8 Supplementary Video Captions

Supplementary Video 1 DeepLabCut model predictions (631 labeled frames), mirror-mouse dataset, corresponding to traces in Fig. 1B. See Supplementary Fig. 5 for a detailed caption of Supplementary Videos 1, 5-7. [link]

Supplementary Video 2 Selected frames with high Pose PCA errors, mirror-mouse dataset. DLC model trained with 631 labeled frames. See Supplementary Fig. 6 for a detailed caption of Supplementary Videos 2-4. [link]

Supplementary Video 3 Selected frames with high Pose PCA errors, mirror-fish dataset. DLC model trained with 354 labeled frames. [link]

Supplementary Video 4 Selected frames with high Pose PCA errors, CRIM13 dataset. DLC model trained with 800 labeled frames. [link]

Supplementary Video 5 Baseline vs semi-supervised TCN model predictions (75 labeled frames), mirror-mouse dataset, corresponding to traces in Fig. 4A. [link]

Supplementary Video 6 Baseline vs semi-supervised TCN model predictions (75 labeled frames), mirror-fish dataset, corresponding to traces in Extended Data Fig. 4A. [link]

Supplementary Video 7 Baseline vs semi-supervised TCN model predictions (800 labeled frames), CRIM13 dataset, corresponding to traces in Extended Data Fig. 5A. [link]

Supplementary Video 8 Ensemble Kalman Smoother predictions (multi-view PCA), mirror-mouse dataset, corresponding to the session in Supplementary Fig. 2. See Supplementary Fig. 7 for a detailed caption of Supplementary Videos 8-12. [link]

Supplementary Video 9 Ensemble Kalman Smoother predictions (Pose PCA), IBL-pupil dataset, corresponding to the session in Supplementary Fig. 3. [link]

Supplementary Video 10 Ensemble Kalman Smoother predictions (multi-view PCA), IBL-paw dataset, corresponding to the session in Supplementary Fig. 4. [link]

Supplementary Video 11 Ensemble Kalman Smoother predictions (multi-view PCA), mirror-fish dataset. [link]

Supplementary Video 12 Ensemble Kalman Smoother predictions (temporal), CRIM13 dataset. [link]

Supplementary Video 13 Trial-by-trial markers and traces for IBL-pupil dataset, corresponding to the example session in Fig. 6. See Supplementary Fig. 8 for a detailed caption of Supplementary Videos 13-14. [link]

Supplementary Video 14 Trial-by-trial markers and traces for IBL-paw dataset, corresponding to the example session in Extended Data Fig. 7. [link]

Lightning Pose: Supplementary Information

1 Supplementary Methods

1.1 Runtime benchmarking

We used the mirror-mouse dataset (two views, 17 total keypoints) and measured training and inference time. We compared each of our model variants – supervised baseline, TCN, semi-supervised (combining the three losses presented in the main text), semi-supervised TCN model – and DeepLabCut (Mathis *et al.* [1]; version 2.3.5). For all models, we used a ResNet-50 backbone. All experiments ran on an NVIDIA A10 GPU (using 16 CPU threads and 24 GB GPU memory, costing 1.624 USD per hour on-demand on Amazon Web Services as of September 2023).

1.1.1 Training time

We measured the time it takes to load a batch of data from disk and complete a stochastic gradient descent step, an operation that forms the building block of neural network training. We varied two parameters: image resizing dimensions (128×128 and 256×256; images are read and resized "on the fly"), and labeled batch size B (16 and 32 images). We timed 100 training batches after discarding the first 15 "warmup" training batches. For DeepLabCut, we used the Tensorpack accelerated data loading package (henceforth DLC+tensorpack). As expected, the supervised model's training time is on par with DLC+tensorpack (Supplementary Fig. 9). All other models train on a significant number of additional unlabeled frames and are therefore slower to train. The TCN (with $J = 5$ context frames) operates on a total of $5B$ frames and is 2x-5x slower; the semi-supervised model appends an additional unlabeled video clip of length $2B$ frames (for a total of $3B$ frames) and is 2x-4x slower; their combination operates on $7B$ frames and is 3x-5x slower. To put these estimates in perspective, a typical network will perform at minimum 300 passes over the entire labeled dataset (a.k.a. "epochs"). With 631 labeled InD images of the mirror-mouse dataset (each 256×256), and a batch size of 32, each epoch will include $\lceil 631/32 \rceil = 20$ batches. Throughout training, the semi-supervised model will ingest additional 20 batches $\times$ 64 video frames $\times$ 300 epochs = $384K$ unlabeled video frames, and will train in approximately $(0.44s \times 20 \times 300)/60s = 44$ minutes on a single NVIDIA A10 GPU (excluding logging operations) which will cost approximately 1.2 USD.

1.1.2 Inference time

Next, we calculate the speed at which a trained network predicts a video (in frames-per-second, FPS). We compare three models: Lightning Pose without context (LP; either supervised or semi-supervised, since the training strategy does not affect inference time), LP with context (LP+TCN), and DLC. We calculate FPS for increasing image sizes (128x128, 256x256, 384x384, 512x512) and sequence lengths (16, 32, 64, 128, 256, 512). LP and LP+TCN use NVIDIA-DALI video loading which includes an additional resizing operation "on the fly," which is not done by DeepLabCut.

Supplementary Fig. 10 summarizes the results. LP (orange) tends to have the highest throughput, with DLC (blue) 5% slower, and LP+TCN (green) 50% slower when averaged across all sequence lengths and frame sizes. LP speedups become more pronounced with longer sequence lengths and larger frame sizes, where GPU utilization increases. For smaller image sizes and sequence lengths, LP and LP+TCN are slower due to a paradoxical "data bottleneck": since NVIDIA-DALI decodes and augments the videos entirely on the GPU, using very small batch sizes is inefficient as it "seeks" the entire video to find each batch. The seeking operation occupies the GPU instead of performing backward and forward passes through the network. In this unconventional regime, and with many CPUs available, standard CPU dataloaders like DeepLabCut's are preferable. To summarize, when GPUs are properly utilized, LP's inference is faster, as expected.

1.2 EKS' Relationship to previous work

We are far from the first to notice that the output of pose tracking networks can contain "glitches," and a number of strategies have been proposed for post-processing the network output to remove these glitches.

The simplest and perhaps most commonly applied strategy [2–4] is to detect "bad" keypoints and frames and remove them, followed by simple temporal interpolation to fill in the resulting gaps in the estimated pose traces. A number of criteria have been proposed to find bad frames, e.g., low network confidence, large temporal jumps, and/or multi-camera inconsistency.

While attractively simple, this "remove-then-interpolate" strategy is suboptimal for several reasons. First, it can be challenging to automatically and reliably choose thresholds to determine which bad frames should be dropped. Second, using simple temporal interpolation to replace removed keypoints ignores useful spatial constraints (such as the multi-view PCA loss) which, as we have seen, can significantly improve the estimation of uncertain keypoints. Third, networks often make errors confidently that may not be corrected with this simple strategy. Finally, even error frames often contain partial information about keypoint location (for example, a keypoint may be near the estimated value, but not match the estimated value exactly), and removing error frames completely discards this useful partial information.

A number of more complex denoising strategies have been proposed in the single-animal pose tracking literature [5–8], in addition to post-processing strategies for the multi-animal tracking case [9, 10] that are beyond the scope of this paper. These advanced techniques vary widely in their complexity, computational demands, assumptions, generality, outlier-handling logic, etc., but they all operate on the output of a single network. As we have seen, the output of a single network (particularly a fully-supervised network) can be highly unreliable, and moreover the reliability (as measured by the ensemble variance) can vary sharply from frame to frame. Without well-calibrated information about the reliability of each frame, it can be difficult to correct network errors.

Previous work [11] showed that training an ensemble of multiple networks and using the mean prediction leads to better pose estimates. Our Ensemble Kalman Smoother (EKS) denoises the ensemble means and leads to further drastic improvements in pixel errors (main text Fig. 5). EKS leverages the per-frame and per-keypoint ensemble variance to determine the degree of smoothing by our spatiotemporal priors. To reiterate, EKS extracts information even from "bad" predictions, all without the need for the user to set any manual thresholds to detect these "bad" points. Once the ensemble has been run our method is simpler, faster, more interpretable, and easier to tune than the more complex strategies discussed in [5–7].

However, it is important to note that these more complex methods are complementary to ours: future work could combine our ensembling strategy with the more realistic nonlinear constraints and non-Gaussian observation models developed in these previous papers, to potentially obtain further accuracy improvements (at the cost of longer computational post-processing times).

Finally, a note on terminology: the Ensemble Kalman Smoother we use here is different from the Ensemble Kalman filter commonly used e.g. in weather prediction [12]. The two approaches differ in whether ensembling is performed in the dynamics step or the observation step of the Kalman filter model. In our case, the ensemble is used to generate the observation model (which is then smoothed with a simple linear-Gaussian dynamical system model), whereas in the weather prediction context ensembling is performed over multiple instances of nonlinear dynamics models, which are then combined with a simple Kalman-like observation update.

1.3 Differences between DeepLabCut and the Baseline model

Backbones. Both packages support multiple backbones. In the results we use AnimalPose10K for all datasets but mirror-fish (where we use ImageNet), whereas DeepLabCut defaults to ImageNet.

Heatmap peak finding. As described in the Methods section, Lightning Pose operates on normalized heatmaps, that are valid 2D probability distributions for each keypoint. We find the heatmaps' peaks (i.e., the predicted width-height coordinates) via a soft argmax. Crucially, this process is differentiable, which is necessary when defining downstream unsupervised loss functions on the width-height coordinates. DeepLabCut, on the other hand, uses a location refinement strategy that requires a hard argmax which is not differentiable, and therefore incompatible with unsupervised losses.

Supervised heatmap losses. For the supervised loss, we use the mean-squared error between the target heatmap and the predicted heatmap. DeepLabCut uses a cross-entropy loss for each pixel.

Training. Though both use the Adam optimizer, DeepLabCut trains for a certain number of "iterations" (typically 50K) independent of the dataset size, while we use the more common "epochs" method which counts full passes over the labeled dataset, typically 300 at a minimum. As a result there are differences in the learning rate schedule as well.

1.4 IBL-paw results

Here, we report results on the "IBL-paw" dataset. As in "IBL-pupil", we compare the following three approaches: DeepLabCut with custom post-processing (DLC), Lightning Pose's semi-supervised TCN model followed by the same post-processing (LP; using temporal difference and Pose PCA losses), and a multi-view EKS variant which uses an ensemble of $m = 5$ LP models (LP+EKS). Here too, we obtain performance gains (Extended Data Fig. 7 and Supplementary Fig. 4).

We use two cameras to track the paws and, therefore, we can use the multi-view PCA loss to help quantify the rate of paw tracking errors. (Since the dataset does not contain synchronous labeled frames from both cameras, we did not train with multi-view PCA and only use it post-hoc; see the Methods section.)

Specifically, we use canonical correlations analysis (CCA) to find directions of motion that must match in the left and right cameras (see Methods), and then we quantify the correlation values of these two directions of motion. We find that the correlation values obtained by DeepLabCut can be fairly low in some sessions (r=0.83±0.02, mean±sem over 44 sessions), largely due to occlusions or to frames in which one paw is confused for the other. LP networks improve these correlations slightly (r=0.86±0.02), and the EKS that enforces the multi-view PCA loss pushes these correlation values to 1.0, by construction (Extended Data Fig. 7C,D; see Methods for multi-view EKS details). Next we align trials to movement onset and compute the trial consistency metric introduced above, finding improvements with LP+EKS (Extended Data Fig. 7E,F; DLC 0.07±0.01; LP 0.07±0.01; LP+EKS 0.17±0.02). As in IBL-pupil, we also quantify the correlation between paw speed and neural activity using a simple decoding analysis, and find that LP+EKS but not LP leads to greater decoding accuracy (Extended Data Fig. 7H; DLC R^2=0.23±0.02; LP 0.22±0.02; LP+EKS 0.26±0.02). All quantities reported here refer to the right paw; see Supplementary Table 2 for left paw values, which are qualitatively similar.

References

1. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience* **21,** 1281–1289 (2018).
2. Warren, R. A. *et al.* A rapid whisker-based decision underlying skilled locomotion in mice. *Elife* **10,** e63596 (2021).
3. Syeda, A. *et al.* Facemap: a framework for modeling neural activity based on orofacial tracking. *Nature Neuroscience,* 1–9 (2023).
4. Weinreb, C. *et al.* Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *bioRxiv,* 2023–03 (2023).
5. Karashchuk, P. *et al.* Anipose: a toolkit for robust markerless 3D pose estimation. *Cell reports* **36,** 109730 (2021).
6. Zhang, L., Dunn, T., Marshall, J., Olveczky, B. & Linderman, S. *Animal pose estimation from video data with a hierarchical von Mises-Fisher-Gaussian model International Conference on Artificial Intelligence and Statistics* (2021), 2800–2808.
7. Monsees, A. *et al.* Estimation of skeletal kinematics in freely moving rodents. *Nature Methods* **19,** 1500–1509 (2022).
8. Ebrahimi, A. S. *et al.* Three-dimensional unsupervised probabilistic pose reconstruction (3D-UPPER) for freely moving animals. *Scientific Reports* **13,** 155 (2023).
9. Pereira, T. D. *et al.* SLEAP: A deep learning system for multi-animal pose tracking. *Nature methods* **19,** 486–495 (2022).
10. Lauer, J. *et al.* Multi-animal pose estimation, identification and tracking with DeepLabCut. *Nature Methods* **19,** 496–504 (2022).
11. Abe, T. *et al.* Neuroscience Cloud Analysis As a Service: An open-source platform for scalable, reproducible data analysis. *Neuron* **110,** 2771–2789 (2022).
12. Katzfuss, M., Stroud, J. R. & Wikle, C. K. Understanding the Ensemble Kalman Filter. *The American Statistician* **70,** 350–357 (2016).
13. Kendall, A., Gal, Y. & Cipolla, R. *Multi-task learning using uncertainty to weigh losses for scene geometry and semantics Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 7482–7491.

2 Supplementary Tables

			In-distribution (Train)			Out-of-distribution (Test)		
	FPS	Size	Labeled frames	Animals	Videos/frames	Labeled frames	Animals	Videos/frames
Mirror-mouse	250	406×396	789	10	17/510k	253	3	5/150k
Mirror-fish	300	384×512	373	10	28/55k	94	3	10/22k
CRIM13	30	480×640	3986	4	37/314k	1274	4	19/155k
IBL-paw	60/150	102×128	6071	35	84/187k	1446	10	19/90k
IBL-pupil	60/150	100×100	2619	26	52/279k	1012	8	8/72k

Supplementary Table 1: **Dataset details**. In-distribution (InD) frames are selected from one set of animals/videos; out-of-distribution (OOD) frames are selected from a non-overlapping subset of animals/videos. The number of videos is generally larger than the number of animals, indicating multiple experimental sessions from some animals. The total number of unlabeled frames is also included under "Videos/frames" (rounded to the nearest thousand). See text for IBL details.
Frame size is (height × width).

		Vert vs Horz diameter r	Trial consistency	Decoding R^2
IBL-pupil	DLC	0.36±0.03	0.35±0.06	0.27±0.02
	LP	0.88±0.01	0.62±0.07	0.33±0.02
	LP+EKS	1.00±0.00*	0.74±0.08	0.35±0.02

		CCA projection r	Trial consistency	Decoding R^2
IBL-paw (left)	DLC	0.85±0.02	0.10±0.01	0.22±0.02
	LP	0.88±0.01	0.11±0.01	0.23±0.02
	LP+EKS	1.00±0.00*	0.14±0.01	0.25±0.02
IBL-paw (right)	DLC	0.83±0.02	0.07±0.01	0.23±0.02
	LP	0.86±0.02	0.07±0.01	0.22±0.02
	LP+EKS	1.00±0.00*	0.17±0.02	0.26±0.02

Supplementary Table 2: **Model performance on IBL data**. We quantify pose estimation performance on the IBL data using a range of metrics. For more details on how the metrics are computed, see main text Fig. 6 (pupil) and Extended Data Fig. 7 (paw). An asterisk (*) indicates values that are fixed to 1.0 by model construction. Values are mean and standard error of the mean computed over n=65 sessions for the pupil, and n=44 sessions for the paws.

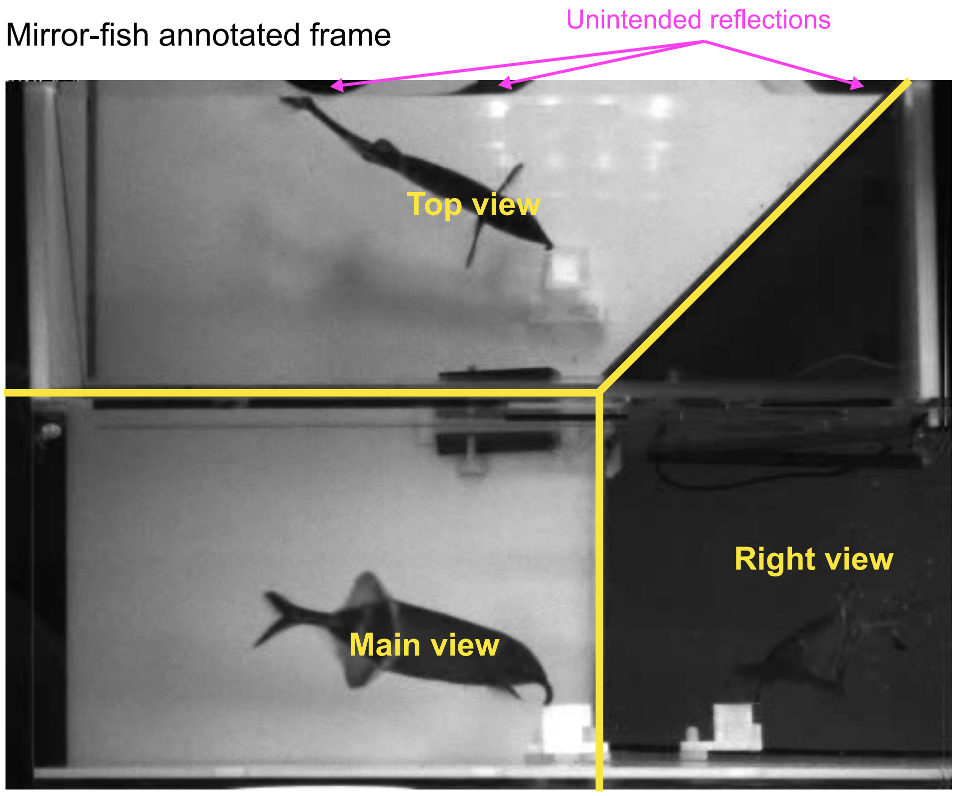
		Single Frame Models		Context (TCN)	
	Height × Width	Labeled	Unlabeled	Labeled (+ Context)	Unlabeled
Mirror-mouse	256×256	8	32	8 (+32)	16
Mirror-fish	256×384	8	16	8 (+32)	12
CRIM13	256×256	8	32	8 (+32)	16
IBL-paw	128×128	32	64	32 (+128)	32
IBL-pupil	128×128	64	64	64 (+256)	32

Supplementary Table 3: **Batch size details**. Height × Width column shows the dimensions of the *resized* images fed to the network. TCN models with a window of $(2J + 1)$ frames (we use $J = 2$) will result in a 5x larger effective labeled batch size (number of unlabeled context frames in parentheses). In general, TCN's effective labeled batch size is $(2J+1)B$ without unlabeled videos and $(2J+1)B+T$ with unlabeled videos. We intentionally kept labeled batch sizes small so that simple consumer GPUs could train a Single Frame and TCN models with an identical labeled batch size.

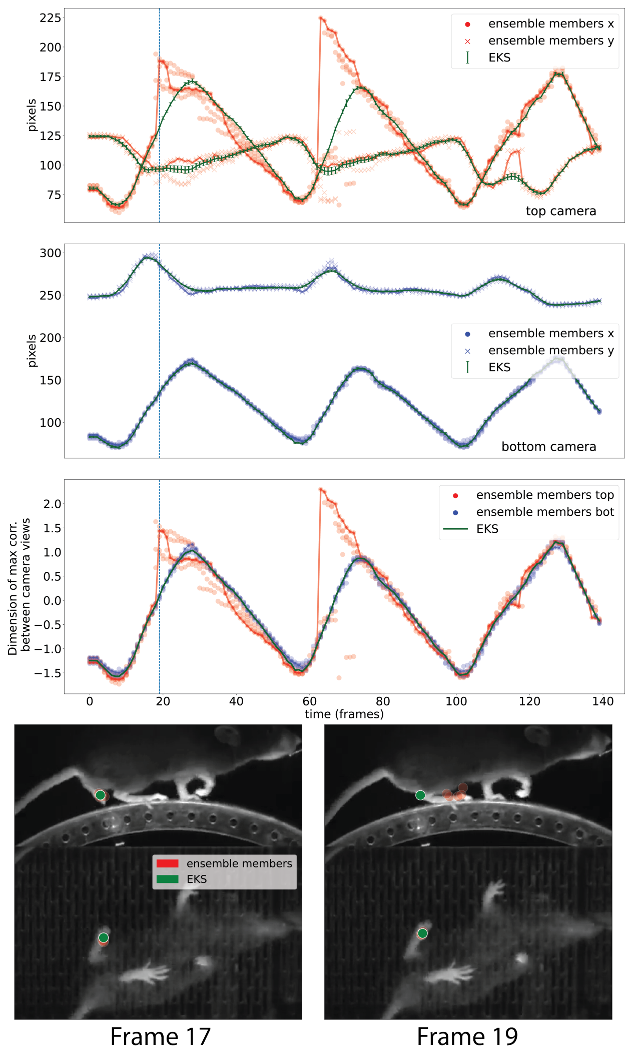
	Temporal		Multi-view PCA		Pose PCA	
Train frames	75	All	75	All	75	All
Mirror-mouse	4.75	4.75	4.5	4.5	4.5	5.0
Mirror-fish	5.5	5.0	6.0	6.0	5.0	6.5
CRIM13	4.5	4.5	-	-	4.5	5.0
IBL-paw	6.0	6.0	-	-	-	-
IBL-pupil	4.5	4.5	-	-	4.0	4.0

Supplementary Table 4: **Hyperparameters for unsupervised losses**. The table presents the log-weights w_l^2 for each of the individual losses $\mathcal{L}_l$, where the total loss is parameterized as sum of terms proportional to Gaussian likelihoods, as in [13]: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{super}} + \sum_{l=1}^{L} \frac{1}{2\exp(w_l^2)} L_l$.

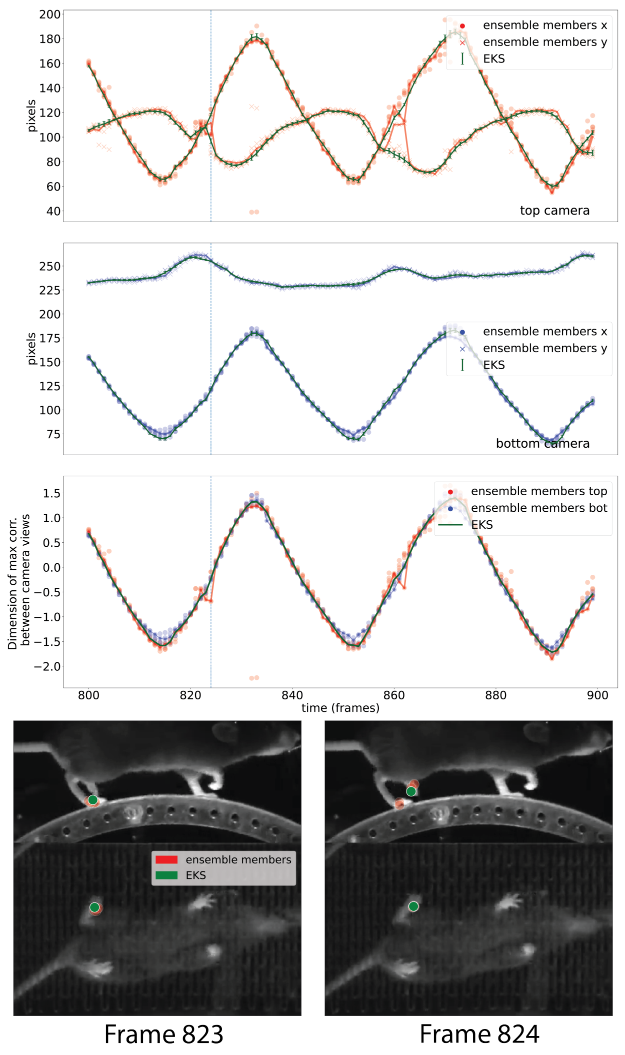
3 Supplementary Figures



Supplementary Figure 1: **Annotated frame from mirror-fish dataset.** The mirror-fish dataset uses a single camera pointed at a tank containing a single fish. The tank contains two mirrors at 45°, allowing the the camera to capture three roughly orthogonal views. Occasionally, a combination of mirror placement and water levels lead to unintended reflections on the top of the frame.

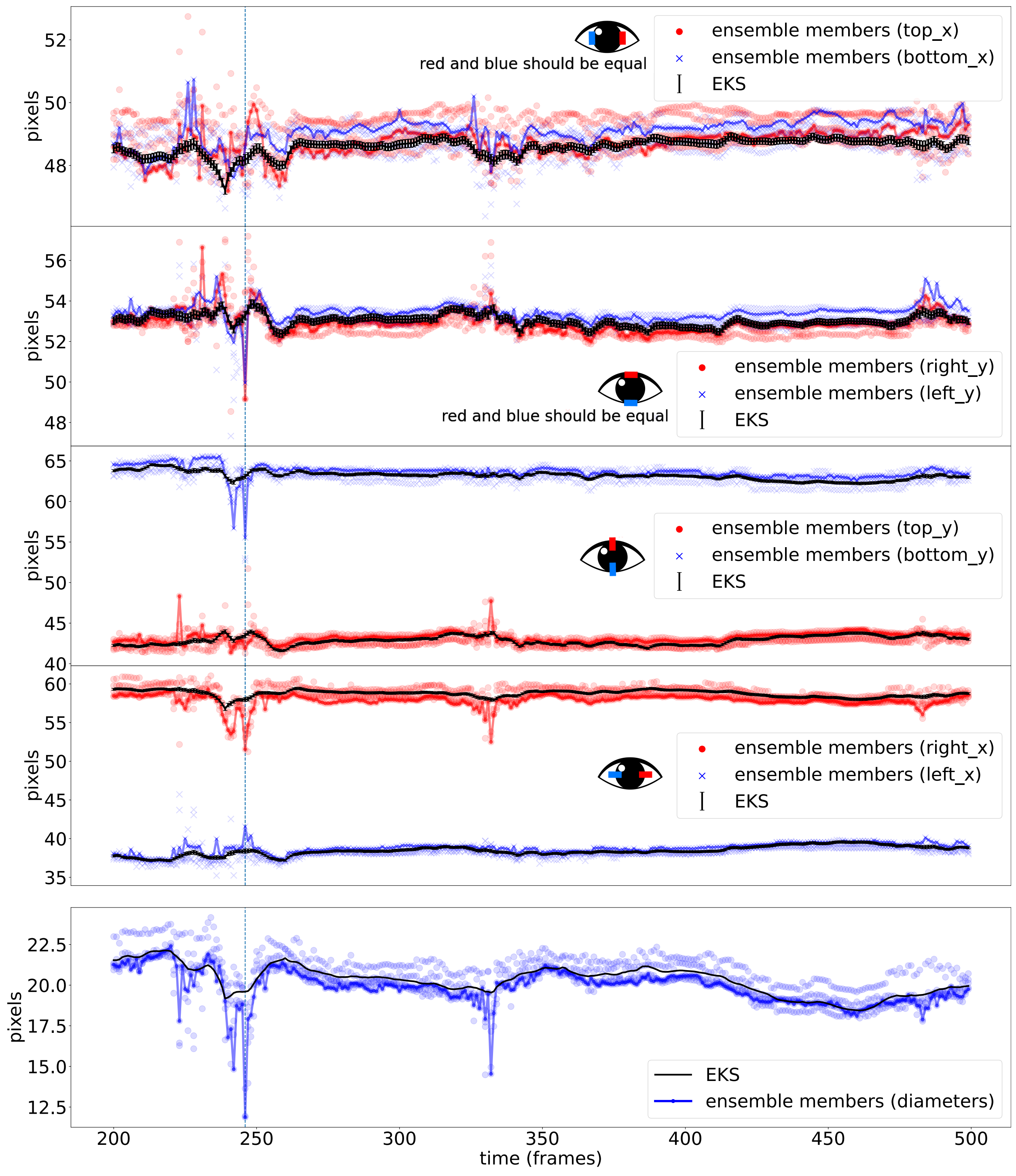


(a) Paw occlusion

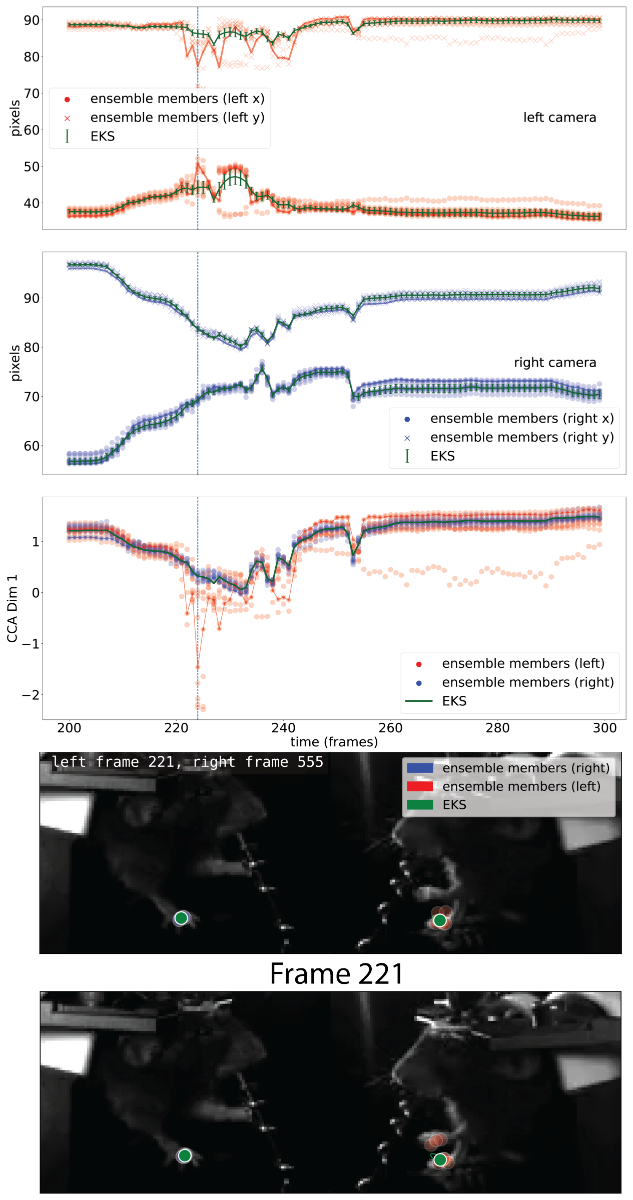


(b) Paw switching

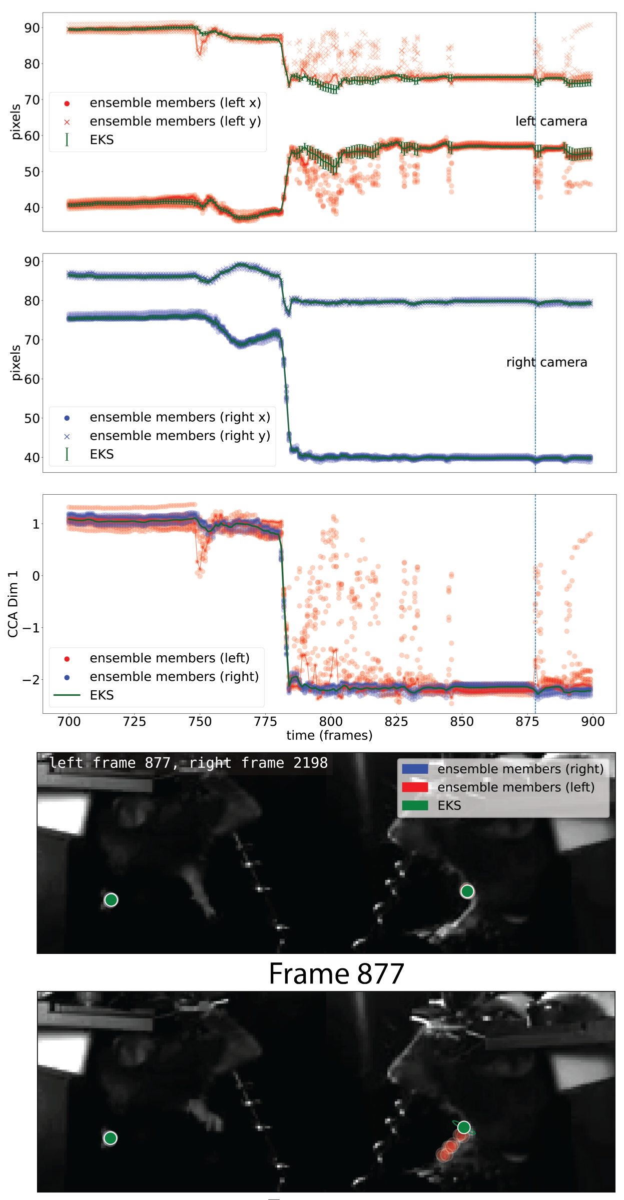
Supplementary Figure 2: **Ensemble Kalman smoothing example with the 75 training frame mirror-mouse dataset.** We use ensembles of $m = 5$ semi-supervised TCN models as input to EKS. The two columns show two illustrative examples. Top panels: x and y coordinates of the left hind paw viewed on the top camera. Conventions as in Supplementary Fig. 3. Second from top: x and y coordinates of the left paw viewed on the bottom camera. Third from top: CCA coordinates computed from the top and bottom camera views. Similarly to the IBL-paw dataset (Supplementary Fig. 4), these CCA coordinates should be equal at each frame. The top view is more challenging (the camera is facing the side of the mouse rather than the bottom, and we are tracking the distant paw here, so more occlusions occur), and the ensemble variance is correspondingly larger for the top view; therefore the EKS tracks the bottom view more closely here. Bottom: example frames (indicated with the vertical dashed line above). In the left column, we see an example of paw occlusion - when the left hind paw goes behind the back of the animal all members of the ensemble jump to the nearest visible keypoint. Tracking is accurate in frame 17 and then the occlusion and ensemble confusion is visible in frame 19. Note that the EKS accurately tracks the correct paw here, since it uses information from the more confident camera view to resolve confusion in the more challenging camera view. In the right column, we see an example in which the EKS is able to correct an error due to paw switching in the indicated frames (823-24). For the full video used here, see Supplementary Video 8.



Supplementary Figure 3: **Ensemble Kalman smoothing example with pupil traces**. The top four panels show the eight tracked coordinates, while the bottom panel shows the estimated diameter. In each panel, the $\times$'s indicate the output of individual ensemble members (semi-supervised TCN models); we have connected the output of a single ensemble member with a solid line across time. The black trace indicates the EKS output. As the pupil keypoints are arranged in a diamond shape, they are paired naturally, e.g. the x coordinate of the top and bottom keypoints should be equal, as should the y coordinate of the left and right keypoint (as indicated by the insets at the bottom right of each of the first four panels). This pairing makes it easy to detect errors: for example, in the first and second panels, any divergence between the red and blue traces are due to tracking errors. Note that tracking errors are common here but are largely resolved by the EKS (an example error is shown using dotted line at frame 246). For the full video used here, see Supplementary Video 9.

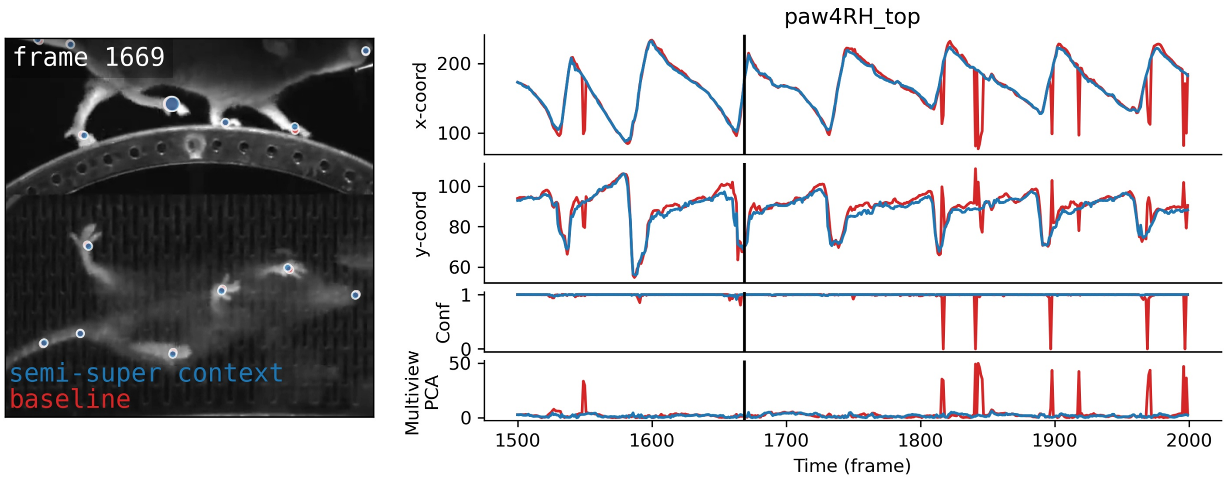


(a) Paw switching

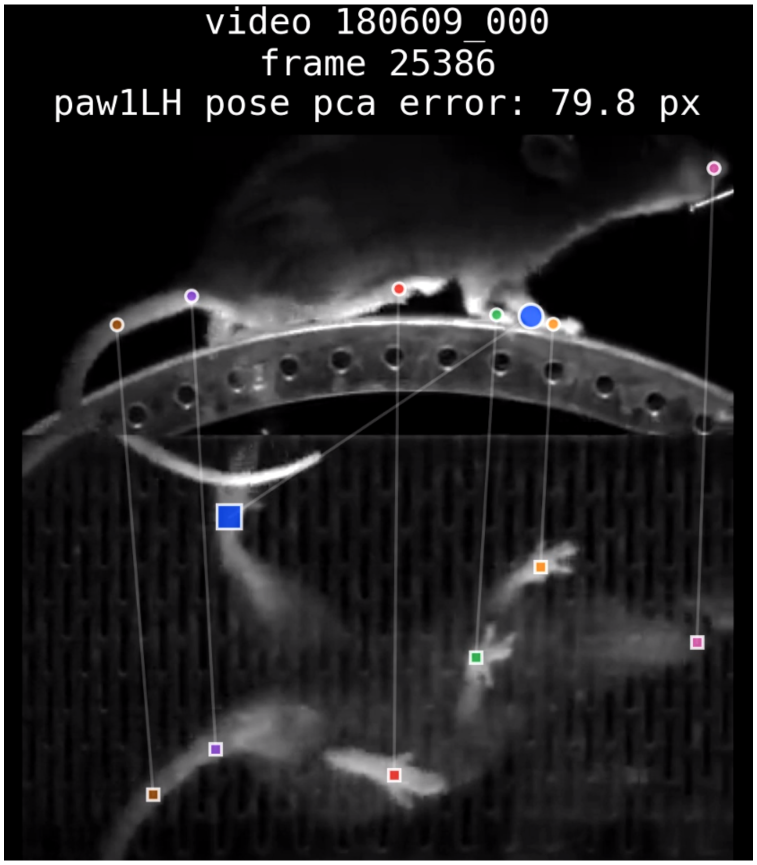


(b) Paw occlusion

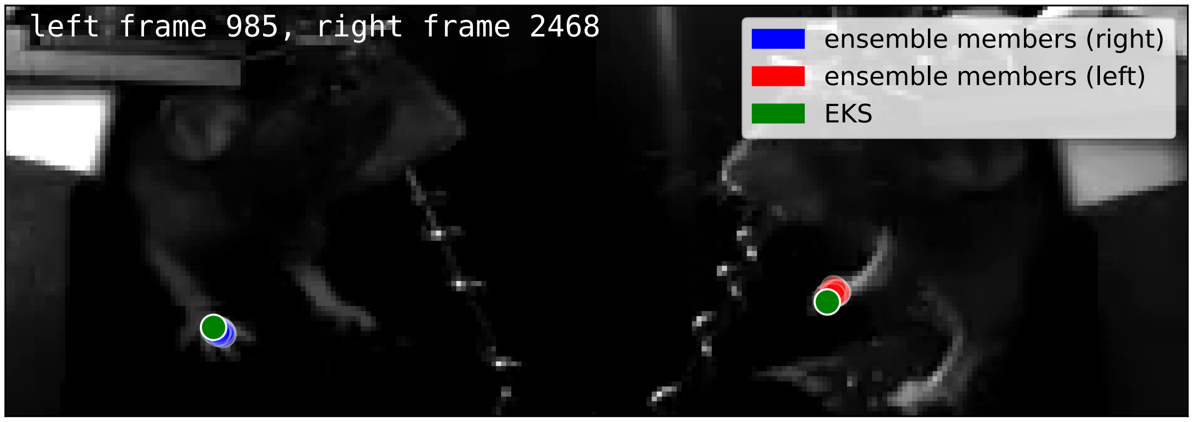
Supplementary Figure 4: **Ensemble Kalman smoothing of baseline supervised models improves pose estimation on IBL-paw data.** The two columns show two illustrative examples. Top panels: x and y coordinates of the left paw viewed on the left camera. Conventions as in Supplementary Fig. 3. Second from top: x and y coordinates of the left paw viewed on the right camera. Third from top: canonical correlation analysis (CCA) coordinates computed from the left and right camera views (see Methods). Due to the geometry of multiple cameras recording the same body parts from different directions, these CCA coordinates should be equal at each frame (this is true for the EKS by construction). The left view is more challenging (the paw is further from the camera and the sampling rate of the video is lower), and the ensemble variance is correspondingly larger for the left view; therefore the EKS tracks the right view more closely here. Bottom: example frames (indicated with the vertical dashed line above). In the left column, we see an example of paw confusion - some members of the ensemble (correctly) track one paw, and some (mistakenly) track the other. Tracking is accurate in frame 221 and then confusion between the two paws is visible in frame 222. Note that the EKS accurately tracks the correct paw here, since it uses information from the more confident camera view to resolve confusion in the more challenging camera view. In the right column, we see an example in which the EKS is able to correct an error due to paw occlusion in the indicated frames (877-8). The right ensemble is highly confident so it is tightly packed behind the ensemble-kalman prediction. For the full video used here, see Supplementary Video 10.

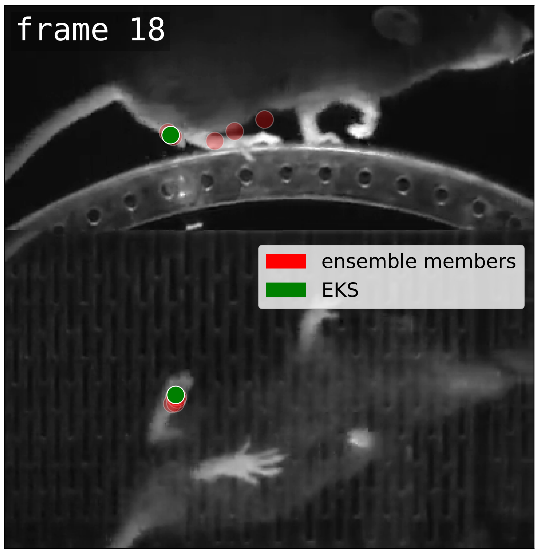


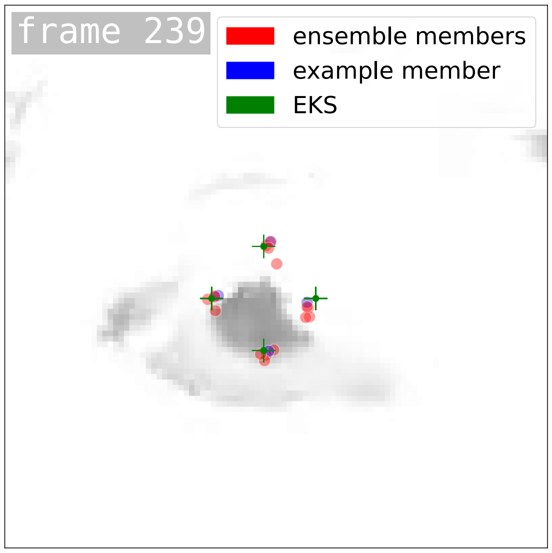
Supplementary Figure 5: **Baseline vs semi-supervised context model predictions.** *Left*: Model predictions (with no confidence filtering) for a snippet of OOD video, for both the fully supervised baseline model (red) and the semi-supervised context model (blue). The larger marker is the one depicted in the traces. *Right*: x- and y-coordinates of the larger marker, along with confidence values and the multi-view PCA loss. Vertical black bar marks the current frame in the video.



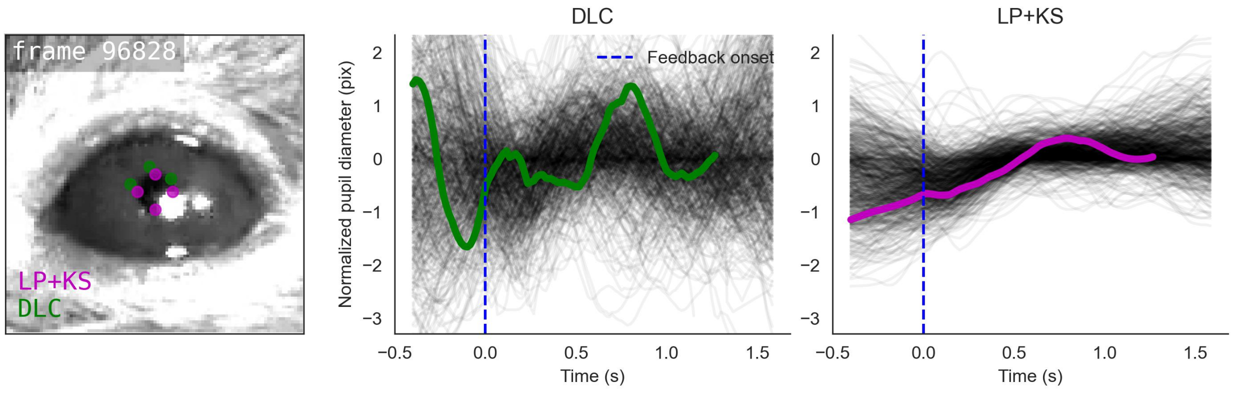
Supplementary Figure 6: **Selected frames with high Pose PCA errors.** The video is composed of 100 unlabeled video frames with high single-body-part Pose PCA errors, averaged across all views in the mirrored datasets (DLC models using all available training frames). The body part in question is indicated in the text of the title, as well as the enlarged markers in the frame. In the mirrored datasets the keypoint is enlarged in all views. Each body part is given a unique marker color, which is shared across views (mirror-mouse, mirror-fish) or animals (CRIM13); marker shape is unique for each view/animal. Skeleton lines highlight prediction errors. The videos demonstrate that high values of the Pose PCA error generally capture erroneous predictions; however, high values are occasionally associated with correct predictions on rare poses that are not included in the PCA subspace. To avoid redundant frames, we choose the 1000 frames with highest Pose PCA errors, then fit k-means on these keypoints using 100 clusters, and select one frame from each cluster to display in the video.

A


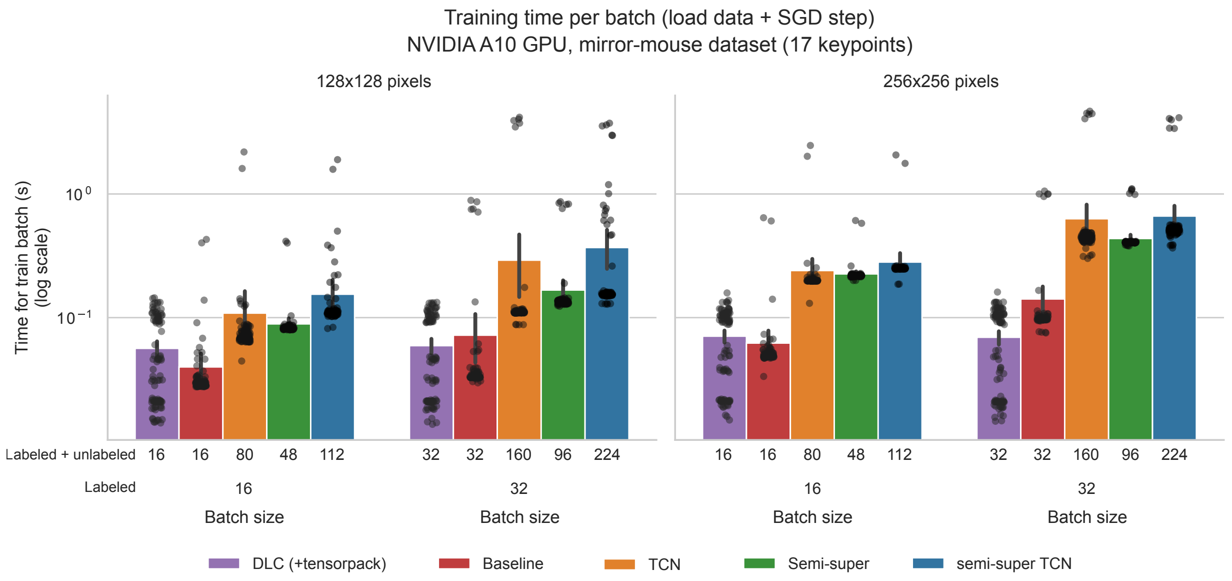
B


C


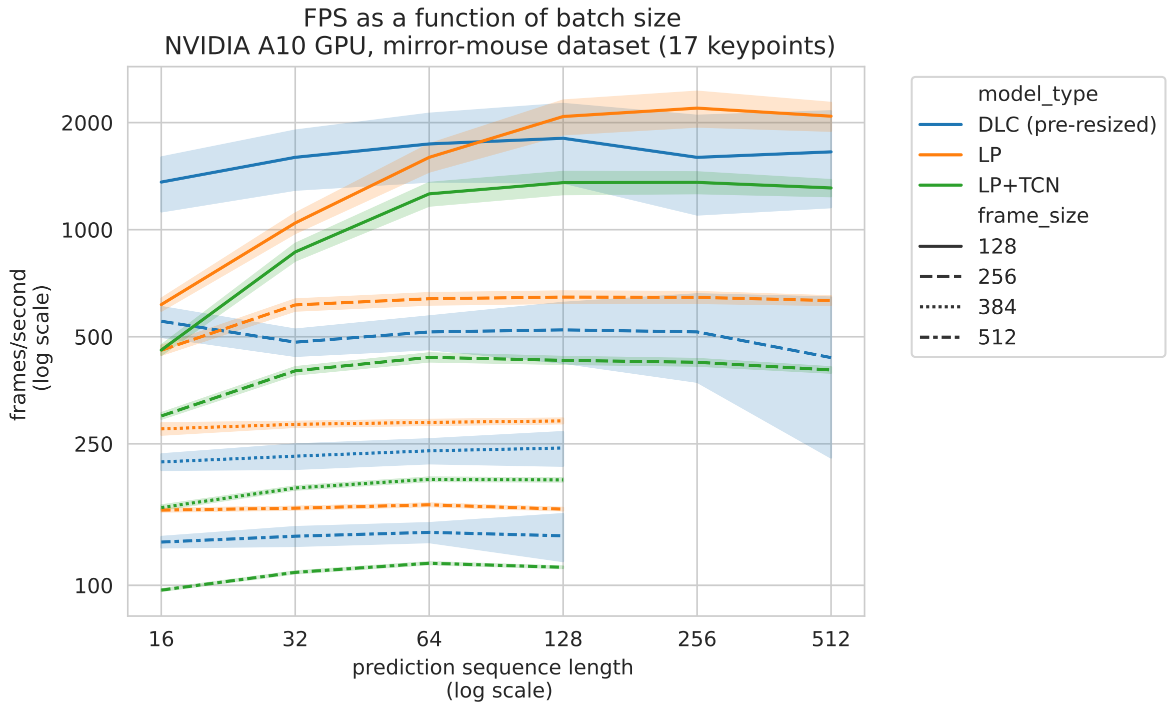
Supplementary Figure 7: **Ensemble kalman model predictions.** **A**: Video frames for IBL-paw dataset from two cameras (right and left) overlaid with predicted markers from ten supervised baseline models (blue and red) and the Ensemble Kalman Smoother (green). The cameras for this dataset have different sampling frequencies so the closest frames are visualized together for this video. All blue points are closely packed together behind the green point. **B**: Video frames for 75 frame mirror-mouse dataset with two camera views (top and bottom) overlaid with predicted markers from five semi-supervsied baseline models (red) and the Ensemble Kalman Smoother (green). **C**: Video frame for IBL-pupil dataset with predicted markers from ten supervised baseline models (red), one example supervised baseline model (blue), and the Ensemble Kalman Smoother (green).



Supplementary Figure 8: **Trial-by-trial markers and traces for IBL datasets. Left**: Video frame overlaid with predicted markers from the DLC model (green) and the ensemble Kalman smoother (LP+EKS; magenta). Markers with confidence <0.9 are not displayed. **Center**: Black traces show smoothed DLC pupil diameter from a subset of correct trials, aligned to feedback onset (reward delivery). The green trace highlights the pupil diameter of the current trial displayed on the left. **Right**: Pupil diameters from the LP+EKS model. For the paw video, instead of displaying per-trial traces in black we display the mean and 95% confidence interval.



Supplementary Figure 9: **Training time per batch.** Each bar depicts the mean batch processing time (in seconds) and 95% CI over n=100 batches, with each of the batches overlaid as a point, using a log-scale spacing for the y-axis. Left panel: 128 × 128 images; right panel: 256 × 256 images. Each panel is divided into a labeled batch size of 16 (left) and 32 (right). The upper x-axis label denotes the total batch size, comprised of both labeled and unlabeled frames.



Supplementary Figure 10: **Prediction throughput.** x-axis: prediction sequence length (frames), log-spaced. y-axis: frames per second, log-spaced. Colors indicate model types, line styles indicate the frame sizes. For image sizes 384 and 512, sequence lengths did not exceed 128 due to GPU memory constraints. Plotted are means ± standard errors across n=5 videos.

4 The International Brain Laboratory consortium members

Larry Abbot, Luigi Acerbi, Valeria Aguillon-Rodriguez, Mandana Ahmadi, Jaweria Amjad, Dora Angelaki, Jaime Arlandis, Zoe C Ashwood, Kush Banga, Hailey Barrell, Hannah M Bayer, Brandon Benson, Julius Benson, Jai Bhagat, Dan Birman, Niccolò Bonacchi, Kcenia Bougrova, Julien Boussard, Sebastian A Bruijns, Robert Campbell, Matteo Carandini, Joana A Catarino, Fanny Cazettes, Gaelle A Chapuis, Anne K Churchland, Yang Dan, Felicia Davatolhagh, Peter Dayan, Sophie Denève, Eric EJ DeWitt, Ling Liang Dong, Tatiana Engel, Michele Fabbri, Mayo Faulkner, Robert Fetcho, Ila Fiete, Charles Findling, Laura Freitas-Silva, Surya Ganguli, Berk Gercek, Naureen Ghani, Ivan Gordeliy, Laura M Haetzel, Kenneth D Harris, Michael Hausser, Naoki Hiratani, Sonja Hofer, Fei Hu, Felix Huber, Julia M Huntenburg, Cole Hurwitz, Anup Khanal, Christopher S Krasniak, Sanjukta Krishnagopal, Michael Krumin, Debottam Kundu, Agnès Landemard, Christopher Langdon, Christopher Langfield, Inês Laranjeira, Peter Latham, Petrina Lau, Hyun Dong Lee, Ari Liu, Zachary F Mainen, Amalia Makri-Cottington, Hernando Martinez-Vergara, Brenna McMannon, Isaiah McRoberts, Guido T Meijer, Maxwell Melin, Leenoy Meshulam, Kim Miller, Nathaniel J Miska, Catalin Mitelut, Zeinab Mohammadi, Thomas Mrsic-Flogel, Masayoshi Murakami, Jean-Paul Noel, Kai Nylund, Farideh Oloomi, Alejandro Pan-Vazquez, Liam Paninski, Alberto Pezzotta, Samuel Piccard, Jonathan W Pillow, Alexandre Pouget, Florian Rau, Cyrille Rossant, Noam Roth, Nicholas A Roy, Kamron Saniee, Rylan Schaeffer, Michael M Schartner, Yanliang Shi, Carolina Soares, Karolina Z Socha, Cristian Soitu, Nicholas A Steinmetz, Karel Svoboda, Marsa Taheri, Charline Tessereau, Anne E Urai, Erdem Varol, Miles J Wells, Steven J West, Matthew R Whiteway, Charles Windolf, Olivier Winter, Ilana Witten, Lauren E Wool, Zekai Xu, Han Yu, Anthony M Zador, Yizi Zhang