CellPress

Neuron

Primer

# A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives

Alexander Mathis,[1,2,3,*] Steffen Schneider,[3,4] Jessy Lauer,[1,2,3] and Mackenzie Weygandt Mathis[1,2,3,*]
[1]Center for Neuroprosthetics, Center for Intelligent Systems, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
[2]Brain Mind Institute, School of Life Sciences, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
[3]The Rowland Institute at Harvard, Harvard University, Cambridge, MA, USA
[4]University of Tübingen and International Max Planck Research School for Intelligent Systems, Tübingen, Germany
*Correspondence: alexander.mathis@epfl.ch (A.M.), mackenzie.mathis@epfl.ch (M.W.M.)
https://doi.org/10.1016/j.neuron.2020.09.017

## SUMMARY

Extracting behavioral measurements non-invasively from video is stymied by the fact that it is a hard computational problem. Recent advances in deep learning have tremendously advanced our ability to predict posture directly from videos, which has quickly impacted neuroscience and biology more broadly. In this primer, we review the budding field of motion capture with deep learning. In particular, we will discuss the principles of those novel algorithms, highlight their potential as well as pitfalls for experimentalists, and provide a glimpse into the future.

## INTRODUCTION

The pursuit of methods to robustly and accurately measure animal behavior is at least as old as the scientific study of behavior itself (Klette and Tee, 2008). Trails of hominid footprints, "motion" captured by Pliocene deposits at Laetoli that date to 3.66 million years ago, firmly established that early hominoids achieved an upright, bipedal, and free-striding gait (Leakey and Hay, 1979). Beyond fossilized locomotion, behavior can now be measured in a myriad of ways: from GPS trackers, videography, and microphones to tailored electronic sensors (Kays et al., 2015; Brown et al., 2013; Camomilla et al., 2018). Videography is perhaps the most general and widely used method, because it allows noninvasive, high-resolution observations of behavior (Johansson, 1973; O'Connell et al., 2010; Weinstein, 2018). Extracting behavioral measures from video poses a challenging computational problem. Recent advances in deep learning have tremendously simplified this process (Wu et al., 2020; Mathis and Mathis, 2020), which quickly impacted neuroscience (Mathis and Mathis, 2020; Datta et al., 2019).

In this primer, we review markerless (animal) motion capture with deep learning. In particular, we review principles of algorithms, highlight their potential, as well as discuss pitfalls for experimentalists and compare them to alternative methods (inertial sensors, markers, etc.). Throughout, we also provide glossaries of relevant terms from deep learning and hardware. Furthermore, we will discuss how to use these deep learning-based motion capture tools, what pitfalls to avoid, and provide perspectives on what we believe will and should happen next.

What do we mean by "markerless motion capture"? Although biological movement can also be captured by dense or surface models (Mathis and Mathis, 2020; Güler et al., 2018; Zuffi et al., 2016), here, we will almost exclusively focus on "keypoint-based pose estimation." Human and many other animals motions are determined by the geometric structures formed by several pendulum-like motions of the extremities relative to a joint (Johansson, 1973). Seminal psychophysics studies by Johansson (1973) showed that just a few coherently moving keypoints are sufficient to be perceived as human motion. This empirically highlights why pose estimation is a great summary of such video data. Which keypoints should be extracted, of course, dramatically depends on the model organism and the goal of the study (e.g., many are required for dense, 3D models) (Güler et al., 2018; Sanakoyeu et al., 2020; Zuffi et al., 2016), whereas a single point can suffice for analyzing some behaviors (Mathis and Mathis, 2020). One of the great advantages of deep learning-based methods is that they are very flexible, and the user can define what should be tracked.

### Principles of Deep Learning Methods for Markerless Motion Capture

In raw video, we acquire a collection of pixels that are static in their location and have varying value over time. For analyzing behavior, this representation is sub-optimal: instead, we are interested in properties of objects in the images, such as location, scale, and orientation. Objects are collections of pixels in the video moving or being changed in conjunction. By decomposing objects into keypoints with semantic meaning—such as body parts in videos of human or animal subjects—a high-dimensional video signal can be converted into a collection of time series describing the movement of each keypoint (Figure 1). Compared to raw video, this representation is easy to analyze and semantically meaningful for investigating behavior and addressing the original research question for which the data have been recorded.

Motion capture systems aim to infer keypoints from videos: in marker-based systems, this can be achieved by manually enhancing parts of interest (by colors, LEDs, reflective markers),

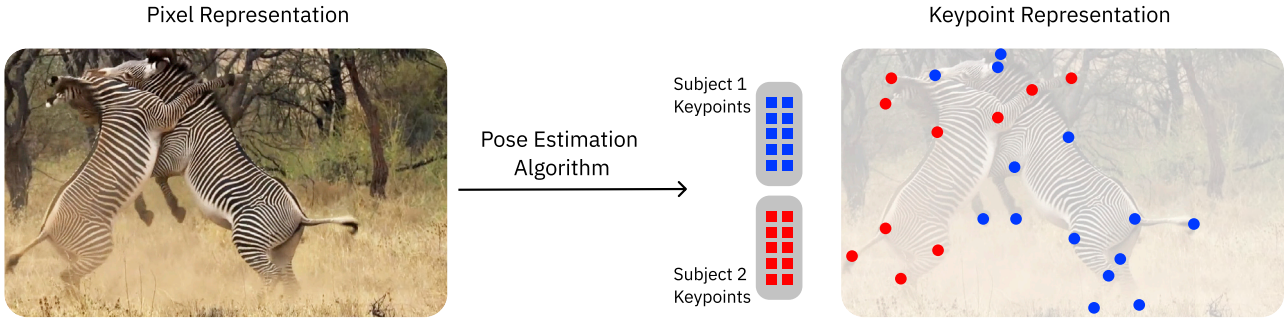



**Figure 1. Schematic Overview of Markerless Motion Capture or Pose Estimation**
The pixel representation of an image (left) or sequence of images (video) is processed and converted into a list of keypoints (right). Semantic information about object identity and keypoint type is associated to the predictions. For instance, the keypoints are structures with a name (e.g., ear), the x and y coordinates, as well as a confidence readout of the network (often this is included, but not for all pose estimation packages), and are then grouped according to individuals (subjects).

which greatly simplifies the computer vision challenge, and then using classical computer vision tools to extract these keypoints. Markerless pose estimation algorithms directly map raw video input to these coordinates. The conceptual difference between marker-based and markerless approaches is that the former requires special preparation or equipment, whereas the latter can even be applied post hoc but typically requires ground truth annotations of example images (i.e., a training set). Notably, markerless methods allow for extracting additional keypoints at a later stage, something that is not possible with markers (Figure 2).

Fundamentally, a pose estimation algorithm can be viewed as a function that maps frames from a video into the coordinates of body parts. The algorithms are highly flexible with regard to what body parts are tracked. Typically, the identity of the body parts (or objects) have semantically defined meaning (e.g., different finger knuckles, the head), and the algorithms can group them accordingly (namely, to assemble an individual) so that the posture of multiple individuals can be extracted simultaneously (Figure 1). For instance, for an image of one human the algorithm would return a list of pixel coordinates (these can have subpixel resolution) per body part and frame (and sometimes an uncertainty prediction) (Insafutdinov et al., 2016; Kreiss et al., 2019; Mathis et al., 2018). The body parts returned by the algorithm depend on both the application and the training data provided—this is an important aspect with respect to how the algorithms can be customized for applications.

## Overview of Algorithms

Although many pose estimation algorithms (Moeslund et al., 2006; Poppe, 2007) have been proposed, algorithms based on deep learning (LeCun et al., 2015) are the most powerful as measured by performance on human pose estimation benchmarks (Toshev and Szegedy, 2013; Jain et al., 2014; Insafutdinov et al., 2016; Newell et al., 2016; Cao et al., 2018; Xiao et al., 2018; Cheng et al., 2020). More generally, pose estimation algorithms fall under "object detection," a field that has seen tremendous advances with deep learning (aptly reviewed in Wu et al., 2020). In brief, pose estimation can often intuitively be understood as a system of an encoder that extracts important (visual) features from the frame, which are then used by the decoder to predict the body parts of interests along with their location in the image frame.

In classical algorithms (see Moeslund et al., 2006; Poppe, 2007; Wu et al., 2020), handcrafted feature representations are used that extract invariant statistical descriptions from images. These features were then used together with a classifier (decoder) for detecting complex objects like humans (Dalal and Triggs, 2005; Moeslund et al., 2006). Handcrafted feature representations are (loosely) inspired by neurons in the visual pathway and are designed to be robust to changes in illumination and translations; typical feature representations are scale invariant feature transform (SIFT) (Lowe, 2004), histogram of gradients (HOG) (Dalal and Triggs, 2005), or speeded up robust features (SURF) (Bay et al., 2008).

In more recent approaches, both the encoder and decoders (alternatively called the backbone and output heads, respectively) are deep neural networks (DNN) that are directly optimized on the pose estimation task. An optimal strategy for pose estimation is jointly learning representations of the raw image or video data (encoder) and a predictive model for posture (decoder). In practice, this is achieved by concatenating multiple layers of differentiable, non-linear transformations and by training such a model as a whole using the back-propagation algorithm (LeCun et al., 2015; Goodfellow et al., 2016; Wu et al., 2020). In contrast to classical approaches, DNN-based approaches directly optimize the feature representation in a way most suitable for the task at hand (for a glossary of deep learning terms, see Box 1).

Machine learning systems are composed of a dataset, model, loss function (criterion), and optimization algorithm (Goodfellow et al., 2016). The dataset defines the input-output relationships that the model should learn: in pose estimation, a particular pose (output) should be predicted for a particular image (input) (Figures 1 and 2B). The model's parameters (weights) are iteratively updated by the optimizer to minimize the loss function. Thereby the loss function measures the quality of a predicted pose (in comparison to the ground truth data). Choices about these four parts influence the final performance and behavior of the pose-estimation system, and we discuss possible design choices in the next sections.

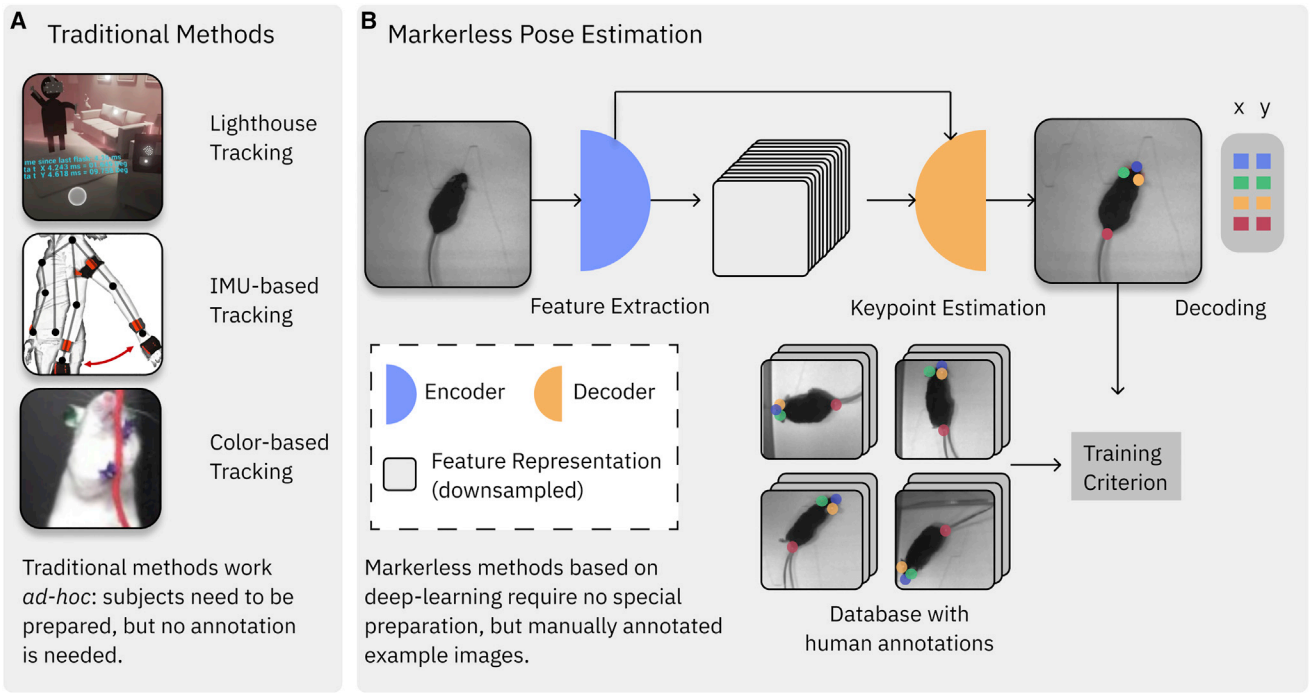



**Figure 2. Comparison of Marker-Based (Traditional) and Markerless Tracking Approaches**

(A) In marker-based tracking, prior to performing an experiment, special measures have to be taken regarding hardware and preparation of the subject (images adapted from Inayat et al. [2020] and Maceira-Elvira et al. [2019]; IMU stands for inertial measurement unit).

(B) For markerless pose estimation, raw video is acquired and processed post hoc: using labels from human annotators, machine learning models are trained to infer keypoint representations directly from video (on-line inference without markers is also possible) (Kane et al., 2020). Typically, the architectures underlying pose estimation can be divided into a feature extractor and a decoder. The former maps the image representation into a feature space and the latter infers keypoint locations given this feature representation. In modern deep learning systems, both parts of the systems are trained end-to-end.

## Datasets and Data Augmentation

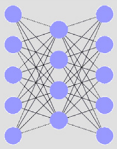
Two kinds of datasets are relevant for training pose estimation systems: first, one or multiple datasets used for related tasks—such as image recognition—can be used for pre-training computer vision models on this task (also known as transfer learning; see Box 1). This dataset is typically considerably larger than the one used for pose estimation. For example, ImageNet (Deng et al., 2009), sometimes denoted as ImageNet-21K, is a highly influential dataset, and a subset was used for the ImageNet Large Scale Visual Recognition Challenge in 2012 (ILSRC-2012) (Russakovsky et al., 2015) for object recognition. The full ImageNet contains 14.2 million images from 21K classes; the ILSRC-2012 subset contains 1.2 million images of 1,000 different classes (such as car, chair, etc.) (Russakovsky et al., 2015). Groups working toward state-of-the-art performance on this benchmark also helped push the field to build better DNNs and openly share code. This dataset has been extensively used for pre-training networks, which we will discuss in the model and optimization section below.

The second highly relevant dataset is the one curated for the task of interest—Mathis et al. (2018) empirically demonstrated that the size of this dataset can be comparably small for typical pose estimation cases in the laboratory. Typically, this dataset contains 10–500 images, versus the standard human pose estimation benchmark datasets, such as MS COCO (Lin et al., 2014) or MPII pose (Andriluka et al., 2014), which has annotated 40,000 images (of 26,000 individuals). This implies that the dataset that is curated is highly influential on the final performance, and great care should be taken to select diverse postures, individuals, and background statistics and labeling the data accurately (discussed below).

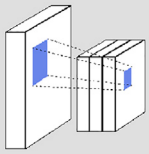
In practice, several factors matter: the performance of a fine-tuned model on the task of interest, the amount of images that need to be annotated for fine-tuning the network, and the convergence rate of optimization algorithm (i.e., how many steps of gradient descent are needed to obtain a certain performance). Using a pre-trained network can help with this in several regards: He et al. (2018) show that in the case of large training datasets, pre-training typically aids with convergence rates, but not necessarily the final performance. Under the right circumstances (i.e., given enough task-relevant data) and with longer training, randomly initialized models can match the performance of fine-tuned ones for key point detection on COCO (He et al., 2018) and horses (Mathis et al., 2019); however, the networks are less robust (Mathis et al., 2019). Beyond robustness, using a pre-trained model is generally advisable when the amount of labeled data for the target task is small, which is true for many applications in neuroscience, because it leads to shorter training times and better performance with less data (He et al., 2018; Mathis et al., 2018, 2019; Arac et al., 2019). Thus, pre-trained pose estimation algorithms save training time, increase robustness, and require

**Box 1. Glossary of Deep Learning Terms**

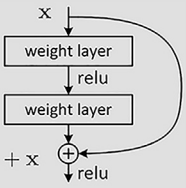
An excellent textbook for Deep Learning is provided by Goodfellow et al. (2016). See Dumoulin and Visin (2016) for an in-depth technical overview of convolution arithmetic.



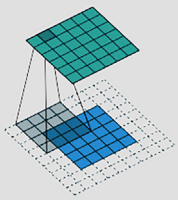
**Artificial neural network (ANN):** an ANN can be represented by a collection of computational units ("neurons") arranged in a directed graph. The output of each unit is computed as a weighted sum of its inputs, followed by a nonlinear function.



**Convolutional neural network (CNN):** A CNN is an ANN composed of one or multiple convolutional layers. Influential early CNNs are the LeNet, AlexNet and VGG16 (LeCun et al., 2015; Goodfellow et al., 2016; Wu et al., 2020).

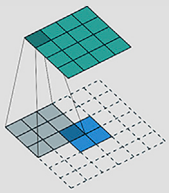



**Residual networks (ResNets):** increasing network depth makes deep ANNs (DNNs) more expressive compared to adding units to a shallow architecture. However, optimization becomes hard for standard convolutional neural networks (CNNs) beyond 20 layers, at which point depth in fact decreases the performance (He et al., 2016). In residual networks, instead of learning a mapping $y = f(x)$, the layer is re-parametrized to learn the mapping $y = x + f(x)$, which improves optimization and regularizes the loss landscape (Li et al., 2018). These networks can have much larger depth without diminishing returns (He et al., 2016) and are the basis for other popular architectures such as MobileNets (Sandler et al., 2018) and EfficientNets (Tan and Le, 2019).



**Convolution:** a convolution is a special type of linear filter. Compared with a full linear transformation, convolutional layers increase computational efficiency by weight sharing (LeCun et al., 2015; Goodfellow et al., 2016). By applying the convolution, the same set of weights is used across all locations in the image.

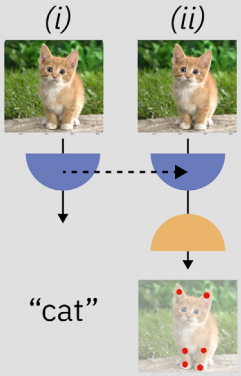
**Deconvolution:** deconvolutional layers allow to upsample a feature representation. Typically, the kernel used for upsampling is optimized during training, similar to a standard convolutional layer. Sometimes, fixed operations such as bilinear upsampling filters are used.



**Stride, downsampling, and dilated (atrous) convolutions:** in DNNs for computer vision, images are presented as real-valued pixel data to the network and are then transformed to symbolic representations and image annotations, such as bounding boxes, segmentation masks, class labels, or keypoints. During processing, inputs are consecutively abstracted by aggregating information from growing "receptive fields." Increasing the receptive field of the unit is possible by different means: increasing the stride of a layer computes outputs only for every *n*-th input and effectively downsamples the input with a learnable filter. Downsampling layers perform the same operation, but with a fixed kernel (e.g., taking the maximum or mean activation across the receptive field). In contrast, atrous or dilated convolutions increase the filter size by adding intermittent zero entries between the learnable filter weights—e.g., for a dilation of 2, a filter with entries (1,2,3) would be converted into (1,0,0,2,0,0,3). This allows increases in the receptive field without losing resolution in the next layers and is often applied in semantic segmentation algorithms (Chen and Ramanan, 2017) and pose estimation (Insafutdinov et al., 2016; Mathis et al., 2018).

*(Continued on next page)*

**Box 1. *Continued***




**Transfer learning:** the ability to use parameters from a network that has been trained on one task—e.g. classification, see (*i*) as part of a network to perform another task—e.g., pose estimation, see *(ii)*. The approach was popularized with DeCAF (Donahue et al., 2014), which used AlexNet (Krizhevsky et al., 2012) to extract features to achieve excellent results for several computer vision tasks. Transfer learning generally improves the convergence speed of model training (He et al., 2018; Zamir et al., 2018; Arac et al., 2019; Mathis et al., 2019) and model robustness compared to training from scratch (Mathis et al., 2019).

substantially less training data. Indeed, most packages in neuroscience now use pre-trained models (Mathis et al., 2018, 2020; Graving et al., 2019; Arac et al., 2019; Bala et al., 2020; Liu et al., 2020), although some do not (Pereira et al., 2019; Günel et al., 2019; Zimmermann et al., 2020) and yet can give acceptable performance for simplified situations with aligned individuals.

More recently, larger datasets like the 3.5 billion Instagram dataset (Mahajan et al., 2018) JFT (with 300 million images) (Hinton et al., 2015; Xie et al., 2020) and OpenImages (Kuznetsova et al., 2018) have become popular, further improving performance and robustness of the considered models (Xie et al., 2020). What task is used for pre-training also matters. Corroborating this insight, Li et al. (2019) showed that pre-training on a large-scale object detection task can improve performance for tasks that require fine spatial information like segmentation.

Besides large datasets for pre-training, a curated dataset with pose annotations is needed for optimizing the algorithm on the pose estimation task. The process is discussed in more detail below and it typically suffices to label a few (diverse) frames. Data augmentation is the process of expanding the training set by applying specified manipulations (like rotating or scaling image size). Based on the chosen corruptions, models become more invariant to rotations, scale changes, or translations and thus more accurate (with less training data). Augmentation can also help with improving robustness to noise, like jpeg compression artifacts and motion blur (Figure 3). Of note, data augmentation schemes should not affect the semantic information in the image: for instance, if color conveys important information about the identity of an animal, augmentations involving changes in color are not advisable. Likewise, augmentations that change the spatial position of objects or subjects should always be applied to both the input image and the labels (Box 2).

## Model Architectures

Systems for markerless pose estimation are typically composed of a backbone network (encoder), which takes the role of the feature extractor, and one or multiple heads (decoders). Understanding the model architectures and design choices common in deep learning-based pose estimation systems requires basic understanding of convolutional neural networks. We summarize the key terms in Box 1 and expand on what encoders and decoders are below.

Instead of using handcrafted features as in classical systems, deep learning-based systems employ "generic" encoder architectures that are often based on models for object recognition. In a typical system, the encoder design affects the most important properties of the algorithms such as its inference speed, training-data requirements, and memory demands. For the pose estimation algorithms so far used in neuroscience, the encoders are either stacked hourglass networks (Newell et al., 2016), MobileNetV2s (Sandler et al., 2018), ResNets (He et al., 2016), DenseNets (Huang et al., 2017), or EfficientNets (Tan and Le, 2019). These encoder networks are typically pre-trained on one or multiple of the larger-scale datasets introduced previously (such as ImageNet), because this has been shown to be an advantage for pose estimation on small lab-scale-sized datasets (Mathis et al., 2019, 2018; Arac et al., 2019). For common architectures, this pre-training step does not need to be carried out explicitly, as trained weights for popular architectures are already available in common deep learning frameworks.

The impact of the encoder on DNN performance is a highly active research area. The encoders are continuously improved in speed and object recognition performance (Huang et al., 2017; Sandler et al., 2018; Tan and Le, 2019; Wu et al., 2020; Kornblith et al., 2019). Naturally, due to the importance of the ImageNet benchmark, the accuracy of network architectures continuously increases (on that dataset). For example, we were able to show that this performance increase is not merely reserved for ImageNet, or (importantly) other object recognition tasks (Kornblith et al., 2019), but in fact that better architectures on ImageNet are also better for pose estimation (Mathis et al., 2020). However, being better on ImageNet also comes at the cost of decreasing inference speed and increased memory demands. DeepLabCut (an open source toolbox for markerless pose estimation popular in neuroscience) thus incorporates backbones from MobileNetV2s (faster) to EfficientNets (best performance on ImageNet) (Mathis et al., 2019, 2020).

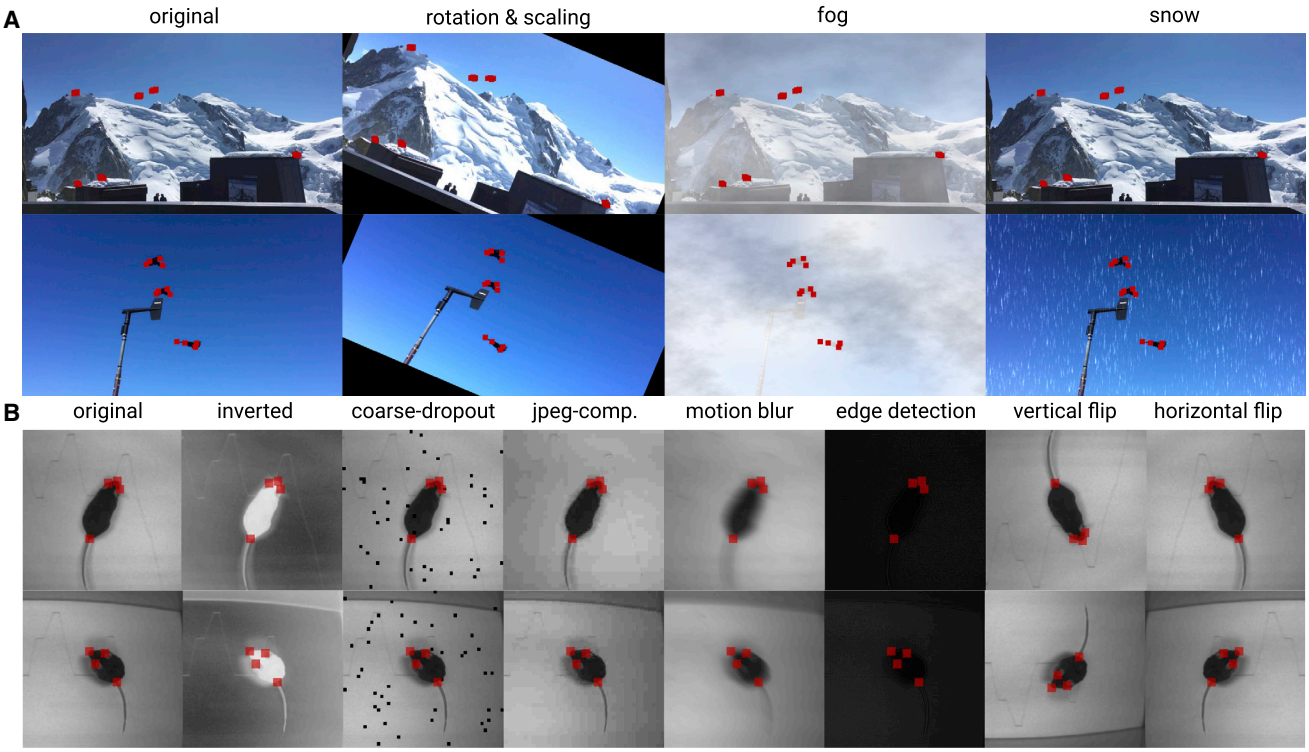



**Figure 3. Example Augmentation Images with Labeled Body Parts in Red**

(A) Two example frames of Alpine choughs (*Pyrrhocorax graculus*) near Mont Blanc with human-applied labels in red (original). The images to the right illustrate three augmentations (title denotes the type of augmentation).

(B) Two example frames of a trail-tracking mouse (*Mus musculus*) from Mathis et al. (2018) with four labeled body parts as well as augmented variants. Open in Google Colaboratory: https://colab.research.google.com/github/DeepLabCut/Primer-MotionCapture/blob/master/COLAB_Primer_MotionCapture_Fig3.ipynb

## Convolutional Neural Network (CNN)

A CNN is an ANN composed of one or multiple convolutional layers. Influential early CNNs are the LeNet, AlexNet, and VGG16 (LeCun et al., 2015; Goodfellow et al., 2016; Wu et al., 2020).

In (standard) convolutional encoders, the high-resolution input images get gradually downsampled while the number of learned features increases. Regression-based approaches that directly predict keypoint locations from the feature representation can potentially deal with this downsampled representation. When the learning problem is instead cast as identifying the keypoint locations on a grid of pixels, the output resolution needs to be increased first, often by deconvolutional layers (Insafutdinov et al., 2016; Xiao et al., 2018). We denote this part of the network as the decoder, which takes downsampled features, possibly from multiple layers in the encoder hierarchy, and gradually upsamples them again to arrive at the desired resolution. The first models of this class were fully convolutional networks (Long et al., 2015), and later DeepLab (Chen and Ramanan, 2017). Many popular architectures today follow similar principles. Design choices include the use of skip connections between decoder layers, but also skip connections between the encoder and decoder layers. Example encoder-decoder setups are illustrated in Figure 4. The aforementioned building blocks—encoders and decoders—can be used to form a variety of different approaches, which can be trained end-to-end directly on the target task (i.e., pose estimation).

Pre-trained models can also be adapted to a particular application. For instance, DeeperCut (Insafutdinov et al., 2016), which was adapted by the animal pose estimation toolbox DeepLabCut (Mathis et al., 2018), was built with a ResNet (He et al., 2016) backbone network, but adapted the stride by atrous convolutions (Chen et al., 2018) to retain a higher spatial resolution (Box 1). This allowed larger receptive fields for predictions while retaining a relatively high speed (i.e., for video analysis), but most importantly, because ResNets can be pre-trained on ImageNet, those initialized weights could be used. Other architectures, like stacked hourglass networks (Newell et al., 2016) used in DeepFly3D (Günel et al., 2019) and DeepPoseKit (Graving et al., 2019), retain feature representations at multiple scales and pass those to the decoder (Figures 4A and 4B).

## Loss Functions: Training Architectures on Datasets

Keypoints (i.e., body parts) are simply coordinates in image space. There are two fundamentally different ways for estimating keypoints (i.e., how to define the loss function). The problem can be treated as a regression problem with the coordinates as targets (Toshev and Szegedy, 2013; Carreira et al., 2016). Alternatively, and more popular, the problem can be cast as a classification problem, where the coordinates are mapped onto a grid (e.g., of the same size as the image) and the model predicts a heatmap (scoremap) of location probabilities for each body part (Figure 4C). In contrast to the regression approach (Toshev

**Box 2. Key Parameters and Choices**

The key design choices of pose estimation systems are dataset curation, data augmentation, model architecture selection, optimization process, and the optimization criterions.

- **Data augmentation:** the technique of increasing the training set by converting images and annotations into new, altered images via geometric transformations (e.g., rotation, scaling, ...), image manipulations (e.g., contrast, brightness, …), etc. (Figure 3). Depending on the annotation data, various augmentations (i.e., rotation symmetry, etc.) are ideal. Packages such as Tensorpack (Wu, 2016) and imgaug (Jung et al., 2020) as well as tools native to PyTorch (Paszke et al., 2019) and TensorFlow (Abadi et al., 2016) provide common augmentation methods and are used in many packages.
- **Model architecture:** users should select an architecture that is accurate and fast (enough) for their goal. Top performing networks (in terms of accuracy) include Stacked Hourglass (Newell et al., 2016), ResNets (He et al., 2016), and EfficientNets (Tan and Le, 2019) with appropriate decoders (Insafutdinov et al., 2016; Xiao et al., 2018; Kreiss et al., 2019; Mathis et al., 2020) as well as recent high-resolution nets (Sun et al., 2019; Cheng et al., 2020). Performance gains in speed at the expense of slightly worse accuracy are possible with (optimized) lightweight models such as MobileNetV2 (Sandler et al., 2018) in DeepLabCut (Mathis et al., 2019) and stacked hourglass networks with DenseNets (Huang et al., 2017) as proposed in DeepPoseKit (Graving et al., 2019); often this performance gap can be rescued with good data augmentation.

and Szegedy, 2013), this is fully convolutional, allows modeling of multi-modal distributions, and aids the training process (Tompson et al., 2014; Newell et al., 2016; Insafutdinov et al., 2016; Cao et al., 2018). Moreover, the heatmaps have the advantage that one can naturally predict multiple locations of the "same" body part in the same image (i.e., two elbows) without mode collapse (Figure 5A).

Loss functions can also reflect additional priors or inductive biases about the data. For instance, DeepLabCut uses location refinement layers (locref), which counteract the downsampling inherent in encoders, by training outputs to predict corrective shifts in image coordinates relative to the downsampled output maps (Figure 5A). In pose estimation, it is possible to define a skeleton or graph connecting keypoints belonging to subjects with the same identity (see below) (Insafutdinov et al., 2016; Cao et al., 2018). When estimating keypoints over time, it is also possible to employ temporal information and encourage the model to only smoothly vary its estimate among consecutive frames (Insafutdinov et al., 2017; Yao et al., 2019; Xu et al., 2020; Zhou et al., 2020). Based on the problem, these priors can be directly encoded and be used to regularize the model.

How can pose estimation algorithms accommodate multiple individuals? Fundamentally, there are two different approaches: bottom-up and top-down methods (Figure 5). In top-down methods, individuals are first localized (often with another neural network trained on object localization), then pose estimation is performed per localized individual (Xiao et al., 2018; Newell et al., 2016; Sun et al., 2019). In bottom-up methods, all body parts are localized, and networks are also trained to predict connections of body parts within individuals (i.e., limbs). These connections are then used to link candidate body parts to form individuals (Cao et al., 2018; Insafutdinov et al., 2017; Kreiss et al., 2019; Cheng et al., 2020). Of note, these techniques can be used on single individuals for increased performance, but often are not needed and usually imply reduced inference speed.

## Optimization

For pre-training, stochastic gradient descent (SGD) (Bottou, 2010) with momentum (Sutskever et al., 2013) is an established method. Different variants of SGD are now common (such as Adam) (Kingma and Ba, 2015) and used for fine-tuning the resulting representations. As mentioned above, pose estimation algorithms are typically trained in a multi-stage setup in which the backbone is trained first on a large (labeled) dataset of a potentially unrelated task (like image classification). Users can also download these pre-trained weights. Afterward, the model is fine-tuned on the pose-estimation task. Once trained, the quality of the prediction can be judged in terms of the root-mean-square error (RMSE), which measures the distance between the ground truth keypoints and predictions (Mathis et al., 2018; Pereira et al., 2019), or by measuring the percentage of correct keypoints (PCK) (Andriluka et al., 2014; Mathis et al., 2019) (i.e., the fraction of detected keypoints that fall within a defined distance of the ground truth).

To properly estimate model performance in an application setting, it is advisable to split the labeled dataset at least into train and test subsets. If systematic deviations can be expected in the application setting (e.g., because the subjects used for training the model differ in appearance from subjects encountered at model deployment) (Mathis et al., 2019), this should be reflected when choosing a way to split the data. For instance, if data from multiple individuals is possible, distinct individuals should form distinct subsets of the data. On the contrary, strategies like splitting data by selecting every *n*-th frame in a video likely overestimates the true model performance.

The model is then optimized on the training dataset, while performance is monitored on the validation (test) split. If needed, hyperparameters—like parameter settings of the optimizer or also choices about the model architecture—of the model can be adapted based on an additional validation set.

All of the aforementioned choices influence the final outcome and performance of the algorithm. While some parts of the training pipeline are well-established and robust—like pre-training a model on ImageNet—choices about the dataset, architecture, augmentation, fine-tuning procedure, etc. will inevitably influence the quality of the pose estimation algorithm (Box 2). See Figure 3 for a qualitative impression of augmentation effects of some of these decisions (see also Figure 8). We will discuss this in more detail in the Pitfalls section.

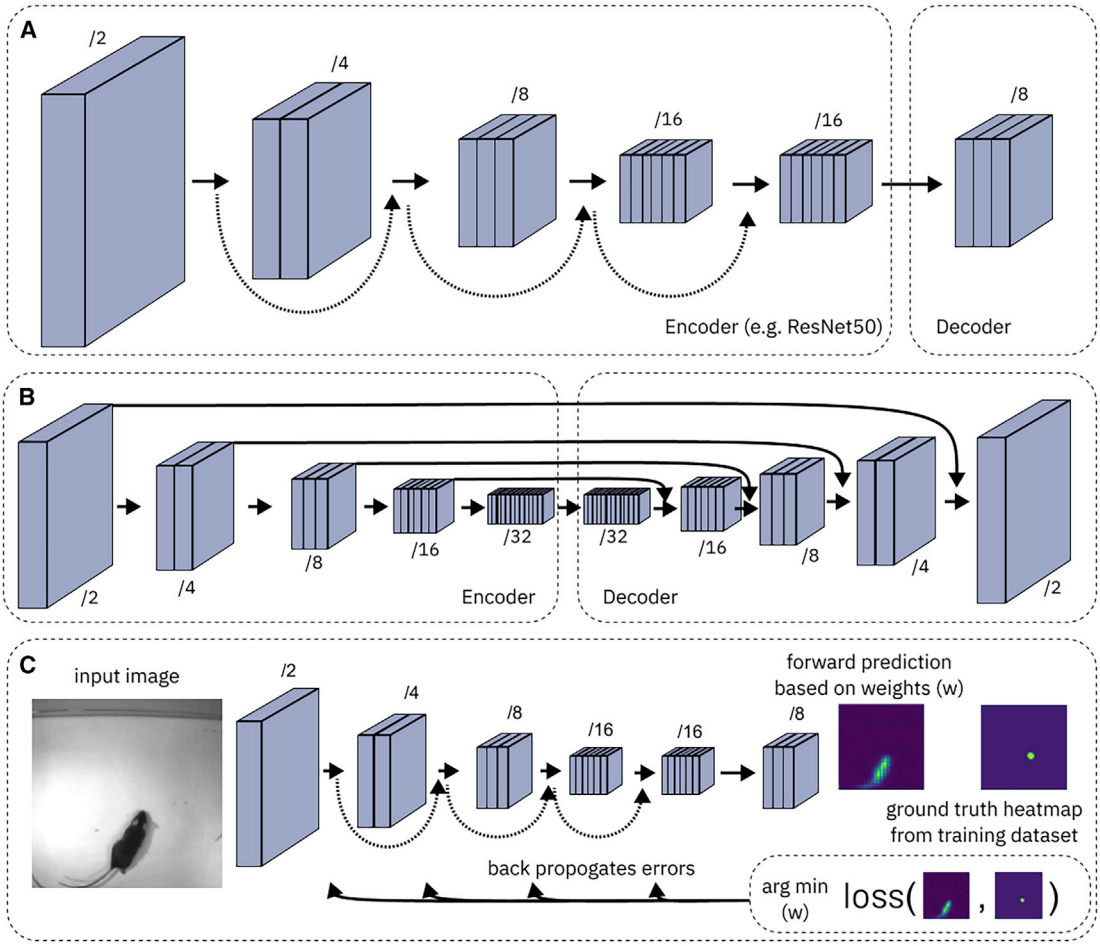



**Figure 4. Schematic Overview of Possible Design Choices for Model Architectures and Training Process**

(A) A simple but powerful variant (Insafutdinov et al., 2016) is a ResNet-50 (He et al., 2016) architecture adapted to replace the final downsampling operations by atrous convolutions (Chen et al., 2018) to keep a stride of 16, and then a single deconvolution layer to upsample to output maps with stride 8. It also forms the basis of other architectures (Xiao et al., 2018). The encoder can also be exchanged for different backbones to improve speed or accuracy (see Box 2).

(B) Other approaches like stacked hourglass networks (Newell et al., 2016) are not pre-trained and employ skip connections between encoder and decoder layers to aid the upsampling process.

(C) For training the network, the training data comprising input images and target heatmaps is used. The target heatmap is compared with the forward prediction. Thereby, the parameters of the network are optimized to minimize the loss that measures the difference between the predicted heatmap and the target heatmap (ground truth).

So far, we considered algorithms able to infer 2D keypoints from videos by training deep neural networks on previously labeled data. Naturally, there is also much work in computer vision and machine learning toward the estimation of 3D keypoints from 2D labels, or to directly infer 3D keypoints. In the interest of space, we do not cover 3D pose estimation, but refer the interested reader to Martinez et al. (2017), Mehta et al. (2016), Tomè et al. (2017), Chen and Ramanan (2017), and Yao et al. (2019) as well as, specifically for neuroscience, Yao et al. (2019), Günel et al. (2019), Nath et al. (2019), Zimmermann et al. (2020), Karashchuk et al. (2020), and Bala et al. (2020).

Lastly, it is not understood how CNNs make decisions, and they often find "shortcuts" (Geirhos et al., 2020). While this active research area is certainly beyond the scope of this primer, from practical experience we know that at least within-domain (i.e., data that is similar to the training set) DNNs work very well for pose estimation, which is the typical setting relevant for downstream applications in neuroscience. It is worth noting that in order to optimize performance, there is no one-size-fits-all solution. Thus, we hope by building intuition in users of such systems, we provide the necessary tools to make these decisions with more confidence (Figure 6).

## SCOPE AND APPLICATIONS

Markerless motion capture can excel in complicated scenes, with diverse animals, and with any camera available (monochrome, RGB, depth cameras, etc.). The only real requirement is the ability of the human to be able to reliably label keypoints (manually or via alternative sources). Simply, you need to be able to see what you want to track. Historically, due to limitations

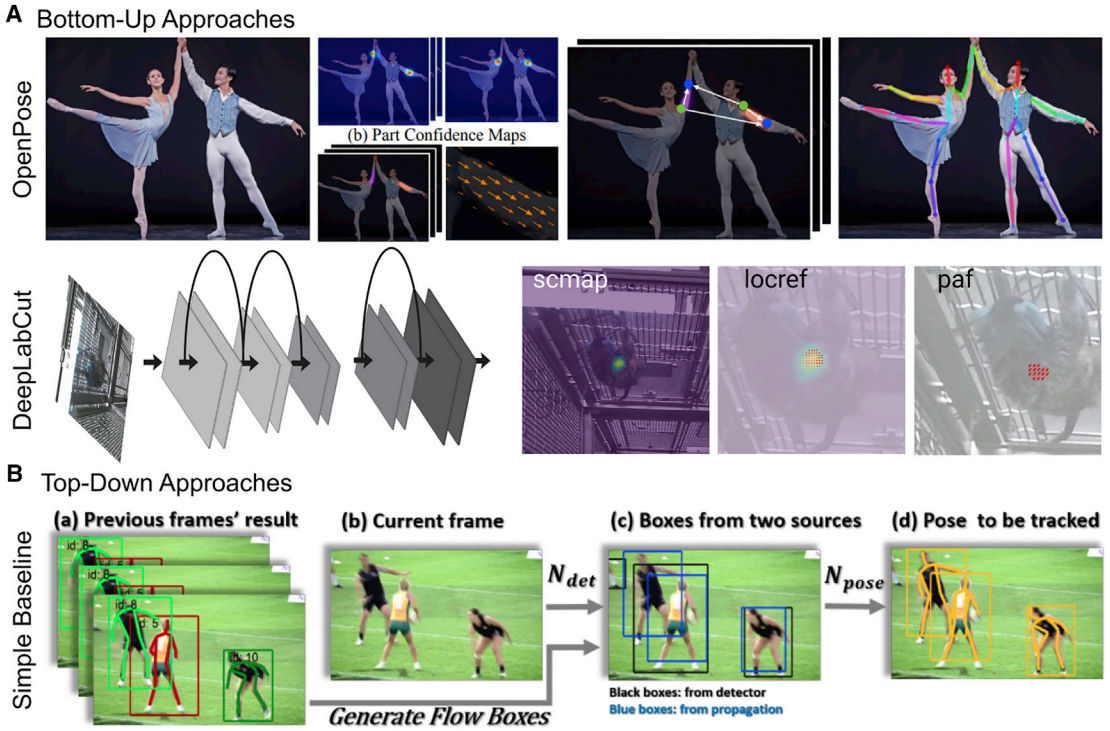



**Figure 5. Multi-Animal Pose Estimation Approaches**

(A) Bottom-up approaches detect all the body parts (e.g., elbow and shoulder in example) as well as "limbs" (part confidence maps). These limbs are then used to associate the body parts within individuals correctly (top rowfrom OpenPose, Cao et al. [2018]; bottom row from DeepLabCut [v2.2, unpublished]). For both OpenPose and DeepLabCut (v2.2), the body parts, part confidence maps (pafs) are predicted as different decoders (also known as output heads) from the encoder.

(B) Top-down approaches localize individuals with bounding-box detectors and then directly predict the posture within each bounding box. This does not require part confidence maps but is subject to errors when bounding boxes are wrongly predicted (see black bounding box encompassing two players in subpael c). The displayed figures, adapted from Xiao et al. (2018), improved this disadvantage by predicting bounding boxes per frame and forward predicting them across time via visual flow.

in computer vision algorithms, experimentalists would go to great lengths to simplify the environment, even in the laboratory (i.e., no bedding, white or black walls, high contrast), and this is no longer required with deep learning-based pose estimation. Now, the aesthetics one might want for photographs or videos taken in daily life are the best option.

Indeed, the field has been able to rapidly adopt these tools for neuroscience. Deep learning-based markerless pose estimation applications in the laboratory have already been published for flies (Mathis et al., 2018; Pereira et al., 2019; Graving et al., 2019; Günel et al., 2019; Karashchuk et al., 2020; Liu et al., 2020), rodents (Mathis et al., 2018; Mathis and Warren, 2018; Pereira et al., 2019; Graving et al., 2019; Arac et al., 2019; Günel et al., 2019; Zimmermann et al., 2020; Liu et al., 2020), horses (Mathis et al., 2019), dogs (Yao et al., 2019), rhesus macaques (Berger et al., 2020; Yao et al., 2019; Bala et al., 2020; Labuguen et al., 2020), and marmosets (Ebina et al., 2019); the original architectures were developed for humans (Insafutdinov et al., 2016; Newell et al., 2016; Cao et al., 2018). Outside of the laboratory, DeepPoseKit was used for zebras (Graving et al., 2019) and DeepLabCut for 3D tracking of cheetahs (Nath et al., 2019), for squirrels (Barrett et al., 2020), and for macaques (Labuguen et al., 2020), highlighting the great "in-the-wild" utility of this new technology (Mathis and Mathis, 2020). As outlined in the principles section and illustrated by these applications, these deep learning architectures are general-purpose and can be broadly applied to any animal and/or condition.

Recent research highlights the prevalent representations of action across the brain (Kaplan and Zimmer, 2020), which emphasizes the importance of quantifying behavior even in non-motor tasks. For instance, pose estimation tools have recently been used to elucidate the neural variability across cortex in humans during thousands of spontaneous reach movements (Peterson et al., 2020). Pupil tracking is of great importance for visual neuroscience. One recent study by Meyer et al. used head-fixed cameras and DeepLabCut to reveal two distinct types of coupling between eye and head movements (Meyer et al., 2020). In order to accurately correlate neural activity to visual input, tracking the gaze is crucial. The recent large open dataset from the Allen Institute includes imaging data of six cortical and two thalamic regions in response to various stimuli classes as well as pupil tracking with DeepLabCut (Siegle et al., 2019). The International Brain Lab has integrated DeepLabCut into their workflow to track multiple body parts of decision-making mice including their pupils (Harris et al., 2019).

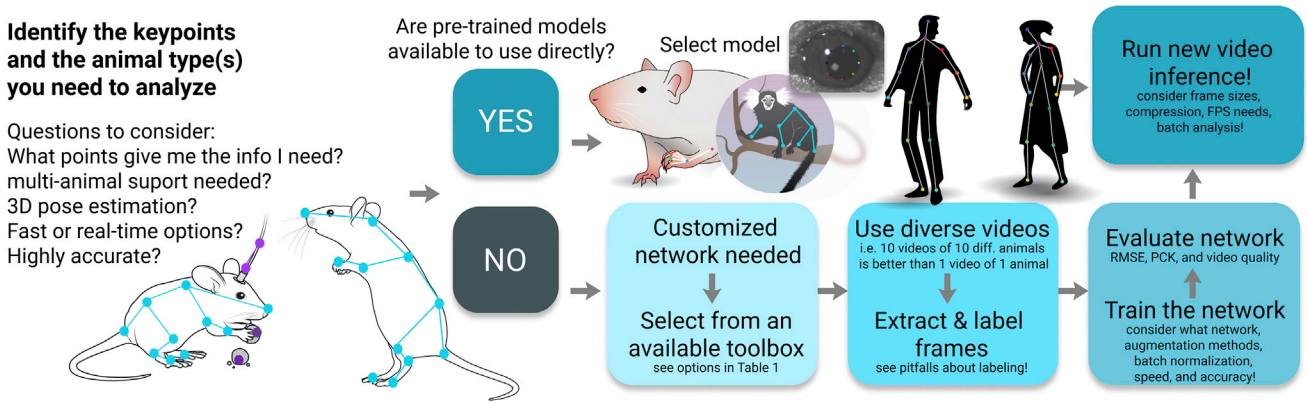



**Figure 6. An Overview of the Workflow for Deep Learning-Based Pose Estimation, which Highlights Several Critical Decision Points**

Measuring relational interactions is another major direction that has been explored less in the literature so far but is feasible. Because the feature detectors for pose estimation are of general nature, one can easily not only track the posture of individuals but also the tools and objects one interacts with (e.g., for analyzing golf or tennis). Furthermore, social behaviors and parenting interactions (for example in mice) can now be studied noninvasively.

Due to the general capabilities, these tools have several applications for creating biomarkers by extracting high-fidelity animal traits (e.g., in the pain field) (Tracey et al., 2019) and for monitoring motor function in healthy and diseased conditions (Micera et al., 2020).

DeepLabCut was also integrated with tools for X-ray analysis (Laurence-Chasen et al., 2020). For measuring joint center locations in mammals, arguably, X-ray is the gold standard. Of course, X-ray data also pose challenges for extracting body part locations. A recent paper shared methodology to integrate DeepLabCut with XROMM, a popular analysis suite, to advance the speed and accuracy for X-ray-based analysis (Laurence-Chasen et al., 2020).

## HOW DO THE (CURRENT) PACKAGES WORK?

Here, we will focus on packages that have been used in behavioral neuroscience, but the general workflow for pose estimation in computer vision research is highly similar. What has made experimentalist-focused toolboxes different is that they provide essential code to generate and train on one's own datasets. Typically, what is available in computer vision-focused pose estimation repositories is code to run inference (video analysis) and/or run training of an architecture for specific datasets around which competitions happen (e.g., MS COCO [Lin et al., 2014] and MPII pose [Andriluka et al., 2014]). Although these are two crucial steps, they are not sufficient to develop tailored neural networks for an individual lab or experimentalist. Thus, the "barrier to entry" is often quite high to use these tools. It requires knowledge of deep learning languages to build appropriate data loaders, data augmentation pipelines, and training regimes. Therefore, in recent years several packages have focused not only on animal pose estimation networks but on providing users a full pipeline that allows for (1) labeling a customized dataset (frame selection and labeling tools), (2) generating test/train datasets, (3) data augmentation and loaders, (4) neural architectures, (5) code to evaluate performance, (6) run video inference, and (7) post-processing tools for simple readouts of the acquired machine-labeled data.

Thus far, around 10 packages have become available in the past 2 years (Mathis et al., 2018; Pereira et al., 2019; Graving et al., 2019; Günel et al., 2019; Arac et al., 2019; Zimmermann et al., 2020; Bala et al., 2020; Liu et al., 2020). Each has focused on providing slightly different user experiences, modularity, available networks, and balances to the speed/accuracy tradeoff for video inference. Several include their (adapted) implementations of the original DeepLabCut or LEAP networks as well (Graving et al., 2019; Liu et al., 2020). However, the ones we highlight have the full pipeline delineated above as a principle and are open source (i.e., at minimum inference code is available) (see Table 1). The progress gained and challenges they set out to address (and some that remain) are reviewed elsewhere (Mathis and Mathis, 2020; Seethapathi et al., 2019). Here, we discuss collective aims of these packages (see also Figure 6).

Current packages for animal pose estimation have focused primarily on providing tools to train tailored neural networks to user-defined features. Because experimentalists need flexibility and are tracking very different animals and features, the most successful packages (in terms of user base as measured by citations and GitHub engagement) are species agnostic. However, given they are all based on advances from prior art in human pose estimation, the accuracy of any one package given the breadth of options that could be deployed (i.e., data augmentation, training schedules, and architectures) will remain largely comparable, if such tools are provided to the user. What will determine performance the most is the input training data provided and the capacity of the architectures.

It is notable that using transfer learning has proven to be advantageous for better robustness (i.e., its ability to generalize) (Mathis et al., 2018, 2019; Arac et al., 2019), which was first deployed by DeepLabCut (see Table 1). Now, training on large animal-specific datasets has been made available in DeepLabCut

**Table 1. Overview of Popular Deep Learning Tools for Animal Motion Capture**

| | Any Species | 3D | >1 Animal | Training Code | Full GUI | Ex. Data | PT-NNs | Released | Citations |
|---|---|---|---|---|---|---|---|---|---|
| DeepLabCut[a] | yes | yes | yes | yes | yes | yes | many | 4/2018 | 491 |
| LEAP[b] | yes | no | yes | yes | yes | yes | no | 6/2018 | 98 |
| DeepBehavior[c] | no | yes | yes | no | no | no | no | 5/2019 | 15 |
| DeepPoseKit[d] | yes | no | no | yes | partial | yes | no | 8/2019 | 48 |
| DeepFly3D[e] | no | yes | no | 2D only | partial | yes | fly | 5/2019 | 21 |
| FreiPose[f] | no | yes | no | partial | no | yes | no | 2/2020 | 1 |
| Optiflex[g] | yes | no | no | yes | partial | yes | no | 5/2020 | 0 |

Here, we denote if each tool can be used to create tailored networks or if only specific animal tools are provided (i.e., only work "as-is" on a fly or rat). We only highlight if beyond human pre-trained neural networks (PT-NNs) are available. We also provide the release date of code and current citations for noted references, including those to related preprints (indexed from Google Scholar in early September 2020). Note: LEAP is deprecated and supplanted by sleap.

[a]Mathis et al., 2018; Nath et al., 2019; [b]Pereira et al., 2019; [c]Arac et al., 2019; [d]Graving et al., 2019; [e]Günel et al., 2019; [f]Zimmermann et al., 2020; [g]Liu et al., 2020

as well (e.g., a horse pose dataset with >8,000 annotated images of 30 horses) (Mathis et al., 2019). This allows the user to bypass the only manual part of curating and labeling ground truth data, and these models can directly be used for inference on novel videos. For DeepLabCut, this is an emerging community-driven effort, with external labs already contributing models and data (http://modelzoo.deeplabcut.org/). In the future, having the ability to skip labeling and training and run video inference with robust models will lead to more reproducible and scalable research. For example, as we show in other sections of the primer, if the labeling accuracy is not of a high quality and the data are not diverse enough, then the networks are not able to generalize to so-called "out-of-domain" data. If, as a community, we collectively build stable and robust models that leverage the breadth of behaviors being carried out in laboratories worldwide, we can work toward models that would work in a plug-in-play fashion. We anticipate new datasets and models to become available in the next months to years.

All packages, just like all applications of deep learning to video, prefer access to GPU computing resources (see Box 3). On GPUs, one experiences faster training and inference times, but the code can also be deployed on standard CPUs or laptops. With cloud computing services, such as Google Colaboratory and JupyterLab, many pose estimation packages can simply be deployed on remote GPU resources. This still requires (1) knowledge about these resources and (2) toolboxes providing so-called "notebooks" that can be easily deployed. However, given these platforms have utility beyond just pose estimation, they are worthwhile to learn about.

For the non-GPU aspects, only a few packages have provided easy-to-use graphical user interfaces that allow users with no programming experience to use the tool (see Table 1). Last, the available packages vary in their access to 3D tools, multi-animal support, and types of architectures available to the user, which is often a concern for speed and accuracy. Additionally, some packages have limitations such as allowing only the same-sized videos for training and inference, whereas others are more flexible. These are all key considerations when deciding which ecosystem to invest in learning (as every package has taken a different approach to the API).

Perhaps the largest barrier to entry for using deep learning-based pose estimation methods is managing the computing resources (see Boxes 3 and 4). From our experience, installing GPU drivers and the deep learning packages (TensorFlow and PyTorch) that all the packages rely on is the biggest challenge. To this end, in addition to documentation that is "user-focused" (i.e., not just an API for programmers), resources like webinars, video tutorials, workshops, Gitter, and community-forums (like StackOverflow and Image Forum SC) have become invaluable resources for the modern neuroscientist. Here, users can ask questions and get assistance from developers and users alike. We believe this has also been a crucial step for the success of DeepLabCut.

Although some packages provide full GUI-based control over the packages, to utilize more advanced features at least minimal programming knowledge is ideal. Thus, better training for the increasingly computational nature of neuroscience will be crucial. Making programming skills a requirement of graduate training, building better community resources, and leveraging the fast-moving world of technology to harness those computing and user resources will be crucial. In animal pose estimation, although there is certainly an attempt to make many of the packages user-friendly (i.e., to onboard users) and have a scalable discussion around common problems, we found user forums to be very valuable (Rueden et al., 2019). Specifically, DeepLabCut is a member of the Scientific Community Image Forum (https://forum.image.sc) alongside other packages that are widely used for image analysis in the life sciences such as Fiji (Schindelin et al., 2012), napari, CellProfiler (McQuin et al., 2018), Ilastik (Sommer et al., 2011), and scikit-image (van der Walt et al., 2014).

## PRACTICAL CONSIDERATIONS FOR POSE ESTIMATION (WITH DEEP LEARNING)

As a recent field gaining traction, it is instructive to regard the operability of deep learning-powered pose estimation in light of well-established, often gold standard, techniques.

### General Considerations and Pitfalls

As discussed in Scope and Applications and as evidenced by the strong adaptation of the tools, deep learning-based pose

**Box 3. Computing Hardware**

- **CPU:** the central processing unit (CPU) is the core of a computer and executes computer programs. CPUs work well on sequential or lightly parallelized routines due to the limited number of cores.
- **GPU:** a graphical processing unit (GPU) is a specialized computing device designed to rapidly process and alter memory. GPUs are ideal for computer graphics and often located in graphics cards. Their highly parallel architecture enables them to be more efficient (NVIDIA data center deep learning product performance: https://developer.nvidia.com/deep-learning-performance-training-inference) than CPUs for algorithms with many small subroutines that can be launched in parallel. They can be applied to run DNNs at higher speed (Krizhevsky et al., 2012) and pose estimation algorithms in particular (Mathis and Warren, 2018; Mathis et al., 2019; Kane et al., 2020).
- **Affordability of GPUs:** modern GPUs are affordable (around 300–800 USD for cards that can be used for the pose estimation tools mentioned here and up to 10,000 USD for high-end cards) and ideally suited to run video processing within a single lab in a decentralized way. They can be placed into standard desktop computers or even "gaming" laptops. However, to get started it might be easier to test software in cloud computing services first for ease of use (i.e., no driver installation).
- **Cloud computing:** ability to use resources online rapidly (minimal installation) often in a pay-per-use scheme. Two relevant examples are Google Colaboratory and My Binder. Google Colaboratory is an online platform for free GPU use with run times of up to 6 h (https://colab.research.google.com/). My Binder allows turning a Git repository into a collection of interactive notebooks by running them in an executable environment, making your code immediately reproducible by anyone, anywhere (https://mybinder.org).

estimation works well in standard setups with visible animals. The most striking advantage over traditional motion capture systems is the absence of any need for body instrumentation. Although seemingly obvious, the previous statement hides the belated recognition that marker-based motion capture suffers greatly from the wobble of markers placed on the skin surface. That behavior, referred to as "soft tissue artifact" among movement scientists and attributable to the deformation of tissues underneath the skin, such as contracting muscles or fat, is now known to be the major obstacle to obtaining accurate skeletal kinematics (Camomilla et al., 2017). Note: intra-cortical pins and biplane fluoroscopy give direct, uncontaminated access to joint kinematics. The first, however, is invasive (and entails careful surgical procedures) (Ramsey et al., 2003), whereas the second is only operated in very constrained and complex laboratory settings (List et al., 2017). Both are local to a specific joint, and as such, do not strictly address the task of pose estimation. To make matters worse, contaminated marker trajectories may be harmful in clinical contexts, potentially invalidating injury risk assessment (e.g., Smale et al., 2017). Although a multitude of numerical approaches exist to tackle this issue, the most common yet incomplete solution is multi-body kinematics optimization (or "inverse kinematics" in computer graphics and robotics) (Begon et al., 2018). This procedure uses a kinematic model and searches for the body pose that minimizes in the least-squares sense the distance between the measured marker locations and the virtual ones from the model while satisfying the constraints imposed by the various joints (Lu and O'Connor, 1999). Its accuracy is, however, decisively determined by the choice of the underlying model and its fidelity to an individual's functional anatomy (Begon et al., 2018). In contrast, motion capture with deep learning elegantly circumvents the problem by learning a geometry-aware representation of the body from the data to associate keypoints to limbs (Cao et al., 2018; Insafutdinov et al., 2016; Mathis and Mathis, 2020), which, of course, presupposes that one can avoid the "soft tissue artifact" when labeling.

At present, deep learning-powered pose estimation can be poorly suited to evaluate rotation about a bone's longitudinal axis. From early markerless techniques based on visual hull extraction, this is a known problem (Ceseracciu et al., 2014). In marker-based settings, the problem has long been addressed by tracking clusters of at least three non-aligned markers to fully reconstruct a rigid segment's six degrees of freedom (Spoor and Veldpaus, 1980). Performing the equivalent feat in a markerless case is difficult, but it is possible by labeling multiple points (e.g., on either side of the wrist to get the lower-limb orientation). Still, recent hybrid, state-of-the-art approaches jointly training under both position and orientation supervision augur very well for video-based 3D joint angle computation (Xu et al., 2020; Zhou et al., 2020).

With the notable exception of approaches leveraging radio wave signals to predict body poses through walls (Zhao et al., 2018), deep learning-powered motion capture requires the individuals be visible; this is impractical for kinematic measurements over wide areas. A powerful alternative is offered by Inertial Measurement Units (IMUs)—low-cost and lightweight devices typically recording linear accelerations, angular velocities, and the local magnetic field. Raw inertial data can be used for coarse behavior classification across species (Kays et al., 2015; Chakravarty et al., 2019). They can also be integrated to track displacement with lower power consumption and higher temporal resolution than GPS (Bidder et al., 2015), thereby providing a compact and portable way to investigate whole body dynamics (e.g., Wilson et al., 2018) or, indirectly, energetics (Gleiss et al., 2011). Recent advances in miniaturization of electronical components now also allow precise quantification of posture in small animals (Pasquet et al., 2016) and open new avenues for kinematic recordings in multiple animals at once at fine motor scales.

Nonetheless, IMU-based full-body pose reconstruction necessitates multiple sensors over the body parts of interest; commercial solutions require up to 17 of them (Roetenberg et al., 2009). That burden was recently eased by utilizing a statistical

**Box 4. Reproducible Software**

Often installation of deep learning languages like TensorFlow/Keras (Abadi et al., 2016) and PyTorch (Paszke et al., 2019) is the biggest hurdle for getting started.

- **Python virtual environments:** software often has many dependencies, and they can conflict if multiple versions are required for different needs. Thus, placing dependencies within a contained environment can minimize issues. Common environments include Anaconda (conda) and virtualenv, both for Python code bases.
- **Docker:** delivers software in packages called containers, which can be run locally or on servers. Containers are isolated from one another and bundle their own software, libraries, and configuration files (https://docker.com) (Merkel, 2014).
- **GitHub:** https://github.com is a platform for developing and hosting software, which uses Git version control. Version control is excellent to have history-dependent versions and workspaces for code development and deployment. Gitlab (https://gitlab.com/explore) also hosts code repositories.

body model that incorporates anatomical constraints, together with optimizing poses over multiple frames to enforce coherence between the model orientation and IMU recordings—reducing the system down to six sensors while achieving stunning motion tracking (von Marcard et al., 2017). Yet two additional difficulties remain. The first arises when fusing inertial data in order to estimate a sensor's orientation (for a comprehensive description of mathematical formalism and implementation of common fusion algorithms, see Sabatini, 2011). The process is susceptible to magnetic disturbances that distort sensor readings and, consequently, orientation estimates (Fan et al., 2017). The second stems from the necessity to align a sensor's local coordinate system to anatomically meaningful axes, a step crucial (among others) to calculating joint angles (Lebleu et al., 2020). The calibration is ordinarily carried out by having the subject perform a set of predefined movements in sequence whose execution determines the quality of the procedure. Yet in some pathological populations (let alone in animals), calibration may be challenging to say the least, deteriorating pose reconstruction accuracy (Vargas-Valencia et al., 2016).

A compromise to making the task less arduous is to combine videos and body-worn inertial sensors. Thanks to their complementary nature, incorporating both cues mitigates the limitations of each individual system; i.e., both modalities reinforce one another in that IMUs help disambiguate occlusions, whereas videos provide disturbance-free spatial information (Gilbert et al., 2019). The idea also applies particularly well to the tracking of multiple individuals—even without the use of appearance features, advantageously—by exploiting unique movement signatures contained within inertial signals to track identities over time (Henschel et al., 2019).

## Pitfalls of Using Deep Learning-Based Motion Capture

Despite being trained on large-scale datasets of thousands of individuals, even the best architectures fail to generalize to "atypical" postures (with respect to the training set). This is wonderfully illustrated by the errors committed by OpenPose on yoga poses (Huang et al., 2019).

These domain shifts are major challenges (also illustrated below), and although this is an active area of research with much progress, the easiest way to make sure that the algorithm generalizes well is to label data that are similar to the videos at inference time. However, due to active learning implemented for many packages, users can manually refine the labels on "outlier" frames.

Another major caveat of deep learning-powered pose estimation is arguably its intrinsic reliance on high-quality labeled images. This suggests that a labeled dataset that reflects the variability of the behavior should be used. If one—due to the quality of the video—cannot reliably identify body parts in still images (i.e., due to massive motion blur, uncertainty about body part [left/right leg crossing], or animal identity), then the video quality should be fixed, or sub-optimal results should be expected.

To give readers a concrete idea about label errors, augmentation methods, and active learning, we also provide some simple experiments with shared code and data. Code for reproducing these analyses is available at https://github.com/DeepLabCut/Primer-MotionCapture.

To illustrate the importance of error-free labeling, we artificially corrupted labels from the trail-tracking dataset from Mathis et al. (2018). The corruptions respectively simulate inattentive labeling (e.g., with left-right body parts being occasionally confounded) and missing annotation or uncertainty as to whether to label an occluded body part. We corrupted 1%, 5%, 10%, and 20% of the training dataset either by swapping two labels or removing one and then trained on 5% of the full data (n = 1,066 images). The effect of missing labels is barely noticeable (Figure 7A). Swapping labels, on the other hand, causes a substantial drop in performance, with an ~10% loss in percentage of correct keypoints (PCK) (Figure 7B). We therefore reason that careful labeling, more so than labeling a very large number of images, is the safest guard against poor ground truth annotations. We believe that explicitly modeling labeling errors, as done in Johnson and Everingham (2011), will be an active area of research and integrated in some packages.

Even if labeled well, augmentation greatly improves results and should be used. For instance, when training on the example dataset of (highly) correlated frames from one short video of one individual, the loss nicely plateaus and shows comparable train/test errors for three different augmentation methods (Figures 8A and 8B). The three models also give good performance and generalize to a test video of a different mouse. However, closer inspection reveals that the "scalecrop" augmentation method, which only performs cropping and scaling during training (Nath et al., 2019), leads to swaps in body parts with this small training set from only one different mouse (Figures 8C and 8D). The other two methods (imgaug and tensorpack), which were configured to perform rotations of the training data, could robustly track the posture of the mouse (Video S1). This discrepancy becomes striking when observing the PCK plots: imgaug and tensorpack

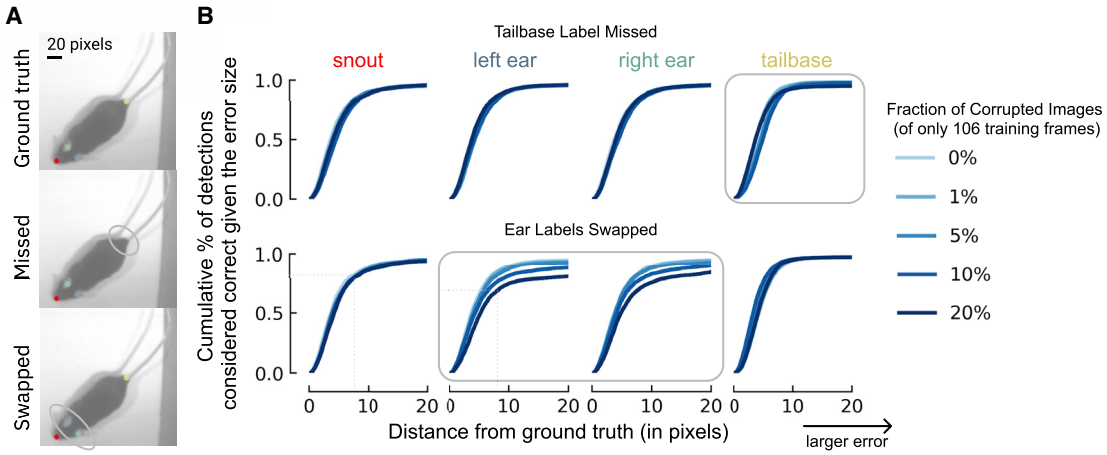



**Figure 7. Labeling Pitfalls: How Corruptions Affect Performance**

(A) Illustration of two types of labeling errors. Top is ground truth, middle is missing a label at the tailbase, and bottom is if the labeler swapped the ear identity (left to right, etc.).

(B) Using a small training dataset of 106 frames, how do the corruptions in (A) affect the percent of correct keypoints (PCK) on the test set as the distance to ground truth increases from 0 pixels (perfect prediction) to 20 pixels (larger error)? The x axis denotes the difference in the ground truth to the predicted location (RMSE in pixels), whereas the y axis is the fraction of frames considered accurate (e.g., ≈80% of frames fall within 9 pixels, even on this small training dataset, for points that are not corrupted, whereas for swapped points this falls to ≈65%). The fraction of the dataset that is corrupted affects this value. Shown is when missing the tailbase label (top) or swapping the ears in 1%, 5%, 10%, and 20% of frames (of 106 labeled training images). Swapping versus missing labels has a more notable adverse effect on network performance.

outperform scalecrop by a margin of up to ≈30% (Figure 8E). One simple way to generalize to this additional case is by active learning (Nath et al., 2019), which is also available for some packages. Thereby, one annotates additional frames with poor performance (outlier frames) and then trains the network from the final configuration, which thus only requires a few thousand iterations. Adding 28 annotated frames from the higher resolution camera, we get good generalization for test frames from both scenarios (Figure 8F). Generally, this illustrates how the lack of diversity in training data leads to worse performance but can be fixed by adding frames with poor performance (active learning).

### Coping with Pitfalls

Fortunately, dealing with the most common pitfalls is relatively straightforward and mostly demands caution and common sense. Rules of thumb and practical guidelines are given in Box 5. Video quality should be envisaged as a trade-off between storage limitations, labeling precision, and training speed; e.g., the lower the resolution of a video, the smaller the occupied disk space and the faster the training speed, but the harder it gets to consistently identify body parts. In practice, DeepLabCut was shown to be very robust to downsizing and video compression, with pose reconstruction degrading only after scaling videos down to a third of their original size or compression by a factor of 1,000 (Mathis and Warren, 2018).

Body parts should be labeled reliably and consistently across frames that preferably capture a variety of behaviors. Note that some packages provide the user means to automatically extract frames differing in visual content based on unsupervised clustering, which simplifies the selection of relevant images in sparse behaviors.

Utilize symmetries for training with augmentation and try to include image augmentations that are helpful. Use the strongest model (given the speed requirements). Check performance and actively grow the training set if errors are found.

Pose estimation algorithms can make different types of errors: jitter, inversion (e.g., left/right), swap (e.g., associating body part to another individual), and miss (Ruggero Ronchi and Perona, 2017). Depending on the type of errors, different causes need to be addressed (i.e., check the data quality for any human-applied mistakes [Mathis et al., 2018] and use suitable augmentation methods). For some cases, post processing filters can be useful (such as Kalman filters), but also graphical models or other methods that learn the geometry of the body parts. We also believe that future work will explicitly model labeling errors during training.

## WHAT TO DO WITH MOTION CAPTURE DATA?

Pose estimation with deep learning is to relieve the user of the painfully slow digitization of keypoints. With markerless tracking, you need to annotate a much smaller dataset, and this can be applied to new videos. Pose estimation also serves as a springboard to a plethora of other techniques. Indeed, many new tools are specifically being developed to aid users of pose estimation packages to analyze movement and behavioral outputs in a high-throughput manner. Plus, many such packages existed pre-deep learning and can now be leveraged with this new technology as well. Although the general topic of what to do with the data is beyond this primer, we will provide a number of pointers. These tools fall into three classes: time series analysis, supervised, and unsupervised learning tools.

A natural step ahead is the quantitative analysis of the keypoint trajectories. The computation of linear and angular displacements, as well as their time derivatives, lays the ground for detailed motor performance evaluation—a great introduction to

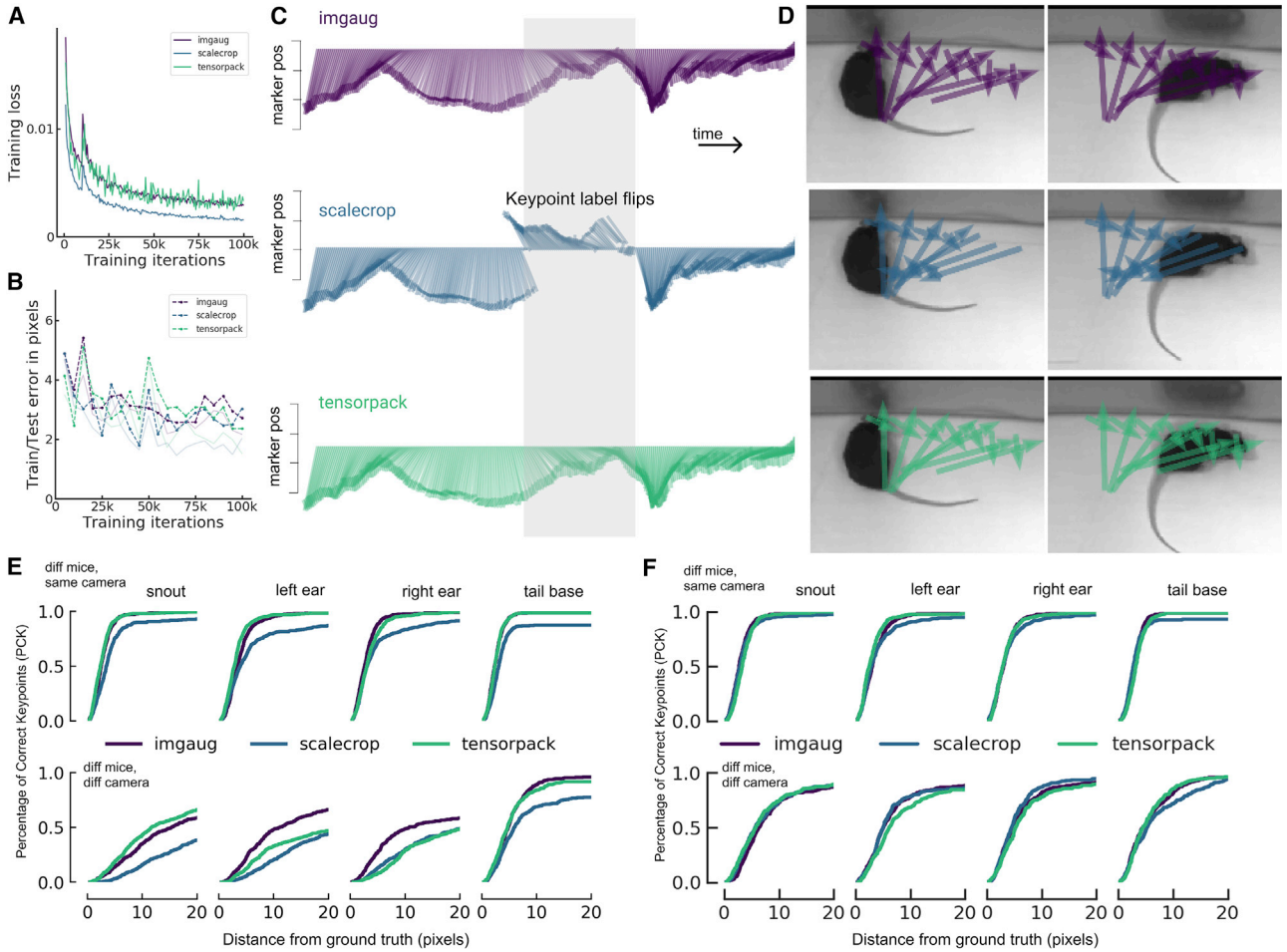



**Figure 8. Data Augmentation Improves Performance**

Performance of three different augmentation methods on the same dataset of around 100 training images from one short video of one mouse (thus correlated). Scalecrop is configured to only change the scale and randomly crop images; Imgaug also performs motion blur and rotation (±180°) augmentation. Tensorpack performs Gaussian noise and rotation (±180°) augmentation.

(A and B) Loss over training iterations has plateaued (A), and test errors in pixels appear comparable for all methods (B).

(C and D) Tail base-aligned skeletons across time for a video of a different mouse with predictions displayed as a "cross" connecting snout to tail and left ear to right ear. (C). Note the swap of the "T" in the shaded gray zone (and overlaid on the image to the right in (D). Imgaug and tensorpack, which also included full 180° rotations, work perfectly (see also Video S1, which shows the three methods in parallel). This example highlights that utilizing the rotational symmetry of the data during training can give excellent performance (without additional labeling).

(E) Performance of the networks on different mice recorded with the same camera (top) and a different camera (≈2.5× magnification; bottom). Networks trained with tensorpack and imgaug augmentation generalize much better and, in particular, generalize very well to different mice. The generalization to the other camera is difficult, but also works better for tensorpack and imgaug augmentation.

(F) Performance of networks on same data as in (E), but after an active learning step, adding 28 training frames from the higher-resolution camera and training for a few thousand iterations. Afterward, the network generalizes well to both scenarios.

elementary kinematics can be found in Winter (2009), and a thorough description of 151 common metrics is given in Schwarz et al. (2019). These have a broad range of applications, of which we highlight a system for assessing >30 behaviors in groups of mice in an automated way (de Chaumont et al., 2019) or an investigation of the evolution of gait invariants across animals (Catavitello et al., 2018). Furthermore, kinematic metrics are the basis from which to deconstruct complex whole-body movements into interpretable motor primitives, non-invasively probing neuromuscular control (Longo et al., 2019). Unsupervised methods, such as clustering methods (Pedregosa et al., 2011), MotionMapper (Berman et al., 2014), MoSeq (Wiltschko et al., 2015), or variational autoencoders (Luxem et al., 2020), allow the extraction of common "kinematic behaviors" such as turning, running, and rearing. Supervised methods allow the prediction of human-defined labels such as "attack" or "freezing." For this, general purpose tools such as scikit-learn (Pedregosa et al., 2011) can be ideal, or tailored solutions with integrated GUIs such as JAABA can be used (Kabra et al., 2013). Sturman et al. (2020) have developed an open source package to utilize motion capture outputs together with classifiers to automate human annotations for various behavioral tests (open field,

**Box 5. Avoiding Pitfalls**

- **Video quality:** while deep learning-based methods are more robust than other methods and can even learn from blurry, low-resolution images, you will make your life easier by recording quality videos.
- **Labeling:** label accurately and use enough data from different videos. 10 videos with 20 frames each is better than 1 video with 200 frames. Check labeling quality. If multiple people label, agree on conventions—i.e., be sure that for a larger body part (e.g., back of mouse) the same location is labeled.
- **Dataset curation:** collect annotation data from the full repertoire of behavior (different individuals, backgrounds, postures). Automatic methods of frame extraction exist, but the videos need to be manually selected.
- **Data augmentation:** are there specific features you know happen in your videos, like motion blur or contrast changes? Can rotational symmetry or mirroring be exploited? Then use an augmentation scheme that can build this into training.
- **Optimization:** train until loss plateaus and do not over-train. Check that it worked by looking at performance on training images (both quantitatively and visually), ideally across "snapshots" (i.e., train iterations of the network). If that works, look at test images. Does the network generalize well? Note that, even if everything is proper, train and test performance can be different due to overfitting on idiosyncrasies of training set. Bear in mind that the latest iterations may not be the ones yielding the smallest errors on the test set. It is therefore recommended to store and evaluate multiple snapshots.
- **Cross-validation:** you can compare different parameters (networks, augmentation, and optimization) to get the best performance (see Figure 7).

elevated plus maze, forced swim test). They showed that these open source methods outperform commercially available platforms (Sturman et al., 2020).

Kinematic analysis, together with simple principles derived from physics, also allows the calculation of the energy required to move about, a methodology relevant to understanding the mechanical determinants of the metabolic cost of locomotion (Saibene and Minetti, 2003) or informing the design of bio-inspired robots (Li et al., 2017; Nyakatura et al., 2019).

### Modeling and Motion Understanding

Looking forward, we also expect that motion capture data will be used to learn task-driven and data-driven models of the sensorimotor as well as the motor pathway. We have recently provided a blueprint combining human movement data, inverse kinematics, biomechanical modeling, and deep learning (Sandbrink et al., 2020). Given the complexity of movement, as well as the highly nonlinear nature of sensorimotor processing (Madhav and Cowan, 2020; Nyakatura et al., 2019), we believe that such approaches will be fruitful to leverage motion capture data to gain insight into brain function.

## PERSPECTIVES

As we highlighted thus far in this primer, markerless motion capture has reached a mature state in only a few years due to the many advances in machine learning and computer vision. Although there are still some challenges left (Mathis and Mathis, 2020), this is an active area of research, and advances in training schemes (such as semi-supervised and self-supervised learning) and model architectures will provide further advances and even less required manual labor. Essentially, now every lab can train appropriate algorithms for their application and turn videos into accurate measurements of posture. If setups are sufficiently standardized, these algorithms already broadly generalize, even across multiple laboratories as in the case of the International Brain Lab (Harris et al., 2019). But how do we get there, and how do we make sure the needs of animal pose estimation for neuroscience applications are met?

### Recent Developments in Deep Learning

Innovations in the field of object recognition and detection affect all aforementioned parts of the algorithm, as we discussed in the context of using pre-trained representations. An emerging relevant research direction in machine learning is large-scale semi-supervised and self-supervised representation learning (SSL). In SSL, the problem of pre-training representations is no longer dependent on large labeled datasets, as introduced above. Instead, even larger databases comprising unlabeled examples—often multiple orders of magnitude larger than the counterparts used in supervised learning—can be leveraged. A variety of SSL algorithms are becoming increasingly popular in all areas of machine learning. Recently, representations obtained by large-scale self-supervised pre-training began to approach or even surpass performance of the best supervised methods. Various SSL methods (Oord et al., 2018; Logeswaran and Lee, 2018; Wu et al., 2018; Hénaff et al., 2019; Tian et al., 2019; Hjelm et al., 2018; Bachman et al., 2019; He et al., 2019; Chen et al., 2020) made strides in image recognition (Chen et al., 2020), speech processing (Schneider et al., 2019; Baevski et al., 2020a, 2020b; Ravanelli et al., 2020), and NLP (Devlin et al., 2019; Liu et al., 2019), already starting to outperform models obtained by supervised pre-training on large datasets. Considering that recent SSL models for computer vision continue to be shared openly (Xie et al., 2020; Chen et al., 2020), these models can be expected to impact and improve pose estimation, especially if merely replacing the backend model is required. On top of that, SSL methods can be leveraged in end-to-end models for estimating keypoints and poses directly from raw, unlabeled video (Umer et al., 2020; Tung et al., 2017; Kocabas et al., 2019). Approaches based on graph neural networks (Scarselli et al., 2009) can encode priors about the observed structure and model correlations between individual keypoints and across time (Cai et al., 2019). For some applications (like modeling soft tissue or

volume), full surface reconstructions are needed, and this area has seen tremendous progress in recent years (Güler et al., 2018; Sanakoyeu et al., 2020; Zuffi et al., 2019). Such advances can be closely watched and incorporated in neuroscience, but we also believe our field (neuroscience) is ready to innovate in this domain too.

## Pose Estimation Specifically for Neuroscience

The goals of human pose estimation—aside from the purely scientific advances for object detection—range from person localization in videos, self-driving cars, and pedestrian safety to socially aware artificial intelligence (AI). These are related to, but do differ from, the applied goals of animal pose estimation in neuroscience. Here, we want tools that give us the highest precision with the most rapid feedback options possible, and we want to train on small datasets but have them generalize well. This is a tall order, but so far we have seen that the glass is (arguably more than) half full. How do we meet these goals going forward? Although much research is still required, there are essentially two ways forward: (1) datasets and associated benchmarks and (2) algorithms.

## Neuroscience Needs (More) Benchmarks

In order to push the field toward innovations in areas the community finds important, setting up benchmark datasets and tasks will be crucial (i.e., the animal version of ImageNet). The community can work toward sharing and collecting data of relevant tasks and curating it into benchmarks. This also has the opportunity of shifting the focus in computer vision research: instead of "only" doing human pose estimation, researchers probably will start evaluating on datasets directly relevant to neuroscience community. Indeed, there has been a recent interest in more animal-related work at top machine learning conferences (Khan et al., 2020; Sanakoyeu et al., 2020), and providing proper benchmarks for such approaches would be ideal.

For animals, such efforts are developing: Khan et al. (2020) recently shared a dataset comprising 22,400 annotated faces from 350 diverse species, and Labuguen et al. (2020) announced a dataset of 13,000 annotated macaque images. We recently released two benchmark datasets that can be evaluated for state-of-the-art performance (https://paperswithcode.com) on within-domain and out-of-domain data (http://horse10.deeplabcut.org/). The motivation is to train on a limited number of individuals and test on held out animals (the so-called "out-of-domain" issue) (Mathis et al., 2019, 2020). We picked horses due to the variation in coat colors (and provide >8,000 labeled frames). To directly study the inherent shift in domain between individuals, we set up a benchmark for common image corruptions, as introduced by Hendrycks et al. (2019), that uses the image corruptions library proposed by Michaelis et al. (2019).

Of course, these aforementioned benchmarks are not sufficient to cover all the needs of the community, so we encourage consortium-style efforts to curate data and provide additional benchmarks. Plus, making robust networks is still a major challenge, even when trained with large amounts of data (Beery et al., 2018; Geirhos et al., 2020). In order to make this a possibility, it will be important to develop and share common keypoint estimation benchmarks for animals as well as expand the human ones to applications of interest, such as sports (Huang et al., 2019).

## Sharing Pre-trained Models

We believe another major step forward will be sharing pre-trained pose estimation networks. If, as a field, we were to annotate sufficiently diverse data, we could train more robust networks that broadly generalize. This success is promised by other large-scale datasets such as MS COCO (Lin et al., 2014) and MPII pose (Andriluka et al., 2014). In the computer vision community, sharing model weights such that models do not need to be retrained has been critical for progress. For example, the ability to download pre-trained ImageNet weights is invaluable—training ImageNet from scratch on a standard GPU can take more than a week. Now, they are downloaded within a few seconds and fine-tuned in packages like DeepLabCut. However, even for custom training setups, sharing of code and easy access to cloud computing resources enables smaller labs to train and deploy models without investment in additional lab resources. Pre-training a typical object recognition model on the ILSVC is now possible on the order of minutes for less than 100 USD (Coleman et al., 2017) thanks to high-end cloud computing, which is also feasible for labs lacking the necessary on-site infrastructure (Box 3).

In neuroscience, we should aim to fine tune even those models; sharing of mouse-specific, primate-specific weights will drive interest and momentum from researchers without access to such data and further drive innovations. Currently, only DeepLabCut provides model weights (albeit not at the time of the original publication) as part of the recently launched Model Zoo (http://modelzoo.deeplabcut.org/). Currently it contains models trained on MPII pose (Insafutdinov et al., 2016), dog and cat models, as well as contributed models for primate facial recognition, primate full body recognition (Labuguen et al., 2020), and mouse pupil detection (Figure 6). Researchers can also contribute in a citizen-science fashion by labeling data on the web (https://contrib.deeplabcut.org) or by submitting models.

Both datasets and models will benefit from common formatting to ease sharing and testing. Candidate formats are HDF5 (also chosen by NeuroData Without Borders [Teeters et al., 2015] and DeepLabCut), TensorFlow data (https://www.tensorflow.org/api_docs/python/tf/data), and/or PyTorch data (https://pytorch.org/docs/stable/torchvision/datasets.html). Specifically, for models, proto-buffer formats for weights are useful and easy to share (Kane et al., 2020; Lopes et al., 2015) for deployment to other systems. Platforms such as OSF and Zenodo allow banking of weights, and some papers (Barrett et al., 2020; Sturman et al., 2020) have also shared their trained models. We envision that having easy-to-use interfaces to such models will be possible in the future.

These pre-trained pose estimation networks hold several promises: it saves time and energy (as different labs do not need to annotate and train networks) as well as contributes to reproducibility in science. Like many other forms of biological data, such as genome sequences and functional imaging data, behavioral data are notoriously hard to analyze in standardized ways. Lack of agreement can lead to different results, as pointed

out by a recent landmark study comparing the results achieved by 70 independent researchers analyzing nine hypotheses in shared imaging data (Botvinik-Nezer et al., 2020). To increase reproducibility in behavioral science, video is a great tool (Gilmore and Adolph, 2017). Analyzing behavioral data is complex, owing to its unstructured, large-scale nature, which highlights the importance of shared analysis pipelines. Thus, building robust architectures that extract the same behavioral measurements in different laboratories would be a major step forward.

## CONCLUSIONS

Deep learning-based markerless pose estimation has been broadly and rapidly adopted in the past 2 years. This impact was, in part, fueled by open-source code: by developing and sharing packages in public repositories on GitHub, they could be easily accessed for free and at scale. These packages are built on advances (and code) in computer vision and AI, which has a strong open science culture. Neuroscience also has strong and growing open science culture (White et al., 2019), which greatly impacts the field, as evidenced by tools from the Allen Institute, the UCLA Miniscope (Aharoni et al., 2019), OpenEphys (Siegle et al., 2017), and Bonsai (Lopes et al., 2015) (just to name a few).

Moreover, neuroscience and AI have a long history of influencing each other (Hassabis et al., 2017), and research in neuroscience will likely contribute to making AI more robust (Sinz et al., 2019; Hassabis et al., 2017). The analysis of animal motion is a highly interdisciplinary field at the intersection of biomechanics, computer vision, medicine, and robotics with a long tradition (Klette and Tee, 2008). The recent advances in deep learning have greatly simplified the measurement of animal behavior, which, as we and others believe (Krakauer et al., 2017), in turn will greatly advance our understanding of the brain.

## SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.neuron.2020.09.017.

## ACKNOWLEDGMENTS

We thank Yash Sharma for discussions around future directions in self-supervised learning and Erin Diel, Maxime Vidal, Claudio Michaelis, and Thomas Biasi for comments on the manuscript. We thank Julia Kuhl and scidraw.io for illustrations. Funding was provided by the Rowland Institute at Harvard University (M.W.M. and A.M.), the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation (#2019-207363; M.W.M., A.M., and J.L.), and the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (St.S.; FKZ: 01IS18039A). St.S. thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) and acknowledges his membership in the European Laboratory for Learning & Intelligent Systems (ELLIS) PhD program. The authors declare no conflicts of interest. M.W.M. dedicates this work to Adam E. Max.

## REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation, pp. 265–283.

Aharoni, D., Khakh, B.S., Silva, A.J., and Golshani, P. (2019). All the light that we can see: a new era in miniaturized microscopy. Nat. Methods *16*, 11–13.

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2D human pose estimation: New benchmark and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3686–3693.

Arac, A., Zhao, P., Dobkin, B.H., Carmichael, S.T., and Golshani, P. (2019). Deepbehavior: A deep learning toolbox for automated analysis of animal and human behavior imaging data. Front. Syst. Neurosci. *13*, 20.

Bachman, P., Hjelm, R.D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. arXiv, arXiv:1906.00910 https://arxiv.org/abs/1906.00910.

Baevski, A., Schneider, S., and Auli, M. (2020a). vq-wav2vec: Self-supervised learning of discrete speech representations. arXiv, arXiv:1910.05453 https://arxiv.org/abs/1910.0545.

Baevski, A., Zhou, H., Mohamed, A., and Auli, M. (2020b). wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv, arXiv:2006.11477 https://arxiv.org/abs/2006.11477.

Bala, P.C., Eisenreich, B.R., Yoo, S.B.M., Hayden, B.Y., Park, H.S., and Zimmermann, J. (2020). Openmonkeystudio: Automated markerless pose estimation in freely moving macaques. bioRxiv. https://doi.org/10.1101/2020.01.31.928861.

Barrett, J.M., Raineri Tapies, M.G., and Shepherd, G.M.G. (2020). Manual dexterity of mice during food-handling involves the thumb and a set of fast basic movements. PLoS ONE *15*, e0226774.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). Comput. Vis. Image Underst. *110*, 346–359.

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. arXiv, arXiv:1807.04975 https://arxiv.org/abs/1807.04975.

Begon, M., Andersen, M.S., and Dumas, R. (2018). Multibody kinematics optimization for the estimation of upper and lower limb human joint kinematics: a systematized methodological review. J. Biomech. Eng. *140*, https://doi.org/10.1115/1.4038741.

Berger, M., Agha, N.S., and Gail, A. (2020). Wireless recording from unrestrained monkeys reveals motor goal encoding beyond immediate reach in frontoparietal cortex. eLife *9*, e51322.

Berman, G.J., Choi, D.M., Bialek, W., and Shaevitz, J.W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. J. R. Soc. Interface *11*, 20140672.

Bidder, O.R., Walker, J.S., Jones, M.W., Holton, M.D., Urge, P., Scantlebury, D.M., Marks, N.J., Magowan, E.A., Maguire, I.E., and Wilson, R.P. (2015). Step by step: reconstruction of terrestrial animal movement paths by dead-reckoning. Mov. Ecol. *3*, 23.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010, Y. Lechevallier and G. Saporta, eds. (Physica-Verlag HD). https://doi.org/10.1007/978-3-7908-2604-3_16.

Botvinik-Nezer, R., Holzmeister, F., Camerer, C.F., Dreber, A., Huber, J., Johannesson, M., Kirchler, M., Iwanir, R., Mumford, J.A., Adcock, R.A., et al. (2020). Variability in the analysis of a single neuroimaging dataset by many teams. Nature *582*, 84–88.

Brown, D.D., Kays, R., Wikelski, M., Wilson, R., and Klimley, A.P. (2013). Observing the unwatchable through acceleration logging of animal behavior. Anim. Biotelemetry. *1*, 20.

Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.-J., Yuan, J., and Thalmann, N.M. (2019). Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2272–2281.

Camomilla, V., Dumas, R., and Cappozzo, A. (2017). Human movement analysis: The soft tissue artefact issue. J. Biomech. *62*, 1–4.

Camomilla, V., Bergamini, E., Fantozzi, S., and Vannozzi, G. (2018). Trends supporting the in-field use of wearable inertial sensors for sport performance evaluation: A systematic review. Sensors (Basel) *18*, 873.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2018). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv, arXiv:1812.08008.

Carreira, J., Agrawal, P., Fragkiadaki, K., and Malik, J. (2016). Human pose estimation with iterative error feedback. arXiv, arXiv:1507.06550 https://arxiv.org/abs/1507.06550.

Catavitello, G., Ivanenko, Y., and Lacquaniti, F. (2018). A kinematic synergy for terrestrial locomotion shared by mammals and birds. eLife *7*, e38190.

Ceseracciu, E., Sawacha, Z., and Cobelli, C. (2014). Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept. PLoS ONE *9*, e87640.

Chakravarty, P., Cozzi, G., Ozgul, A., and Aminian, K. (2019). A novel biomechanical approach for animal behaviour recognition using accelerometers. Methods Ecol. Evol. *10*, 802–814.

Chen, C.-H., and Ramanan, D. (2017). 3D human pose estimation= 2D pose estimation+ matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7035–7043.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A.L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. *40*, 834–848.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. arXiv, arXiv:2002.05709.

Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., and Zhang, L. (2020). Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. arXiv, arXiv:1908.10357 https://arxiv.org/abs/1908.10357.

Coleman, C., Narayanan, D., Kang, D., Zhao, T., Zhang, J., Nardi, L., Bailis, P., Olukotun, K., Ré, C., and Zaharia, M. (2017). Dawnbench: An end-to-end deep learning benchmark and competition. https://dawn.cs.stanford.edu/benchmark/.

Dalal, N., and Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp. 886–893.

Datta, S.R., Anderson, D.J., Branson, K., Perona, P., and Leifer, A. (2019). Computational neuroethology: a call to action. Neuron *104*, 11–24.

de Chaumont, F., Ey, E., Torquet, N., Lagache, T., Dallongeville, S., Imbert, A., Legou, T., Le Sourd, A.-M., Faure, P., Bourgeron, T., and Olivo-Marin, J.C. (2019). Real-time analysis of the behaviour of groups of mice via a depth-sensing camera and machine learning. Nat. Biomed. Eng. *3*, 930–942.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv, arXiv:1810.04805 https://arxiv.org/abs/1810.04805.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. arXiv, arXiv:1310.1531 https://arxiv.org/abs/1310.1531.

Dumoulin, V., and Visin, F. (2016). A guide to convolution arithmetic for deep learning. arXiv, arXiv:1603.07285.

Ebina, T., Obara, K., Watakabe, A., Masamizu, Y., Terada, S.-I., Matoba, R., Takaji, M., Hatanaka, N., Nambu, A., Mizukami, H., et al. (2019). Arm movements induced by noninvasive optogenetic stimulation of the motor cortex in the common marmoset. Proc. Natl. Acad. Sci. USA *116*, 22844–22850.

Fan, B., Li, Q., and Liu, T. (2017). How magnetic disturbance influences the attitude and heading in magnetic and inertial sensor-based orientation estimation. Sensors (Basel) *18*, 76.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F.A. (2020). Shortcut learning in deep neural networks. arXiv, arXiv:2004.07780.

Gilbert, A., Trumble, M., Malleson, C., Hilton, A., and Collomosse, J. (2019). Fusing visual and inertial sensors with semantics for 3d human pose estimation. Int. J. Comput. Vis. *127*, 381–397.

Gilmore, R.O., and Adolph, K.E. (2017). Video can make behavioural science more reproducible. Nat. Hum. Behav. *1*, 0128.

Gleiss, A.C., Wilson, R.P., and Shepard, E.L. (2011). Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. Methods Ecol. Evol. *2*, 23–33.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning (MIT Press).

Graving, J.M., Chae, D., Naik, H., Li, L., Koger, B., Costelloe, B.R., and Couzin, I.D. (2019). DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. eLife *8*, e47994.

Güler, R.A., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. arXiv, arXiv:1802.00434 https://arxiv.org/abs/1802.00434.

Günel, S., Rhodin, H., Morales, D., Campagnolo, J., Ramdya, P., and Fua, P. (2019). DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult *Drosophila*. eLife *8*, e48571.

Harris, K.D., Hunter, M., Rossant, C., Sasaki, M., Shen, S., Steinmetz, N.A., Walker, E.Y., Winter, O., and Wells, M. (2019). Data architecture for a large-scale neuroscience collaboration. bioRxiv. https://doi.org/10.1101/827873.

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. Neuron *95*, 245–258.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. arXiv, arXiv:1512.03385 https://arxiv.org/abs/1512.03385.

He, K., Girshick, R., and Dollár, P. (2018). Rethinking imagenet pre-training. arXiv, arXiv:1811.08883.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2019). Momentum contrast for unsupervised visual representation learning. arXiv, arXiv:1911.05722.

Hénaff, O.J., Razavi, A., Doersch, C., Eslami, S., and Oord, A.d. (2019). Data-efficient image recognition with contrastive predictive coding. arXiv, arXiv:1905.09272.

Hendrycks, D., Lee, K., and Mazeika, M. (2019). Using pre-training can improve model robustness and uncertainty. arXiv, arXiv:1901.09960 https://arxiv.org/abs/1901.09960.

Henschel, R., von Marcard, T., and Rosenhahn, B. (2019). Simultaneous Identification and Tracking of Multiple People Using Video and IMUs. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 780–789, https://doi.org/10.1109/CVPRW.2019.00106.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. arXiv, arXiv:1503.02531.

Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. arXiv, arXiv:1808.06670.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K.Q. (2017). Densely connected convolutional networks. arXiv, arXiv:1608.06993 https://arxiv.org/abs/1608.06993.

Huang, Y., Sun, B., Kan, H., Zhuang, J., and Qin, Z. (2019). FollowMeUp Sports: New Benchmark for 2D Human Keypoint Recognition. In Pattern Recognition and Computer Vision. PRCV 2019. Lecture Notes in Computer Science, Z. Lin, ed. (Springer), pp. 110–121.

Inayat, S., Singh, S., Ghasroddashti, A., Qandeel, Q., Egodage, P., Whishaw, I.Q., and Mohajerani, M.H. (2020). A Matlab-based toolbox for characterizing behavior of rodents engaged in string-pulling. eLife *9*, e54540.

Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., and Schiele, B. (2016). DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. arXiv, arXiv:1605.03170 https://arxiv.org/abs/1605.03170.

Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., and Schiele, B. (2017). Arttrack: Articulated multi-person tracking in the wild. arXiv, arXiv:1612.01465 https://arxiv.org/abs/1612.01465.

Jain, A., Tompson, J., LeCun, Y., and Bregler, C. (2014). Modeep: A deep learning framework using motion features for human pose estimation. arXiv, arXiv:1409.7963 https://arxiv.org/abs/1409.7963.

Johansson, G. (1973). Visual perception of biological motion and a model for its analysis. Percept. Psychophys. *14*, 201–211.

Johnson, S., and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation. In CVPR 2011, pp. 1465–1472, https://doi.org/10.1109/CVPR.2011.5995318.

Jung, A.B., Wada, K., Crall, J., Tanaka, S., Graving, J., Reinders, C., Yadav, S., Banerjee, J., Vecsei, G., Kraft, A., et al. (2020). imgaug. https://github.com/aleju/imgaug.

Kabra, M., Robie, A.A., Rivera-Alba, M., Branson, S., and Branson, K. (2013). JAABA: interactive machine learning for automatic annotation of animal behavior. Nat. Methods *10*, 64–67.

Kane, G., Lopes, G., Saunders, J.L., Mathis, A., and Mathis, M.W. (2020). Real-time deeplabcut for closed-loop feedback based on posture. bioRxiv. https://doi.org/10.1101/2020.08.04.236422.

Kaplan, H.S., and Zimmer, M. (2020). Brain-wide representations of ongoing behavior: a universal principle? Curr. Opin. Neurobiol. *64*, 60–69.

Karashchuk, P., Rupp, K.L., Dickinson, E.S., Sanders, E., Azim, E., Brunton, B.W., and Tuthill, J.C. (2020). Anipose: a toolkit for robust markerless 3d pose estimation. bioRxiv. https://doi.org/10.1101/2020.05.26.117325.

Kays, R., Crofoot, M.C., Jetz, W., and Wikelski, M. (2015). ECOLOGY. Terrestrial animal tracking as an eye on life and planet. Science *348*, aaa2478.

Khan, M.H., McDonagh, J., Khan, S., Shahabuddin, M., Arora, A., Khan, F.S., Shao, L., and Tzimiropoulos, G. (2020). Animalweb: A large-scale hierarchical dataset of annotated animal faces. arXiv, arXiv:1909.04951 https://arxiv.org/abs/1909.04951.

Kingma, D.P., and Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, eds..

Klette, R., and Tee, G. (2008). Understanding Human Motion: A Historic Review. In Human Motion. Computational Imaging and Vision, *Vol. 36*, B. Rosenhahn, R. Klette, and D. Metaxas, eds. (Springer, Dordrecht), pp. 1–22.

Kocabas, M., Karagoz, S., and Akbas, E. (2019). Self-supervised learning of 3d human pose using multi-view geometry. arXiv, arXiv:1903.02330 https://arxiv.org/abs/1903.02330.

Kornblith, S., Shlens, J., and Le, Q.V. (2019). Do better imagenet models transfer better? arXiv, arXiv:1805.08974 https://arxiv.org/abs/1805.08974.

Krakauer, J.W., Ghazanfar, A.A., Gomez-Marin, A., MacIver, M.A., and Poeppel, D. (2017). Neuroscience needs behavior: correcting a reductionist bias. Neuron *93*, 480–490.

Kreiss, S., Bertoni, L., and Alahi, A. (2019). Pifpaf: Composite fields for human pose estimation. arXiv, arXiv:1903.06593 https://arxiv.org/abs/1903.06593.

Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems *25*, https://doi.org/10.1145/3065386.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv, arXiv:1811.00982.

Labuguen, R., Matsumoto, J., Negrete, S., Nishimaru, H., Nishijo, H., Takada, M., Go, Y., Inoue, K.-i., and Shibata, T. (2020). Macaquepose: A novel 'in the wild'macaque monkey pose dataset for markerless motion capture. bioRxiv. https://doi.org/10.1101/2020.07.30.229989.

Laurence-Chasen, J.D., Manafzadeh, A.R., Hatsopoulos, N.G., Ross, C.F., and Arce-McShane, F.I. (2020). Integrating xmalab and deeplabcut for high-throughput xromm. J. Exp. Biol. *223*, jeb226720.

Leakey, M.D., and Hay, R.L. (1979). Pliocene footprints in the laetolil beds at Laetoli, Northern Tanzania. Nature *278*, 317–323.

Lebleu, J., Gosseye, T., Detrembleur, C., Mahaudens, P., Cartiaux, O., and Penta, M. (2020). Lower limb kinematics using inertial sensors during locomotion: Accuracy and reproducibility of joint angle calculations with different sensor-to-segment calibrations. Sensors (Basel) *20*, 715.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444.

Li, C., Kessens, C.C., Fearing, R.S., and Full, R.J. (2017). Mechanical principles of dynamic terrestrial self-righting using wings. Adv. Robot. *31*, 881–900.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. arXiv, arXiv:1712.09913 https://arxiv.org/abs/1712.09913.

Li, H., Singh, B., Najibi, M., Wu, Z., and Davis, L.S. (2019). An analysis of pre-training on object detection. arXiv, arXiv:1904.05871.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C.L. (2014). Microsoft coco: Common objects in context. arXiv, arXiv:1405.0312 https://arxiv.org/abs/1405.0312.

List, R., Postolka, B., Schütz, P., Hitz, M., Schwilch, P., Gerber, H., Ferguson, S.J., and Taylor, W.R. (2017). A moving fluoroscope to capture tibiofemoral kinematics during complete cycles of free level and downhill walking as well as stair descent. PLoS ONE *12*, e0185952.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv, arXiv:1907.11692 https://arxiv.org/abs/1907.11692.

Liu, X., Yu, S.-y., Flierman, N., Loyola, S., Kamermans, M., Hoogland, T.M., and De Zeeuw, C.I. (2020). Optiflex: video-based animal pose estimation using deep learning enhanced by optical flow. bioRxiv. https://doi.org/10.1101/2020.04.04.025494.

Logeswaran, L., and Lee, H. (2018). An efficient framework for learning sentence representations. arXiv, arXiv:1803.02893 https://arxiv.org/abs/1803.02893.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. arXiv, arXiv:1411.4038 https://arxiv.org/abs/1411.4038.

Longo, A., Haid, T., Meulenbroek, R., and Federolf, P. (2019). Biomechanics in posture space: Properties and relevance of principal accelerations for characterizing movement control. J. Biomech. *82*, 397–403.

Lopes, G., Bonacchi, N., Frazão, J., Neto, J.P., Atallah, B.V., Soares, S., Moreira, L., Matias, S., Itskov, P.M., Correia, P.A., et al. (2015). Bonsai: an event-based framework for processing and controlling data streams. Front. Neuroinform. *9*, 7.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. *60*, 91–110.

Lu, T.-W., and O'Connor, J.J. (1999). Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints. J. Biomech. *32*, 129–134.

Luxem, K., Fuhrmann, F., Kürsch, J., Remy, S., and Bauer, P. (2020). Identifying behavioral structure from deep variational embeddings of animal motion. bioRxiv. https://doi.org/10.1101/2020.05.14.095430.

Maceira-Elvira, P., Popa, T., Schmid, A.-C., and Hummel, F.C. (2019). Wearable technology in stroke rehabilitation: towards improved diagnosis and treatment of upper-limb motor impairment. J. Neuroeng. Rehabil. *16*, 142.

Madhav, M.S., and Cowan, N.J. (2020). The synergy between neuroscience and control theory: the nervous system as inspiration for hard control challenges. Annu. Rev. Control. Robot. Auton. Syst. *3*, 243–267.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. (2018). Exploring the limits of weakly supervised pretraining. arXiv, arXiv:1805.00932 https://arxiv.org/abs/1805.00932.

Martinez, J., Hossain, R., Romero, J., and Little, J.J. (2017). A simple yet effective baseline for 3d human pose estimation. arXiv, arXiv:1705.03098 https://arxiv.org/abs/1705.03098.

Mathis, M.W., and Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. Curr. Opin. Neurobiol. *60*, 1–11.

Mathis, A., and Warren, R.A. (2018). On the inference speed and video-compression robustness of deeplabcut. bioRxiv. https://doi.org/10.1101/457242.

Mathis, A., Biasi, T., Yüksekgönül, M., Rogers, B., Bethge, M., and Mathis, M.W. (2020). Imagenet performance correlates with pose estimation robustness and generalization on out-of-domain data. In International Conference on Machine Learning 2020 Workshop on Uncertainty and Robustness in Deep Learning (ICML).

Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W., and Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nat. Neurosci. *21*, 1281–1289.

Mathis, A., Yüksekgönül, M., Rogers, B., Bethge, M., and Mathis, M.W. (2019). Pretraining boosts out-of-domain robustness for pose estimation. arXiv, arXiv:1909.11229.

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B.A., Karhohs, K.W., Doan, M., Ding, L., Rafelski, S.M., Thirstrup, D., et al. (2018). CellProfiler 3.0: Next-generation image processing for biology. PLoS Biol. *16*, e2005970.

Mehta, D., Rhodin, H., Casas, D., Sotnychenko, O., Xu, W., and Theobalt, C. (2016). Monocular 3d human pose estimation using transfer learning and improved CNN supervision. arXiv, arXiv:1611.09813 https://arxiv.org/abs/1611.09813.

Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. Linux J. *2014*, 2.

Meyer, A.F., O'Keefe, J., and Poort, J. (2020). Two distinct types of eye-head coupling in freely moving mice. Curr. Biol. *30*, 2116–2130.

Micera, S., Caleo, M., Chisari, C., Hummel, F.C., and Pedrocchi, A. (2020). Advanced neurotechnologies for the restoration of motor function. Neuron *105*, 604–620.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., and Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv, arXiv:1907.07484.

Moeslund, T.B., Hilton, A., and Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. Comput. Vis. Image Underst. *104*, 90–126.

Nath, T., Mathis, A., Chen, A.C., Patel, A., Bethge, M., and Mathis, M.W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nat. Protoc. *14*, 2152–2176.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. arXiv, arXiv:1603.06937 https://arxiv.org/abs/1603.06937.

Nyakatura, J.A., Melo, K., Horvat, T., Karakasiliotis, K., Allen, V.R., Andikfar, A., Andrada, E., Arnold, P., Lauströer, J., Hutchinson, J.R., et al. (2019). Reverse-engineering the locomotion of a stem amniote. Nature *565*, 351–355.

O'Connell, A.F., Nichols, J.D., and Karanth, K.U. (2010). Camera Traps in Animal Ecology: Methods and Analyses (Springer Science & Business Media).

Oord, A.d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv, arXiv:1807.03748.

Pasquet, M.O., Tihy, M., Gourgeon, A., Pompili, M.N., Godsil, B.P., Léna, C., and Dugué, G.P. (2016). Wireless inertial measurement of head kinematics in freely-moving rats. Sci. Rep. *6*, 35689.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv, arXiv:1912.01703 https://arxiv.org/abs/1912.01703.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. J. Mach. Learn. Res. *12*, 2825–2830.

Pereira, T.D., Aldarondo, D.E., Willmore, L., Kislin, M., Wang, S.S.-H., Murthy, M., and Shaevitz, J.W. (2019). Fast animal pose estimation using deep neural networks. Nat. Methods *16*, 117–125.

Peterson, S.M., Singh, S.H., Wang, N.X., Rao, R.P., and Brunton, B.W. (2020). Behavioral and neural variability of naturalistic arm movements. bioRxiv. https://doi.org/10.1101/2020.04.17.047357.

Poppe, R. (2007). Vision-based human motion analysis: An overview. Comput. Vis. Image Underst. *108*, 4–18.

Ramsey, D.K., Wretenberg, P.F., Benoit, D.L., Lamontagne, M., and Németh, G. (2003). Methodological concerns using intra-cortical pins to measure tibiofemoral kinematics. Knee Surg. Sports Traumatol. Arthrosc. *11*, 344–349.

Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., and Bengio, Y. (2020). Multi-task self-supervised learning for robust speech recognition. arXiv, arXiv:2001.09239 https://arxiv.org/abs/2001.09239.

Roetenberg, D., Luinge, H., and Slycke, P. (2009). Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Xsens Motion Technol. BV Tech. Rep. *3*.

Rueden, C.T., Ackerman, J., Arena, E.T., Eglinger, J., Cimini, B.A., Goodman, A., Carpenter, A.E., and Eliceiri, K.W. (2019). Scientific Community Image Forum: A discussion forum for scientific image software. PLoS Biol. *17*, e3000340.

Ruggero Ronchi, M., and Perona, P. (2017). Benchmarking and error diagnosis in multi-instance pose estimation. arXiv, arXiv:1707.05388 https://arxiv.org/abs/1707.05388.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. *115*, 211–252.

Sabatini, A.M. (2011). Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing. Sensors (Basel) *11*, 1489–1525.

Saibene, F., and Minetti, A.E. (2003). Biomechanical and physiological aspects of legged locomotion in humans. Eur. J. Appl. Physiol. *88*, 297–316.

Sanakoyeu, A., Khalidov, V., McCarthy, M.S., Vedaldi, A., and Neverova, N. (2020). Transferring dense pose to proximal animal classes. arXiv, arXiv:2003.00080.

Sandbrink, K.J., Mamidanna, P., Michaelis, C., Mathis, M.W., Bethge, M., and Mathis, A. (2020). Task-driven hierarchical deep neural networkmodels of the proprioceptive pathway. bioRxiv. https://doi.org/10.1101/2020.05.06.081372.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. arXiv, arXiv:1801.04381 https://arxiv.org/abs/1801.04381.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. IEEE Trans. Neural Netw. *20*, 61–80.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. Nat. Methods *9*, 676–682.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. arXiv, arXiv:1904.05862 https://arxiv.org/abs/1904.05862.

Schwarz, A., Kanzler, C.M., Lambercy, O., Luft, A.R., and Veerbeek, J.M. (2019). Systematic review on kinematic assessments of upper limb movements after stroke. Stroke *50*, 718–727.

Seethapathi, N., Wang, S., Saluja, R., Blohm, G., and Kording, K.P. (2019). Movement science needs different pose tracking algorithms. arXiv, arXiv:1907.10226.

Siegle, J.H., López, A.C., Patel, Y.A., Abramov, K., Ohayon, S., and Voigts, J. (2017). Open Ephys: an open-source, plugin-based platform for multichannel electrophysiology. J. Neural Eng. *14*, 045003.

Siegle, J.H., Jia, X., Durand, S., Gale, S., Bennett, C., Graddis, N., Heller, G., Ramirez, T.K., Choi, H., Luviano, J.A., et al. (2019). A survey of spiking activity reveals a functional hierarchy of mouse corticothalamic visual areas. bioRxiv. https://doi.org/10.1101/805010.

Sinz, F.H., Pitkow, X., Reimer, J., Bethge, M., and Tolias, A.S. (2019). Engineering a less artificial intelligence. Neuron *103*, 967–979.

Smale, K.B., Potvin, B.M., Shourijeh, M.S., and Benoit, D.L. (2017). Knee joint kinematics and kinetics during the hop and cut after soft tissue artifact suppression: Time to reconsider ACL injury mechanisms? J. Biomech. *62*, 132–139.

Sommer, C., Straehle, C., Koethe, U., and Hamprecht, F.A. (2011). Ilastik: Interactive learning and segmentation toolkit. In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 230–233.

Spoor, C.W., and Veldpaus, F.E. (1980). Rigid body motion calculated from spatial co-ordinates of markers. J. Biomech. *13*, 391–393.

Sturman, O., von Ziegler, L., Schläppi, C., Akyol, F., Privitera, M., Slominski, D., Grimm, C., Thieren, L., Zerbi, V., Grewe, B., and Bohacek, J. (2020). Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. Neuropsychopharmacology *45*, 1942–1952.

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). Deep high-resolution representation learning for human pose estimation. arXiv, arXiv:1908.07919 https://arxiv.org/abs/1908.07919.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. Proceedings of the 30th International Conference on Machine Learning*28*, pp. 1139–1147.

Tan, M., and Le, Q.V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv, arXiv:1905.11946.

Teeters, J.L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., et al. (2015). Neurodata without borders: creating a common data format for neurophysiology. Neuron *88*, 629–634.

Tian, Y., Krishnan, D., and Isola, P. (2019). Contrastive multiview coding. arXiv, arXiv:1906.05849.

Tomè, D., Russell, C., and Agapito, L. (2017). Lifting from the deep: Convolutional 3d pose estimation from a single image. arXiv, arXiv:1701.00295 https://arxiv.org/abs/1701.00295.

Tompson, J.J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. arXiv, arXiv:1406.2984 https://arxiv.org/abs/1406.2984.

Toshev, A., and Szegedy, C. (2013). Deeppose: Human pose estimation via deep neural networks. arXiv, arXiv:1312.4659 https://arxiv.org/abs/1312.4659.

Tracey, I., Woolf, C.J., and Andrews, N.A. (2019). Composite pain biomarker signatures for objective assessment and effective treatment. Neuron *101*, 783–800.

Tung, H.-Y., Tung, H.-W., Yumer, E., and Fragkiadaki, K. (2017). Self-supervised learning of motion capture. arXiv, arXiv:1712.01337 https://arxiv.org/abs/1712.01337.

Umer, R., Doering, A., Leibe, B., and Gall, J. (2020). Self-supervised keypoint correspondences for multi-person pose estimation and tracking in videos. arXiv, arXiv:2004.12652.

van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., and Yu, T.; Scikit-Image Contributors (2014). scikit-image: image processing in Python. PeerJ *2*, e453.

Vargas-Valencia, L.S., Elias, A., Rocon, E., Bastos-Filho, T., and Frizera, A. (2016). An imu-to-body alignment method applied to human gait analysis. Sensors (Basel) *16*, 2090.

von Marcard, T., Rosenhahn, B., Black, M.J., and Pons-Moll, G. (2017). Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. Comput. Graph. Forum *36*, 349–360.

Weinstein, B.G. (2018). A computer vision for animal ecology. J. Anim. Ecol. *87*, 533–545.

White, S.R., Amarante, L.M., Kravitz, A.V., and Laubach, M. (2019). The future is open: Open-source tools for behavioral neuroscience research. eNeuro *6*, ENEURO.0223-19.2019.

Wilson, A.M., Hubel, T.Y., Wilshin, S.D., Lowe, J.C., Lorenc, M., Dewhirst, O.P., Bartlam-Brooks, H.L.A., Diack, R., Bennitt, E., Golabek, K.A., et al. (2018). Biomechanics of predator-prey arms race in lion, zebra, cheetah and impala. Nature *554*, 183–188.

Wiltschko, A.B., Johnson, M.J., Iurilli, G., Peterson, R.E., Katon, J.M., Pashkovski, S.L., Abraira, V.E., Adams, R.P., and Datta, S.R. (2015). Mapping sub-second structure in mouse behavior. Neuron *88*, 1121–1135.

Winter, D. (2009). Biomechanics and Motor Control of Human Movement (John Wiley & Sons).

Wu, Y. (2016). Tensorpack. https://github.com/tensorpack/.

Wu, Z., Xiong, Y., Yu, S.X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. arXiv, arXiv:1805.01978 https://arxiv.org/abs/1805.01978.

Wu, X., Sahoo, D., and Hoi, S.C. (2020). Recent advances in deep learning for object detection. arXiv, arXiv:1908.03673 https://arxiv.org/abs/1908.03673.

Xiao, B., Wu, H., and Wei, Y. (2018). Simple baselines for human pose estimation and tracking. arXiv, arXiv:1804.06208 https://arxiv.org/abs/1804.06208.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q.V. (2020). Self-training with noisy student improves imagenet classification. arXiv, arXiv:1911.04252 https://arxiv.org/abs/1911.04252.

Xu, L., Xu, W., Golyanik, V., Habermann, M., Fang, L., and Theobalt, C. (2020). Eventcap: Monocular 3d capture of high-speed human motions using an event camera. arXiv, arXiv:1908.11505 https://arxiv.org/abs/1908.11505.

Yao, Y., Jafarian, Y., and Park, H.S. (2019). Monet: Multiview semi-supervised keypoint detection via epipolar divergence. arXiv, arXiv:1806.00104 https://arxiv.org/abs/1806.00104.

Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., and Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. arXiv, arXiv:1804.08328 https://arxiv.org/abs/1804.08328.

Zhao, M., Li, T., Abu Alsheikh, M., Tian, Y., Zhao, H., Torralba, A., and Katabi, D. (2018). Through-wall human pose estimation using radio signals. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7356–7365.

Zhou, Y., Habermann, M., Xu, W., Habibie, I., Theobalt, C., and Xu, F. (2020). Monocular real-time hand shape and motion capture using multi-modal data. arXiv, arXiv:2003.09572 https://arxiv.org/abs/2003.09572.

Zimmermann, C., Schneider, A., Alyahyay, M., Brox, T., and Diester, I. (2020). Freipose: A deep learning framework for precise animal motion capture in 3d spaces. bioRxiv. https://doi.org/10.1101/2020.02.27.967620.

Zuffi, S., Kanazawa, A., Jacobs, D., and Black, M. (2016). 3d menagerie: Modeling the 3d shape and pose of animals. arXiv, arXiv:1611.07700 https://arxiv.org/abs/1611.07700.

Zuffi, S., Kanazawa, A., Berger-Wolf, T., and Black, M. (2019). Three-D Safari: Learning to Estimate Zebra Pose, Shape, and Texture from Images "In the Wild". arXiv, arXiv:1908.07201 https://arxiv.org/abs/1908.07201.