

**Title:** Fast animal pose estimation using deep neural networks

**Authors:**

Pereira, T.<sup>1,#</sup>, Aldarondo, D.<sup>1,#</sup>, Willmore, L.<sup>1</sup>, Kislin, M.<sup>1</sup>, Wang, S. S.<sup>1,2</sup>, Murthy, M.<sup>1,2\*</sup>, and Shaevitz, J. W.<sup>1,3,4\*</sup>

1 Princeton Neuroscience Institute, Princeton University

2 Department of Molecular Biology, Princeton University

3 Lewis-Sigler Institute for Integrative Genomics, Princeton University

4 Department of Physics, Princeton University

#equal authors

\*lead contacts and co-corresponding authors: Mala Murthy (mmurthy@princeton.edu) and Joshua W. Shaevitz (shaevitz@princeton.edu)

**Abstract:**

1 Recent work quantifying postural dynamics has attempted to define the repertoire of behaviors  
2 performed by an animal. However, a major drawback to these techniques has been their  
3 reliance on dimensionality reduction of images which destroys information about which parts of  
4 the body are used in each behavior. To address this issue, we introduce a deep learning-based  
5 method for pose estimation, LEAP (**L**EAP **E**stimates **A**nimal **P**ose). LEAP automatically predicts  
6 the positions of animal body parts using a deep convolutional neural network with as little as 10  
7 frames of labeled data for training. This framework consists of a graphical interface for  
8 interactive labeling of body parts and software for training the network and fast prediction on  
9 new data (1 hr to train, 185 Hz predictions). We validate LEAP using videos of freely behaving  
10 fruit flies (*Drosophila melanogaster*) and track 32 distinct points on the body to fully describe the  
11 pose of the head, body, wings, and legs with an error rate of <3% of the animal's body length.  
12 We recapitulate a number of reported findings on insect gait dynamics and show LEAP's  
13 applicability as the first step in unsupervised behavioral classification. Finally, we extend the  
14 method to more challenging imaging situations (pairs of flies moving on a mesh-like  
15 background) and movies from freely moving mice (*Mus musculus*) where we track the full  
16 conformation of the head, body, and limbs.

17 **Introduction:**

18 Connecting neural activity with behavior requires methods to parse what an animal does into its  
19 constituent components (movements of its body parts), which can then be connected with the  
20 electrical activity that generates each action. This is particularly challenging for natural behavior,  
21 which is dynamic, complex, and seemingly noisy. Human classification of behavior is  
22 painstakingly slow and subject to bias – but recent methods make it feasible to automate the  
23 analysis of behavior<sup>1</sup>. These include methods to track animal centroids over time<sup>2–4</sup>, machine  
24 learning techniques for identifying user-defined behaviors, such as fighting or courting<sup>5,6</sup>, and  
25 software to segment the acoustic signals produced by an animal<sup>7–9</sup>. However, one may not  
26 know *a priori* which behaviors to analyze – this is particularly true when screening mutant  
27 animals or investigating the results of neural perturbations that can alter behavior in unexpected  
28 ways.

29

30 Recent developments in the unsupervised clustering of postural dynamics have overcome many  
31 of these challenges by analyzing the raw frames of movies in a reduced dimensional space  
32 (e.g., generated using Principal Component Analysis (PCA)). By comparing frequency spectra  
33 or fitting auto-regressive models<sup>10,11</sup>, these methods both define and provide the ability to  
34 record the occurrence of tens to hundreds of unique, stereotyped behaviors in animals such as  
35 fruit flies or mice. These unsupervised methods have been used to uncover new structure in  
36 behavioral data, facilitating the investigation of temporal sequences<sup>12</sup>, social interactions<sup>13</sup>, the  
37 analysis of genetic mutants<sup>11,14</sup>, and the results of neural perturbation<sup>15,16</sup>.

38

39 While powerful, a major drawback to the aforementioned techniques is their reliance on PCA to  
40 reduce the dimensionality of the image time series. While this produces a more manageable  
41 substrate for machine learning, the modes derived from PCA come from the statistics of the  
42 images and are not related directly to any individual body part of the animal. As such, the  
43 discovered stereotyped behaviors must be labeled, classified, and compared manually through  
44 the human observation of representative movie snippets. Given the highly quantitative approach  
45 that precedes this step, it is ultimately unsatisfying and subjective for the experimenter to  
46 manually label each behavior (e.g., foreleg grooming, hindleg grooming, forward locomotion,  
47 right turns, etc.). Instead, what is desired is a mathematical representation of the relative  
48 motions of all parts of the animal that characterizes a particular behavior. Such a description  
49 would facilitate the investigation of the similarities and differences between behaviors and likely  
50 improve the behavioral identification algorithm itself.

51

52 Measuring all of the body part positions from raw images is a challenging computer vision  
53 problem. Previous attempts at automated body-part tracking in insects and mammals have  
54 relied on either physically constraining the animal and having it walk on a spherical treadmill<sup>17</sup>  
55 or linear track<sup>18</sup>, applying physical markers to the animal<sup>17,19</sup>, or utilizing specialized equipment  
56 such as depth cameras<sup>20–22</sup>, frustrated total internal reflection imaging<sup>23,24</sup> or multiple cameras  
57<sup>25</sup>. Meanwhile, approaches designed to operate without constraining the natural space of  
58 behaviors make use of image processing techniques that are sensitive to imaging conditions  
59 and require manual correction even after full training<sup>26</sup>.

60

61 To address these issues, we turned to deep learning-based methods for pose estimation that  
62 have proven successful on images of humans<sup>27–33</sup>. Major breakthroughs in the field have come  
63 from adopting fully convolutional neural network architectures for efficient training and  
64 evaluation of images<sup>34,35</sup> and producing a probabilistic estimate of the position of each tracked  
65 body part<sup>28,30</sup>. However, the problems of pose estimation in the typical human setting and that  
66 for laboratory animals are subtly different. Algorithms that work on human images are meant to  
67 deal with large amounts of heterogeneity in body shape, environment, and image quality, but for  
68 which there are very large labeled training sets of images available. On the contrary, behavioral  
69 laboratory experiments are often more controlled, but the imaging conditions may be highly  
70 specific to the experimental paradigm and labeled data is not readily available and must be  
71 generated for every experimental apparatus and animal type. One recent attempt to apply these  
72 techniques to images of behaving animals successfully used transfer learning, whereby  
73 networks initially trained for human pose estimation are refined by further training with relatively  
74 few samples from animal images<sup>36</sup>.

75

76 We have taken a different approach that combines a graphical user interface (GUI)-driven  
77 workflow for labeling images with a simple network architecture that is easy to train and requires  
78 fewer computations to generate predictions. Our method can automatically predict the positions  
79 of animal body parts via iterative training of deep convolutional neural networks with as little as  
80 10 frames of labeled data for initial prediction and training. After initial *de novo* training,  
81 incrementally refined predictions can be used to guide labeling in new frames, drastically  
82 reducing the time required to label sufficient examples (~500 frames) to achieve an accuracy of  
83 less than 3 pixels (distance from ground truth). Our framework consists of a GUI for interactive  
84 labeling of ground truth body part positions as well as software for efficient training of a

85 convolutional neural network on a workstation with a modern GPU (<1 hour) and fast prediction  
86 on new data (up to 185 Hz). We validate the results of our method using a previously published  
87 dataset of high quality videos of freely behaving adult fruit flies (*Drosophila melanogaster*<sup>10</sup>) and  
88 we recapitulate a number of reported findings on insect gait dynamics as a test of its  
89 experimental validity. We then show its applicability as a front end to an unsupervised  
90 behavioral classification algorithm and demonstrate how it can be used to describe stereotyped  
91 behaviors in terms of the dynamics of individual body parts. Finally, we show the generalizability  
92 of this method in challenging imaging conditions as well as in freely moving rodents.  
93

94 **Results:**

95 Our method, which we refer to as LEAP (**LEAP Estimates Animal Pose**), consists of three  
96 phases (**Fig. 1a**): *(i) Registration and alignment*, in which raw video of a behaving animal is  
97 preprocessed into egocentric coordinates; *(ii) Labeling and training*, in which the user provides  
98 ground truth labels to train the network to find body part positions in a subset of images; and *(iii)*  
99 *Pose estimation*, in which the network can be applied to new and unlabeled data. In the  
100 following sections, we demonstrate the power of this tool using a previously published data set  
101 of 59 male fruit flies, each recorded for one hour at 100 Hz, for a total of >21 million images<sup>10</sup>.  
102

103 ***The Components of LEAP:***

104 ***(i) Registration and alignment***

105 The first step in our pipeline is to extract the image region that contains the animal within the  
106 field of view of the camera, as well as its angular heading within the image. This can be  
107 accomplished using standard image processing techniques<sup>37,38</sup> or existing software packages  
108<sup>2,13,39,40</sup>. Our implementation<sup>10</sup> is provided in the accompanying code repository. This step  
109 produces egocentric, oriented bounding boxes around each fly image used to train the neural  
110 network. While this step improves pose calculation accuracy as it saves the network from being  
111 required to learn rotational invariance, we note that this can also be learned at the cost of  
112 prediction accuracy (**Supplementary Fig. 1**).  
113

114 ***(ii) Labeling, training, and neural network architecture***

115 The neural network learns to predict body part positions from a set of user-labeled images. To  
116 identify a small set of example ‘training’ images that are representative of the set of poses  
117 across the entire data set, we use a technique we refer to as *cluster sampling*. A simple random  
118 subset of the movie images are grouped via k-means clustering and then these images are

119 sampled uniformly across groups for labeling. The grouping is based on linear correlations  
120 between pixel intensities in the images as a proxy measure for similarity in body pose. The  
121 diversity of poses represented using this method can be observed in the centroids of each of the  
122 clusters identified (**Supplementary Fig. 2**).  
123

124 Poses in each training image are labeled using a custom GUI with draggable body part markers  
125 that form a skeleton (**Fig. 1b**). For the fruit fly, we track four points on each of the six legs, two  
126 points on the wing tips, three points on the thorax and abdomen, and three points on the head  
127 for a total of 32 points in every frame. These points were chosen to align with known *Drosophila*  
128 body joints (**Supplementary Fig. 3**). For every training image, the user drags each skeleton  
129 point to the appropriate body part and the program saves the label positions into a self-  
130 contained file. To enhance the size of the training image set further without the need for hand  
131 labeling more frames, we augment the dataset by applying small random rotations and body-  
132 axis reflections to generate new samples from the labeled data. As the neural network  
133 processes the raw images, the rotated and reflected images add new information that the  
134 network can use during training.  
135

136 We first labeled only 10 images, and used these data to train the neural network and generate  
137 body part position estimates for the remaining images chosen via cluster sampling (see below  
138 for details on network training). When trained with only 10 images for just 15 epochs, estimation  
139 error rates were large (**Supplementary Fig. 4a-b**) but these estimates helped to decrease the  
140 time required to label each subsequent frame. We therefore repeated this procedure of  
141 alternating labeling and initializing via briefly trained network estimates at 50, 100, 250, 500 and  
142 1000 labeled frames, decreasing the time required to label each frame from 2 minutes per frame  
143 for the first 10 frames, to 6 seconds per frame for the last 500 frames (**Supplementary Fig. 4c**).  
144 Labelling 1500 frames required a total of 7 hours of manual labeling and an additional 1.5 hours  
145 of network training (including 6 “fast” and 1 “full” training epochs).  
146

147 The core component of LEAP is a deep convolutional neural network. The network takes as  
148 input a single image of the animal and produces as output a set of confidence maps (probability  
149 distributions) which describe the location of each body part within the input (**Fig. 1c**). The global  
150 maximum in each confidence map represents the network’s prediction of that body part’s  
151 position (**Fig. 1c, insets**). We employ a fully convolutional network architecture. This type of  
152 neural network eschews fully connected layers in lieu of repeated convolutions and pooling

153 steps, which greatly improves training and prediction performance when working in the image  
154 domain<sup>34</sup>.

155

156 We devised a simple 15 layer network architecture that is designed to be fast. The network  
157 consists of two blocks of 3x3x64 convolutions, ReLU nonlinear activation, and 2-strided max  
158 pooling, which is then followed by two blocks of transposed convolutions for upsampling and  
159 additional convolutions for refinement (see **Online Methods, Supplementary Fig. 5a**). Pooling  
160 and downsampling allow us to keep filter sizes fixed and small, minimizing the number of  
161 computations required while allowing both local and global spatial features to be learned and  
162 combined. Recently published architectures for pose estimation follow these same general  
163 principles, but are often much larger and more complex, using skip connections, residual  
164 modules, and stacked version of the hourglass with intermediate supervision<sup>41</sup>. We find that  
165 without these features, our network performs equivalently or better than those architectures  
166 (**Supplementary Fig. 5b**).

167

168 Network training consisted of a series of epochs, during which initially random weights are  
169 updated to minimize the mean-squared-error loss between ground truth and estimated  
170 confidence maps. During each epoch, 50 batches of 32 randomly sampled training images are  
171 augmented with small random rotations or reflections and evaluated for weight updates. Then,  
172 10 batches are sampled and augmented from the held out validation set and used to compute  
173 the validation loss. This loss is used to decrease the learning rate if no significant improvements  
174 occur for multiple epochs, fine-tuning the learning process. An epoch was completed in 60 to 90  
175 seconds on modern GPUs (see **Online Methods**).

176

177 For fast training during the labeling and initialization phase, 10% of the data are held out for  
178 validation and training is concluded after 15 epochs. After 1500 images were labeled, we  
179 proceeded to full training, for which we split the data into training (76.5%), validation (13.5%),  
180 and testing (10%) sets. We train the network for 50 epochs to increase the chance of  
181 convergence and use the held out test set to evaluate the final accuracy. All accuracy measures  
182 reported here were computed from this held out test set.

183

### 184 **(iii) Pose estimation**

185 After amortizing for initialization (loading the network onto the GPU), we find that the network is  
186 able to generate predictions at speeds suitable even for real time processing:  $185 \pm 1.1$  Hz

187 (mean+s.d.) for 192x192 images. Without any further refinement, poses generated by the  
188 network faithfully represented many features of *Drosophila* behavior that have been difficult to  
189 track automatically due to issues of occlusion, e.g., thin body parts, such as the legs, being  
190 occluded by the body or wings (**Fig. 1e, Supplementary Movie 1-3**). For example, we found  
191 that the network was able to continuously and accurately track the motion of all 6 legs during  
192 extended bouts of locomotion (**Fig. 1d, Supplementary Movie 1,2**). In addition, the network  
193 can accurately track bouts of head grooming during which the forelegs are highly occluded by  
194 the head (**Fig. 1e, Supplementary Movie 3**).  
195

#### 196 **Performance of LEAP: Accuracy, speed, and training sample size**

197 We evaluated the accuracy of LEAP after full training with 1,500 labeled images by measuring  
198 error as the Euclidean distance between estimated and ground truth coordinates of each body  
199 part on a held-out test set of 300 frames. We found that the accuracy level depends on the body  
200 part being tracked, with parts that are more often occluded, such as hind legs, resulting in  
201 slightly higher error rates (**Fig. 2a**). Overall, we found that error distances for all body parts were  
202 well below 3 pixels for the vast majority of tested images (**Fig. 2b**). This error is achieved rather  
203 quickly during training, requiring as few as 15 epochs (15-20 minutes of training time) to achieve  
204 approximately 1.97 pixel overall accuracy, and less than 50 epochs (50-75 minutes) for  
205 convergence to 1.63 pixel accuracy with the full training set (**Fig. 2c**). To measure the ground  
206 truth accuracy during the alternating labeling-training phase, we also measured the errors on  
207 the full test set as a function of the number of labeled images used for training under the fast  
208 training regime (15 epochs). We found that with as few as 10 labeled images the network is able  
209 to achieve <2.5 pixel error (2-3% of body length) in 74% of the test set, while 1,000 labeled  
210 images yields an accuracy of <2.5 pixels in 87% of the test set (**Fig. 2d**). This level of accuracy  
211 when training for few epochs with few samples contributes to the drastic reduction in time spent  
212 hand-labeling after fast training (**Supplementary Fig. 4**).  
213

#### 214 **Leg tracking with LEAP recapitulates previously described gait structure**

215 To evaluate the usefulness of our pose estimator for producing experimentally valid  
216 measurements, we used it to analyze the gait dynamics of freely moving flies. Previous work on  
217 *Drosophila* gait relied on imaging systems that use a combination of optical touch sensors and  
218 high speed video recording to follow fly legs as they walk<sup>24</sup>. Although this system can  
219 accurately track fly footprints over a few seconds at a time, it cannot track the limbs when they  
220 are not in contact with the surface (during swing). Other methods to investigate gait dynamics

221 use a semi-automated approach to label fly limbs<sup>26,42</sup>. This requires a large time investment to  
222 manually correct automatically generated predictions, and therefore the semi-automated  
223 approach typically involves smaller datasets.

224

225 We began by evaluating our network on the dataset of 59 adult male fruit flies<sup>10</sup> and extracting  
226 the predicted positions of each leg tip in each of 21 million frames. For every frame in which the  
227 fly was moving forward (7.2 hours/2.6 million frames total), we encoded each leg as either in  
228 swing or stance depending on whether the leg was moving forward or backward relative to the  
229 fly's direction of motion (**Fig. 3a**). Using this encoding, we measured the relationship between  
230 the fly's speed and the duration of stance and swing (**Fig. 3b**). Similar to previous work, we find  
231 that swing duration is relatively constant across walking speeds, whereas stance duration  
232 decreases with walking speed<sup>24</sup>. Because our methods allow us to estimate animal pose during  
233 both stance and swing (versus only during stance<sup>24</sup>), we have the opportunity to investigate the  
234 dynamics of leg motion during the swing phase. We found that swing velocity increases with  
235 body speed, corroborating previous results (**Fig. 3c**). We also found that fly leg velocities follow  
236 a parabolic trajectory parametrized by body speed (**Fig. 3c**).

237

238 Following the work of<sup>42</sup>, we then trained a 3 state Hidden Markov Model (HMM) to capture the  
239 different gait modes exhibited by *Drosophila*. The emission probabilities from the model of the  
240 resulting hidden states were indicative of tripod, tetrapod, and non-canonical/wave gaits (**Fig.**  
241 **3d**). As expected, we observed tripod gait at high body velocities and tetrapod or non-canonical  
242 gaits at intermediate and low velocities, in accordance with previous work<sup>24,42,43</sup> (**Fig. 3e-g**).  
243 These results demonstrate that our pose estimator is able to effectively capture the dynamics of  
244 known complex behaviors, such as locomotion.

245

#### 246 ***Body dynamics reveal structure in the fly behavioral repertoire***

247 We next used the output of LEAP as the first step in an unsupervised analysis of the fly  
248 behavioral repertoire<sup>10</sup>. We calculated the position of each body part relative to the center of the  
249 fly abdomen for each point in time and then computed a spectrogram for each of these  
250 timeseries via the Continuous Wavelet Transform (CWT). We then concatenated these  
251 spectrograms and embedded the resulting feature vectors into a two-dimensional space of  
252 actions we term a behavior space (**Online Methods, Fig. 4a**). As has been shown previously,  
253 the distribution of time points in this space is concentrated into a number of strong peaks that  
254 represent stereotyped behaviors seen across time and in multiple individuals<sup>10</sup>.

255

256 We identify clusters in the behavior space distribution by grouping together regions of high  
257 occupancy and stereotypy (**Fig. 4b**). This distribution is sharper than what we found previously  
258 using a PCA-based compression of the images (**Supplementary Fig. 6**), with many of the least  
259 resolved behaviors now grouped together appropriately. An additional advantage to using pose  
260 estimation over PCA-based image compression is the ability to describe stereotyped behaviors  
261 by the dynamics of each body part. We calculated the average concatenated spectrogram for  
262 each cluster and found that specific behaviors are recapitulated in the motion power spectrum  
263 for each body part (**Fig. 4c-h**).

264

265 This method can be used to accurately describe grooming, a class of behaviors that is highly  
266 represented in our dataset. Posterior grooming behaviors exhibited a distinctly symmetric  
267 topology (**Fig. 4b-g**), revealing both bilateral (**Fig. 4e**) as well as unilateral grooming of the  
268 wings (**Fig. 4c,f**) and the rear of the abdomen (**Fig. 4d,g**). These behaviors involve unilateral,  
269 broadband (1-8 Hz) motion of the hind legs on one side of the body and a slower (~1.5 Hz)  
270 folding of the wing on the same side of the body. In contrast, anterior grooming is characterized  
271 by broadband motions of both front legs with a peak at ~9 Hz, representing the legs rubbing  
272 against each other (**Fig. 4h**).

273

274 We also discovered a number of unique clusters related to locomotion (**Fig 5a,b**). The slowest  
275 state (cluster 10) involves a number of frequencies with a broad peak centered at 5.1 Hz (**Fig. 5**  
276 **c-e**). This can be seen both in the concatenated spectrograms (**Fig. 5c**) and the power  
277 spectrum averaged over all leg positions (**Fig. 5d**). The fly center-of-mass velocity distribution  
278 for this behavior is shown in **Figure 5e**. As the fly speeds up (clusters 10-15, **Fig. 5e**), the peak  
279 frequency for the legs increases monotonically to 11.5 Hz (cluster 15). We next asked if the  
280 tripod and tetrapod gaits we found in our previous analysis were represented by distinct regions  
281 in the behavior space. We found that tripod gait was used predominantly in the three fastest  
282 locomotion behaviors whereas the tetrapod (and to a lesser extent the non-canonical) gait was  
283 used for the three slower locomotion behaviors (**Supplementary Fig. 5f**).

284

#### 285 ***LEAP generalizes to images with complex backgrounds or of other animals***

286 To test the robustness and generalizability of our approach under more varied imaging  
287 conditions, we evaluated the performance of LEAP on a dataset in which pair of flies were  
288 imaged against a non-uniform and low contrast background of porous mesh (~4.2 million

289 frames, ~11.7 hours) (**Fig. 6a<sub>1</sub>**). Using the same workflow as in the first dataset, we found that  
290 the pose estimator was able to reliably recover body part positions with high accuracy despite  
291 poorer illumination and a complex background that was at times indistinguishable from the fly  
292 (**Fig. 6a<sub>2,3</sub>, Supplementary Movie 4**). We then applied a previously described method for  
293 segmentation and tracking <sup>13</sup> to these images to evaluate the performance when masking out  
294 the background (**Fig. 6b<sub>1</sub>**). Even with substantial errors in the masking (e.g., leg or wing  
295 segmentation artifacts), we find that the accuracy remains high and is improved slightly by  
296 excluding the background pixels from the images when compared to the raw images (**Fig. 6b<sub>2,3</sub>**,  
297 **Supplementary Movie 4**). Finally, we tested the applicability of our framework to animals with  
298 different morphology by tracking videos of freely behaving mice (*Mus musculus*) imaged from  
299 below in an open arena (**Fig. 6c<sub>1</sub>**). We observed comparable accuracy in these mice despite  
300 considerable occlusion during behaviors such as rearing (**Fig. 6c<sub>2,3</sub>, Supplementary Movie 5**).  
301

## 302 Discussion

303 Here we present a pipeline (termed LEAP) that uses a deep neural network to track the body  
304 parts of a behaving animal in all frames of a movie via labeling of a small number of images  
305 from across the dataset. We show that this method is fast (requiring one hour to train and  
306 producing body part position estimates at a maximum rate of 185 Hz), accurate (training on 10  
307 frames results in 74% of estimates within 2.5 pixel error while training on 100 frames results in  
308 85% of the frames within 2.5 pixel error), and generalizes across animal species (including flies  
309 and mice) and different regimes of signal to noise ratio. Importantly, we do not construct a single  
310 network to perform pose estimation on all datasets, but rather we present a single architecture  
311 that can be trained to perform pose estimation on any dataset if given a small number of training  
312 samples. All that is required of future users is that the training sets be compiled in a specific  
313 manner that can be facilitated with our user interface (for which we provide code and utilities).  
314

315 Discovering the proximate mechanisms underlying behavior relies on an analysis of behavioral  
316 dynamics matched to the timescales of neural and muscular activity. Tracking only the centroid  
317 of an animal and its change in position or heading over time is likely an insufficient level of  
318 description for determining how the nervous system controls most behaviors. Previous studies  
319 have addressed the issue of pose estimation either through centroid tracking <sup>2</sup>, pixel-wise  
320 correlations <sup>10,11</sup>, or specialized apparatus for tracking body parts <sup>17,20,24,42,44</sup>. For the latter,  
321 applying markers to an animal can limit natural behavior and systems that track particular body  
322 parts are not in general scalable to all body parts or animals with a very different body plan.

323

324 We demonstrate the value of LEAP by showing how it can be applied to the study of locomotor  
325 gait dynamics (**Fig. 3, 5**) and unsupervised behavioral mapping (**Fig. 4, 5**) in *Drosophila*.  
326 Previous studies of gait dynamics have been limited to short stretches of locomotor bouts that  
327 were captured using a specialized imaging system <sup>24</sup> or to the number of behavioral frames that  
328 could be hand-labeled <sup>42</sup>. We show that LEAP not only recapitulates previous findings on  
329 locomotor gait, but that it also discovers new aspects of the behavior (for example, that the  
330 dynamics of the leg during swing have a nonlinear relationship with swing velocity). We also  
331 demonstrate the clear interpretability afforded when using LEAP in combination with  
332 unsupervised behavior classification (**Fig. 4, 5**). This provides a solution to a major shortcoming  
333 in existing approaches, namely that identified behaviors had to be interpreted simply by  
334 watching videos <sup>10,11</sup>. Using LEAP as the first step in such unsupervised algorithms, each  
335 discovered behavior can now be interpreted by analyzing the dynamics of each body part.  
336

337 There are a number of applications for this pipeline beyond those demonstrated here. Because  
338 the network learns body positions from a small amount of human labeled frames, the network  
339 can be easily trained to track a wide variety of animal species and classes of behavior. Further,  
340 LEAP can be extended to tracking of body parts in 3D by either using multiple cameras or  
341 depth-sensitive devices. This will likely be useful for tracking body parts of head-fixed animals  
342 moving on an air supported treadmill <sup>45,46</sup>. These experiments are particularly suited for our  
343 approach, as the movies from head-fixed animals are inherently recorded in egocentric  
344 coordinates. Additionally, we note that the fast prediction performance of our method makes it  
345 compatible with closed-loop experimentation, where joint positions may be computed in realtime  
346 to control experimental parameters such as stimuli presented to the animal or optogenetic  
347 modulation. Lastly, through the addition of a segmentation step for analyzing movies of multiple  
348 animals <sup>2,13,39</sup>, LEAP can estimate poses for multiple interacting individuals.  
349

350 The primary practical limitation of this framework is the egocentric alignment step that may be  
351 sensitive to imaging conditions and the context of the experiment. We note, however, that many  
352 standard techniques exist to find the centroid and orientation of animals in images, including  
353 deep learning-based approaches <sup>40</sup>. Other concerns may pertain to generalizability, in particular  
354 due to how we train each network from scratch rather than performing transfer learning to reuse  
355 a set of more general, shallow layer feature detectors <sup>36</sup>. While transfer learning could easily be  
356 incorporated into LEAP (as well as any other network architecture designed for pose

357 estimation), we found it to be unnecessary given the inherently low variability of imaging  
358 conditions in the lab and the empirically determined low training data requirements.

359

360 In summary, we present a method for tracking body part positions of freely moving animals with  
361 little manual effort and without the use of physical markers. We show LEAP's robustness, state-  
362 of-the-art performance, validity, and utility for quantitative behavioral analysis. We anticipate that  
363 this tool will reduce the technical barriers to addressing a broad range of previously intractable  
364 questions in ethology and neuroscience through quantitative analysis of the dynamic changes in  
365 the full pose of an animal over time.

366

### 367 **Contributions**

368 Designed study: TP, DA, SW, JS, and MM

369 Conducted experiments: TP, DA, LW, and MK

370 Developed GUI and analyzed data: TP and DA

371 Wrote manuscript: TP, DA, JS, and MM

372

### 373 **Acknowledgments**

374 The authors acknowledge Jonathan Pillow for helpful discussions; Gordon Berman and Daniel  
375 Choi for providing fly behavior data; Byung Cheol Cho for contributions to the mouse  
376 experimental setup, acquisition and preprocessing pipeline; Peter Chen for a previous version of  
377 a neural network for pose estimation that was useful in designing our method; Heejae Jang,  
378 Malavika Murugan, and Ilana Witten for feedback on the GUI and other helpful discussions;  
379 Georgia Guan for assistance maintaining flies; and the Murthy, Shaevitz and Wang labs for  
380 general feedback.

381

382 This work was supported by the NIH: R01 NS104899-01 BRAIN Initiative Award (to MM and  
383 JS), R01 MH115750 (to SW and JS), NIH R01 NS045193; NSF BRAIN Initiative EAGER Award  
384 (to MM and JS); the Nancy Lurie Marks Family Foundation (to SW); an HHMI Faculty Scholar  
385 Award (to MM); and an NSF GRFP DGE-1148900 (to TP).

386

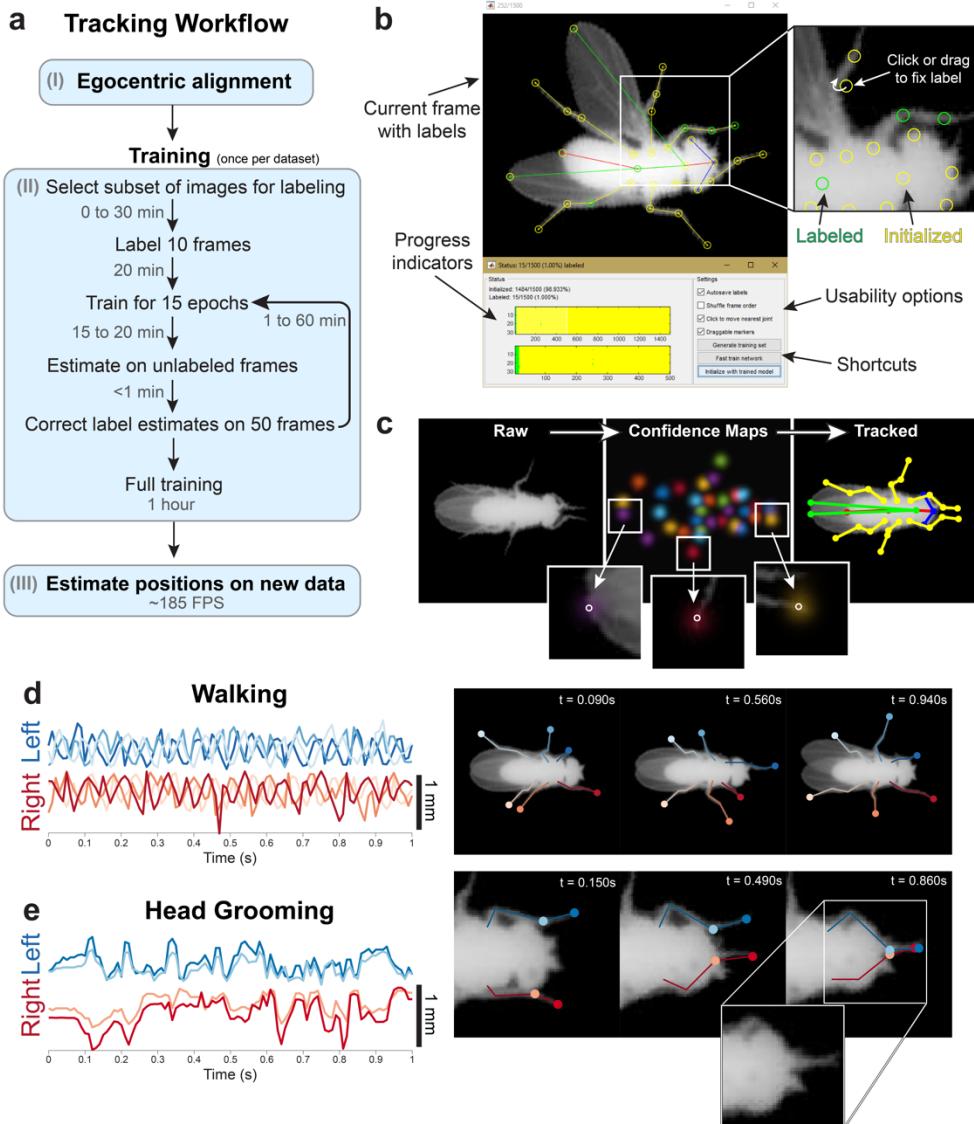
### **References**

1. Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* 84, 18–31 (2014).
2. Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. H. High-throughput

- ethomics in large groups of *Drosophila*. *Nat. Methods* 6, 451–457 (2009).
3. Swierczek, N. A., Giles, A. C., Rankin, C. H. & Kerr, R. A. High-throughput behavioral analysis in *C. elegans*. *Nat. Methods* 8, 592–598 (2011).
  4. Deng, Y., Coen, P., Sun, M. & Shaevitz, J. W. Efficient multiple object tracking using mutually repulsive active membranes. *PLoS One* 8, e65769 (2013).
  5. Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in *Drosophila*. *Nat. Methods* 6, 297–303 (2009).
  6. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* 10, 64–67 (2013).
  7. Arthur, B. J., Sunayama-Morita, T., Coen, P., Murthy, M. & Stern, D. L. Multi-channel acoustic recording and automated analysis of *Drosophila* courtship songs. *BMC Biol.* 11, 11 (2013).
  8. Anderson, S. E., Dave, A. S. & Margoliash, D. Template-based automatic recognition of birdsong syllables from continuous recordings. *J. Acoust. Soc. Am.* 100, 1209–1219 (1996).
  9. Tachibana, R. O., Oosugi, N. & Okanoya, K. Semi-automatic classification of birdsong elements using a linear support vector machine. *PLoS One* 9, e92584 (2014).
  10. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* 11, (2014).
  11. Wiltschko, A. B. et al. Mapping Sub-Second Structure in Mouse Behavior. *Neuron* 88, 1121–1135 (2015).
  12. Berman, G. J., Bialek, W. & Shaevitz, J. W. Predictability and hierarchy in *Drosophila* behavior. *Proc. Natl. Acad. Sci. U. S. A.* 113, 11943–11948 (2016).
  13. Klibaite, U., Berman, G. J., Cande, J., Stern, D. L. & Shaevitz, J. W. An unsupervised method for quantifying the behavior of paired animals. *Phys. Biol.* 14, 015006 (2017).
  14. Wang, Q. et al. The PSI–U1 snRNP interaction regulates male mating behavior in *Drosophila*. *Proc. Natl. Acad. Sci. U. S. A.* 113, 5269–5274 (2016).
  15. Vogelstein, J. T. et al. Discovery of brainwide neural-behavioral maps via multiscale unsupervised structure learning. *Science* 344, 386–392 (2014).
  16. Cande, J. et al. Optogenetic dissection of descending behavioral control in *Drosophila*. bioRxiv doi: 10.1101/230128 230128 (2018).
  17. Kain, J. et al. Leg-tracking and automated behavioural classification in *Drosophila*. *Nat. Commun.* 4, 1910 (2013).
  18. Machado, A. S., Darmohray, D. M., Fayad, J., Marques, H. G. & Carey, M. R. A quantitative framework for whole-body coordination reveals specific deficits in freely walking ataxic mice. *Elife* 4, (2015).

19. Nashaat, M. A. et al. Pixying Behavior: A Versatile Real-Time and Post Hoc Automated Optical Tracking Method for Freely Moving and Head Fixed Animals. *eNeuro* 4, (2017).
20. Nanjappa, A. et al. Mouse Pose Estimation From Depth Images. *arXiv:1511.07611* (2015).
21. Nakamura, A. et al. Low-cost three-dimensional gait analysis system for mice with an infrared depth sensor. *Neurosci. Res.* 100, 55–62 (2015).
22. Wang, Z., Mirbozorgi, S. A. & Ghovanloo, M. An automated behavior analysis system for freely moving rodents using depth image. *Med. Biol. Eng. Comput.* (2018).:
23. Mendes, C. S. et al. Quantification of gait parameters in freely walking rodents. *BMC Biol.* 13, 50 (2015).
24. Mendes, C. S., Bartos, I., Akay, T., Márka, S. & Mann, R. S. Quantification of gait parameters in freely walking wild type and sensory deprived *Drosophila melanogaster*. *Elife* 2, e00231 (2013).
25. Petrou, G. & Webb, B. Detailed tracking of body and leg movements of a freely walking female cricket during phonotaxis. *J. Neurosci. Methods* 203, 56–68 (2012).
26. Uhlmann, V., Ramdya, P., Delgado-Gonzalo, R., Benton, R. & Unser, M. FlyLimbTracker: An active contour based approach for leg segment tracking in unmarked, freely behaving *Drosophila*. *PLoS One* 12, e0173433 (2017).
27. Toshev, A. & Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. *arXiv:1312.4659* (2013).
28. Tompson, J. J., Jain, A., LeCun, Y. & Bregler, C. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. in *Advances in Neural Information Processing Systems* 27 (eds. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q.) 1799–1807 (Curran Associates, Inc., 2014).
29. Carreira, J., Agrawal, P., Fragkiadaki, K. & Malik, J. Human Pose Estimation with Iterative Error Feedback. *arXiv:1507.06550* (2015).
30. Wei, S.-E., Ramakrishna, V., Kanade, T. & Sheikh, Y. Convolutional Pose Machines. *arXiv:1602.00134* (2016).
31. Bulat, A. & Tzimiropoulos, G. Human pose estimation via Convolutional Part Heatmap Regression. *arXiv:1609.01743* (2016).
32. Cao, Z., Simon, T., Wei, S.-E. & Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv:1611.08050* (2016).
33. Tome, D., Russell, C. & Agapito, L. Lifting from the Deep: Convolutional 3D Pose Estimation from a Single Image. *arXiv:1701.00295* (2017).
34. Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic

- segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition 3431–3440 (cv-foundation.org, 2015).
35. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 234–241 (Springer International Publishing, 2015).
36. Mathis, A. et al. Markerless tracking of user-defined features with deep learning. arXiv:1804.03142 (2018).
37. Taubin, G. Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 1115–1138 (1991).
38. Fitzgibbon, A., Pilu, M. & Fisher, R. B. Direct Least Square Fitting of Ellipses. *IEEE Trans. Pattern Anal. Mach. Intell.* 21, 476–480 (1999).
39. Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S. & de Polavieja, G. G. idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nat. Methods* 11, 743–748 (2014).
40. Romero-Ferrero, F., Bergomi, M. G., Hinz, R., Heras, F. J. H. & de Polavieja, G. G. idtracker.ai: Tracking all individuals in large collectives of unmarked animals. arXiv:1803.04351 (2018).
41. Newell, A., Yang, K. & Deng, J. Stacked Hourglass Networks for Human Pose Estimation. arXiv:1603.06937 (2016).
42. Isakov, A. et al. Recovery of locomotion after injury in *Drosophila melanogaster* depends on proprioception. *J. Exp. Biol.* 219, 1760–1771 (2016).
43. Wosnitza, A., Bockemühl, T., Dübbert, M., Scholz, H. & Büschges, A. Inter-leg coordination in the control of walking speed in *Drosophila*. *J. Exp. Biol.* 216, 480–491 (2013).
44. Qiao, B., Li, C., Allen, V. W., Shirasu-Hiza, M. & Syed, S. Automated analysis of long-term grooming behavior in using a -nearest neighbors classifier. *Elife* 7, (2018).
45. Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L. & Tank, D. W. Imaging large-scale neural activity with cellular resolution in awake, mobile mice. *Neuron* 56, 43–57 (2007).
46. Seelig, J. D. & Jayaraman, V. Neural dynamics for landmark orientation and angular path integration. *Nature* 521, 186–191 (2015).



387

388

389 **Figure 1: Body part tracking via LEAP, a deep learning framework for animal pose  
estimation**

391 (a) Overview of the tracking workflow. In the initial preprocessing phase (I), video frames are  
392 centered relative to the animal to render the images in egocentric coordinates. In the beginning  
393 of the training phase (II), representative frames are sampled. After labeling an initial set of  
394 images, the neural network is trained and used to estimate body positions on the remaining  
395 images of the training set to facilitate subsequent correction of labels. Correcting labels takes  
396 progressively less time as the network is trained with increasingly more labeled examples. Once  
397 all training images are labeled, full training involves fine tuning the network to optimize

398 performance. Once trained (III), estimation on new, unlabeled data is fully automated and can  
399 be performed at high speed on a GPU.

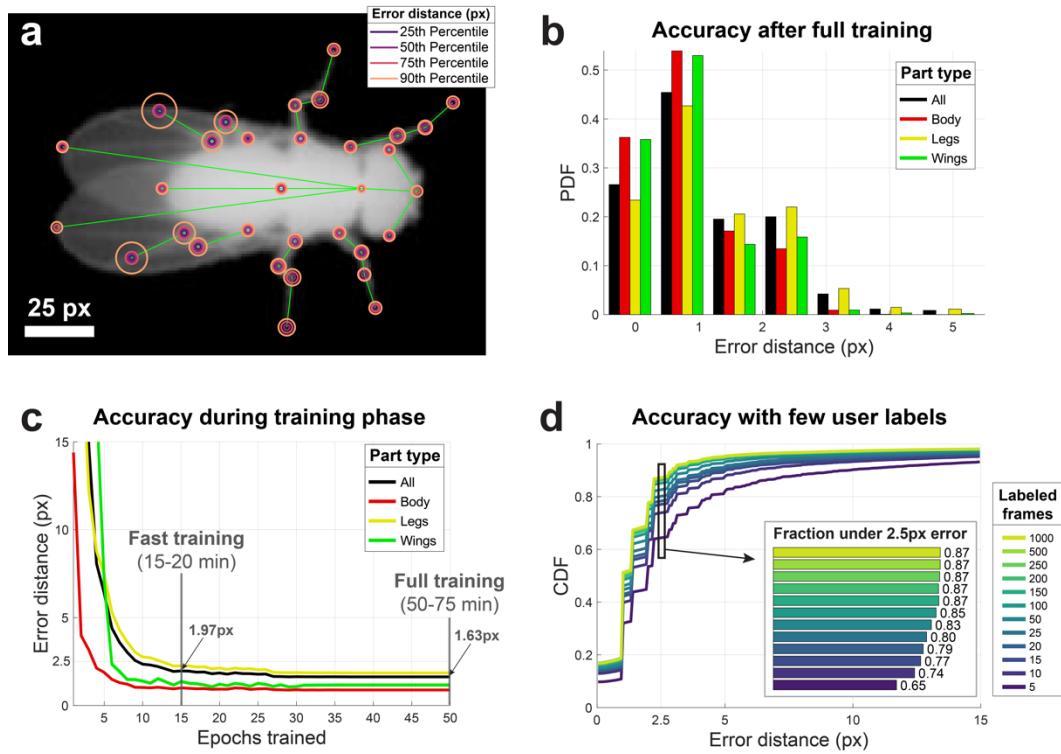
400 (b) Graphical user interface for providing ground truth labels and correcting estimates. The  
401 software displays images in the training set with interactive markers denoting the default or best  
402 estimate for each body part (top-left). User input is provided by clicking or dragging the markers  
403 to the correct location (top-right). Colors indicate labeling progress and denote whether the  
404 marker is at the ground truth location (green) or is an estimate from network initialization  
405 (yellow). Progress indicators mark which frames and body parts have been labeled thus far,  
406 while shortcut buttons enable the user to export the labels to use a trained network to initialize  
407 unlabeled body parts with automated estimates.

408 (c) Data flow through the LEAP pipeline. Raw images are provided as input without markers or  
409 indicators (left). For each input image, the network outputs a stack of confidence maps, a max  
410 projection through which is used here for visualization (middle). Insets overlay individual  
411 confidence maps on the image to reveal how confidence density is centered on each body part,  
412 with the peak indicated by a circle. The predicted coordinate for each body part is the peak  
413 value in each confidence map, enabling a visualization of the tracked skeleton (right).

414 (d) Walking behavior can be quantitatively described by leg tip trajectories. The distance of each  
415 of the 6 leg tips from its own mean position during a walking bout reveals a cyclic pattern of leg  
416 movements (left). The tracked points on the images span a diversity of poses that change over  
417 fast timescales (right).

418 (e) Head grooming behavior can also be quantitatively described by leg tip trajectories. Position  
419 estimates are not confounded by occlusions when the legs pass under the head (right, inset).

420



421  
422

423 **Figure 2: LEAP is highly accurate, and requires little training or labeled data**

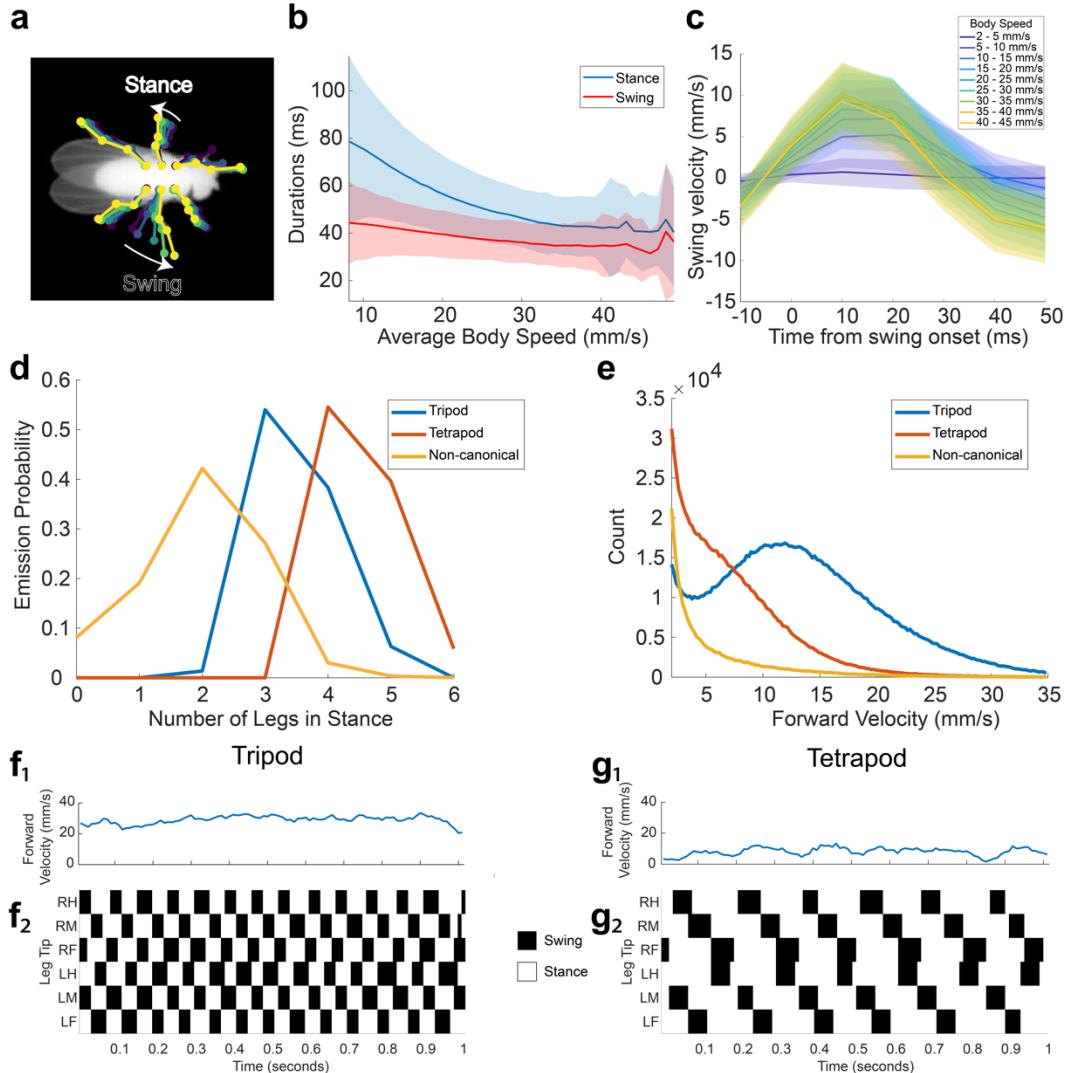
424 (a) Part-wise accuracy distribution after full training. Circles are plotted on a reference image to  
425 indicate the fraction of held out testing data ( $n = 300$  images) for which estimated positions of  
426 the particular body part are closer to the ground truth than the radii. Most body parts have error  
427 rates below 3 pixels for over 90% of tested images. Body parts that often suffer from occlusion  
428 (e.g., hind legs) have higher rates of error.  
429 (b) Accuracy summary on held out test set after full training. Both total and grouped error rates  
430 fall well below 3 pixels (1/64th of 192x192 pixel images) in terms of Euclidean distance to  
431 ground truth as in (a).  
432 (c) Accuracy as a function of training time demonstrates fast convergence and time/accuracy  
433 trade-off during training. In the “fast training” regime, the training procedure runs for only 15  
434 epochs, allowing the network to approximate convergence-level accuracy in a fraction of the  
435 time, optimal for training for initialization with few samples. For these tests,  $n = 1215$  labeled  
436 frames were used for training. Lines and shaded area indicate mean and SEM for all held out  
437 test images pooled over 5 runs. After 50 epochs, convergence is achieved at the cost of

438 additional run time. Run times depend mainly on the performance of the hardware being used,  
439 with a range provided by estimates from high end consumer or enterprise GPUs.

440 (d) Accuracy as a function of number of training examples demonstrates the trade-off between  
441 estimation accuracy and time spent labeling. Distributions indicate estimation errors in a held  
442 out test set ( $n = 300$  frames) while varying the number of labeled images used for training,  
443 pooled over 5 “fast training” runs. Using as few as 10 labeled images, 74% of body part  
444 estimates fell within 2.5 pixels of their ground truth locations, increasing to 87% with 1000  
445 labeled images (inset).

446

447



448

449

450 **Figure 3: LEAP recapitulates known gait patterning in flies**

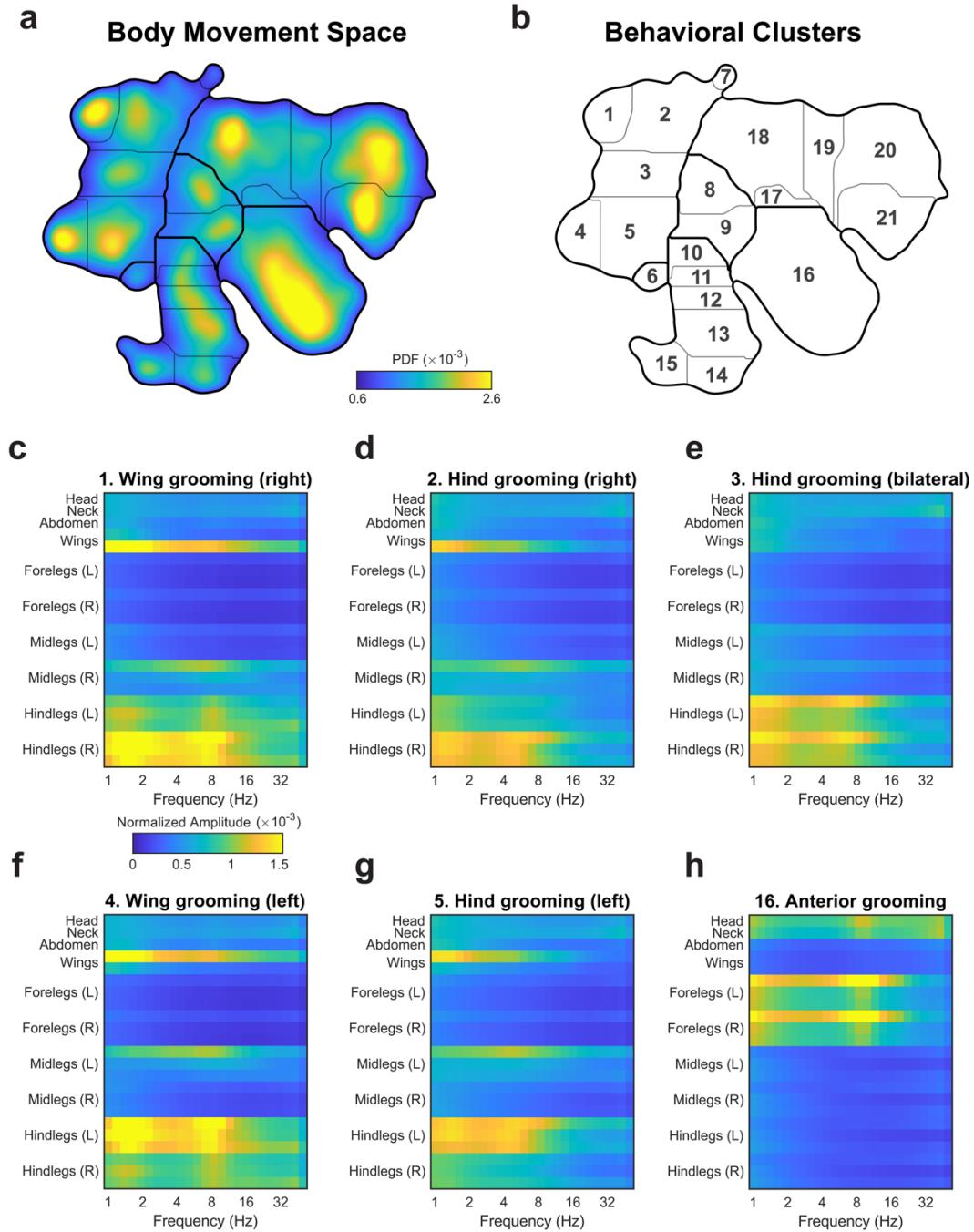
- 451 (a) Schematic of swing and stance encoding.
- 452 (b) Duration of swing and stance as a function of average body speed. Stance duration
- 453 decreases with increasing body speed, corroborating previous findings (Mendes et al. 2013).
- 454 This data comprises approximately 7.2 hours in which the fly is moving forward (2.6 million
- 455 frames). Shaded regions indicate one standard deviation.
- 456 (c) Swing velocity as a function of time from swing onset, and binned by body speed. Shaded
- 457 regions indicate one standard deviation.
- 458 (d) Emission probabilities of numbers of legs in stance for each hidden state in the HMM (see
- 459 Methods). Hidden state emissions resemble tripod, tetrapod, and non-canonical gaits.

460 (e) Distributions of velocities for each hidden state. Flies primarily exhibit tripod gait at high  
461 velocities, and tetrapod or non-canonical gaits at intermediate and slow velocities.

462 (f,g) Examples of tripod and tetrapod gaits identified by the HMM.

463

464



465

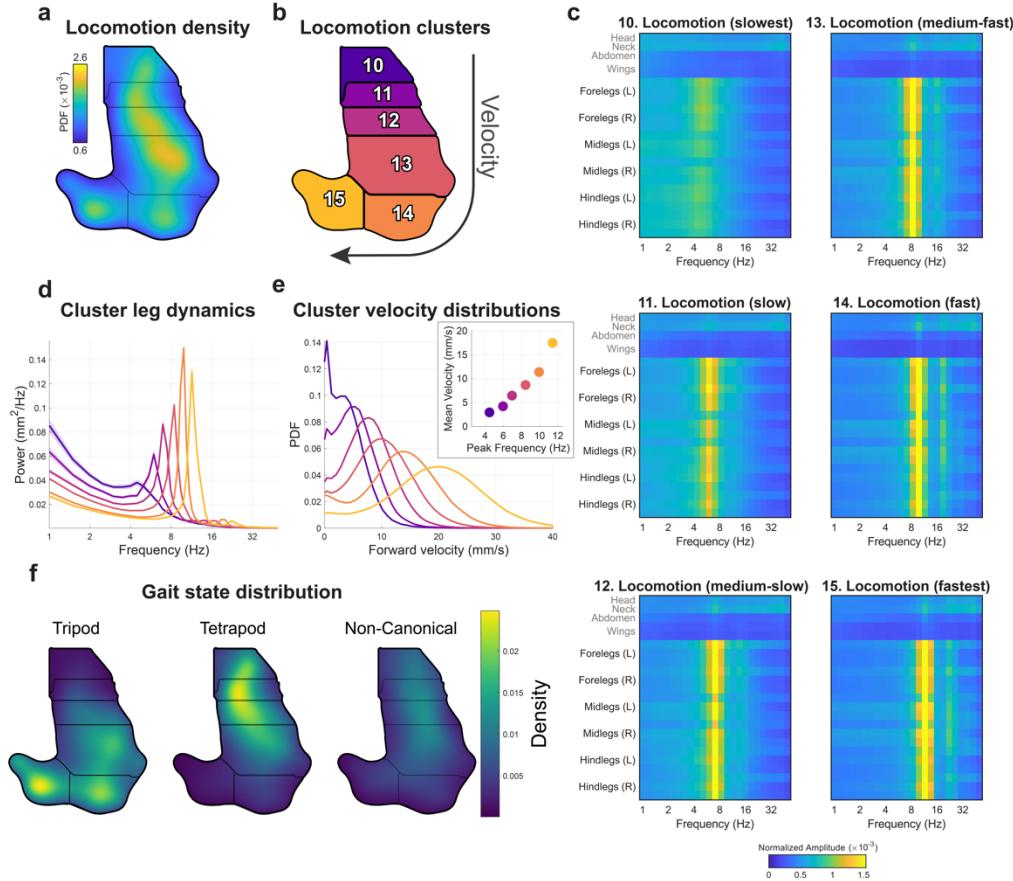
466

467 **Figure 4: Unsupervised embedding of body position dynamics**

468 (a) Density of freely moving fly body part trajectories, after projecting their spectrograms into to  
469 two dimensions via unsupervised nonlinear manifold embedding (Berman et al., 2014). The  
470 distribution shown is generated from 21.1 million frames. Regions in the space with higher  
471 density correspond to stereotyped movement patterns, whereas low density regions form

472 natural divisions between distinct dynamics. A watershed algorithm is used to separate the  
473 peaks in the probability distribution (see Methods).  
474 (b) Cluster boundaries from (a) with cluster numbers indicated.  
475 (c-h) Average spectrograms from time points that fall within the dominant grooming clusters;  
476 cluster numbers are indicated in (b). Posterior grooming behaviors subdivide into symmetric  
477 clusters corresponding to the lateralization of limbs employed (c-g). Qualitative labels for each  
478 cluster based on visual inspection are provided for convenience. Colormap corresponds to  
479 normalized power for each body part.

480



481

482

483 **Figure 5: Locomotor clusters in behavior space separate distinct gait modes.**

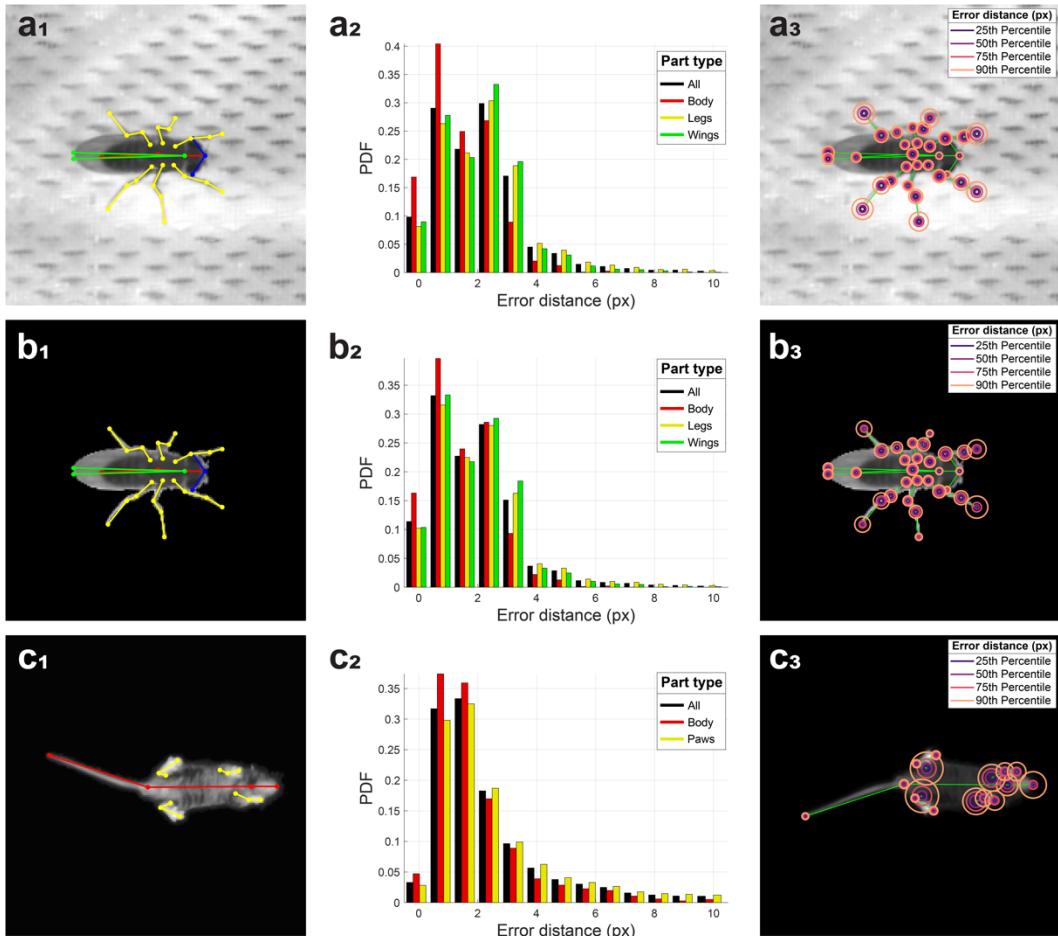
484 (a, b) Density and cluster labels of locomotion clusters (from the same behavioral space shown  
485 in Fig. 4a).

486 (c) Average spectrograms (similar to Fig. 4c-h) quantify the dynamics in each cluster. The  
487 frequency spectrum of leg movements in each cluster is sharp and shifts from 5.1 to 11.5 Hz  
488 from slowest to fastest locomotion speeds.

489 (d) Average power spectra calculated from the leg joint positions for each cluster in (c). Colors  
490 correspond to the cluster numbers in (b). Each spectrum has a single dominant peak between  
491 5.1 and 11.5 Hz, with harmonics from 12-25 Hz seen in the fastest subtypes.

492 (e) The distribution of forward locomotion velocity exhibits a peak that shifts to the right as a  
493 function of cluster number. Colors correspond to cluster numbers in (b). (inset) Forward  
494 locomotion velocity increases with peak leg frequency.

495 (f) Gait modes identified by HMM from swing/stance state correspond to distinct clusters.



496

497

498 **Figure 6: LEAP generalizes to images with complex backgrounds or of other animals**

499 (a) LEAP estimates on a separate dataset of 42 freely moving male flies, each imaged against a  
500 heterogeneous background of mesh and microphones, with side illumination (~4.2 million  
501 frames, ~11.7 hours). 32 body parts (see Supp Fig. 3) were tracked (a<sub>1</sub>), and 1,530 labeled  
502 frames were used for training. Error rates for position estimates were calculated on a held out  
503 test set of 400 frames (a<sub>2</sub>) and were comparable to those achieved for images with higher signal  
504 to noise (compare with Fig. 2b). Part-wise error distances (a<sub>3</sub>) illustrate that accuracy is lower in  
505 distal body parts, likely due to ambiguity with the background mesh holes.

506 (b) LEAP estimates on masked images from the dataset described in (a). Background was  
507 subtracted using standard image processing algorithms (see Methods) to reduce the effect of  
508 background artifacts. Similar accuracy measures are observed (compare b<sub>2</sub> with a<sub>2</sub>). Error  
509 distances are higher for distal body parts that are often masked out due to the difficulty in  
510 resolving those pixels from the background (b<sub>3</sub>).

511 (c) LEAP estimates on a dataset of freely moving mice imaged from below (~3 million frames,  
512 ~4.8 hours). Three points are tracked per leg, in addition to the tip of the snout, neck, and base  
513 and tip of the tail ( $c_1$ ) - 1000 labeled frames were used for training. Accuracy rates on a held out  
514 test set (of 242 frames) are higher but still comparable to fly datasets ( $c_2$ ). Most errors come  
515 from the leg base point, which is often occluded ( $c_3$ ).

## Methods

**Title:** Fast animal pose estimation using deep neural networks

**Authors:**

Pereira, T.<sup>1,#</sup>, Aldarondo, D.<sup>1,#</sup>, Willmore, L.<sup>1</sup>, Kislin, M.<sup>1</sup>, Wang, S. S.<sup>1,2</sup>, Murthy, M.<sup>1,2\*</sup>, and Shaevitz, J. W.<sup>1,3,4\*</sup>

1 Princeton Neuroscience Institute, Princeton University

2 Department of Molecular Biology, Princeton University

3 Lewis-Sigler Institute for Integrative Genomics, Princeton University

4 Department of Physics, Princeton University

#equal authors

\*lead contacts and co-corresponding authors: Mala Murthy ([mmurthy@princeton.edu](mailto:mmurthy@princeton.edu)) and Joshua W. Shaevitz ([shaevitz@princeton.edu](mailto:shaevitz@princeton.edu))

1   **Code:** The code for running LEAP can be found in the following repository:

2   <https://github.com/talmo/leap>

3

4   **Datasets:** Details on the dataset of 59 adult male *Drosophila* can be found in<sup>1,2</sup>. Animals  
5   were allowed to move freely in a backlit 100mm diameter circular arena covered by a 2mm tall  
6   clear PETG dome. Videos were captured from the top with a Point Grey Gazelle camera at a  
7   resolution of ~35 pixels/mm at 100 FPS for 1 hour for each fly, totaling ~21 million frames for the  
8   dataset. To calculate the spatial resolution of LEAP we assumed a mean male fly length of  
9   2.82mm<sup>3</sup>.

10

11   The second fly dataset reported here (**Fig. 5**) consists of 42 videos of freely moving pairs of  
12   virgin male and female fruit flies (NM91 strain), 3-5 days post-eclosion. Only males from these  
13   videos were analyzed in this study. Flies moved freely within a 30mm diameter circular arena  
14   with a 2mm tall clear PETG dome against a white mesh floor covering an array of microphones,  
15   resulting in an inhomogeneous image background. Videos were captured from above using a  
16   Point Grey Flea3 camera at a resolution of ~25 pixels/mm at 100 FPS, totaling ~4.2 million  
17   frames.

18

19 The mouse dataset for **Figure 5** consisted of 29 videos of C57/BL6 strain mice (*Mus musculus*),  
20 15 weeks (108 days) old. Animals moved freely in a 45.7x45.7 cm open field arena with a clear  
21 acrylic floor for 10 minutes each. Videos were captured from below with IR illumination using a  
22 Point Grey Blackfly S camera at a resolution of 1.95 pixels/mm at 170 FPS, totaling ~3 million  
23 frames. Experimental procedures were approved by the Princeton University Institutional Animal  
24 Care and Use Committee and conducted in accordance to the National Institutes of Health  
25 guidelines for the humane care and use of laboratory animals. Mice used in this study were  
26 ordered through Jackson Laboratory (The Jackson Laboratory, Bar Harbor, ME) and had at  
27 least one week of acclimation to the Princeton Neuroscience Institute vivarium before  
28 experimental procedures were performed. Mice were kept in group cages with food and water  
29 ad libitum under a reversed 12:12 hour dark-light cycle (light: 19:30-7:30).

30

31 **Preprocessing and alignment to generate egocentric images for labeling and training in**  
32 **LEAP:** For the main fly dataset (59 males), we used the alignment algorithm from <sup>1</sup>. The raw  
33 videos consisted of unoriented bounding boxes around the flies from a closed-loop camera  
34 tracking system. Individual frames were then aligned to a template image of an oriented fly by  
35 matching the peak of the Radon transformed fly image to recover the orientation and then  
36 computing the cross correlation to center the fly. The centroid and orientation parameters were  
37 used to crop a 200x200 pixel oriented bounding box in each frame. Code for alignment is  
38 available in the repository accompanying the original paper:

39 <https://github.com/gordonberman/MotionMapper>

40

41 For the second fly dataset (42 males), we adapted a previously published method for tracking  
42 and segmentation of videos of courting fruit flies <sup>4</sup>. We first modeled the mesh background of the  
43 images by fitting a normal distribution to each pixel in the frame across time with a constant  
44 variance to account for camera shot noise. The posterior was evaluated at each pixel of each  
45 frame and then thresholded to segment the foreground pixels. Due to the inhomogeneity of the  
46 arena floor mesh, significant segmentation artifacts were introduced, particularly when  
47 translucent or very thin body parts (i.e., wings and legs) could not be disambiguated from the  
48 dark background mesh holes. The subsequent steps of histogram thresholding, morphological  
49 filtering and ellipse fitting were performed as described previously in <sup>4</sup>. We developed a simple  
50 GUI for proofreading the automated ellipse tracking before extracting 200 x 200 pixel oriented  
51 bounding boxes. We extracted bounding boxes for both animals in each frame and saved both  
52 the raw pixels containing the background mesh as well as the foreground-only images which

53 contain segmentation artifacts. This pipeline was implemented in MATLAB and the code is  
54 available in the code repository accompanying this paper.

55

56 For the mouse videos, a separate preprocessing pipeline was developed. Raw videos were  
57 processed in three stages: (1) animal tracking, (2) segmentation from background, and (3)  
58 alignment to the body centroid and tail-body interface. In stage (1), the mouse's torso centroid  
59 was tracked by subtracting a background image (median calculated at each pixel value across  
60 that video), retrieving pixels with a brightness above a chosen threshold from background (mice  
61 were brighter than background), and using morphological opening to eliminate noise and the  
62 mouse's appendages. The largest contiguous region reliably captured the mouse's torso  
63 (referred to below as the torso mask) and was used to fit an ellipse whose center was used to  
64 approximate the center of the animal. In stage (2), a similar procedure as in stage (1) was  
65 employed to retrieve a full body mask. In this stage, a more permissive threshold and smaller  
66 morphological opening radius were used than in stage (1) to capture the mouse's body edges,  
67 limbs, and tail while still eliminating noise. The pixels outside of this body mask were set to 0. In  
68 stage (3) each segmented video frame was translated and rotated such that frame's center  
69 coincided with the center of the animal and the x-axis lay on the line connecting the center and  
70 tail-body attachment point. The tail-body attachment point was defined as the center of a region  
71 overlapping between the torso mask and a dilated tail mask. The tail mask was defined as the  
72 largest region remaining after subtracting the torso mask from the full body mask and  
73 performing a morphological opening. After applying these masks to segment the raw images,  
74 bounding boxes were extracted by using the ellipse center and orientation.

75

76 Oriented bounding boxes were cropped to 192 x 192 pixels for all datasets to ensure  
77 consistency in output image size after repeated pooling and upsampling steps in the neural  
78 network. These data were stored in self-describing HDF5 files.

79

80 ***Sampling diverse images for labeling and training in LEAP:*** To ensure diversity in image  
81 and pose space when operating at low sample sizes, we employ a multistage cluster sampling  
82 technique. First,  $n_0$  images were sampled uniformly from each dataset by using a fixed stride  
83 over time to minimize correlations being temporally adjacent samples. We then used principal  
84 component analysis (PCA) to reduce their dimensionality, and the images were then projected  
85 down to the first  $D$  principal components. After dimensionality reduction, the images were  
86 grouped via k-means clustering into  $k$  subgroups from which  $n$  images were randomly sampled

87 from each group. To minimize the time necessary for the network to generalize to images from  
88 all groups, we sorted the dataset such that consecutive samples cycled through the groups.  
89 This way, uniform sampling was maintained even at the early phases of user labeling, ensuring  
90 that even a network trained on only the first few images will be optimized to estimate body part  
91 positions for a diversity of poses. We used  $n_0 = 500$ , yielding 29,500 initial samples;  $D = 50$ ,  
92 which is sufficient to explain 80% of the variance in the data (**Supplementary Fig. 2**);  $k =$   
93 10 and  $n = 150$  to produce a final dataset of 1,500 frames for labeling and training.

94

95 **LEAP neural network design and implementation:** We based our network architecture on  
96 previous designs of neural networks for human pose estimation<sup>5–7</sup>. We adopt a fully  
97 convolutional architecture that learns a mapping from raw images to a set of confidence maps.  
98 These maps are images that can be interpreted as the 2-d probability distribution (i.e., heatmap)  
99 centered at the spatial coordinates of each body part within the image. We train the network to  
100 output one confidence map per body part stacked along the channel axis.

101

102 Our network consists of 15 layers of repeated convolutions and pooling (**Supplementary Fig.**  
103 **4**). The convolution block consists of 3x convolution layers (64 filters, 3x3 kernel size, 1x1 stride,  
104 ReLU activation). The full network consists of 1x convolution block, 1x max pooling across  
105 channels (2x2 pooling size, 2x2 stride), 1x convolution block (128 filters), 1x max pooling (2x2  
106 pooling size, 2x2 stride), 1x convolution block (256 filters), 1x transposed convolution (128  
107 filters, 3x3 kernel size, 2x2 stride, ReLU activation, Glorot normal initialization), 2x convolution  
108 (128 filters, 3x3 kernel size, 1x1 stride, ReLU activation), and 1x transposed convolution (128  
109 filters, 3x3 kernel size, 2x2 stride, linear activation, Glorot normal initialization).

110

111 We base our decisions of these hyperparameters on the idea that repeated convolutions and  
112 strided max pooling enable the network to learn feature detectors across spatial scales. This  
113 allows the network to learn how to estimate confidence maps using global image structure  
114 which provides contextual information that can be used to improve estimates even for occluded  
115 parts<sup>5,7</sup>. Despite the loss of resolution from pooling, the upsampling learned through transposed  
116 convolutions is sufficient to recover the spatial precision in the confidence maps. We do not  
117 employ skip connections, residual modules, stacked networks, regression networks, or affinity  
118 fields in our architecture as used in other approaches of human pose estimation<sup>5,6,8,9</sup>.

119

120 For comparison, we also implemented the stacked hourglass network<sup>7</sup>. We tested both the  
121 single hourglass version and 2x stacked hourglass with intermediate supervision. The hourglass  
122 network consisted of 4x residual bottleneck modules (64 output filters) with max pooling (2x2  
123 pool, 2x2 stride), followed by their symmetric upsampling blocks and respective skip  
124 connections. The stacked version adds intermediate supervision in the form of a loss term on  
125 the output of the first network in addition to the final output.

126

127 We implemented all versions of neural networks in Python via Keras and TensorFlow, popular  
128 deep learning packages that allow transparent GPU acceleration and easy portability across  
129 operating systems and platforms. All Python code was written for Python 3.6.4. Required  
130 libraries were installed via the pip package manager: numpy (1.14.1), h5py (2.7.1), tensorflow-  
131 gpu (1.6.0), keras (2.1.4). We tested our code on machines running either Windows 10 (v1709)  
132 and a RedHat-based Linux distribution (Springdale 7.4) with no additional steps required to port  
133 the software other than installing the required libraries.

134

135 Code for all network implementations is available in the main repository accompanying this  
136 paper.

137

138 **LEAP training procedure:** Prior to training, we generated an augmented dataset from the user-  
139 provided labels and corresponding images. We first doubled the number of images by mirroring  
140 the images along the body symmetric axis and adjusting the body part coordinates accordingly,  
141 including swapping left/right body part labels (e.g., legs). Then, we generated confidence maps  
142 for each body part in each image by rendering the 2-d Gaussian probability distribution centered  
143 at the ground truth body part coordinates,  $\mu = (x, y)$ , and fixed covariance,  $\Sigma = \text{diag}(\sigma)$  with a  
144 constant  $\sigma = 5px$ . These were pre-generated and cached to disk to minimize the necessary  
145 processing time during training.

146

147 Once confidence maps were computed for each image, we split the dataset into training,  
148 validation and test sets. The training set was used for backpropagation of the loss for updating  
149 network weights, the validation set was used to estimate performance and adjust the learning  
150 rate over epochs, and the test set was held out for analysis. For the fast training, the dataset  
151 was split into only training (90%) and validation (10%) sets to make the best use of data when  
152 training with very few labels. For full training, the dataset was split into training (76.5%),

153 validation (13.5%) and testing (10%) sets. All analyses reported here share the same held out  
154 test set to ensure it is never trained against for any replicate.

155

156 All training was done using the Adam optimizer with default parameters as described in the  
157 original paper <sup>10</sup>. We started with a learning rate of 1e-3 but use a scheduler to reduce it by a  
158 factor of 0.1 when the validation loss fails to improve by a minimum threshold of 1e-5 for 3  
159 epochs. The loss function optimized against is simply the mean squared error between  
160 estimated and ground truth confidence maps.

161

162 During training, we considered an epoch to be a set of 50 batches of 32 images, which were  
163 sampled randomly with replacement from the training set and augmented by applying a random  
164 rotation to the input image and the corresponding ground truth confidence maps. At the end of  
165 50 batches of training, 10 batches were sampled from the separate validation set, augmented  
166 and evaluated and the loss was used for learning rate scheduling. An epoch evaluated in 60 to  
167 90 seconds including all augmentation, forward and reverse passes, and the validation forward  
168 pass when running on a modern GPU (NVIDIA GeForce GTX 1080 Ti or P100).

169

170 We ran this entire procedure for 15 epochs during the fast training stage, and for 50 epochs  
171 during the full training stage. For analyses, a minimum of 5 replicates were fully trained on each  
172 dataset to estimate the stability of optimization convergence.

173

174 **Pose estimation from confidence maps:** Predictions of body part positions were computed  
175 directly on the GPU. We implement a channel-wise global maximum operation to convert the  
176 confidence maps into image coordinates as a TensorFlow function, further improving runtime  
177 prediction performance by avoiding the costly transfer of large confidence map arrays. All  
178 prediction functions including normalization and saving were implemented as a self-contained  
179 Python script with a command-line interface for ease of batch processing.

180

181 **Computing hardware:** All performance tests were conducted on a high end consumer-grade  
182 workstation equipped with a Intel Core i7-5960X CPU, 128 GB DDR4 RAM, NVMe SSD drives,  
183 and a single NVIDIA GeForce 1080 GTX Ti (12 GB) GPU. We also use Princeton University's  
184 High Performance Computing cluster with nodes equipped with NVIDIA P100 GPUs for batch  
185 processing. These higher end cards afford a speed-up of ~1.5x in the training phase.

186

187 **Accuracy analysis:** For all analyses of accuracy (**Figs. 2, 6; Supplementary Figs. 4, 5**), we  
188 trained at least 5 replicates of the network with the same training/validation/testing datasets. All  
189 analyses were performed in MATLAB R2018a (MathWorks). We used the gramm toolbox for  
190 figure plotting <sup>11</sup>.

191

192 **Gait analysis:** We translated the body position coordinates to egocentric coordinates by  
193 subtracting the predicted location of the intersection between the thorax and abdomen from all  
194 other body position predictions for each frame. We then calculated the instantaneous velocity  
195 along the rostrocaudal axis of each leg tip within these truly egocentric reference coordinates.  
196 The speed of each body part was smoothed using a Gaussian filter with a five frame moving  
197 window. For each leg tip, instances in which the smoothed velocity was greater than zero were  
198 defined as swing while those less than zero were defined as stance. Information from this  
199 egocentric axis was combined with allocentric tracking data to incorporate speed and orientation  
200 information. The centroids and orientations of the flies were smoothed using a moving mean  
201 filter with a five frame window to find the instantaneous speed and forward velocity. To remove  
202 idle bouts and instances of backward walking, all gait analyses were limited to times when the  
203 fly was moving in the forward direction at a velocity greater than 2 mm/s (approximately one  
204 body length/s) unless otherwise noted. The analyses relating stance and swing duration to body  
205 velocity were limited to forward velocities greater than 7.2 mm/s, to remain in line with previous  
206 work <sup>12</sup>.

207

208 To measure gait modes, we trained an HMM to model gait as described previously <sup>13</sup>. The  
209 training data consisted of a vector denoting the number of legs in stance for bouts in which the  
210 fly was moving forward at a velocity greater than 2 mm/s lasting longer than 0.5 seconds.

211 Training data were sampled such that up to 3,000 frames were taken from each video, resulting  
212 in a total of 159,270 frames. We trained a three-state HMM using the Baum-Welch algorithm  
213 and randomly initialized transition and emission probabilities <sup>14</sup>. We designated each hidden  
214 state as tripod, tetrapod, and non-canonical in accordance with the estimated emission  
215 probabilities. We then used the Viterbi algorithm along with our estimated transition and  
216 emission matrices to predict the most likely sequence of hidden states from which the observed  
217 stance vectors for the entire dataset would emerge <sup>15</sup>.

218

219 **Unsupervised embedding of body part dynamics:** In order to create a map of motor  
220 behaviors described by body part movements, we used a previously described method for

221 discovering stereotypy in postural dynamics <sup>1</sup>. First, body part positions were predicted for each  
222 frame in our dataset to yield a set of 32 timeseries of egocentric trajectories in image  
223 coordinates for each video. These timeseries were recentered by subtracting the thorax  
224 coordinate at each timepoint and rescaled to comparable ranges by z-scoring each timeseries.  
225 The timeseries were then expanded into spectrograms by applying the Continuous Wavelet  
226 Transform (CWT) parametrized by the Morlet wavelet as the mother wavelet and 25 scales  
227 chosen to match dyadically spaced center frequencies spanning 1 to 50 Hz. This time-frequency  
228 representation augments the instantaneous representation of pose at each timepoint to one that  
229 captures oscillations across many timescales. The instantaneous spectral amplitudes of each  
230 body part were then concatenated into a single vector of length  $2(J - 1)F$  where  $J$  is the number  
231 of body parts before subtracting the body part used as reference (i.e., the thorax) and doubled  
232 to account for both  $x$  and  $y$  coordinates, and  $F$  is the number of frequencies being measured via  
233 CWT. In our data, this resulted in a 1,550-dimensional representation at each timepoint.  
234

235 Finally, we performed nonlinear dimensionality reduction on these high dimensional vectors by  
236 using a nonlinear manifold embedding algorithm <sup>16</sup>. We first selected representative timepoints  
237 via importance sampling, wherein a random sampling of timepoints in each video is embedded  
238 into a 2D manifold via t-distributed stochastic neighbor embedding (t-SNE) and clustered via the  
239 watershed transform. This allowed us to choose a set of timepoints from each video that were  
240 representative of their local clusters, i.e., spanning the space of postural dynamics. A final  
241 behavior space distribution was then computed by embedding the selected representative  
242 timepoints using t-SNE to produce the full manifold of postural dynamics in two dimensions.  
243

244 After projecting all remaining timepoints in the dataset into this manifold, we computed their 2-d  
245 distribution and smoothed with a Gaussian kernel with  $\sigma = 0.65$  to approximate the probability  
246 density function of this space. We clipped the range of this density map to the range  $[0.5 \times$   
247  $10^{-3}, 2.75 \times 10^{-3}]$  to exclude low density regions and merge very high density regions. We then  
248 clustered similar points by segmenting the space into regions of similar body part dynamics by  
249 applying the watershed transform to the density. Although both the manifold coordinates  
250 representation of each timepoint are not immediately meaningful, we were able to derive an  
251 intuitive interpretation of each cluster by referring to the high dimensional representation of their  
252 constituent timepoints. To do this, we sampled timepoints from each cluster and averaged their  
253 corresponding high dimensional feature vector, which we can then visualize by reshaping it into  
254 a body part-frequency matrix (**Fig. 4**).

## **References**

1. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**, (2014).
2. Berman, G. J., Bialek, W. & Shaevitz, J. W. Predictability and hierarchy in Drosophila behavior. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 11943–11948 (2016).
3. Chyb, S. & Gompel, N. *Atlas of Drosophila Morphology: Wild-type and Classical Mutants*. (Academic Press, 2013).
4. Klibaite, U., Berman, G. J., Cande, J., Stern, D. L. & Shaevitz, J. W. An unsupervised method for quantifying the behavior of paired animals. *Phys. Biol.* **14**, 015006 (2017).
5. Tompson, J. J., Jain, A., LeCun, Y. & Bregler, C. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. in *Advances in Neural Information Processing Systems 27* (eds. Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q.) 1799–1807 (Curran Associates, Inc., 2014).
6. Wei, S.-E., Ramakrishna, V., Kanade, T. & Sheikh, Y. Convolutional Pose Machines. *arXiv: 1602.00134* (2016).
7. Newell, A., Yang, K. & Deng, J. Stacked Hourglass Networks for Human Pose Estimation. *arXiv: 1603.06937* (2016).
8. Bulat, A. & Tzimiropoulos, G. Human pose estimation via Convolutional Part Heatmap Regression. *arXiv: 1609.01743* (2016).
9. Cao, Z., Simon, T., Wei, S.-E. & Sheikh, Y. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *arXiv: 1611.08050* (2016).
10. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv: 1412.6980* (2014).
11. Morel, P. Gramm: grammar of graphics plotting in Matlab. *JOSS* **3**, 568 (2018).
12. Mendes, C. S., Bartos, I., Akay, T., Márka, S. & Mann, R. S. Quantification of gait parameters in freely walking wild type and sensory deprived *Drosophila melanogaster*. *Elife* **2**, e00231 (2013).
13. Isakov, A. *et al.* Recovery of locomotion after injury in *Drosophila melanogaster* depends on proprioception. *J. Exp. Biol.* **219**, 1760–1771 (2016).
14. Baum, L. E., Petrie, T., Soules, G. & Weiss, N. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Ann. Math. Stat.* **41**, 164–171 (1970).

15. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **13**, 260–269 (1967).
16. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

## **Supplemental Figures and Movie Legends**

**Title:** Fast animal pose estimation using deep neural networks

**Authors:**

Pereira, T.<sup>1,#</sup>, Aldarondo, D.<sup>1,#</sup>, Willmore, L.<sup>1</sup>, Kislin, M.<sup>1</sup>, Wang, S. S.<sup>1,2</sup>, Murthy, M.<sup>1,2\*</sup>, and Shaevitz, J. W.<sup>1,3,4\*</sup>

1 Princeton Neuroscience Institute, Princeton University

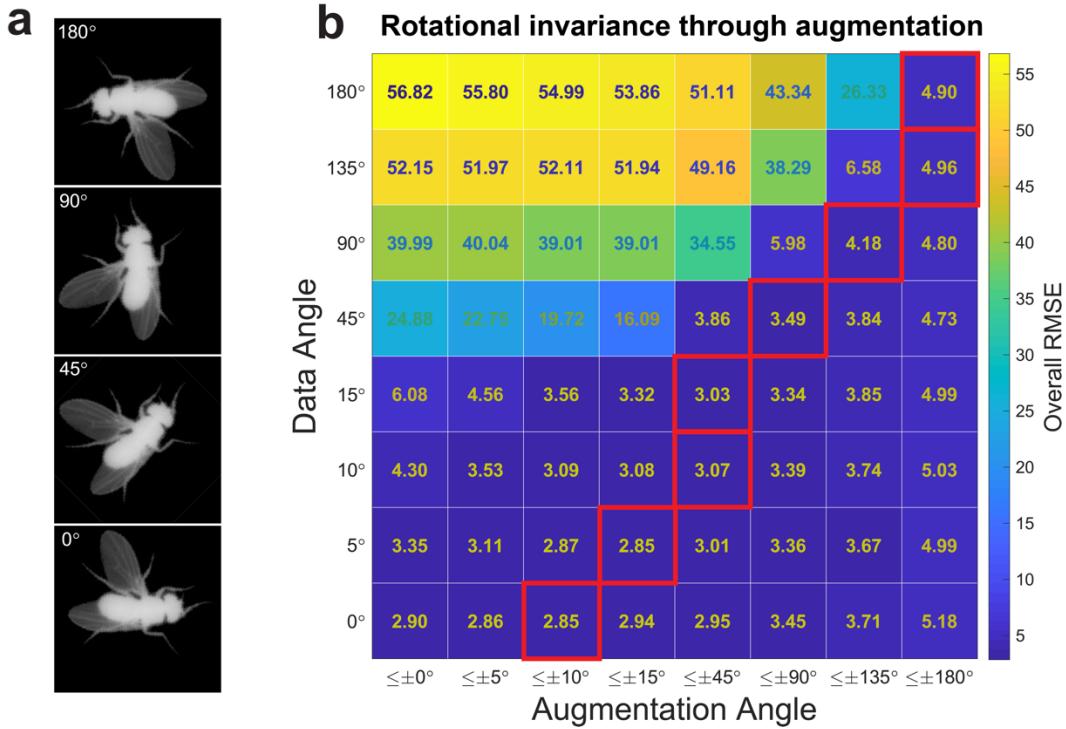
2 Department of Molecular Biology, Princeton University

3 Lewis-Sigler Institute for Integrative Genomics, Princeton University

4 Department of Physics, Princeton University

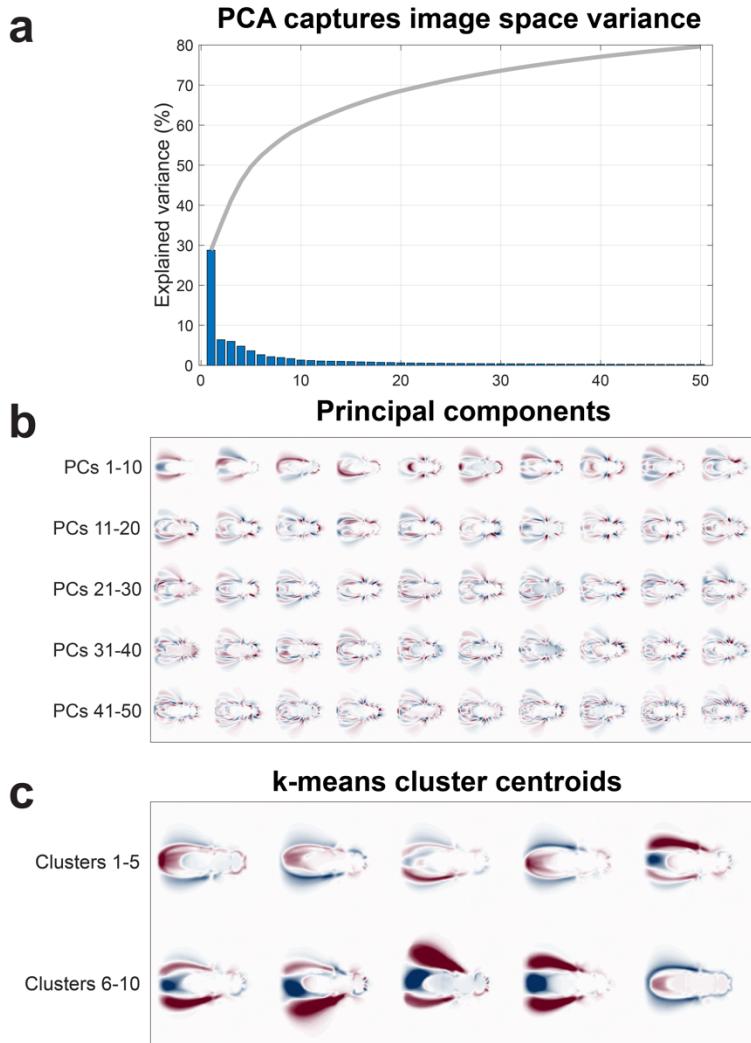
#equal authors

\*lead contacts and co-corresponding authors: Mala Murthy ([mmurthy@princeton.edu](mailto:mmurthy@princeton.edu)) and Joshua W. Shaevitz ([shaevitz@princeton.edu](mailto:shaevitz@princeton.edu))



**Supplementary Figure 1: Rotational invariance is learned at the cost of prediction accuracy**

- (a) Rotations are applied about the center of the image. During training, confidence maps are rotated accordingly.
- (b) The accuracy measured as the RMSE of position estimates when evaluated on data rotated at a fixed angle (rows) with networks trained on data augmented by rotations between a range of angles (columns). Red boxes denote the best accuracy for each data angle, denoting that optimal performance is achieved when the network is trained on augmented images with the minimally inclusive range of angles. Top accuracy decreases relative to the degree of rotational invariance the network must learn.

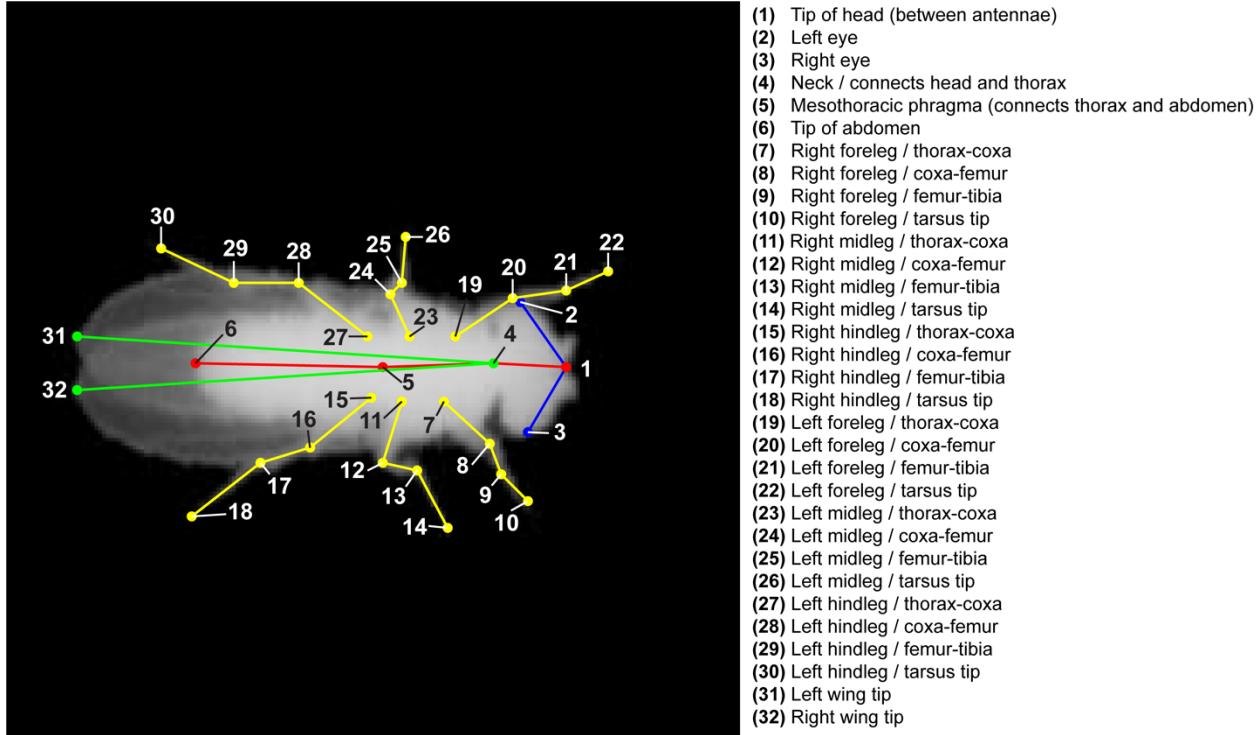


**Supplementary Figure 2: Cluster sampling to promote pose diversity in labeling dataset**

(a) Principal component analysis (PCA) of unlabeled images captures the majority of the variance in the data within 50 components. The cumulative variance explained (line) suggests that using PCA for dimensionality reduction does not sacrifice substantial loss of information within the images.

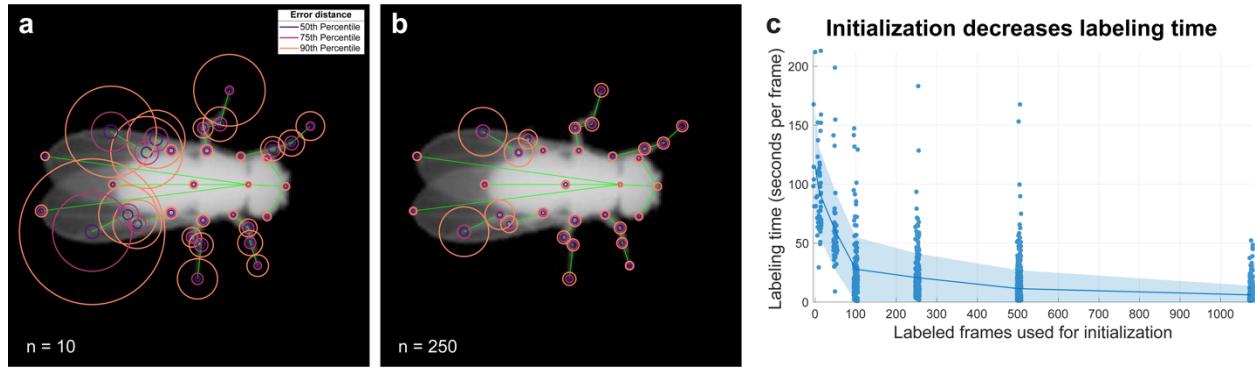
(b) Top PCA eigenmodes visualized as coefficient images. Red and blue shading denote positive and negative coefficients at each pixel. Areas of similar colors indicate correlated pixel intensities within a given mode. After mean subtraction, each image in the initially sampled dataset is projected onto all 50 eigenmodes.

(c) Cluster centroids identified by k-means after PCA. Red and blue shading denote pixels with higher or lower intensity than the overall mean. Cluster centroids illustrate the diversity of poses that are detected in image space by this sampling method. Samples are then drawn evenly from each cluster to select representative images for labeling with the GUI.



### Supplementary Figure 3: User-defined skeleton

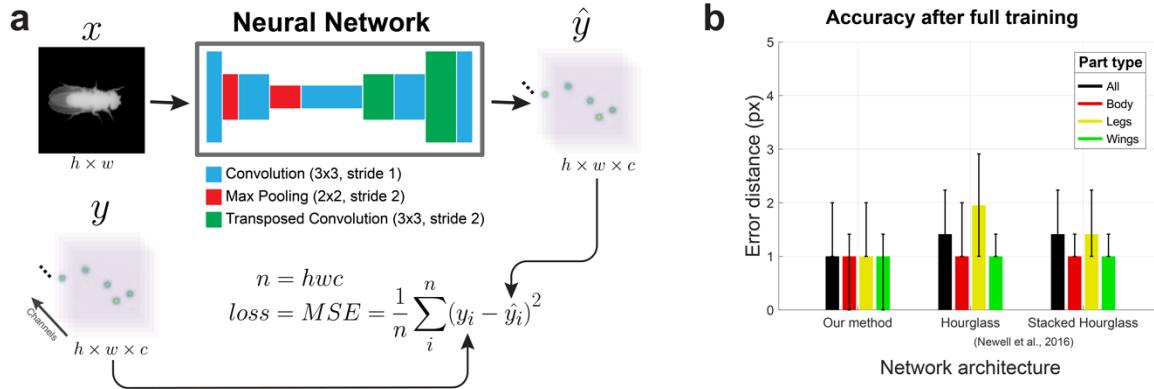
We selected 32 points to cover the body parts of the fly; these parts were chosen to approximately match the set of visible joints and interest points in the anatomy of the animal.



#### Supplementary Figure 4: Estimation accuracy improves with few samples

(a-b) Error distance distributions per body part when estimated with networks trained for 15 epochs on 10 (a) or 250 (b) labeled frames. The majority of estimates fall within few pixels of the ground truth, reducing the labeling procedure to simply correcting estimates.

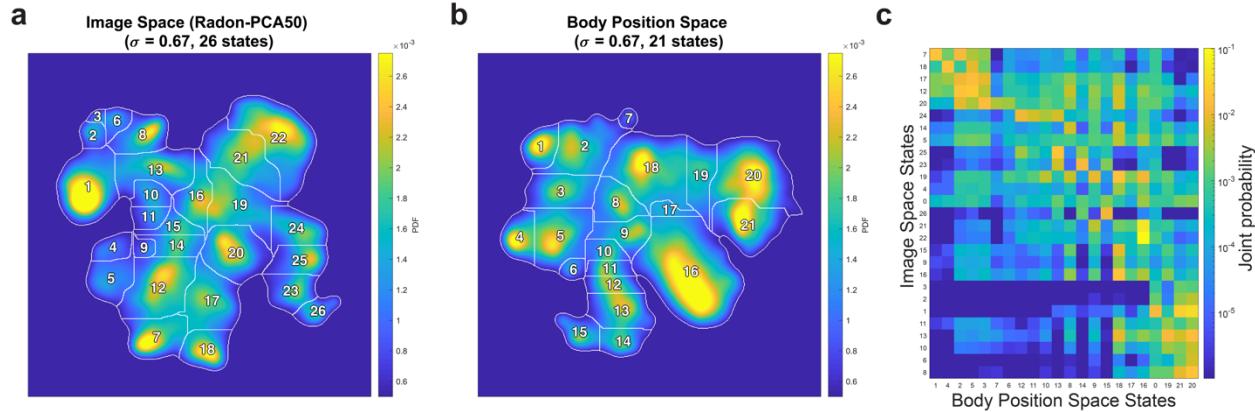
(c) Time spent labeling each frame decreases with the quality of initialization. Line and shaded region correspond to mean and standard deviation. Starting frames require  $115.4 \pm 45.0$  (mean+s.d.) seconds to label, decreasing to  $6.1 \pm 7.7$  seconds after initializing with a network trained on 1000 labeled frames.



### Supplementary Figure 5: Neural network architecture comparison

(a) Diagram of our neural network architecture. Raw images are provided as input into the network, which then computes a set of confidence maps of the same height and width as the input image (top row). The network consists of a set of convolutions, max pooling and transposed convolutions whose weights are learned during training (top middle). Estimated confidence maps are compared to ground truth maps generated from user labels using a mean squared error loss function, which is then minimized during training (bottom row).

(b) Accuracy comparison between architectures. We compared the accuracy of our architecture to the hourglass and stacked hourglass versions of the network described in<sup>1</sup>. The accuracy of our network is equivalent or better than those achieved when training with these reference architectures.

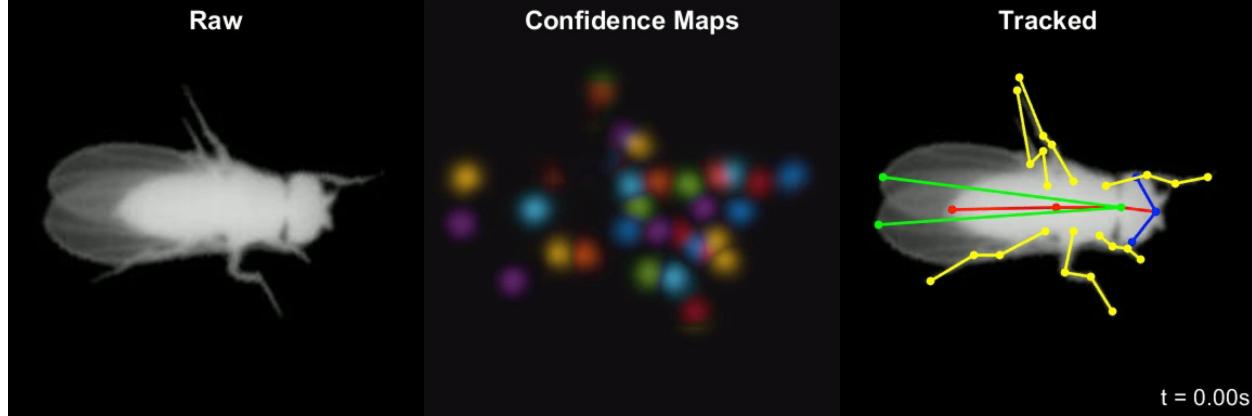


**Supplementary Figure 6: Comparison of behavioral space distributions generated from compressed images versus body part positions.**

(a) Behavioral space distribution from 59 male flies calculated using the original MotionMapper pipeline (data and pipeline from <sup>2</sup>), including Radon-transform compression and PCA-based projection onto the first 50 principal components followed by a nonlinear embedding of the resultant spectrograms.

(b) Behavioral space distribution from 59 male flies (data and pipeline from <sup>2</sup>) calculated using spectrograms generated from tracked body part positions rather than PCA modes (see **Online Methods**). We note that this distribution has fewer peaks than that from (a) and a more symmetric topology (e.g. in the top-left clusters, **Fig. 4c-g**).

(c) Joint probability distribution of the cluster labels from (a) and (b); sorted by row and column peaks. Many clusters identified using the pixel-based representation (rows) match up with those of the position-based representation (columns), but some are distributed into newly separated clusters.



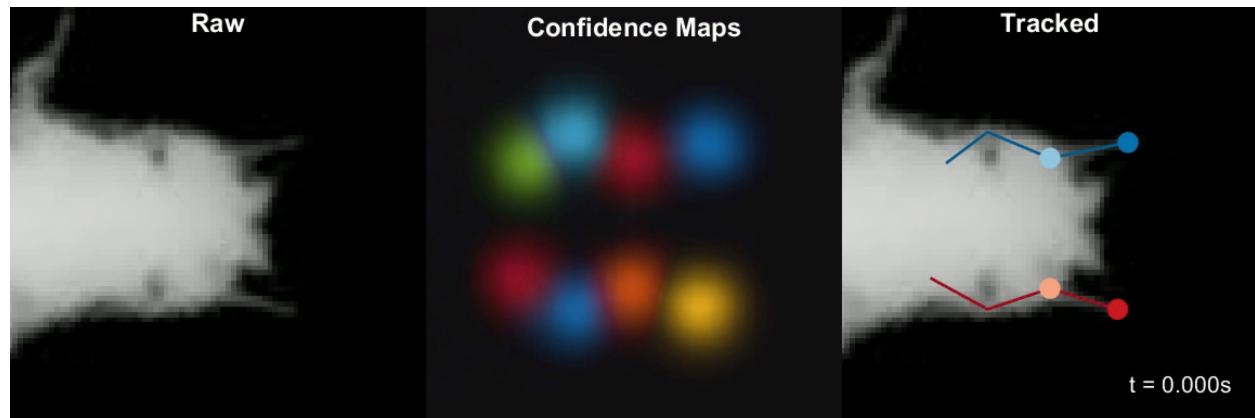
**Supplementary Movie 1: Body part tracking is reliable over long periods without temporal constraints.**

Raw images (left), max projection of all confidence maps (center), and tracked images (right) during a 20 second bout of free movement. Video playback at 0.2x realtime speed.



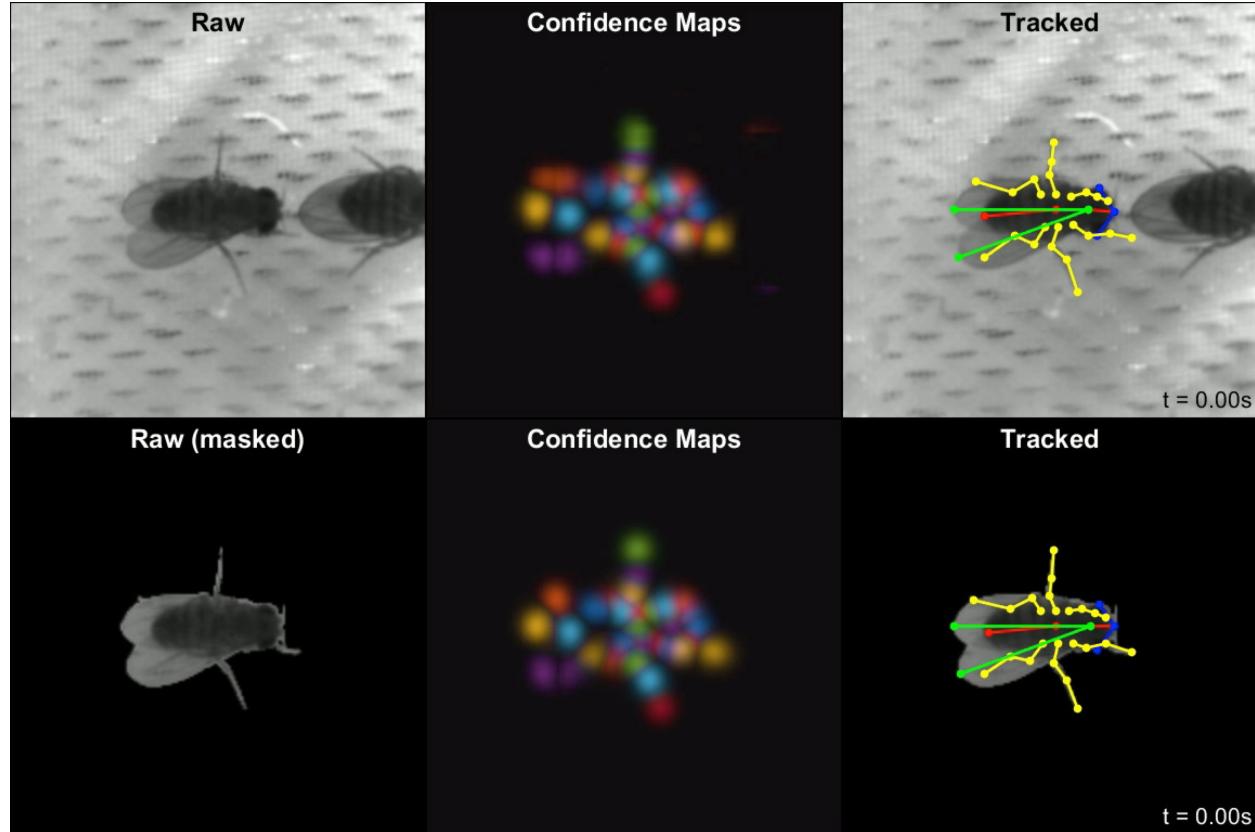
**Supplementary Movie 2: Body part tracking during freely moving locomotion.**

Raw images (left), max projection of all confidence maps (center), and tracked images (right) during a bout of locomotion. Video playback at 0.15x realtime speed. Video corresponds to Fig. 1d.



**Supplementary Movie 3: Body part tracking during head grooming.**

Raw images (left), max projection of all confidence maps (center), and tracked images (right) during a bout of head grooming. Video playback at 0.15x realtime speed. Video corresponds to Fig. 1e.



**Supplementary Movie 4: Tracking joints robustly in images with heterogeneous background and noisy segmentation.**

Raw images (left), max projection of all confidence maps (center), and tracked images (right) of a freely moving courting male fly. Rows correspond to results from a network trained on unmasked and masked images, respectively. Video playback at 0.2x realtime speed.



#### **Supplementary Movie 5: Tracking joints in freely moving rodents.**

Raw images (left), max projection of all confidence maps (center), and tracked images (right) of a freely moving mouse in an open field arena imaged from below through a clear acrylic floor. Video playback at 0.2x realtime speed. Tracking is reliable over time but degenerate when certain parts are occluded, such as when the animal rears.

#### **References**

1. Newell, A., Yang, K. & Deng, J. Stacked Hourglass Networks for Human Pose Estimation. in *Computer Vision – ECCV 2016* 483–499 (Springer International Publishing, 2016).
2. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**, (2014).