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

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

We present additional discussions (Section 1), additional experimental results (Section 2), method description for the approaches we studied for 3D keypoint discovery in addition to the volumetric method (Section 3), additional implementation details (Section 4), and qualitative results (Section 5).

1. Additional Discussion

Limitations and Future Directions. Currently, our approach uses multi-view videos with camera parameters for training and focuses on behavioral videos with stationary cameras and backgrounds. Future directions to jointly estimate camera parameters, camera movement, and pose from visual data can improve the applicability of 3D keypoint discovery. We were also limited by the small amount of publicly available multi-view datasets of non-human animals. More open datasets in this space would encourage the development of pose estimation models with broader impacts beyond humans. Finally, during our model training, once an edge between points becomes non-activated, then it is not displayed in the edge heatmap. To activate additional keypoints, one approach could be to perform random dropout on learned features of keypoints without activated edges and reset learned edge weights during training to activate additional keypoints. While challenges exist, we highlight the potential for 3D keypoint discovery in studying the 3D movement of diverse organisms without supervision.

Broader Impacts. 3D keypoint discovery has the potential to accelerate the study of agent movements and behavior in 3D [7], since these methods does not require time-consuming manual annotations for training. Additionally, behavioral scientists have long used summarizations of an animal's skeleton in order to analyze behavior, as tracking the full skeleton is often impractical or infeasible. To classify behavior, they have relied on estimates of center of mass trajectory, PCA features, or even raw image frames [9]. This advance enables scientists to study behavior in novel organisms and experimental setups, for which annotations and pre-trained models are not available. However, risks are inherent in applications of behavior analysis, especially regarding human behavior, and thus important considerations must be taken to respect privacy and human rights. In research, responsible use of these models involves being informed and following policies, which often includes obtaining internal review board (IRB) approval, as well as obtaining written informed consent from human participants in studies. Overall, we hope to inspire more efforts in self-supervised 3D keypoint discovery in order to understand the capabilities and limitations of vision models as well as enable new applications, such as studying natural behaviors of organisms from diverse taxa in biology.

2. Additional Experimental Results

We perform additional experiments of BKinD-3D on Human3.6M and Rat7M using our keypoint discovery model, focusing on the volumetric approach. We evaluate our keypoints using the 3D keypoint regression procedure specific in the main paper, unless otherwise specified.

*Equal contribution
†Work done outside of SAIT

	PMPJPE (MLP) ↓	MPJPE (MLP) ↓
BKinD-3D (15 kpts)	98	116
BKinD-3D (30 kpts)	94	111

Table 1. **MLP regressor results on Human3.6M**.

	Top 1 Accuracy ↑	Top 5 Accuracy ↑
BKinD-3D (15 kpts)	31.8	61.1
BKinD-3D (30 kpts)	36.5	64.9
Ground truth 3D kpts	43.5	64.8

Table 2. **Action recognition results on Human3.6M**.

2.1. Keypoint regression using MLP vs linear model

We additionally evaluate the volumetric model trained on Human 3.6M using a 2-layer multilayer perceptron (MLP) for keypoint regression. Our MLP network has 50 hidden units as our regressor. We train the regressor on the train subjects and evaluate on unseen subjects (Table 1), matching the procedure for linear regressor in the main paper. Using the MLP regressor, we find that the keypoints discovered by the 30 keypoints model perform better relative to with 15 keypoints (in the evaluation of the main paper the regressor was a linear model). This suggests that the linear model may have been underfitting our 30 keypoints model.

2.2. Action recognition results

We compare our discovered keypoints to ground truth 3D keypoints as input to a 1D Convlutional Network (previously used in [11]) to predict action labels on Human3.6M (Table 2). For train and test split, we use the same subject split as the main paper (subject 1,5,6,7,8 for train, and subject 9, 11 for test), and extract keypoints for all frames to classify actions. The action classification network is a 3-layer 1D convolution with a window size of 15 with hidden dimensions 128, 64, and 32. We find that 30 keypoints perform better for action recognition than 15 keypoints, and in particular, performs comparably to ground truth 3D keypoints in this setting.

2.3. Varying number of cameras

During both training and inference. On Human3.6M (Table 3), we vary the number of cameras from 4 to 2, and compute the mean performance over all camera pairs. For the 4 camera experiment, we used all 4 cameras for training and inference, while for the 2 camera experiment, we used the same selections of 2 cameras for training and inference. The mean performance with 2 cameras is slightly lower than using all cameras. Notably, on the best performing camera pair, we observe that the performance is similar to using all 4 cameras. This result is promising for 3D keypoint discovery in settings that might limit the number of cameras, such as due to cost of additional cameras, maintenance effort, or difficulty of hardware setups.

During inference only. We vary the number of camera during inference only, training on all 4 cameras and using pairs of 2 cameras for inference (Table 3). We compute the mean, max, and min performance over all camera pairs. We note that this 2 camera inference setup performed slightly better compared to training on camera pairs and performing inference on the same 2 cameras. These results suggests that in experimental setups where cameras might fail or removed during recording, it may still benefit the model to be trained on all views, but then inference can be performed without re-training on the remainig views.

2.4. Error distribution across joints

We visualize the error distribution across joint types from our 3D volumetric model (Figure 1). We observe that generally joints on the limbs (e.g. wrist, ankle) have higher errors than joints closer to the center of the body (e.g. thorax, neck), for both MPJPE and PMPJPE. This could be due to the wider range of motion of these limbs compared to the center in Human3.6M. There is not a significant difference in error across joints on the left side or right side. Since we currently perform inference

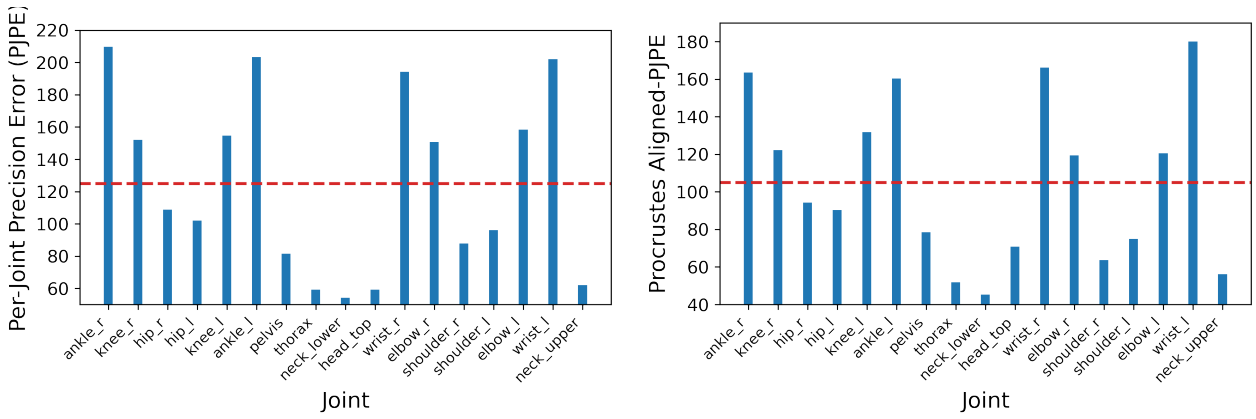

Figure 1. **Per joint errors on Human3.6M**. Errors of each joint in mm using BKinD-3D, corresponding to the skeleton definition from the Human3.6M dataset. The dotted red line corresponds to the mean across joints (the MPJPE and PMPJPE respectively).

per frame, future work to incorporate temporal constraints, or extend our method to identify meshes without supervision, could reduce errors on the limbs.

2.5. Training with different keypoint counts in Rat7M

On Rat7M (Table 4), we compare model performance when discovering 15 keypoints and 30 keypoints. We observe a small improvement in PMPJPE and MPJPE with an increased number of discovered keypoints, and also observe that the discovered keypoints cover a greater portion of the rat body in qualitative results (Figure 5). This is similar to our observations on varying keypoints on Human 3.6M (Figure 3 in the main paper). It is possible that further increasing the number of keypoints could lead to a better body representation. Future work that explores using more efficient models with a much higher number of learned keypoints could further improve performance.

Method	PMPJPE ↓	MPJPE ↓
BKinD-3D (4 cams)	105	125
Train and inference with 2 cams		
BKinD-3D (2 cams) mean	117	155
(2 cams) best	108	133
(2 cams) worst	125	167
Train with 4 cams, inference with 2 cams		
BKinD-3D (2 cams) mean	114	153
(2 cams) best	103	130
(2 cams) worst	121	160

Table 3. **Camera variations on Human3.6M**. We vary the number of cameras used for training and inference, as well as during inference only (trained with 4 cameras). Since there are multiple choices of 2 camera configurations, we chose the mean, best, and worst performance metrics.

Method	PMPJPE ↓	MPJPE ↓
BKinD-3D (15 kpt)	24	76
BKinD-3D (30 kpt)	23	70

Table 4. **Additional results on Rat7M**. We vary the number of discovered keypoints.

Method	PMPJPE ↓	MPJPE ↓
BKinD-3D (32 volume features)	105	125
BKinD-3D (64 volume features)	107	125

Table 5. **Varying volumetric representation size on Human3.6M.**

	PMPJPE (linear) ↓	PMPJPE (MLP) ↓	MPJPE (linear) ↓	MPJPE (MLP) ↓
BKinD-3D ($\sigma = 0.04$)	106	98	128	118
BKinD-3D ($\sigma = 0.08$)	105	98	125	116
BKinD-3D ($\sigma = 0.16$)	106	100	128	122

Table 6. **Varying σ in separation loss on Human 3.6M**.

2.6. Other hyperparameter variations

Volumetric representation. We evaluate the performance of our model when varying the size of the volumetric features (Table 5). We did not observe a significant difference in performance with a bigger volumetric representation. This volume feature corresponds to C, which is the number of channels of the volumetric representation before input to the volume-to-volume network ρ.

Varying σ in separation loss. The value of σ in the separation loss (section 3.2.3) controls the separation of the learned keypoints. For our experiments, we used a value of $\sigma = 0.08$ based on previous approaches in 2D [12]. To evaluate the effect of this hyperparameter, we trained the BKinD-3D model on Human 3.6M with 15 keypoints with varying values of σ, evaluating with both our standard linear model and the MLP model described above (Table 6). We find that changing the value of σ does not change the test error signficantly. Qualitatively, we did find that the recovered keypoints were more evenly spread out on the human body for lower values of σ (corresponding to a greater effect of the separation loss).

3. Additional 3D Keypoint Discovery Approaches

3.1. Triangulation and reprojection

One of the simplest approaches to extending current 2D keypoint discovery methods [12] to three dimensions is to triangulate the discovered 2D keypoints to obtain 3D keypoints, then reproject the points back to 2D. This model can be trained using the same loss (spatiotemporal difference reconstruction) using the discovered 2D keypoints and the projected 2D keypoint in each view. We implement this approach, along with an additional loss for minimizing reprojection error to encourage detecting consistent keypoints across views.

We use an encoder-decoder architecture, with a shared appearance encoder Φ, geometry decoder Ψ, and reconstruction decoder ψ. For each camera view i and time t, a frame $I_t^{(i)}$ is processed to obtain a heatmap $H_t^{(i)} = \Psi(\Phi(I_t^{(i)}))$. We apply a spatial softmax to obtain 2D keypoints $y_t^{(i)}$ for each view. The 2D keypoints across all views are triangulated to produce 3D keypoints. The triangulation is done by applying singular value decomposition (SVD) to find a solution to the following problem:

$$\text{argmin}_{\tilde{z}_t^{(i)}} ||y_t^{(i)} - P^{(i)}\tilde{U}_t||_2$$

where $\tilde{U}_t$ represents the 3D keypoints in homogeneous coordinates and $P^{(i)}$ the projection matrix for camera view i. The 3D keypoints are projected back into 2D for each view forming $y*_t^{(i)}$.

To train the network, we minimize a sum of three losses:

- $\mathcal{L}_{recon}^{(i)}$: the multi-view reconstruction loss (described in Section 3.2.1 of the main paper) using the detected 2D keypoints $y_t^{(i)}$ and $y_{t+k}^{(i)}$

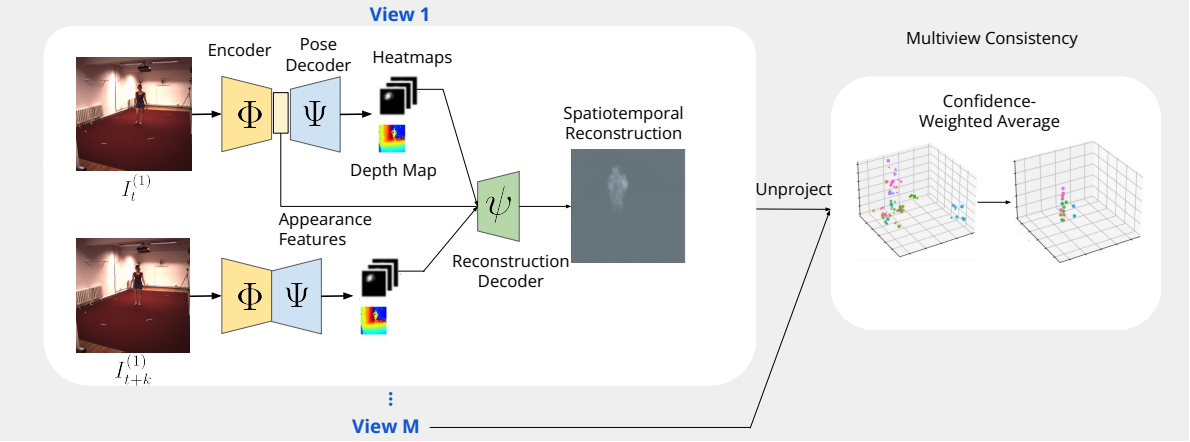

Figure 2. **3D keypoint discovery using depth maps**. The model is trained using multi-view spatiotemporal difference reconstruction to learn 2D heatmaps and depth representations at each view. Then the 3D information from each view is aggregated using a confidence-weighted average to produce the final 3D pose.

- $\mathcal{L}^{(i)}_{projrecon}$: the same multi-view reconstruction loss as above, but applied to the projected 2D keypoints $y*^{(i)}_t$ and $y*^{(i)}_{t+k}$
- $\mathcal{L}^{(i)}_{reproj} = ||y*^{(i)}_t - y^{(i)}_t||_2$: the reprojection error
- $\mathcal{L}_s$: the separation loss (described in Section 3.2.3 of the main paper)

The final loss is

$$\mathcal{L} = \sum_i \mathcal{L}^{(i)}_{recon} + \mathbb{1}_{epoch>e}(\omega_s \mathcal{L}_s + \omega_p \sum_i \mathcal{L}^{(i)}_{projrecon} + \omega_r \sum_i \mathcal{L}^{(i)}_{reproj})$$

Our model is trained using curriculum learning [1]. We only apply the losses based on projected 2D points after e epochs, when the model learns some consistent keypoints with each view. We train our model for 5 epochs and apply the losses after $e = 2$ epochs.

3.2. Depth Approach

Based on the success in 2D unsupervised behavioral video keypoint discovery [12] and 3D keypoint discovery for robotic control [2], we experiment with a framework that encodes appearance as well as 2D and depth representations (Figure 2). Given multiple camera views with known extrinsic and instrinsic parameters, our framework learns 2D keypoints and depth maps to estimate 3D keypoints.

For each camera i, there is an appearance encoder $\Phi^{(i)}$, a pose decoder $\Psi^{(i)}$, and a depth decoder $D^{(i)}$. A frame $I^{(i)}_t$ and future frame $I^{(i)}_{t+k}$ are fed into the appearance encoder and subsequently the pose decoder. The pose decoder outputs J heatmaps corresponding to the J keypoints: $\Psi^{(i)}(\Phi^{(i)}(\cdot))$. A spatial softmax operation is applied to the output of the pose decoder, representing confidence or a probability distribution for the location of each keypoint. We interpret each of the heatmaps as a 2D Gaussian. The depth decoder outputs one depth map $D(\Phi(\cdot))$, representing a dense prediction of distance from the camera plane for the scene. The appearance features $\Phi^{(i)}(I^{(i)}_t)$ are fused with the 2D geometry features for both I_t and I_{t+k}. These are fed into the reconstruction decoder ψ to reconstruct the 2D spatiotemporal difference between I_t and I_{t+k}. The spatiotemporal difference encourages the network to focus on meaningful regions of movement and be invariant to the background and other irrelevant features. This framework is repeated across camera views.

The reconstruction objective uses spatiotemporal difference reconstruction similar to our volumetric approach. To make the model more robust to rotation, we rotate the geometry bottleneck h_g for image I to create pseudo labels $h^{R^\circ}_g$ for the rotated input images I^{R°. where $R = 90°, 180°, 270°$. We apply mean squared error between the predicted geometry bottlenecks $\hat{h}_g$ and the rotated images and the generated pseudo labels h_g:

$$\mathcal{L}_{rot} = \text{MSE}(h^{R^\circ}_g, \hat{h}_g(I^{R^\circ})) \tag{1}$$

Type	Input dimension	Output dimension	Output size
Upsampling	-	-	16x16
Conv_block	2048 + # keypoints $\times$ 2	1024	16x16
Upsampling	-	-	32x32
Conv_block	1024 + # keypoints $\times$ 2	512	32x32
Upsampling	-	-	64x64
Conv_block	512 + # keypoints $\times$ 2	256	64x64
Upsampling	-	-	128x128
Conv_block	256 + # keypoints $\times$ 2	128	128x128
Upsampling	-	-	256x256
Conv_block	128 + # keypoints $\times$ 2	64	256x256
Convolution	64	3	256x256

Table 7. **Reconstruction decoder architecture.** "Conv_block" refers to combination of 3$\times$3 convolution, batch normalization, and ReLU activation. This architecture setup is also used for reconstruction decoding in [10, 12].

The rotational loss can lead to a degenerate solution, with the keypoints converging to the center of the image. As such, we employ a separation loss as was done in our volumetric method.

For camera i and a 3D point (x, y, z) in the world coordinate system, we can use the projection matrix $P^{(i)}$ to project the 3D point to camera i's normalized coordinate system (u, v, d). Let the $\Omega^{(i)}$ operator denote the transformation to the camera plane and $\Omega^{*(i)}$ denote the inverse transformation. These transformations are differentiable and can be expressed analytically [2].

After outputting the 2D keypoint heatmap $\Psi(\Phi(\cdot))$ and the depth map $D(\Phi(\cdot))$ for an input frame, we integrate over the probability distributions on the $\mathbb{R}^{S\times S}$ heatmaps and the depth maps to get the expected value for each coordinate j and camera i:

$$\mathbb{E}[u_j^{(i)}] = \frac{1}{S}\sum_{u,v} u \cdot H_j^{(i)}(u, v) \tag{2}$$

$$\mathbb{E}[v_j^{(i)}] = \frac{1}{S}\sum_{u,v} v \cdot H_j^{(i)}(u, v) \tag{3}$$

$$\mathbb{E}[d_j^{(i)}] = \sum_{u=1}^{S}\sum_{v=1}^{S} D_j^{(i)}(u, v) \cdot H_j^{(i)}(u, v) \tag{4}$$

The keypoints are unprojected into the world coordinate system: $\Omega^{-}1_n(u, v, d)$. To penalize disagreement between predictions from different views, we use a multi-view consistency loss via mean-squared error.

4. Additional Implementation Details

Architecture Details. Our model architecture is based on ones studied before for 2D keypoint discovery [10, 12]. Our encoder Φ is a ResNet-50 [5], which outputs our appearance features. Our pose decoder Ψ uses GlobalNet [3], which outputs our 2D heatmaps. Our volume-to-volume network ρ is based on V2V [8]. Finally, our reconstruction decoder ψ is a a series of convolution blocks, where the architecture details are in Table 7. Our code is available at https://github.com/neuroethology/BKinD-3D.

Hyperparameters. The hyperparameters for the volumetric 3D keypoint discovery model is in Table 8. All keypoint discovery models are trained until convergence, with 5 epochs for Human3.6M and 8 epochs for Rat 7M. We use $\sigma = 0.08$ for the keypoint separation hyperparameter based on previous works [12]. We include additional details on each dataset:

Human3.6M. The Human 3.6M dataset [6] contains 3.6 million frames of 3D human poses with corresponding video captured from 4 different camera views, recorded from a set of different scenarios (discussion, sitting, eating, ...). Each scenario consists of videos from all 4 views with the same background, across a set of human participants. The person in

Dataset	# Keypoints	Batch size	Volume dimension	Volume size	Resolution	Frame Gap	Learning Rate
Human3.6M	15	1	7500	64	256	20	0.001
Rat7M	15	1	1000	64	256	80	0.001

Table 8. **Hyperparameters for 3D Keypoint Discovery.**

the video is approximately 1700mm tall while the room is approximately 4000mm in dimension. The dataset is captured at 50Hz. This dataset is licensed for academic use, and more details on the dataset and license are provided by the Human 3.6M authors within [6].

Rat7M. The Rat7M dataset [4] consists of 3D pose and videos from a behavioral experiment with a set of rats, recorded across 6 views. This is currently one of the largest dataset with animal 3D poses. The dataset consists of 5 rats, with videos from some of the rats across multiple days. The rats are approximately 250mm long with the cage being around 1000mm in dimension. The video is captured at 120Hz. Some of the ground truth poses in Rat7M contains nans, and during processing, similar to [4], we remove frames with nans from evaluation. Our training procedure is not affected since we do not use any 3D poses during training. This dataset is open-sourced for research.

5. Qualitative Results

We present additional qualitative results from BKinD-3D in Figures 4 and 5. For Human 3.6M (Figure 4), qualitative results demonstrate that the volumetric method discovers 3D keypoints and connections that qualitatively match the ground truth, even with self-occlusion or unusual poses, such as when the subject is laying or sitting down. The 30 keypoint model generally tracks the legs, shoulders, hips, arms, and head of the subject. The 15 keypoint model tracks the shoulders, arms, and head of the subject but fails to discover the legs and hips. This may be because we use spatiotemporal difference reconstruction, and there is more movements in these discovered parts. We observe that most discovered edges correspond to limbs, although there are extra discovered edges within the body. For example, the shoulders to feet connection in the 15 keypoint model. This edge likely allows the volumetric bottleneck to model the human shape with the limited keypoints available. In addition to extra edges that may be discovered by our model, we may also miss parts, such as the knees of the subject, and occasionally the wrist keypoints (e.g. the left wrist for both 15 and 30 keypoint model in the last row). Despite this, we note that the discovered skeleton is reasonable across a wide range of poses.

In contrast to the volumetric bottleneck, the method Keypoint3D [2] does not work well on our real videos. In [2], Keypoint3D jointly trains image reconstruction with a reinforcement learning (RL) policy loss in simulated environments. We find that training in real videos using only image reconstruction leads to poor performance: the discovered keypoints do not track any semantically meaningful parts (Figure 3).

For Rat7M (Figure 5), we also find that the volumetric bottleneck discovers interpretable keypoints that qualitatively match the ground truth. The head and front legs in particular are well tracked in both 15 and 30 keypoint models, across a variety of rat poses from having 4 feet on the ground to crouching to standing up. However, the back legs are only partially discovered in the 30 keypoint model. Furthermore, the discovered rat skeleton has much more edges compared to the ground truth. This highlights a limitation in our model, as the rat's skin and fat hide its underlying skeleton, making it difficult to discover the skeleton from video data alone. Future work could explore applying self-supervised learning constrained by body priors, such as animal X-rays, in order to discover a more precise skeleton.

Overall, qualitative results from the volumetric method demonstrates the potential of 3D keypoint discovery for discovering the pose and structure of different agents without supervision, across organisms that are significantly different in appearance and scale.

References

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

[2] 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. 5, 6, 7, 8

[3] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 6

[4] 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. 7

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, 2016. 6

[6] 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, 7

[7] 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

[8] 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. 6

[9] Talmo D Pereira, Joshua W Shaevitz, and Mala Murthy. Quantifying behavior to understand the brain. *Nature neuroscience*, 23(12):1537–1549, 2020. 1

[10] Serim Ryou and Pietro Perona. Weakly supervised keypoint discovery. *CoRR*, abs/2109.13423, 2021. 6

[11] 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

[12] 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. 4, 5, 6

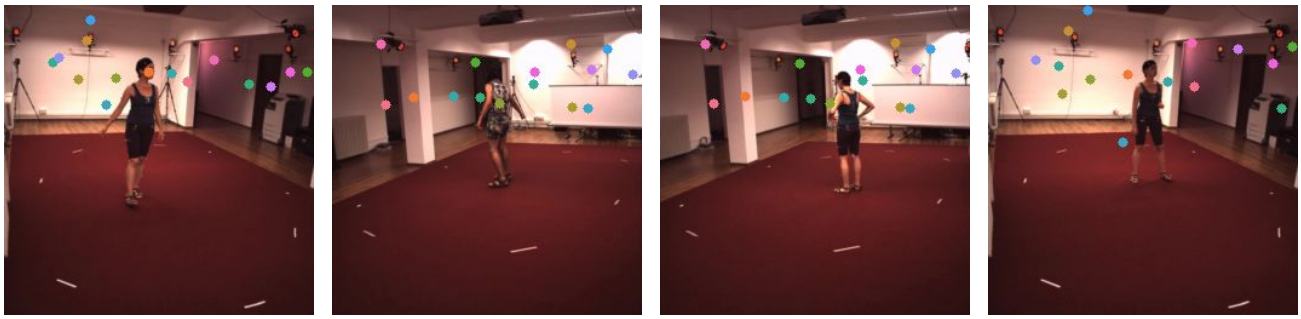

Figure 3. **Qualitative results for Keypoint3D on Human3.6M**. Representative samples of 3D keypoints discovered using Keypoint3D method [2] on real videos.

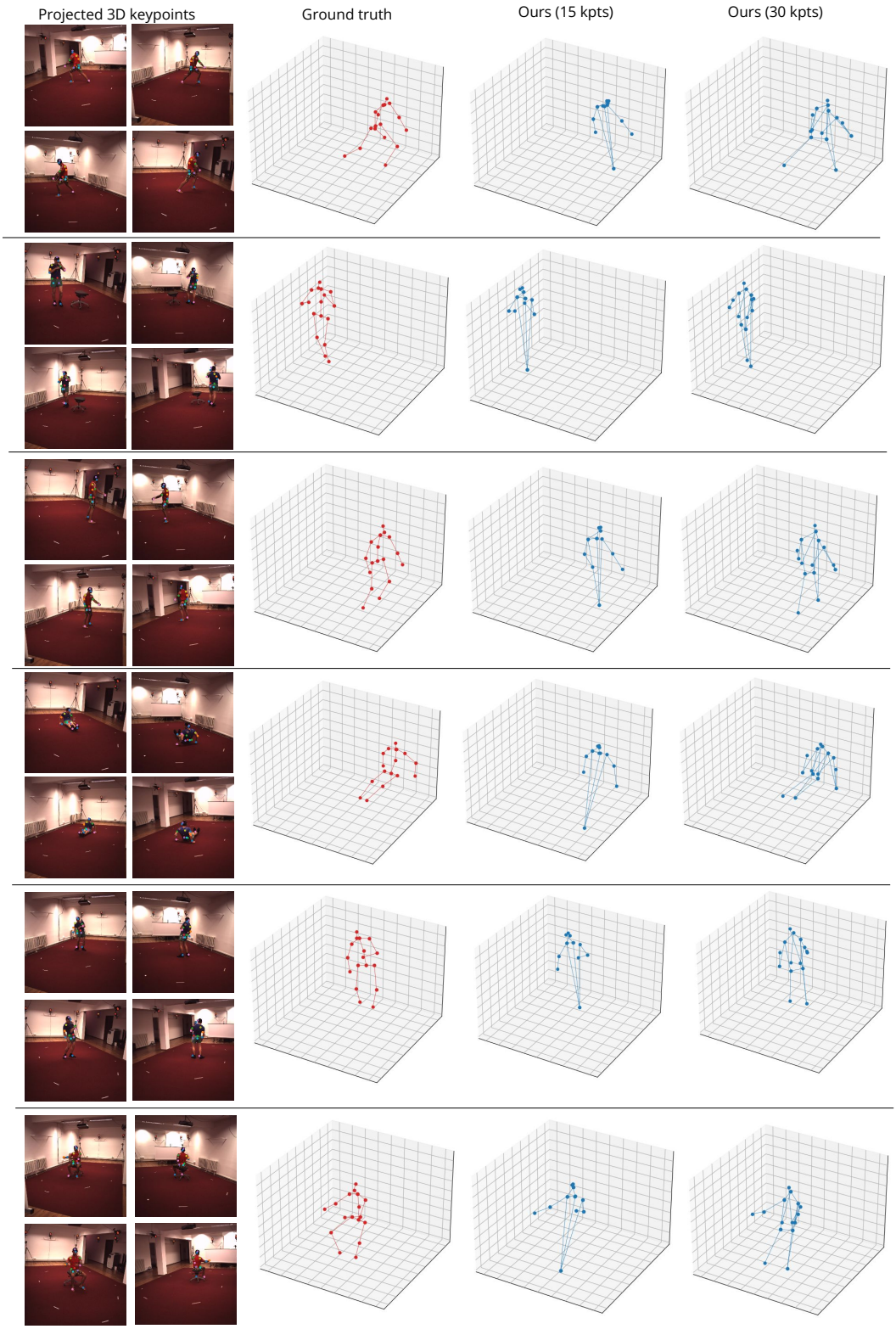

Figure 4. **Qualitative results for 3D keypoint discovery on Human3.6M**. Representative samples from BKinD-3D without regression or alignment for 15 and 30 total discovered keypoints. We visualize all keypoints that are connected using the learned edge weights, and the projected 3D keypoints in the leftmost column are from the keypoint model with 30 discovered keypoints.

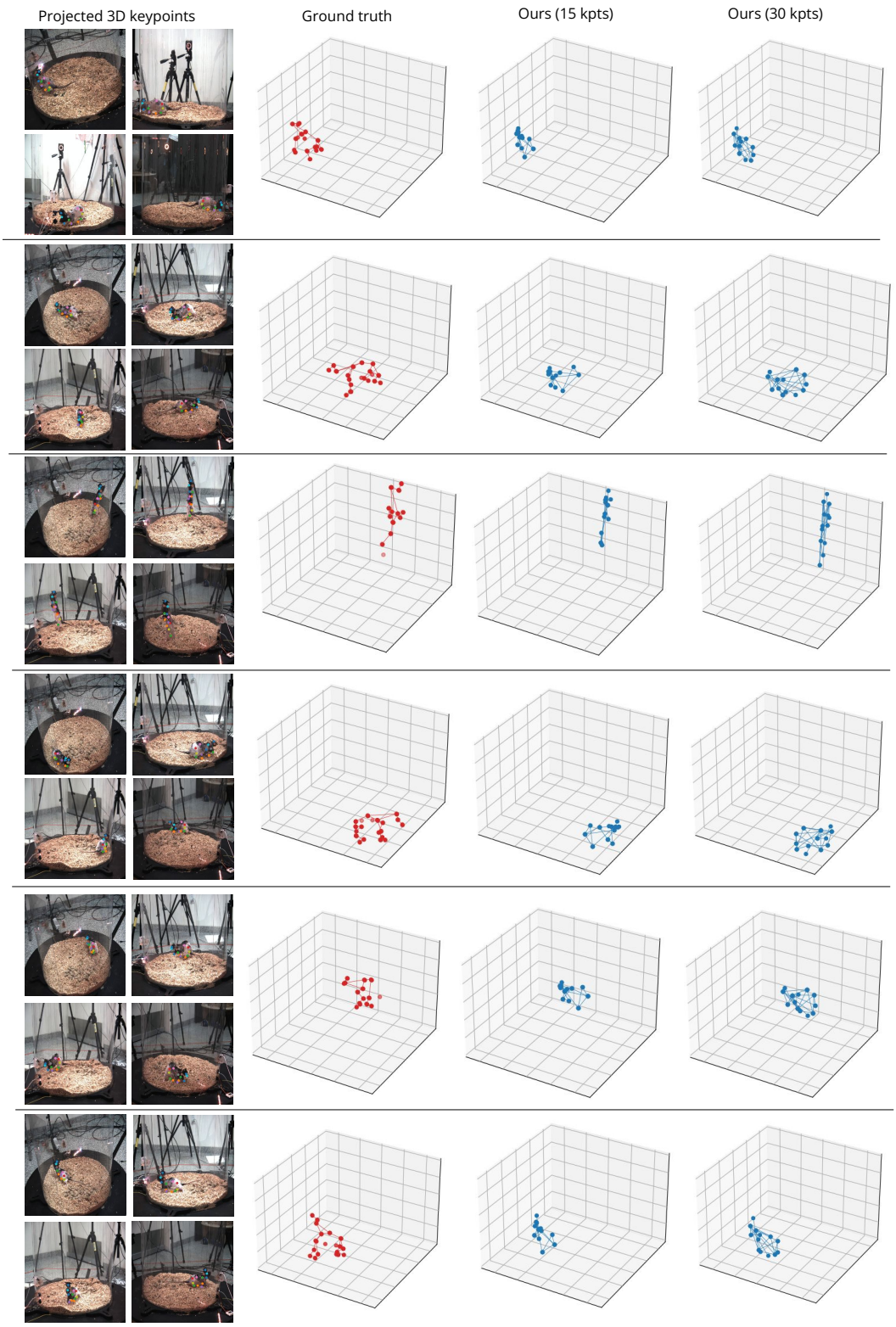

Figure 5. **Qualitative results for 3D keypoint discovery on Rat7M**. Representative samples of 3D keypoints discovered from BKinD-3D without regression or alignment for 15 and 30 total discovered keypoints. We visualize all connected keypoints using the learned edge weights and visualize the first 4 cameras (out of 6 cameras) in Rat7M for projected 3D keypoints from the 30 keypoint model.