Self-Supervised Keypoint Discovery in Behavioral Videos

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

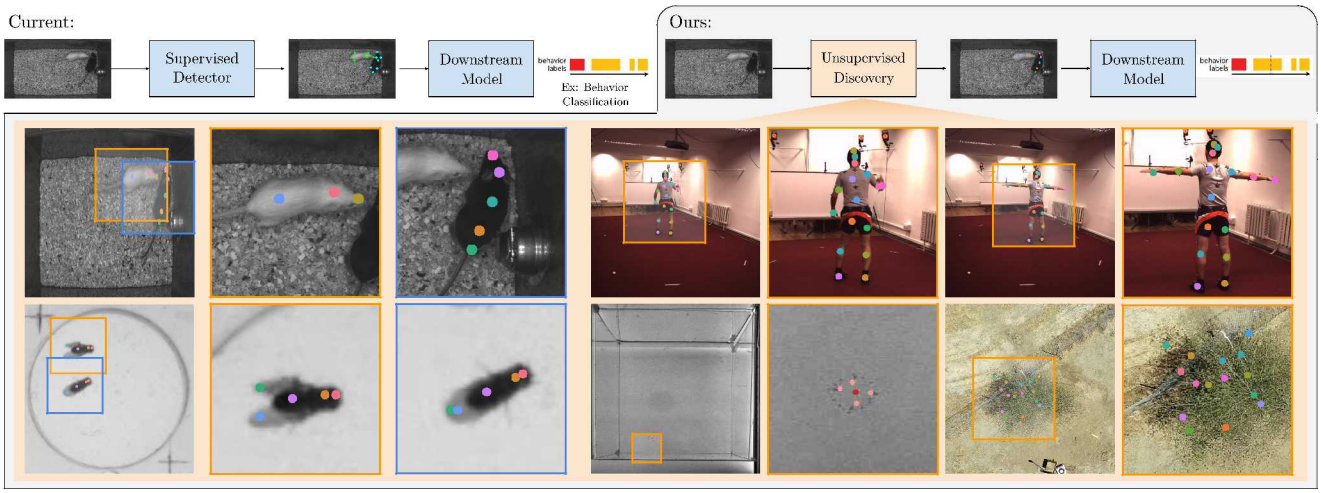

Figure 1. **Self-supervised Behavioral Keypoint Discovery**. Intermediate representations in the form of keypoints are frequently used for behavior analysis. We propose a method to discover keypoints from behavioral videos without the need for manual keypoint or bounding box annotations. Our method works across a range of organisms (including mice, humans, flies, jellyfish and tree), works with multiple agents simultaneously (see flies and mice above), does not require bounding boxes (boxes visualized above purely for identifying the enlarged regions of interest) and achieves state-of-the-art performance on downstream tasks.

Abstract

We propose a method for learning the posture and structure of agents from unlabelled behavioral videos. Starting from the observation that behaving agents are generally the main sources of movement in behavioral videos, our method, Behavioral Keypoint Discovery (B-KinD), uses an encoder-decoder architecture with a geometric bottleneck to reconstruct the spatiotemporal difference between video frames. By focusing only on regions of movement, our approach works directly on input videos without requiring manual annotations. Experiments on a variety of agent types (mouse, fly, human, jellyfish, and trees) demonstrate the generality of our approach and reveal that our discovered keypoints represent semantically meaningful body parts, which achieve state-of-the-art performance on keypoint regression among self-supervised methods. Additionally, B-KinD achieve comparable performance to supervised keypoints on downstream tasks, such as behavior classification, suggesting that our method can dramatically reduce model training costs vis-a-vis supervised methods.

1. Introduction

Automatic recognition of object structure, for example in the form of keypoints and skeletons, enables models to capture the essence of the geometry and movements of objects. Such structural representations are more invariant to background, lighting, and other nuisance variables and are much lower-dimensional than raw pixel values, making them good intermediates for downstream tasks, such as behavior classification [4, 11, 15, 40, 44], video alignment [27, 45], and physics-based modeling [7, 12].

However, obtaining annotations to train supervised pose detectors can be expensive, especially for applications in behavior analysis. For example, in behavioral neuroscience [35], datasets are typically small and lab-specific, and the training of a custom supervised keypoint detector

*Equal contribution. Correspondence to jjsun@caltech.edu.
†Current affiliation: Samsung Advanced Institute of Technology

presents a significant bottleneck in terms of cost and effort. Additionally, once trained, supervised detectors often do not generalize well to new agents with different structures without new supervision. The goal of our work is to enable keypoint discovery on new videos without manual supervision, in order to facilitate behavior analysis on novel settings and different agents.

Recent unsupervised/self-supervised methods have made great progress in keypoint discovery [21, 22, 53] (see also Section 2), but these methods are generally not designed for behavioral videos. In particular, existing methods do not address the case of multiple and/or non-centered agents, and often require inputs as cropped bounding boxes around the object of interest, which would require an additional detector module to run on real-world videos. Furthermore, these methods do not exploit relevant structural properties in behavioral videos (e.g., the camera and the background are typically stationary, as observed in many real-world behavioral datasets [5,15, 23, 30, 35, 40]).

To address these challenges, the key to our approach is to discover keypoints based on reconstructing the *spatiotemporal difference* between video frames. Inspired by previous works based on image reconstruction [21, 38], we use an encoder-decoder setup to encode input frames into a geometric bottleneck, and train the model for reconstruction. We then use spatiotemporal difference as a novel reconstruction target for keypoint discovery, instead of single image reconstruction. Our method enables the model to focus on discovering keypoints on the behaving agents, which are generally the only source of motion in behavioral videos.

Our self-supervised approach, **B**ehavioral **K**eypo**in**t **D**iscovery (B-KinD), works without manual supervision across diverse organisms (Figure 1). Results show that our discovered keypoints achieve state-of-the-art performance on downstream tasks among other self-supervised keypoint discovery methods. We demonstrate the performance of our keypoints on behavior classification [43], keypoint regression [21], and physics-based modeling [7]. Thus, our method has the potential for transformative impact in behavior analysis: first, one may discover keypoints from behavioral videos for new settings and organisms; second, unlike methods that predict behavior directly from video, our low-dimensional keypoints are semantically meaningful so that users can directly compute behavioral features; finally, our method can be applied to videos without the need for manual annotations.

To summarize, our main contributions are:

1. **Self-supervised method for discovering keypoints** from real-world behavioral videos, based on spatiotemporal difference reconstruction.
2. **Experiments across a range of organisms** (mice, flies, human, jellyfish, and tree) demonstrating the generality of the method and showing that the discovered keypoints are semantically meaningful.
3. **Quantitative benchmarking** on downstream behavior analysis tasks showing performance that is comparable to supervised keypoints.

2. Related work

Analyzing Behavioral Videos. Video data collected for behavioral experiments often consists of moving agents recorded from stationary cameras [1,4, 5, 11, 15,23, 34, 40]. These behavioral videos contain different model organisms studied by researchers, such as fruit flies [4, 11, 15, 25] and mice [5, 18, 23,40]. From these recorded video data, there has been an increasing effort to automatically estimate poses of agents and classify behavior [13,14,18,25,31,40].

Pose estimation models that were developed for behavioral videos [16, 31, 36,40] require human annotations of anatomically defined keypoints, which are expensive and time-consuming to obtain. In addition to the cost, not all data can be crowd-sourced due to the sensitive nature of some experiments. Furthermore, organisms that are translucent (jellyfish) or with complex shapes (tree) can be difficult for non-expert humans to annotate. Our goal is to enable keypoint discovery on videos for behavior analysis, without the need for manual annotations.

After pose estimation, behavior analysis models generally compute trajectory features and train behavior classifiers in a fully supervised fashion [5,15, 18,40, 44]. Some works have also explored using unsupervised methods to discover new motifs and behaviors [3, 19,29, 52]. Here, we apply our discovered keypoints to supervised behavior classification and compare against baseline models using supervised keypoints for this task.

Keypoint Estimation. Keypoint estimation models aim to localize a predefined set of keypoints from visual data, and many works in this area focus on human pose. With the success of fully convolutional neural networks [41], recent methods [8,33,46,51] employ encoder-decoder networks by predicting high-resolution outputs encoded with 2D Gaussian heatmaps representing each part. To improve model performance, [33,46,51] propose an iterative refinement approach, [8,37] design efficient learning signals, and [9,49] exploit multi-resolution information. Beyond human pose, there are also works that focus on animal pose estimation, notably [16,31,36]. Similar to these works, we also use 2D Gaussian heatmaps to represent parts as keypoints, but instead of using human-defined keypoints, we aim to discover keypoints from video data without manual supervision.

Unsupervised Part Discovery. Though keypoints provide a useful tool for behavior analysis, collecting annotations is time-consuming and labor-intensive especially for new domains that have not been previously studied. Unsupervised keypoint discovery [21, 22, 53] has been proposed to reduce keypoint annotation effort and there have been

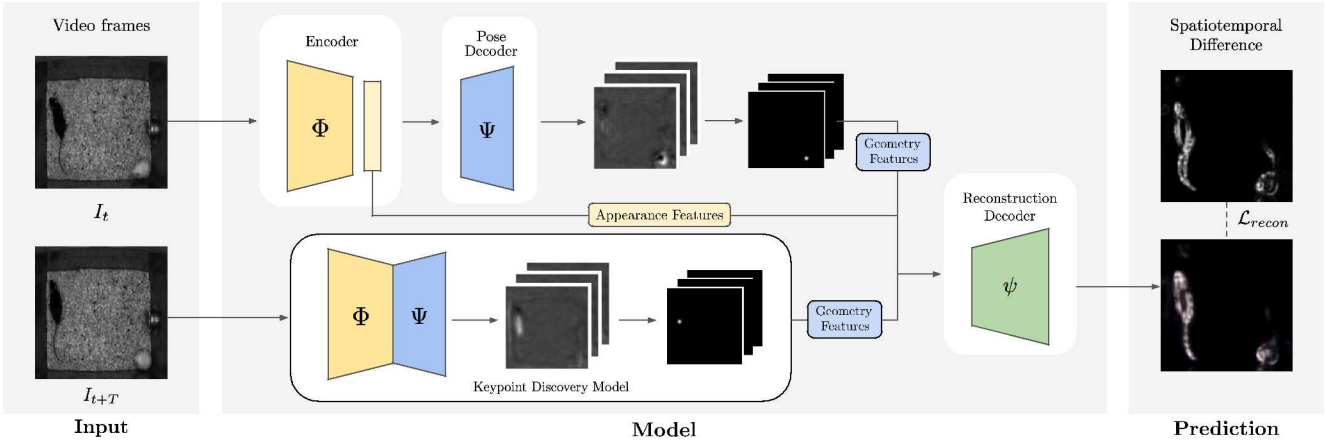

Figure 2. **B-KinD, an approach for keypoint discovery from spatiotemporal difference reconstruction**. I_t and I_{t+T} are video frames at time t and $t+T$. Both frame I_t and frame I_{t+T} are fed to an appearance encoder Φ and a pose decoder Ψ. Given the appearance feature from I_t and geometry features from both I_t and I_{t+T} (Sec 3.1), our model reconstructs the spatiotemporal difference (Sec 3.2.1) computed from two frames using the reconstruction decoder ψ.

many promising results on centered and/or aligned objects, such as facial images and humans with an upright pose. These methods train and evaluate on images where the object of interest is centered in an input bounding box. Most of the approaches [21, 28, 53] use an autoencoder-based architecture to disentangle the appearance and geometry representation for the image reconstruction task. Our setup is similar in that we also use an encoder-decoder architecture, but crucially, we reconstruct spatiotemporal difference between video frames, instead of the full image as in previous works. We found that this enables our discovered keypoints to track semantically-consistent parts without manual supervision, requiring neither keypoints nor bounding boxes.

There are also works for parts discovery that employ other types of supervision [22, 38, 39]. For example, [38] proposed a weakly-supervised approach using class label to discriminate parts to handle viewpoint changes, [22] incorporated pose prior obtained from unpaired data from different datasets in the same domain, and [39] proposed a template-based geometry bottleneck based on a pre-defined 2D Gaussian-shaped template. Different from these approaches, our method does not require any supervision beyond the behavioral videos. We chose to focus on this setting since other supervisory sources are not readily available for emerging domains (ex: jellyfish, trees).

In previous works, keypoint discovery has been applied to downstream tasks, such as image and video generation [22, 32], keypoint regression to human-annotated poses [21, 53], and video-level action recognition [26, 32]. While we also apply keypoint discovery to downstream tasks, we note that our work differs in approach (we discover keypoints directly on behavioral videos using spatiotemporal difference reconstruction), focus (behavioral videos of diverse organisms), and application (real-world behavior analysis tasks [7, 43]).

3. Method

The goal of B-KinD (Figure 2) is to discover semantically meaningful keypoints in behavioral videos of diverse organisms without manual supervision. We use an encoder-decoder setup similar to previous methods [21, 38], but instead of image reconstruction, here we study a novel reconstruction target based on spatiotemporal difference. In behavioral videos, the camera is generally fixed with respect to the world, such that the background is largely stationary and the agents (e.g. mice moving in an enclosure) are the only source of motion. Thus spatiotemporal differences provide a strong cue to infer location and movements of agents.

3.1. Self-supervised keypoint discovery

Given a behavioral video, our work aims to reconstruct regions of motion between a reference frame I_t (the video frame at time t) and a future frame I_{t+T} (the video frame T timesteps later, for some set value of T.) We accomplish this by extracting appearance features from frame I_t and keypoint locations ("geometry features") from both frames I_t and I_{t+T} (Figure 2). In contrast, previous works [21, 22, 28, 38, 39] use appearance features from I_t and geometry features from I_{t+T} to reconstruct the full image I_{t+T} (instead of difference between I_t and I_{t+T}).

We use an encoder-decoder architecture, with shared appearance encoder , geometry decoder Ψ, and reconstruction decoder ψ. During training, the pair of frames I_t and I_{t+T} are fed to the appearance encoder to generate appearance features, and those features are then fed into the geometry decoder Ψ to generate geometry features. In our approach, the reference frame I_t is used to generate both appearance and geometry representations, and the future

frame I_{t+T} is only used to generate a geometry representation. The appearance feature h_a^t for frame I_t are defined simply as the output of Φ: $h_a^t = \Phi(I_t)$.

The pose decoder Ψ outputs K raw heatmaps $X_i \in \mathbb{R}^2$, then applies a spatial softmax operation on each heatmap channel. Given the extracted $p_i = (u_i, v_i)$ locations for $i = \{1, \ldots, K\}$ keypoints from the spatial softmax, we define the geometry features h_g^t to be a concatenation of 2D Gaussians centered at (u_i, v_i) with variance σ.

Finally, the concatenation of the appearance feature h_a^t and the geometry features h_g^t and h_g^{t+T} is fed to the decoder ψ to reconstruct the learning objective $\hat{S}$ discussed in the next section: $\hat{S} = \psi(h_a^t, h_{tg}, h_g^{t+T})$.

3.2. Learning formulation

3.2.1 Spatiotemporal difference

Our method works with different types of spatiotemporal differences as reconstruction targets. For example:

Structural Similarity Index Measure (SSIM) [50]. This is a method for measuring the perceived quality of the two images based on luminance, contrast, and structure features. To compute our reconstruction target based on SSIM, we apply the SSIM measure locally on corresponding patches between I_t and I_{t+T} to build a similarity map between frames. Then we compute dissimilarity by taking the negation of the similarity map.

Frame differences. When the video background is static with little noise, simple frame differences, such as absolute difference ($S_{|d|} = |I_{t+T} - I_t|$) or raw difference ($S_d = I_{t+T} - I_t$), can also be directly applied as a reconstruction target.

3.2.2 Reconstruction loss

We apply perceptual loss [24] for reconstructing the spatiotemporal difference S. Perceptual loss compares the L2 distance between the features computed from VGG network ϕ [42]. The reconstruction $\hat{S}$ and the target S are fed to VGG network, and mean squared error is applied to the features from the intermediate convolutional blocks:

$$L_{\text{recon}} = \left\| \phi(S(I_t, I_{t+T})) - \phi(\hat{S}(I_t, I_{t+T})) \right\|_2^2. \quad (1)$$

3.2.3 Rotation equivariance loss

In cases where agents can move in many directions (e.g. mice filmed from above can translate and rotate freely), we would like our keypoints to remain semantically consistent. We enforce rotation-equivariance in the discovered keypoints by rotating the image with different angles and imposing that the predicted keypoints should move correspondingly. We apply the rotation equivariance loss (similar to the deformation equivariance in [48]) on the generated heatmap.

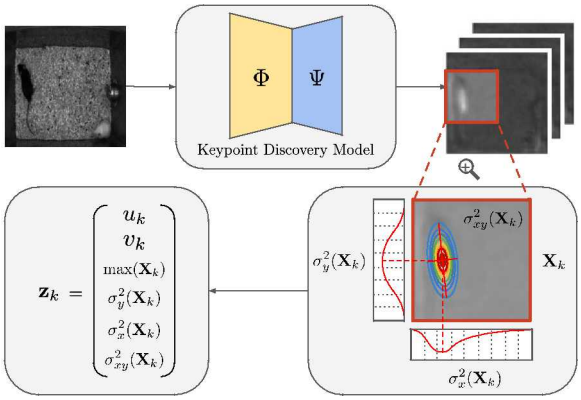

Figure 3. Behavior Classification Features. Extracting information from the raw heatmap (Section 3.3): the confidence scores and the covariance matrices are computed from normalized heatmaps. Note that the features are computed for all x, y coordinates. We visualize the zoomed area around the target instance for illustrative purposes.

Given reference image I and the corresponding geometry bottleneck h_g, we rotate the geometry bottleneck to generate pseudo labels h_g^R for rotated input images I^R with degree $R = \{90, 180, 270\}$. We apply mean squared error between the predicted geometry bottlenecks h_g from the rotated images and the generated pseudo labels h_g^R:

$$L_r = \left\| h_g^{R^\circ} - \hat{h}_g(I^{R^\circ}) \right\|_2^2. \quad (2)$$

3.2.4 Separation loss

Empirical results show that rotation equivariance encourages the discovered keypoints to converge at the center of the image. We apply separation loss to encourage the keypoints to encode unique coordinates, and prevent the discovered keypoints from being centered at the image coordinates [53]. The separation loss is defined as follows:

$$L_s = \sum_{i \neq j} \exp\left(\frac{C - (p_i - p_j)^2}{2\sigma_s^2} \right). \quad (3)$$

3.2.5 Final objective

Our final loss function is composed of three parts: reconstruction loss L_{recon}, rotation equivariance loss L_r, and separation loss L_s:

$$L = L_{\text{recon}} + \mathbb{1}_{\text{epoch}>n}(w_r L_r + w_s L_s). \quad (4)$$

We adopt curriculum learning [2] and apply L_r and L_s once the keypoints are consistently discovered from the semantic parts of the target instance.

3.3. Feature extraction for behavior analysis

Following standard approaches [5, 18, 40], we use the discovered keypoints from B-KinD as input to a behavior quantification module: either supervised behavior classifiers or a physics-based model. Note that this is a separate process from keypoint discovery; we feed discovered geometry information into a downstream model.

In addition to discovered keypoints, we extracted additional features from the raw heatmap (Figure 3) to be used as input to our downstream modules. For instance, we found that the confidence and the shape information from the of the network prediction of keypoint location was informative. When a target part is well localized, our keypoint discovery network produces a heatmap with a single high peak with low variance; conversely, when a target part is occluded, the raw heatmap contains a blurred shape with lower peak value. This "confidence" score (heatmap peak value) is also a good indicator for whether keypoints are discovered on the background (blurred over the background with low confidence) or tracking anatomical body parts (peaked with high confidence), visualized in Supplementary materials. The shape of a computed heatmap can also reflect shape information of the target (e.g. stretching).

Given a raw heatmap $\mathbf{X}_k$ for part k, the confidence score is obtained by choosing the maximum value from the heatmap, and the shape information is obtained by computing the covariance matrix from the heatmap. Figure 3 visualizes the features we extract from the raw heatmaps. Using the normalized heatmap as the probability distribution, additional geometric features are computed:

$$\begin{aligned}\sigma_x^2(\mathbf{X}_k) &= \sum_{ij}(x_i - u_k)^2\mathbf{X}_k(i,j),\\ \sigma_y^2(\mathbf{X}_k) &= \sum_{ij}(y_j - v_k)^2\mathbf{X}_k(i,j),\\ \sigma_{xy}^2(\mathbf{X}_k) &= \sum_{ij}(x_i - u_k)(y_j - v_k)\mathbf{X}_k(i,j).\end{aligned} \tag{5}$$

4. Experiments

We demonstrate that B-KinD is able to discover consistent keypoints in real-world behavioral videos across a range of organisms (Section 4.1.1). We evaluate our keypoints on downstream tasks for behavior classification (Section 4.2) and pose regression (Section 4.3), then illustrate additional applications of our keypoints (Section 4.4).

4.1. Experimental setting

4.1.1 Datasets

CalMS21. CalMS21 [43] is a large-scale dataset for behavior analysis consisting of videos and trajectory data from a pair of interacting mice. Every frame is annotated by an expert for three behaviors: sniff, attack, mount. There are 507k frames in the train split, and 262k frames in the test split (video frame: 1024×570, mouse: approx 150×50). We use only the train split on videos without miniscope cable to train B-KinD. Following [43], the downstream behavior classifier is trained on the entire training split, and performance is evaluated on the test split.

MARS-Pose. This dataset consists of a set of videos with similar recording conditions to the CalMS21 dataset. We use a subset of the MARS pose dataset [40] with keypoints from manual annotations to evaluate the ability of our model to predict human-annotated keypoints, with $\{10, 50, 100, 500\}$ images for train and 1.5k images for test.

Fly vs. Fly. These videos consists of interactions between a pair of flies, annotated per frame by domain experts. We use the Aggression videos from the Fly vs. Fly dataset [15], with the train and test split having 1229k and 322k frames respectively (video frame: 144×144, fly: approx 30×10). Similar to [44], we evaluate on behaviors of interest with more than 1000 frames in the training set (lunge, wing threat, tussle).

Human 3.6M. Human 3.6M [20] is a large-scale motion capture dataset, which consists of 3.6 million human poses and images from 4 viewpoints. To quantitatively measure the pose regression performance against baselines, we use the Simplified Human 3.6M dataset, which consists of 800k training and 90k testing images with 6 activities in which the human body is mostly upright. We follow the same evaluation protocol from [53] to use subjects 1, 5, 6, 7, and 8 for training and 9 and 11 for testing. We note that each subject has different appearance and clothing.

Jellyfish. The jellyfish data is an in-house video dataset containing 30k frames of recorded swimming jellyfish (video frame: 928×1158, jellyfish: approx 50 pix in diameter). We use this dataset to qualitatively test the performance of B-KinD on a new organism, and apply our keypoints to detect the pulsing motion of the jellyfish.

Vegetation. This is an in-house dataset acquired over several weeks using a drone to record the motion of swaying trees. The dataset consists of videos of an oak tree and corresponding wind speeds recorded using an anemometer, with a total of 2.41M video frames (video frame: 512×512, oak tree: varies, approx $\frac{1}{4}$ of the frame). We evaluate this dataset using a physics-based model [7] that relates the visually observed oscillations to the average wind speeds.

4.1.2 Training and evaluation procedure

We train B-KinD using the full objective in Section 3.2.5. During training, we rescale images to 256×256 and use T of around 0.2 seconds, except Human3.6M, where we use 128×128. Unless otherwise specified, all experiments are ran with all keypoints discovered from B-KinD with SSIM reconstruction and with 10 keypoints for mouse, fly, and jel-

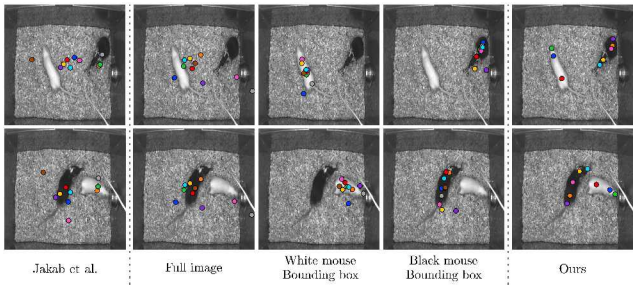

Figure 4. **Comparison with existing methods** [21], full image, bounding box, and SSIM reconstruction (ours). "Jakab *et al.*" and "full image" results are based on full image reconstruction. "White mouse bounding box" and "black mouse bounding box" show the results when the cropped bounding boxes were fed to the network for image reconstruction.

lyfish, 16 keypoints for Human3.6M, 15 keypoints for Vegetation. We train on the train split of each dataset as specified, except for jellyfish and vegetation, where we use the entire dataset. Additional details are in the Supplementary materials.

After training the keypoint discovery model, we extract the keypoints and use it for different evaluations based on the labels available in the dataset: behavior classification (CalMS21, Fly), keypoint regression (MARS-Pose, Human), and physics-based modeling (Vegetation).

For keypoint regression, similar to previous works [21, 22], we compare our regression with a fully supervised 1-stack hourglass network [33]. We evaluate keypoint regression on Simplified Human 3.6M by using a linear regressor without a bias term, following the same evaluation setup from previous works [28,53]. On MARS-Pose, we train our model in a semi-supervised fashion with 10,50, 100, 500 supervised keypoints to test data efficiency. For behavior classification, we evaluate on CalMS21 and Fly, using available frame-level behavior annotations. To train behavior classifiers, we use the specified train split of each dataset. For CalMS21 and Fly, we train the 1D Convolutional Network benchmark model provided by [43] using B-KinD keypoints. We evaluate using mean average precision (MAP) weighted equally over all behaviors of interest.

4.2. Behavior classification results

CalMS21 Behavior Classification. We evaluate the effectiveness of B-KinD for behavior classification (Table 1). Compared to supervised keypoints trained for this task, our keypoints (without manual supervision) is comparable when using both pose and confidence as input. Compared to other self-supervised methods, even those that use bounding boxes, our discovered keypoints on the full image generally achieve better performance.

Keypoints discovered with image reconstruction, similar to baselines [21,38] cannot track the agents well without using bounding box information (Figure 4) and does not perform well for behavior classification (Table 1). When we provide bounding box information to the model based on image reconstruction, the performance is significantly improved, but this model does not perform as well as B-KinD keypoints from spatiotemporal difference reconstruction.

For the per-class performance (see the Supplementary materials), the biggest gap exists between B-KinD and MARS on the "attack" behavior. This is likely because during attack, the mice are moving quickly, and there exists a lot of motion blur and occlusion which is difficult to track without supervision. However, once we extract more information from the heatmap, through computing keypoint confidence, our keypoints perform comparably to MARS.

Fly Behavior Classification. The FlyTracker [15] uses hand-crafted features computed from the image, such as contrast, as well as features from tracked fly body parts,

CalMS21	Pose	Conf	Cov	MAP
Fully supervised				
MARS † [40]				.856 ± .010
				.874 ± .003
				.880 ±.005
Self-supervised				
Jakab et al. [21]				.186 ±.008
Image Recon.				.182 ±.007
				.184 ±.006
				.165 ±.012
Image Recon. bbox†				.819 ±.008
				.812 ± .006
				.812 ±.010
Ours				.814 ±.007
				.857 ±.005
				.852 ±.013

Table 1. **Behavior Classification Results on CalMS21**. "Ours" represents classifiers using input keypoints from our discovered keypoints. "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 std dev from 5 classifier runs are shown.

Fly	MAP
Hand-crafted features	
FlyTracker [15]	.809 ± .013
Self-supervised + generic features	
Image Recon.	.500 ± .024
Image Recon. bbox†	.750 ± .020
Ours	.727 ± .022

Table 2. **Behavior Classification Results on Fly**. "FlyTracker" represents classifiers using hand-crafted inputs from [15]. The self-supervised keypoints all use the same "generic features" computed on all keypoints: speed, acceleration, distance, and angle. † refers to models that require bounding box inputs before keypoint estimation. Mean and std dev from 5 classifier runs are shown.

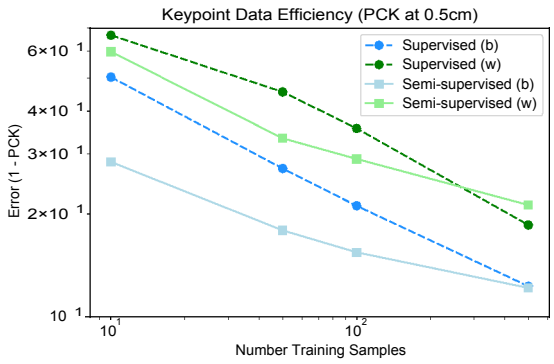

Figure 5. **Keypoint data efficiency on MARS-Pose**. The supervised model is based on [40] using stacked hourglass [33], while the semi-supervised model uses both our self-supervised loss and supervision. PCK is computed at *0.5cm* threshold, averaged across nose, ears, and tail keypoints, over 3 runs. "b" and "w" indicates the black and white mouse respectively.

such as wing angle or distance between flies. Using discovered keypoints, we compute comparable features without assuming keypoint identity, by computing speed and acceleration of every keypoint, distance between every pair, and angle between every triplet. For all self-supervised methods, we use keypoints, confidence, and covariance for behavior classification. Results demonstrate that while there is a small gap in performance to the supervised estimator, our discovered keypoints perform much better than image reconstruction, and is comparable to models that require bounding box inputs (Table 2).

4.3. Pose regression results

MARS Pose Regression. We evaluate the pose estimation performance of our method in the setting where some human annotated keypoints exist (Figure 5). For this experiment, we train B-KinD in a semi-supervised fashion, where the loss is a sum of both our keypoint discovery objective (Section 3.2.5) as well as standard keypoint estimation objectives based on MSE [40]. For both black and white mouse, when using our keypoint discovery objective in a semi-supervised way during training, we are able to track keypoints more accurately compared to the supervised method [40] alone. We note that the performance of both methods converge at around 500 annotated examples.

Simplified Human 3.6M Pose Regression. To compare with existing keypoint discovery methods, we evaluate our discovered keypoints on Simplified Human3.6M (a standard benchmarking dataset) by regressing to annotated keypoints (Table 3). Though our method is directly applicable to full images, we train the discovery model using cropped bounding box for a fair comparison with baselines, which all use cropped bounding boxes centered on the subject. Compared to both self-supervised + prior information and self-supervised + regression, our method shows state-of-the-art performance on the keypoint regression task, suggesting spatiotemporal difference is an effective reconstruction target for keypoint discovery.

Simplified H36M	all	wait	pose	greet	direct	discuss	walk
Fully supervised:							
Newell [33]	2.16	1.88	1.92	2.15	1.62	1.88	2.21
Self-supervised + unpaired labels							
Jakab [22]$	2.73	2.66	2.27	2.73	2.35	2.35	4.00
Self-supervised + template							
Schmidtke [39]	3.31	3.51	3.28	3.50	3.03	2.97	3.55
Self-supervised + regression							
Thewlis [48]	7.51	7.54	8.56	7.26	6.47	7.93	5.40
Zhang [53]	4.14	5.01	4.61	4.76	4.45	4.91	4.61
Lorenz [28]	2.79	–	–	–	–	–	–
Ours (best)	2.44	2.50	2.22	2.47	2.22	2.77	2.50
Ours (mean)	2.53	2.58	2.31	2.56	2.34	2.83	2.58
Ours (std)	.056	.047	.062	.048	.066	.048	.063

Table 3. **Comparison with state-of-the-art methods for landmark prediction on Simplified Human 3.6M**. The error is in %-MSE normalized by image size. All methods predict 16 keypoints except for [22]‡, which uses 32 keypoints for training a prior model from the Human 3.6M dataset. B-Kind results are computed from 5 runs.

Learning Objective	%-MSE
Image Recon.	2.918 ± 0.139
Abs. Difference	2.642 ± 0.174
Difference	2.770 ± 0.158
SSIM	**2.534 ± 0.056**
Self-supervised + extracted features	
SSIM	**2.494 ± 0.047**

Table 4. **Learning objective ablation on Simplified Human3.6M**. %-MSE error is reported by changing the reconstruction target. Extracted features correspond to keypoint locations, confidence, and covariance. Results are from 5 B-KinD runs.

Learning Objective Ablation Study We report the pose regression performance on Simplified Human3.6M (Table 4) by varying the spatiotemporal difference reconstruction target for training B-KinD. Here, image reconstruction also performs well since cropped bounding box is used as an input to the network. Overall, spatiotemporal difference reconstruction yield better performance over image reconstruction, and performance can be further improved by extracting additional confidence and covariance information from the discovered heatmaps.

4.4. Additional applications

We show qualitative performance and demonstrate additional downstream tasks using our discovered keypoints, on pulse detection for Jellyfish and on wind speed regression for Vegetation.

Qualitative Results. Qualitative results (Figure 6) demonstrates that B-KinD is able to track some body parts consistently, such as the nose of both mice and keypoints

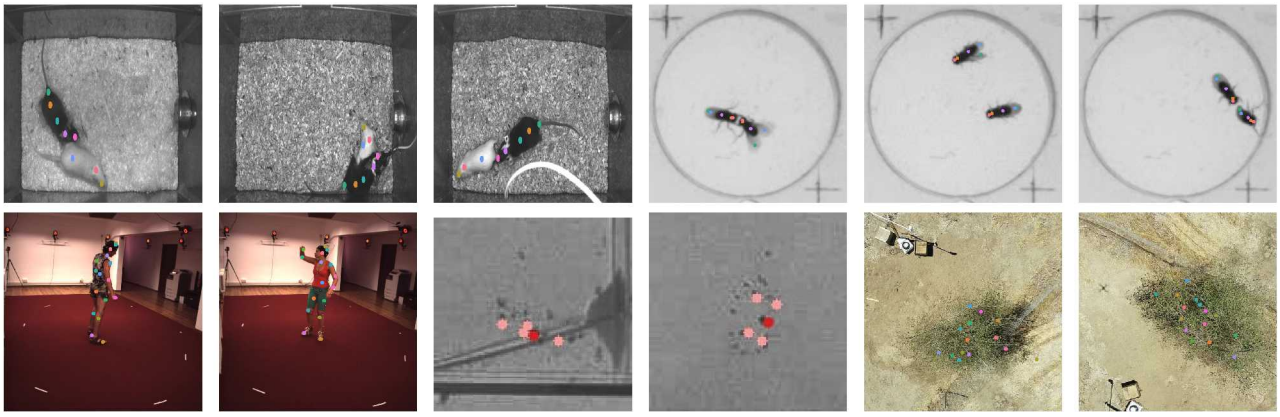

Figure 6. **Qualitative Results of B-KinD**. Qualitative results for B-KinD trained on CalMS21 (mouse), Fly vs. Fly (fly), Human3.6M (human), jellyfish and Vegetation (tree). Additional visualizations are in the Supplementary materials.

along the spine; the body and wings of the flies; the mouth and gonads of the jellyfish; and points on the arms and legs of the human. For visualization only, we show only keypoints discovered with high confidence values (Section 3.3); for all other experiments, we use all discovered keypoints.

Pulse Detection. Jellyfish swimming is among the most energetically efficient forms of transport, and its control and mechanics are studied in hydrodynamics research [10]. Of key interest is the relationship between body plan and swim pulse frequency across diverse jellyfish species. By computing distance between B-KinD keypoints, we are able to extract a frequency spectrogram to study jellyfish pulsing, with a visible band at the swimming frequency (Supplementary materials). This provides a way to automatically annotate swimming behavior, which could be quickly applied to video from multiple species to characterize the relationship between swimming dynamics and body plan.

Wind Speed Modeling. Measuring local wind speed is useful for tasks such as tracking air pollution and weather forecasting [6]. Oscillations of trees encode information on wind conditions, and as such, videos of moving trees could function as wind speed sensors [6, 7]. Using the Vegetation dataset, we evaluate the ability of our keypoints to predict wind speed using a physics-based model [7]. This model defines the relationship between the mean wind speed and the structural oscillations of the tree, and requires tracking these oscillations from video, which was previously done manually. We show that B-KinD can accomplish this task automatically. Using our keypoints, we are able to regress the measured ground truth wind speed with an $R^2 = 0.79$, suggesting there is a good agreement between the proportionality assumption from [7] and the experimental results using the keypoint discovery model.

Limitations. One issue we did not explore in detail, and which will require further work, is keypoint discovery for agents that may be partially or completely occluded at some point during observation, including self-occlusion. Additionally, similar to other keypoint discovery models [28, 39, 53], we observe left/right swapping of some body parts, such as the legs in a walking human. One approach that might overcome these issues would be to extend our model to discover the 3D structure of the organism, for instance by using data from multiple cameras. Despite these challenges, our model performs comparably to supervised keypoints for behavior classification.

5. Discussion and conclusion

We propose B-KinD, a self-supervised method to discover meaningful keypoints from unlabelled videos for behavior analysis. We observe that in many settings, behavioral videos have fixed cameras recording agents moving against a (quasi) stationary background. Our proposed method is based on reconstructing spatiotemporal difference between video frames, which enables B-KinD to focus on keypoints on the moving agents. Our approach is general, and is applicable to behavior analysis across a range of organisms without requiring manual annotations.

Results show that our discovered keypoints are semantically meaningful, informative, and enable performance comparable to supervised keypoints on the downstream task of behavior classification. Our method will reduce the time and cost dramatically for video-based behavior analysis, thus accelerating scientific progress in fields such as ethology and neuroscience.

6. Acknowledgements

This work was generously supported by the Simons Collaboration on the Global Brain grant 543025 (to PP and DJA), NIH Award #R00MH117264 (to AK), NSF Award #1918839 (to YY), NSF Award #2019712 (to JOD and RHG), NINDS Award #K99NS119749 (to BW), NIH Award #R01MH123612 (to DJA and PP), NSERC Award #PGSD3-532647-2019 (to JJS), as well as a gift from Charles and Lily Trimble (to PP).

References

[1] David J Anderson and Pietro Perona. Toward a science of computational ethology. *Neuron*, 84(1):18–31, 2014. 2

[2] Yoshua Bengio, Je´ro ̂me Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 4

[3] Gordon J Berman, Daniel M Choi, William Bialek, and Joshua W Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014. 2

[4] Kristin Branson, Alice A Robie, John Bender, Pietro Perona, and Michael H Dickinson. High-throughput ethomics in large groups of drosophila. *Nature methods*, 6(6):451–457, 2009. 1,2

[5] Xavier P Burgos-Artizzu, Piotr Dolla´r, Dayu Lin, David J Anderson, and Pietro Perona. Social behavior recognition in continuous video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1322–1329. IEEE, 2012. 2, 5

[6] 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'Alche´-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 8, 14

[7] 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. 1, 2, 3, 5, 8, 14, 15

[8] 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. 2, 15

[9] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[10] 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. 8, 14

[11] Heiko Dankert, Liming Wang, Eric D Hoopfer, David J Anderson, and Pietro Perona. Automated monitoring and analysis of social behavior in drosophila. *Nature methods*, 6(4):297–303, 2009. 1, 2

[12] Brian M de Silva, David M Higdon, Steven L Brunton, and J Nathan Kutz. Discovery of physics from data: universal laws and discrepancies. *Frontiers in artificial intelligence*, 3:25, 2020. 1

[13] SE Roian Egnor and Kristin Branson. Computational analysis of behavior. *Annual review of neuroscience*, 39:217–236, 2016. 2

[14] Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. *ICLR*, 2017. 2

[15] 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. 1, 2, 5, 6, 16

[16] Jacob M Graving, Daniel Chae, Hemal Naik, Liang Li, Benjamin Koger, Blair R Costelloe, and Iain D Couzin. Deepposekit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife*, 8:e47994, 2019. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016. 15

[18] Weizhe Hong, Ann Kennedy, Xavier P Burgos-Artizzu, Moriel Zelikowsky, Santiago G Navonne, Pietro Perona, and David J Anderson. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proceedings of the National Academy of Sciences*, 112(38):E5351–E5360, 2015. 2,5

[19] Alexander I Hsu and Eric A Yttri. B-soid, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature communications*, 12(1):1–13, 2021. 2

[20] 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. 5, 16

[21] 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. 2, 3, 6, 13

[22] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 6, 7

[23] Hueihan Jhuang, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D Steele, and Thomas Serre. Automated home-cage behavioural phenotyping of mice. *Nature communications*, 1(1):1–10, 2010. 2

[24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 4

[25] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10(1):64, 2013. 2

[26] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. *arXiv preprint arXiv:1910.02027*, 2019. 3

[27] Jingyuan Liu, Mingyi Shi, Qifeng Chen, Hongbo Fu, and Chiew-Lan Tai. Normalized human pose features for human action video alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11521–11531, 2021. 1

[28] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Björn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, 2019. 3, 6, 7, 8, 17

[29] Kevin Luxem, Falko Fuhrmann, Johannes Kürsch, Stefan Remy, and Pavol Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *bioRxiv*, 2020. 2

[30] Julian Marstaller, Frederic Tausch, and Simon Stock. Deepbees-building and scaling convolutional neuronal nets for fast and large-scale visual monitoring of bee hives. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[31] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018. 2

[32] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *arXiv preprint arXiv:1906.07889*, 2019. 3

[33] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, 2016. 2, 6, 7, 16

[34] Simon RO Nilsson, Nastacia L Goodwin, Jia J Choong, Sophia Hwang, Hayden R Wright, Zane Norville, Xiaoyu Tong, Dayu Lin, Brandon S Bentzley, Neir Eshel, et al. Simple behavioral analysis (simba): an open source toolkit for computer classification of complex social behaviors in experimental animals. *BioRxiv*, 2020. 2

[35] Talmo D Pereira, Joshua W Shaevitz, and Mala Murthy. Quantifying behavior to understand the brain. *Nature neuroscience*, 23(12):1537–1549, 2020. 1, 2

[36] Talmo D Pereira, Nathaniel Tabris, Junyu Li, Shruthi Ravindranath, Eleni S Papadoyannis, Z Yan Wang, David M Turner, Grace McKenzie-Smith, Sarah D Kocher, Annegret Lea Falkner, et al. Sleap: Multi-animal pose tracking. *BioRxiv*, 2020. 2

[37] Serim Ryou, Seong-Gyun Jeong, and Pietro Perona. Anchor loss: Modulating loss scale based on prediction difficulty. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[38] Serim Ryou and Pietro Perona. Weakly supervised keypoint discovery. *CoRR*, abs/2109.13423, 2021. 2, 3, 6, 15

[39] 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. 3, 7, 8, 17

[40] 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. 1, 2, 5, 6, 7, 13, 16

[41] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 39(4):640–651, 2017. 2

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 4

[43] 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. 2, 3, 5, 6, 11, 16

[44] 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. 1, 2, 5, 16

[45] Jennifer J Sun, Jiaping Zhao, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, and Ting Liu. View-invariant probabilistic embedding for human pose. In *European Conference on Computer Vision*, pages 53–70. Springer, 2020. 1

[46] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[47] Philippe The´venaz. Stackreg: an imagej plugin for the recursive alignment of a stack of images. *Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne*, 2012, 1998. 17

[48] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 4, 7

[49] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, D. Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3349–3364, 2021. 2

[50] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600–612, 2004. 4

[51] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. IEEE CVPR*, 2016. 2

[52] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015. 2

[53] 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. 2, 3, 4, 5, 6, 7, 8, 16, 17

Supplementary Material

We present additional experimental results (Section A), additional implementation details (Section B), and visualizations (Section C). 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.

A. Additional Experimental Results

A.1. CalMS21 Ablation Study

Similar to the main paper, we evaluate CalMS21 on the behavior classification train/test split described in task 1 [43], 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 5). 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 7), and due to the large proportion of background keypoints, we do not use background keypoints in the 20 keypoints case for 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.

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 6). In particular, we are able to achieve comparable performance on behavior classification to supervised keypoints (Table 8) 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 7). The rotation equivariance loss is qualitatively important for tracking semantically consistent parts of the mouse (Figure 8) 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

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 5. **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.

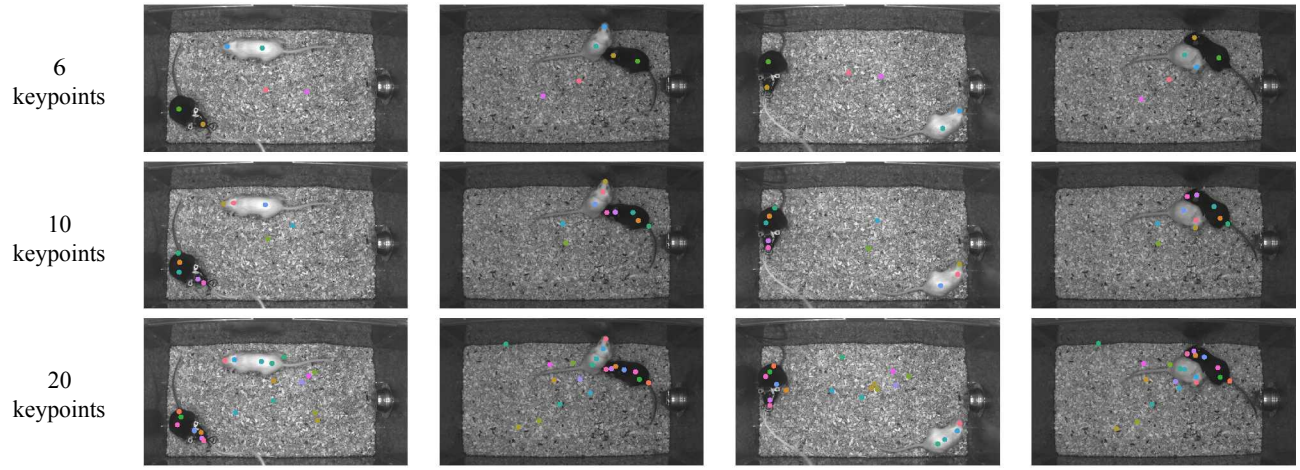

Figure 7. **Qualitative Results on CalMS21 by varying the number of keypoints**. We train the keypoint discovery model with different numbers of discovered keypoints. Each row shows qualitative results with all the keypoints including the background keypoints. We note that there are 2 background (low-confidence) keypoints for 6 and 10 discovered keypoints, and 9 background keypoints for 20 discovered keypoints.

# 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 6. **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.

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

Table 7. **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.

only, we note that adding confidence and covariance significantly improves the performance of the reconstruction loss only model (Table 7).

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 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).

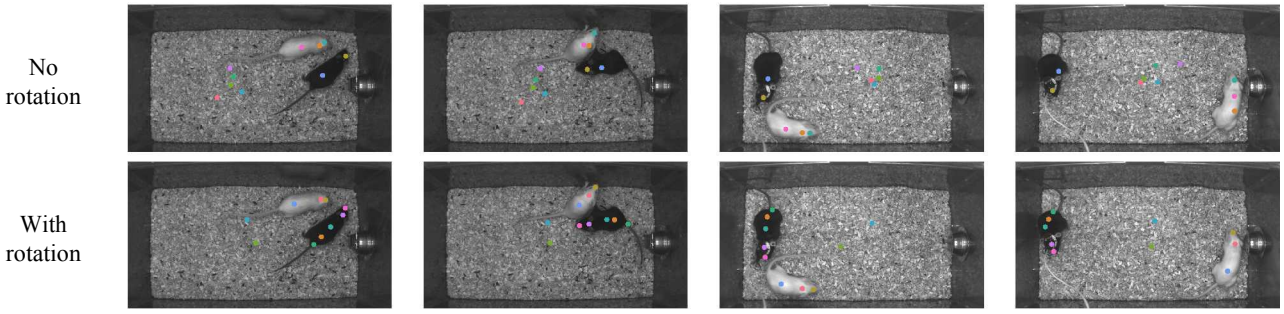

Figure 8. **Qualitative Results on CalMS21 for loss ablation study**. With the full training objective for our discovered keypoints, we are able to track 8/10 keypoints consistently, while without rotation loss, there are only 5/10 tracked keypoints on both mice. Additionally, some of the discovered 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 7.

CalMS21	Pose	Conf	Cov	MAP	Attack AP	Investigation AP	Mount AP
				Fully supervised			
				.856 ± .010	.724 ± .023	.893 ± .005	.950 ± .004
MARS † [40]				.874 ± .003	.790 ± .004	.890 ± .006	.943 ± .004
				.880 ± .005	.804 ± .012	.902 ± .004	.934 ± .006
				Self-supervised			
Jakab et al. [21]				.186±.008	.135±.019	.254±.019	.170±.029
				.182±.007	.111±.016	.217±.011	.219±.021
Image Recon.				.184±.006	.114±.006	.209±.012	.229±.021
				.165±.012	.110±.016	.218±.013	.167±.038
				.819 ± .008	.680 ± .028	.861 ± .007	.918 ± .007
Image Recon. bbox†				.812±.006	.694±.011	.818±.016	.923±.013
				.812 ± .010	.709 ± .008	.806 ± .019	.922 ± .013
				.814 ± .007	.654 ± .025	.861 ± .003	.925 ± .014
Ours				.857 ± .005	.763 ± .015	.879 ± .009	.928 ± .006
				.852 ± .013	.751 ± .025	.870 ± .009	.935 ± .010

Table 8. **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.

A.2. CalMS21 Per-Class Performance

B-KinD keypoints achieve comparable performance to supervised keypoints when using pose and confidence features from the heatmap (Table 8). 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 16). 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.

A.3. Human3.6M Ablation Study

We evaluate the effect of number of keypoints and frame gaps on simplified Human 3.6M (Table 9). 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.

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 9. **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.

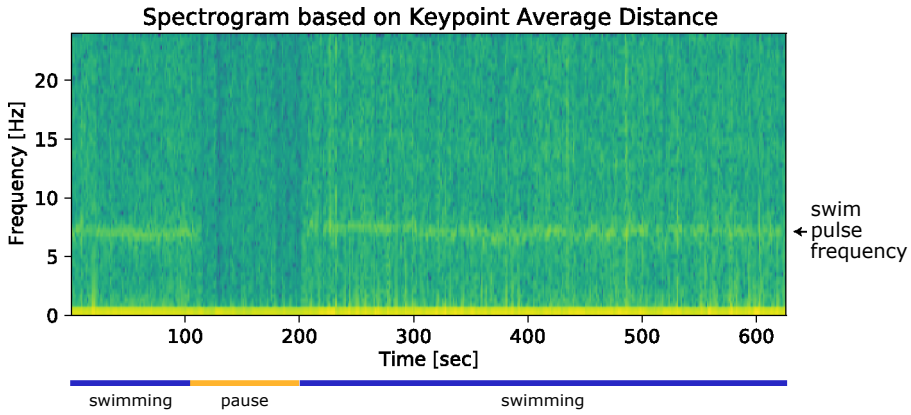

Figure 9. **Spectrogram from Distance of Discovered Keypoints**. From a recorded video of jellyfish swimming at 48Hz, we discover keypoints at each frame using our model and compute a spectrogram based on the average distance between discovered keypoints on the jellyfish.

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.

A.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 [10]. 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 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 9). 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.

A.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 [6, 7]. 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

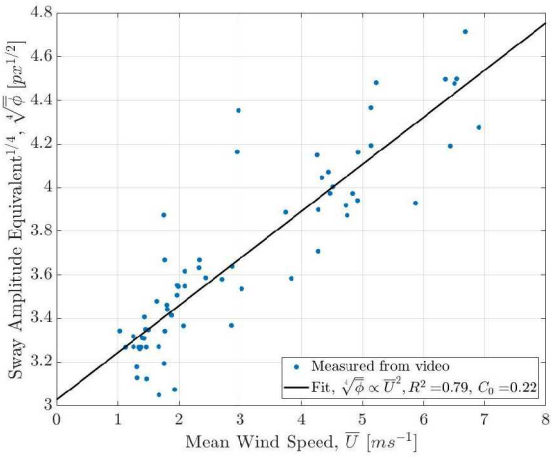

Figure 10. Wind Speed Regression from Discovered Keypoints. Mean wind speed, $\bar{U}$, vs. the fourth root of the sway amplitude equivalent measured from the standard deviation of the convex hull area of the 15 discovered keypoints in each clip, based on model from [7]. The scatter represents 10-mintute 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.

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 [7] to study the relationship between oscillations of trees and wind speed. This model defines the relationship between structural oscillation and wind speed as:

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

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

$$C_0 \sqrt[4]{\bar{\phi}} \propto \bar{U}$$

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 10).

B. Additional Implementation Details

Architecture Details Our method uses ResNet-50 [17] as an encoder Φ, GlobalNet [8] as a pose decoder Ψ, and a series of convolution blocks as a reconstruction decoder ψ, following the unsupervised keypoint discovery model from [38]. Architecture details about reconstruction decoder is shown in Table 10. 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 11. 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.

Table 10. **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 x 2	1024	16x16
Upsampling	-	-	32x32
Conv block	1024 + # keypoints x 2	512	32x32
Upsampling	-	-	64x64
Conv block	512 + # keypoints x 2	256	64x64
Upsampling	-	-	128x128
Conv block	256 + # keypoints x 2	128	128x128
Upsampling	-	-	256x256
Conv block	128 + # keypoints x 2	64	256x256
Convolution	64	3	256x256

CalMS21. The CalMS21 dataset [43] 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 x 570 at 30 Hz. The supervised keypoints provided with CalMS21 are from the MARS detector [40] 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 [43] 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 [40] and these images are recorded in similar recording conditions to CalMS21 [43]. 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 [33] and for the semi-supervised model, we add a supervised keypoint estimation loss based on MSE to our keypoint discovery framework.

Fly vs. Fly. This dataset consists of videos of two interacting flies [15] with frame-level behavior annotations. We use the "Aggression" videos from this dataset (144 x 144 at 30 Hz) and use the behaviors with more than 1000 annotated training samples, with the same setup as [44]. 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 [20] 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 [20]. Simplified Human 3.6M dataset, introduced by [53], consists of 6 different activities with mostly upright poses by cropping the full image using bounding box. Since our method requires

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 11. **Hyperparameters for Keypoint Discovery.**

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 (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 A.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 [47] to align video frames before computing the spatiotemporal difference reconstruction target for our method. More details on the wind speed regression is in Section A.5.

C. Visualizations

We present additional visualization results on mouse (Figure 11), fly (Figure 12), tree (Figure 13), and human (Figure 14). 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 16). 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 16), the confidence varies from 0.5 0.6 when the nose is visible in the first two examples to 0.3 0.4 when the nose is harder to see in the last two examples.

Challenges. Difficult examples for our model are visualized in Figure 15. 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 16) 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 [28,39, 53]). 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.

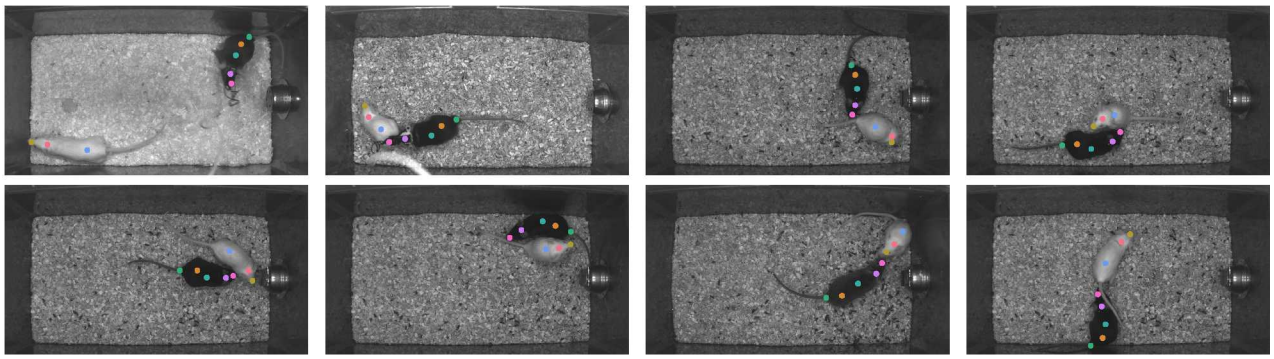

Figure 11. **Qualitative Results on CalMS21**. We observe that keypoints are discovered for noses of both mice and generally along the spine of the mice.

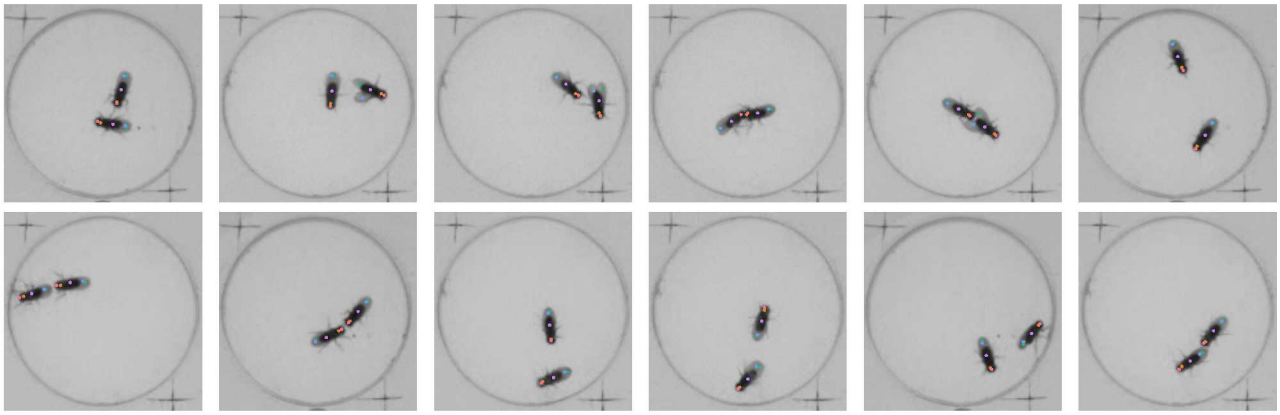

Figure 12. **Qualitative Results on Fly-vs-Fly**. We observe that 3 keypoints are discovered on the body of the fly, with 2 on the wings (one for each wing).

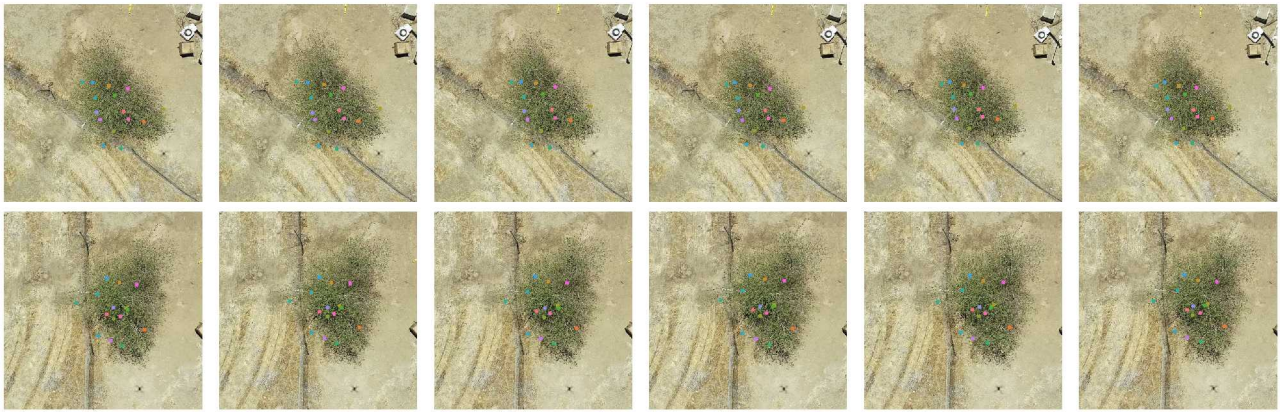

Figure 13. **Qualitative Results on Vegetations**. Each row shows different frames with discovered keypoints from a single video. Our model can discover and track consistent keypoints within the same video.

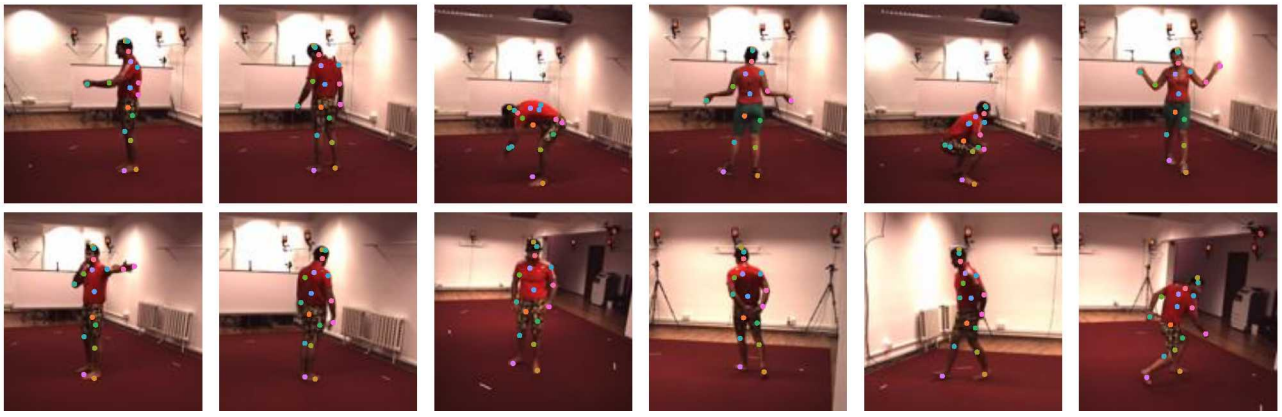

Figure 14. **Qualitative Results on Simplified Human 3.6M**. We observe that keypoints are generally discovered on visible joints and end points of humans, such as head, elbows, hands, upper legs, knees and feet. We note that there is left/right swapping of body parts, since when the human is facing forwards or backwards, keypoints are generally on the same side.

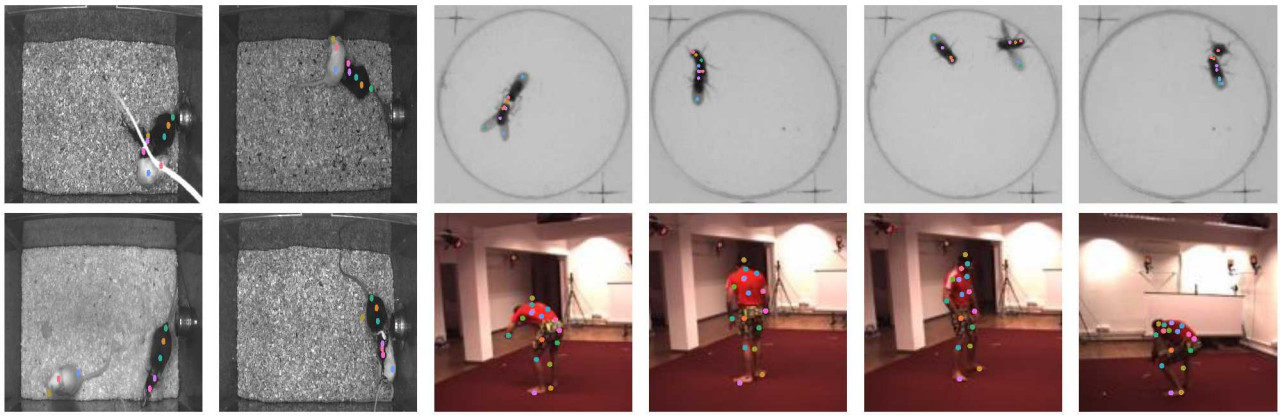

Figure 15. **Limitations**. We visualize examples that are difficult for our model, for example from occlusion/agents being in close proximity (mouse, fly), self-occlusion (human), unusual poses (human, fly), and left-right swapping (human).

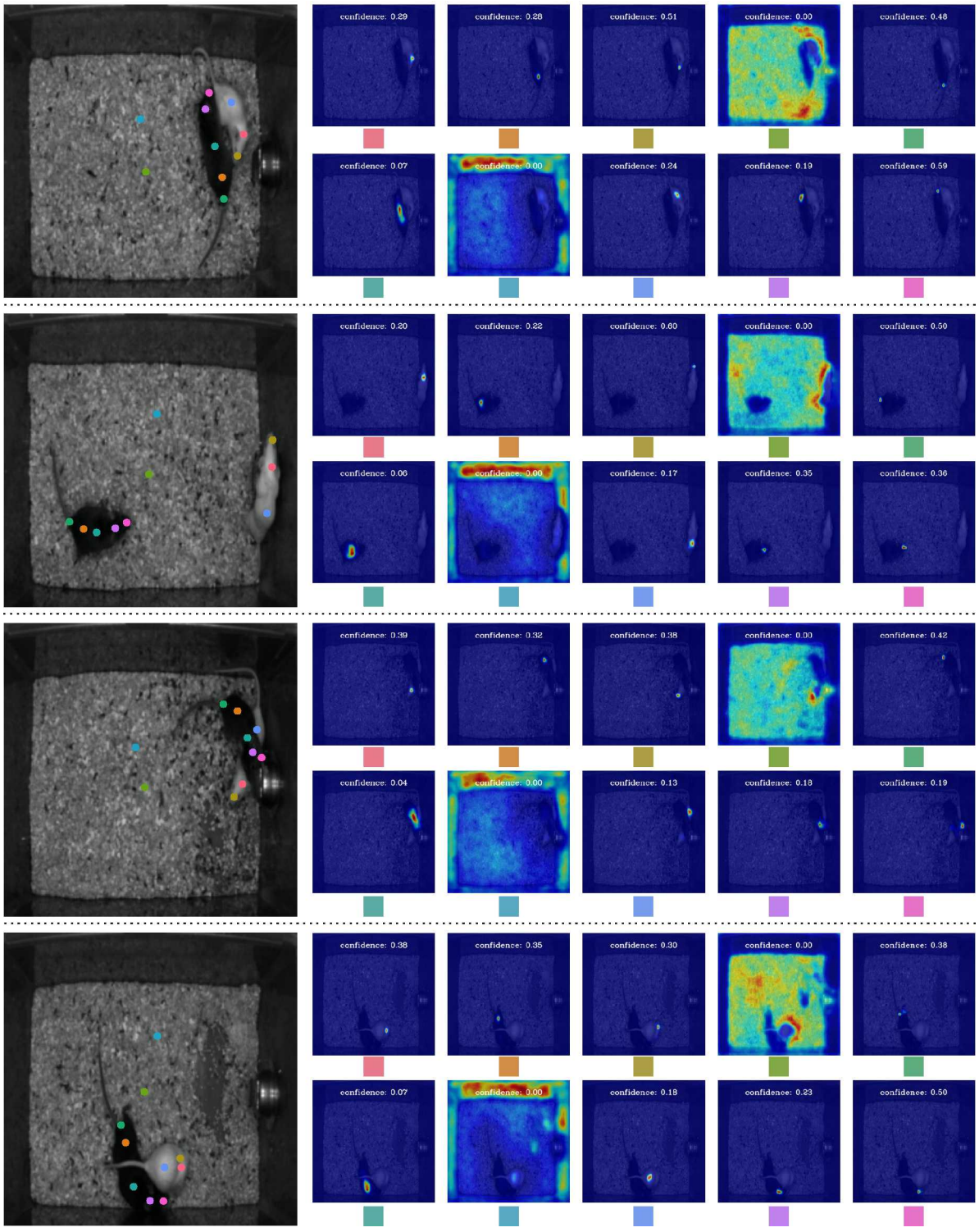

Figure 16. **Confidence visualization on CalMS21**. Confidence score (maximum prediction value) is shown with the normalized heatmap. Background keypoints (fourth on row 1 and second on row 2) have very low confidence.