90
PCK@0.05	76.95	84.34	85.20	87.20	88.76	85.68	86.08	78.62	84.11
PCK@0.10	86.51	92.72	92.69	93.76	93.20	91.45	91.86	86.47	91.08
MPJPE↓	13.28	9.34	9.25	8.11	7.27	9.41	9.69	15.79	10.27
Rodent3D:	Snout	RightEar	LeftEar	HeadBase	Mid	TailBase	TailMid	TailTip	Avg
Noisy Input	65.26	65.77	65.19	65.73	66.16	65.23	65.82	65.33	65.55
PCK@0.05	78.68	82.13	82.19	78.13	83.80	82.26	80.37	74.34	80.94
PCK@0.10	89.40	92.69	92.94	94.22	93.16	92.05	90.41	86.36	91.90
MPJPE↓	12.09	9.87	9.49	14.80	10.55	9.83	10.77	14.20	11.25

Table 3 Average PCK accuracy and MPJPE of OptiPose-10-3-1, trained on 30 Hz RGB data of Rodent3D v2, over 6,500 sets of $n \leq 30$ consecutive poses, recorded at 120 Hz from thermal cameras, compared to the baseline PCK@0.1 accuracy of noise-added keypoints

Rodent3D:	Snout	RightEar	LeftEar	HeadBase	Mid	TailBase	TailMid	TailTip	Avg
Noisy Input	64.86	65.51	64.87	65.41	64.55	64.62	65.30	65.12	65.03
PCK@0.05	79.79	85.24	84.80	87.43	88.59	84.86	85.22	79.80	84.47
PCK@0.10	88.08	92.41	92.09	93.61	92.79	90.20	90.01	86.49	90.72
MPJPE↓	11.994	9.224	9.764	7.974	7.384	10.104	10.934	16.114	10.43

Pose improves the noise-added inputs by 19 percent points within 5% (PCK@0.05) and by 25 percent points within 10% (PCK@0.1) of their groundtruth range for color data and 15 pp (PCK@0.05) and 26 pp (PCK@0.1) for thermal data. The average per joint position error (MPJPE) ranges from 7 to 15 millimeters for specific keypoints. The positions of the snout and the tail tip are the most difficult to estimate. When trained on RGB and tested on thermal infrared data, the results are similar (see Table 3).

Estimates of raw keypoint locations can be highly inaccurate due to occlusion, missed detection by the 2D keypoint extractor, or errors introduced in the 3D reconstruction, as Fig. 2 illustrates. One may argue that such inaccuracies can be smoothed out or interpolated by a temporal filter, and thus making a keypoint refiner like OptiPose less relevant. However, for keypoints that are missing for a lengthy time period, simple smoothing or interpolation will lead to information loss.

In fact, outlier coordinates of a keypoint cannot easily be substituted. For example, the trajectory of the snout of a rodent in x , y , z , plotted as functions of time in Fig. 8, highlights that OptiPose is able to estimate snout position (blue line) during lengthy periods when the accuracy of raw keypoint locations (orange) is poor and interpolation (dotted red) loses information.

Comparing OptiPose and Anipose. We evaluated the reprojection error of OptiPose and Anipose (Karashchuk et al., 2021). To do that, we used the same set of 2D estimates obtained by running home-trained DeepLabCut models, processed the input via OptiPose and Anipose respectively,

and obtained 3D estimates from the two models. We then reprojected the 3D reconstructed keypoints back to their original 2D planes and compared these reprojected keypoints in 2D with the manually annotated ones. We report the 2D Euclidean distance in number of pixels in Table 4. OptiPose outperforms Anipose for all keypoints in all views. It is noticeable that data from the front-view camera (C3), see Fig. 3, yields the most error in both OptiPose and Anipose. This is consistent with our observation that the occasional frame drops happened mostly in the recordings from cam3. Frame drops may have been caused by the wall on the arena reflecting lights to the front camera.

5.2 OptiPose on other datasets

We provide experimental results with our OptiPose on the three publicly available datasets Rat7M (Dunn et al., 2021), AcinoSet (Joska et al., 2021), and the DeepFly3D Drosophila dataset (Günel et al., 2019). We report our result quantitatively and qualitatively.

OptiPose on Rat7M. We conducted experiments to evaluate OptiPose on the motion capture data provided in the Rat7M dataset (Marshall et al., 2021a). We trained OptiPose on the subjects 2, 3, and 5, and evaluated the model performance on subject 4. We generated our training dataset by applying our data augmentation and masking method on the 3D poses defined by 16 keypoints. We excluded joints of left and right elbows and arms because a large portion of these joints are missing (‘ElbowR’: 23.74%, ‘ElbowL’: 10.44%, ‘ArmL’: 9.82%, ‘ArmR’: 22.31%) and missing in blocks

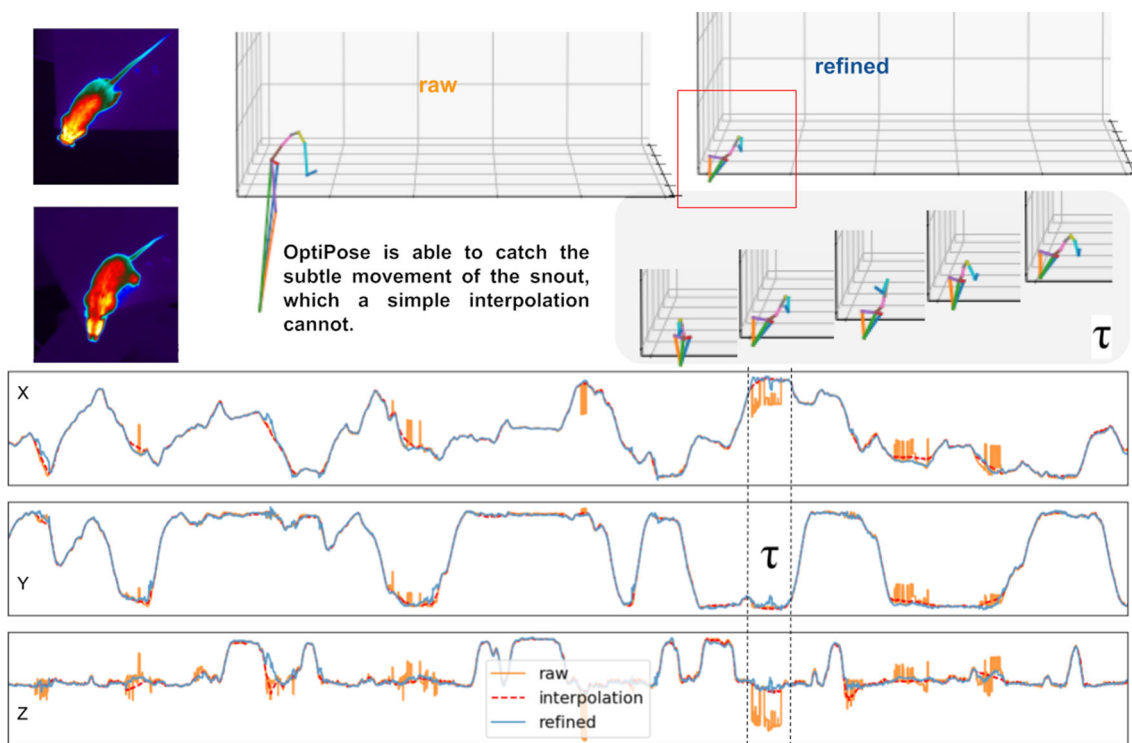


Fig. 8 OptiPose is able to catch movements of a keypoint that simple interpolation is unable to. Top: The 3D pose (red box) of the rodent shown in the infrared images is a refined version of the inaccurate raw 3D pose shown. Bottom: The plots show raw (orange), interpolation

(dotted red) and refined (blue, by OptiPose) keypoint coordinates for a sequence of 12,000 frames (100 s). Interpolation will not provide accurate snout positions, particularly during challenging periods like the 360-frame-long period marked τ

Table 4 Reprojection error in pixels of OptiPose-10-3-1 and Anipose (Karashchuk et al., 2021) per keypoint, computed by projecting the 3D keypoint coordinates back onto the image plane of each RGB-D camera. (The color data we used for this experiment is from the Rodent3D v3)

View	Model	Snout	RightEar	LeftEar	HeadBase	Mid	TailBase	TailMid	TailTip
cam1	Anipose	9.26	9.89	9.21	8.63	7.76	7.52	7.62	7.98
	OptiPose	5.93	5.78	5.19	4.99	6.99	4.17	5.26	5.06
cam2	Anipose	13.23	12.96	13.20	12.34	12.73	11.44	11.86	10.83
	OptiPose	4.89	4.14	4.24	3.95	4.54	3.07	4.37	3.41
cam3	Anipose	59.05	49.33	49.37	47.97	36.45	32.43	32.43	31.33
	OptiPose	57.04	47.27	46.88	45.53	33.49	28.01	28.44	27.41

(thousands of consecutive frames) in the motion capture data. The training set consists of 976 K frames and the test set of 60 K frames. We report the PCK@0.05 and PCK@0.1 values of our Rat 7M experiment in Table 5. Optipose improves the noisy input by ~19% points considering PCK@0.05 and by ~28% points considering PCK@0.1.

Furthermore, we ran this trained OptiPose model on DAN-NCE (Dunn et al., 2021) predictions to see if further improvements could be achieved. DANNCE is a volumetric convolutional neural network that learns from projective geometry from multi-views. The DANNCE model was pre-trained on the Rat7M motion capture data so we used the

pre-trained model directly and ran inference on subject-1 videos to obtain the 3D positions. We then ran OptiPose on those predictions and report our result in Table 6. OptiPose does not provide a significant improvement over a DAN-NCE model using all camera angles. We note that since the DANNCE predictions are already postural-accurate, OptiPose only modifies the baseline predictions slightly.

OptiPose on AcinoSet. We conducted experiments to compare the performance of our OptiPose model on the AcinoSet dataset (Joska et al., 2021) with the performance of the state-of-the-art Full Trajectory Estimation (FTE) technique provided with the AcinoSet (Joska et al., 2021). FTE is a

Table 5 Average PCK accuracy and MPJPE of OptiPose-10-10-2 per keypoint over 6500 sets of $T \leq 30$ consecutive poses on the Rat7M (Marshall et al., 2021a) motion capture dataset

Rat7M:	HeadF	HeadB	HeadL	SpineF	SpineM	SpineL	Offset1	Offset2	
Noisy Input	67.24	66.78	66.55	66.68	66.89	67.07	66.69	66.8	
PCK@0.05	83.72	86.63	85.96	86.23	85.93	86.06	89.66	86.92	
PCK@0.10	94.41	96.27	95.82	95.73	94.61	94.78	97.66	95.72	
MPJPE↓	4.37	3.74	3.81	3.55	3.55	3.76	3.25	3.52	
	HipL	HipR	ShoulderL	ShoulderR	KneeR	KneeL	ShinL	ShinR	Avg
Noisy Input	66.91	66.80	67.32	66.94	66.81	66.93	66.79	67.03	66.89
PCK@0.05	85.50	87.93	85.69	81.99	87.18	84.84	83.61	85.78	85.85
PCK@0.10	94.36	95.88	94.91	94.14	95.72	94.42	93.38	94.72	95.16
MPJPE↓	4.34	3.64	3.67	4.35	3.57	3.97	4.57	3.89	3.85

The baseline “Noisy Input” accuracy is PCK@0.1

Table 6 PCK@0.1 accuracy of OptiPose-10-10-2 on baseline prediction obtained through DANNCE (Dunn et al., 2021) on 50,000 frames from Rat7M (Marshall et al., 2021a)

Rat7M:	HeadF	HeadB	HeadL	SpineF	SpineM	SpineL	Offset1	Offset2	
DANNCE	88.80	88.54	89.67	94.53	92.48	95.15	93.79	95.31	
OptiPose-10-10-2	89.03	88.60	89.96	94.55	92.98	94.76	93.84	95.64	
	HipL	HipR	ShoulderL	ShoulderR	KneeR	KneeL	ShinL	ShinR	Avg
DANNCE Dunn et al. (2021)	95.44	94.02	90.52	91.04	93.96	89.57	92.51	92.63	92.37
OptiPose-10-10-2	95.35	93.83	90.45	91.6	93.44	90.16	91.85	92.68	92.42

During OptiPose inference, the window size is 30 frames and overlap is 20

biomechanics-based constraint optimization method running on data generated from multiple cameras. As recommended (Joska et al., 2021), we treated the 3D cheetah poses reconstructed by FTE from six camera views (FTE6) as the groundtruth 3D poses.

To create a cheetah training dataset for OptiPose, we applied our data augmentation method on the published 6-camera FTE reconstructed 3D poses. For testing, we compared our model performance with the FTE performance in settings where fewer than six cameras are used by FTE to provide 3D pose reconstructions (groundtruth is still obtained by all six views). We denote these baseline models as FTE2 and FTE4, corresponding to the settings where the predictions were generated by running FTE on two or four cameras, respectively (and are therefore sub-par relative to FTE6).

In our experiment, we treated the FTE2 and FTE4 outputs as raw 3D keypoints and set the goal to improve their accuracy with OptiPose. We used 14 videos (2204 frames) for training and two videos (331 frames) for testing. The resulting augmented training dataset had 22,400 sequences of poses. The results indicate that OptiPose outperforms FTE in both settings. It improves the FTE2 by ~ 5 percent points and FTE4 by ~ 2 percent points (Table 7). We note that inference with OptiPose is faster than with iterative constraint optimization (Joska et al., 2021) (seconds versus minutes).

OptiPose on DeepFly3D Drosophila. We conducted experiments to further evaluate OptiPose on the DeepFly3D Drosophila Dataset (Ramdya, 2019). We considered the 3D poses reconstructed by DeepFly3D model (Günel et al., 2019) from seven camera views as the groundtruth. We used eight recordings for creating the training set and two recordings for testing. Each recording contains 900 frames with a spatial resolution of 960×480 . We created our training set as previously described. For testing, we used a subset of five cameras to triangulate 3D locations of all 38 keypoints and aimed to improve their accuracy with OptiPose. The five cameras we chose include one front view and two side views per side, which results in 14,368 out of 68,324 missing keypoints due to occlusion. We evaluated our result qualitatively as shown in Fig. 9 as well as quantitatively in Table 8. Our method improves the initial prediction on average by 19.53 percent points. It indicates that OptiPose is able to learn the poses of a tethered Drosophila and locate keypoints when they are missing. This experiment reinforces that OptiPose can be used to reduce the number of cameras required, even when a large number of keypoints are used to represent the animal.

Table 7 PCK@0.1 accuracy of baselines (Joska et al., 2021) and OptiPose-10-7-1 on the cheetah “run” data of the AcinoSet (Joska et al., 2021)

Method	FTE2	FTE4
Baseline	73.41	90.09
OptiPose	78.37	92.38

The groundtruth is provided by the FTE approach with 6 cameras. During the OptiPose inference, the window size is 30 frames and overlap is 20

Table 8 PCK@0.1 accuracy for the baseline obtained through SBA reconstruction on a subset of cameras and OptiPose-15-5-4 on the DeepFly3d (Günel et al., 2019) dataset

Session	Model	Accuracy
Fly-2-004	Baseline	77.47
	OptiPose	96.25
Fly-2-005	Baseline	77.68
	OptiPose	97.96

During the OptiPose inference, the window size is 30 frames and overlap is 20

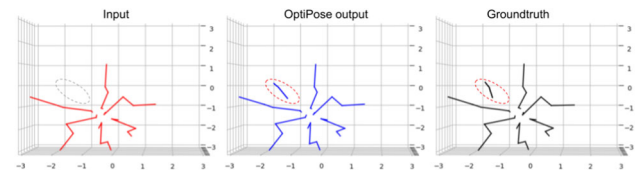


Fig. 9 Qualitative evaluation of OptiPose results on DeepFly3D Drosophila data. The input pose (red) contains missing keypoints on the fly’s body (dashed circle). OptiPose is not only able to fill in the missing keypoints (blue), but also computes location estimates of the keypoints on the body that appear to be anatomically more realistic than the “groundtruth” locations inferred by the DeepFly3D model (black).

5.3 Ablation Studies

We first investigated the question of how many parallel CMs, sub-CMs and attention heads OptiPose should use. We trained a variety of combinations on different datasets and report our result in Table 9. The notation of the OptiPose variants is in the form of $n_{cm}-K-n_{heads}$. It is evident from Table 9 that a higher number of parallel CMs and sub-CMs is beneficial for increasing the overall accuracy of keypoint estimation. The more keypoints that are used to model animal shape and motion, the higher the number of parallel context models should be. In addition, increasing the number of heads may also increase the accuracy. However, the trade-off is that it requires more workload in computation since the number of parameters is increased by $n_{cm} \times K \times n_{heads}$. Furthermore, there is a point of diminishing returns. In particular, the model versions OptiPose-10-5-1 for AcinoSet (20 keypoints) and OptiPose-10-5-2 for DF3D (38 keypoints) are not large enough, and we see a noticeable improvement by doubling the number of heads. This is not true for Rodent3D (8 keypoints) and Rat7M (16 keypoints). For

Table 9 Ablation study. PCK@0.1 and MPJPE averaged across all keypoints for different OptiPose variations $n_{cm}-K-n_{heads}$

Dataset	Model	PCK@0.1	MPJPE	
Rodent3D	3-3-1	91.52	13.01	
	3-3-2	91.58	12.25	
	10-3-1	95.37	8.63	
	10-3-2	95.54	7.72	
	10-5-1	96.10	7.21	
	<u>10-5-2</u>	<u>95.73</u>	<u>7.51</u>	
	Rat7M	7-3-1	92.38	5.81
		7-3-2	91.86	5.72
		10-5-1	92.98	4.92
		10-5-2	93.10	4.75
<u>10-10-1</u>		<u>93.87</u>	<u>4.36</u>	
10-10-2		95.16	3.85	
AcinoSet	7-3-1	67.20	17.28	
	7-3-2	68.71	16.05	
	10-5-1	77.52	12.54	
	10-5-2	<u>85.10</u>	10.71	
	<u>12-5-1</u>	84.67	<u>9.08</u>	
	12-5-2	85.42	8.93	
DF3D	10-5-2	92.99	15.28	
	<u>10-5-4</u>	<u>94.82</u>	<u>10.31</u>	
	15-5-4	94.94	10.03	

We use bold text to indicate the best and underline for the second best

Rodent3D, OptiPose-10-5-2 performs even slightly worse than OptiPose-10-5-1.

Impact of the Loss Function. To study the dependence of our results on the proposed spatio-temporal loss function \mathcal{L} (Eq. 3), we trained OptiPose-10-5-2 with the standard Euclidean loss function ($\alpha = 0$ and $\beta = 0$, in Eq. 3), instead of \mathcal{L} , on four datasets. The results show that the accuracy is higher by 1.3 (Rodent3D), 3.76 (Rat7M), 1 (AcinoSet), and 10 (DF3D) percent points when \mathcal{L} is used; see Table 10. It is evident from Tables 9 and 10 that OptiPose-10-5-2 is sufficiently large for Rodent3D (8 keypoints), Rat7M (16 keypoints) and AcinoSet (20 keypoints); in that, the proposed loss function does not show significant improvements. In comparison, the results on DF3D (38 keypoints) suggest that the proposed loss function can allow us to train a relatively smaller model and still achieve higher accuracy.

Table 10 PCK@1.0 accuracy for OptiPose-10-5-2 trained with the Euclidean loss function $\|L\|$ (i.e., $\alpha = 0$ and $\beta = 0$ in Eq. 3) and the proposed loss function \mathcal{L} on Rodent3D, Rat7M, AcinoSet, and DF3D

Loss Function	Rodent3D	Rat7M	AcinoSet	DF3D
$\ L\ $	94.37	89.34	84.22	82.99
Proposed \mathcal{L}	<u>95.73</u>	<u>93.10</u>	<u>85.10</u>	<u>92.98</u>

6 Behavioral Analysis

Accurate 3D poses allow us to perform complex analysis of animal behavior. Foraging in open field arenas, as recorded in the Rodent3D dataset, is a commonly used task in the study of the neural correlates of animal behavior. While foraging and exploring an environment, animals perform a multitude of behaviors that have been found to correlate with neural activity such as rearing, grooming, darting, and freezing (Dunn et al., 2021). Analyzing 3D pose sequences provided by OptiPose, we were able to answer questions about where the rat in our experiments spent most of its time: The hot-spots in the top left plot in Fig. 10 are next to the positions of the automatic food dispensers. Similarly, rearing occurred at the food dispensers (Fig. 10 top right), particularly at the back wall, and also at the arena corners.

In addition to coding for specific poses of the animal, neurons have been found to encode relationships between the animal and the environment, such as head direction, movement direction, current spatial location, distance as well as the direction from boundaries, and the alterations within the environment (Alexander et al., 2020; Carstensen et al., 2021; Dannenberg et al., 2020). An unreported aspect of rodent behavior is determining how frequent the body direction is guided by the head direction of the animal, which is important for understanding how body direction of the animal differs from the mechanisms for directed perception involving the head direction (viewing angle) (Raudies et al., 2015). We define the head direction as a vector from HeadBase to Snout keypoints and the body direction as a vector from Midpoint to HeadBase keypoints. We identified rearing and heading behaviors based on the two vectors. The plot on the bottom of Figure 10 indicates all locations in which the head direction led the body direction of the animal. The positions shown with higher values are near a known reward location. It is evident that the animal likely turned its head to face a reward location, then moved in the corresponding direction to navigate to the reward.

We also investigated in which direction the rodent is facing mostly throughout the session. We assumed that the rat’s eye-line is aligned with the head direction vector and traced it to one of the five surfaces of the arena—left wall, far wall, right wall, near wall (imaginary—the arena boundary is a table

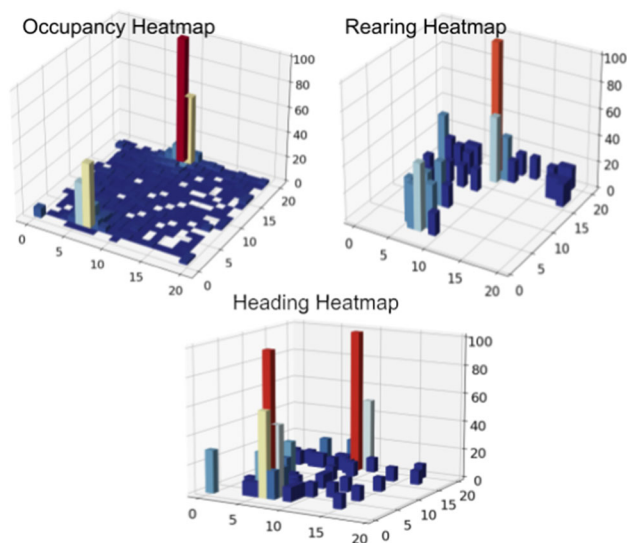


Fig. 10 Spatial heatmaps measuring occurrence of rodent behaviors across a 1 m by 1 m laboratory arena. Top left: Discrete locations of the rodent in bin units of 5 cm by 5 cm. The regions of highest occupancy values correspond to the two locations where food pellets were dropped by apparatuses on opposite walls. Top right: Locations where rodent performed a “rearing” action, notably near the food dispensers and arena corners. Bottom: Locations of “heading events” where the body direction consistently followed the head direction, notably between and at food dispensers

edge), and the floor. The resulting heatmap, normalized by the frame rate, is shown in Fig. 11. The fixation result confirms our previous conclusion that the rat focuses more on the area near the rewards location. Furthermore, our fixation tracking method enables neuroscientists to perform quantitative behavioral experiments with audio-visual stimuli.

The spatial heatmaps shown in Figs. 10 and 11 are examples of types of behavioral analysis that can be performed with Rodent3D and OptiPose. We provide this analysis code, as well as the tracked coordinates and timestamps in the supplemental materials (<http://www.cs.bu.edu/faculty/betke/Rodent3D>).

7 Discussion

7.1 Rodent3D

Rodent3D is the only publicly available dataset that includes thermal infrared videos of rodents (to the best of our knowledge).

Some animals may be more comfortable in low or no light conditions. It would be desirable to record those animals in a somewhat typical ethological setting. In the case of rodents, which are nocturnal animals, thermal recordings could be important when recording with traditional IR or RGB cam-

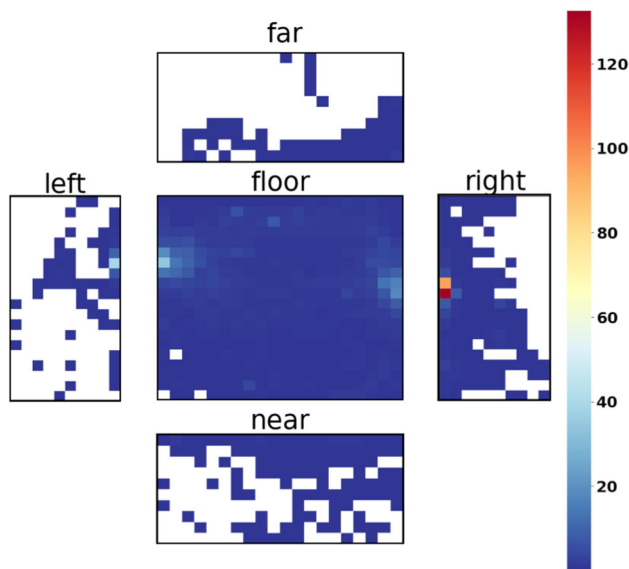


Fig. 11 Heatmap showing rat's fixation over 5 cm by 5 cm blocks on the surfaces of the arena. The surfaces include the 1 m by 1 m floor and four 50 cm high walls—left, far, right, and near (imaginary). The frequencies were normalized by the frame rate. The color represents seconds spent by the rodent, fixating on the corresponding location

eras is not possible or induces experimental variables such as the animals being able to perceive IR emission from traditional IR cameras. Thermal cameras also provide additional tracking capability compared to RGB in situations where there is little contrast between the subject and the background (such as a black animal at night or against a dark background).

Current annotations on our Rodent3D data are limited to keypoints along the head and spine of a rodent and the tips of the two ears. We did not annotate joints on the limbs like Rat7M does. These keypoints were chosen by the neuroscientists on our team to address questions from experimental neuroscience about the position, orientation, and velocity of the rodent's head and body. Most previous neurophysiological studies of rodents use a single overhead camera for visible light 2D tracking of the position and direction of the head with LEDs mounted on an implant attached to the skull (Alexander et al., 2020; Carstensen et al., 2021; Høydal et al., 2019; O'Keefe & Burgess, 2005). These previous studies did not answer the question whether neurons respond to only head orientation and velocity versus body orientation and velocity (Dannenberg et al., 2020; Raudies et al., 2015). Existing methods (Karashchuk et al., 2021; Mathis et al., 2018), although they can generate the required 2D/3D kinematics, may not be sufficiently accurate during occlusions (particularly long occlusions). This can be resolved by using our OptiPose method to refine their predictions.

7.2 OptiPose

Our results show that OptiPose generalizes well to various animal species, performs well on two different input modalities, and can handle train/test switches from modalities and frame rates. In particular, the fact that the OptiPose model, trained on 30 Hz color recordings, can predict 3D locations of keypoints in 120 Hz thermal recordings with similar accuracy (Table 3 compared to Table 2 Bottom) suggests that OptiPose is capable of understanding relations among keypoints both spatially and temporally.

The hyperparameters selection for OptiPose is overall a forgiving process. Based on our observation, the parameter n_{cm} should be selected first. From our experiments, we found that $n_{cm} = 10$ is sufficient for up to 38 keypoints (Günel et al., 2019). To fine-tune this parameter, an analysis of each CM's contribution can be performed (refer to supplementary material). If n_{cm} is selected appropriately, we can reduce the number of sub-CMs (K) and increase the number of heads (n_{heads}) for a significant performance boost (see AcinoSet-10-5-1 and AcinoSet-10-5-2 in Table 9). For training, we observed that setting α and β to low values increases the convergence rate of the model. We believe it is because the model needs to first understand each keypoints independently by the Euclidean-distance loss function and then focus on the spatial and temporal context. However, fine-tuning the model later with higher α and β values can provide further refinement.

We showed that using OptiPose on predictions generated by FTE (Joska et al., 2021) and DeepFly3D (Günel et al., 2019) on their respective datasets but with fewer cameras, generated comparable results to their output based on the full set of cameras. OptiPose allows users to simplify their recording system without giving up a high level of accuracy. Furthermore, we showed that our model is not dependent on any specific recording setting. It was shown to work for open-field data (AcinoSet), tethered subjects (DeepFly3D), and small arenas (Rodent3D or Rat7M).

The PCK measure commonly used to quantify *single* pose accuracy has limited utility for assessing a pose tracking model. For the reader to obtain a qualitative understanding of the accuracy of 3D pose *tracking* with OptiPose, we encourage viewing the videos submitted as supplemental materials at <http://www.cs.bu.edu/faculty/betke/OptiPose>.

It could be viewed as a limitation of our approach that we did not include CAD models into the training, as proposed by recent work (Li & Lee, 2021). In Sect. 2, we argue that design and/or verification of dynamic CAD models of animals requires biomechanical expertise to ensure realistic modeling, particularly of limb movements. However, our problem statement was to avoid such expertise. Nonetheless, it should be noted, if an accurate dynamic CAD model of the animal species of interest exists, it could be used for train-

ing of OptiPose and would likely make its results even more accurate.

7.3 Future Work

OptiPose does not have access to the original input frame images. Therefore, the performance of the model is dependent upon the quality of the 3D raw keypoints in the current input window. Furthermore, if its inputs are generated by a procedure that fits postural dynamics but does not match groundtruth, OptiPose may only show a slight improvement, caused by its temporal capabilities (Table 6). This issue may be circumvented by future work on integrating an existing pose estimation architecture with a pretrained OptiPose model. Such an end-to-end system may benefit from OptiPose's insights on the structure and movement patterns of the animal.

Future work will utilize the depth data, which is included in the Rodent3D dataset but has not been used yet. It will integrate the OptiPose model with a mesh or point-cloud auto-encoder model to filter the subject's mesh in a 3D reconstructed scene.

8 Conclusions

We provide a multimodal dataset of recordings of one or two rodents exploring a laboratory arena from up to three thermal and three RGB-D synchronized cameras. The dataset, camera calibration data, annotations, our OptiPose model and behavior analysis code will be made available with publication of this paper. The former may support computer vision researchers in developing animal pose estimation models, the latter may be valuable for scientists in need of rodent image analysis tools.

The proposed contextual model OptiPose employs a token-based self-attention mechanism to learn spatio-temporal patterns of sequences of 3D keypoints. Compared to other methods, OptiPose refines 3D keypoint predictions quickly and without requiring hard-coded spatial and temporal constraints on the subject movement that would require biomechanical expertise. OptiPose model performance on the tested datasets featuring broad variance in imaging scene and modality (infrared and visible-light videos), as well as animal of interest (rat, cheetah, and fly) suggests a relatively high degree of generalizability.

The way we adapted a self-attention model to interpret spatio-temporal content contextually may be useful for other problems in computer vision that involve 2D or 3D spatial

patterns and their movements in time, beyond pose estimation.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11263-022-01714-5>.

Acknowledgements This work has been partially supported by ONR MURI grant N00014-19-1-2571 associated with AUSMURIB000001.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alexander, A. S., Carstensen, L. C., Hinman, J. R., Raudies, F., Chapman, G. W., & Hasselmo, M. E. (2020). Egocentric boundary vector tuning of the retrosplenial cortex. *Science Advances*, 6(8), eaaz2322.
- Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A., & Cipolla, R. (2020). Who left the dogs out: 3D animal reconstruction with expectation maximization in the loop. In *16th European conference on computer vision*, Glasgow UK August 23 to 28, 2020, Proceedings Part XI
- Breslav, M., Hedrick, T. L., Sclaroff, S., & Betke, M. (2016). Discovering useful parts for pose estimation in sparsely annotated datasets. In *Proceedings of the IEEE winter conference on applications of computer vision (WACV)*, Lake Placid, NY
- Carstensen, L. C., Alexander, A. S., Chapman, G. W., Lee, A. J., & Hasselmo, M. E. (2021). Neural responses in retrosplenial cortex associated with environmental alterations. *iScience* p. 103377
- Cheng, Y., Yan, B., Wang, B., & Tan, R. T. (2020). 3D human pose estimation using spatio-temporal networks with explicit occlusion training. In *The thirty-fourth AAAI conference on artificial intelligence (AAAI-20)*, (pp. 10631–10638)
- Dannenberg, H., Lazaro, H., Nambiar, P., Hoyland, A., & Hasselmo, M. E. (2020). Effects of visual inputs on neural dynamics for coding of location and running speed in medial entorhinal cortex. *Elife*, 9, e62500.
- Dunn, T. W., Marshall, J. D., Severson, K. S., Aldarondo, D. E., Hildebrand, D. G., Chettih, S. N., Wang, W. L., Gellis, A. J., Carlson, D. E., Aronov, D., et al. (2021). Geometric deep learning enables 3D kinematic profiling across species and environments. *Nature methods*, 18(5), 564–573.
- Gong, K., Zhang, J., & Feng, J. (2021). PoseAug: A differentiable pose augmentation framework for 3D human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (pp. 8575–8584)
- Gosztolai, A., Günel, S., Ríos, V. L., Abrate, M. P., Morales, D., Rhodin, H., Fua, P., & Ramdya, P.: LiftPose3D, a deep learning-based approach for transforming 2D to 3D pose in laboratory

- animals. *bioRxiv* (2021). <https://www.biorxiv.org/content/early/2021/04/12/2020.09.18.292680>
- Graving, J. M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B. R., & Couzin, I. D. (2019). DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife*, 8, 1–42. <https://doi.org/10.7554/eLife.47994>
- Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., & Fua, P. (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult drosophila. *eLife*, 8, e48571.
- Høydal, Ø. A., Skytøen, E. R., Andersson, S. O., Moser, M. B., & Moser, E. I. (2019). Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752), 400–404.
- Hu, B., Seybold, B., Yang, S., Ross, D. A., Sud, A., Ruby, G., & Liu, Y. (2021). Optical Mouse: 3D mouse pose from single-view video. <https://arxiv.org/abs/2106.09251>
- Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1325–1339.
- Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y. (2019). Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 7718–7727)
- Joska, D., Clark, L., Muramatsu, N., Jericevich, R., Nicolls, F., Mathis, A., Mathis, M. W., & Patel, A. (2021). AcinoSet: A 3D pose estimation dataset and baseline models for cheetahs in the wild. [arXiv: 2103.13282](https://arxiv.org/abs/2103.13282)
- Karashchuk, P., Rupp, K. L., Dickinson, E. S., Azim, E., Brunton, B. W., & Tuthill, J. C. (2021). Anipose: A toolkit for robust markerless 3D pose estimation. *Cell Reports* 36(13)
- Kearney, S., Li, W., Parsons, M., Kim, K., & Cosker, D. (2020). RGBD-Dog: Predicting canine pose from RGBD sensors. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (pp. 8333–8342), <https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00836>
- Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International conference on learning representations (ICLR)*
- Lauer, J., Zhou, M., Ye, S., Menegas, W., Nath, T., Rahman, M. M., Di Santo, V., Soberanes, D., Feng, G., Murthy, V. N., Lauder, G., Dulac, C., Mathis, M. W., & Mathis, A. (2021). Multi-animal pose estimation and tracking with DeepLabCut. *bioRxiv*, <https://www.biorxiv.org/content/early/2021/04/30/2021.04.30.442096>
- Li, C., & Lee, G. H. (2021). From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (pp. 1482–1491)
- Li, S., Günel, S., Ostrek, M., Ramdya, P., Fua, P., & Rhodin, H. (2020). Deformation-aware unpaired image translation for pose estimation on laboratory animals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (pp. 13158–13168)
- Li, W., Liu, H., Ding, R., Liu, M., Wang, P., & Yang, W. (2022). Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*. <https://doi.org/10.1109/TMM.2022.3141231>
- Lin, K., Wang, L., & Liu, Z. (2021). End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (pp. 1954–1963)
- Liu, X., Yu, S. y., Flierman, N., Loyola, S., Kamermans, M., Hoogland, T. M., & De Zeeuw, C. I. (2020). OptiFlex: Video-based animal pose estimation using deep learning enhanced by optical flow. *BioRxiv*
- Marshall, J. D., Aldarondo, D., Wang, W. P., Ölveczky, B., & Dunn, T. (2021). Rat 7m. <https://doi.org/10.6084/m9.figshare.c.5295370.v3>
- Marshall, J. D., Klibaite, U., Gellis, A. J., Aldarondo, D. E., Ölveczky, B. P., & Dunn, T. W. (2021). The pair-r24m dataset for multi-animal 3d pose estimation. *bioRxiv*
- Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, (pp. 2640–2649)
- Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience* 21, 1281–1289, <http://www.nature.com/articles/s41593-018-0209-y>
- Mathis, A., Schneider, S., Lauer, J., & Mathis, M. W. (2020). A primer on motion capture with deep learning: Principles, pitfalls, and perspectives. *Neuron*, 108(1), 44–65.
- Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 fifth international conference on. IEEE (2017)*. <https://doi.org/10.1109/3dv.2017.00064>, http://gvy.mpi-inf.mpg.de/3dhp_dataset
- Monsees, A., Voit, K. M., Wallace, D. J., Sawinski, J., Leks, E., Scheffler, K., Macke, J. H., & Kerr, J. N. (2021). Anatomically-based skeleton kinetics and pose estimation in freely-moving rodents. *bioRxiv*
- Moreno-Noguer, F. (2017). 3D human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, (pp. 2823–2832)
- Mu, J., Qiu, W., Hager, G., & Yuille, A.L. (2020). Learning from synthetic animals. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, (pp. 12383–12392)
- Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2018). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *bioRxiv*. <https://doi.org/10.1101/476531>, <https://www.biorxiv.org/content/early/2018/11/24/476531>
- O’Keefe, J., & Burgess, N. (2005). Dual phase and rate coding in hippocampal place cells: Theoretical significance and relationship to entorhinal grid cells. *Hippocampus*, 15(7), 853–866.
- Pavlakos, G., Zhou, X., Derpanis, K. G., & Daniilidis, K. (2017). Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 6988–6997)
- Pereira, T. D., Aldarondo, D. E., Willmore, L., Kislin, M., Wang, S. S. H., Murthy, M., & Shaevitz, J. W. (2019). Fast animal pose estimation using deep neural networks. *Nature Methods*, 16(1), 117–125.
- Ramdya, P.P. (2019). aDN-GAL4 Control. <https://doi.org/10.7910/DVN/PKKXOE>
- Raudies, F., Brandon, M. P., Chapman, G. W., & Hasselmo, M. E. (2015). Head direction is coded more strongly than movement direction in a population of entorhinal neurons. *Brain Research*, 1621, 355–367.
- Rempe, D., Birdal, T., Hertzmann, A., Yang, J., Sridhar, S., & Guibas, L. J. (2021). Humor: 3d human motion model for robust pose estimation. In *International conference on computer vision (ICCV)*
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, 404, 132306. <https://doi.org/10.1016/j.physd.2019.132306>, www.sciencedirect.com/science/article/pii/S0167278919305974

- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data* 6(60), <https://doi.org/10.1186/s40537-019-0197-0>
- Shuai, H., Wu, L., & Liu, Q. (2022). Adaptive multi-view and temporal fusing transformer for 3d human pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://doi.org/10.1109/TPAMI.2022.3188716>
- Theriault, D. H., Fuller, N. W., Jackson, B. E., Bluhm, E., Evangelista, D., Wu, Z., Betke, M., & Hedrick, T. L. (2014). A protocol and calibration method for accurate multi-camera field videography. *The Journal of Experimental Biology* 217, 1843–1848, open access online, <http://jeb.biologists.org/content/early/2014/02/20/jeb.100529.abstract.html?papetoc>
- Tome, D., Russell, C., & Agapito, L. (2017). Lifting from the deep: Convolutional 3d pose estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 2500–2509)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*. pp. 5998–6008
- Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology
- Wu, Z., Kunz, T. H., & Betke, M. (2011). Efficient track linking methods for track graphs using network-flow and set-cover techniques. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, (pp. 1185–1192). Colorado Springs, <http://www.cs.bu.edu/fac/betke/papers/WuKunzBetke-CVPR2011.pdf>
- Wu, Z., & Betke, M. (2016). Global optimization for coupled detection and data association in multiple object tracking. *Computer Vision and Image Understanding*, 143, 25–37.
- Yuan, Y., Wei, S. E., Simon, T., Kitani, K., & Saragih, J. (2021). Sim-PoE: Simulated character control for 3D human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. (pp. 7159–7169)
- Zhang, L., Dunn, T., Marshall, J., Olveczky, B., & Linderman, S. (2021). Animal pose estimation from video data with a hierarchical von Mises-Fisher-Gaussian model. In: Banerjee, A., Fukumizu, K. (eds.) *Proceedings of The 24th international conference on artificial intelligence and statistics proceedings of machine learning research*, (vol. 130, pp. 2800–2808.) PMLR, <https://proceedings.mlr.press/v130/zhang21h.html>
- Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., & Ding, Z. (2021). 3D human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, (pp. 11656–11665)
- Zuffi, S., Kanazawa, A., Jacobs, D. W. & Black, M. J. (2017). 3D Menagerie: Modeling the 3D shape and pose of animals. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, (pp. 5524–5532). IEEE Computer Society, Los Alamitos, CA, USA. <https://doi.org/10.1109/CVPR.2017.586>, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.586>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.