LiCamPose: Combining Multi-View LiDAR and RGB Cameras for Robust Single-timestamp 3D Human Pose Estimation

Zhiyu Pan Zhicheng Zhong Wenxuan Guo Yifan Chen Jianjiang Feng* Jie Zhou
Department of Automation, BNRist, Tsinghua University, China
{pzy20, zhongzc18, gwx22, chenyf21}@mails.tsinghua.edu.cn
{jfeng, jzhou}@tsinghua.edu.cn

Abstract

Several methods have been proposed to estimate 3D human pose from multi-view images, achieving satisfactory performance on public datasets collected under relatively simple conditions. However, there are limited approaches studying extracting 3D human skeletons from multimodal inputs, such as RGB and point cloud data. To address this gap, we introduce LiCamPose, a pipeline that integrates multi-view RGB and sparse point cloud information to estimate robust 3D human poses via single timestamp. We demonstrate the effectiveness of the volumetric architecture in combining these modalities. Furthermore, to circumvent the need for manually labeled 3D human pose annotations, we develop a synthetic dataset generator for pretraining and design an unsupervised domain adaptation strategy to train a 3D human pose estimator without manual annotations. To validate the generalization capability of our method, LiCamPose is evaluated on four datasets, including two public datasets, one synthetic dataset, and one challenging self-collected dataset named BasketBall, covering diverse scenarios. The results demonstrate that LiCamPose exhibits great generalization performance and significant application potential. The code, generator, and datasets are available at https://github.com/Yu-Yy/LiCamPose.

1. Introduction

Human pose estimation is a fundamental task in computer vision and has been widely applied in various fields, such as human-computer interaction, human activity recognition, sports analytics, augmented reality, etc. Specifically, multi-view image datasets [4, 24, 27, 54] allow more precise 3D human pose estimation compared to single-view ones [1, 26, 36, 62], due to the ability of multiple views to capture 3D information from epipolar geometry. As technology advances, researchers [25, 35, 47, 52, 59] have achieved promising results on current public multi-view images datasets. However, practical scenarios are more challenging than existing public datasets, with diverse human motions, severe occlusions, and large scene.

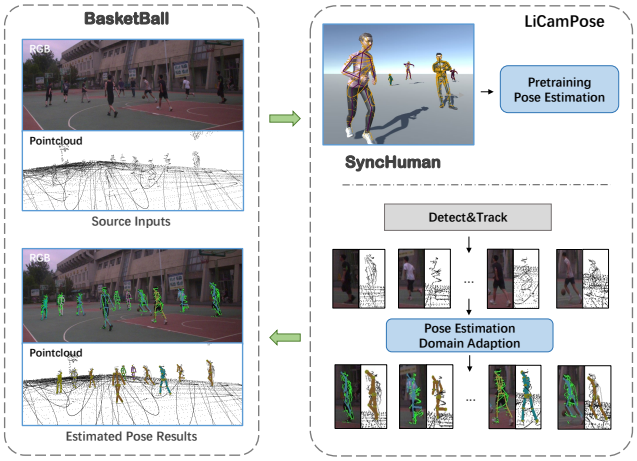

Figure 1. The LiCamPose pipeline for extracting 3D poses, as exemplified by the BasketBall dataset, involves pretraining on synthetic data from SyncHuman, followed by detecting and tracking individuals, and finally using unsupervised domain adaptation to estimate poses.

Fusing LiDAR point clouds with RGB camera information has been demonstrated to significantly enhance object detection and tracking performance [38, 57], leveraging their complementary nature. LiDAR sensor can obtain precise but quite sparse 3D measurements over long range, while camera can capture images of high resolution but lacks depth perception. Zhang et al. [61] introduced a LiDAR-Camera capturing system capable of simultaneously providing texture and depth information over a wide coverage range. This advancement enables the capture of complex human activities, such as basketball games, using a minimal number of sensors, which is advantageous for

*Jianjiang Feng is the corresponding author.
This work was supported in part by the National Natural Science Foundation of China under Grant 62376132 and 62321005.

system setup and cost reduction. However, their application has thus far been limited to human detection and tracking. In recent years, some researchers [8, 14] have aimed to improve human pose estimation performance by temporally fusing these two modalities from a single view. Multi-view information fusion [47,52,58] is the most effective approach to address the inaccuracy of pose estimation caused by occlusion in single-view methods. It is essential to consider integrating the information from multiple modalities and multiple views, and fusing them into a cohesive representation which can significantly facilitate the accuracy of 3D human pose estimation.

In this paper, we introduce LiCamPose (Figure 1), a 3D human pose estimation pipeline that utilizes multi-view single-timestamp LiDAR-Camera inputs. We unify different modalities into a volumetric space that reserve each modality's geometric characteristics which facilitates better realization of single-person 3D pose estimation. Researchers [25, 47, 52, 58] have shown the effectiveness of voxel-based method. Volumetric architecture can naturally model a space's geometric characteristics. Besides, it is straightforward to either map the point cloud information or back-project the 2D information into the 3D volumetric representation. Regarding how to achieve multi-person detection, some pointcloud-based methods [32,38,46] illustrates that they can detect objects accurately with the assist of point cloud. Hence, LiCamPose combines human detection and voxel-based 3D human pose estimation in a top-down manner.

Manually annotating or capturing 3D poses for multiple individuals in large scenes is challenging, time-consuming, and difficult. To achieve better results across diverse scenarios without relying on pose labels, we adopt a two-stage approach: first, pretraining on a synthetic dataset, and then performing unsupervised domain adaptation training on the target dataset. We have developed a synthetic dataset generator named SyncHuman for pretraining. SyncHuman can produce a large volume of synthetic data with multi-modal information, allowing us to adjust LiDAR and camera settings in scenes to meet practical requirements. Moreover, avatar actions are varied and sourced from existing action files or pose annotation files from public motion capture datasets like AMASS [41]. Consequently, the synthetic data includes accurate 3D pose ground truth for groups of people engaging in complex actions.

To bridge the gap between synthetic data and real-world scenarios, we adopt unsupervised domain adaptation training and propose an efficient strategy that includes entropy-guided pseudo 3D pose supervision, pseudo 2D pose supervision, and constraints based on human pose priors. Using volumetric representation, our approach yields 3D human joint heatmaps, with each channel representing a joint's probability distribution. We calculate each channel's entropy to gauge confidence in the corresponding joints, filtering out implausible poses and deriving pseudo 3D pose labels during unsupervised domain adaptation. We utilize off-the-shelf 2D pose estimation methods [7, 13, 51, 56] to generate pseudo 2D pose labels for supervision. Additionally, we introduce an intuitive human prior loss to enforce coherence in the predicted 3D poses. By integrating these methodologies, we develop a 3D human pose estimation algorithm that leverages multi-view point cloud and RGB data without requiring annotations on the target dataset. To further validate LiCamPose, we built a four-view LiDAR-Camera system to capture a basketball game, creating the BasketBall dataset for qualitative evaluation of our method.

We summarize the contributions of this paper as follows:

- We propose LiCamPose, a simple and effective pipeline for fusing multi-view, sparse point cloud and RGB information to estimate 3D poses of multiple individuals from single timestamp.
- We developed SyncHuman, a generator for synthetic data under various camera and LiDAR settings. Additionally, we created a four-view LiDAR-Camera system to capture real data from a basketball game, resulting in the BasketBall dataset.
- We propose a training strategy that avoids manual annotations by pretraining on synthetic data from SyncHuman and using unsupervised domain adaptation on the target dataset.

2. Related Works

In this section, we conduct the literature review according to the three contribution points we proposed.

2.1. 3D Human Pose Estimation

Image-based. The basic 3D pose estimation method typically follows a two-stage process: first estimating the 2D pose and then lifting it into 3D space [10,25,35,47,52,55]. For multi-person settings, some methods [10, 35] match pedestrians from different views and locate them through 2D pose similarity. However, these methods are not robust to inaccurate 2D pose results. Zhang et al. [59] directly use 2D images as inputs and regress the 3D pose with a transformer architecture, but their training process is time-consuming. Voxel-based methods [47, 52] locate each person in a 3D volumetric space and estimate 3D poses, significantly improving the precision of 3D human pose estimation. However, these voxel-based methods are not suitable for large scenes due to their high computational cost when detecting people. LiCamPose utilizes pointcloud-based method to locate and track pedestrians, and then employs voxel-based pose estimation for each individual.

Table 1. Comparison of synthetic datasets related to human.

Synthetic Dataset	Scene setup					Labels	
	Multi-View	Multi-Person	RGB Image	Depth Image	LiDAR Pointcloud	2D Pose	3D Pose
CAPE [40]	✗	✗	✗	✗	✗	✗	✓
SURREAL [53]	✗	✗	✓	✓	✗	✓	✓
PSP [12]	✗	✗	✓	✗	✗	✓	✗
AGORA [44]	✗	✓	✓	✗	✗	✓	✓
BEDLAM [6]	✗	✓	✓	✗	✗	✗	✓
BlendMimic3D [37]	✓	✓	✓	✗	✗	✓	✓
CALAR [11]	✓	✓	✓	✓	✓	✗	✗
Ours	✓	✓	✓	✓	✓	✓	✓

Pointcloud-based. Initially, several 3D pose estimation methods were based on single-view depth maps [15, 20, 22, 42], treating the depth map as 2D information with depth values. Conversely, some methods back-project the depth map into 3D space as a dense 3D point cloud and utilize PointNet networks [3, 64]. Moon et al. [43] proposed a single-person pose estimation approach that treats the depth map as a point cloud and fills it into a volumetric space. Bekhtaoui et al. [3] employed a PointNet-based approach to detect and estimate 3D human pose. More recently, Li et al. [33] used sparse LiDAR-scanned point clouds to estimate the 3D pose of a single person. Zhang et al. [60] enhanced performance by incorporating the point cloud surrounding the person as a neighborhood context. However, temporal information is essential for the effectiveness of both approaches. Sparse point clouds from single timestamp do not provide sufficient information for accurate 3D pose estimation. Multi-modal fusion is beneficial for precise 3D human pose perception.

Multi-modal based. Several works have been introduced for RGBD human pose estimation [2, 17, 21, 65], demonstrating that multi-modal information not only aids in detecting individuals but also ensures the accuracy of 3D human pose estimation. And recently, researchers [8, 14] tried to fuse the RGB and point cloud temporal information from singe view to improve the pose estimation accuracy. However, they rely on temporal information for supplementation, and when a single view is occluded for an extended period, the methods fail to make accurate predictions. Regarding point cloud and RGB information fusion methods, some approaches [3, 21, 65] fuse these modalities at the point level by attaching extracted 2D features to each 3D point. A few methods [34, 45] employ a feature-level fusion strategy. Nevertheless, these methods do not integrate the different modalities in a feature space that preserves each modality's spatial properties. In contrast, LiCamPose employs a volumetric representation to integrate point cloud and RGB data from multiple views within a single timestamp, preserving the spatial properties of the real-world environment and mitigating the impact of occlusion.

2.2. Synthetic Dataset Generation

Manually annotating 3D human poses is extremely challenging. Therefore, synthetic datasets are beneficial and useful for pretraining models. Several synthetic datasets for 3D human pose estimation have been developed [6, 12, 37, 40, 44, 53]. However, these datasets typically feature an RGB photo from one view with random backgrounds and do not consider the realistic interaction between avatars and their environment. Additionally, they all lack the provision of point cloud information. Dosovitskiy et al. [11] provide a large-scale synthetic dataset for autonomous driving, but it lacks 3D human pose labels, and the actions of the characters are monotonous.

We introduce our synthetic dataset generator, SyncHuman, which can produce data with greater richness and diversity (Table 1). Furthermore, we will release the generation tool, allowing researchers to adjust settings according to their needs via our provided APIs

2.3. Unsupervised Domain Adaption Training

Numerous works have focused on unsupervised or domain adaptation methods for single-view 3D human pose estimation. Kocabas et al. [29] use multi-view geometry to supervise single-view predictions. Several methods [30, 49] employ a teacher-student framework for domain adaptation. Some approaches [9, 63] utilize optical flow or depth as inputs, which are less affected by domain shifts compared to RGB. Bigalke et al. [5] incorporate human prior loss based on human anatomy, while Kundu et al. [31] define the uncertainty of predictions and control this uncertainty during training. For multi-view 3D human pose generation, some works use non-deeplearning methods [4, 48]. 3DPS [4] and basic triangulation achieve 3D poses with significant computational complexity or inaccuracy. Remelli et al. [48] proposed an efficient direct linear transformation (DLT) method that quickly produces relatively accurate 3D results.

Our approach utilizes multi-view 2D pose heatmaps derived from RGB data alongside point clouds, ensuring minimal impact from domain variations. We utilize information entropy to help select reliable results as pseudo 3D labels. Additionally, we incorporate a human prior loss to ensure the plausibility of 3D poses.

3. Methodology

We introduce LiCamPose from two perspectives: 3D human pose estimation and unsupervised domain adaptation (Figure 2). Section 3.1 details the top-down approach for 3D human pose estimation. Section 3.2 explains how we generate the synthetic dataset to support the training of the pose estimation model using point cloud and RGB data, and how we utilize information entropy and human prior loss to achieve domain adaptation.

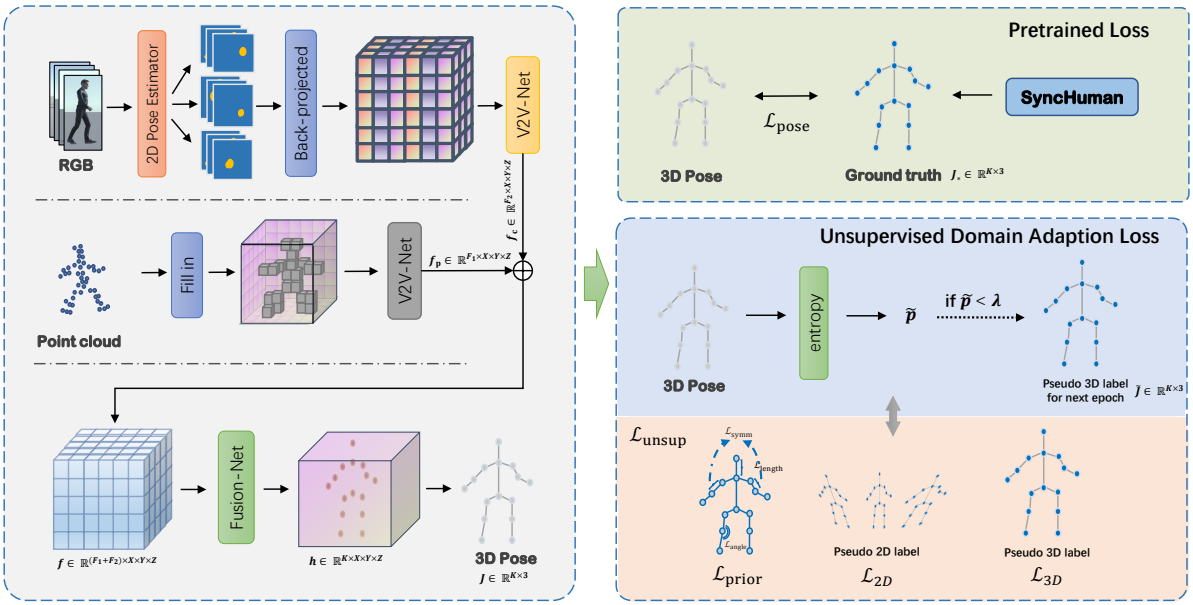

Figure 2. The detailed structure of LiCamPose in 3D human pose estimation and its corresponding losses calculations.

3.1. 3D Human Pose Estimation

Similar to [10, 47, 52], we adopt a top-down approach to estimate 3D poses. Specifically, we use PointPillars [32] to detect the 3D bounding box of each person. Once we obtain the 3D bounding box, we derive the 2D bounding box in each view for 2D pose estimation and extract the corresponding point cloud. Initially, we define a volumetric space centered at the point cloud's centroid, with a size consistent with the detected bounding boxes, and discretize it into an $X \times Y \times Z$ resolution. We fill each voxel based on the coordinates of the point cloud, assigning a value of 1 to voxels containing points from the point cloud. This allows us to obtain the pointcloud-related feature map $f_{\mathrm{p}} \in \mathbb{R}^{F_1 \times X \times Y \times Z}$ using the 3D convolutional backbone V2V-Net [43].

With the development of 2D pose estimation methods [13, 56], robust 2D pose results can be predicted using off-the-shelf techniques. We use these methods to obtain 2D pose heatmaps from RGB information of each view. These multi-view heatmaps are then back-projected into the volumetric space according to each camera's settings, and V2V-Net is used to extract the RGB-related features $f_{\mathrm{c}} \in \mathbb{R}^{F_2 \times X \times Y \times Z}$. We concatenate the features from both modalities in the same 3D feature space to get fusion feature f and obtain the final 3D human pose heatmap h via Fusion-Net, which has the same structure as V2V-Net. To mitigate quantization error, we adopt *Soft-argmax* to calculate each joint's 3D coordinate J^k from the 3D heatmaps and minimize the L_1 loss with the ground truth J_*^k:

$$\mathcal{L}_{\text{pose}} = \sum_{k=1}^{K} \left\| J^k - J_*^k \right\|_1, \tag{1}$$

where K is the number of joints and J^k is the predicted joint, $J^k, J_*^k \in \mathbb{R}^3$.

3.2. Unsupervised Domain Adaptation

Due to the lack of annotated multi-view LiDAR-Camera 3D human pose datasets, we opted to generate a synthetic dataset to assist with training. Our designed SyncHuman can create 3D dynamic scenes with multiple persons and accurate labels. Additionally, we developed an efficient unsupervised domain adaptation method by designing a loss function to transfer the pretrained model from the synthetic dataset to a real-world dataset.

3.2.1 SyncHuman Generator

We developed the synthetic system based on the Unity Engine. Its flexibility and productivity allow us to create scenes with specific sizes and backgrounds according to our requirements. For sensors, we use Unity's built-in camera to obtain RGB images and extract depth information from the GPU's depth buffer using custom shaders in the rendering pipeline. Additionally, we can generate colored point clouds by sampling the depth and RGB images. To simulate the LiDAR scanning process, we sample points according to a time-based scanning function. We currently focus on the Livox Mid-40 LiDAR due to its affordability for sports or surveillance applications. We use the following function to simulate its scanning process:

$$r = \alpha \times \cos(3.825 \times (\theta_0 + 0.0017 \times n)), \tag{2}$$

where $n \in [0, t \times 1e5]$ and t is in second. α is the scanning radius in pixel, and θ_0 is a random initial angle. This equation is defined in polar coordinate. Final sampling points

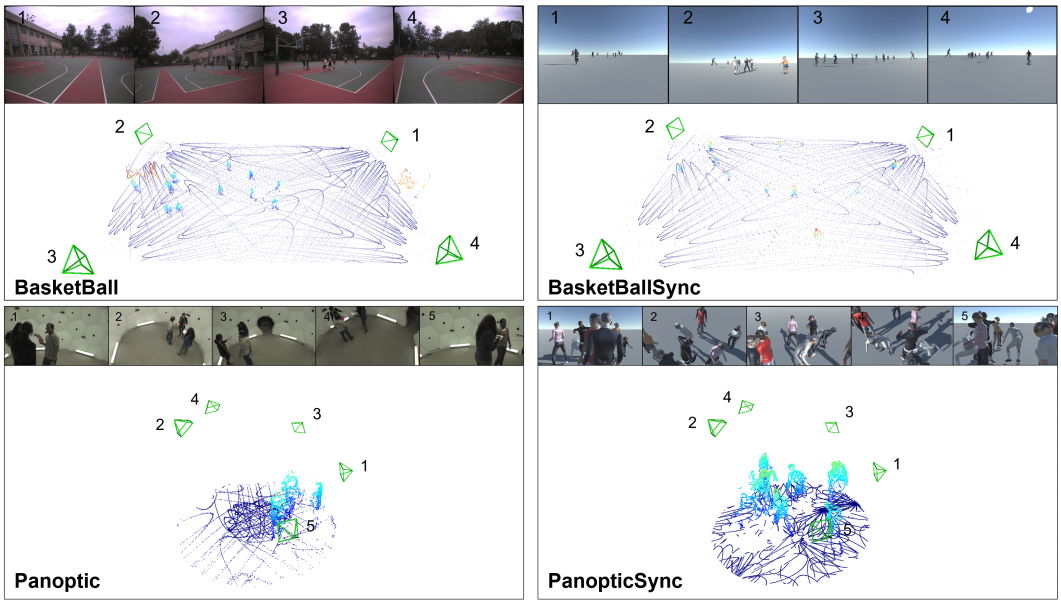

Figure 3. Real datasets and the corresponding synthetic datasets generated by SyncHuman.

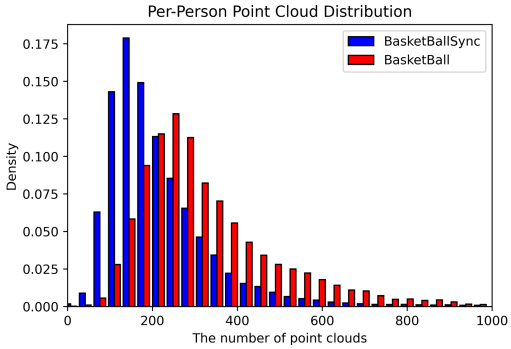

Figure 4. Statistic analysis of per-person point cloud counts between BaseketBall and BaseketBallSync.

can be acquired by transforming it from polar coordinates to Cartesian coordinates and translating it to the center of the image space.

For the avatars, we download various human 3D models from Adobe Mixamo[1]. To ensure the diversity of the generated actions, these avatars can be driven by either premade action files or keypoint annotations from other public datasets such as AMASS [41]. We have developed driving APIs for COCO17, COCO19 [36], and SMPL [39] standard keypoint annotated inputs. Regarding ground truth data, we can obtain the 3D human pose from the avatars' humanoid skeleton; the 2D pose label can be acquired by projecting the 3D pose into 2D views. Additionally, we can fetch the mesh vertices of each avatar and compute the semantic segmentation label for each point cloud by considering the pose label simultaneously.

[1]https://mixamo.com

We can use SyncHuman to simulate any arrangement of sensors to observe a scene. As demonstrated in the experiments in our paper, using the same scene setting for both training and testing yields better transfer performance. Figure 3 compares the datasets we generated, PanopticSync and BasketBallSync, with the original datasets. Since SyncHuman allows for flexible sensor placement, the distribution of the number of point clouds scanned per person in the simulation closely approximates the real-world distribution (Figure 4).

3.2.2 Unsupervised Domain Adaptation

Similar to the 2D projection supervision used by other unsupervised or weakly supervised methods [18,29], we directly utilize an off-the-shelf 2D human pose estimation method to obtain the pseudo 2D pose label $\widetilde{J}$2D. The 2D loss $\mathcal{L}$2D is computed by calculating the L_2 norm between the 3D pose projection results and the pseudo 2D label from each view:

$$\mathcal{L}_{\text{2D}} = \sum_{v=1}^{V} \sum_{k=1}^{K} \left\| \mathcal{P}_v(J^{v,k}) - \widetilde{J}_{\text{2D}}^{v,k} \right\|_2, \tag{3}$$

where $\mathcal{P}$ is the projection function. V is the number of views. To obtain the pseudo 3D label, we use information entropy as an uncertainty index. The entropy of a keypoint prediction heatmap (a spatial probability distribution) h^k is defined as:

$$\mathcal{H}(h^k) = -\sum_i h_i^k \times \log h_i^k, \tag{4}$$

where i represents the voxel index. Higher entropy indicates greater uncertainty in the keypoint location. Moreover, as

observed in our experiments (Section 4.5), there is a correlation between entropy and the rationality of the keypoint location.

To measure a person's uncertainty, we use the maximum entropy value of all keypoints. Unlike Kundu et al. [31], who train a model to learn the uncertainty values, we directly calculate and use entropy as an inherent index. Our experiments demonstrate that the magnitude of entropy values can serve as an indicator of pose estimation quality for a specific network. Consequently, an entropy threshold λ can be set to filter out the reasonably predicted 3D human poses. Specifically, we select the predicted pose $\widetilde{p}$ whose entropy value is less than λ as the pseudo 3D pose label for the next epoch's training, allowing it to be updated after each training epoch. Therefore, the 3D pseudo pose loss can be obtained by:

$$\mathcal{L}_{\text{3D}} = \sum_{k=1}^{K} \left\| J^k - \widetilde{J}^k \right\|_1, \tag{5}$$

where J^k represent the predicted joints and $\widetilde{J}^k$ is the 3D joint of the selected pseudo 3D pose label $\widetilde{p}$.

To further ensure the anatomical plausibility of the pose, we design a human prior loss $\mathcal{L}_{\text{prior}}$[2]. Specifically, we formulate three losses to penalize asymmetric limb lengths ($\mathcal{L}_{\text{symm}}$), implausible joint angles ($\mathcal{L}_{\text{angle}}$), and implausible bone lengths ($\mathcal{L}_{\text{length}}$).

In summary, the final loss function is defined as:

$$\mathcal{L}_{\text{unsup}} = \omega_1 \mathcal{L}_{\text{2D}} + \omega_2 \mathbf{1}(\widetilde{p} < \lambda) \mathcal{L}_{\text{3D}} + \omega_3 \mathcal{L}_{\text{prior}}, \tag{6}$$

where ω_1, ω_2 and ω_3 are the weights of each loss, and $\mathbf{1}(\cdot)$ represents the indicator function.

4. Experiments

In this section, we introduce all datasets and evaluation metrics used in our experiments. We analyze the performance of our LiCamPose pipeline in supervised and unsupervised manner on different datasets. Additionally, we examine the impact of different configurations on the results in the context of unsupervised domain adaptation.

4.1. Implementation details.

We use V2V-Net [43] as the feature extractor for point cloud and RGB information. The detailed design of V2V-Net follows the PRN model [52]. The resolution of the voxel grid is set to $64 \times 64 \times 64$ within a 2m $\times$ 2m $\times$ 2m space. The weights are set as $\omega_1 = 0.02$, $\omega_2 = 1$, $\omega_3 = 10$, and $\lambda = 6$. We train the voxel-based pose estimation network on an NVIDIA GeForce RTX-3090 with a batch size of eight. The learning rate is set to 0.001, and the optimizer used is Adam [28].

[2]Please refer to the Supplementary Material for details.

4.2. Datasets and Metrics.

We use the following datasets in our experiments:

I. CMU Panoptic Studio [27]. The setup is indoors with a valid scene range of approximately 5m $\times$ 5m. For multi-modal inputs, we select subsets[3] from [35] that include depth information. We unify the depth data from Kinect 1 to 5 and process these depth maps using equation 2 to obtain a sparse point cloud, similar to Mid-40 Livox LiDAR scanning. For 2D pose estimation, we use the results from [35], predicted with HRNet [51].

II. MVOR [50]. The dataset was collected indoors in an operating room. We use three sampling functions (equation 2) centered at three trisection points of the image width to simulate Mid-100 Livox LiDAR scanning. For 2D pose estimation, we use results from [50], predicted using AlphaPose [13]. Due to a semantic keypoints' definition gap between MVOR (ten keypoints) and our COCO17 standard, we only perform qualitative analysis on MVOR. Data from "day1, day2, day3" are used for training, and "day4" for testing.

III. BasketBall. The dataset was collected outdoors, covering a valid scene range of approximately 35m $\times$ 17m, recording a basketball match. We gathered this real-world dataset with four-view RGB-poincloud information. The point clouds were collected using four Mid-100 Livox LiDARs, and the 2D pose estimation results were predicted by VitPose [56]. The dataset comprises two segments, each containing two thousand frames. It includes manually marked detection and tracking ground truth for each frame but lacks 3D keypoints ground truth. Therefore, it is primarily used for qualitative analysis.

IV. BasketBallSync. This synthetic dataset was generated using SyncHuman with the same sensor configuration and settings as BasketBall dataset. The point cloud is obtained by simulating the scan pattern of Mid-100 Livox LiDARs. It contains ten avatars downloaded from Mixamo[4] performing random actions, with the first eight used for training and the remaining two for testing, across all 3336 frames at 10Hz. We utilize the pose data from the "CMU" track of AMASS [41] to animate the avatars. The 2D human poses are predicted using VitPose [56].

Evaluation metrics. To evaluate the 3D human pose, we calculate the mean per-joint position error (MPJPE) and the error after Procrustes Alignment (PA-MPJPE), both in millimeters [19]. For fairness, we compare our results to methods using single-timestamp data inputs similar to ours.

[3]"160422_ultimatum1", "160224_haggling1", "160226_haggling1", "161202_haggling1", "160906_ian1", "160906_ian2", "160906_ian3", and "160906_band1" for training; "160906_pizza1", "160422_haggling1", "160906_ian5", "160906_band2" for testing.

[4]Character 2, 6, 11, 22, 23, 31, 37, 38, 42 and a kid model named "Gregory".

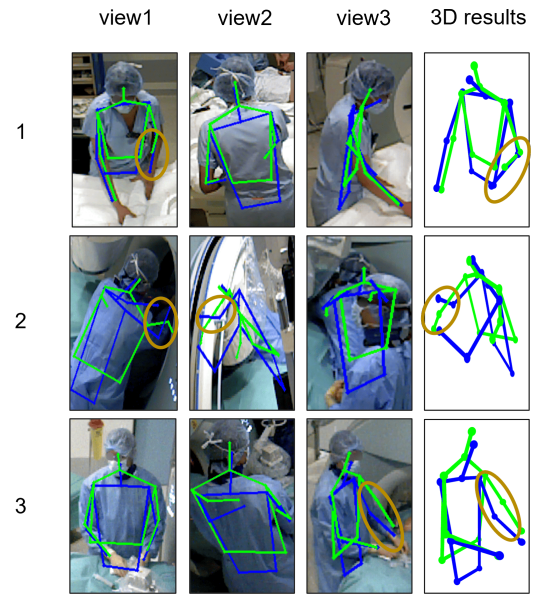

Figure 5. Three examples of 3D human pose estimation on MVOR. Blue lines represent predictions, green lines represent ground truth. The first three columns show 2D projections from different views, and the fourth column shows the 3D pose results.

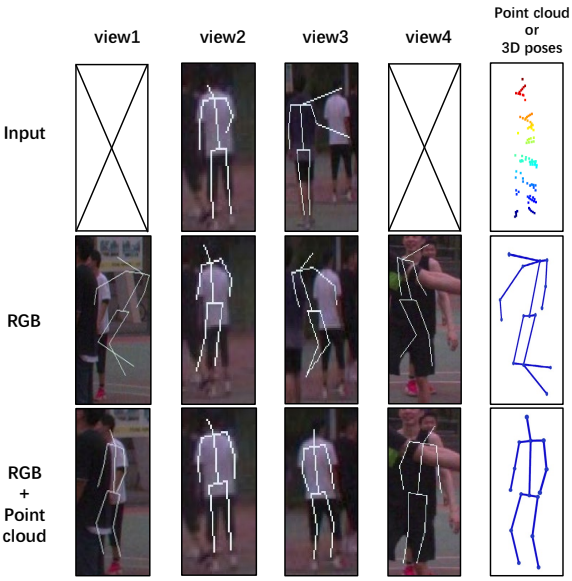

Figure 6. Qualitative illustration on the BasketBall dataset from different input modalities. The first row shows 2D pose estimations (missing where not estimable) and point clouds. The second row displays results from using only RGB input, with 2D poses projected from the estimated 3D poses. The third row presents results from using both RGB and point cloud inputs.

4.3. 3D Pose Estimation Analysis

In this section, we evaluate the 3D human pose estimation results using different approaches. For PlaneSweepPose [35], VoxelPose [52], and supervised LiCamPose, we incorporate ground truth person locations during both training and testing phases. However, for MvP [59], its architecture does not accommodate direct integration of ground truth detection information, making it unable to utilize depth or point cloud data directly. For fairness, we evaluate only the MPJPE@500 metric (considering only per-joint errors less than 500 mm) for the matched person in the case of MvP.

We evaluate these approaches on Panoptic Studio and BasketBallSync datasets, representing small and large scenes respectively. Table 2 demonstrates that LiCamPose surpasses them in terms of both MPJPE and PA-MPJPE, particularly in the large scene setting. Notably, unsupervised learning LiCamPose even outperforms some supervised multi-view approaches with RGB inputs.

We evaluate LiCamPose using unsupervised domain adaption on MVOR, and it achieves impressive performance, occasionally outperforming ground truth in some frames (Figure 5).

Additionally, we analyze the performance using different modal inputs (point cloud, RGB, and both) in volumetric representation. Table 3 illustrates that using Livox point cloud information alone struggles to accurately extract the 3D human skeleton due to its sparsity, making it unsuitable for unsupervised learning without 2D pseudo-labels from heatmaps. However, combining it with RGB information significantly enhances performance. Figure 6 demonstrates that multi-modal inputs improve the model's robustness to 2D pose estimation errors, even in challenging scenarios such as mispredictions or incorrect person predictions, during unsupervised domain adaptation training.

Table 2. Comparison of different 3D pose estimation methods on Panoptic and BasketBallSync.

Methods	Training Manner	Panoptic		BasketBallSync	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
MvP [59]	supervised	25.78	25.02	291.11	240.32
PlanePose [35]	supervised	18.54	11.92	119.55	65.80
VoxelPose(PRN) [52]	supervised	14.91	11.88	40.80	34.34
LiCamPose	supervised	**14.44**	**11.61**	**31.84**	**27.36**
DLT [48]	unsupervised	38.26	44.77	101.49	62.77
LiCamPose	unsupervised†	**22.00**	**15.96**	**72.92**	**62.72**

† Pretrained on BasketBallSync during Panoptic evaluating and on Panoptic during BasketBallSync testing. It is the same for Table 3 and 5.

Table 3. Comparison of Different Input Modalities for LiCamPose on Panoptic and BasketBallSync.

Input Modality	Training Manenr	Panoptic		BasketBallSync	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
point cloud	supervised	164.30	148.78	136.01	123.22
RGB	supervised	14.91	11.88	40.80	34.34
RGB + point cloud	supervised	**14.44**	**11.61**	**31.84**	**27.36**
RGB	unsupervised	25.72	17.24	263.05	183.85
RGB + point cloud	unsupervised	**22.00**	**15.96**	**72.92**	**62.72**

4.4. Unsupervised Domain Adaption

In this section, we conduct a detailed analysis of synthetic dataset settings and ablation studies on the unsuper-

Table 4. Comparison of different scene setups. We simulated four different scene setups using the SyncHuman generator. It shows the performance of LiCamPose when pre-trained on different scenes and directly applied to Panoptic Studio.

Scene's Size[1]	**Sensors' Setting**[2]			**Metric**	
	Number[3]	Position	Orientation	MPJPE	PA-MPJPE
BaseketBall	4	BaseketBall	BaseketBall	82.61	93.41
BaseketBall	5	others	Panoptic	77.90	60.42
Panoptic	5	others	Panoptic	26.45	22.56
Panoptic	5	Panoptic	Panoptic	**23.15**	**17.63**

[1] Scene sizes are denoted with the names of the corresponding datasets.
[2] Sensor settings are denoted with the names of the corresponding datasets. "Others" refers to configurations with different settings.
[3] BasketBall has 4 groups of sensors, and Panoptic Studio has 5 groups of sensors.

Table 5. Comparison of different unsupervised training strategies.

Unsupervised loss		**Panoptic**		**BasketBallSync**	
$\mathcal{L}_{3D}$	$\mathcal{L}_{prior}$	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
✗	✗	29.14	16.48	84.49	75.55
✓	✗	28.26	25.85	77.38	68.84
✗	✓	22.88	24.69	76.45	65.54
✓	✓	**22.00**	**15.96**	**72.92**	**62.72**

vised training Losses.

4.4.1 Synthetic Setting

Tu et al. [52] conducted an experiment training VoxelPose with a dataset's camera configuration and human poses from other sources, then evaluating the model directly on another dataset. In this section, we further analyze the impact of scene settings in volumetric architecture. We simulated four different scene setups using the SyncHuman generator, varying sensor positions and orientations to modify camera parameters and scene sizes. Table 4 demonstrates that the closer the scene setups resemble each other, the better the model performs. This suggests that in practical applications, aligning synthetic data with the sensor arrangement and scene range of the target dataset can enhance the performance of unsupervised domain adaptation or direct inference in new scenarios.

4.4.2 Ablation Study on Unsupervised Training Losses

For unsupervised domain adaptation, pseudo 2D pose supervision serves as the baseline necessity. Additionally, we incorporate an interpretable human prior loss and a pseudo 3D pose loss (selected based on entropy values) to aid in learning. Thus, we conduct an ablation study to analyze the impact of these two losses. According to Table 5, the entropy-selected pseudo 3D pose loss improves performance, and the prior loss ensures predicted poses remain within a reasonable action range. Therefore, we adopt both losses as part of our effective training strategy.

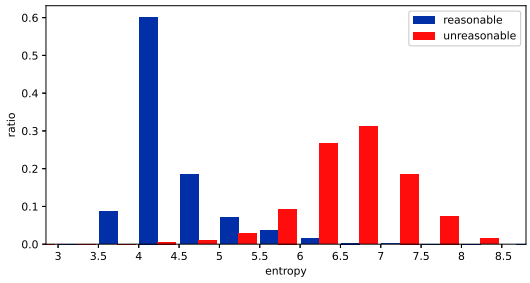

Figure 7. Entropy distributions of reasonable and unreasonable predicted 3D poses.

4.5. Entropy Analysis

To explore the relationship between entropy values and pose rationality, we categorized some 3D poses predicted by LiCamPose as either reasonable or unreasonable manually. Figure 7 illustrates the distribution of entropy values for these categories. Natural division of poses occurs based on an entropy threshold. Therefore, the average entropy value can serve as an indicator of a model's performance on an unlabeled dataset.

5. Conclusion

In this paper, we propose LiCamPose, a pipeline for 3D human pose estimation using multi-view RGB and sparse point cloud data. To facilitate unsupervised training, we introduce SyncHuman, a synthetic data generator, and develop specific unsupervised training losses for domain adaptation. Through extensive experiments on synthetic and real-world datasets, we demonstrate the robust learning capabilities of our pipeline for 3D human pose estimation. Additionally, we investigate the influence of scene settings and unsupervised training losses, and explore the relationship between entropy values and pose plausibility. In future work, we aim to expand SyncHuman's scenarios and integrate temporal information to further enhance estimation accuracy.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 1

[2] Renat Bashirov, Anastasia Ianina, Karim Iskakov, Yevgeniy Kononenko, Valeriya Strizhkova, Victor Lempitsky, and Alexander Vakhitov. Real-time rgbd-based extended body pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2807–2816, 2021. 3

[3] Walid Bekhtaoui, Ruhan Sa, Brian Teixeira, Vivek Singh, Klaus Kirchberg, Yao-jen Chang, and Ankur Kapoor. View invariant human body detection and pose estimation from multiple depth sensors. *arXiv preprint arXiv:2005.04258*, 2020. 3

[4] Vasileios Belagiannis, Sikandar Amin, Mykhaylo Andriluka, Bernt Schiele, Nassir Navab, and Slobodan Ilic. 3D pictorial structures for multiple human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1669–1676, 2014. 1, 3

[5] Alexander Bigalke, Lasse Hansen, Jasper Diesel, and Mattias P Heinrich. Domain adaptation through anatomical constraints for 3D human pose estimation under the cover. In *International Conference on Medical Imaging with Deep Learning*, pages 173–187, 2022. 3

[6] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 3

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299, 2017. 2

[8] Peishan Cong, Yiteng Xu, Yiming Ren, Juze Zhang, Lan Xu, Jingya Wang, Jingyi Yu, and Yuexin Ma. Weakly supervised 3d multi-person pose estimation for large-scale scenes based on monocular camera and single lidar. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 461–469, 2023. 2, 3

[9] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3D human pose estimation: motion to the rescue. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[10] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3D pose estimation from multiple views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7792–7801, 2019. 2, 4

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16, 2017. 3

[12] Salehe Erfanian Ebadi, You-Cyuan Jhang, Alex Zook, Saurav Dhakad, Adam Crespi, Pete Parisi, Steven Borkman, Jonathan Hogins, and Sujoy Ganguly. PeopleSansPeople: a synthetic data generator for human-centric computer vision. *arXiv preprint arXiv:2112.09290*, 2021. 3

[13] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 4, 6

[14] Michael Fürst, Shriya TP Gupta, René Schuster, Oliver Wasenmüller, and Didier Stricker. Hperl: 3d human pose estimation from rgb and lidar. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7321–7327. IEEE, 2021. 2, 3

[15] Nicola Garau, Niccolo Bisagno, Piotr Bródka, and Nicola Conci. DECA: Deep viewpoint-equivariant human pose estimation using capsule autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11677–11686, 2021. 3

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2

[17] Beerend GA Gerats, Jelmer M Wolterink, and Ivo AMJ Broeders. 3D human pose estimation in multi-view operating room videos using differentiable camera projections. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pages 1–9, 2022. 3

[18] Mohsen Gholami, Ahmad Rezaei, Helge Rhodin, Rabab Ward, and Z Jane Wang. Self-supervised 3D human pose estimation from video. *Neurocomputing*, 488:97–106, 2022. 5

[19] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 6

[20] Hengkai Guo, Guijin Wang, Xinghao Chen, and Cairong Zhang. Towards good practices for deep 3D hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017. 3

[21] Lasse Hansen, Marlin Siebert, Jasper Diesel, and Mattias P Heinrich. Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room. *International Journal of Computer Assisted Radiology and Surgery*, 14:1871–1879, 2019. 3

[22] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3D human pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 160–177, 2016. 3

[23] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *Proceedings of the European Conference on Computer Vision*, pages 1–18, 2020. 2

[24] 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. 1

[25] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727, 2019. 1, 2

[26] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision*, pages 196–214, 2020. 1

[27] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015. 1, 6

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[29] 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. 3, 5

[30] Jogendra Nath Kundu, Ambareesh Revanur, Govind Vitthal Waghmare, Rahul Mysore Venkatesh, and R Venkatesh Babu. Unsupervised cross-modal alignment for multi-person 3D pose estimation. In *Proceedings of the European Conference on Computer Vision*, pages 35–52, 2020. 3

[31] Jogendra Nath Kundu, Siddharth Seth, Pradyumna YM, Varun Jampani, Anirban Chakraborty, and R Venkatesh Babu. Uncertainty-aware adaptation for self-supervised 3D human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20448–20459, 2022. 3, 6

[32] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 2, 4

[33] Jialian Li, Jingyi Zhang, Zhiyong Wang, Siqi Shen, Chenglu Wen, Yuexin Ma, Lan Xu, Jingyi Yu, and Cheng Wang. Lidarcap: Long-range marker-less 3d human motion capture with lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20502–20512, 2022. 3

[34] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *Proceedings of the European Conference on Computer Vision*, pages 641–656, 2018. 3

[35] Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3D pose estimation with Plane Sweep Stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11886–11895, 2021. 1, 2, 6, 7

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 1, 5

[37] Filipa Lino, Carlos Santiago, and Manuel Marques. 3d human pose estimation with occlusions: Introducing blendmimic3d dataset and gcn refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4646–4656, 2024. 3

[38] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 2

[39] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, 2015. 5

[40] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3D people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 3

[41] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5442–5451, 2019. 2, 5, 6

[42] Angel Martínez-González, Michael Villamizar, Olivier Canévet, and Jean-Marc Odobez. Residual pose: A decoupled approach for depth-based 3D human pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10313–10318, 2020. 3

[43] 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. 3, 4, 6

[44] Priyanka Patel, Chun-Hao P Huang, Joachim Tesch, David T Hoffmann, Shashank Tripathi, and Michael J Black. AGORA: Avatars in geography optimized for regression analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13468–13478, 2021. 3

[45] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4D-net for learned multi-modal alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15435–15445, 2021. 3

[46] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 2

[47] N Dinesh Reddy, Laurent Guigues, Leonid Pishchulin, Jayan Eledath, and Srinivasa G Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3D pose tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15190–15200, 2021. 1, 2, 4

[48] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3D pose estimation through camera-disentangled representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6040–6049, 2020. 3, 7

[49] Vinkle Srivastav, Afshin Gangi, and Nicolas Padoy. Self-supervision on unlabelled or data for multi-person 2D/3D human pose estimation. In *Medical Image Computing and Computer Assisted Intervention*, pages 761–771, 2020. 3

[50] Vinkle Srivastav, Thibaut Issenhuth, Abdolrahim Kadkhodamohammadi, Michel de Mathelin, Afshin Gangi, and Nicolas Padoy. Mvor: A multi-view rgb-d operating room dataset for 2D and 3D human pose estimation. *arXiv preprint arXiv:1808.08180*, 2018. 6

[51] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 2, 6

[52] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. VoxelPose: Towards multi-camera 3D human pose estimation in wild environment. In *Proceedings of the European Conference on Computer Vision*, pages 197–212, 2020. 1, 2, 4, 6, 7, 8

[53] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 109–117, 2017. 3

[54] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision*, pages 601–617, 2018. 1

[55] Size Wu, Sheng Jin, Wentao Liu, Lei Bai, Chen Qian, Dong Liu, and Wanli Ouyang. Graph-based 3D multi-person pose estimation using multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11148–11157, 2021. 2

[56] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *arXiv preprint arXiv:2204.12484*, 2022. 2, 4, 6

[57] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023. 1

[58] Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster VoxelPose: Real-time 3d human pose estimation by orthographic projection. In *Proceedings of the European Conference on Computer Vision*, pages 142–159, 2022. 2

[59] Jianfeng Zhang, Yujun Cai, Shuicheng Yan, Jiashi Feng, et al. Direct multi-view multi-person 3D pose estimation. *Advances in Neural Information Processing Systems*, 34:13153–13164, 2021. 1, 2, 7

[60] Jingyi Zhang, Qihong Mao, Guosheng Hu, Siqi Shen, and Cheng Wang. Neighborhood-enhanced 3d human pose estimation with monocular lidar in long-range outdoor scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7169–7177, 2024. 3

[61] Meng Zhang, Wenxuan Guo, Bohao Fan, Yifan Chen, Jianjiang Feng, and Jie Zhou. A flexible multi-view multimodal imaging system for outdoor scenes. In *2022 International Conference on 3D Vision (3DV)*, pages 322–331. IEEE, 2022. 1

[62] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2019. 1

[63] Xiheng Zhang, Yongkang Wong, Mohan S Kankanhalli, and Weidong Geng. Unsupervised domain adaptation for 3D human pose estimation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 926–934, 2019. 3

[64] Zihao Zhang, Lei Hu, Xiaoming Deng, and Shihong Xia. Sequential 3D human pose estimation using adaptive point cloud sampling strategy. In *International Joint Conferences on Artificial Intelligence Organization*, pages 1330–1337, 2021. 3

[65] Jingxiao Zheng, Xinwei Shi, Alexander Gorban, Junhua Mao, Yang Song, Charles R Qi, Ting Liu, Visesh Chari, Andre Cornman, Yin Zhou, et al. Multi-modal 3D human pose estimation with 2D weak supervision in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4478–4487, 2022. 3

LiCamPose: Combining Multi-View LiDAR and RGB Cameras for Robust Single-timestamp 3D Human Pose Estimation

Supplementary Material

6. Different Scanning Patterns of Point Cloud

There are various methods to obtain or scan the point cloud: 1) randomly sampling the depth map; 2) sampling the depth map using multiple equidistant horizontal lines to mimic Velodyne LiDARs; and 3) sampling the depth map with the "Rose curve" sampling equation as discussed in our paper to replicate Livox LiDARs. Figure 9 illustrates that the "Rose curve" sampling equation yields minimal information due to its localized concentrated scan. However, Livox LiDARs are more cost-effective than Velodyne LiDARs and have been employed in numerous applications, including surveillance. Additionally, our BaseketBall dataset is captured using Livox LiDARs. Therefore, we adopt the Livox scanning pattern to simulate the point cloud scanning in our experiments.

7. BaseketBall

BasketBall is an outdoor dataset capturing a basketball match using four sensor nodes, each comprising one Livox LiDAR and one RGB camera, in a convergent acquisition setup. The dataset presents challenges due to its extensive coverage, occlusions, and the dynamic motions of the players (Figure 3). We have developed an annotation tool to label the players' 3D bounding boxes and IDs. In the future, we plan to integrate 3D human keypoint annotation into the tool with the assistance of LiCamPose.

8. The Detailed Structure of V2V-Net and Fusion-Net

Figure 8 shows the detailed structure of V2V-Net and Fusion-Net. $i = 1$ for the one of point cloud information and $i = K$ for the one of RGB information, K is the number of joints. X, Y, Z represents the setting of volumetric space, and F represents F_1 and $F2$. As indicated in the legend, the yellow arrow represents a standard 3D convolutional layer, while the blue arrow denotes a Residual Block consisting of two 3D convolutional layers. As indicated in the legend, the yellow arrow represents a standard 3D convolutional layer, while the blue arrow denotes a Residual Block consisting of two 3D convolutional layers.

9. Human Prior Loss

We designed the human prior loss to encourage the network to generate human-like 3D keypoints. The human prior loss comprises three components: 1) the predicted bone lengths should be within a reasonable range; 2) the predicted lengths of symmetric bones should be similar; and 3) the predicted bone angles should be reasonable according to human kinematics.

We set a limited length range for all bones. In our case, we set $l_{\text{min}} = 0.05\text{m}$ and $l_{\text{max}} = 0.7\text{m}$. So the $\mathcal{L}_{\text{length}}$ can be defined as:

$$\mathcal{L}_{\text{length}} = \sum_{b=1}^{N} \mathcal{C}(B_i - l_{\text{max}}, 0) + \mathcal{C}(l_{\text{min}} - B_i, 0), \tag{7}$$

where $\mathcal{C}(\cdot)$ is the clipping function that clip the value greater than 0, N is the number of bones. As to the symmetric bones, we set the symmetric bones as a pair, and set L_2 loss among them. So the $\mathcal{L}_{\text{symm}}$ can be defined as:

$$\mathcal{L}_{\text{symm}} = \sum_{b=1}^{N} \|B_i - B_{\text{symm}(i)}\|_2, \tag{8}$$

where B_{symm} is the symmetric bone of B_i. As to angle loss, we limit the nose-neck-midhip angle and hip-knee-ankle angle specifically to let nose be in front of the body and legs be bent forward. Figure 10 shows the definition of each joint and vectors. Specially, we do not directly calculate the angle of the bones, but calculate the dot product of corresponding vectors. First, we calculate the forward direction vector $\vec{d}_{\text{forward}}$ of the body, which is the cross product of the unit vector from neck to midhip $\overrightarrow{J_0J_2}$ and unit vector from neck to left shoulder $\overrightarrow{J_0J_3}$:

$$\vec{d}_{\text{forward}} = \overrightarrow{J_0J_2} \times \overrightarrow{J_0J_3}, \tag{9}$$

Then, as to the nose-neck-midhip angle, we calculate the unit vector from neck to nose $\overrightarrow{J_1J_0}$ denoted by $\vec{d}_{\text{nose}}$, and we calculate the dot product of $\vec{d}_{\text{nose}}$ and $\vec{d}_{\text{forward}}$ and get the head angle loss:

$$\mathcal{L}_{\text{head\_ang}} = \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{\text{nose}}, 0, 1), \tag{10}$$

where $\mathcal{C}(\cdot)$ is the clipping function that clip the value into 0 to 1. As to the hip-knee-ankle angle, we need to get the midpoint of the hip and ankle denoted by c_{l} and c_{r} for left leg and right leg respectively. Then, we calculate the unit vectors from knee point to the leg's midpoint as $\vec{d}_{\text{l\_leg}}$ and $\vec{d}_{\text{r\_leg}}$. Therefore, we get the leg angle loss:

$$\mathcal{L}_{\text{leg\_ang}} = \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{\text{l\_leg}}, 0, 1) + \mathcal{C}(\vec{d}_{\text{forward}} \cdot \vec{d}_{\text{r\_leg}}, 0, 1), \tag{11}$$

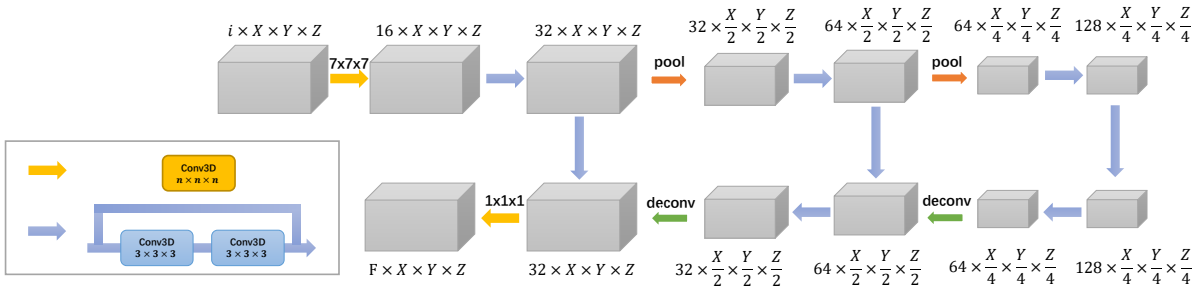

Figure 8. The structure and detailed setting of V2V-Net and Fusion-Net.

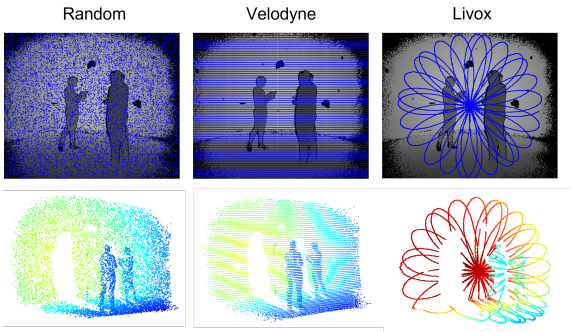

Figure 9. Different scanning patterns of point clouds. All samples shown in this figure are from the same scene, captured at the same time, and contain the same number of points.

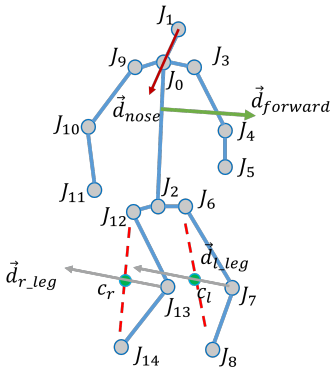

Figure 10. Definition of $\mathcal{L}_{\text{angle}}$.

where $\mathcal{C}(\cdot)$ is the clipping function that clip the value into 0 to 1. Therefore, we can calculate the angle loss:

$$\mathcal{L}_{\text{angle}} = \mathcal{L}_{\text{head\_ang}} + \mathcal{L}_{\text{leg\_ang}}, \tag{12}$$

Finally, we combine the three losses together as the human prior loss:

$$\mathcal{L}_{\text{prior}} = \gamma_1 \mathcal{L}_{\text{length}} + \gamma_2 \mathcal{L}_{\text{symm}} + \gamma_3 \mathcal{L}_{\text{angle}}, \tag{13}$$

where γ_1, γ_2 and γ_3 are the weights of each loss. In our case, we set all the weights as 1.

Table 6. Human detection results on different datasets. "#" means using synthetic datasets (BaseketBallSync and PanopticSync) to pretrain and directly evaluate on corresponding real-world datasets.

Datasets	**Methods**	**Metric**	
		AP_{50}	AP_{70}
BasketBall	MVDet	69.41	37.66
	PointPillars#	88.17	44.26
	PointPillars	**89.77**	**69.96**
Panoptic	VoxelPose	21.17	0.19
	PointPillars#	40.17	6.25
	PointPillars	**73.83**	**13.97**

10. Extended Experiments

In this section, we conduct experiments to verify the advantages of using point cloud input for pedestrian detection. Additionally, we present more examples to explain the relationship between entropy value and pose rationality.

10.1. Human Detection

For evaluating human detection, we assess performance using the established average precision (AP) metric as described in KITTI [16]. We consider detections as true positives if they overlap by more than 70% (AP_{70}) or 50% (AP_{50}).

In our current experiment, we adopt PointPillars [32] to detect human bounding boxes. For comparison with multi-view RGB-based methods, we utilize VoxelPose's CPN [52] and MVDet [23], which is more suitable for large scene applications. In the CMU Panoptic Studio setup, VoxelPose [52] achieves relatively accurate center localization. However, it sets the bounding box size to a constant value (we use 0.8m $\times$ 0.8m $\times$ 1.9m for tighter results, compared to 2m $\times$ 2m $\times$ 2m in [52]), which affects detection performance. In the BasketBall dataset, we adopt MVDet to detect humans. Table 6 shows that the point cloud-based method outperforms the multi-view RGB-based method in

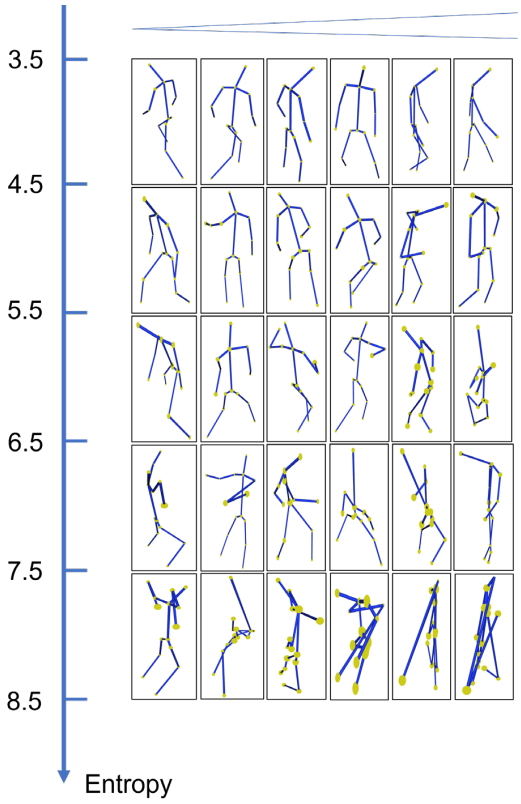

Figure 11. The entropy value and the specific poses. < represents an increasing entropy value from left to right among row's samples. The size of joints' ball represents the magnitude of the joint's entropy value.

terms of AP_{50} and AP_{70}, benefiting from the 3D information of the original point cloud. Additionally, we verify the generalization ability of the point cloud-based method by pretraining it on our synthetic dataset, and it still produces acceptable results.

10.2. The analysis of unsupervised training losses.

Figure 12 qualitatively shows that these designed unsupervised training losses significantly enhance robustness to 2D pose estimation errors.

10.3. Entropy Analysis

Figure 11 shows the entropy value and the specific poses, and we can find that the 3D poses become more and more irrational while the entropy goes up.

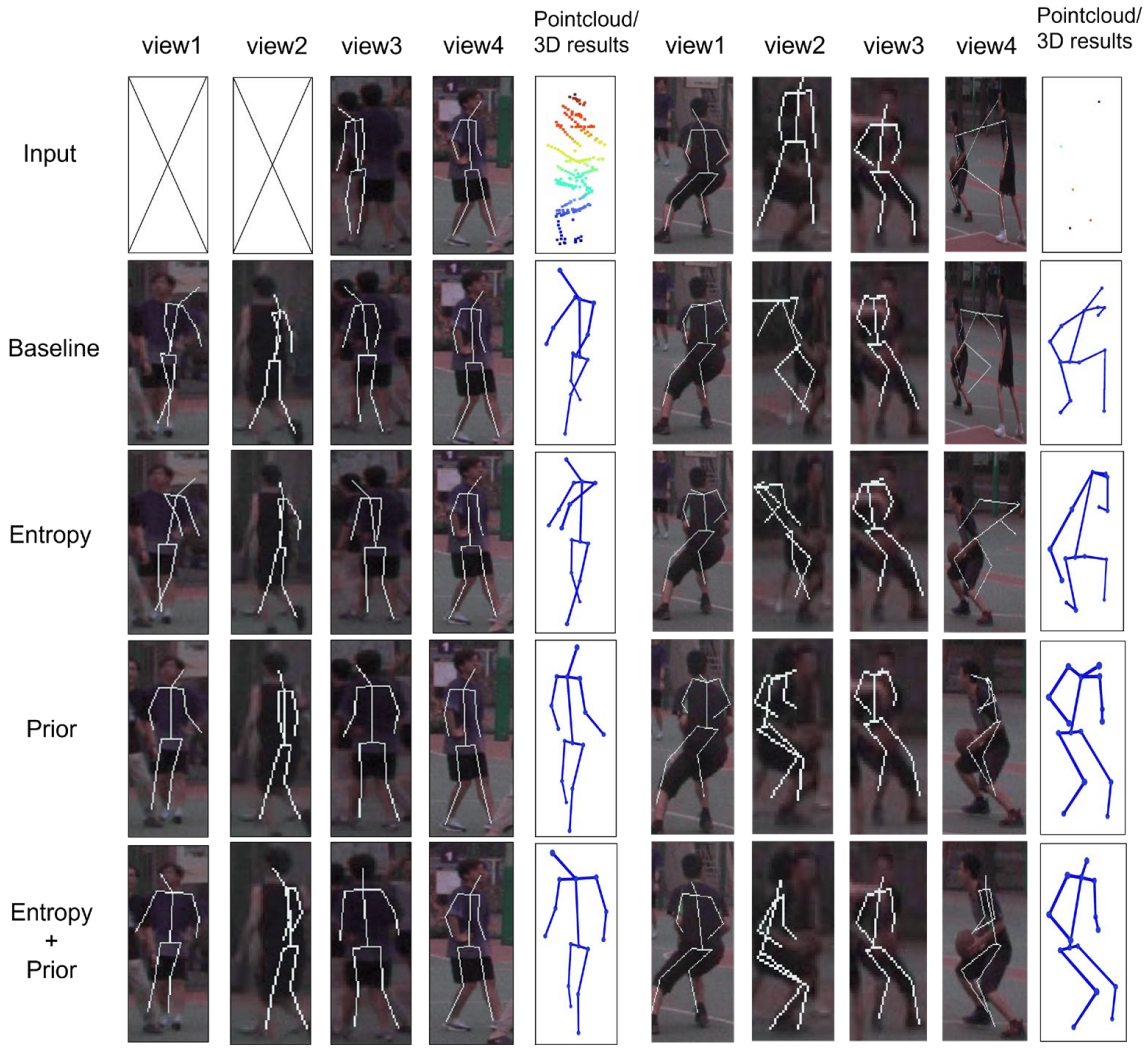

Figure 12. Qualitative visualization on BasketBall about different unsupervised training losses. "Baseline" uses only pseudo 2D pose supervision. "Entropy" indicates the addition of entropy-selected pseudo 3D pose supervision. "Prior" denotes the incorporation of human prior loss.