PROACTIVE MULTI-CAMERA COLLABORATION FOR 3D HUMAN POSE ESTIMATION

Anonymous authors
Paper under double-blind review

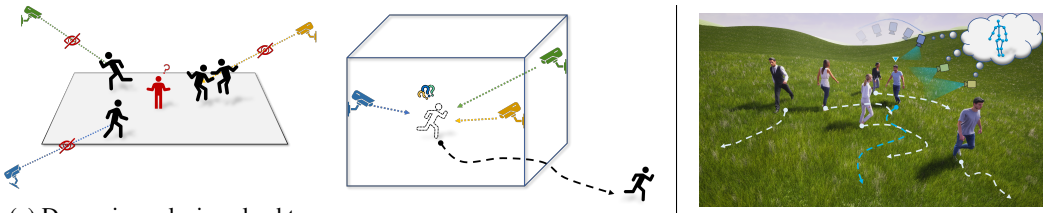
ABSTRACT

For human motion capture (MoCap), particularly outdoors, the fixed-viewpoint multi-camera solutions are susceptible to dynamic occlusions and constrained in capture space. While an active camera approach aims to proactively control the camera poses to find optimal viewpoints for 3D reconstruction. This work introduces a multi-agent reinforcement learning (MARL) scheme to proactive Multi-Camera Collaboration for 3D Human Pose Estimation (MCC-HPE) in dynamic human crowds. At its core is a novel Collaborative Triangulation Contribution Reward (CTCR) that incentivizes agents according to their weighted average marginal contribution to the 3D reconstruction. CTCR improves convergence and alleviates the multi-agent credit assignment issue resulted from using 3D reconstruction accuracy as the shared reward. To better capture environment dynamics and to encourage anticipatory behaviors for occlusion avoidance, we jointly train our model with multiple world dynamics learning tasks. We evaluate our proposed method in four photo-realistic UE4 environments to ensure validity and generalizability. The empirical results show that our methods steadily outperform the fixed and active baselines in different scenarios with various numbers of cameras and humans.

1 INTRODUCTION

Marker-less motion capture (MoCap) has broad commercial applications in many areas such as cinematography, medical research, virtual reality (VR), sports, and *etc*. Their successes can be partly attributed to recent developments in 3D Human pose estimation (HPE) techniques (Tu et al., 2020; Iskakov et al., 2019; Qiu et al., 2019; Jafarian et al., 2019; Pavlakos et al., 2017b; Dong et al., 2019; Lin & Lee, 2021b). A straightforward implementation to solve multi-views 3D HPE is to use fixed cameras. Although being a convenient solution, it is less effective against dynamic occlusions. Moreover, fixed camera solutions confine tracking targets within a constrained space, therefore less applicable to outdoor MoCap. On the contrary, active cameras such as ones mounted on drones can maneuver proactively against incoming occlusions. Owing to its remarkable flexibility, the active approach has thus attracted overwhelming interest (Tallamraju et al., 2020; Ho et al., 2021; Nägeli et al., 2018; Xu et al., 2017; Kiciroglu et al., 2019; Zhou et al., 2018; Saini et al., 2022; Cheng et al., 2018; Zhang et al., 2021).

Previous works have demonstrated the effectiveness of using active cameras for 3D HPE on a single target in indoor (Kiciroglu et al., 2019; Cheng et al., 2018), clean landscapes (Tallamraju et al., 2020; Nägeli et al., 2018; Zhou et al., 2018; Saini et al., 2022) or landscapes with scattered static obstacles (Ho et al., 2021). However, to our best knowledge, we have not seen any existing work



(a) Dynamic occlusions lead to failed reconstruction

(b) Constrained MoCap area

Active MoCap in the wild

Figure 1: **Left**: Two critical challenges in fixed camera approaches. **Right**: Three active cameras collaborate to best reconstruct the 3D pose of the target (marked in ▼).

that experimented with *multiple* $(n > 3)$ active cameras to conduct 3D HPE *in human crowd*. There are two key challenges : **First**, frequent human-to-human interactions lead to **random dynamic occlusions**. Different from previous works that only consider clean landscapes or static obstacles, dynamic scenes require frequent adjustments of cameras' viewpoints for occlusion avoidance while keeping an overall good team formation for multi-view reconstruction. Therefore, it is challenging to achieve optimality with a fixed camera formation or a hand-crafted control policy. In addition, the complex behavioral pattern of a human crowd makes the occlusion patterns less comprehensible and predictable, further increasing the difficulty in control. **Second**, as the size of a camera team grows, the **multi-agent credit assignment issue** becomes prominent which hinders policy learning of the camera agents. Concretely, as a team effort, multi-view 3D HPE requires inputs from multiple cameras to generate an accurate reconstruction. Having more camera agents participate in a reconstruction introduces more redundancy. Though it reduces the susceptibility to reconstruction failure caused by dynamic occlusions, it consequently weakens the association between the individual performance and the reconstruction accuracy of the team, which likely leads to the "lazy agent" problem (Sunehag et al., 2017). A shared reward risks disincentivizing some individual cameras on exploring better viewpoints due to an adequate accuracy already being attained by their teammates alone.

We introduce a proactive multi-camera collaboration framework based on multi-agent reinforcement learning (MARL) for real-time distributive adjustments of multi-camera formation for 3D HPE in a human crowd. In our approach, multiple camera agents perform seamless collaboration for successful reconstructions of 3D human poses. Additionally, it is a decentralized framework which offers flexibility over the size of formation and eliminates dependency to a control hierarchy or a centralized entity. In regards of the **first** challenge, we argue the utmost importance of the model's capability to anticipate human motions and future states of the environment. To model these properties into the state representations, we incorporate *World Dynamics Learning* into the training of our model, *i.e.*, learning with five auxiliary tasks to predict target's position, pedestrians' positions, self state, teammates' states and team reward.

To tackle the **second** challenge, we further introduce the *Collaborative Triangulation Contribution Reward (CTCR)*, which incentivizes each agent according to its characteristic contribution to a 3D reconstruction. Inspired by the Shapley Value (Rapoport, 1970), CTCR computes the average weighted marginal contribution to the 3D reconstruction for any given agent over all possible coalitions that contain it. This reward aims to directly associate agents' levels of participation with their return and guides their policy learning when the team reward alone is insufficient to produce such direct association. Moreover, CTCR penalizes occluded camera agents more efficiently than the shared reward, encouraging emergent occlusion avoidance behaviors. Empirical results show that CTCR can accelerate convergence and increase reconstruction accuracy. Furthermore, CTCR is a general approach that can benefit policy learning in active 3D HPE and serve as a new assessment metric for view selection in other multi-view reconstruction tasks.

In experiments, we build photo-realistic environments based on Unreal Engine 4 (UE4) and UnrealCV (Qiu et al., 2017) for training and testing. Our environments can simulate realistic-behaving crowds with assurances of high fidelity and customizability. We train the agents on a *Blank* environment and validate the generalizability on three unseen scenarios with different landscapes, levels of illumination, human appearances, and various quantities of cameras and humans. The empirical results show that our method can achieve more accurate and stable 3D pose estimates than off-the-shelf passive- and active-camera baselines. To help facilitate more fruitful research on this topic, we will open-source our environments that are integrated with OpenAI Gym-API (Brockman et al., 2016) and together with a dedicated visualization tool. Here we summarize the key contributions of our work:[1]

- We formulate active 3D human pose estimation in a human crowd problem as a Dec-POMDP, and proposed a novel multi-camera $(n \geq 3)$ collaboration framework.
- We propose CTCR to improve credit assignment in multi-camera collaboration and demonstrate notable improvements in reconstruction accuracy compared to both passive and active baselines.
- We introduce five auxiliary tasks called to help the model learn environment dynamics, further enhancing the model's ability to handle highly dynamic scenes.
- We contribute high-fidelity environments built for simulating realistic-looking human crowds with authentic behaviors, along with visualization software for frame-by-frame video analysis.

[1]Project Website for demo videos: https://sites.google.com/view/active3dpose/home

2 Related Work

3D Human Pose Estimation (HPE) Recent works demonstrated recovering 3D human pose from a single monocular image (Ma et al., 2021; Pavlakos et al., 2017a; Martinez et al., 2017; Kanazawa et al., 2018; Pavlakos et al., 2018; Sun et al., 2018; Ci et al., 2019; Zeng et al., 2020; Ci et al., 2020) or monocular video (Mehta et al., 2017; Hossain & Little, 2018; Pavllo et al., 2019; Kocabas et al., 2020). Another line of work utilized multi-camera systems (i.e. triangulation) to eliminate ambiguity and provide redundancy to visibility (Qiu et al., 2019; Jafarian et al., 2019; Pavlakos et al., 2017b; Dong et al., 2019; Lin & Lee, 2021b; Tu et al., 2020; Iskakov et al., 2019). However, these multi-view methods are limited to the indoor laboratory environment with fixed cameras. This work presents an active camera system with multiple mobile cameras for mocap in outdoor scenes.

Proactive Motion Capture Few previous works studied proactive motion capture with a single mobile camera (Zhou et al., 2018; Cheng et al., 2018; Kiciroglu et al., 2019). In comparison, more works studied the control of a multi-camera team. Among them, many are based on optimization with various system designs, including marker-based (Nägeli et al., 2018), RGBD-based (Xu et al., 2017), two-stage system (Saini et al., 2019; Tallamraju et al., 2019), hierarchical system (Ho et al., 2021), *etc*. It is important to note that all the above methods deal with static occlusion sources or clean landscapes. Also, the common similarity shared by these works is the use of hand-crafted optimization objectives and some fixed-form camera formations. These factors resulted in poor adaptability to dynamic scenes saturated with uncertainties. Recently, RL-based methods received more attention due to their potential for dynamic formation adjustments. These works focused on active 3D HPE in the Gazebo simulation (Tallamraju et al., 2020) or Panoptic dome (Joo et al., 2015; Pirinen et al., 2019; Gärtner et al., 2020) for active view selection. Among them, AirCapRL (Tallamraju et al., 2020) shares similarities with our work. However, it is restricted to coordinating between two cameras in clean landscapes without occlusion. We study the collaboration between more cameras and resolve the credit assignment issue with novel CTCR incentives. Meanwhile, we consider a more challenging scenario with multiple distracting humans as dynamic occlusions, which requires dedicated algorithms to handle.

Multi-Camera Collaboration Many works in computer vision have studied multi-camera collaboration and designed active camera systems accordingly. Earlier works (Qureshi & Terzopoulos, 2007)(Collins et al., 2003; Matsuyama et al., 2012) focused on developing a network of pan-tile-zoom (PTZ) cameras. Owing to recent advances in MARL algorithms (Lowe et al., 2017; Sunehag et al., 2017; Rashid et al., 2018; Wang et al., 2020; Yu et al., 2021), many works have formulated multi-camera collaboration as a multi-agent learning problem and solved it using MARL algorithms accordingly (Li et al., 2020; Xu et al., 2020; Wenhong et al., 2022; Fang et al., 2022; Sharma et al., 2022). However, most works focus on the target tracking problem, whereas this work attempts to solve the task of 3D HPE. Compared with tracking, 3D HPE has more strict criteria for optimal view selections due to correlations across multiple views, which requires stronger collaboration between cameras. To our best knowledge, this work is the first to experiment with various camera agents ($n \geq 3$) to learn multi-camera collaboration strategies for active 3D HPE.

3 Proactive Multi-Camera Collaboration

This section will explain the formulation of multi-camera collaboration in 3D HPE as a Dec-POMDP. Then we will describe our proposed solutions for modeling the virtual environment's complex dynamics and strengthening credit assignment in the multi-camera collaboration task.

3.1 Problem Formulation

We formulate the multi-cameras 3D HPE problem as a Decentralized Partially-Observable Markov Decision Process (Dec-POMDP), where each camera is considered as an agent which is decentralized controlled and has partial observability over the environment. Formally, a Dec-POMDP is defined as $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}^n, n, P, r, \gamma \rangle$, where $\mathcal{S}$ denotes the global state space of the environment, including all human states and camera states in our problem. $o_i \in \mathcal{O}$ denotes the agent i's local observation, *i.e.*, the RGB image observed by camera i. $\mathcal{A}$ denotes the action space of an agent and $\mathcal{A}^n$ represents the joint action space of all n agents. $P : \mathcal{S} \times \mathcal{A}^n \rightarrow \mathcal{S}$ is the transition probability function $P(s^{t+1}|s^t, \boldsymbol{a}^t)$, in which $\boldsymbol{a}^t \in \mathcal{A}^n$ is a joint action by all n agents. At each timestep t, every agent obtains a local view o_i^t from the environment s^t and then pre-process o_i^t to form i-th agent's local observation $\tilde{o}_i^t$. The agent performs action $a_i^t \sim \pi_i^t|\tilde{o}_i^t$ and receives its reward $r(s^t, \boldsymbol{a}^t)$. $\gamma \in (0, 1]$ is the discount factor used to calculate the cumulative discounted reward $G(t) = \sum_{t' \geq t} \gamma^{t'-t}\, r(t')$.

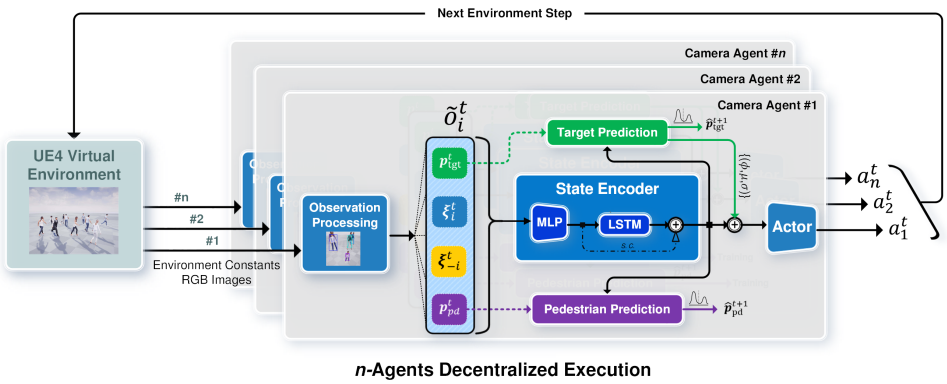


Figure 2: Simplified Pipeline Architecture. For i agent at t step, the environment returns local camera view o_i^t and environment constants that are pre-processed to generate local observation $\tilde{o}_i^t$ (described here). Target Prediction and Pedestrian Prediction modules generate predictions $\hat{p}_{\text{tgt}}^t$ and $\hat{p}_{\text{pd}}^t$, respectively. Mixture Density Network(MDN) parameters for the Target Prediction Module, $\{(\phi, \mu, \sigma)\}_{\text{MDN}_{\text{tgt}}}$, is concatenated to the state encoder's output to enhance the feature.

In a cooperative team, the objective is to learn a group of decentralized policies $\{\pi_i(a_i^t|\tilde{o}_i^t)\}_{i=1}^n$ that maximizes $\mathbb{E}_{(s,\boldsymbol{a})\sim\boldsymbol{\pi}}[G(t)]$. For convenience, we let i means the agent index, $[\![n]\!] = \{1, \ldots, n\}$ is the set for all agents, and $-\boldsymbol{i} = [\![n]\!] \setminus \{i\}$ are all agents except agent i.

Observation Camera agents have partial observability over the environment. The pre-processed observation $\tilde{o}_i = (p_i, \xi_i, \xi_{-\boldsymbol{i}})$ of the camera agent i consists of: **(1)** p_i, a set of states of visible humans to agent i, containing information including the detected human bounding box in the 2D local view, the 3D positions and orientations of all visible humans measured in both local coordinate frame of camera i and world coordinates; **(2)** own camera pose ξ_i showing the camera's position and orientation in world coordinates; **(3)** peer cameras poses $\xi_{-\boldsymbol{i}}$ showing their positions and orientations in world coordinates and are obtained via multi-agent communication.

Action Space The action space of each camera agent consists of the velocity of 3D egocentric translation (x, y, z) and velocity of 2D pitch-yaw rotation (θ, ψ). To reduce the exploration space for state-action mapping, the agent's action space is discretized into three levels across all five dimensions. At each timestep, the camera agent can move its position by $[+\delta, 0, -\delta]$ in (x, y, z) directions and rotate about pitch-yaw axes by $[+\eta, 0, -\eta]$ degrees.

3.2 Framework

This section will describe the technical framework that constitutes our camera agents, which contains a Perception Module and a Controller Module. The Perception Module maps the original RGB images taken by the camera to numerical observations. The Controller Module takes the numerical observations and correspondingly outputs the controls. Fig. 2 illustrates this framework.

Perception Module The perception module executes a procedure consisting of four sequential stages: **(1)** 2D HPE. The agent performs 2D human detection and pose estimation on the observed RGB image with the YOLOv3 (Redmon & Farhadi, 2018) detector and the HRNet-w32 (Sun et al., 2019) pose estimator, respectively. Both models are pre-trained on the COCO dataset, (Lin et al., 2014) and their parameters are kept frozen during policy learning of camera agents to ensure cross-scene generalization. **(2)** Person ReID. A ReID model (Zhong et al., 2018) is used to distinguish people in a scene. For simplicity, an appearance dictionary of all to-be-appeared people is built in advance following (Gärtner et al., 2020). At test time, the ReID network computes features for all detected people and identifies different people by comparing features to the pre-built appearance dictionary. **(3)** Multi-agent Communication. Detected 2D human pose, IDs, and own camera pose are broadcasted to other agents. **(4)** 3D human pose reconstruction. 3D human pose is finally reconstructed via local triangulation after receiving other agents' communications. Human position and orientation that are parts of p_i^t can then be calculated from the reconstructed human pose. The whole process is illustrated in Appendix Fig. 10.

Controller Module The controller module consists of a state encoder E and an actor network A. The state encoder E takes $\tilde{o}_i^t$ as input, encoding the temporal dynamics of the environment via LSTM. The future states of the target, pedestrians, and cameras are modeled using Mixture Density Network (MDN) (Bishop, 1994) to account for uncertainty. During model inference, it computes

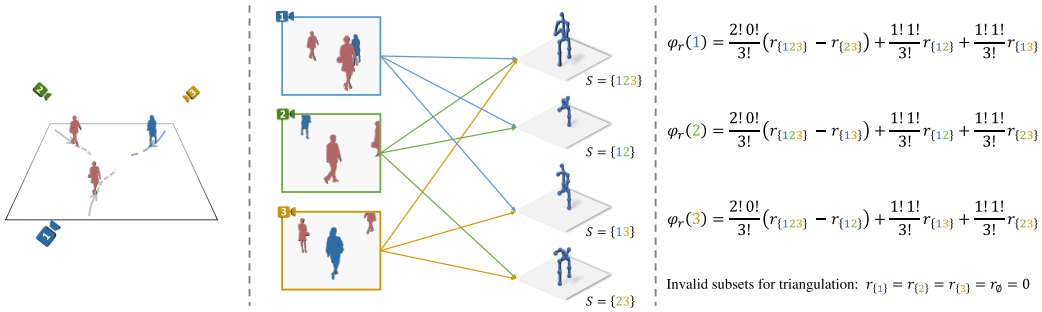


Figure 3: An instance of computing Collaborative Triangulation Contribution Reward (CTCR) for every camera. **Left:** Three cameras tasked to perform HPE on the target person avoiding occlusions from two pedestrians. **Middle:** Three cameras can constitute four valid coalitions, each is generating a different 3D HPE. **Right:** Compute CTCR (without re-scaling by n) for each camera.

target position prediction, and then the (ϕ, μ, σ) parameters of target prediction MDN are used as a part of the inputs to the actor network to enhance feature encoding. Please refer to Section 3.4 for more details regarding training the MDN.

$$\text{Feature Embedding} \quad z_i^t = \text{MLP}(\tilde{o}_i^t), \tag{1}$$

$$\text{Temporal Modeling} \quad h_i^t = \text{LSTM}(z_i^t, h_i^{t-1}), \tag{2}$$

$$\text{Human Trajectory Prediction} \quad \hat{p}_{\text{tgt/pd}}^{t+1} = \text{MDN}(z_i^t, h_i^t, p_{\text{tgt/pd}}^t), \tag{3}$$

$$\text{Final Embedding} \quad e_i^t = E\left(\tilde{o}_i^t, h_i^{t-1}\right) = \text{Concat}(z_i^t, h_i^t, \{(\phi, \mu, \sigma)\}_{\text{MDN}_{\text{tgt}}}), \tag{4}$$

where p_{tgt} and p_{pd} refer to the state of the target and the pedestrian, respectively. The actor network A consists of 2 fully-connected layers that output the action, $a_i^t = A(E(\tilde{o}_i^t, h_i^{t-1}))$.

3.3 Reward Structure

To alleviate the credit assignment issue that arises in multi-camera collaboration, we propose the Collaborative Triangulation Contribution Reward (CTCR). We start by defining a base reward that reflects the reconstruction accuracy of the triangulated pose generated by the camera team. Then we explain how our CTCR is computed based on this base team reward.

Reconstruction Accuracy as A Team Reward To directly reflect the reconstruction accuracy, the reward function negatively correlates with the pose estimation error (Mean Per Joint Position Error, MPJPE) of the multi-camera triangulation. Formally,

$$r(X) = \begin{cases} 0, & |X| \leq 1, \\ 1 - \text{Gemen}\left(\text{MPJPE}(X)\right), & |X| \geq 2. \end{cases} \tag{5}$$

Where X denotes a set of cameras that participate in the triangulation. $\text{Gemen}(\cdot) = \frac{2(\cdot/c)^2}{(\cdot/c)^2+4}$, is the Geman-McClure smoothing (Barron, 2019) function used to stabilize policy updates. Empirically we set $c = 50\text{mm}$ in our experiments. In the context of our MAPPO baseline, X represents the entire camera team, and each camera receives a shared team reward $r(X)$, from which the credit assignment issue arises. Intuitively, when one camera is occluded, all cameras receive the same reduced reward. It is hard to tell who the culprit is. To deal with this issue, we introduce CTCR.

Collaborative Triangulation Contribution Reward (CTCR) CTCR computes each agent's individual reward based on its marginal contribution to the collaborative multi-view triangulation. Please see Fig. 3 for a rundown of computing CTCR for a 3-cameras team. The contribution of agent i can be measured by:

$$\text{CTCR}(i) = n \cdot \varphi_r(i), \qquad \varphi_r(i) = \sum_{S \subseteq [\![n]\!] \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} [r(S \cup \{i\}) - r(S)], \tag{6}$$

Where n denotes the total number of agents. S denotes all the subsets of $[\![n]\!]$ not containing agent i. $\frac{|S|!(n-|S|-1)!}{n!}$ is the normalization term. $[r(S \cup \{i\}) - r(S)]$ means the marginal contribution of agent i. Note that $\sum_{i \in [\![n]\!]} \varphi_r(i) = r([\![n]\!])$. We additionally multiply a constant n to rescale the CTCR to have the same scale as the team reward. Specially, in the 2-cameras case, the individual CTCR should be equivalent to the team reward, *i.e.*, $\text{CTCR}(i = 1) = \text{CTCR}(i = 2) = r(\{1, 2\})$.

(a) Blank (b) School Gym (c) Urban Street (d) Wilderness

Figure 4: UNREALPOSE: Our virtual environments for active 3D HPE in human crowd

Table 1: A comparison with other active control methods on the control of three cameras. We also report fixed camera results as a reference. MPJPE is averaged over six environment instances, each with a different number of humans (1 to 6, respectively).

	Fixed Cameras	MAPPO	AirCapRL (Reproduced)	Rule-based Formation	**Ours**
MPJPE (mm) ↓	122.4	71.6	70.2	70.0	**56.7**

3.4 LEARNING MULTI-CAMERA COLLABORATION VIA MARL

We employ the multi-agent learning variant of PPO (Schulman et al., 2017) called Multi-Agent PPO (MAPPO) (Yu et al., 2021) to learn the collaboration strategy. Alongside the RL loss, we jointly train the model with five auxiliary tasks that encourage comprehension of the world dynamics and the stochasticity in human behaviors. The pseudocode of the learning algorithm can be found in Appendix A.

World Dynamics Learning (WDL) We use the encoder's hidden states (z_i^t, h_i^t) as the basis to model the world. Three WDL objectives correspond to modeling agent dynamics : **(1)** learning the forward dynamics of the camera $P_1(\xi_i^{t+1}|z_i^t, h_i^t, a_i^t)$, **(2)** prediction of team reward $P_2(r^t|z_i^t, h_i^t, a_i^t)$, **(3)** prediction of future position of peer agents $P_3(\xi_{-\boldsymbol{i}}^{t+1}|z_i^t, h_i^t, a_i^t)$. Two WDL objectives correspond to modeling human dynamics: **(4)** prediction of future position of target person $P_4(p_{\text{tgt}}^{t+1}|z_i^t, h_i^t, p_{\text{tgt}}^t)$, **(5)** prediction of future position of pedestrians, $P_5(p_{\text{pd}}^{t+1}|z_i^t, h_i^t, p_{\text{pd}}^t)$. All the probability functions above are approximated using Mixture Density Networks (MDNs) (Bishop, 1994).

Total Training Objectives $\mathcal{L}_{\text{Train}} = \mathcal{L}_{\text{RL}} + \lambda_{\text{WDL}}\,\mathcal{L}_{\text{WDL}}$. The $\mathcal{L}_{\text{RL}}$ is the reinforcement learning loss consisting of PPO-Clipped loss and centralized-critic network loss similar to MAPPO (Yu et al., 2021). $\mathcal{L}_{\text{WDL}} = -\frac{1}{n}\sum_l \lambda_l \sum_i \mathbb{E}[\log P_l(\cdot|\tilde{o}_i^t, h_i^t, a_i^t)]$ is the world dynamics learning loss that consists of MDN supervised losses on the five prediction tasks mentioned above.

4 EXPERIMENT

In this section, we first introduce our novel environment, UNREALPOSE, used for training and testing the learned policies. Then we compare our method with various passive multi-camera baselines and perform an ablation study on the effectiveness of the proposed CTCR and WDL objectives. Additionally, we evaluate the effectiveness of the learned policies by comparing them against other active multi-camera methods. Lastly, we test our method in four different scenarios to showcase its robustness.

4.1 UNREALPOSE: A VIRTUAL ENVIRONMENT FOR PROACTIVE HUMAN POSE ESTIMATION

We built a virtual environment for active HPE based on Unreal Engine 4 (UE4), a powerful 3D game engine that can provide real-time and photo-realistic renderings provided for making visually-stunning video games. The environment handles the interactions between realistic-behaving human crowds and camera agents. Here is a list of characteristics of UnrealPose that we would like to highlight: **Realistic**: Diverse generations of human trajectories, built-in collision avoidance, and several scenarios with different human appearance, terrain, and level of illumination. **Flexibility**: extensive configuration in numbers of humans, cameras, or their physical properties, more than 100 MoCap action sequences incorporated. **RL-Ready**: integrated with OpenAI Gym API, overhaul the communication module in the UnrealCV (Qiu et al., 2017) plugin with inter-process communication (IPC) mechanism. For more detailed descriptions, please refer to Appendix Section B.

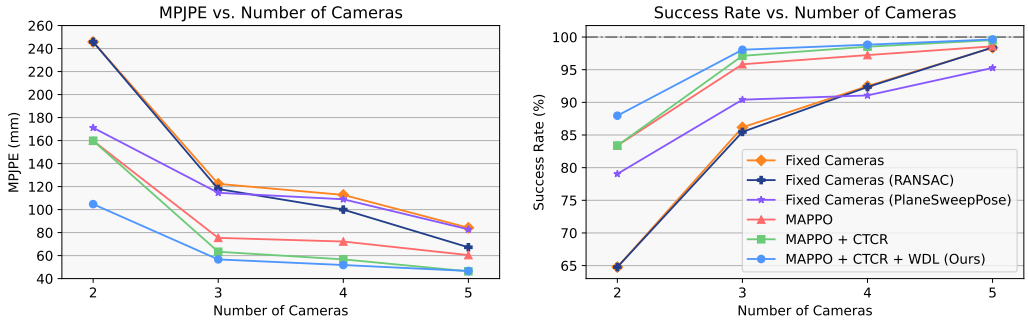


Figure 5: Train results show the performance difference, relative to the number of cameras, between the baselines and our methods. Each data point aggregates the results in environments with a different number of humans (1 to 6, respectively). The threshold for success rate is set to $\tau = 200\text{mm}$.

4.2 Evaluation Metrics

We adopt Mean Per Joint Position Error (MPJPE) between the ground truth and the reconstructed 3D pose as our primary evaluation metric. While cameras making adjustments toward higher-quality perceptions could take several frames to complete or have high peaks in MPJPE that were overlooked by the mean aggregation. Hence we additionally use the "success rate" metric to evaluate the robustness and smooth execution of the learned policies. The success rate is the ratio of frames in an episode with MPJPE lower than τ. Formally, $\text{SuccessRate}(\tau) = P(\text{MPJPE} \leq \tau)$. This metric is a proxy measure for the integrity of multi-view coordination. One instance of bad coordination may cause partial occlusions or have too many overlapping perceptions due to a bad formation. Both will result in a dramatic increase in MPJPE and consequently cause a drop in the success rate.

4.3 Results and Analysis

We trained all the learning-based control policies in a mix of 60 instances of the Blank Environment containing 1 to 6 humans (uniformly sampled). Each training run has 500,000 steps of environment interactions. For fair evaluation, we report all of the mean metrics based on the data of a hundred 500-step episodes. In all experiments, the cameras and humans interact in a $10\text{m} \times 10\text{m}$ area.

Active vs. Passive To show the necessity of proactive camera control, we compare the active camera control methods with three passive methods, *i.e.*, Fixed Camera, Fixed Camera (RANSAC), and Fixed Camera (PlaneSweepPose). "Fixed Cameras" denotes that the poses of the cameras are fixed, hanging 3 meters above ground with $-35°$ pitch angles (i.e. overseeing the entire area). The placements of these fixed cameras are carefully determined with strong priors, for example, right-angle, triangle, square, and pentagon formations for 2, 3, 4, and 5 cameras, respectively. "RANSAC" denotes the method that uses RANSAC (Fischler & Bolles, 1981) for enhanced triangulation. "PlaneSweepPose" represents the off-the-shelf learning-based method (Lin & Lee, 2021b) for multi-view 3D HPE. Please refer to the Appendix E.1 for more implementation details. We show the MPJPE(mm) and Success Rate versus a different number of cameras in Fig. 5. We observe that all passive baselines are being outperformed by the active approaches due to their inability to adjust camera views against dynamic occlusions. The improvement of active approaches is especially significant when the number of cameras is less, i.e. when the camera system has little or no redundancy against occlusions. Notably, the MPJPE achieved by our 3-camera control strategy is even lower than the MPJPE from 5 Fixed cameras. This suggests that proactive strategies can help reduce the number of cameras necessary for deployment.

The Effectiveness of CTCR and WDL We also perform ablation studies on the two proposed modules (CTCR and WDL) to analyze their benefits to performance. We take "MAPPO" as the baseline for comparison, which is our method but trained instead by a shared global reconstruction reward and the absence of any world dynamics modeling. Fig. 5 shows the consistent performance gap between the "MAPPO" baseline and our methods (MAPPO + CTCR). This explains the increasing difficulty in credit assignments with more cameras. The proposed CTCR mitigates such an issue by computing the weighted marginal contribution of each camera. Also, CTCR promotes faster convergence. Training curves are shown in Appendix Fig. 13. Training with WDL objectives further improves the MPJPE metric for our 2-Camera model by approx. 60%. However, its supporting effect

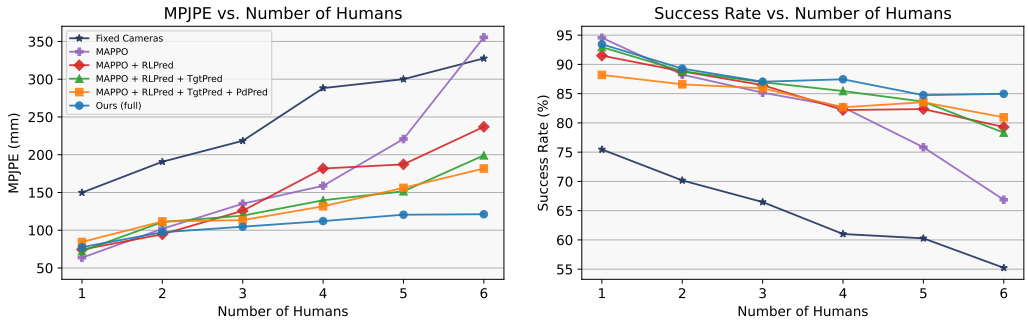


Figure 6: Detailed ablation study on WDL modules in the 2-cameras environment. **RLPred**: learn agent's forward dynamics, **TgtPred**: predict future target position, **PdPred**: predict future pedestrian position. The **full** model additionally concatenates predicted future target position to observation embedding e^t.

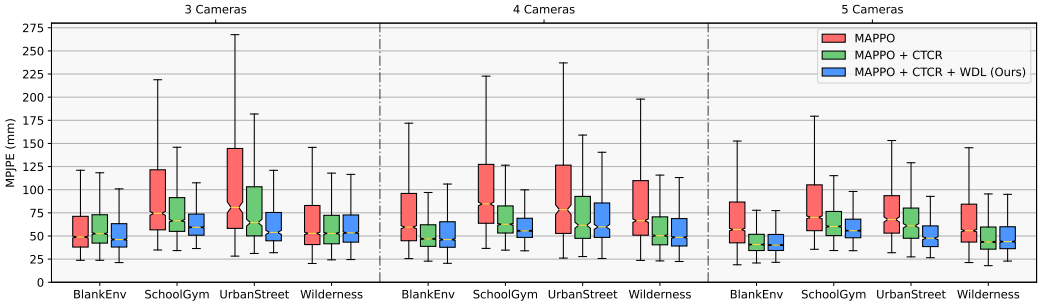
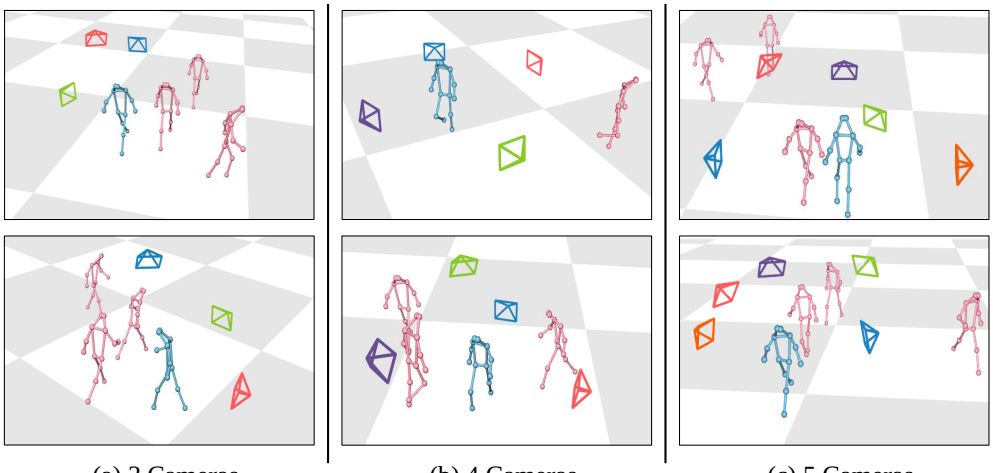


Figure 7: Evaluation results for testing our multi-camera collaboration policy (three to five cameras) in four different environments that have six walking humans (including the target human).

gradually weakens with the increasing number of cameras. We argue that is caused by the more complex dynamics involved with more cameras simultaneously interacting in the same environment. Notably, we observe that the agents trained with WDL are of better generalization in unseen scenes, as shown in Fig. 7.

Ablation Study within the WDL Objectives We perform a detailed ablation study regarding the effect of each WDL sub-task on the model performance. As shown in Fig. 6, we can observe that the MPJPE metric gradually decreases as we incorporate more WDL losses. This aligns with our assumptions that training the model with world dynamics learning objectives will promote the model's ability to capture a better representation of future states, which in turn increases performance. Our method (full, + RLPred + TgtPred + PdPred + Merge) additionally demonstrates the importance of incorporating information regarding the target's future state into the encoder's output features. Predicting the target's future states should not only be used as an auxiliary task but should also directly influence the inference process of the actor model. Moreover, "Ours" method suffers less performance drop from increasing the number of pedestrians. In contrast, the accuracy of the baselines drops dramatically as the number of pedestrians increases. We argue that the proposed WDL objectives can improve the model's generalizability.

Versus Other Active Methods To show the effectiveness of the learned policies, we further compare our method with other active multi-camera formation control methods in 3-cameras BlankEnv. **(1)** "MAPPO" (Yu et al., 2021) and **(2)** AirCapRL (Tallamraju et al., 2020) are two learning-based methods, built on PPO (Schulman et al., 2017). The main difference between these two methods is the reward shaping technique, *i.e.*, AirCapRL additionally employs multiple carefully-designed rewards (Tallamraju et al., 2020) for learning. We also program a **(3)** rule-based fixed formation control method (keeping an equilateral triangle, spread by 120°) to track and surround the target person. Results are shown in Table 1. Interestingly, these three baselines achieve similar reconstruction performance. Our method notably outperforms them, indicating a more effective multi-camera collaboration strategy for 3D HPE. For example, our method learns a spatially spread-out formation while automatically adjusting to avoid impending occlusion.



(a) 3 Cameras (b) 4 Cameras (c) 5 Cameras

Figure 8: Visualizations for the emergent formations in controlling three to five cameras. The target person is colored in blue and others are colored in pink. The camera agents learn to spread out and avoid occlusions while maintaining good reconstruction accuracy without a centralized controller.

Generalize to Various Scenarios We train the control policies in BlankEnv while testing them in three other realistic environments (SchoolGym, UrbanStreet, and Wilderness) to evaluate their generalization capability in unseen scenarios. Fig. 7 shows that our method consistently outperforms baseline methods with lower variance in MPJPE during the evaluations in three test environments. Note that we report the results in the BlankEnv as a reference.

Qualitative Analysis In Figure 8, we show six examples of the emergent camera formations under our trained policies using the proposed method (CTCR + WDL). The camera agents learn to spread out and ascent above humans to avoid occlusions and collision. Their placements in an emergent formation are not assigned by other entities but rather determined by the decentralized control policies themselves based on observations. For more vivid examples of emergent formations, please refer to the project website for the demo videos.[2]

5 Conclusion and Discussion

To the best of our knowledge, this paper presents the first active multi-camera system targeting the 3D reconstruction in a dynamic crowd. It is also the first study regarding active 3D HPE that promptly experimented with multi-camera collaborations at different scales and scenarios. We propose CTCR to alleviate the multi-agent credit assignment issue when the camera team scales and identifies multiple auxiliary tasks that are helpful for representation learning of complex dynamics. As a final note, we have decided to open-source our UE4 virtual environment and the visualization software.

Limitation Admittedly, a couple of aspects of this work have room for improvement. Firstly, the camera agents received the positions for their teammates via non-disrupting broadcasts of information, which may be prone to packet losses during deployment. One idea is to incorporate a specialized protocol for multi-agent communication into our pipeline, such as TarMAC (Das et al., 2019), I2C (Ding et al., 2020), or ToM2C(Wang et al., 2022). Secondly, intended initially to reduce the communication bandwidth (for example, to eliminate the need for image transmission between cameras), the current pipeline comprises a Human Re-Identification module requiring a pre-scan memory on all to-be-appeared human subjects. For some out-of-distribution humans' appearances, the current ReID module may not be able to recognize. Though Active3DPose can accommodate a more sophisticated ReID module to resolve this shortcoming. Lastly, we assume near-perfect calibrations for a group of active and mobile cameras, which is difficult to sustain in practice.

[2]Project Website for demo videos: https://sites.google.com/view/active3dpose/home

6 ETHICS DISCUSSION

There are many great benefits brought by the advances in active 3D HPE technologies, for example, biomechanical analysis in sports and automated video-assisted coach (AVAC). However in reality, misusing these technologies through repressive surveillance may lead to privacy infringements and human rights violations. We condemn these malicious acts and advocate fair uses of our virtual environment UNREALPOSE and all other 3D HPE technologies. For maximum safety compliance in the experiments involving human-machine interactions, we have introduced safety-related features to minimize the probability of collision (Appendix Section F).

7 REPRODUCIBILITY STATEMENT

We will publicly release the project upon official publishing of this work. The release will include the UNREALPOSE environment, RLLIB training scripts, and the visualization tool. The UE4 binaries and the pre-trained models used in our experiments will also be available for download. The RLLIB library is specifically chosen as the base architecture of the implementation of our MARL algorithm. This is to ensure cross-platform compatibility, scalability, maintainability and transparency.

REFERENCES

Epic games incorporated. unreal engine 4. https://www.unrealengine.com.

Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

Christopher M. Bishop. Mixture density networks. Workingpaper, Aston University, 1994.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2527–2530, 2012.

Wei Cheng, Lan Xu, Lei Han, Yuanfang Guo, and Lu Fang. ihuman3d: Intelligent human body 3d reconstruction using a single flying camera. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1733–1741, 2018.

Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2262–2271, 2019.

Hai Ci, Xiaoxuan Ma, Chunyu Wang, and Yizhou Wang. Locally connected network for monocular 3d human pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1429–1442, 2020.

Robert T Collins, Omead Amidi, and Takeo Kanade. Acquiring multi-view video with an active camera system. In *Multisensor Surveillance Systems*, pp. 135–147. 2003.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pp. 1538–1546, 2019.

Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 33:22069–22079, 2020.

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*, pp. 7792–7801, 2019.

Zeyu Fang, Jian Zhao, Mingyu Yang, Wengang Zhou, Zhenbo Lu, and Houqiang Li. Coordinate-aligned multi-camera collaboration for active multi-object tracking. *arXiv preprint arXiv:2202.10881*, 2022.

Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Erik Gärtner, Aleksis Pirinen, and Cristian Sminchisescu. Deep reinforcement learning for active human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10835–10844, 2020.

Cherie Ho, Andrew Jong, Harry Freeman, Rohan Rao, Rogerio Bonatti, and Sebastian Scherer. 3d human reconstruction in the wild with collaborative aerial cameras. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5263–5269, 2021.

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

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*, pp. 7718–7727, 2019.

Yasamin Jafarian, Yuan Yao, and Hyun Soo Park. Monet: Multiview semi-supervised keypoint via epipolar divergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 753–762, 2019.

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*, pp. 3334–3342, 2015.

Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7122–7131, 2018.

Sena Kiciroglu, Helge Rhodin, Sudipta N Sinha, Mathieu Salzmann, and Pascal Fua. Activemocap: optimized drone flight for active human motion capture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16–20, 2019.

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5253–5263, 2020.

Jing Li, Jing Xu, Fangwei Zhong, Xiangyu Kong, Yu Qiao, and Yizhou Wang. Pose-assisted multi-camera collaboration for active object tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 759–766, 2020.

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 (CVPR)*, pp. 11886–11895, June 2021a.

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*, pp. 11886–11895, 2021b.

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 *European conference on computer vision*, pp. 740–755, 2014.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems*, 2017.

Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Hai Ci, and Yizhou Wang. Context modeling in 3d human pose estimation: A unified perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6238–6247, 2021.

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

Takashi Matsuyama, Shohei Nobuhara, Takeshi Takai, and Tony Tung. Active camera system for object tracking and multi-view observation. In *3D Video and Its Applications*, pp. 45–85. Springer, 2012.

Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

Tobias Nägeli, Samuel Oberholzer, Silvan Plüss, Javier Alonso-Mora, and Otmar Hilliges. Flycon: real-time environment-independent multi-view human pose estimation with aerial vehicles. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7025–7034, 2017a.

Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6988–6997, 2017b.

Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 459–468, 2018.

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

Aleksis Pirinen, Erik Gärtner, and Cristian Sminchisescu. Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction. *Advances in Neural Information Processing Systems*, 32, 2019.

Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4342–4351, 2019.

Weichao Qiu, Fangwei Zhong, Yi Zhang, Siyuan Qiao, Zihao Xiao, Tae Soo Kim, and Yizhou Wang. Unrealcv: Virtual worlds for computer vision. In *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1221–1224, 2017.

Faisal Qureshi and Demetri Terzopoulos. Smart camera networks in virtual reality. In *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 87–94, 2007. doi: 10.1109/ICDSC.2007.4357510.

Anatol Rapoport. *N-person game theory : concepts and applications / Anatol Rapoport*. University of Michigan Press Ann Arbor, 1970. ISBN 0472050176.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304, 2018.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

Nitin Saini, Eric Price, Rahul Tallamraju, Raffi Enficiaud, Roman Ludwig, Igor Martinovic, Aamir Ahmad, and Michael J Black. Markerless outdoor human motion capture using multiple autonomous micro aerial vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 823–832, 2019.

Nitin Saini, Elia Bonetto, Eric Price, Aamir Ahmad, and Michael J Black. Airpose: Multi-view fusion network for aerial 3d human pose and shape estimation. *arXiv preprint arXiv:2201.08093*, 2022.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Anil Sharma, Saket Anand, and Sanjit K Kaul. Intelligent camera selection decisions for target tracking in a camera network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3388–3397, January 2022.

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*, pp. 5693–5703, 2019.

Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision*, pp. 529–545, 2018.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Rahul Tallamraju, Eric Price, Roman Ludwig, Kamalakar Karlapalem, Heinrich H Bülthoff, Michael J Black, and Aamir Ahmad. Active perception based formation control for multiple aerial vehicles. *IEEE Robotics and Automation Letters*, 4(4):4491–4498, 2019.

Rahul Tallamraju, Nitin Saini, Elia Bonetto, Michael Pabst, Yu Tang Liu, Michael J Black, and Aamir Ahmad. Aircaprl: autonomous aerial human motion capture using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6678–6685, 2020.

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*, pp. 197–212, 2020.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, Apr 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6220. URL http://dx.doi.org/10.1609/aaai.v34i05.6220.

Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=M3tw78MH1Bk.

Zhou Wenhong, LI Jie, LIU Zhihong, and SHEN Lincheng. Improving multi-target cooperative tracking guidance for uav swarms using multi-agent reinforcement learning. *Chinese Journal of Aeronautics*, 35(7):100–112, 2022.

Jing Xu, Fangwei Zhong, and Yizhou Wang. Learning multi-agent coordination for enhancing target coverage in directional sensor networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 10053–10064, 2020.

Lan Xu, Yebin Liu, Wei Cheng, Kaiwen Guo, Guyue Zhou, Qionghai Dai, and Lu Fang. Flycap: Markerless motion capture using multiple autonomous flying cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24(8):2284–2297, 2017.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M Bayen, and Yi Wu. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In *European Conference on Computer Vision*, pp. 507–523, 2020.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5157–5166, 2018.

Xiaowei Zhou, Sikang Liu, Georgios Pavlakos, Vijay Kumar, and Kostas Daniilidis. Human motion capture using a drone. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2027–2033, 2018.

A TRAINING ALGORITHM PSUEDOCODE

Algorithm 1 Learning Multi-Camera Collaboration (CTCR + WDL)

Initialize: n agents with a tied-weights MAPPO policy π and mixture density models (MDN) for WDL prediction models $\{(P_{\text{self}}, P_{\text{reward}}, P_{\text{peer}}, P_{\text{tgt}}, P_{\text{pd}})\}_{\pi}$, $\mathcal{E}$ parallel environment rollouts
for Iteration $= 1, 2, \ldots M$ **do**
  In each $\mathcal{E}$ environment rollouts, each agent $i \in [\![n]\!]$ collects trajectory with length T:

$$\boldsymbol{\tau} = \left[(\tilde{o}_i^t, \tilde{o}_i^{t+1}, a_i^t, r_i^t, h_i^{t-1}, s^t, \boldsymbol{a}^{t-1}, r^t)\right]_{t=1}^{T}$$

  Substitute individual reward of each agent r^t with CTCR: $r_i^t \leftarrow \text{CTCR}(i)$ (equation 6)
  For each step in $\boldsymbol{\tau}$, compute advantage estimates $\hat{A}_i^1, \ldots, \hat{A}_i^T$ with GAE (Schulman et al., 2015) for each agent $i \in [\![n]\!]$
  Yield a training batch $\mathcal{D}$ of the size $\mathcal{E} \times T \times n$
  for Mini-batch SGD Epoch $= 1, 2, \ldots, K$ **do**
    Sample a stochastic mini-batch of size B form $\mathcal{D}$, which $B = |\mathcal{D}|/K$
    Compute z_i^t and h_i^t using the encoder model E_π
    Compute PPO-CLIP objective loss $\mathcal{L}_{\text{PPO}}$, global critic value loss $\mathcal{L}_{\text{Value}}$, adaptive KL loss $\mathcal{L}_{\text{KL}}$
    Compute the objectives that constitutes $\mathcal{L}_{\text{WDL}}$:
      - Self-State Prediction Loss $\mathcal{L}_{\text{self}} = -\mathbb{E}_{\boldsymbol{\tau}}[\log P(\xi_i^{t+1}|z_i^t, h_i^t, a_i^t)]$
      - Reward Prediction Loss $\mathcal{L}_{\text{reward}} = -\mathbb{E}_{\boldsymbol{\tau}}[\log P(r^t|z_i^t, h_i^t, a_i^t)]$
      - Peer-State Prediction Loss $\mathcal{L}_{\text{peer}} = -\mathbb{E}_{\boldsymbol{\tau}}[\log P(\xi_{-\boldsymbol{i}}^{t+1}|z_i^t, h_i^t, a_i^t)]$
      - Target Prediction Loss $\mathcal{L}_{\text{tgt}} = -\mathbb{E}_{\boldsymbol{\tau}}[\log P(p_{\text{tgt}}^{t+1}|z_i^t, h_i^t, p_{\text{tgt}}^t)]$
      - Pedestrians Prediction Loss $\mathcal{L}_{\text{pd}} = -\mathbb{E}_{\boldsymbol{\tau}}[\log P(p_{\text{pd}}^{t+1}|z_i^t, h_i^t, p_{\text{pd}}^t)]$
    $\mathcal{L}_{\text{Train}} = \lambda_{\text{PPO}}\mathcal{L}_{\text{PPO}} + \lambda_{\text{Value}}\mathcal{L}_{\text{Value}} + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{WDL}}\mathcal{L}_{\text{WDL}}$
    Optimize $\mathcal{L}_{\text{Train}}$ w.r.t to the current policy parameter θ_π

B UNREALPOSE: ACCESSORIES AND MISCELLANEOUS ITEMS

Our UnrealPose virtual environment supports different active vision tasks, such as active human pose estimation and active tracking. This environment also supports various settings ranging from single-target single-camera settings to multi-target multi-camera settings. Here we provide a more detailed description of the three key characteristics of UnrealPose:

Realistic The built-in navigation system governs the collision avoidance movements of virtual humans against dynamic obstacles. In the meantime, it also ensures diverse generations of walking trajectories. These features enable the users to simulate a realistic-looking crowd exhibiting socially acceptable behaviors. We have also provided several pre-set scenes, *e.g.*, school gym, wilderness, urban crossing, and *etc*. These scenes have notable differences in illuminations, terrains, and crowd appearances to reflect the dramatically different-looking scenarios in real-life.

Extensive Configuration The environment can be configured with different numbers of humans and cameras and swapped across other scenarios with ease, which we demonstrated in Fig 4(a-d). Besides simulating walking human crowds, the environment incorporates over 100 Mocap action sequences with smooth animation interpolations to enrich the data variety for other MoCap tasks.

RL-Ready We use the UnrealCV (Qiu et al., 2017) plugin as the medium to acquire images and annotations from the environment. The original UnrealCV plugin suffers from unstable data transfer and unexpected disconnections under high CPU workloads. To ensure fast and reliable data acquisition for large-scale MARL experiments, we overhauled the communication module in the UnrealCV plugin with inter-process communication (IPC) mechanism, which eliminates the aforementioned instabilities.

B.1 VISUALIZATION TOOL

We provide a visualization tool to facilitate per-frame analysis of the learned policy and reconstruction results (shown in Fig. 9). The main interface consists of four parts: (1) Live 2D views from all cameras. (2) 3D spatial view of camera positions and reconstructions. (3) Plot of statistics. (4) Frame

control bar. This visualization tool supports different numbers of humans and cameras. Meanwhile, it is written in Python to support easy customization.

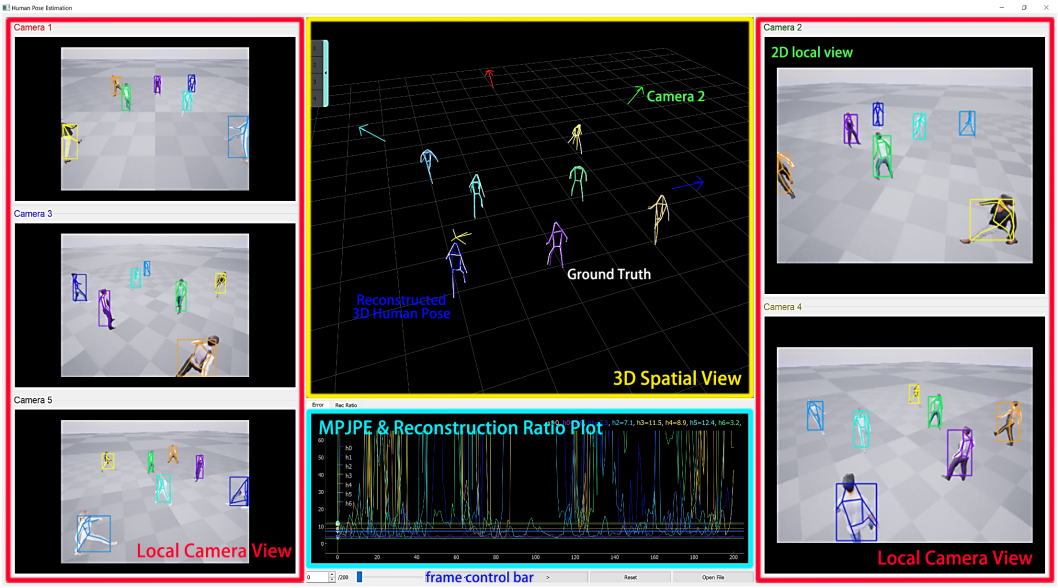


Figure 9: Visualization Tool. RGB videos at the two sides show the local camera view for each individual camera. The center part shows the live 3D reconstructions and corresponding MPJPE.

B.2 LICENSE

All assets used in the environment are commercially-available and obtained from the UE4 Marketplace. The environment and tools developed in this work are licensed under Apache License 2.0.

C OBSERVATION PROCESSING

Fig.10 shows the pipeline of the observation processing. Each camera observes an RGB image and detects the 2D human poses and IDs via the Perception Module described in the main paper. The camera pose, 2D human poses and IDs are then broadcast to other cameras for multi-view 3D triangulation. The human position is calculated as the median of the reconstructed joints. Human orientation is calculated from the cross product of the reconstructed shoulder and spine.

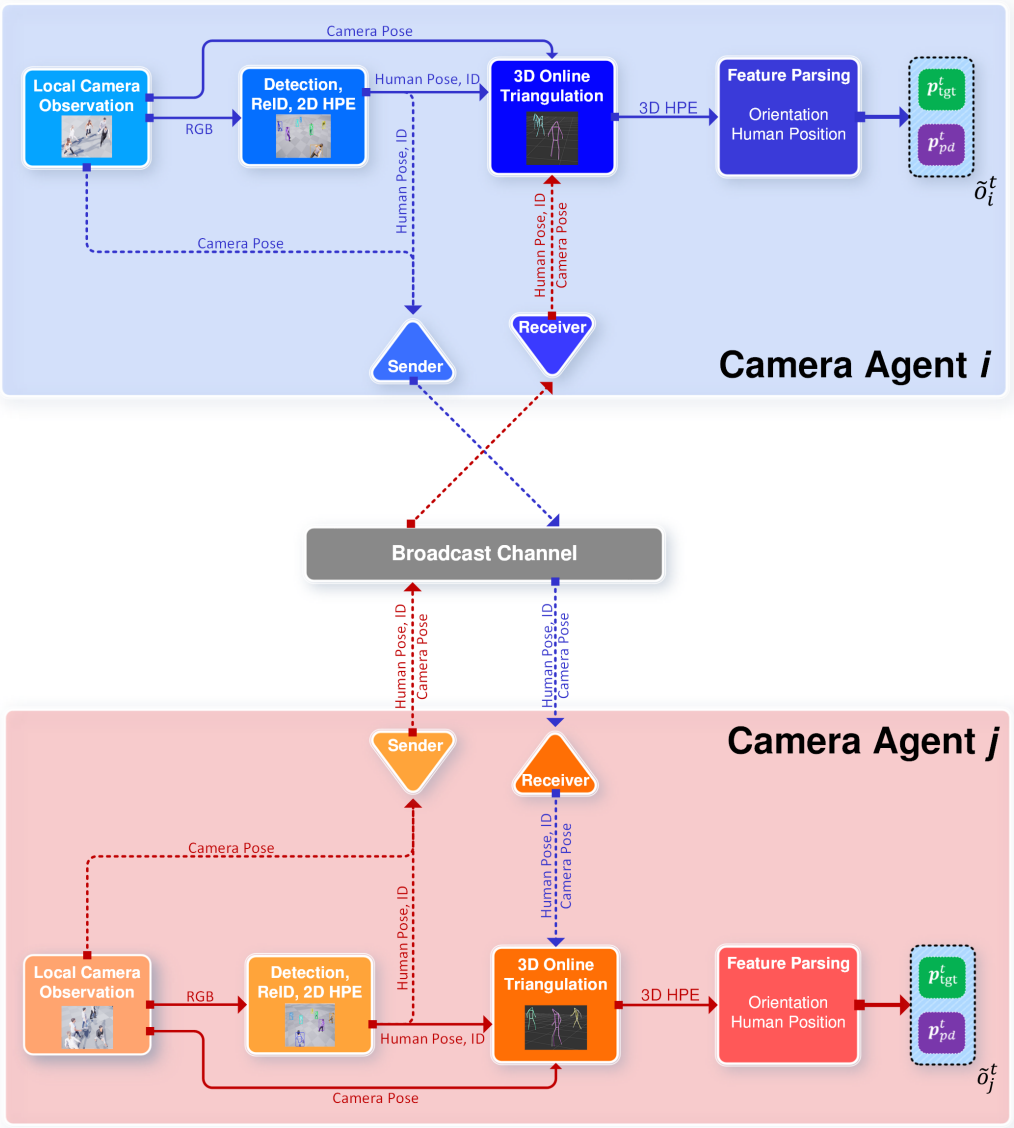


Figure 10: Decentralized Observation Processing

D IMPLEMENTATION DETAILS

D.1 TRAINING DETAILS

All control policies are trained in the `BlackEnv` scene. At the testing stage, we apply zero-shot transfer for the learned policies to three realistic scenes: `SchoolGym`, `UrbanStreet`, and `Wilderness`.

To simulate a dynamic human crowd that performs high random behaviors, we sample arbitrary goals for each human and employ the built-in navigation system to generate collision-free trajectories. Each human walks at a random speed. To ensure the generalization across a different number of humans, we train our RL policy with a mixture of environments of 1 to 6 humans. The learning rate is set to 5×10^{-4} with scheduled decay during the training phase. The annealing schedule for the learning rate is detailed in Table. 2. The maximum episode length is 500 steps, discounted factor γ is 0.99 and the GAE horizon is 25 steps. Each sampling iteration produces a training batch with the size of $1,500$ steps, then we perform 16 iterations on every training batch with 4 SGD mini-batch updates for each iteration.

Table. 2 shows the common training hyper-parameters shared between the baseline models (MAPPO) and all of our methods. Table. 4 shows the hyperparameters for the WDL module.

Table 2: Common hyperparameters for models in baselines and our methods

Common Hyperparameter	**Baseline, Our Methods**
Max Episode Length	500
Rollout Fragment Length	25
Number of Rollouts	60
Train Batch Size	1500
SGD Mini-batch Size	375
Number of SGD Iterations (Sample Reuse)	16
gamma (γ)	0.99
GAE lambda (λ)	1.0
Initial KL Coefficient	0.2
$\text{KL}_{\text{Target}}$	0.01
Entropy Regulation Coefficient	0.0
Value Function Loss Coefficient	0.1
Value Function Clipping	1000
Gradient Clipping	50
Initial Learning Rate	0.0005
Learning Rate Schedule (lr, end step)	[(0, 5E-4), (200E3, 5E-4), (200E3, 1E-4), (400E3, 1E-4), (600E3, 5E-5), (600E3, 5E-5)]
LSTM Cell Size	128
Encoder Hidden Layers	[128, 128, 128]
Value Network Hidden Layers	[128]
Actor Network Hidden Layers	[128]

D.2 DIMENSIONS OF FEATURE TENSORS IN THE CONTROLLER MODULE

Table 3 serves as a complementary description for Eqn. 1, 2, 3, 4. The table shows the dimensions of the feature tensors used in the controller module."B" denotes the batch size. In the current

model design, the dimension of local observation is adjusted based on the maximum number of camera agents (N_cam_{max}) and the maximum number of observable humans (N_human_{max}) in an environment. In our experiments, N_human_{max} has been set to 7. The observation pre-processor will zero-pad each observation to a length equal $N_human_{\text{max}} \times 18$ if the current environment instance has less than N_human_{max} humans. "9" and "18" correspond to the feature length for a camera and a human, respectively. The MDN of the Target Prediction module has 16 Gaussian components, in which each component outputs $(\phi, \mu_x, \sigma_x, \mu_y, \sigma_y)$ and ϕ is the weight parameter of a component. The current implementation of MDN only predicts the x-y location of a human, which is a simplification since the z coordinate of a simulated human barely changes across an episode compared to the x and y coordinates. The dimension of an MDN output has a length of 80 and the exact prediction is produced by ϕ-weighted averaging. In Eqn. 4, $\{(\phi, \mu, \sigma)\}_{\text{MDN}_{\text{tgt}}}$ is an encoded feature produced by passing the MDN output to a 2-layer MLP that has an output dimension of 128.

Table 3: Dimensions of the feature tensors used in the controller module

Feature	**Shape**
$\hat{o}_i$	(B, $N_cam_{\text{max}} \times 9 + N_human_{\text{max}} \times 18$)
z_i	(B, 128)
h_i	(B, 128)
$\hat{p}_{\text{tgt}}, \hat{p}_{\text{pd}}$	(B, 128)
$\{(\phi, \mu, \sigma)\}_{\text{MDN}_{\text{tgt}}}$	(B, 128)
e_i	(B, 384)

D.3 Computational Resources

We used 15 Ray workers for each experiment to ensure consistency in the training procedure. Each worker carries a Gym vectorized environment consisting of 4 actual environment instances. Each worker demands approximately 3.7 GB of VRAM. We run each experiment with 8 NVIDIA RTX 2080 Ti GPUs. Depending on the number of camera agents, the total training hours required for an experiment to run 500k steps will vary between 4 to 12 hours.

Table 4: Hyperparamters for WDL

WDL Hyperparameter	**Our Methods**
Prediction Loss Coefficient (λ_{WDL})	1.0
Camera Prediction Coefficient	1.0
Other Camera Prediction Coefficient	1.0
Reward Prediction Coefficient	1.0
Human Prediction Coefficient	1.0
Pedestrian Prediction Coefficient	0.1
MDN: Hidden Layers	[128, 128]
MDN: Number of Gaussian Distributions	16

D.4 Training Curve

Fig. 13 shows the training curves of our method and the baseline method. We can find that our methods converge faster and improve reconstruction accuracy than the baseline method.

D.5 Total Inference Time

Our solution can run in real-time. Table 5 reports the inference time for each module of the proposed Active3DPose pipeline.

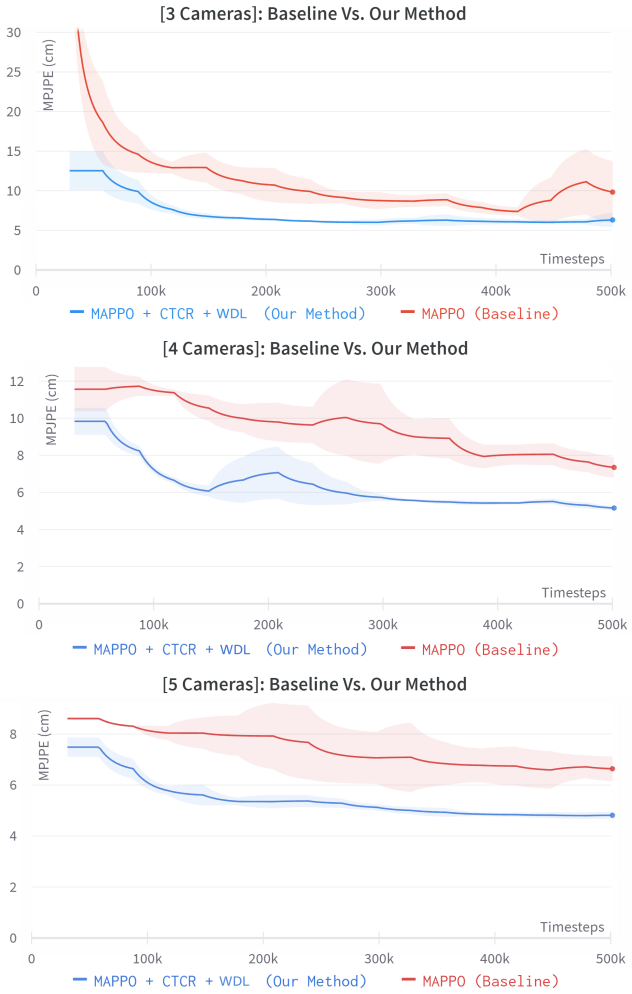


Figure 11: Comparing the training curves (MPJPE (cm) vs. Environment Steps) for **3, 4, 5** cameras policies trained to 500k timesteps for the baseline and our method. Intervals for one standard deviation are shown, and each interval is computed based on samples from 3 different runs on different seeds.

Table 5: The total inference speed of our current implementation is around 8 FPS. The average time data are generated on a computing node with one NVIDIA RTX 2080TI GPU and one Intel Xeon E5-2699A CPU. These data are from testing our 5-camera model over 2500 environment steps in the Blank Environment.

Average Time (ms)	**MPJPE (mm)**
Human Detection	20.05
ReID Module	28.88
2D Pose Estimation	31.16
3D Triangulation	1.43
Observation Processing	3.97
RL Model Inference	40.76
Total Inference Time	126.25

E ADDITIONAL EXPERIMENT RESULTS

E.1 BASELINE — FIXED-CAMERAS

In addition to the triangulation and RANSAC baselines introduced in the main text, we compare and elaborate on two more baselines that use fixed cameras: (1) fixed cameras with RANSAC-based triangulation and temporal smoothing (TS) (2) an off-shelf 3D pose estimator PlaneSweepPose (Lin & Lee, 2021a).

In the temporal smoothing baseline, we applied a low-pass filter (Casiez et al., 2012) and temporal fusion where the algorithm will fill in any missing keypoints in the current frame with the detected keypoints from the last frame.

In the PlaneSweep baseline, as per the official instructions, we train three separate models (3 cams to 5 cams) with the same camera setup as in our testing scenarios. We have tested the trained models in different scenarios and reported the MPJPE results in Table 6, 7, 8 and 9. Note that, this off-shelf pose estimator performs better than the Fixed-Camera Baseline (Triangulation) but still underperforms compared to our active method. Fig. 12 illustrates the formations of the fixed camera baselines.

Table 6: Four fixed-camera baselines compared to our method in the 6-human Blank environment. Results for all baselines are evaluated over 20 episodes of 500 steps. **RANSAC:** RANSAC-based Triangulation **TS:** Temporal Smoothing with temporal fusion **PlaneSweep:** PlaneSweepPose (Lin & Lee, 2021a).

BlankEnv	Fixed Cameras (Triangulation)	Fixed Cameras (RANSAC)	Fixed Cameras (RANSAC+TS)	Fixed Cameras (PlaneSweep)	Ours
3 Cameras	142.0	129.0	123.5	123.3	**73.3**
4 Cameras	122.3	116.6	106.0	116.4	**54.7**
5 Cameras	87.0	72.3	77.7	86.5	**51.6**

Table 7: Four fixed-camera baselines compared to our method in the 6-human SchoolGym environment. Results for all baselines are evaluated over 20 episodes of 500 steps. **RANSAC:** RANSAC-based Triangulation **TS:** Temporal Smoothing with temporal fusion **PlaneSweep:** PlaneSweep-Pose (Lin & Lee, 2021a).

SchoolGym	Fixed Cameras (Triangulation)	Fixed Cameras (RANSAC)	Fixed Cameras (RANSAC+TS)	Fixed Cameras (PlaneSweep)	Ours
3 Cameras	149.1	149.8	144.1	132.6	**99.4**
4 Cameras	133.3	124.5	124.7	120.3	**67.7**
5 Cameras	100.2	76.8	83.0	92.8	**64.6**

Table 8: Four fixed-camera baselines compared to our method in the 6-human UrbanStreet environment. Results for all baselines are evaluated over 5 episodes of 500 steps. **RANSAC:** RANSAC-based Triangulation **TS:** Temporal Smoothing with temporal fusion **PlaneSweep:** PlaneSweepPose (Lin & Lee, 2021a).

UrbanStreet	Fixed Cameras (Triangulation)	Fixed Cameras (RANSAC)	Fixed Cameras (RANSAC+TS)	Fixed Cameras (PlaneSweep)	Ours
3 Cameras	513.4	210.1	194.2	160.3	**103.6**
4 Cameras	279.4	190.8	174.5	154.5	**66.4**
5 Cameras	136.7	94.1	94.4	106.5	**61.9**

Table 9: Four fixed-camera baselines compared to our method in the 6-human Wilderness environment. Results for all baselines are evaluated over 20 episodes of 500 steps. **RANSAC:** RANSAC-based Triangulation **TS:** Temporal Smoothing with temporal fusion **PlaneSweep:** PlaneSweepPose (Lin & Lee, 2021a).

Wilderness	Fixed Cameras (Triangulation)	Fixed Cameras (RANSAC)	Fixed Cameras (RANSAC+TS)	Fixed Cameras (PlaneSweep)	Ours
3 Cameras	234.0	159.6	160.5	139.2	**72.6**
4 Cameras	165.6	143.9	146.6	129.5	**71.3**
5 Cameras	108.1	74.9	86.5	95.1	**54.3**

Camera placements for all fixed-camera baselines are shown in Fig. 12. These formations are carefully designed not to disadvantage the fixed-camera baselines on-purpose. Especially for the 5-cameras pentagon formation, which helps the fixed-camera baseline to obtain satisfactory performance on the 5-camera setting as shown in Tables 6,7,8 and 9.

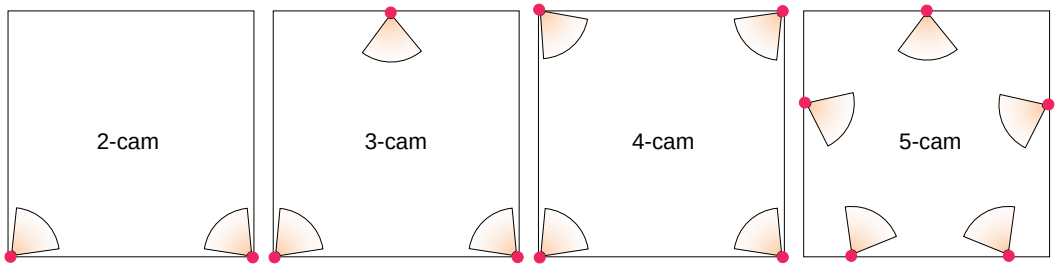
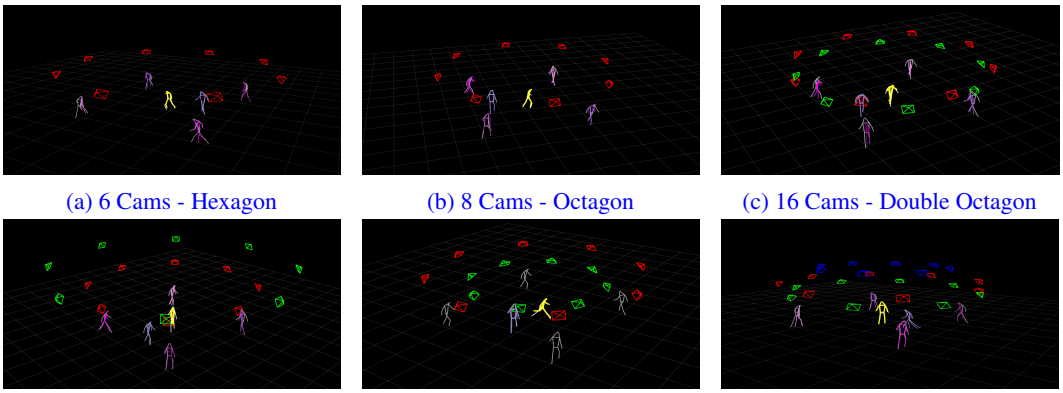


Figure 12: Illustration of fixed camera placements



(a) 6 Cams - Hexagon (b) 8 Cams - Octagon (c) 16 Cams - Double Octagon

(d) 16 Cams - Hexadecagon (e) 16 Cams - Double Octagon (OH) (f) 24 Cams - "Xmas Tree"

Figure 13: In search of a topology of a fixed-camera system that could outperform our 5-camera model, we have designed six more 3D topologies that contain 8 to 24 cameras.

Table 10: Comparing six more fixed-camera network topologies with our 5-camera method. Although these topologies have significantly more cameras than our 5-active-camera model, the fixed-camera approach cannot close the performance gap between fixed-camera systems and the learned policy. We also observe inversions in MPJPE when increasing the number of cameras from 8 to 16 to 24. These results suggest that the fixed-camera system has saturated the reconstruction accuracy beyond 8 cameras, and there are no further upsides to increasing the scale. We reason this is due to the errors and biases in the 2D pose model that subsequently affect the triangulation algorithm's effectiveness.

	MPJPE (mm)
6 Cams - Hexagon	88.0
8 Cams - Octagon	78.5
16 Cams - Double Octagon	86.7
16 Cams - Hexadecagon	81.7
16 Cams - Double Octagon (OH)	94.9
24 Cams - "Xmas Tree"	87.0
Ours, 5 Cams, active cameras	51.6

E.2 Our Method Enhanced with RANSAC-based Triangulation

RANSAC is a generic technique that can be used to improve triangulation performance. In this experiment, we also train and test our model with RANSAC. The final result (Table 11) shows a further improvement on our original triangulation version.

Table 11: Evaluate the effect of adding RANSAC-based Triangulation to our method. Both models are trained from scratch on a mix of Blank environment instances with 1 to 6 humans. MPJPE(mm) is averaged over the training results in six different configurations of Blank environment instances (1-6 humans).

	Ours (5 Cams)	**Ours (5 Cams + RANSAC)**
MPJPE (mm)↓	46.7	**35.0**

F Generating Safe and Smooth Trajectories

F.1 Collision Avoidance

In order to generate safe trajectories, in this section, we introduce and evaluate two different ways to enforce collision avoidance between cameras and humans.

Obstacle Collision Avoidance (OCA) OCA resembles a feed-forward PID controller on the final control outputs before execution. Concretely, OCA adds a constant reverse quantity to the control output if it detects any surrounding objects within its safety range. This "detouring" mechanism safeguards the cameras from possible collisions and prevents them from making dangerous maneuvers.

Action-Masking (AM) AM also resembles a feed-forward controller but is instead embedded into the forward step of the deep-learning model. At each step, AM module first identifies the dangerous actions among all possible actions, then modifies the probabilities (output by the policy model) of choosing the hazardous actions to be zero so that the learning-based control policy will never pick them. Note that AM must be trained with the MARL policy model.

We proposed the minimum Camera-Human distance (in which the scope includes the target and the pedestrians) as the safety metric. It measures the distance between the closest human and the camera at a timestep. Fig. 14 shows the histograms of Min Camera-Human distance sampled over five episodes of 500 steps.

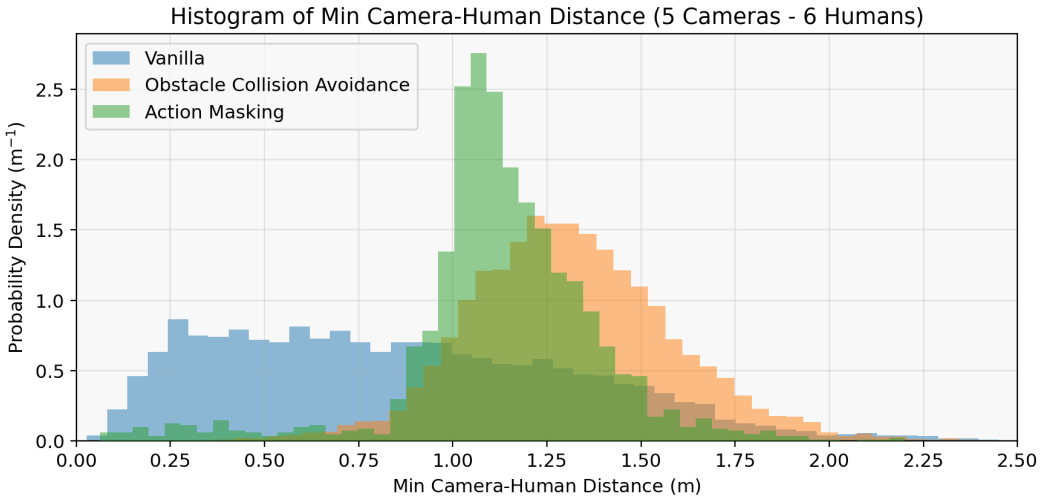


Figure 14: Histograms of minimum Camera-to-Human distances to demonstrate the safety integrity level for three different **5-Cameras** methods. **Vanilla**: original deep MARL policy without any safety feature. **Obstacle Collision Avoidance**: deep MARL policy added with OCA module **Action Masking**: deep MARL policy trained with AM module.

F.2 Trajectory Smoothing

Exponential Moving Average (EMA) To generate smooth trajectories, we introduce EMA to smooth the outputs of our learned policy model. EMA is a common technique used in smoothing time-series data. In our case, the smoothing operator is defined as :

$$\hat{a}_t = \hat{a}_{t-1} + \eta \cdot (a_t - \hat{a}_{t-1})$$

Where a_t is the action (or control signal) output by the model at the current step, $\hat{a}_{t-1}$ is the smoothed action from the last step and η is the smoothing factor. A smaller η results in greater smoothness. $\hat{a}_t$ is the smoothed action that the camera will execute.

F.3 Robustness of the Learned Policy

In this section, we evaluate the robustness of our model on different external perturbations to the control signal. In conclusion, our model shows resilience against the delay effect and random noise.

Delay EMA also brings a delay effect while smoothing the generated trajectory. The level of smoothing positively correlates with a larger delay factor. Here, we evaluate our model's robustness to the EMA simulated delay.

Random Action Noise Control devices in real-life are inevitably affected by errors. For example, the control signal of a controller may be over-damped or may overshoot. We simulated this type of random error by multiplying the output action by a uniformly-sampled noise.

In Table 12, we intend to observe the effects of EMA delay and random action noise on the reconstruction accuracy of our model, marked as "Vanilla".

Table 12: Adding delay (EMA) and action noise to our proposed method. In this experiment, we set $\eta = 0.50$ and sample noise factor between $[0.80, 1.20)$. Results are reported on 5 episodes of 500 steps.

	Δ MPJPE (mm)
Ours(Vanilla)	–
Add delay(EMA)	+0.8
Add action noise	+2.1
Add action noise and delay(EMA)	+3.9

G More Explanations on CTCR

Figure 3 is an example of how to plug numbers into Eq. 6 to compute CTCR for each of the three cameras. There is a breakdown of Eq. 6 just below it. But just in case if it seems too obscure, here is a more intuitive description of Eq. 6. The CTCR is incentivized by the Shapley Value, so the main idea is that the overall optimality needs to also account for the optimality of every possible sub-formation. In the context of an active HPE task, for a camera agent to receive the highest CTCR possible, its current position and view must be optimal both in terms of its current formation and any sub-formation possible.

Note: Game theory folks like to call a group of collaborating players a "coalition". Here we apply the same concept but to a group of cameras, so we used the more initiative term "formation" instead.

Eq. 6 can be further breakdown as follows:

$$\begin{aligned}
\varphi_r(i) &= \sum_{S \subseteq [n] \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{n!} [r(S \cup \{i\}) - r(S)] \\
&= \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \frac{|S|!(n-|S|-1)!}{(n-1)!} [r(S \cup \{i\}) - r(S)] \\
&= \frac{1}{n} \sum_{S \subseteq [n] \setminus \{i\}} \binom{n-1}{|S|}^{-1} [r(S \cup \{i\}) - r(S)]
\end{aligned}$$

where $[n] = 1, 2, \ldots, n$ denotes the set of all cameras and the binomial coefficient $\binom{k}{n} = \frac{n!}{k!(n-k)!}, 0 \leq k \leq n$. S denotes a formation (a subset) without camera i. $\binom{n-1}{|S|}$ is the number of combinations of subset S (i.e., the binomial coefficient), which serves as a normalization term. $r(S)$ computes the reconstruction accuracy of the formation S. And $[r(S \cup \{i\}) - r(S)]$ computes the marginal improvement after adding the camera i to sub-formation S. So this equation means we iterate over all possible S and compute the marginal contribution of camera i and average over all possible combinations of (S, i).

Suppose we have a 3-cameras formation, similarly shown in Figure 3. So $n = 3$, the number of cameras. Let's name these cameras (1,2,3), and let's say we only care about Camera 1 for now. Since we are computing the average marginal contribution for Camera 1, we are looking for those formations that do not have Camera 1 because we want to see how much of an increase in performance resulted from the addition of Camera 1 to those formations. For all possible formations denoted by the set $[n]$, four formations satisfy this condition, $S \subseteq [n] \setminus \{i\} \longrightarrow S \in (\emptyset, 2, 3, 23)$.

The binomial coefficient $\binom{n-1}{|S|}$ for a 2-camera sub-formation in a 3-cameras case is $\binom{2}{2}$, which makes sense because there exists only one unique combination that does not contain Camera 1, which is sub-formation 23. $r(23)$ computes the reconstruction accuracy of the formation 23 and $r(23 \cup \{1\})$ computes the reconstruction accuracy after adding Camera 1 to sub-formation 23. Their difference gives us the marginal contribution of Camera 1. As we sum over all subsets S of $[n]$ not containing Camera 1, and then divide by $\binom{n-1}{|S|}$ and the number of cameras n, we have the average marginal

contribution of Camera 1 ($\varphi_r(1)$) to the collaborative triangulated reconstruction. Further multiplying this term by n, you have the $CTCR_1$ as shown in Eq. 6.

H More Qualitative Analysis on the proposed methods

Here in Figure 15 we provide statistics and analysis on the behaviour mode of the agents controlled by our 3-, 4- and 5-camera policies, respectively. We are interested in understanding the characteristics of the emergent formations learned by our model. Hence we proposed three quantitative measures to understand the topology of the emergent formations: (1) the min-angle between the cameras' orientation and the per-frame-mean of the min-camera angle, (2) the camera's pitch angle, and (3) the camera-human distance. Hence we provide rigorous definitions as follows:

$$\text{min-camera angle}(i) = \min_{j \neq i} \langle \text{axis of camera } i, \text{axis of camera } j \rangle$$

$$\text{per-frame-mean of min-camera angle} = \frac{1}{n} \sum_{i \in [n]} \text{min-camera angle}(i)$$

In simpler terms, $\text{min-camera angle}(i)$ finds the minimum angle between camera i and any other camera j. "per-frame-mean of min-camera angle" is the mean of $\text{min-camera angle}(i)$ for all camera i in one frame. A positive camera's pitch angle means looking upward, and a negative camera's pitch angle means looking downward. The camera-human distance measures the distance between the target human and the given camera.

Regarding the distance between cameras and humans, the camera agents actively adjust their position relative to the target human. The cameras learn to keep a camera-human distance between 2m 3m. It is neither too distant from the target human nor too close, violating the safety constraint. The camera agents surround the target at an appropriate distance to have a better resolution on the 2D views. In the meantime, as the safe-distance constraint is enforced during training, the camera agents are prohibited from aggressively minimizing the camera-human distance.

The histograms for the camera's pitch angle suggest that the cameras mostly maintain negative pitch angles. Their preferred strategy is to hover over the humans and capture images at a higher altitude. This is likely because of emergent occlusion avoidance. The cameras also emerge to fly at an even level with the humans (where the pitch angle approximately equals 0) to capture more accurate 2D poses of the target human. This propensity is apparent from the peaks at x=0 angles in the histograms. A relatively wide distribution of the pitch angle histogram suggests that the camera formation is spread out in the 3D space and dynamic adjustments of flying heights and pitch angles by the camera agents.

For the average angle between the cameras' orientations, this statistic shows that the cameras in various formations will maintain reasonable non-zero spatial angles between each other. Therefore, their camera views are less likely to coincide and provide more diverse perspectives to generate a more reliable 3D triangulation.

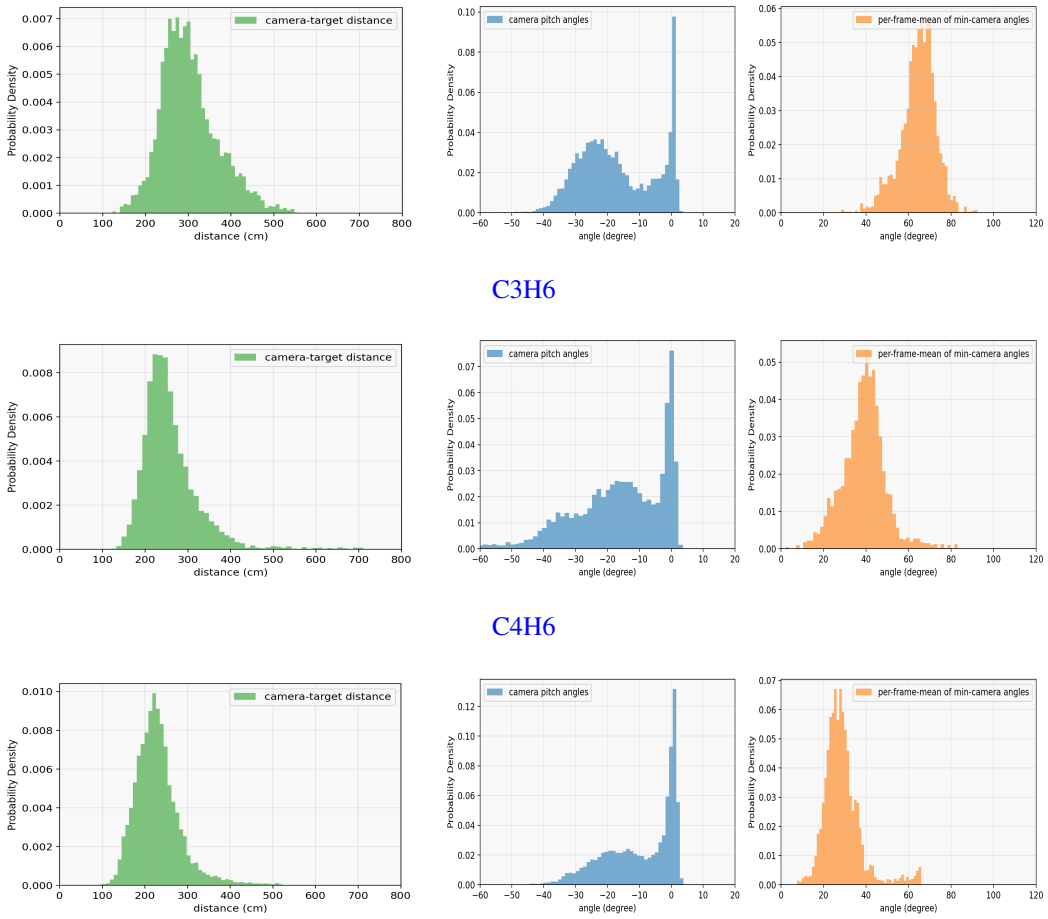


Figure 15: Histograms of the three metrics for 3-, 4- and 5-camera methods. Each row from left to right: Camera-target distance, camera-pitch-angle, per-frame-mean of min-camera angle,