

1 ConstrastivePose: A contrastive learning approach for self-supervised
2 feature engineering for pose estimation and behaviorial classification of
3 interacting animals

4
5 Tianxun Zhou¹, Calvin Chee Hoe Cheah², Eunice Wei Mun Chin², Jie Chen³, Hui Jia Farm^{1,4},
6 Eyleen Lay Keow Goh², Keng Hwee Chiam^{1,3*}

7
8
9 ¹ Bioinformatics Institute, A*STAR, Singapore

10 ² Neuroscience and Mental Health Faculty, Lee Kong Chian School of Medicine, Nanyang
11 Technological University, Singapore

12 ³ School of Biological Sciences, Nanyang Technological University, Singapore

13 ⁴ Department of Computer Science, University of Oxford, Oxford, United Kingdom

14 * Corresponding author, email: chiamkh@bii.a-star.edu.sg

15

16

17

18 **Abstract**

19 In recent years, supervised machine learning models trained on videos of animals with pose
20 estimation data and behavior labels have been used for automated behavior classification.
21 Applications include, for example, automated detection of neurological diseases in animal models.
22 However, there are two problems with these supervised learning models. First, such models
23 require a large amount of labeled data but the labeling of behaviors frame by frame is a laborious
24 manual process that is not easily scalable. Second, such methods rely on handcrafted features
25 obtained from pose estimation data that are usually designed empirically. In this paper, we
26 propose to overcome these two problems using contrastive learning for self-supervised feature
27 engineering on pose estimation data. Our approach allows the use of unlabeled videos to learn
28 feature representations and reduce the need for handcrafting of higher-level features from pose
29 positions. We show that this approach to feature representation can achieve better classification
30 performance compared to handcrafted features alone, and that the performance improvement is
31 due to contrastive learning on unlabeled data rather than the neural network architecture.

32 **Author Summary**

33 Animal models are widely used in medicine to study diseases. For example, the study of social
34 interactions between animals such as mice are used to investigate changes in social behaviors
35 in neurological diseases. The process of manually annotating animal behaviors from videos is
36 slow and tedious. To solve this problem, machine learning approaches to automate the video
37 annotation process have become more popular. Many of the recent machine learning approaches
38 are built on the advances in pose-estimation technology which enables accurate localization of
39 key points of the animals. However, manual labeling of behaviors frame by frame for the training
40 set is still a bottleneck that is not scalable. Also, existing methods rely on handcrafted feature

41 engineering from pose estimation data. In this study, we propose ConstrastivePose, an approach
42 using contrastive learning to learn feature representation from unlabeled data. We demonstrate
43 the improved performance using the features learnt by our method versus handcrafted features
44 for supervised learning. This approach can be helpful for work seeking to build supervised
45 behavior classification models where behavior labelled videos are scarce.

46 **Introduction**

47 Analysis of animal behavior is critical in the field of neuroscience to study brain function, and
48 crucial for the assessment of treatment efficacy in preclinical testing. With the advancement of
49 molecular tools for intervention in animal models, accurate and efficient detection and
50 quantification of animal behavior is increasingly sought after. While human annotators remain the
51 gold standard in behavior scoring, they can get fatigued or overwhelmed by the vast number of
52 behaviors to score, in addition to the complexity of differentiating specific behaviors. It takes about
53 22 man-hours to annotate a one-hour video by frame with high confidence[1]. Other problems
54 with human annotation are the difficulty of ensuring high quality of annotation due to well
55 documented factors such as variability between different annotators, observer bias and observer
56 drift[1–4].

57 Automated video analysis has been introduced to help allow a semi-high throughput workflow for
58 behavioral screening in research[5]. Commercial behavior tracking software packages (e.g.
59 EthoVision, ANY-maze), or those that are incorporated in the behavioral assay equipment
60 hardware (e.g. Med Associates Inc., Campden Instruments Ltd.,) are often costly, and have low
61 customizability to user-specific experimental setting. Additionally, some studies have shown that
62 many commercial software lack sensitivity due to poor animal tracking and are unable to
63 dissociate complex animal behaviors[5–9]. Due to such drawback, machine learning-based

64 approaches using open-source software and videos acquired with consumer grade cameras have
65 steadily being embraced by animal behavioral scientists for automated tracking and analysis of
66 complex behaviors in their research models. For example, Wu et al.[10] developed a machine-
67 learning image-analysis program that automatically tracks leg claw positions of freely moving flies
68 recorded on high-speed video, producing a series of gait measurements. Their fully automated
69 leg tracking of *Drosophila* neurodegeneration models reveals distinct conserved movement
70 signatures. Hong et al.[11] studied interactions of mice with gene mutations associated with
71 autism using machine learning based video tracking and classification and detected social
72 interaction deficits compared to those without the mutations. Van den Boom et al.[6] applied open-
73 source machine learning classification software to study SAPAP3 knockout mice and confirmed
74 that they engage in more grooming than wildtype mice from the same litter both in number of
75 bouts and grooming duration.

76 The common workflow[1,5] for machine learning animal behavior classification is to first extract
77 features from the video, commonly in the form of pose estimation. Feature engineering is then
78 performed by computing hand-crafted features, such as animal orientation and length, from
79 animal pose estimation. Finally, a machine learning algorithm, either supervised learning or
80 unsupervised learning, is applied on those features. In supervised learning, the classifier is trained
81 on the features and behavior labels, and the trained classifier can be used to classify behaviors
82 in new videos. We focus on the supervised learning workflow as it is more commonly used in
83 literature.

84 There are two weaknesses of this typical workflow for practitioners. Firstly, the requirement of
85 creating a large labeled training set for the machine learning model to achieve good classification
86 accuracy. E.g., in the supervised classification of mice behavior, up to 260 minutes, and 135
87 minutes of video were annotated in [12], and [13], requiring approximately 95 and 50 man-hours

88 of work respectively to build the training and validation sets. Secondly, engineering handcrafted
89 features from pose-estimation data relies on experience and trial-and-errors. A summary of
90 various feature engineering approaches found in literature is provided in S1 Table.

91 In this study, we develop a method, which we refer to as ConstrastivePose, that seeks to address
92 these two weaknesses. The ConstrastivePose method is trained using contrastive learning on
93 unlabeled data, and then fine-tuned with a small amount of labeled data.

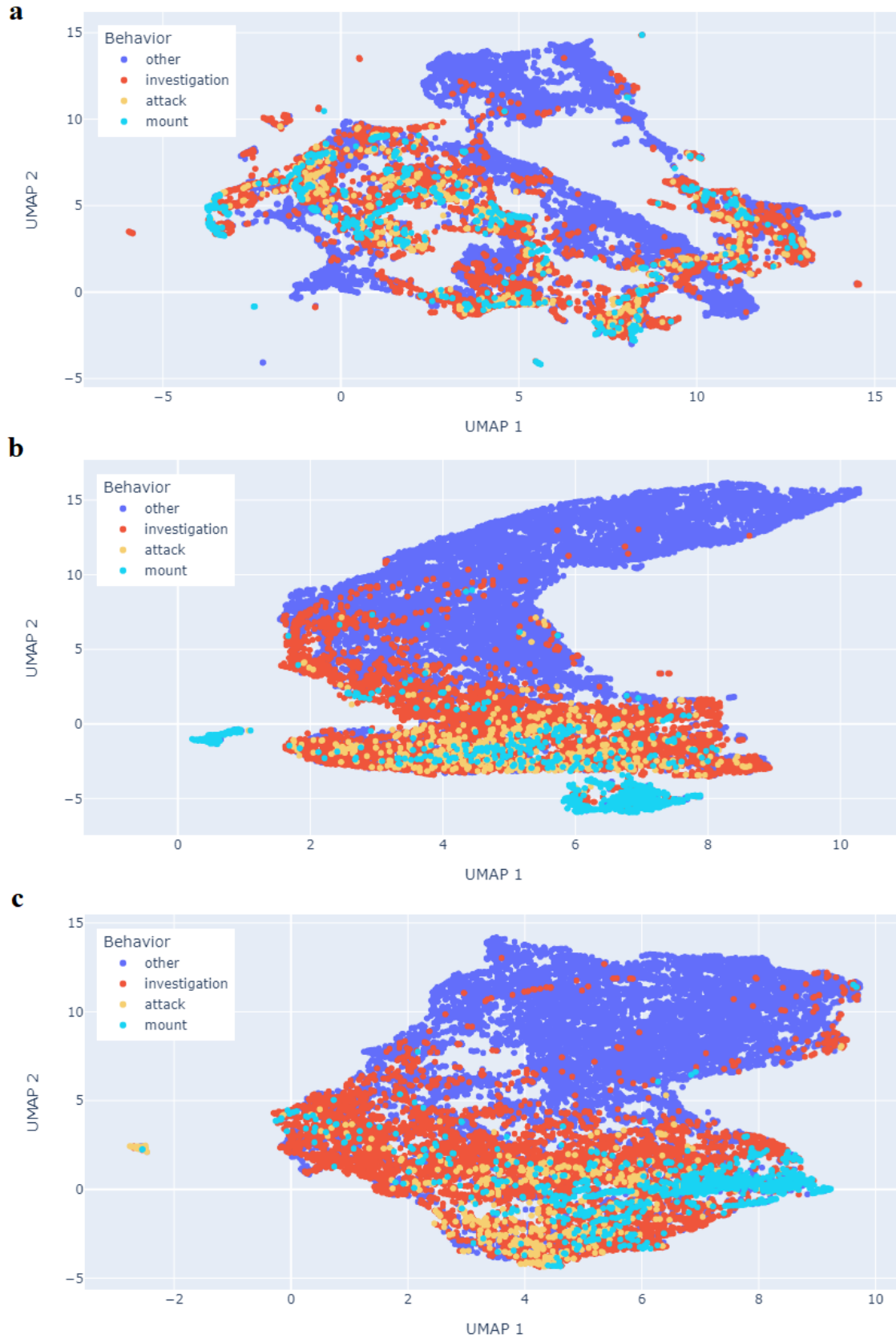
94 Contrastive learning, a form of self-supervised learning that learns useful representations of data
95 for classification without the need for labels. It is trained by *contrasting* similar data against
96 dissimilar data. For a given datapoint in a training batch, similar data is generated by data
97 augmentation of itself while dissimilar data are simply other datapoints in the training batch.
98 Through contrastive learning, ConstrastivePose leverages the availability of large sets of
99 unlabeled data generated with the automated and easily scalable pose estimation data generation
100 process. The data representation learnt by ConstrastivePose also reduces the need for manual
101 feature engineering from pose estimation. With fine-tuning after contrastive learning,
102 ConstrastivePose achieves better performance on downstream supervised learning task than
103 handcrafted feature engineering. Through this work, we hope to improve behavior classification
104 performance and alleviate the reliance on manual annotations by trained behavioral scientists to
105 decipher animal behavior.

106 **Results**

107 **ConstrastivePose learns features that exhibit similar structure as handcrafted features**

108 ConstrastivePose uses contrastive learning to reduce differences in representation between a set
109 of pose estimation and its random augmented version and enlarges their differences with other

110 examples in the batch. We trained a neural network to take in the pose-estimation and output an
111 embedding which can be interpreted as features constructed by the neural network from the
112 original data. To demonstrate how this works, we first apply ContrastivePose to the Caltech
113 Mouse Social Interactions (CaIMS21) Dataset[14] Task 1, which contains videos of two mice
114 interacting that have been labeled for key body part positions and one of four behaviors:
115 investigation, attack, mount and others (more details provided in materials and methods section).
116 Visualization of the embedding space with UMAP for the CaIMS21 dataset in **Fig 1** showed that
117 contrastive learning was able to learn a representation that is similar to the embedding spaces
118 formed by handcrafted feature engineering methods, as opposed to the original feature
119 representation with no feature engineering. We can see that in **Fig 1** panel *a*, the original feature
120 representation does not show any coherent groups or clusters between different behaviors. In
121 panel *b*, the representation of handcrafted features shows distinction between interacting
122 behaviors (investigation, attack and mount) and non-interacting behaviors (others). The learnt
123 representation in panel *c* was able to achieve similar results as in panel *B*, with clearer distinction
124 and more separable structure.



126 **Fig 1 Visualization of feature representations for CalMS21 dataset using UMAP. (a)** Representation of the
127 original untransformed features. **(b)** Representation of the handcrafted features with the best classification
128 performance in Table 3. **(c)** Representation of learnt features through contrastive learning. The learnt representation
129 in panel c was able to achieve similar results as in panel b, with clearer distinction and more separable structure.

130

131 **ConstrastivePose outperforms no feature engineering, and is on-par with handcrafted**
132 **feature engineering for supervised learning**

133 To test how well the feature representations learned by ConstrastivePose performs on supervised
134 learning, we compared our method against handcrafted engineered features that were commonly
135 used in literature (S1 Table). For our method, we trained ConstrastivePose on unlabeled data and
136 then fine-tuned it on a small set of labeled data.

137

138 A random forest model was employed to compare the test performances of the different feature
139 engineering methods. Tree based ensemble methods such as random forest are easy to train
140 and are one of the most popular supervised learning methods used in animal behavior supervised
141 classification[1]. Each of the random forest model is trained on a separate set of engineered
142 features as inputs and then tested with unseen test data.

143 The performance of the models trained on different combinations of engineered features are
144 summarized in **Table 1**. Macro-averaging was used for the metrics because of class imbalance
145 to treat all classes as equally important and avoid overoptimistic estimation of the classifier
146 performance due to the majority class. We found similar or higher scores for precision, recall, F1,
147 and accuracy for our method compared to supervised learning. In particular, the macro F1 score
148 for our method was at least 0.05 higher than the next highest supervised engineering feature,
149 indicating greater multiclass classification performance. (Complete classification results for each
150 class are provided in S7)

151

152 **Table 1 Comparison of classification performance for various handcrafted feature engineering methods for**
153 **CaIMS21 dataset**

Feature Engineering	Precision	Recall	F1 Score	Accuracy
<ul style="list-style-type: none">• Position• Frame-wise velocity [13,15,16]	0.75	0.49	0.49	0.76
<ul style="list-style-type: none">• Distance between points [12,13,15–17]• Frame-wise velocity	0.88	0.69	0.69	0.88
<ul style="list-style-type: none">• Position• Distance between points• Frame-wise velocity	0.88	0.70	0.70	0.88
Our Method				
ConstrastivePose with fine-tuning	0.82	0.72	0.75	0.88

154

155 To further validate the performance of our model, we applied it to a new set of data that we
156 generated from our in-house experiments. Two wild type mice were housed in a cage and videos
157 of them interacting were captured. Either animal can be behaving individually, such as self-
158 grooming, or one can be following the other, or engaging in sniffing the body or the anogenital
159 region of the other, or both animals can come together and perform nose-to-nose sniffing. (See
160 S2 for list of behaviors.) Thus, this dataset is more challenging as it contained more than twice as
161 many behavior classes compared to CaIMS21. Furthermore, the mice in the videos were of the
162 same color and size, which made it difficult for pose-estimation software to extract pose with high
163 accuracy. Hence, the pose estimation input was noisy and contained some missing or erroneous
164 data. This is the case for both DeepLabCut[4], a popular pose estimation software, as well as
165 using YOLO-based object detection algorithm[18], suggesting the inherent difficulty of the video

166 rather than an issue with the pose estimation software choice. Nevertheless, despite these
167 challenges, ContrastivePose provides similar performance advantages on this dataset. The
168 results are summarized in **Table 2**.

169 Well-designed handcrafted features, in this case the overlap between bounding boxes, can have
170 good prediction power. The overlap in bounding boxes is computed using the intersection area
171 between two bounding boxes. When interacting animals come into close contact with each other,
172 bounding boxes of the body parts will tend to overlap and the intersection area provide information
173 about the type and extent of contact between a pair of body parts. As seen in **Table 2**, when
174 overlap in bounding box feature was added, the classification scores increased substantially. By
175 supplementing the features learnt by ContrastivePose with well-designed handcrafted features
176 like the overlap between bounding boxes, which is easily done, it can achieve better performance
177 than just the handcrafted feature set measurably.

178 **Table 2 Comparison of classification performance for various handcrafted feature engineering methods for**
179 **in-house experiments**

Feature Engineering	Precision	Recall	F1 Score	Accuracy
<ul style="list-style-type: none">• Position of corners• Distance between points• Frame-wise velocity	0.15	0.17	0.11	0.35
<ul style="list-style-type: none">• Position of corners• Distance between points• Frame-wise velocity• Overlap in bounding boxes	0.24	0.39	0.25	0.65
Our Method				
ConstrastivePose with fine-tuning	0.30	0.44	0.32	0.72

180

181 **Discussions**

182 Machine learning methods for animal behavior classification typically follow a two-step process:
183 feature extraction from the video, commonly in the form of pose estimation, followed by machine
184 learning classification. In recent years, pose-estimation or pose-tracking has advanced rapidly
185 with the introduction of deep learning methods in computer vision that allows for markerless
186 tracking of various user-selected body parts of animals to be accurately tracked in video. Open-
187 source tools such as DeepLabCut[4], DeepPoseKit[19] and YOLO[18,20] are now popular and
188 widely used among researchers. We focus on pose estimation features as input due to their
189 popularity. In most cases, hand-crafted features such as animal orientation and length are then
190 computed from the pose-estimation to be used in the second step.

191 In the second step, machine learning is used to classify behaviors using the features extracted
192 and computed in the first step. Machine learning methods generally fall under supervised and
193 unsupervised learning[1]. Supervised learning trains a model with true labels provided. There
194 have been many works using supervised learning methods such as random forest[12,15], support
195 vector machines (SVMs)[21], and neural networks[22]. Unsupervised learning seeks to discover
196 inherent structure within the data, typically by finding various spatial groupings in a feature space
197 after some form of dimensionality reduction. These spatial groupings may correspond to various
198 human defined behaviors or behavior “motifs” upon inspection[1]. However, user oversight is still
199 necessary at the end to ensure accuracy and explainability of output variables.

200 A main weakness of supervised classification of behaviors from pose-estimation is the
201 requirement of accurate annotation for the creation of labels needed for training, which currently
202 relies on human input. Supervised learning is known to perform better with more available labeled
203 data. However, as mentioned previously, creating high quality manual labeling is a time-
204 consuming process. Generating more training data will require more man-hours spent on labeling.

205 Moreover, the use of multiple human labelers or even the same labeler on different working
206 sessions inevitably introduces variability due to observer bias and observer drift.

207 Another aspect of most existing machine learning workflows for animal behavior classification
208 using pose-estimation data as input, is feature engineering. Various methods in literature mostly
209 rely on handcrafted features computed from the pose-estimation data (S1 Table). Feature
210 engineering from pose-estimation data can be a tedious and difficult process. Handcrafting
211 features depend much on the intuition and experience of the designer. In this process, poorly
212 designed feature engineering can potentially fail to capture necessary information and
213 relationships that are needed to obtain high classification accuracy.

214 To overcome the burden of tedious manual annotation of behavior from videos and reliance on
215 trained observers, we developed ConstrastivePose, which uses contrastive learning to train on
216 pose estimation data alone, and output behavior classifications. This method reduces the need
217 for feature engineering and is able to learn from large unlabeled datasets to improve the model
218 learnt representation. Contrastive learning was first successfully applied in computer vision to
219 leverage the fact that there are huge amounts of unlabeled images available compared to labeled
220 images. Through self-supervised learning on a larger set of unlabeled images, and then fine-
221 tuning the representation learnt for downstream tasks like image classification and object
222 detection, it is possible to obtain quality performance with much lesser labeled data (18–20). Our
223 method, ConstrastivePose, is a novel application of contrastive learning on the problem of
224 classifying animal behavior from pose estimation data.

225

226 ConstrastivePose has two main advantages over existing methods. Firstly, this approach enables
227 the leveraging of larger amounts of pose-estimation extracted from unlabeled video to improve
228 predictions, alleviating the bottleneck of lesser available labeled video data. Secondly, this

229 approach reduces the need for feature engineering from user-defined to learning from the data
230 itself. In comparison, current methods with engineered features are static in the sense that once
231 the user defines the rules or calculations to generate the features, e.g. pairwise distance, angle
232 between subjects, these rules are fixed no matter how much pose-estimation data is available.
233 The self-supervised learning approach, however, can leverage on more available data to improve
234 its feature extraction ability. In the results section, we have shown that by performing contrastive
235 learning on a larger set of unlabeled pose-estimation data, and then fine-tuning with a small set
236 of labeled training data, ConstrastivePose can achieve better performance on downstream
237 supervised classification than using handcrafted features.

238 We also trained a model with the same neural network architecture using a small set of labeled
239 training data alone, without the contrastive pre-training, as a feature extractor to understand if the
240 performance improvement was due to the use of contrastive learning or simply the strength of the
241 neural network architecture itself as a feature extractor (Refer to experiment set-up details in S3
242 Fig 1). The results summarized in **Table 3** show that training from scratch on labeled data alone
243 performs worse than training with contrastive learning. This demonstrates that the performance
244 improvement comes from representation learnt during contrastive learning, and that the method
245 is an effective way of boosting performance by incorporating information from unlabeled pose-
246 estimation data.

247 **Table 3 Comparison of classification performance for ConstrastivePose and neural network without pre-**
248 **training**

Method	Precision	Recall	F1 Score	Accuracy
CaIMS21 Dataset				
ConstrastivePose	0.82	0.72	0.75	0.88
No pre-training	0.79	0.72	0.72	0.88

In-house Dataset				
ConstrastivePose + overlap in bounding boxes	0.30	0.44	0.32	0.72
No pre-training + overlap in bounding boxes	0.28	0.40	0.29	0.68

249

250 Our method can be used to study the behavior of mice and other animal models, and their social
251 interactions in a setting that allows for free interaction. We demonstrated individual specificity in
252 the data output using our model, which is critical for studies requiring individual-based
253 identification, like research on models of social behavior disorders (e.g. autism spectrum
254 disorders, anxiety disorders). Our method is not limited to any particular type of pose estimation
255 (key points, bounding boxes etc.) or set of behaviors. It can be easily applied for pose estimation
256 and additional behaviors not discussed in this study. This adds to the adaptability of our model to
257 suit various research needs, thereby achieving our goal of existing supervised learning workflow
258 to intelligently automate a task that has high human dependency.

259 Future work can seek to investigate and incorporate other techniques of self-supervised learning
260 such as pre-text task learning, for e.g. predicting missing values or predicting video clip order, to
261 improve the learnt representation further. The contrastive method proposed in this paper only
262 performs spatial augmentation and thus may not be very effective in extracting useful temporal
263 features. Hence, temporal based tasks may be especially useful for behaviors that happen over
264 a period of time such as one animal following another.

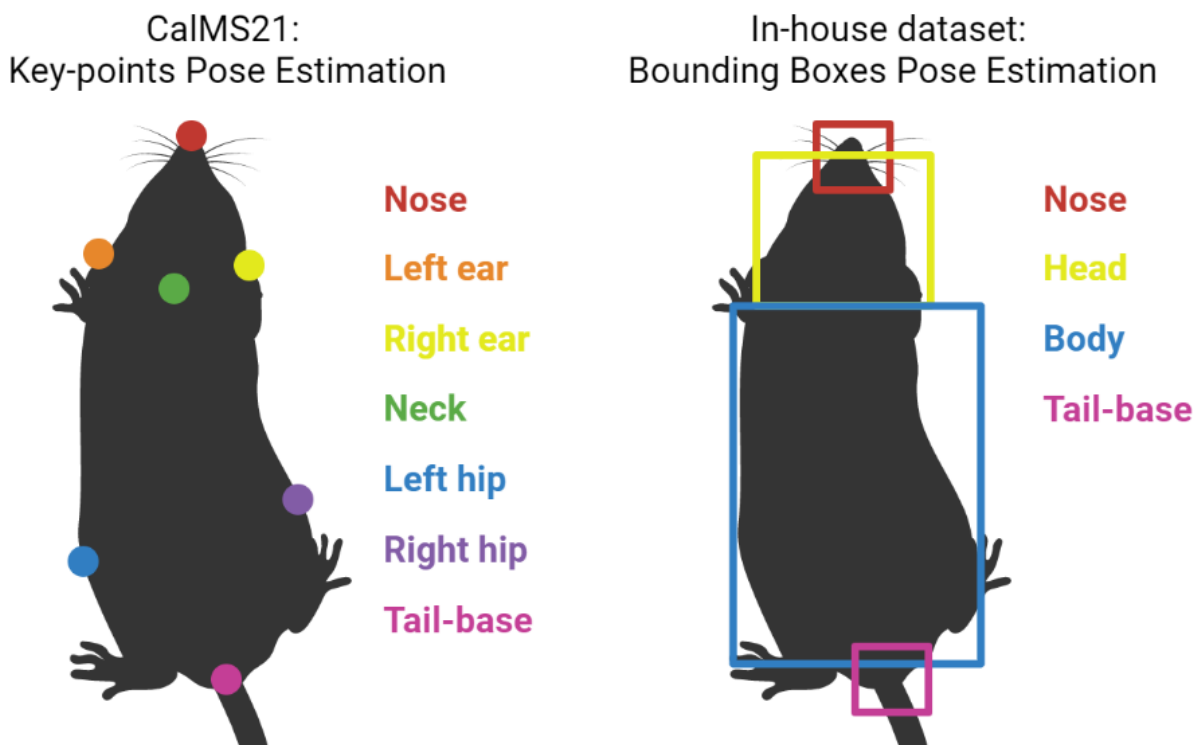
265 **Materials and Methods**

266 **Datasets**

267 The first dataset is Task 1 of the Caltech Mouse Social Interactions (CaIMS21) Dataset[14]. The
268 dataset consists of 7 labeled key points (nose, left ear, right ear, neck, left hip, right hip, and tail-
269 base) for two interacting mice in a box. For each key point, there is the x and y pixel positions.
270 Each frame is labeled for 4 behaviors: attack, investigation, mount and others (non-interaction).
271 Please refer to [14] for details on the dataset.

272 The second dataset is video recording from an in-house experiment conducted on two interacting
273 mice in a box. The pose of mice in the video is labeled using the YOLOv3 algorithm[20]. YOLO
274 generated pose-estimation data consists of 4 bounding boxes capturing the nose, head, body,
275 and tail-base for each mouse. For each bounding box, there is the x and y pixel positions of the
276 top left corner of the box, and the height and width of the box. The videos are labeled for 10
277 behaviors: nose-nose sniff, body sniff 1, body sniff 2, anogenital sniff 1, anogenital sniff 2, mutual
278 circle, affiliative, following 1, following 2 and exploration (behaviors with suffixes 1 and 2 indicate
279 the identity of mice performing the action). Illustrations of behaviors are provided in S2 Table. The
280 use of bounding box tracking by YOLO instead of key points also serves to demonstrate
281 generalizability to different types of pose-estimation methods.

282 The pose estimation for both dataset are illustrated **Fig 2**.



283

284 **Fig 2 Pose estimation of mice used in CaIMS21 and in-house dataset.** CaIMS21 dataset uses 7 key points, each
285 defined by a x, y positional value. The in-house dataset uses 4 bounding boxes, each defined by the x, y positional
286 value of the top left corner, and the height and width of the box.

287

288 **Overview of methods**

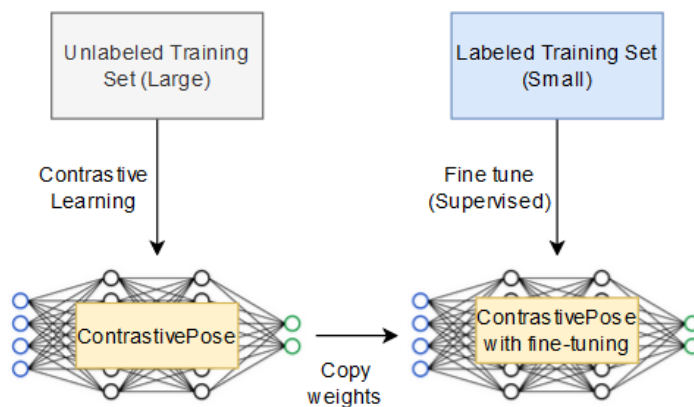
289 ContrastivePose takes as input pose estimation data and outputs a feature representation of the
290 data, which can then be used for downstream supervised classification. It is akin conceptually to
291 the feature engineering step.

292 The training for ContrastivePose model uses a large set of unlabeled training data. After training
293 with the unlabeled data through contrastive learning, the model would then be fine-tuned with a
294 relatively small set of labeled training data in a supervised fashion. For the downstream task of

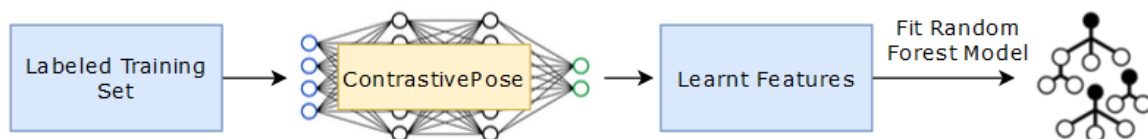
295 supervised classification of behaviors, the small set of labeled training data would be used to train
296 a random forest model (other type of machine learning model may also be used instead). The
297 random forest model takes the learnt representations as computed by the previously trained
298 ConstrastivePose as input. The process is illustrated in **Fig 3**.

299 For inference, we simply take the video that we wish to be labeled, obtain the pose estimation for
300 each frame of the video, pass it through the trained ConstrastivePose model and use its output
301 representation as the input for the trained random forest classifier to obtain the behavior
302 predictions.

Training of ConstrastivePose Model



Downstream Task: Supervised Behavior Classification



303

304 **Fig 3 Overview of ConstrastivePose method.** The ConstrastivePose model is trained on large set of unlabeled data
305 through contrastive learning, and then fine-tuned on a small set of labeled data. When applying to downstream

306 supervised behavior classification task, the labeled training set is passed to the ConstrastivePose model which outputs
307 the learnt features that can be then used to fit a classifier such as random forest

308 The specific details of the training and testing workflow for the data used in this paper are
309 presented in S3.

310 **Sliding window for input data**

311 To capture temporal aspects of the animal poses which are important for behaviors that take place
312 over many frames, such as following, and attack, we used a sliding window approach to generate
313 the input data for the training and test sets. The length of the sliding window is a hyperparameter.
314 We set this at 30 frames for a 30-frames per second video based on visual inspection that
315 temporal activities can be identified within a second of the video. This hyperparameter has not
316 been tuned. Each datapoint is therefore a matrix of size $30 \times \text{number of original features}$. For
317 example, for CalMS21, there are 2 mice, 7 key points for each mouse, and 2-D coordinates for
318 each point. We take the frame $t - 29$ to frame t and concatenate into a 30×28 matrix as X_t , and
319 the label Y_t will be the behavior labeled for frame t .

320 **Contrastive learning**

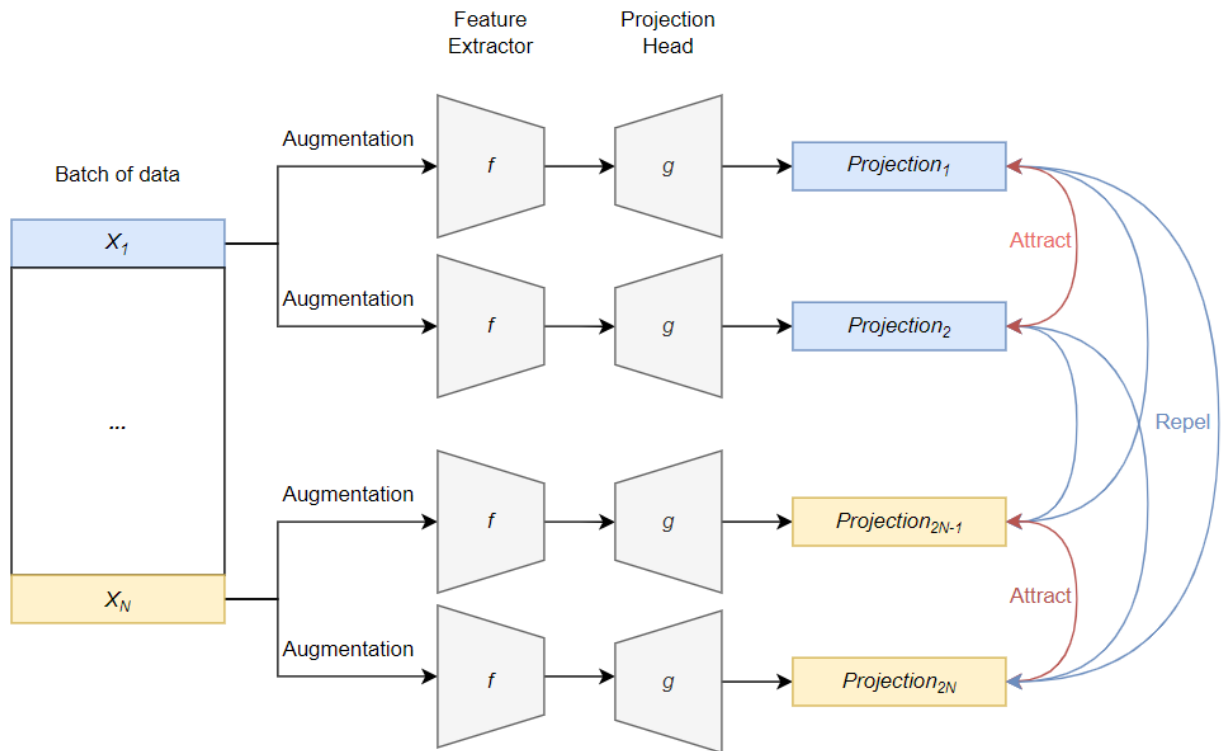
321 Contrastive learning has been the most successful self-supervised learning technique used in
322 computer vision, achieving state-of-the-arts performance. Self-supervised learning is an approach
323 to learn from unlabeled samples by generating tasks or pseudo-labels from the data and training
324 a neural network to learn to solve those tasks or pseudo-labels. Some examples of tasks include
325 inpainting missing sections of images or unscrambling scrambled images[23,24]. Through this
326 self-supervised training, the model can learn a representation of the data that is also helpful for
327 other downstream tasks. The model can then be fine-tuned with small amounts of training data to
328 be optimized for downstream tasks. Recently, contrastive learning has been applied for feature

329 extraction from animal videos by Jia et al.[25], by performing contrastive learning on the frame
330 images directly in similar fashion to existing work in computer vision.

331 Contrastive learning's goal is to learn representations of data such that similar datapoints are
332 close to each other, while dissimilar ones are far apart, without the need for labels[24]. This is
333 achieved during training by using data augmentation. The data augmentation should not change
334 the fundamental characteristic of the data that is relevant for the task at hand. For example, data
335 augmentation in contrastive learning for image classification tasks include random crops and
336 rotations. These augmentations do not change the fundamental characteristic of the data for
337 image classification because a rotated or cropped image still represents the same class of
338 object[24,26].

339 For pose estimation data, augmentation is achieved by random flipping, rotation and translation
340 of the poses, which do not change the fundamental characteristic for behavior classification. It is
341 the same behavior no matter how we mirror, rotate, or translate the setup. Hence, we can define
342 a data augmentation that performs random flipping along the x or y axis, rotation by random
343 angles, and random translation along both axes. For details on the implementation of
344 augmentation, please refer to Supplementary Materials.

345 During training, the model learns to reduce the difference in representations between any image
346 and its random augmented version and enlarge the difference with other images in the
347 batch[24,26]. The training process is illustrated in **Fig 4**. For detailed steps of the implementation
348 of contrastive learning, and neural network architecture used, refer to S4 – S6.



349

350 **Fig 4 Contrastive learning process.** For a given batch of data, each data point goes through parallel random
351 augmentation and gets passed through the networks to obtain two projections. The contrastive loss is computed over
352 the whole batch. By minimizing the contrastive loss, the model seeks to make the matching pairs' projections more
353 similar while making all other pairs' projections dissimilar

354 Feature engineering methods

355 We describe the methods used to compute the engineered features used as comparison in this
356 paper.

357 Velocity features are computed by the difference position between any frame and its previous
358 frame. For example, in the CalMS21 dataset, with 2 mice each with 7 body parts described by x ,
359 y coordinates, there are 28 velocity features.

360 Distances between key points are computed as the Euclidean distance between the combination
361 of any key point of one mouse with any key point of the other mouse. For example, in the CalMS21
362 dataset, there are 7×7 key point pairs, which give 49 pairwise distance features.

363 For bounding boxes, the overlap in bounding boxes is computed as the intersection area between
364 two bounding boxes, divided by total area covered by both boxes. Similar to pairwise distances,
365 the overlap ratio is computed for all combinations of key points of one mouse with key points of
366 the other mouse. For example, in our in-house dataset, there are 4×4 key point pairs giving 16
367 pairwise overlap ratio features.

368 **Acknowledgments**

369 N/A

370 **References**

- 371 1. von Ziegler L, Sturman O, Bohacek J. Big behavior: challenges and opportunities in a
372 new era of deep behavior profiling. *Neuropsychopharmacology*. 2021;46: 33–44.
373 doi:10.1038/s41386-020-0751-7
- 374 2. Bohlen M, Hayes ER, Bohlen B, Bailoo JD, Crabbe JC, Wahlsten D. Experimenter effects
375 on behavioral test scores of eight inbred mouse strains under the influence of ethanol.
376 *Behav Brain Res*. 2014;272: 46–54. doi:10.1016/j.bbr.2014.06.017
- 377 3. Garcia VA, Junior CFC, Marino-Neto J. Assessment of observers' stability and reliability
378 — A tool for evaluation of intra- and inter-concordance in animal behavioral recordings.
379 *Annu Int Conf IEEE Eng Med Biol Soc* 2010. 2010. pp. 6603–6606.
380 doi:10.1109/IEMBS.2010.5627131

- 381 4. Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, et al. DeepLabCut:
382 markerless pose estimation of user-defined body parts with deep learning. *Nat Neurosci.*
383 2018;21: 1281–1289. doi:10.1038/s41593-018-0209-y
- 384 5. Sturman O, von Ziegler L, Schläppi C, Akyol F, Privitera M, Slominski D, et al. Deep
385 learning-based behavioral analysis reaches human accuracy and is capable of
386 outperforming commercial solutions. *Neuropsychopharmacology.* 2020;45: 1942–1952.
387 doi:10.1038/s41386-020-0776-y
- 388 6. van den Boom BJG, Pavlidi P, Wolf CJH, Mooij AH, Willuhn I. Automated classification of
389 self-grooming in mice using open-source software. *J Neurosci Methods.* 2017;289: 48–
390 56. doi:10.1016/j.jneumeth.2017.05.026
- 391 7. Bailoo JD, Bohlen MO, Wahlsten D. The precision of video and photocell tracking
392 systems and the elimination of tracking errors with infrared backlighting. *J Neurosci*
393 *Methods.* 2010;188: 45–52. doi:10.1016/j.jneumeth.2010.01.035
- 394 8. Geuther BQ, Deats SP, Fox KJ, Murray SA, Braun RE, White JK, et al. Robust mouse
395 tracking in complex environments using neural networks. *Commun Biol.* 2019;2: 1–11.
396 doi:10.1038/s42003-019-0362-1
- 397 9. Sturman O, Germain PL, Bohacek J. Exploratory rearing: a context- and stress-sensitive
398 behavior recorded in the open-field test. *Stress.* 2018;21: 443–452.
399 doi:10.1080/10253890.2018.1438405
- 400 10. Wu S, Tan KJ, Govindarajan LN, Stewart JC, Gu L, Ho JWH, et al. Fully automated leg
401 tracking of drosophila neurodegeneration models reveals distinct conserved movement
402 signatures. *PLoS Biol.* 2019;17: e3000346. doi:10.1371/journal.pbio.3000346
- 403 11. Hong W, Kennedy A, Burgos-Artizzu XP, Zelikowsky M, Navonne SG, Perona P, et al.
404 Automated measurement of mouse social behaviors using depth sensing, video tracking,
405 and machine learning. *Proc Natl Acad Sci U S A.* 2015;112: E5351–E5360.
406 doi:10.1073/pnas.1515982112

- 407 12. Nilsson SR, Goodwin NL, Choong JJ, Hwang S, Wright HR, Norville ZC, et al. Simple
408 Behavioral Analysis (SimBA) – an open source toolkit for computer classification of
409 complex social behaviors in experimental animals. *BioRxiv [Preprint]*. 2020 [cited 22 Oct
410 2022]. doi:10.1101/2020.04.19.049452
- 411 13. Lorbach M, Kyriakou EI, Poppe R, van Dam EA, Noldus LPJJ, Veltkamp RC. Learning to
412 recognize rat social behavior: Novel dataset and cross-dataset application. *J Neurosci*
413 *Methods*. 2018;300: 166–172. doi:10.1016/j.jneumeth.2017.05.006
- 414 14. Sun JJ, Karigo T, Chakraborty D, Mohanty SP, Wild B, Sun Q, et al. The Multi-Agent
415 Behavior Dataset: Mouse Dyadic Social Interactions. *arXiv [Preprint]*. 2021 [cited 22 Oct
416 2022]. doi:10.48550/arXiv.2104.02710
- 417 15. Segalin C, Williams J, Karigo T, Hui M, Zelikowsky M, Sun JJ, et al. The mouse action
418 recognition system (MARS) software pipeline for automated analysis of social behaviors
419 in mice. *Elife*. 2021;10. doi:10.7554/eLife.63720
- 420 16. Hsu AI, Yttri EA. B-SOiD, an open-source unsupervised algorithm for identification and
421 fast prediction of behaviors. *Nat Commun*. 2021;12. doi:10.1038/s41467-021-25420-x
- 422 17. Batpurev T, Shibata T, Matsumoto J, Nishijo H. Automatic Identification of Mice Social
423 Behavior Through Multi-Modal Latent Space Clustering. 2021 Joint 10th International
424 Conference on Informatics, Electronics and Vision, ICIEV 2021 and 2021 5th
425 International Conference on Imaging, Vision and Pattern Recognition, icIVPR 2021.
426 Institute of Electrical and Electronics Engineers Inc.; 2021.
427 doi:10.1109/ICIEVICIVPR52578.2021.9564213
- 428 18. Arac A, Zhao P, Dobkin BH, Carmichael ST, Golshani P. DeepBehavior: A Deep
429 Learning Toolbox for Automated Analysis of Animal and Human Behavior Imaging Data.
430 *Front Syst Neurosci*. 2019;13: 20. doi:10.3389/fnsys.2019.00020

- 431 19. Graving JM, Chae D, Naik H, Li L, Koger B, Costelloe BR, et al. Deepposekit, a software
432 toolkit for fast and robust animal pose estimation using deep learning. *Elife*. 2019;8.
433 doi:10.7554/eLife.47994
- 434 20. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object
435 detection. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. 2016;2016-
436 December: 779–788. doi:10.1109/CVPR.2016.91
- 437 21. Jhuang H, Garrote E, Yu X, Khilnani V, Poggio T, Steele AD, et al. Automated home-
438 cage behavioural phenotyping of mice. *Nat Commun*. 2010;1: 1–10.
439 doi:10.1038/ncomms1064
- 440 22. Rousseau JBI, van Lochem PBA, Gispen WH, Spruijt BM. Classification of rat behavior
441 with an image-processing method and a neural network. *Behav Res Methods Instrum*
442 *Comput*. 2000;32: 63–71. doi:10.3758/BF03200789
- 443 23. Doersch C, Zisserman A. Multi-task Self-Supervised Visual Learning. *Proc IEEE Int Conf*
444 *Comput Vis*. Institute of Electrical and Electronics Engineers Inc.; 2017. pp. 2070–2079.
445 doi:10.1109/ICCV.2017.226
- 446 24. Khan A, Albarri S, Manzoor MA. Contrastive Self-Supervised Learning: A Survey on
447 Different Architectures. *2nd IEEE International Conference on Artificial Intelligence, ICAI*
448 *2022*. 2022; 1–6. doi:10.1109/ICAI55435.2022.9773725
- 449 25. Jia Y, Li S, Guo X, Lei B, Hu J, Xu XH, et al. Selfee, Self-supervised Features Extraction
450 of animal behaviors. *Elife*. 2022;11. doi:10.7554/ELIFE.76218
- 451 26. Chen T, Kornblith S, Norouzi M, Hinton G. A Simple Framework for Contrastive Learning
452 of Visual Representations. *arXiv [Preprint]*. 2020 [cited 22 Oct 2022].
453 doi:10.48550/arXiv.2002.05709
- 454 27. Luxem K, Mocellin P, Fuhrmann F, Kürsch J, Remy S, Bauer P. Identifying Behavioral
455 Structure from Deep Variational Embeddings of Animal Motion. *BioRxiv [Preprint]*. 2022.
456 doi:10.1101/2020.05.14.095430

