BKinD-3D: Self-Supervised 3D Keypoint Discovery from Multi-View Videos

Jennifer J. Sun* Caltech
Lili Karashchuk* U Washington
Amil Dravid* Northwestern
Serim Ryou SAIT†
Sonia Fereidooni U Washington
John C. Tuthill U Washington
Aggelos Katsaggelos Northwestern
Bingni W. Brunton U Washington
Georgia Gkioxari Caltech
Ann Kennedy Northwestern
Yisong Yue Caltech
Pietro Perona Caltech

Code & Project Website: https://sites.google.com/view/b-kind/3d

Abstract

Quantifying motion in 3D is important for studying the behavior of humans and other animals, but manual pose annotations are expensive and time-consuming to obtain. Self-supervised keypoint discovery is a promising strategy for estimating 3D poses without annotations. However, current keypoint discovery approaches commonly process single 2D views and do not operate in the 3D space. We propose a new method to perform self-supervised keypoint discovery in 3D from multi-view videos of behaving agents, without any keypoint or bounding box supervision in 2D or 3D. Our method, BKinD-3D, uses an encoder-decoder architecture with a 3D volumetric heatmap, trained to reconstruct spatiotemporal differences across multiple views, in addition to joint length constraints on a learned 3D skeleton of the subject. In this way, we discover keypoints without requiring manual supervision in videos of humans and rats, demonstrating the potential of 3D keypoint discovery for studying behavior.

1. Introduction

All animals behave in 3D, and analyzing 3D posture and movement is crucial for a variety of applications, including the study of biomechanics, motor control, and behavior [27]. However, annotations for supervised training of 3D pose estimators are expensive and time-consuming to obtain, especially for studying diverse animal species and varying experimental contexts. Self-supervised keypoint discovery has demonstrated tremendous potential in discovering 2D keypoints from video [19,20,40], without the need for manual annotations. These models have not been well-explored in 3D, which is more challenging compared to 2D due to depth ambiguities, a larger search space, and the need to incorporate geometric constraints. Our goal is to enable 3D keypoint discovery of humans and animals from synchronized multi-view videos, without 2D or 3D supervision.

*Equal contribution
†Work done outside of SAIT

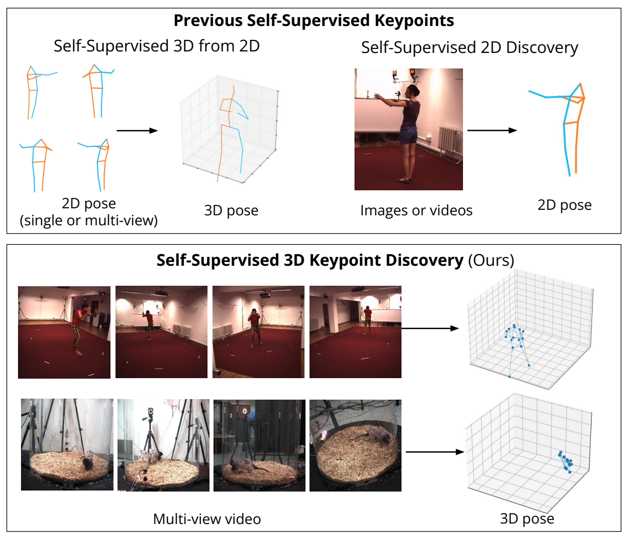

Figure 1. **Self-supervised 3D keypoint discovery**. Previous work studying self-supervised keypoints either requires 2D supervision for 3D pose estimation or focuses on 2D keypoint discovery. Currently, self-supervised 3D keypoint discovery is not well-explored. We propose methods for discovering 3D keypoints directly from multi-view videos of different organisms, such as human and rats, without 2D or 3D supervision. The 3D keypoint discovery examples demonstrate the results from our method.

Self-Supervised 3D Keypoint Discovery. Previous works for self-supervised 3D keypoints typically start from a pre-trained 2D pose estimator [25,42], and thus do not perform *keypoint discovery* (Figure 1). These models are suitable for studying human poses because 2D human pose estimators are widely available and the pose and body structure of humans is well-defined. However, for many scientific applications [27, 33, 40], it is important to track diverse organisms in different experimental contexts. These situations require time-consuming 2D or 3D annotations for training pose estimation models. The goal of our work is to enable 3D keypoint discovery from multi-view videos directly, without any 2D or 3D supervision, in order to accelerate the analysis of 3D poses from diverse animals in novel settings. To the best of our knowledge, self-supervised 3D keypoint discovery have not been well-explored for real-world multi-view videos.

Behavioral Videos. We study 3D keypoint discovery in the setting of behavioral videos with stationary cameras and backgrounds. We chose this for several reasons. First, this setting is common in many real-world behavior analysis datasets [2, 10, 21, 28, 33, 37, 39], where there has been an emerging trend to expand the study of behavior from 2D to 3D [27]. Thus, 3D keypoint discovery would directly benefit many scientific studies in this space using approaches such as biomechanics, motor control, and behavior [27]. Second, studying behavioral videos in 3D enables us to leverage recent work in 2D keypoint discovery for behavioral videos [40]. Finally, this setting enables us to tackle the 3D keypoint discovery challenge in a modular way. For example, in behavior analysis experiments, many tools are already available for camera calibration [24], and we can assume that camera parameters are known.

Our Approach. The key to our approach, which we call **B**ehavioral **K**eypo**in**t **D**iscovery in **3D** (BKinD-3D), is to encode self-supervised learning signals from videos across multiple views into a single 3D geometric bottleneck. We leverage the spatiotemporal difference reconstruction loss from [40] and use multi-view reconstruction to train an encoder-decoder architecture. Our method does not use any bounding boxes or keypoint annotations as supervision. Critically, we impose links between our discovered keypoints to discover connectivity across points. In other words, keypoints on the same parts of the body are connected, so that we are able to enforce joint length constraints in 3D. To show that our model is applicable across multiple settings, we demonstrate our approach on multi-view videos from different organisms. To summarize:

- We introduce self-supervised 3D keypoint discovery, which discovers 3D pose from real-world multi-view behavioral videos of different organisms, without any 2D or 3D supervision.
- We propose a novel method (BKinD-3D) for end-to-end 3D discovery from video using multi-view spatiotemporal difference reconstruction and 3D joint length constraints.
- We demonstrate quantitatively that our work significantly closes the gap between supervised 3D methods and 3D keypoint discovery across different organisms (humans and rats).

Method	3D sup.	2D sup.	camera params	data type
Isakov et al. [17] DANNCE [7]	✓	✓	intrinsics extrinsics	real
Rhodin et al. [35]	✓	optional	intrinsics	real
Anipose [24] DeepFly3D [12]	×	✓	intrinsics extrinsics	real
EpipolarPose [25] CanonPose [43]	×	✓	optional	real
MetaPose [42]	×	✓	×	real
Keypoint3D [3]	×	×	intrinsics extrinsics	simulation
Ours (3D discovery)	×	×	intrinsics extrinsics	real

Table 1. **Comparison of our work with representative related work for 3D pose using multi-view training**. Previous works require either 3D or 2D supervision, or simulated environments to train jointly with reinforcement learning. Our method addresses a gap in discovering 3D keypoints from real videos without 2D or 3D supervision.

2. Related Work

3D Pose Estimation. There has been a large body of work studying 3D human pose estimation from images or videos, as reviewed in [36, 44], with recent works also focusing on 3D animal poses [7, 11, 12, 24, 27]. Most of these methods are fully supervised from visual data [6, 17, 41], with some models perform lifting starting from 2D poses [4, 29, 32, 34]. We focus our discussion on multi-view 3D pose estimation methods, but all of these models require either 3D or 2D supervision during training. This 2D supervision is typically in the form of pre-trained 2D detectors [25], or ground truth 2D poses [42]. In comparison, our method uses multi-view videos to discover 3D keypoints without 2D or 3D supervision.

Methods more closely related to our work are those that also leverage multi-view structure to estimate 3D pose (Table 1). [17] proposed a supervised method that uses learnable triangulation to aggregate 2D information across views to 3D. Here we study similar approaches for representing 3D information, but using self-supervision instead of supervised 3D annotations. Other methods in this space propose training methods such as enforcing consistency of predicted poses across views [35], regression to 3D pose estimated from epipolar geometry of multi-view 2D [25], constraining 3D poses to project to realistic 2D pose [5], or estimates camera parameters using detected and ground truth 2D poses [42]. While we also leverage multi-view information, our goal is different from the work above, in that our approach aims to discover 3D poses without 2D or 3D supervision, given camera parameters.

Self-supervised Keypoint Discovery. 2D keypoint dis-

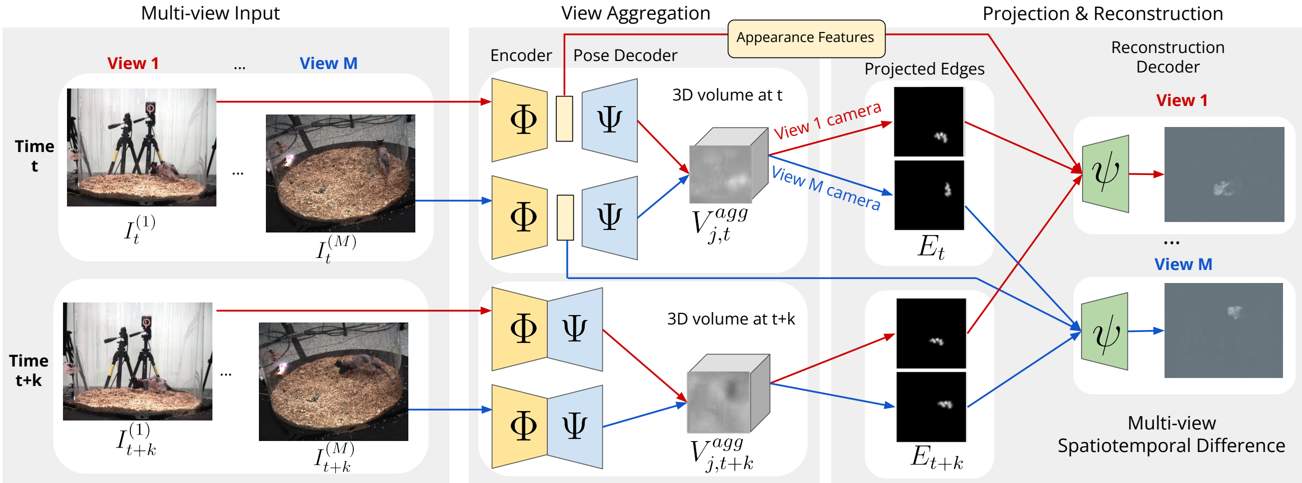

Figure 2. **BKinD-3D: 3D keypoint discovery using 3D volume bottleneck**. We start from input multi-view videos with known camera parameters, then unproject feature maps from geometric encoders into 3D volumes for timestamps t and $t + k$. We next aggregate 3D points from volumes into a single edge map at each timestamp, and use edges as input to the decoder alongside appearance features at time t. The model is trained using multi-view spatiotemporal difference reconstruction. Best viewed in color.

covery has been studied from images [13, 19, 46] and videos [20,40]. Our approach focuses on behavioral videos, similar to [40], but we aim to use multi-view information to discover 3D keypoints, instead of 2D. Many approaches use an encoder-decoder setup to disentangle appearance and geometry information [19, 26, 40, 46]. Our setup also consists of encoders and decoders, but our encoder maps information across views to aggregate 2D information into a 3D geometry bottleneck. The discovery model most similar to our approach is Keypoint3D [3], which discovers 3D keypoints for control from virtual agents, using a combination of image reconstruction and reinforcement learning. However, this setup is designed for simulated data and does not translate well to real videos, since updating the keypoints through a reinforcement learning policy requires videos generated through the simulated environment. Keypoint discovery models typically represent discovered parts as 2D Gaussian heatmaps [19, 40] or 2D edges [13]. While we also use an edge-based representation, our edges are in 3D, which enables our training objective to enforce joint length consistency.

Behavioral Video Analysis. Pose estimation is a common intermediate step in automated behavior quantification; behavioral videos are commonly captured with stationary camera and background, with moving agents. To date, supervised 2D pose estimators are most often used for analyzing behavior videos [8, 9, 14, 23, 30, 37]. However, 2D pose estimation is inadequate for many applications: it cannot reliably capture the angle of joints for kinematics, fails to generalize across views, is sensitive to occlusion, and cannot incorporate body plan constraints as skeleton length or range of motion of joints. Thus, there has recently been an accelerating trend to study behavior in 3D [7, 11, 24, 27]. These models typically require more expensive 3D training annotations compared to 2D poses. While 2D self-supervision has been studied for behavioral videos [40], 3D keypoint discovery in real-world behavioral videos have not been well-explored.

3. Method

Our goal is to discover 3D keypoints from multi-view behavioral videos without 2D or 3D supervision (Figure 2). Our approach is inspired by BKinD [40], which uses spatiotemporal difference reconstruction to discover 2D keypoints in behavioral videos. In these videos, the camera and background is stationary, and spatiotemporal difference provides a strong signal for encoding agent movement.

We develop several approaches for 3D keypoint discovery, but focus on our volumetric model (Figure 2) in this section, as this model generally performed the best in our evaluations. More details on other approaches are in Section 4.1.2 and supplemental materials.

In our volumetric model (BKinD-3D, Figure 2) we use multi-view spatiotemporal reconstruction to train an encoder-decoder architecture with 2D information aggregated to a 3D volumetric heatmap. Projections from the 3D heatmap in the form of agent skeletons are then used to reconstruct movement, represented by spatiotemporal difference, in each view.

3.1. 3D Keypoint Discovery

Given behavioral videos captured from M synchronized camera views, with known camera projection matrix $P^{(i)}$ for each camera $i \in \{1...M\}$, we aim to discover a set of J 3D keypoints $U_t \in \mathbb{R}^{J\times 3}$ on a single behaving agent, at each timestamp t. We assume access to camera projection matrices so that our model discovers 3D keypoints in the global coordinate frame.

During training, our model uses two timestamps in the video t and $t+k$ to compute the spatiotemporal difference in each view as the reconstruction target. In other words, for each camera view i, our training starts with a frame $I_t^{(i)}$ and a future frame $I_{t+k}^{(i)}$. During inference, only a single timestamp is required: once the model is trained, the model only needs $I_t^{(i)}$ for each camera view i.

In our model setup, the appearance encoder Φ, geometry decoder Ψ, and reconstruction decoder ψ are shared across views and timestamps (in previous work [40], these networks are shared across timestamps, but only a single view is addressed). The appearance encoder Φ is used to generate appearance features, which are decoded into 2D heatmaps by the geometry decoder Ψ. These 2D heatmaps are then aggregated across views to form a 3D volumetric bottleneck (Section 3.1.2), which is processed by a volume-to-volume network ρ. We compute the 3D keypoints using spatial softmax on the 3D volume. Then, we project these keypoints to 2D, compute edges between points, and output these edges into the reconstruction decoder ψ (Section 3.1.3) for training. The reconstruction decoder ψ is only used during training, and not required for inference.

3.1.1 Feature Encoding

To start, we first compute appearance features from frame pairs $I_t^{(i)}$ and $I_{t+k}^{(i)}$ using the appearance encoder Φ: $\Phi(I_t^{(i)})$ and $\Phi(I_{t+k}^{(i)})$. These appearance features are then fed into the geometry decoder Ψ to generate 2D heatmaps $\Psi(\Phi(I_t^{(i)})) = H_t^{(i)}$ and $H_{t+k}^{(i)}$. Each 2D heatmap has C channels, where $H_{t,c}^{(i)}$ represents channel c of $H_t^{(i)}$.

3.1.2 View Aggregation using Volumetric Model

To aggregate information across views, we unproject our 2D heatmaps to a 3D volumetric bottleneck. We perform view aggregation separately across timestamps t and $t+k$.

We aggregate 2D heatmaps into a 3D volume similar to [17], which used previously for supervised 3D human pose estimation. One important difference is that in the supervised setting, an $L \times L \times L$ sized volume is drawn around the human pelvis, with L being around twice the size of a person. As we perform keypoint discovery, we do not have information on the location or size of the agent. Instead, we initialize our volume with L representing the maximum size of the space/room for the behaving agent.

This process aggregates 2D heatmaps $H_{t,c}^{(i)}$ for cameras $i \in \{1...M\}$ and channels $c \in \{1...C\}$ to 3D keypoints U_t, for timestamp t. Our volume is first discretized into voxels $V_{coords} \in \mathbb{R}^{B \times B \times B \times 3}$, where B represents the number of distinct coordinates in each dimension. Each voxel corresponds to a global 3D coordinate. These 3D coordinates are projected to a 2D plane using the projection matrices in each camera view i: $V_{proj}^{(i)} = P^{(i)} V_{coords}$. A volume $V_c^{(i)}$ is then created and filled for each camera view i and each channel c using bilinear sampling [18] from the corresponding 2D heatmap: $V_c^{(i)} = H_{t,c}^{(i)}\{V_{proj}^{(i)}\}$, where $\{\cdot\}$ denotes bilinear sampling.

We then aggregate these $V_c^{(i)}$ across views for each channel c using a softmax approach [17]:

$$V_c^{agg} = \sum_i \frac{\exp(V_c^{(i)})}{\sum_j \exp(V_c^{(j)})} \odot V_c^{(i)}.$$

V^{agg} is then mapped to 3D heatmaps corresponding to each joint using a volumetric convolutional network [31] ρ: $V^{agg*} = \rho(V^{agg})$. We compute the 3D spatial softmax over the volume, for each channel j of V_j^{agg*}, $j \in \{1...J\}$, to obtain the 3D keypoint locations U_t for timestamp t, as in [17]. In many supervised works, the keypoint locations U_t are optimized to match to ground truth 3D poses; however, we aim to discover 3D keypoints, and train our network by using U_t to decode spatiotemporal difference across views.

3.1.3 Projection and Reconstruction

In this step, we project the discovered 3D keypoints to a 2D representation in each view using camera parameters. For training, 2D representations in timestamps t and $t+k$ are used as input to the reconstruction decoder ψ. We train the 3D keypoints U_t at each timestamp t using multi-view spatiotemporal difference reconstruction. The target spatiotemporal difference is computed using the 2D image pair $I_t^{(i)}$ and $I_{t+k}^{(i)}$ at each view i.

First, we project the 3D keypoints using camera projection matrices into 2D keypoints $u_t^{(i)} = P^{(i)} U_t$. We create an edge representation for each view for each timestamp, which enables us to discover connections between points and enforce 3D joint length constraints. For each keypoint pair $u_{t,m}^{(i)}$ and $u_{t,n}^{(i)}$, we draw a differentiable edge map as a Gaussian along the line connecting them, similar to [13]:

$$E_{t,(m,n)}^{(i)}(\mathbf{p}) = \exp(d_{m,n}^{(i)}(\mathbf{p})^2/\sigma^2),$$

where σ controls the line thickness and $d_{m,n}(\mathbf{p})^{(i)}$ is the distance between pixel $\mathbf{p}$ and the line connecting $u_{t,m}^{(i)}$ and $u_{t,n}^{(i)}$. We then aggregate the edge heatmaps at each timestamp using a set of learned weights $w_{m,n}$ for each edge, where $w_{m,n}$ is shared across all timestamps and all views. An edge is active and connects two points if $w_{m,n} > 0$, otherwise the points are not connected. Finally, we aggregate all edge heatmaps using the max across all edge pairs [13]:

$$E_t^{(i)}(\mathbf{p}) = \max_{m,n} w_{m,n} E_{t,(m,n)^{(i)}}(\mathbf{p}).$$

In our framework, for each view i, the decoder ψ uses the edge maps $E_t^{(i)}$ and $E_{t+k}^{(i)}$ as well as the appearance feature $\Phi(I_t^{(i)})$ for reconstructing the spatiotemporal difference across each view. The ground truth spatiotemporal difference is computed from the original images $S(I_t^{(i)}, I_{t+k}^{(i)})$. The reconstruction from the model is $\hat{S} = \psi(E_t^{(i)}, E_{t+k}^{(i)}, \Phi(I_t^{(i)}))$, through the 3D volumetric bottleneck in order to discover informative 3D keypoints for reconstructing agent movement.

3.2. Learning Formulation

The entire training pipeline (Figure 2) is differentiable, and we train the model end-to-end. We note that our model is only given multi-view video and corresponding camera parameters, without keypoint or bounding box supervision.

3.2.1 Multi-View Reconstruction Loss

Our multi-view spatiotemporal difference reconstruction is based on the single-view spatiotemporal difference studied for 2D keypoint discovery [40]. We compute the Structural Similarity Index Measure (SSIM) [45] as a reconstruction target in each view. SSIM has been used to measure perceived differences between images based on luminance, contrast, and structure features. Here, we use SSIM as a reconstruction target and we compute a similarity map using local SSIM on corresponding patches between $I_t^{(i)}$ and $I_{t+k}^{(i)}$. This similarity map is negated to obtain the dissimilarity map used as the target: $S(I_t^{(i)}, I_{t+k}^{(i)})$.

We use perceptual loss [22] in each view between the target S and the reconstruction $\hat{S}$. This loss computes the L2 distance between features of the target and reconstruction computed from the VGG network ϕ [38]:

$$\mathcal{L}_{recon}^{(i)} = \left\| \phi(S(I_t^{(i)}, I_{t+T}^{(i)})) - \phi(\hat{S}(I_t^{(i)}, I_{t+T}^{(i)})) \right\|_2 . \quad (1)$$

The error is computed by comparing features from intermediate convolutional blocks of the network. Our final perceptual loss is summed over each view $\mathcal{L}_{recon} = \sum_i \mathcal{L}_{recon}^{(i)}$.

3.2.2 Learned Length Constraint

Since many animals have a rigid skeletal structure, we encourage that the length of active edges ($w_{m,n} > 0$ for point pairs m and n) are consistent across samples. We do not assume that these lengths and connections are known, such as previous work [42]; rather, they are learned during training. We do this by maintaining a running average of the length of all active edges $l_{avg(m,n)}$, and minimizing the difference between the average length and each sample $l_{m,n}$:

$$\mathcal{L}_{length} = \sum_m \sum_n \mathbb{1}_{w_{m,n}>0} \left\| l_{avg(m,n)} - l_{m,n} \right\|_2 . \quad (2)$$

During training, we update $l_{avg(m,n)}$ using an exponential running average and $w_{m,n}$ indicating edge weights for every pair is learned. Both of these parameters are shared across all viewpoints and timestamps. Notably, the length constraint is only applied to active edges, since there are many point pairs without rigid connections (e.g. elbow to feet), while we want to enforce this constraint only for rigid connections (e.g. elbow to wrist).

3.2.3 Separation Loss

To encourage unique keypoints to be discovered, we apply separation loss to our 3D keypoints, which has been previously studied in 2D [40, 46]. On a set of 3D keypoints U_{it}, where i is the index of a keypoint and t is the time, the separation loss is:

$$\mathcal{L}_s = \sum_{i \neq j} \exp\left(\frac{-(U_{it} - U_{jt})^2}{2\sigma_s^2} \right), \quad (3)$$

where σ_s is a hyperparameter that controls the strength of separation.

3.2.4 Training Objective

Our full training objective is the sum of the multi-view spatiotemporal reconstruction loss $\mathcal{L}_{recon}$, learned length constraints $\mathcal{L}_{length}$, and separation loss $\mathcal{L}_s$:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathbb{1}_{epoch>e}(\omega_r \mathcal{L}_{length} + \omega_s \mathcal{L}_s). \quad (4)$$

Our model is trained using curriculum learning [1]. We only apply $\mathcal{L}_{length}$ and $\mathcal{L}_s$ when the keypoints are more consistent, after e epochs of training using reconstruction loss.

4. Experiments

We demonstrate BKinD-3D using real-world behavioral videos, using a human dataset and a recently released large-scale rat dataset (Section 4.1). We evaluate our discovered keypoints using a standard linear regression protocol based on previous works for 2D keypoint discovery [19, 40] (also described in Section 4.1.3). Here, we present results on pose regression (Section 4.2) with ablation studies (Section 4.3), with additional results in supplementary materials.

4.1. Experimental Setup

4.1.1 Datasets

We demonstrate our method by evaluating it on two representative datasets: Human 3.6M and Rat7M. The datasets have different environments and focus on subjects of different sizes, with humans being about 1700mm tall and rats about 250mm long.

Human 3.6M. We evaluate our method on Human3.6M to compare to recent works in self-supervised 3D from

2D [42]. Human 3.6M [15] is a large-scale motion capture dataset with videos from 4 viewpoints. We follow the standard evaluation protocol [17, 25] to use subjects 1, 5, 6, 7, and 8 for training and 9 and 11 for testing. Our test set matches the set specified in [42] using every 16th frame (8516 test frame sets). Notably, unlike baselines such as [17], our method does not require any pre-processing with 2D bounding box annotations but rather is directly applied to the full image frame.

Rat7M. We also evaluate our method on Rat7M [7], a 3D pose dataset of rats moving in a behavioral arena. This dataset most closely matches the expected use case for our method, which is a dataset of non-human animal behavior in a static environment. Rat7M consists videos from 6 viewpoints captured at 1328×1048 resolution and 120Hz, along with ground truth annotations obtained from marker-based tracking. We train on subjects 1, 2, 3, 4, and test on subject 5, as in [7]. We train and evaluate on every 240th frame of each video (3083 train, 1934 test frame sets).

4.1.2 Model Comparisons

We compare our method with three main categories of baselines: supervised 3D pose estimation methods (ex: [17]), 3D pose estimation methods from 2D supervision (ex: [42]), and a 3D keypoint discovery method developed for control in simulation [3]. A more detailed comparison of methods in this space is in Table 1. For baselines with model variations, we use evaluation results from the version that is the closest to our model (multi-view inference, and camera parameters during inference). We note that all previous methods require additional 3D or 2D supervision, or jointly training a reinforcement learning policy in simulation [3], which we do not require for 3D keypoint discovery in real videos. Another notable difference is that previous methods typically pre-process video frames using detected or ground truth 2D bounding boxes [17], while our method does not require this pre-processing step.

Since 3D keypoint discovery has not been thoroughly explored, we additionally study methods in this area using multi-view 2D discovery and triangulation (Triang.+Reproj.), and multi-view 2D discovery with a depth map estimates (Depth Map), in addition to our volumetric approach (Section 3, BKinD-3D). For multi-view 2D discovery and triangulation, we use BKinD [40] to discover 2D keypoints in each view, and perform triangulation using camera parameters to obtain 3D keypoints. We then project the 3D keypoints for multi-view reconstruction. We add an additional loss on the reprojection error to learn keypoints consistent across multiple views. For the depth map approach, in each camera view, we estimate 2D heatmaps corresponding to each keypoint alongside a view-specific depthmap estimate. The final 3D keypoints are then computed from a confidence-weighted average of each view's estimated 3D keypoint coordinates (from the per-view 2D heatmaps and depth estimates). More details on each method are in the supplementary materials.

4.1.3 Training and Evaluation Procedure

We train our volumetric approach using the full objective (Eq 4). We scale images to 256×256 for training, with a frame gap of 0.4s for Human3.6M and 0.66s for Rat7M. We use a maximum volume size of 7500mm for Human3.6M and 1000mm for Rat7M. The results are computed for all 3D keypoint discovery methods with 15 keypoints unless otherwise specified. We train using videos from the train split with camera parameters provided by each dataset.

We evaluate our 3D keypoint discovery through keypoint regression based on similar methods from 2D, using a linear regressor without a bias term [19, 40, 46]. For this regression step, we extract our discovered 3D keypoints from a frozen network, and learn a linear regressor to map our discovered keypoints to the provided 3D keypoints in each of the training sets. We then perform evaluation on regressed keypoints on the test set.

For metrics, we compute Mean Per Joint Position Error (MPJPE) in line with previous works in 3D pose estimation [16, 17], which is the L2 distance between the regressed and ground truth 3D poses, accounting for the mean shift between the regressed and ground truth points. To compare to methods that require addition alignment before MPJPE computation (e.g. [42] which does not use camera parameters during inference), we also compute Procrustes aligned MPJPE (PMPJPE) [16,25,42]. PMPJPE applies the optimal rigid alignment to the predicted and ground truth 3D poses before metric computation.

4.2. Results

We evaluate our discovered keypoints quantitatively using keypoint regression on Human3.6M (Table 2) and Rat7M (Table 3). Over both datasets with diverse organisms, our approach generally outperforms all other fully self-supervised 3D keypoint discovery approaches. Additionally, among all the approaches we developed for 3D keypoint discovery, BKinD-3D using the volumetric bottleneck performs the best overall. Results demonstrate that BKinD-3D is directly applicable to discover 3D keypoints on novel model organisms, potentially very different in appearance or size, without 2D or 3D supervision.

Notably, on Humam3.6M, Keypoint3D [3], developed for control of simulated videos, does not work well in our setting with real videos, and qualitative results demonstrate that this method was not able to discover keypoints that tracked the agent (supplementary materials).

Qualitative results. We find that the discovered points and skeletons are reasonable and look similar to the ground truth annotations for Human3.6M (Figure 3) and Rat7M

Method	Supervision	PMPJPE ↓	MPJPE ↓
	Supervised 3D		
Anipose [24]	2D only	-	33
Rhodin et al. [35]	3D/2D	52	67
Isakov et al. [17]	3D/2D	-	21
	Supervised 2D + self-supervised 3D		
CanonPose [43]	2D	53	74
EpipolarPose [25]	2D	67	77
Iqbal et al. [16]	2D	55	69
MetaPose [42]	2D	74	-
	3D Discovery + Regression		
Keypoint3D [3]	×	168	368
Ours:			
Triang+reproj	×	134	241
Depth Map	×	122	161
BKinD-3D	×	105	125

Table 2. **Comparing performance with related work on Human3.6M**. We note that previous approaches typically require additional 2D or 3D supervision, whereas our model discovers 3D keypoints directly from multi-view video. The 3D keypoint discovery models are evaluated using a linear regression protocol (Section 4.1.3).

(Figure 4). Furthermore, we find that a volumetric model with 30 keypoints learns a more detailed human skeleton representation than a model with 15 keypoints. For example, the model with 30 keypoints is able to track both legs, while the 15 keypoint model only tracks 1 leg; however, both models miss the knees. Importantly, our model discovers the skeleton in global coordinates, and is able to track the agent as they move around the space. More examples are in supplementary materials.

While there exists a gap in terms of quantitative metrics between supervised methods and self-supervised 3D keypoint discovery, supervised methods require users to invest time and resources for annotations. In comparison, our method can be deployed out-of-the-box on new datasets and experiments with multi-view cameras. Our approach has closed the gap substantially to supervised methods compared to previous work, without requiring time-consuming 2D or 3D annotations. Qualitative results demonstrate that our approach is able to discover structure across diverse model organisms, providing a method for accelerating the study of organism movements in 3D.

Downstream Analysis. To further evaluate our keypoint discovery method, we use BKinD-3D keypoints as input to a 1D convolutional neural network (previously used in [39]) to predict action labels on Human3.6M. Notably, we found that our keypoints performs similarly to ground truth 3D points for action recognition, where Top 5 accuracy is 64.8% (GT), 61.0% (15 kpts), and 64.9% (30 kpts) (supplementary material).

Method	Supervision	PMPJPE ↓	MPJPE ↓
	Supervised 3D		
DANNCE [7]	3D	11	-
	3D Discovery + Regression		
Ours:			
Triang+reproj	×	21	108
Depth Map	×	27	56
BKinD-3D	×	24	76

Table 3. **Comparison with 3D keypoint discovery methods on Rat7M**. Results from the top three 3D keypoint discovery methods on Rat7M. The 3D keypoint discovery models are evaluated using a linear regression protocol (Section 4.1.3).

Method	PMPJPE ↓	MPJPE ↓
BKinD-3D (8 kpts)	120	149
BKinD-3D (15 kpts)	105	125
BKinD-3D (30 kpts)	109	130
BKinD-3D (point)	110	137
BKinD-3D (edge, without length)	108	129
BKinD-3D (edge, full objective)	105	125

Table 4. **Ablation results on Human3.6M**. We perform an ablation study of our volumetric bottleneck method comparing different numbers of keypoints as well as variations to the edge bottleneck with length constraints.

4.3. Ablation

We perform an ablation study of our model (Table 4), focused on BKinD-3D as it is the best performing approach on Human3.6M. Results show that 15 keypoints performed the best quantitatively, but 30 keypoints is comparable and qualitatively provides a more informed skeleton (Figure 3). We perform additional regression experiments using a 2-layer MLP regressor (supplementary material), and we found that the keypoints discovered by the 30 keypoints model (94 PMPJPE) perform better relative to 15 keypoints (98 PMPJPE). This suggests that the linear model may have been underfitting our 30 keypoints model.

We additionally find that adding edge information has a quantitative improvement on performance and provides more qualitative information on connectivity between joints (Figures 3, 4). In our 3D setting, we found that the point bottleneck (studied in previous works in 2D [19, 40]) did not work as well as the edge bottleneck (studied in previous works in 2D [13]). By studying edge bottlenecks in 3D and expanding beyond 2D, our approach is able to enforce joint length constraints through the discovered edge connectivity.

5. Discussion

We present a method for 3D keypoint discovery directly from multi-view video, without any requirement for 2D or

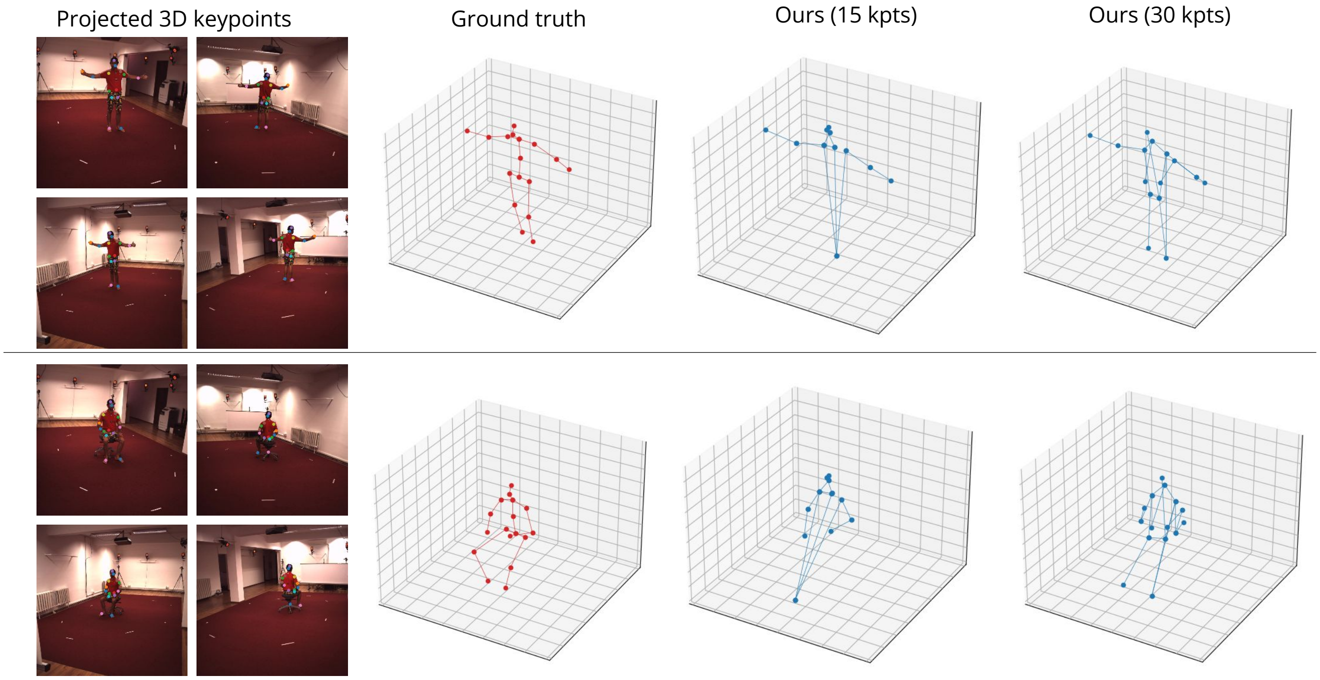

Figure 3. **Qualitative results for 3D keypoint discovery on Human3.6M**. Representative samples of 3D keypoints discovered from BKinD-3D without regression or alignment for 15 and 30 total discovered keypoints. We visualize all keypoints that are connected using the learned edge weights, and the projected 3D keypoints in the leftmost column are from the keypoint model with 30 discovered keypoints.

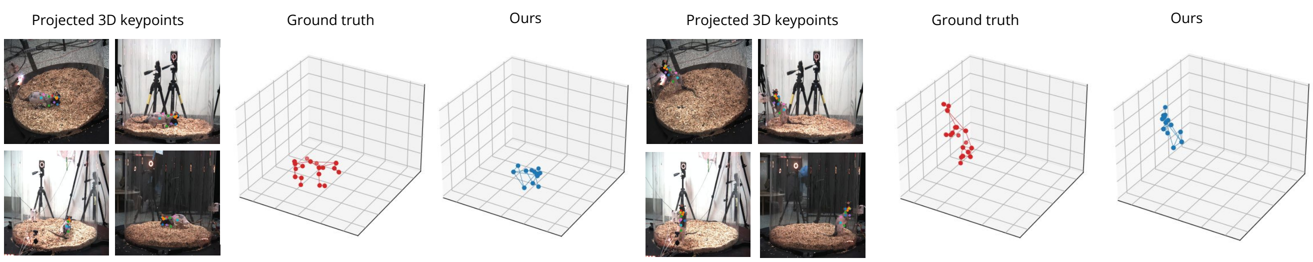

Figure 4. **Qualitative results for 3D keypoint discovery on Rat7M**. Representative samples of 3D keypoints discovered from BKinD-3D without regression or alignment. We visualize all connected keypoints using the learned edge weights and visualize the first 4 cameras (out of 6 cameras) in Rat7M for projected 3D keypoints.

3D supervision. Our method discovers 3D keypoint locations as well as joint connectivity in behaving organisms using a volumetric heatmap with multi-view spatiotemporal difference reconstruction. Results show that our work has closed the gap significantly to supervised methods for studying 3D pose, and is applicable to different organisms.

Our approach focuses on behavioral videos with stationary cameras and background, with known camera parameters. The applicability of 3D keypoint discovery can be further improved with future work to jointly estimate camera parameters, camera movement, and pose from visual data. Additionally, the lack of publicly available multi-view datasets of animals could limit model development and evaluation. Open-sourcing more datasets in this area would encourage the development of pose estimation models with broader impacts beyond humans. Despite these challenges, 3D keypoint discovery has the potential to enable studying behavior of diverse organisms, without the need for expensive and time-consuming annotations. Our goal is to encourage more efforts in 3D keypoint discovery, to study the capabilities of vision models and to facilitate the study of behavior in new organisms and across diverse experimental setups.

6. Acknowledgements

This work is generously supported by the Amazon AI4Science Fellowship (to JJS), NIH NINDS (R01NS102333 to JCT), the Air Force Office of Scientific Research (AFOSR FA9550-19-1-0386 to BWB), and NSF (1918865 to YY).

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 5

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

[3] Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3d keypoints for control. In *International Conference on Machine Learning*, pages 1539–1549. PMLR, 2021. 2, 3, 6, 7

[4] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017. 2

[5] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5714–5724, 2019. 2

[6] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3279–3288, 2020. 2

[7] Timothy W Dunn, Jesse D Marshall, Kyle S Severson, Diego E Aldarondo, David GC Hildebrand, Selmaan N Chettih, William L Wang, Amanda J Gellis, David E Carlson, Dmitriy Aronov, et al. Geometric deep learning enables 3d kinematic profiling across species and environments. *Nature methods*, 18(5):564–573, 2021. 2, 3, 6, 7

[8] SE Roian Egnor and Kristin Branson. Computational analysis of behavior. *Annual review of neuroscience*, 39:217–236, 2016. 3

[9] Eyrun Eyjolfsdottir, Kristin Branson, Yisong Yue, and Pietro Perona. Learning recurrent representations for hierarchical behavior modeling. *ICLR*, 2017. 3

[10] Eyrun Eyjolfsdottir, Steve Branson, Xavier P Burgos-Artizzu, Eric D Hoopfer, Jonathan Schor, David J Anderson, and Pietro Perona. Detecting social actions of fruit flies. In *European Conference on Computer Vision*, pages 772–787. Springer, 2014. 2

[11] Adam Gosztolai, Semih Günel, Victor Lobato-Ríos, Marco Pietro Abrate, Daniel Morales, Helge Rhodin, Pascal Fua, and Pavan Ramdya. Liftpose3d, a deep learning-based approach for transforming two-dimensional to three-dimensional poses in laboratory animals. *Nature methods*, 18(8):975–981, 2021. 2, 3

[12] Semih Günel, Helge Rhodin, Daniel Morales, João Campagnolo, Pavan Ramdya, and Pascal Fua. DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. *eLife*, 8:e48571, Oct. 2019. 2

[13] Xingzhe He, Bastian Wandt, and Helge Rhodin. Autolink: Self-supervised learning of human skeletons and object outlines by linking keypoints. *arXiv preprint arXiv:2205.10636*, 2022. 3, 4, 7

[14] Weizhe Hong, Ann Kennedy, Xavier P Burgos-Artizzu, Moriel Zelikowsky, Santiago G Navonne, Pietro Perona, and David J Anderson. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proceedings of the National Academy of Sciences*, 112(38):E5351–E5360, 2015. 3

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 6

[16] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3d human pose learning via multi-view images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5243–5252, 2020. 6, 7

[17] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. *arXiv preprint arXiv:1905.05754*, 2019. 2, 4, 6, 7

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015. 4

[19] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 3, 5, 6, 7

[20] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

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

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[23] Mayank Kabra, Alice A Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. Jaaba: interactive machine learning for automatic annotation of animal behavior. *Nature methods*, 10(1):64, 2013. 3

[24] Pierre Karashchuk, Katie L Rupp, Evyn S Dickinson, Sarah Walling-Bell, Elischa Sanders, Eiman Azim, Bingni W Brunton, and John C Tuthill. Anipose: a toolkit for robust markerless 3d pose estimation. *Cell reports*, 36(13):109730, 2021. 2, 3, 7

[25] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1077–1086, 2019. 1, 2, 6, 7

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

[27] Jesse D Marshall, Tianqing Li, Joshua H Wu, and Timothy W Dunn. Leaving flatland: Advances in 3d behavioral measurement. *Current Opinion in Neurobiology*, 73:102522, 2022. 1, 2, 3

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

[29] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 2

[30] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie W. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 2018. 3

[31] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018. 4

[32] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 2

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

[34] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018. 2

[35] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3d human pose estimation from multi-view images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8437–8446, 2018. 2, 7

[36] Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A Kakadiaris. 3d human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016. 2

[37] Cristina Segalin, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J. Sun, Pietro Perona, David J. Anderson, and Ann Kennedy. The mouse action recognition system (mars): a software pipeline for automated analysis of social behaviors in mice. *bioRxiv https://doi.org/10.1101/2020.07.26.222299*, 2020. 2, 3

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[39] Jennifer J Sun, Tomomi Karigo, Dipam Chakraborty, Sharada P Mohanty, David J Anderson, Pietro Perona, Yisong Yue, and Ann Kennedy. The multi-agent behavior dataset: Mouse dyadic social interactions. *arXiv preprint arXiv:2104.02710*, 2021. 2, 7

[40] Jennifer J Sun, Serim Ryou, Roni H Goldshmid, Brandon Weissbourd, John O Dabiri, David J Anderson, Ann Kennedy, Yisong Yue, and Pietro Perona. Self-supervised keypoint discovery in behavioral videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2171–2180, 2022. 1, 2, 3, 4, 5, 6, 7

[41] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 2

[42] Ben Usman, Andrea Tagliasacchi, Kate Saenko, and Avneesh Sud. Metapose: Fast 3d pose from multiple views without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6759–6770, 2022. 1, 2, 5, 6, 7

[43] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2021. 2, 7

[44] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 210:103225, 2021. 2

[45] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600–612, 2004. 5

[46] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018. 3, 5, 6