Appendix

A. JRDB-Pose Annotation Protocol

We begin our annotation process on individual camera views, initializing with pose estimates from existing pose estimation methods. In the first phase of annotation, we used a customized multi-frame annotation tool to adjust the skeletons and ensure their temporal consistency across video frames. An in-house track editor was used to merge or split tracks, as well as to mark when they enter or exit the scene. In the second phase, we merged the individual camera views into the panoramic views, and used a custom tool to check overlapping and verify accuracy. After each phase, all annotations were checked by separate reviewers. When we merge the individual camera views into a panoramic image, we apply the following protocol to merge annotatinos from adjacent views: if a joint is labeled in two views, we use the label from the joint with a higher visibility score; if the joints have identical visibility scores, we use the pose with higher overall visibility.

B. Annotation Visualization

JRDB-Pose provides high-quality pose annotations at a high (15fps) frequency over all JRDB scenes. In Figure Fig. 8 and Figure Fig. 9 we visualize of our pose annotations for each of the training scenes. The visualizations show the indoor and outdoor scenes, varying light conditions, and a diverse poses representing a wide distribution of actions. To highlight the high frequency and quality of annotations we also refer the reader to https://jrdb.erc.monash.edu/ for video examples of JRDB-Pose.

C. Additional Evaluations

In addition to the evaluation we provide in the main paper, we also include metrics for additional modalities and camera types. In Tab. 7 we report the tracking results for our baseline models and pre-training methods using the panoramic stitched camera images. Similar to the results on individual camera images, OCTrack outperforms the other baselines. We also similarly observe that fine-tuning from COCO yields mixed results with improvements in MOTA and IDF1 but drops in the other metrics as compared to training on JRDB-Pose only.

D. Qualitative Analysis

We show predictions made by Yolo-Pose, our best performing pose estimation method, on frames from each of the testing sequences in Fig. 11 and Fig. 10. The model used for visualization was initialized from scratch. The predictions are accurate for most poses, demonstrating that JRDB-Pose is sufficiently large enough for the model to learn to predict a wide distribution of poses, even without pre-training.

E. Cross-dataset generalizability

The table below shows cross-validation results on JRDB-Pose and PoseTrack21 datasets, where the models are trained on one and evaluated on the other. The performances of the models trained on PoseTrack21 are "moderately low" on JRDB-Pose dataset (around half of the in-domain performances reported in Table 5). The same behaviour is seen on the models trained on JRDB-Pose and subsequently evaluated on PoseTrack21. This reveals a significant domain gap and different challenges between the two datasets.

Trained on	Evaluated on	Method	MOTA	IDF1	IDSW
JRDB-Pose (Individual)	PoseTrack21	**ByteTrack**	23.89	41.69	666
		OC-SORT	**29.28**	**48.92**	**330**
Posetrack21	JRDB	**ByteTrack**	28.07	34.81	4359
		OC-SORT	**35.88**	**41.49**	**3290**

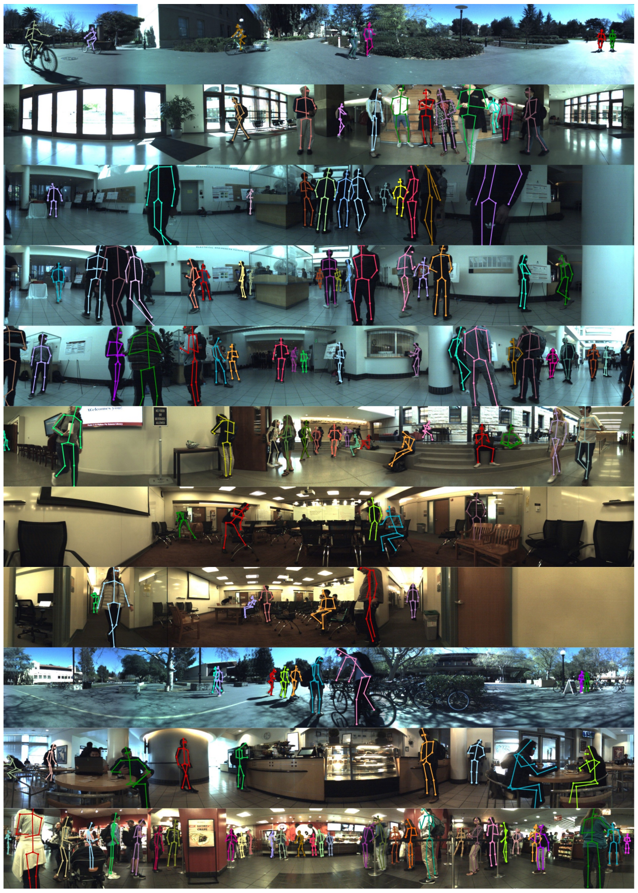

Figure 8. Visualization of JRDB-Pose annotations on the 27 training sequences

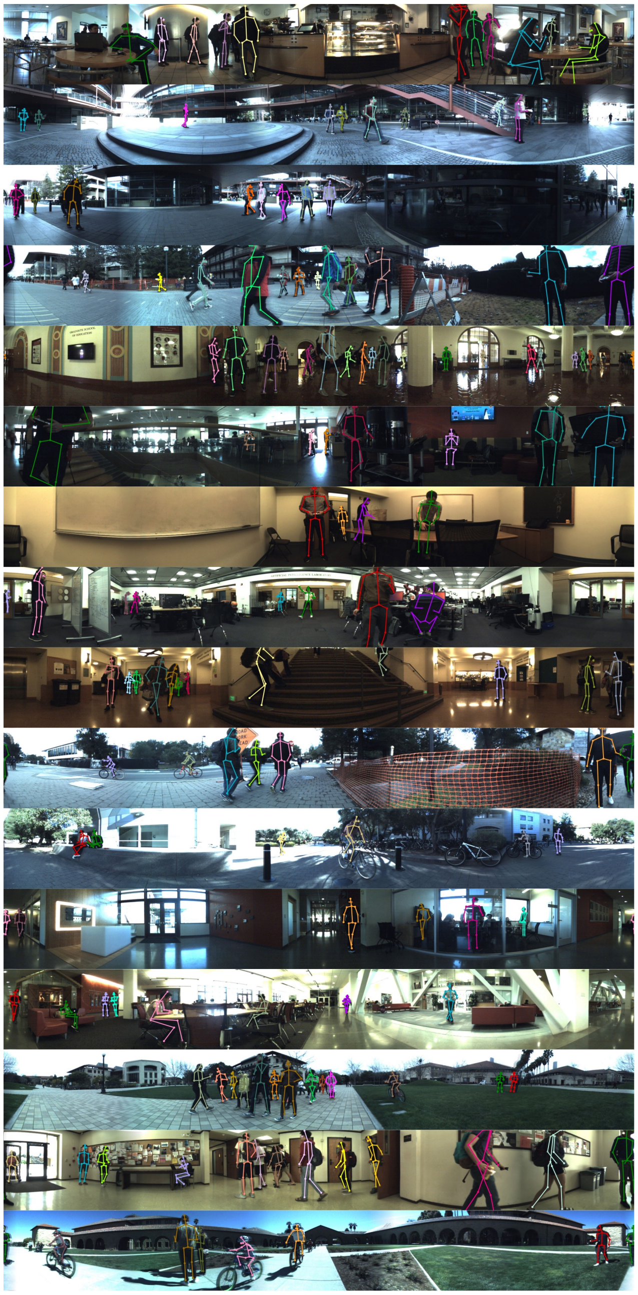

Figure 9. Visualization of JRDB-Pose annotations from each of the 27 training sequences in JRDB-Pose (cont.). The images feature indoor and outdoor areas on a university campus with varying lighting conditions, motion, pedestrian density, and activities. Locations include roads, strip malls, sidewalks, restaurants, plazas, parks, halls, classrooms, laboratories, and office buildings. These scenes show a wide range of scenarios and human poses that a social robot would encounter during operations.

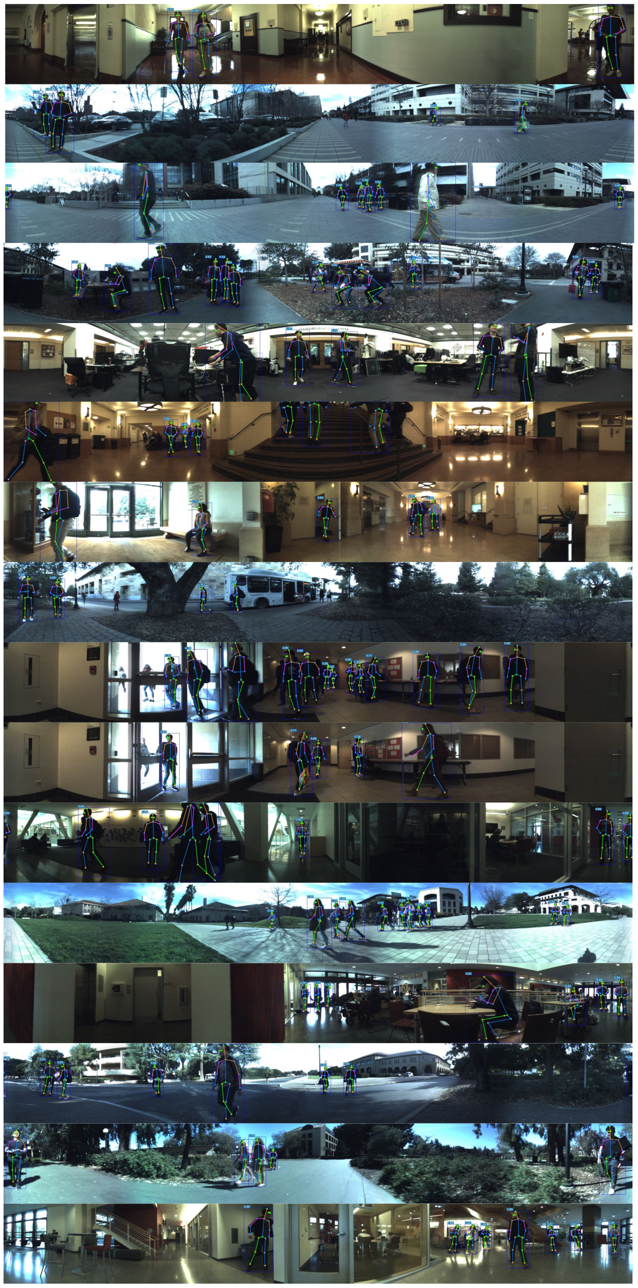

Figure 10. Visualization of Yolo-Pose pose estimation predictions on 27 testing sequences in JRDB-Pose. The weights of the model are initialized from scratch, demonstrating that JRDB-Pose is sufficiently large for the model to learn accurate pose representations even without finetuning.

Pose Estimation Method (Training)	**Tracking Method**	MOTA ↑	IDF1↑	IDSW↓	$\mathcal{O}^2_{pose}$↓	Components		$\mathcal{O}^2_{pose}$↓ by Visibility		
						Card↓	Loc↓	V↓	O↓	I↓
Yolo-Pose [27] (COCO only)	ByteTrack [55]	51.00	43.63	5160	0.914	0.815	0.099	0.910	0.912	0.911
	UniTrack [50]	45.74	41.35	4475	0.940	0.877	**0.064**	0.938	0.939	0.938
	OCTrack [8]	**55.51**	**45.33**	**3906**	**0.895**	**0.766**	0.129	**0.892**	**0.894**	**0.892**
Yolo-Pose [27] (JRDB-Pose only)	ByteTrack [55]	54.33	39.84	4730	0.920	0.796	0.124	0.917	0.920	0.917
	UniTrack [50]	55.90	44.32	4206	0.928	0.849	**0.079**	0.924	0.928	0.925
	OCTrack [8]	**61.28**	**48.08**	**3296**	**0.861**	**0.692**	0.169	**0.856**	**0.862**	**0.855**
Yolo-Pose [27] (COCO→ JRDB-Pose)	ByteTrack [55]	57.69	43.68	4333	0.910	0.791	0.120	0.909	0.911	0.907
	UniTrack [50]	59.37	46.82	3779	0.921	0.841	**0.080**	0.919	0.921	0.918
	OCTrack [8]	**63.02**	**49.04**	**3394**	**0.870**	**0.715**	0.155	**0.867**	**0.871**	**0.865**

Table 7. Multi-person pose tracking baselines evaluated on JRDB-Pose stitched camera images.

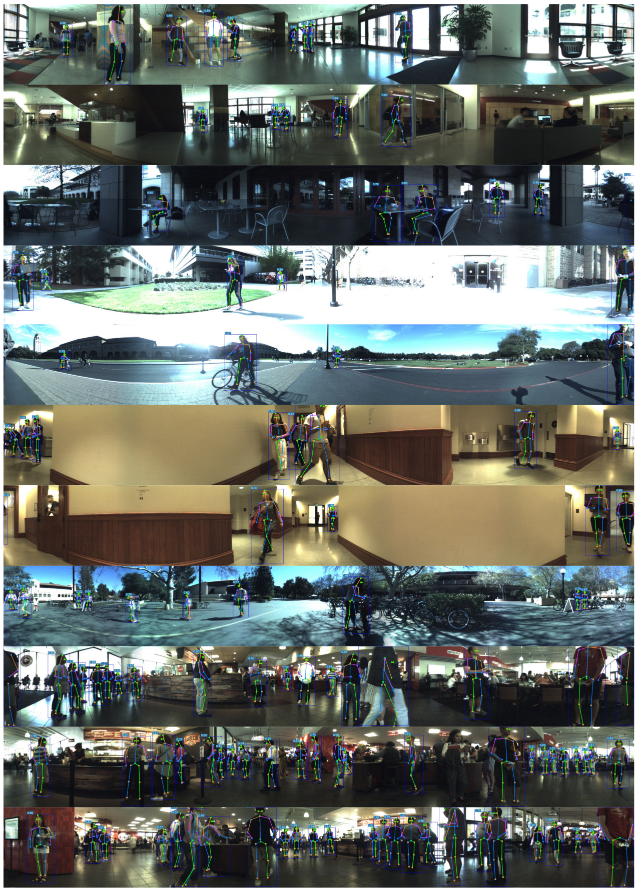

Figure 11. Visualization of Yolo-Pose predictions on 27 testing sequences in JRDB-Pose dataset (cont.)