Self-Supervised Keypoint Discovery in Behavioral Videos
Supplementary Material

Jennifer J. Sun[1]* Serim Ryou[1]*† Roni H. Goldshmid[1] Brandon Weissbourd[1] John O. Dabiri[1]
David J. Anderson[1] Ann Kennedy[2] Yisong Yue[1,3] Pietro Perona[1]

[1]Caltech [2]Northwestern University [3]Argo AI

Code & Project Website: https://sites.google.com/view/b-kind

We present additional experimental results (Section 1), additional implementation details (Section 2), and visualizations (Section 3). Additional video visualizations and code are available in the project website: https://sites.google.com/view/b-kind.

Benefits and risks of this technology. Automating the analysis of behavior is useful across many fields: in neuroscience, to study the neural control of behavior; in ethology and conservation, to study animal behavior and their response to human encroachment; in rehabilitation, to track patients' recovery of motor function; and in helping improve safety in the workplace. Risks are inherent in any application where humans behavior is analyzed, and care must be taken to respect privacy and human rights. Responsible use in research requires following all applicable rules and policies, including filing for permission with the relevant internal review board (IRB), and obtaining written informed consent from human subjects being filmed.

1. Additional Experimental Results

1.1. CalMS21 Ablation Study

Similar to the main paper, we evaluate CalMS21 on the behavior classification train/test split described in task 1 [14], and show results using Mean Average Precision (MAP) across the annotated behavior classes. We use B-KinD keypoints as input to behavior classification to compare against supervised and other self-supervised baselines.

Effect of Hyperparameters. For all experiments on CalMS21, we use a frame gap of 6 with 10 discovered keypoints. Here, we vary the number of discovered keypoints and frame gap for our model, and apply the learned keypoints to behavior classification (Table 1). There are small variations in performance, in particular, the downstream performance generally improves with increasing the number of keypoints, and a frame gap of 6 or 12 works better than larger frame gaps. We note that the number of low confidence background keypoints also increases with the number of discovered keypoints (Figure 1), and due to the large proportion of background keypoints, we do not use background keypoints in the 20 keypoints case for

Hyperparam.	Value	MAP	Hyperparam.	Value	MAP
Frame Gap	6	.852 ± .013	# keypoints	6	.850 ± .017
	12	.862 ± .012		10	.852 ± .013
	30	.839 ± .003		20*	.868 ± .008

Table 1. **Effect of Hyperparameters on CalMS21**. For frame gap experiments, the number of keypoints is set to 10. For experiments with varying number of keypoints, frame gap is set to 6. All keypoints, confidence, and covariance are used as inputs, except (*) for the experiments with 20 keypoints, where only high-confidence keypoints are used (11 keypoints) since a high proportion of keypoints are discovered on the background. Mean and standard dev from 5 classifier runs are shown.

*Equal contribution. Correspondence to jjsun@caltech.edu.

†Current affiliation: Samsung Advanced Institute of Technology

# Training Pairs	Corresponding Video Length (30Hz)	MAP
7.8k	4.3 min	.867 ± .003
18k	10 min	.840 ± .016
26k	14 min	.852 ± .013

Table 2. **Effect of Varying Training Data Amount for Keypoint Discovery**. We train the keypoint discovery model with different amounts of input training frame pairs from video. Different training amounts are selected by choosing random video subsets from the full set of CalMS21 training videos. Image pairs are sampled from videos with a gap of 6 frames, and between pairs to ensure no overlaps, there is a gap of 7 frames. All keypoints, confidence, and covariance values on 10 discovered keypoints are used. Mean and standard dev from 5 classifier runs are shown.

the classification task. In all cases, we note that we do better than other self-supervised baselines even with bounding box information (MAP = .819) for this task.

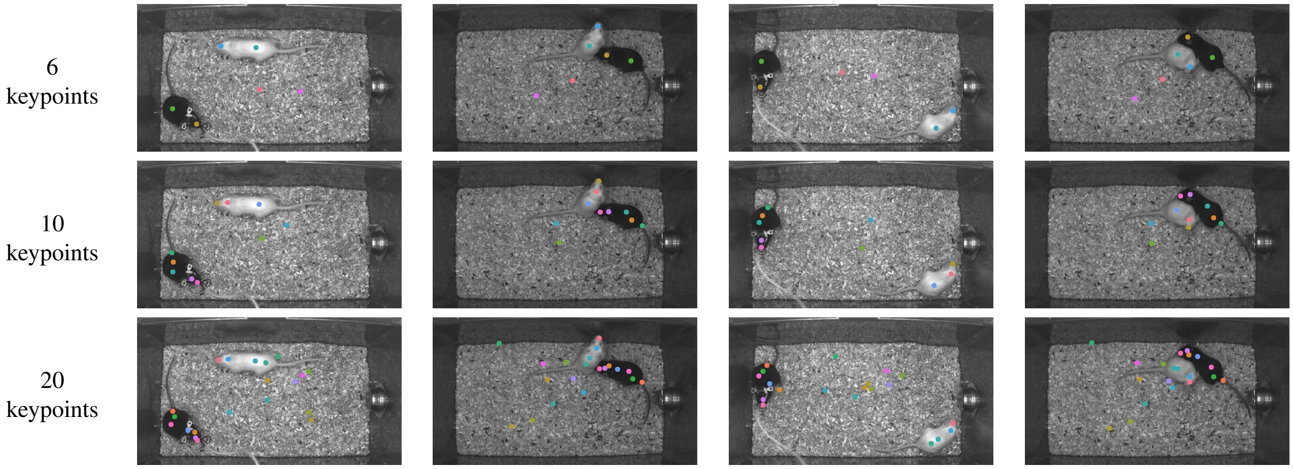

Figure 1. **Qualitative Results on CalMS21 by varying the number of keypoints**. We train the keypoint discovery model with different numbers of discovered keypoints. Each row shows qualitative results with all the keypoints including the background keypoints. We note that there are 2 background (low-confidence) keypoints for 6 and 10 discovered keypoints, and 9 background keypoints for 20 discovered keypoints.

Varying Amount of Unlabeled Video Data. We vary the amount of input data (unlabelled image pairs) used to train B-KinD, and observe comparable performance at different amounts of data availability (Table 2). In particular, we are able to achieve comparable performance on behavior classification to supervised keypoints (Table 4) by using only $7.8k$ input training pairs in our model (approximately 4 minutes of video recorded at 30Hz; approximately 30 minutes of video considering no overlaps on selected image pairs). We note that this experiment is varying the amount of unlabelled data for training the keypoint discovery model, while the train/test split for evaluating the behavior classifier stays the same.

Loss Ablation Study. We compare B-KinD trained with the full objective (reconstruction, rotation equivariance, separation) to one trained only on spatiotemporal difference reconstruction (Table 3). The rotation equivariance loss is qualitatively important for tracking semantically consistent parts of the mouse (Figure 2) and the separation loss prevents the model from predicting keypoints at the center of the image, which are rotationally consistent but do not track semantic body parts. The full objective is important to achieving comparable performance to supervised baselines. We would like to note that the image reconstruction baselines in our main results are also trained with the full objective, except the reconstruction is based on image reconstruction. Additionally, since keypoint locations are not consistent for spatiotemporal difference reconstruction only, we note that adding confidence and covariance significantly improves the performance of the reconstruction loss only model (Table 3).

Single Geometry Branch. Our proposed model extracts appearance features from the frame at time t, and two geometry features (the keypoints), where one is for frame at time t and the other is for frame at time $t + k$. It is also possible to train

CalMS21	Pose	Conf	Cov	Ours (MAP)	Reconstruction (MAP)
	✓			$.814 \pm .007$	$.695 \pm .022$
Loss Variation	✓	✓		$.857 \pm .005$	$.776 \pm .012$
	✓	✓	✓	$.852 \pm .013$	$.794 \pm .008$

Table 3. **Loss Variations on CalMS21**. "Ours" represents training B-KinD keypoints with the full objective (reconstruction, rotation equivariance, separation) and "Reconstruction" indicates training with spatiotemporal difference reconstruction only. Mean and standard dev from 5 classifier runs are shown.

the model using only one geometry branch only for the frame at time $t + k$, without the geometry branch for time t. On CalMS21, training B-KinD with one geometry branch reduced the classification performance from MAP of 0.852 ± 0.013 (full model) to 0.835 ± 0.013 (single branch).

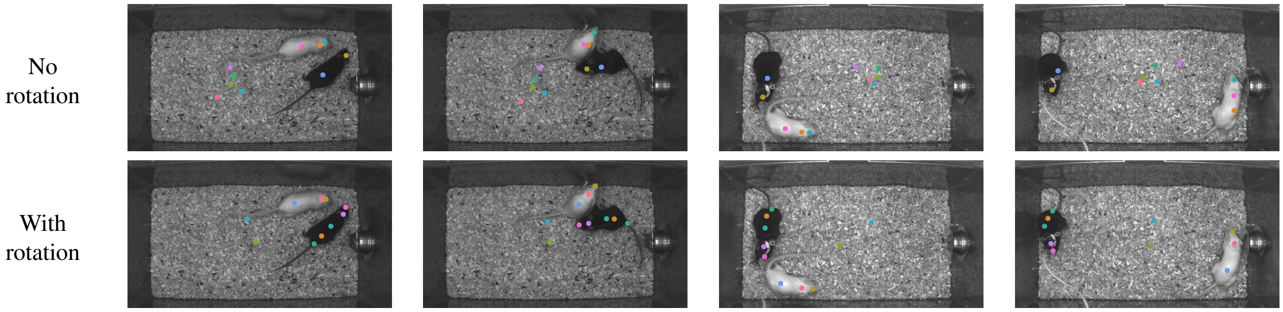

Figure 2. **Qualitative Results on CalMS21 for loss ablation study**. With the full training objective for our discovered keypoints, we are able to track 8/10 keypoints consistently, while without rotation loss, there are only 5/10 tracked keypoints on both mice. Additionally, some of the discovered keypoints without rotation are not semantically consistent (for example, the pink and orange keypoints, two keypoints on the body of the white mouse, shift in order as the white mouse moves around). See quantitative results in Table 3.

1.2. CalMS21 Per-Class Performance

B-KinD keypoints achieve comparable performance to supervised keypoints when using pose and confidence features from the heatmap (Table 4). For both supervised keypoints and our keypoints, the behavior classes with the biggest improvement when adding confidence features is on the "Attack" class, which contains frames with occlusion and motion blur since the mice are moving quickly and chasing/tussling. Heatmap confidence and covariance values provides more information about the detected part (Figure 10). For example, when a part is well localized (ex: visible nose of mouse), our keypoint discovery network produces a heatmap with a single high peak with low variance; conversely, when a target part is occluded, the heatmap contains a blurred shape with lower peak value. We note that performance is similar for the supervised keypoints and our keypoints on the "Investigation" and "Mount" classes.

1.3. Human3.6M Ablation Study

We evaluate the effect of number of keypoints and frame gaps on simplified Human 3.6M (Table 5). Note that we use frame difference, instead of SSIM, as a reconstruction target for studying the effect of hyperparameters. When the frame gap is too small, the region of motion becomes too narrow, which results in slightly lower performance. Also, discovering more keypoints does not always guarantee better performance. Empirical results show that informative keypoints are discoverable with 16 keypoints.

We also perform an ablation study using a single geometry branch at time $t + k$ for two different reconstruction targets (image and SSIM), using the same %-MSE error metric as the main paper. Training with one geometry branch reduced the pose regression performance from 2.534 ± 0.056 to 2.596 ± 0.1089 (image) and 2.556 ± 0.0320 (SSIM) where the standard deviation is computed over 5 runs. As a loss ablation study, we train our model without the rotation equivariance loss on simplified Human 3.6M, and the pose regression performance is reduced to 2.61.

CalMS21	Pose	Conf	Cov	MAP	Attack AP	Investigation AP	Mount AP
				Fully supervised			
MARS † [13]	✓			.856 ± .010	.724 ± .023	.893 ± .005	.950 ± .004
	✓	✓		.874 ± .003	.790 ± .004	.890 ± .006	.943 ± .004
	✓	✓	✓	.880 ± .005	.804 ± .012	.902 ± .004	.934 ± .006
				Self-supervised			
Jakab et al. [8]	✓			.186 ± .008	.135 ± .019	.254 ± .019	.170 ± .029
Image Recon.	✓			.182 ± .007	.111 ± .016	.217 ± .011	.219 ± .021
	✓	✓		.184 ± .006	.114 ± .006	.209 ± .012	.229 ± .021
	✓	✓	✓	.165 ± .012	.110 ± .016	.218 ± .013	.167 ± .038
Image Recon. bbox†	✓			.819 ± .008	.680 ± .028	.861 ± .007	.918 ± .007
	✓	✓		.812 ± .006	.694 ± .011	.818 ± .016	.923 ± .013
	✓	✓	✓	.812 ± .010	.709 ± .008	.806 ± .019	.922 ± .013
Ours	✓			.814 ± .007	.654 ± .025	.861 ± .003	.925 ± .014
	✓	✓		.857 ± .005	.763 ± .015	.879 ± .009	.928 ± .006
	✓	✓	✓	.852 ± .013	.751 ± .025	.870 ± .009	.935 ± .010

Table 4. **Per-Class Behavior Classification Results on CalMS21**. "Ours" represents classifiers using input keypoints from B-KinD. "conf" represents using the confidence score, and "cov" represents values from the covariance matrix of the heatmap. † refers to models that require bounding box inputs before keypoint estimation. Mean and standard dev from 5 classifier runs are shown.

Hyperparam.	Value	%-MSE	Hyperparam.	Value	%-MSE
Frame Gap	10	2.81	# keypoints	10	2.96
	20	**2.57**		16	**2.57**
	30	2.64		30	2.63

Table 5. **Hyperparameters Study on Simplified Human 3.6M**. %-MSE error from a single run is shown. For frame gap experiments, the number of keypoints is set to 16. Frame gap is set to 20 for experiments with a varying number of keypoints. We use frame difference here as a reconstruction target for studying the effect of hyperparameters.

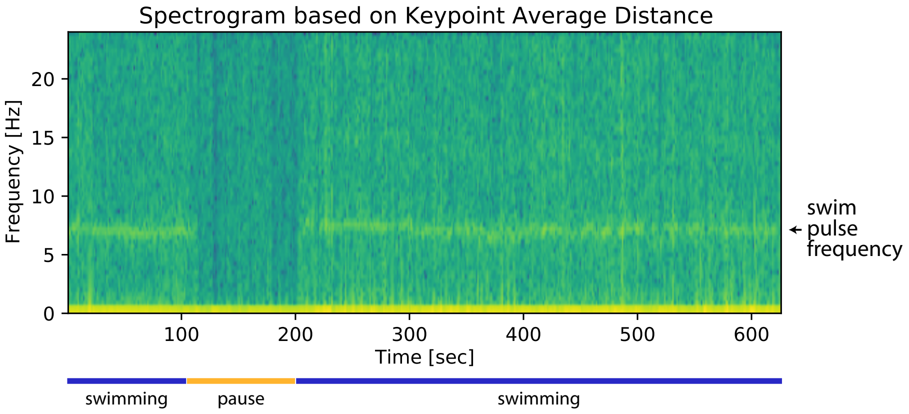

Figure 3. **Spectrogram from Distance of Discovered Keypoints**. From a recorded video of jellyfish swimming at 48Hz, we discover keypoints at each frame using our model and compute a spectrogram based on the average distance between discovered keypoints on the jellyfish.

1.4. Jellyfish Pulse Detection

The energy efficiency of swimming jellyfish combined with their structural simplicity makes them a good organism for understanding the hydrodynamics of animal propulsion [4]. In particular, researchers would like to study the relationship between body plan and swim pulse frequency across jellyfish species. This has applications in ethology, hydrodynamics, as

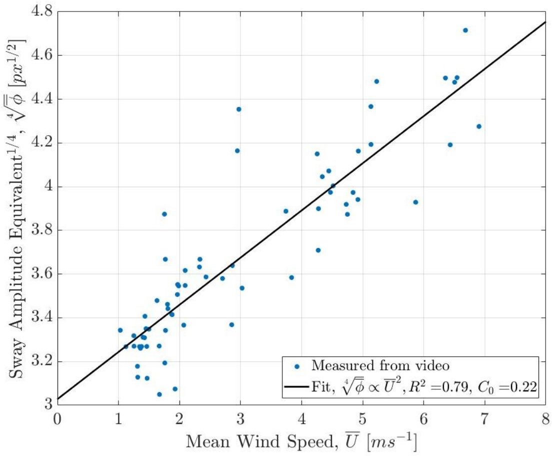

Figure 4. **Wind Speed Regression from Discovered Keypoints**. Mean wind speed, $\bar{U}$, vs. the fourth root of the sway amplitude equivalent measured from the standard deviation of the convex hull area of the 15 discovered keypoints in each clip, based on model from [2]. The scatter represents 10-mintute averages of the same data used for training the keypoint model. The black lines represent the best linear regression fit for the proportionality assumption. The proportionality coefficient and the R^2 values are presented in the legend.

well as bio-inspired vehicles. Here, we use Clytia hemisphaerica as our jellyfish species to study jellyfish pulsing during swimming using our discovered keypoints. After videos are recorded from a swimming jellyfish from in a tank, we apply our keypoint discovery model to track keypoints automatically on the jellyfish (visualization provided in project website). We also compute the swim pulse frequency by computing the distance between all pairs of our discovered keypoints with high confidence (5 keypoints) and extracting a frequency spectrogram based on average keypoint distance (Figure 3). We observe a visible band at the swimming frequency around 7Hz, and we note that between 110 to 200 seconds, the jellyfish is not swimming (floating), and thus the swimming frequency band is not visible in that duration. Since our discovered keypoints are able to detect pulsing, this provides a way to automatically annotate swimming behavior. This method can be applied to videos from other jellyfish species to study the relationship between swimming dynamics and body plan.

1.5. Vegetations Wind Speed Regression

Videos of oscillation of tree branches and leaves encode information on local wind conditions, and could function as wind speed sensors. Local wind speed measurements are useful for a variety of tasks, including air pollution monitoring, weather forecasting, and predicting movement of forest fires [1, 2]. We use the Vegetation dataset to study the effectiveness of our discovered keypoints for capturing oscillating movement of trees. This dataset consists of videos of swaying trees recorded from an overhead camera from a drone, while the wind speed is measured using an anemometer. We observe that the discovered keypoints from our approach are of different parts of the tree in separate views but are consistent within a single clip, as to capture oscillations of branches/leaves (visualization provided in project website).

We use a physics-based model [2] to study the relationship between oscillations of trees and wind speed. This model defines the relationship between structural oscillation and wind speed as:

$$\sigma \sim I_u \bar{U}$$

where σ is the standard deviation of the amplitude of the structural oscillations, $\bar{U}$ is the mean wind speed, and I_u is the measure of the turbulence intensity of the streamwise component, defined as the standard deviation of the streamwise velocity fluctuations normalized to the mean wind speed. The model requires tracing of the structural oscillations of the branches/leaves, which was previously done manually and we show that the keypoint discovery model can do this automatically. The 15 detected keypoints track these oscillations in a 2D space and a representative measure of these oscillations in both coordinates is calculated using the convex hull area, or the sway amplitude equivalent, ϕ. The average sway amplitude

Table 6. **Architecture details of the reconstruction decoder.** "Conv_block" refers to a basic convolution block which is composed of 3×3 convolution, batch normalization, and ReLU activation. Note that output size for Human3.6M experiments is downsampled by a factor of 2 for all the layers.

Type	Input dimension	Output dimension	Output size
Upsampling	-	-	16x16
Conv_block	2048 + # keypoints × 2	1024	16x16
Upsampling	-	-	32x32
Conv_block	1024 + # keypoints × 2	512	32x32
Upsampling	-	-	64x64
Conv_block	512 + # keypoints × 2	256	64x64
Upsampling	-	-	128x128
Conv_block	256 + # keypoints × 2	128	128x128
Upsampling	-	-	256x256
Conv_block	128 + # keypoints × 2	64	256x256
Convolution	64	3	256x256

equivalent of the keypoints, $\bar{\phi}$, provides the following proportionality relationship:

$$C_0\sqrt{\bar{\phi}} \sim \bar{U}$$

where C_0 is the coefficient of proportionality. The best regression fit of the experimental data calculated using the least squares method has $R^2 = 0.79$ suggesting there is a good agreement between the proportionality assumption and the experimental results using the keypoint detection model (Figure 4).

2. Additional Implementation Details

Architecture Details Our method uses ResNet-50 [6] as an encoder Φ, GlobalNet [3] as a pose decoder Ψ, and a series of convolution blocks as a reconstruction decoder ψ, following the unsupervised keypoint discovery model from [11]. Architecture details about reconstruction decoder is shown in Table 6. For more implementation details, the code is available on our project website: `https://sites.google.com/view/b-kind`.

The hyperparameters for the keypoint discovery model is included in Table 7. All models use SSIM image as the reconstruction target, unless stated otherwise. All keypoint discovery models are trained until convergence of the training loss on a NVIDIA V100 Tensor Core GPU. Below, we include a additional details on the keypoint discovery model and downstream task used to evaluate each dataset.

CalMS21. The CalMS21 dataset [14] consists of videos and trajectory data from a pair of interacting mice, annotated with behavior labels at each frame by neuroscientists. There is one black mouse and one white mouse engaging in social behaviors, recorded at 1024 × 570 at 30 Hz. The supervised keypoints provided with CalMS21 are from the MARS detector [13] developed for this dataset, which detects 7 anatomically-defined keypoints for each mouse. For training keypoint discovery, we use a subset of the training split without miniscope cable (26k images), and we use the full train/test split defined by [14] on Task 1 for evaluating behavior classification. For behavior classification, we use the same setup (1D Conv Net architecture, hyperparameters, random seeds, data split, etc.) as the CalMS21 dataset benchmarks, except we replace the supervised input keypoints with our discovered keypoints for evaluation. We additionally experiment with adding heatmap confidence and convariance during classification by appending these additional features to input keypoints during classifier training. This dataset is available under the CC-BY-NC-SA license.

MARS-Pose. MARS-Pose is a set of mouse interaction images with human keypoint annotations [13] and these images are recorded in similar recording conditions to CalMS21 [14]. We use a subset of the images for training (10,50,100,500) and test on the full 1.5k images test set. We evaluate this dataset based on pose estimation performance to the human-annotated keypoints. For the supervised model, we use the stacked hourglass model [10] and for the semi-supervised model, we add a supervised keypoint estimation loss based on MSE to our keypoint discovery framework.

Dataset	# Keypoints	Batch size	Resolution	Frame Gap	Learning Rate
CalMS21	10	5	256	6	0.001
Fly	10	5	256	3	0.001
Human	16	36	128	20	0.001
Jellyfish	10	5	256	20	0.001
Vegetations	15	5	256	60	0.001

Table 7. **Hyperparameters for Keypoint Discovery.**

Fly vs. Fly. This dataset consists of videos of two interacting flies [5] with frame-level behavior annotations. We use the "Aggression" videos from this dataset (144 × 144 at 30 Hz) and use the behaviors with more than 1000 annotated training samples, with the same setup as [15]. The provided FlyTracker with this dataset computes hand-crafted behavioral features directly from video for behavior classification. Since keypoints may be discovered from any body part, we compute corresponding generic features not based on keypoint identity: speed of every keypoint, acceleration of every keypoint, distance between every pair, and angle between every triplet. Additionally, since the flies are similar in appearance, when extracting keypoint locations from the B-KinD heatmaps, we detect 2 max locations for the 2 peaks. We then take the spatial softmax over the region around each max location, instead of taking the spatial softmax over the whole heatmap. In terms of identity, we always use the fly with smaller y values at centroid as the first fly, and the fly with larger y values as the second. For the classifier model, we use the same setup (1D Conv Net architecture (except frame gap in the Conv Net is 1 instead of 2 since flies have faster behaviors), hyperparameters, random seeds, data split, etc.) as the CalMS21 dataset benchmarks, except using the fly features as input to classify annotated behavior at each frame. This dataset is available under the CC0 1.0 Universal license.

Human3.6M. Human 3.6M dataset [7] is a large-scale dataset containing 3.6 million 3D and 2D human poses with corresponding images. The videos are taken from 4 different viewpoints for 17 scenarios (discussion, taking photo, walking, ...) with the same background. This dataset is available for academic use, and the dataset license is provided by the Human 3.6M authors on the dataset website, link available within [7]. Simplified Human 3.6M dataset, introduced by [17], consists of 6 different activities with mostly upright poses by cropping the full image using bounding box. Since our method requires static background assumption, we crop a pair of full images using the same bounding box for training a keypoint discovery model. The final image has 128×128 resolution. We evaluate the pose regression performance on the same testing set from the Simplified Human 3.6M dataset.

Jellyfish. This is an in-house video dataset consisting of a freely swimming Clytia hemisphaerica in a water tank. We train and run our keypoint discovery model on the same 30k frames, recorded at 48Hz, to demonstrate our keypoints on new organisms and on detecting swimming frequency. Since the jellyfish is very small ($\sim$ 50 pix) relative to the size of the image (928 × 1158), we first use the SSIM image to identify a rough bounding box around the jellyfish (150 × 150) before re-scaling the input to the keypoint discovery model to 256 × 256. We note that this step would not be necessary given a GPU with more memory, since the jellyfish would still be visible at higher resolutions. More details on the pulse detection is in Section 1.4.

Vegetations. This is an in-house video dataset captured from a drone flying overhead of an Oak tree as the tree is swaying in the wind, and local wind speed is recorded using an anemometer. The video frames are processed at 512 × 512 and 120 Hz, and re-scaled to be 256 × 256 for the keypoint discovery model. The drone may shift slightly over the video recording, and we use existing image alignment methods [16] to align video frames before computing the spatiotemporal difference reconstruction target for our method. More details on the wind speed regression is in Section 1.5.

3. Visualizations

We present additional visualization results on mouse (Figure 5), fly (Figure 6), tree (Figure 7), and human (Figure 8). Additional videos are available on our project website: `https://sites.google.com/view/b-kind`.

Confidence Visualizations. We observe that keypoints discovered on the background and not tracking agent parts generally have very low confidence (Figure 10). This is because heatmaps of background keypoints are not well-localized, and is spread over the image, thus have a low peak value (low confidence). In comparison, discovered keypoints on body parts (such as the nose), is localized to a specific part of the image and has higher peak values. Additionally, confidence values can provide information on occluded parts. For example, for the nose of the white mouse (third column, first row, Figure 10), the confidence varies from $0.5 \sim 0.6$ when the nose is visible in the first two examples to $0.3 \sim 0.4$ when the nose is harder to see in the last two examples.

Challenges. Difficult examples for our model are visualized in Figure 9. When there is occlusion, such as in the mouse examples, the keypoint is generally discovered on the visible parts, and when there is heavy occlusion, such as from the miniscope cable, discovered keypoint location may be shifted. This is likely why including additional information from the heatmap, such as confidence (Figure 10) is helpful for behavior classification. We can see similar effects on self-occlusion for humans, and also left-right swapping of some keypoints for when humans are facing towards or away from the camera (this has also been observed with other keypoint discovery models [9, 12, 17]). Unusual poses may also be difficult, such as when the fly is completely tilted towards the camera in the last column of row 1. Future directions to integrate 3D structure, for instance by using multi-view videos, could help address these issues. Despite this, we note that our current discovery model achieves state-of-the-art results among other self-supervised methods for behavior classification and keypoint regression.

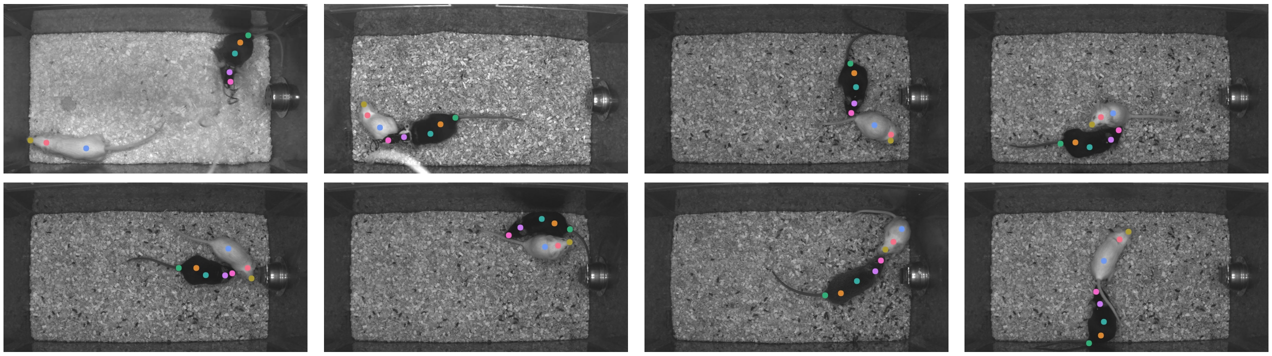

Figure 5. **Qualitative Results on CalMS21**. We observe that keypoints are discovered for noses of both mice and generally along the spine of the mice.

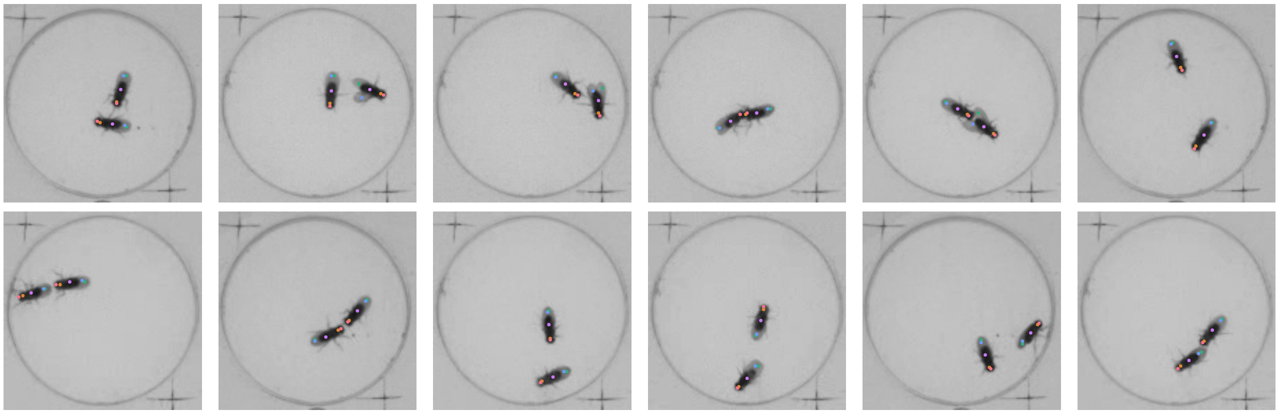

Figure 6. **Qualitative Results on Fly-vs-Fly**. We observe that 3 keypoints are discovered on the body of the fly, with 2 on the wings (one for each wing).

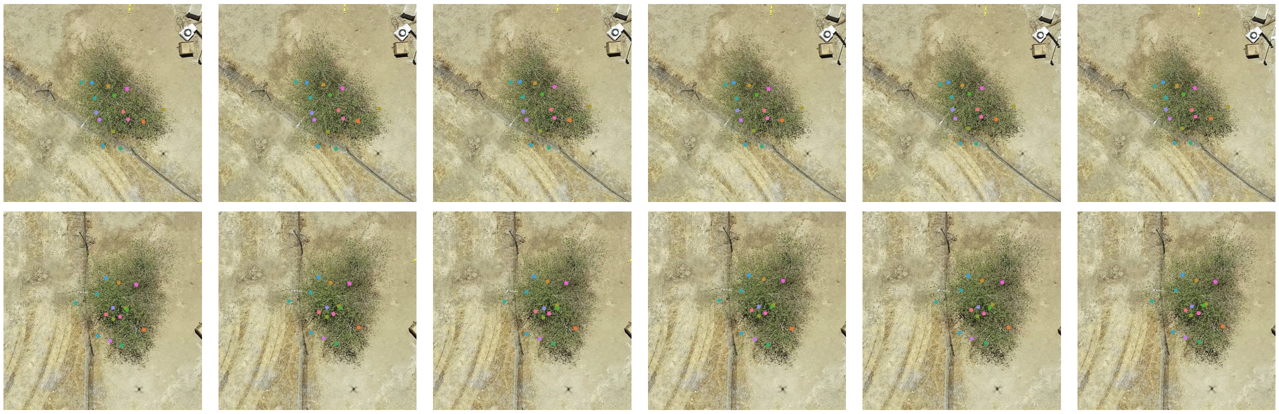

Figure 7. **Qualitative Results on Vegetations**. Each row shows different frames with discovered keypoints from a single video. Our model can discover and track consistent keypoints within the same video.

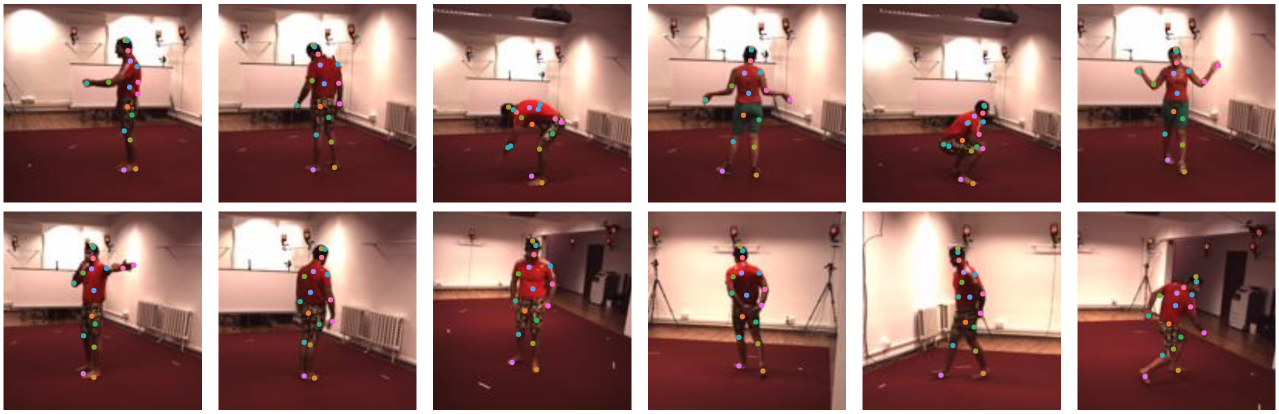

Figure 8. **Qualitative Results on Simplified Human 3.6M**. We observe that keypoints are generally discovered on visible joints and end points of humans, such as head, elbows, hands, upper legs, knees and feet. We note that there is left/right swapping of body parts, since when the human is facing forwards or backwards, keypoints are generally on the same side.

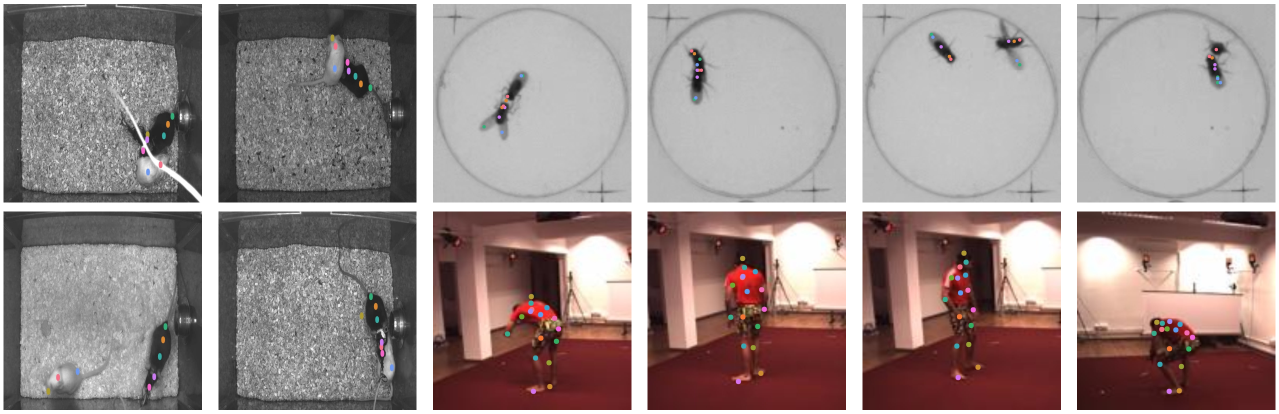

Figure 9. **Limitations**. We visualize examples that are difficult for our model, for example from occlusion/agents being in close proximity (mouse, fly), self-occlusion (human), unusual poses (human, fly), and left-right swapping (human).

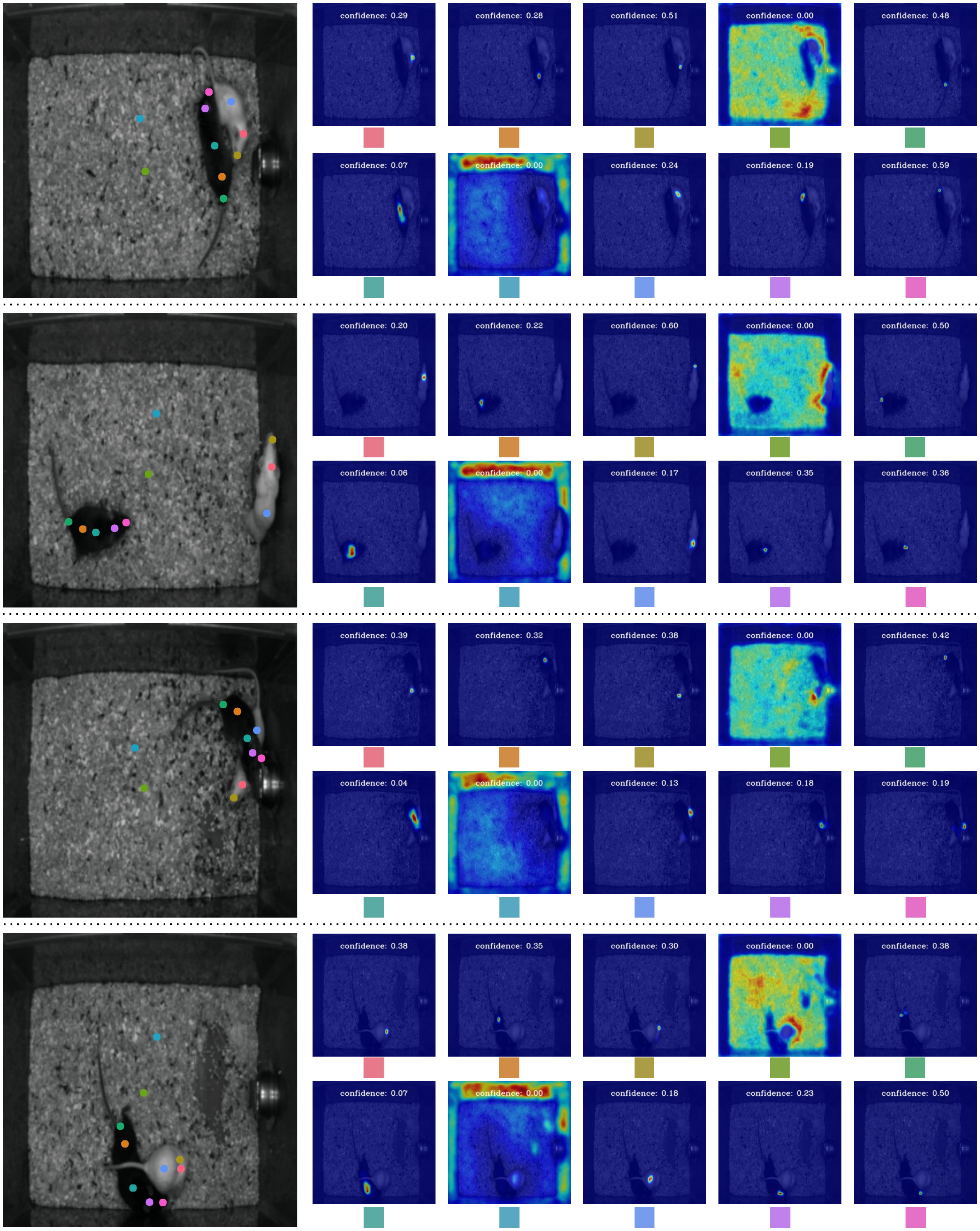

Figure 10. **Confidence visualization on CalMS21**. Confidence score (maximum prediction value) is shown with the normalized heatmap. Background keypoints (fourth on row 1 and second on row 2) have very low confidence.

References

[1] Jennifer Cardona, Michael Howland, and John Dabiri. Seeing the wind: Visual wind speed prediction with a coupled convolutional and recurrent neural network. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 5

[2] Jennifer L Cardona and John O Dabiri. Wind speed inference from environmental flow-structure interactions, part 2: leveraging unsteady kinematics. *arXiv preprint arXiv:2107.09784*, 2021. 5

[3] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 6

[4] John H Costello, Sean P Colin, John O Dabiri, Brad J Gemmell, Kelsey N Lucas, and Kelly R Sutherland. The hydrodynamics of jellyfish swimming. *Annual Review of Marine Science*, 13:375–396, 2021. 4

[5] Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. Detecting social actions of fruit flies. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014. 7

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016. 6

[7] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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, 2013. 7

[8] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 4

[9] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, 2019. 8

[10] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016. 6

[11] Serim Ryou and Pietro Perona. Weakly supervised keypoint discovery. *CoRR*, abs/2109.13423, 2021. 6

[12] Luca Schmidtke, Athanasios Vlontzos, Simon Ellershaw, Anna Lukens, Tomoki Arichi, and Bernhard Kainz. Unsupervised human pose estimation through transforming shape templates. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2484–2494. Computer Vision Foundation / IEEE, 2021. 8

[13] Cristina Segalin, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J. Sun, Pietro Perona, David J. Anderson, and Ann Kennedy. The mouse action recognition system (mars): a software pipeline for automated analysis of social behaviors in mice. *bioRxiv https://doi.org/10.1101/2020.07.26.222299*, 2020. 4, 6

[14] Jennifer J Sun, Tomomi Karigo, Dipam Chakraborty, Sharada P Mohanty, David J Anderson, Pietro Perona, Yisong Yue, and Ann Kennedy. The multi-agent behavior dataset: Mouse dyadic social interactions. *arXiv preprint arXiv:2104.02710*, 2021. 1, 6

[15] Jennifer J Sun, Ann Kennedy, Eric Zhan, David J Anderson, Yisong Yue, and Pietro Perona. Task programming: Learning data efficient behavior representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2876–2885, 2021. 7

[16] Philippe Thévenaz. Stackreg: an imagej plugin for the recursive alignment of a stack of images. *Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne*, 2012, 1998. 7

[17] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018. 7, 8