We thank the reviewers for their valuable feedback and note that all reviewers agree about the novelty of our work.

AR1: *Results for more than 10 cameras:* Triangulation converges when adding more cameras, thus narrowing ACTOR's improvement over the baselines. Using many cameras however requires significant computation, hence we are interested in finding small yet informative subsets of cameras that yield good results. We chose to run the model for 1-10 cameras as evaluating the Oracle is slow for large camera sets, cf. Fig. 3 right. Results for 15 cameras on multi-people data: ACTOR $= 69.99$, Random $= 74.71$, Max-Azim $= 73.79$, Oracle $= 57.24$ (mm/joint hip-translated). Compared to running for 10 cameras, ACTOR keeps its gain over Max-Azim and significantly closes in on the Oracle, see Table 1.

Minor issues: (1) Fig. 5 is not missing; see page 8 and line 240. (2) We will clarify line $\sim$ 223-224 in the paper.

AR2: *Runtimes and baseline comparisons:* Fig. 3-4 right show runtimes; OpenPose adds 0.134 s/image and our policy adds 0.005 s/image (25x faster), see line 237-240. ACTOR is almost as fast as the baselines while *significantly more accurate* when considering fewer views, e.g. 94 mm/joint (34 %) better than Max-Azim at 3 views (Table 1), yet only 6% slower (Fig. 3 right). When using more cameras triangulation converges, decreasing the need for intelligent camera selection. The Oracle cheats by using 3d ground-truth when selecting views. It is greedy, as exhaustively evaluating the exponential set of views is infeasible. It is still very slow since it evaluates all cameras before selecting the next view.

Dataset description: We use the HD cameras, of which there are about 30 per scene. The HD cameras are used since they provide better image quality than VGA and are sufficiently dense, yet spread apart far enough to make each viewpoint unique. We will include this in an extended dataset description for the camera-ready.

More realistic setup: To mimic a single camera setup, we experimented on multi-people data assuming time elapses 0.5 s between *every* selected view. Note that some joints which move drastically between views cannot be triangulated. Therefore we extend ACTOR with a monocular 3d pose estimator (DMHS, Popa et al., CVPR, 2017) as a fallback. *Without refining the policy* our results (in auto-mode) are: ACTOR $= 123.3$, Random $= 138.7$, Max-Azim $= 150.1$, Oracle $= 106.1$ (mm/joint hip-translated). ACTOR outperforms the baselines also in this setting (cf. Table 1). While these are only proof-of-concept results they clearly indicate the potential of our model for single camera setups.

State-of-the-art: Our results are significantly better than state-of-the-art 3d pose estimators such as MubyNet (Zanfir et al., NeurIPS, 2018). Their errors are about 150 mm/joint on multi-people data, while ACTOR obtains 96 mm/joint in auto-mode, see Table 1. There exists no prior work on active 3d human pose estimation, so results are hard to compare as they are framed in completely different settings (monocular 3d estimation vs active 2d-to-3d triangulation).

AR3: *No annotations needed for training:* We emphasize that our reward is *not* based on ground-truth visibility – it relies solely on the automatic 2d pose (body joint) detection. Thus our approach requires no annotations in training.

Next-best-view (NBV) and object pose estimation: NBV is a very general concept, and our formulation can be seen as one way of modeling and implementing it in the framework of deep reinforcement learning. We further address the novel problem of intelligent view selection for 3d human pose reconstruction (or generally articulated structures), which is more challenging than rigid 3d reconstruction and pose estimation or object detection[1]: i) Dimensionality is much higher than object detection and the problem is more complex than rigid 3d reconstruction - humans are articulated and deformable, scenes contain multiple people; ii) Realistic training data for supervised 3d human pose estimation is scarce, and our proposed methodology requires *no annotations* and instead uses self-supervision to learn the policy; iii) Reconstruction difficulty is impacted by occlusions and the pose of targets, hence our model adaptively selects a *variable number of views* depending on scene complexity; iv) NBV for e.g. object pose estimation is formulated such that, at each step, one selects views by assessing a utility function over a set of candidates. Most design choices are handcrafted, decisions are local. We frame the task as a deep RL problem where the policy is trained to maximize a *global objective*, searching over entire sequences of viewpoints, and by triangulating as many joints as possible.

AR2, AR3: *Practical applications:* We use our setup as a proxy for a moving observer and to develop viewpoint selection strategies for e.g. intelligent holoportation (AR1) while providing a framework for *reproducible* experiments. Practical developments of our methodology would include e.g. real-time intelligent processing of multi-camera (Panoptic/Lightstage) video feeds or control policies for a drone observer. In the latter case the model would further benefit from being extended to account for physical constraints, e.g. a single camera and limited speed. Our paper is a key first step since it presents fundamental methodology required for future applied research.

AR1, AR3: *Why not exhaustive triangulation:* Using all views requires significant computation, even with one of the fastest pose estimators such as OpenPose (0.134 s/image vs only 0.005 s/image for our policy, see line 238-239). Thus exhaustive computation does not suit real-time scenarios (processing 30 cameras takes over 4 s / frame). Also, assuming a physical observer (e.g. a drone), the need for a view selection strategy is crucial since covering all views is infeasible.

[1] Obviously there is no one size fits all – details on the output representation, level of detail, single versus multiple structures, occlusions, observation setup, or desired reward would affect formulation and modeling choices for each problem.