STABLECROWD: SPATIO-TEMPORALLY CONSISTENT FEED-FORWARD CROWD RECONSTRUCTION WITH MULTI-JOINT HEIGHT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing crowd motion from monocular large-scene video requires efficiency and spatio-temporal consistency. Among feed-forward methods, per-person video methods recover each person's motion but not the relative positions of the people, while large-scene crowd methods anchor each person in the shared scene through a point regressed as a pixel distance in a crop, a target that varies with the crop and so jitters over time. On the other hand, existing methods that achieve spatio-temporal consistency rely on expensive per-sequence optimization. We present StableCrowd, a feed-forward method that reconstructs crowd motion from such videos. We introduce a new target, Multi-Joint Height, the metric height of every joint above the scene surface. It is crop-invariant, stays stable over time, and can be lifted in closed form to a metric 3D estimate of the joint. We propose Global Trajectory Fusion, which regresses the trajectory in the same space as these estimates, committing directly to absolute positions in the shared scene and covering a sequence of any length. We also propose Canonical Camera Normalization, which absorbs the position and size of each crop into a camera rotation and a zoom, so that close-range single-person data trains the model for large-scene video. StableCrowd is the first feed-forward method to achieve spatio-temporal consistency from large-scene video, zero-shot on VirtualCrowd and WorldPose, showing that per-sequence optimization is no longer required to reach it. Code and models will be released upon acceptance.

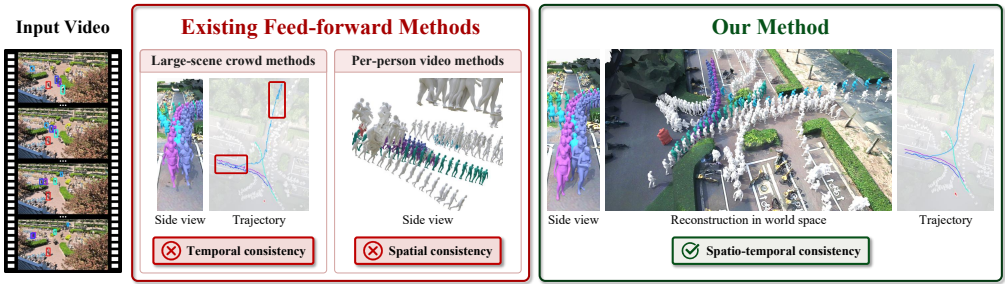

Figure 1: From monocular large-scene video, our method reconstructs the crowd with spatio-temporal consistency, in a feed-forward manner. The trajectories of large-scene crowd methods (Crowd3D (Wen et al., 2023)) jitter from frame to frame (red boxes), and per-person video methods (GEM (Li et al., 2025)) do not recover the relative positions of the people.

1 INTRODUCTION

Large-scene crowd videos capture many interacting individuals across a wide area, and reconstructing them supports applications such as surveillance, crowd behavior analysis, and embodied agent training. Recovering how these people move relative to one another requires reconstructing all of them efficiently and with spatio-temporal consistency: every person should be placed correctly relative to the others, and coherent from frame to frame.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Capabilities of existing methods and ours in a large scene.

Category (representative)	Consistency		Feed-forward
	Spatial	Temporal	
Per-person video			
Per-person depth (TRAM)	✗	✓	✓
Per-person coordinates (GVHMR)	✗	✓	✓
Large-scene crowd (Crowd3D)	✓	✗	✓
Per-sequence optimization (Crowd4D)	✓	✓	✗
Ours	✓	✓	✓

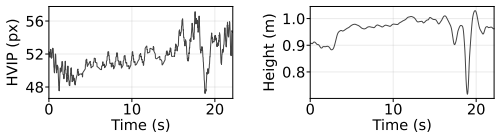

Figure 2: Ground truth of one person over time. Left: the HVIP as parameterized in Crowd3D, a pixel distance in the crop, jitters. Right: the root's height above the scene surface changes only with the person's motion.

Efficiency motivates feed-forward regression. Whole-image methods (Su et al., 2025; Baradel et al., 2024) regress all people at a fixed input resolution, which in a large scene leaves too few pixels per person, so cropping is necessary. Per-person video methods recover coherent motion for each person, but not the relative positions of the people (Fig. 1), since they regress each person's depth independently (Wang et al., 2024) or express the trajectory in per-person coordinates (Shen et al., 2024; Li et al., 2025). Large-scene crowd methods such as Crowd3D (Wen et al., 2023), working on one image at a time, localize people via a shared scene geometry and anchor each person by the Human-scene Virtual Interaction Point (HVIP), the point on the scene surface directly below the torso. That 3D point is regressed as a pixel distance below the torso in the crop, a distance that varies with the crop as well as with the person, so the target jitters even as the person moves smoothly (Fig. 2, left). Even a perfect regressor therefore cannot recover temporally consistent 3D positions from it. Crowd3D's video extensions (Kang et al., 2026; Wen et al., 2025) are no longer feed-forward, relying on per-sequence optimization that takes about four hours for 100 people over 200 frames. No existing method is at once feed-forward and spatio-temporally consistent in a large scene (Table 1). To close this gap, a feed-forward method must regress from each crop a crop-invariant representation anchored in the shared scene, and recover from these estimates a spatio-temporally consistent trajectory.

We observe that a single scalar suffices as such a crop-invariant representation: the metric height of each joint above the scene surface. It changes only with the person's own motion and is therefore stable over time (Fig. 2, right); yet it discards nothing: given the joint's 2D observation, the camera, and a known or estimated scene surface, it can be lifted analytically back to a 3D estimate in the shared scene. No single joint suffices, however: lifting a lower joint yields a more accurate position, but low joints are the most easily occluded. We therefore regress the heights of all joints directly from each crop, and introduce this representation as **Multi-Joint Height (MJH)**. We complete the representation with **Metric Height Lifting (MHL)**, which turns the regressed height of a joint into a metric 3D estimate of that joint in the shared scene, yielding one estimate per joint and frame.

Averaging these estimates does not yield the trajectory: each joint implies a different root position, and since both the height and the joint-to-root mapping are subject to pose-dependent error, these discrepancies are systematic and do not cancel out each other. We therefore propose **Global Trajectory Fusion (GTF)**, which regresses the trajectory jointly from the per-joint 3D estimates and in the same space as those estimates. Such a regression generalizes poorly to long sequences, so GTF adopts **phase-shifted coverage**: fixed-length windows, each regressed in its own coordinates, related to the shared scene by a known rigid transform, so that the regressed positions stay bounded, with the windows shifted by half their length in successive rounds to bridge the window boundaries.

A domain gap remains: crops in a large scene leave the principal point well outside them, at offsets that close-range data does not cover, while large-scene video with 3D annotation is scarce. We therefore propose **Canonical Camera Normalization (CCN)**, an exact warp that treats each crop as a camera rotation and a zoom, absorbing the cropping difference into the camera model, so that training is no longer confined to large-scene data.

Assembled from these three components, **StableCrowd** is the first feed-forward method to achieve spatio-temporally consistent reconstruction from large-scene video (Table 1). CCN normalizes each crop, and the network regresses MJH and the local human pose and shape from each normalized crop. GTF then aggregates the per-joint 3D estimates into trajectories in the shared scene. Experiments demonstrate the effectiveness of StableCrowd. Our technical contributions are as follows:

- **Multi-Joint Height.** A crop-invariant per-joint target that has the range of the human body and stays stable over time, regressed from each crop by a multi-task network and lifted by MHL to metric 3D estimates in the shared scene.
- **Global Trajectory Fusion.** A learned fusion that regresses the trajectory in the same space as the per-joint 3D estimates, keeping their absolute placement, and covers a sequence of any length through phase-shifted coverage.
- **Canonical Camera Normalization.** A strategy that closes the domain gap between close-range and large-scene data, through an exact warp that treats each crop as a camera rotation and a zoom and recenters the principal point.

2 Related Work

Whole-image human regression. Whole-image approaches regress every person from the image at once. ROMP and BEV localize people through body-center representations (Sun et al., 2021; 2022), while Multi-HMR queries full-image features at detected head locations (Baradel et al., 2024). Recent work further varies the token resolution with person scale (Su et al., 2025), or takes spatial prompts alongside the image (Wang et al., 2025b). Human3R instead produces people, scene geometry and the camera together from video, building on a general-purpose geometry foundation model (Chen et al., 2026). It reads people out of that model, kept frozen and pretrained on scene-centric data (Wang et al., 2025a), and takes the image stream alone: the scene geometry and the camera, with intrinsics solved from the predicted point maps, are its outputs, not its inputs. However, these approaches take a fixed-size input, so a large scene must be downsampled into it, discarding most of the pixels a person occupies, and few people are recovered.

Per-person video regression. Per-person video methods instead estimate each person's pose and shape from their own image crop, at the original resolution, and recover their global motion from camera trajectories, motion priors, or scene geometry. One route composes the motion regressed from each crop with a single camera trajectory recovered from the video (Wang et al., 2024). The people then share a camera, but each person's depth is still regressed independently. Another route regresses a per-frame root displacement rather than a position, and rolls it out into a trajectory in each person's own coordinates (Shen et al., 2024; Li et al., 2025): in GVHMR the orientation is predicted in a gravity-aligned coordinate system, and the roll-out starts at that person's first frame. UniSH (Li et al., 2026) crops as well but, trained on clips in which a single subject fills much of the frame, fails on the small people of a large scene (Sec. K). While these approaches recover temporally coherent motion, each person's position is regressed independently of the others, so cannot achieve spatial consistency.

Large-scene crowd reconstruction. Large-scene crowd approaches target spatial consistency, typically by anchoring people to a shared scene. Crowd3D makes the anchor explicit: it pre-estimates a scene-level camera and a ground plane, and predicts a Human-scene Virtual Interaction Point on the scene surface, from which a metric position follows (Wen et al., 2023). GroupRec (Huang et al., 2023) imposes the same relation softly, as a prior over the crowd rather than an explicit anchor, on top of box-conditioned regression (Li et al., 2022). Both GroupRec and CoMHR additionally encode relations among the people in the crowd (Huang et al., 2023; Sun et al., 2026). In video the localization is handed to a per-sequence optimizer, with group-level regularizers and motion priors supplying robustness to occlusion (Kang et al., 2026; Wen et al., 2025). While these methods achieve spatial consistency, none of them provides a representation that can support feed-forward decoding: one that is both anchored to the shared scene and temporally stable.

In contrast, StableCrowd regresses a representation that is anchored to the shared scene geometry and temporally stable, and decodes it into spatio-temporally consistent trajectories for all the people in a large scene, in a feed-forward manner.

3 Method

Given a monocular crowd video, our goal is to recover, for each individual, a spatio-temporally consistent SMPL (Loper et al., 2015) sequence in the world coordinate system, consisting of per-frame pose and translation $\{\boldsymbol{\theta}_i^t,\ \boldsymbol{\tau}_i^t\}_{i=1,\,t=1}^{N,\,T}$ together with per-subject shape $\{\boldsymbol{\beta}_i\}_{i=1}^{N}$.

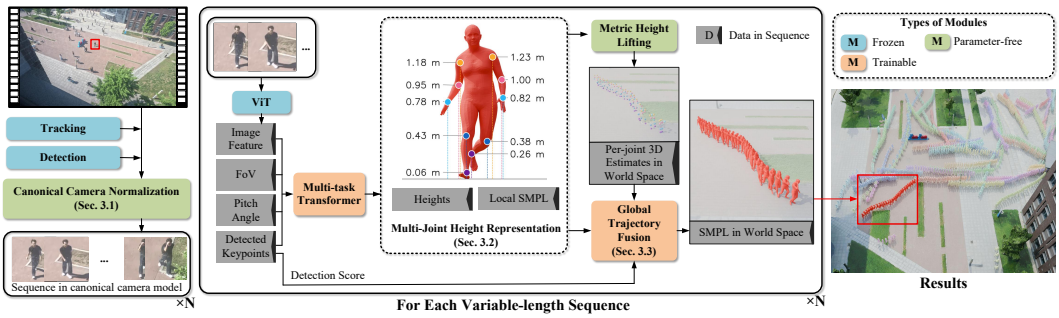

Figure 3: Overview of StableCrowd.

Different from large-scene crowd methods, which regress each person's scene anchor as a pixel distance in the crop and extend to video only through per-sequence optimization, StableCrowd regresses from every crop a target that is defined in the shared scene and stable over time. It then fuses the resulting per-joint 3D estimates into trajectories without per-sequence optimization.

Per-person bounding-box tracks and 2D keypoints come from off-the-shelf models (Zhang et al., 2022; Jiang et al., 2023); each track is processed independently, so we drop the subject index i hereafter for simplicity without ambiguities. The per-video source camera has intrinsics $K_{\text{src}} \in \mathbb{R}^{3\times3}$. A joint's height is measured from the scene geometry below it; we write that geometry throughout as a single ground plane G for simplicity, as it makes lifting a height to 3D linear. Sec. B covers non-planar geometry, and Sec. F covers how K_{src} and G are estimated for an uncalibrated scene. We place the world coordinate system with G as its plane $y = 0$, so the ground normal is the y axis and that height is its y coordinate.

As Fig. 3 shows, a multi-task transformer takes as input the sequence of crops normalized by *Canonical Camera Normalization* (CCN, Sec. 3.1), and regresses *Multi-Joint Height* (MJH, Sec. 3.2) together with the local human pose and shape. *Metric Height Lifting* (MHL, Sec. 3.2) lifts the heights to per-joint 3D estimates, which is then aggregated by *Global Trajectory Fusion* (GTF, Sec. 3.3) into a single trajectory in the world coordinate system.

3.1 Canonical Camera Normalization

To close the domain gap that large-scene cropping opens by pushing the principal point outside the crop, we propose Canonical Camera Normalization (CCN, Fig. 4), which absorbs the position and size of each bounding box into the camera model: every per-frame crop is mapped to a *canonical camera*, a pinhole camera that faces a target anchor on the person with its roll zeroed against the ground normal, whose principal point sits at the image center and whose only free parameter is the focal length, $K_{\text{ccn}}(f) = \text{diag}(f, f, 1)$. Unlike box-conditioned regression (Li et al., 2022), which leaves the difference to the network, CCN removes it before the network sees it; Sec. 4 compares the two.

The source-to-canonical pixel warp in homogeneous coordinates is

$$\phi^t(\mathbf{u}) = K_{\text{ccn}}(f^t)\, R^t_{\text{ccn}\leftarrow\text{src}}\, K^{-1}_{\text{src}}\, \mathbf{u}, \tag{1}$$

with $\mathbf{u}$ a source pixel and $R^t_{\text{ccn}\leftarrow\text{src}} \in \text{SO}(3)$ the source-to-canonical rotation at frame t, fixed by the two conditions above. The warped crop is the **CCN image**. The warp rotates the camera about

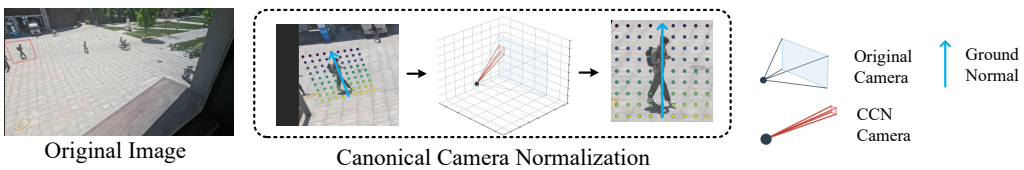

Figure 4: The source and canonical cameras, and the effect of the warp.

its optical center $\mathbf{c}$ and zooms, and neither moves $\mathbf{c}$. So for every world point $\mathbf{X}$, with $R_{\mathrm{src}\leftarrow\mathrm{w}}$ the source camera's world-to-camera rotation,

$$\phi^t\big(K_{\mathrm{src}}\,R_{\mathrm{src}\leftarrow\mathrm{w}}\,(\mathbf{X}-\mathbf{c})\big) = K_{\mathrm{ccn}}(f^t)\,R^t_{\mathrm{ccn}\leftarrow\mathrm{src}}\,R_{\mathrm{src}\leftarrow\mathrm{w}}\,(\mathbf{X}-\mathbf{c}), \tag{2}$$

whose right-hand side is the canonical camera's projection of $\mathbf{X}$, so the 3D-to-2D mapping is preserved, and the only cost is the resampling of a discrete image.

The rotation and the zoom are both determined by the crop and the scene (Fig. 4): the rotation by the anchor and the ground normal of the two conditions above, and the zoom by a human-scale rule set from the person's bounding box. The focal length f^t, together with the residual pitch ρ^t that the two conditions leave, is passed to the network of Sec. 3.2. What makes so elementary a construction load-bearing here is the magnitude of the cross-crop variation it removes, which is specific to large scenes. Sec. A gives the construction.

3.2 Multi-Joint Height

To localize every person in the shared scene, correctly relative to the others, from crops that show them one at a time, we propose **Multi-Joint Height** (MJH): the metric height h^t_j of every joint j above the ground plane G. Unlike the HVIP of Crowd3D (Wen et al., 2023) where a single scene point is regressed as a pixel distance in the crop, a height is stable: it varies continuously and only with the person's own motion, it is the same whichever crop observes it, and its range is that of the human body, not of the scene. A lower joint gives a more accurate position but is more easily occluded, so we regress every joint rather than commit to a specific one.

We select the ten joints on which the detector and the SMPL skeleton agree: the shoulders, elbows, wrists, knees and ankles; the hips are left out, since the detector's hip and SMPL's hip are not the same point. A height stands for a 3D position through the proposed **Metric Height Lifting** (MHL), which recovers the joint's metric position in a closed form from the height and its 2D observation. In the canonical camera's coordinates, with $\mathbf{d}^t_j$ the unit direction of the ray through $\mathbf{u}^t_j$ in the CCN image, $\mathbf{n}^t$ the ground normal, pointing up, and H^t the camera's height above G,

$$\mathbf{x}^t_j = \Lambda^t(\mathbf{u}^t_j, h^t_j) = \begin{cases} \dfrac{h^t_j - H^t}{\mathbf{n}^t\cdot\mathbf{d}^t_j}\,\mathbf{d}^t_j, & |\mathbf{n}^t\cdot\mathbf{d}^t_j| \geq \epsilon \text{ and } (h^t_j - H^t)\,\mathbf{n}^t\cdot\mathbf{d}^t_j > 0, \\ \text{invalid}, & \text{otherwise}, \end{cases} \tag{3}$$

where $|\mathbf{n}^t\cdot\mathbf{d}^t_j| = \sin\psi^t_j$, ψ^t_j is the angle between the ray and G, and ϵ caps the amplification $\|\partial\mathbf{x}^t_j/\partial h^t_j\| = 1/\sin\psi^t_j$ of a height error at $1/\epsilon$; the second condition keeps the point in front of the camera. Sec. B derives MHL.

We regress the representation with a multi-task transformer decoder. It takes an arbitrarily long sequence of ViT-encoded, CCN-normalized image features, together with embeddings of the focal length f^t and the residual pitch ρ^t that Sec. 3.1 leaves, and of the 2D observations $\mathbf{u}^t_j$ with their per-joint detection scores s^t_j. For each observation in each frame it regresses the height h^t_j; for the sequence it regresses the local human pose and shape, per-frame pose and a sequence-shared shape, and per frame a scaled orthographic camera for the CCN image. The pose and shape are regressed without translation, so the space they live in is fixed by a rotation alone: a canonical space, obtained from the canonical camera by removing its pitch relative to the ground. Since CCN has already zeroed the roll against that ground, that pitch is the residual ρ^t, known at every frame rather than estimated. The network is supervised with

$$\mathcal{L}_{\mathrm{MJH}} = \mathcal{L}_h + \lambda_t\,\mathcal{L}_t + \lambda_J\,\mathcal{L}_J + \lambda_V\,\mathcal{L}_V + \lambda_\beta\,\mathcal{L}_\beta + \lambda_{\mathrm{2D}}\,\mathcal{L}_{\mathrm{2D}} + \lambda_c\,\mathcal{L}_c, \tag{4}$$

with $\mathcal{L}_h$ the L1 error of the heights, $\mathcal{L}_t$ a penalty on the second difference of each height along time, $\mathcal{L}_J$ and $\mathcal{L}_V$ the errors of the SMPL joints and vertices, $\mathcal{L}_\beta$ a norm penalty on the shape, $\mathcal{L}_{\mathrm{2D}}$ the error of the joints projected into the CCN image, rotated by ρ^t and then projected by the orthographic camera, and $\mathcal{L}_c$ the error of that camera. Nothing forces the per-joint 3D estimates to agree, so the network trains independently of GTF, which consumes them. At inference, applying Λ^t to every height gives, for each frame, one 3D position per joint, or invalid by Eq. 3; the canonical camera's known pose carries these to the world coordinate system. These *per-joint 3D estimates* are what Sec. 3.3 consumes, along with the joint-to-root offsets the regressed pose implies. The pose and shape themselves are not fused; they reach the output as they are.

3.3 GLOBAL TRAJECTORY FUSION

To recover from the per-joint 3D estimates of Sec. 3.2, a trajectory that is coherent from frame to frame and keeps the absolute placement they carry, averaging them will not serve: they disagree by amounts that do not cancel. We propose Global Trajectory Fusion (GTF), which regresses the trajectory jointly from all the estimates and commits at every frame to a position in the shared scene, one fixed-length window at a time. Methods that regress a per-frame displacement (Shen et al., 2024; Li et al., 2025) run on a sequence of any length without difficulty, because what they regress spans only two frames: it does not grow with the sequence, so length asks nothing of their generalization. What they give up is placement. Their trajectory is fixed only relative to its own first frame, and the displacements accumulate into global drift. GTF takes the other side of that trade. It commits at every frame to an absolute position, which is what the estimates require, and length is then the thing it has to handle. Regressing a whole sequence at once does not handle it: the regression no longer agrees with the estimates it is regressed from.

GTF takes the per-joint 3D estimates of a whole sequence and returns the body's root position in the shared scene at every frame. It works over a window of 120 consecutive frames, in the window's own coordinates — related to the world coordinate system by a known rigid transform fixed by the frames the window covers — and everything below is expressed in those coordinates. Each joint contributes the root position it implies, $\mathbf{r}_j^t = \mathbf{x}_j^t - \boldsymbol{\delta}_j^t$: its lifted position less the joint-to-root offset $\boldsymbol{\delta}_j^t$ of the local pose of Sec. 3.2, rotated into those coordinates; each implied root therefore carries the error of a height and the error of an offset, both of which vary with the pose. A joint contributes ten numbers in all — $\mathbf{r}_j^t$ and $\boldsymbol{\delta}_j^t$, three each, the height h_j^t, the detection score s_j^t, and flags for whether the joint was detected and whether its lifting was well conditioned. Three more are given once per frame: whether the frame is valid, the camera height H^t above G, and the residual pitch ρ^t. Length is handled by phase-shifted coverage. A round covers a sequence with windows laid end to end; the next round covers it again with the windows shifted by half their length, so the boundaries of one round fall inside the windows of the next. Every round after the first reads, alongside the estimates, the trajectory the previous round returned — as further evidence rather than as a state to correct, so what a round returns is a regression and not a residual on the trajectory it was given. A temporal convolution stack over the window regresses the trajectory, in blocks that each refine the previous one, and the same stack is applied in every round. It is trained with

$$\mathcal{L}_{\mathrm{GTF}} = \mathcal{L}_{\mathrm{pos}} + \lambda_a \mathcal{L}_a + \lambda_{\mathrm{shift}} \mathcal{L}_{\mathrm{shift}}, \tag{5}$$

with $\mathcal{L}_{\mathrm{pos}}$ the mean distance to the ground-truth root over the window, $\mathcal{L}_a$ a hinge on the magnitude of the acceleration, and $\mathcal{L}_{\mathrm{shift}}$ a **shift-consistency loss** tying the window's first and last five frames, in position, velocity and acceleration, to the trajectory of the previous round from the second round on.

The trajectory, with the local pose and shape of Sec. 3.2, completes the per-frame pose and translation and the per-subject shape Sec. 3 set out to recover. Carried back by the transform of the window it was solved in, the trajectory is $\boldsymbol{\tau}^t$; the pose carries no translation, so $\boldsymbol{\theta}^t$ follows by the rotation about y that the canonical camera of frame t already determines; the shape $\boldsymbol{\beta}$ is shared over the sequence. Sec. C gives the implementation details.

4 EXPERIMENTS

4.1 BENCHMARKS

Datasets We evaluate zero-shot on two large-scene video datasets with 3D annotation, on which no feed-forward method has been trained: VirtualCrowd (VC) (Wen et al., 2025), a synthetic benchmark with dense crowds of 60 to 200 people per scene and the only one on which the optimization-based large-scene methods report results, and WorldPose (WP) (Jiang et al., 2024), a real benchmark at a person-to-image scale close to that of surveillance video. On VC we hold out the two views of Scene 1 to tune the camera and ground estimator, and evaluate on the remaining six videos. For a valid comparison we keep the track segments that are sufficiently observed according to the ground truth (Sec. D).

Metrics All metrics are computed in the world coordinate system. For spatial consistency among people we report PPDS and its Procrustes-aligned variant PA-PPDS (Wen et al., 2023), which de-

Table 2: **VirtualCrowd**, Crowd4D-compatible protocol. * scored only where Crowd3D's output matches the ground truth, which favors it. Best in bold, second best underlined.

Method	MPJPE (m) ↓	T-MPJPE (mm) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓	ACC norm ratio (→ 1)	PPDS ↑	PA-PPDS ↑
Crowd3D*	<u>2.61</u>	115.36	232.7	169.70	32.575	**0.977**	**0.979**
Crowd4D	14.46	64.98	<u>135.2</u>	7.07	1.674	<u>0.951</u>	0.965
DyCrowd	22.08	76.22	308.5	10.56	2.473	0.824	0.964
TRAM	5.13	63.79	1670.6	960.57	207.309	0.842	0.844
GVHMR	16.38	<u>61.51</u>	354.5	7.23	2.420	0.817	0.844
GEM	18.17	71.48	386.1	**2.29**	**0.953**	0.810	0.842
Ours	**2.19**	**58.58**	**133.5**	<u>2.56</u>	<u>0.789</u>	**0.977**	<u>0.978</u>

Table 3: **WorldPose**, standard protocol. Crowd3D cannot be tested on the moving camera (Sec. G.1). Best in bold, second best underlined.

Method	MPJPE (m) ↓	T-MPJPE (mm) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓	ACC norm ratio (→ 1)	PPDS ↑	PA-PPDS ↑
TRAM	<u>2.71</u>	<u>72.65</u>	<u>1928.0</u>	628.77	397.648	<u>0.841</u>	<u>0.845</u>
GVHMR	17.26	90.44	3952.3	3.44	5.068	0.680	0.755
GEM	19.64	105.89	4876.2	<u>3.04</u>	<u>4.442</u>	0.640	0.700
Ours	**0.33**	**59.95**	**199.2**	**2.26**	**3.142**	**0.984**	**0.985**

pend only on inter-person distances. For temporal consistency we report the ACC error, the frame-to-frame acceleration error of the root, together with the ACC norm ratio, the ratio of the predicted to the ground-truth acceleration norm, whose target is 1 (Sec. E). For absolute localization we report MPJPE, computed without any alignment. As a per-person control we report T-MPJPE, which aligns each estimate to the ground truth by translation alone, and SEQ-RT-MPJPE, which fits a single rigid transform without scale to an entire track, so that drift along the track remains in the error.

4.2 Comparison Protocol

Compared methods We compare against three kinds of method: large-scene crowd methods that work on a single image (Wen et al., 2023); large-scene crowd methods that rely on per-sequence optimization (Kang et al., 2026; Wen et al., 2025); and per-person video methods (Wang et al., 2024; Shen et al., 2024; Li et al., 2025). Of the per-person video methods, GVHMR and GEM place each person in a separate coordinate system rather than one shared across the scene, so we additionally align the outputs of these two by the first frame before evaluation (Sec. G).

Standard protocol For a fair comparison, every method is given the same inputs: ground-truth person tracks, ground-truth camera intrinsics, per-frame extrinsics on WP, and a ground plane. On WP the ground plane is the $z = 0$ plane of WP's world coordinates; on VC it is a single shared estimate. The tracks eliminate differences in detection and allow every output to be matched to the ground truth. Crowd3D is the exception: its output has no one-to-one correspondence with the ground truth, so we score only what matches, which favors it (We use * to highlight this). We focus on surveillance-like capture, where the camera is usually fixed; giving the extrinsics removes camera estimation as a source of difference between methods.

Crowd4D-compatible protocol To include Crowd4D and DyCrowd we change two things. Camera and ground are estimated. Crowd4D and DyCrowd retain their own estimates; we produce one estimate, chosen on the held-out split, and give it to every method we can re-run. The results their authors provided are not in one-to-one correspondence with the given ground-truth tracks, so the other methods receive only those ground-truth tracks that both of them cover.

Training conditions We train our method on H36M (Ionescu et al., 2014), RICH (Huang et al., 2022), BEDLAM (Black et al., 2023), BEDLAM2 (Tesch et al., 2025), and JTA (Fabbri et al.,

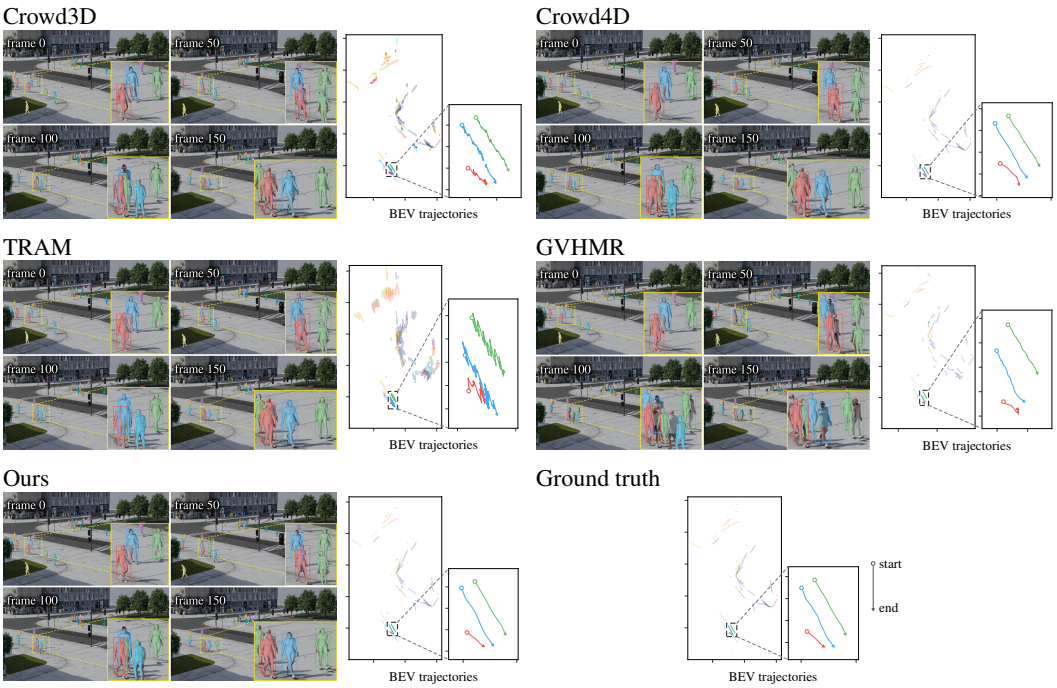

Figure 5: Qualitative comparison on **VirtualCrowd**, Crowd4D-compatible protocol.

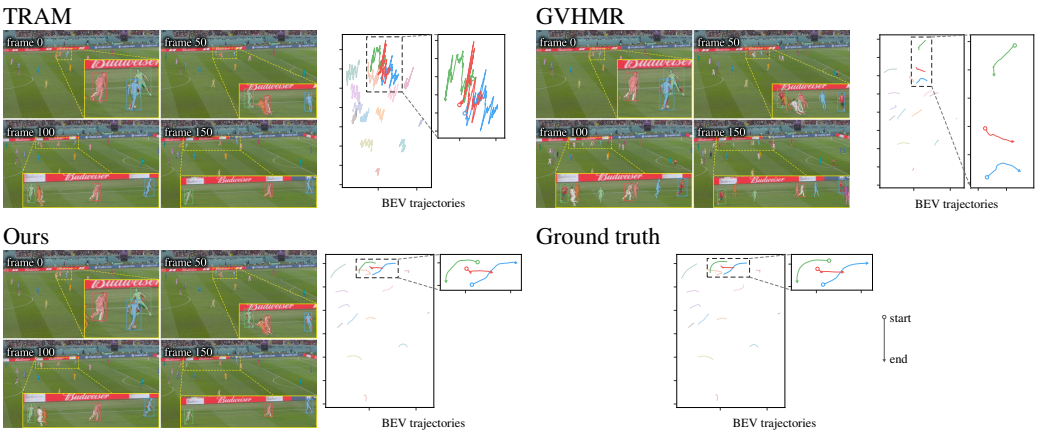

Figure 6: Qualitative comparison on **WorldPose**, standard protocol.

2018). Only JTA is a large-scene dataset; the rest are standard close-range sets. We therefore fine-tune every trainable baseline on JTA. Fine-tuning improves every baseline on the JTA validation set, but the gain does not always carry over to VC and WP, so for each baseline we report the checkpoint with the lower MPJPE on each benchmark and protocol. Sec. H reports the validation record, and Sec. I the results of both checkpoints.

4.3 Comparison Results

Each feed-forward baseline collapses on at least one of three axes (Tables 2 and 3 and Figs. 5 and 6): localization, the layout of the crowd, and temporal stability. Among the per-person video methods, TRAM localizes best, but with by far the largest acceleration error. Crowd3D places the crowd as consistently as ours but jitters severely from frame to frame. GVHMR and GEM are relatively smooth because they have no absolute position: aligned to the image at the first frame, they drift away from the people, more than fifteen meters on average. Ours is the only feed-forward method consistent in both space and time, and it localizes people and their pose best on both datasets. Our feed-forward method reaches the accuracy of Crowd4D, which optimizes every sequence; since each runs on a scene it estimated for itself, we do not read this as one paradigm beating the other (Sec. M).

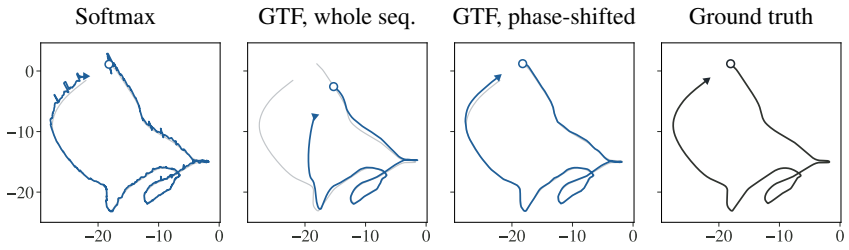

Figure 7: Bird's-eye trajectory of one person over a 1,948-frame **WorldPose** sequence, in meters. All panels fuse MJH. Pale: the ground truth.

Table 4: Crop handling, on local pose. The box encoding follows CLIFF (Li et al., 2022). Cells: VirtualCrowd (standard protocol) / WorldPose. Best in bold. Shaded: the full model.

Crop handling	T-MPJPE (mm) ↓
No extra input	74.7 / 71.3
Box encoding	95.9 / 81.3
CCN	**61.8 / 59.9**

Table 5: Representation and fusion. HVIP follows Crowd3D (Wen et al., 2023). N/A: a single-point representation gives one estimate per frame, so there is nothing to fuse. Cells: VirtualCrowd (standard protocol) / WorldPose. Best in bold. Shaded: the full model.

Representation	Fusion	MPJPE (m) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓
HVIP	N/A	1.80 / 0.57	450 / 277	201.36 / 75.63
Root, height	N/A	1.67 / 0.41	209 / 229	61.94 / 36.65
MJH	softmax	**1.49** / 0.39	228 / 219	131.58 / 79.87
MJH	GTF, whole seq.	1.52 / 3.27	**137** / 3649	**2.17 / 1.87**
MJH	GTF, phase-shifted	1.51 / **0.33**	156 / **199**	2.65 / 2.26

4.4 Ablation

Our representation, MJH fused by GTF, gives global positions that are both accurate and stable (Table 5). Every single-point representation jitters, though regressing the height of the root instead of the HVIP of Crowd3D already more than halves its acceleration error. Relying on the low joints through a softmax over MJH that favors lower joints comes close in accuracy but not in stability: the acceleration error stays more than an order of magnitude above ours (Table 5 and Fig. 7). GTF over the whole sequence as one window is smooth but holds only within the 120 to 360 frames it is trained on: it matches ours on the 200-frame VC videos and drifts on the far longer WP sequences (Table 5 and Fig. 7). The local pose, unlike the height, still depends on the crop; CCN removes the cross-crop variation and gives the lowest T-MPJPE (Table 4), while encoding the box as CLIFF does is worse than giving no extra input, since the offset it encodes, from the principal point to the crop center in crop sizes, averages 0.69 in training (BEDLAM and BEDLAM2) but 12.86 on VC and 3.96 on WP.

5 Conclusion

We presented **StableCrowd**, the first feed-forward method to reconstruct spatio-temporally consistent crowd motion from monocular large-scene video. Where large-scene crowd methods regress each person's scene anchor as a pixel distance in the crop, a target that varies with the crop and so jitters over time, we regress Multi-Joint Height, the metric height of every joint above the scene surface, which changes only with the person's own motion, and Metric Height Lifting lifts it to per-joint 3D estimates in the shared scene. Global Trajectory Fusion regresses the trajectory in the same space as these estimates, committing to an absolute position at every frame rather than only a relative displacement, and through phase-shifted coverage handles sequences of any length. Canonical Camera Normalization treats each crop as a camera rotation and a zoom, bringing back to the image center the principal point that a large scene leaves far outside the crop, so that the local pose and shape learned from close-range single-person data carry over to large-scene video. Zero-shot on VirtualCrowd and WorldPose, StableCrowd shows that spatio-temporal consistency in large scenes no longer requires per-sequence optimization.

AI USE STATEMENT

We used large language models (LLMs) in this work as follows.

- *Code.* The authors determined the ideas, the data flow and how the datasets are processed; LLMs only implemented them, and were also used for code review.
- *Empirical hyperparameters.* For settings such as the learning rate and the scheduler, LLMs surveyed the values used in existing open-source repositories, and the authors chose among them.
- *Writing.* LLMs generated text from the code and the authors' insights, which the authors used as a reference; the final text was always settled by the authors. LLMs were also used for polishing, including when co-authors revised the text. For citations, LLMs only checked the format, including whether cited arXiv preprints have since been formally published; no additional literature search was done with LLMs during writing.
- *Figures.* LLMs were used only for layout and for vector elements such as arrows; none of the image content in the figures was generated by AI.
- *Literature survey.* LLMs searched for papers by topic and summarized them along dimensions and with a template set by the authors.

For every other task whose disclosure the ICLR policy requires, LLMs were either not used or the task does not apply to this work. We have reviewed all LLM-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of LLMs.

REFERENCES

Fabien Baradel, Matthieu Armando, Salma Galaaoui, Romain Brégier, Philippe Weinzaepfel, Grégory Rogez, and Thomas Lucas. Multi-HMR: Multi-person whole-body human mesh recovery in a single shot. In *Proc. European Conference on Computer Vision*, 2024.

Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 8726–8737, June 2023.

Yue Chen, Xingyu Chen, Yuxuan Xue, Anpei Chen, Yuliang Xiu, and Gerard Pons-Moll. Human3R: Everyone everywhere all at once. In *Proc. International Conference on Learning Representations*, 2026.

Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *European Conference on Computer Vision (ECCV)*, 2018.

Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: Exceeding YOLO series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *Proc. IEEE/CVF International Conference on Computer Vision*, 2023.

Buzhen Huang, Jingyi Ju, Zhihao Li, and Yangang Wang. Reconstructing groups of people with hypergraph relational reasoning. In *Proc. IEEE/CVF International Conference on Computer Vision*, pp. 14827–14837, 2023.

Chun-Hao P. Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J. Black. Capturing and inferring dense full-body human-scene contact. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 13274–13285, June 2022.

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, 2014.

Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. RTMPose: Real-time multi-person pose estimation based on MMPose. *arXiv preprint arXiv:2303.07399*, 2023.

Tianjian Jiang, Johsan Billingham, Sebastian Müksch, Juan Zarate, Nicolas Evans, Martin Oswald, Marc Pollefeys, Otmar Hilliges, Manuel Kaufmann, and Jie Song. WorldPose: A World Cup dataset for global 3D human pose estimation. In *Proc. European Conference on Computer Vision*, 2024.

Hongbo Kang, Tianyi Zhou, Qingyang Yang, Hongwei Wen, Jing Huang, Yu-Kun Lai, and Kun Li. Crowd4D: Scene-aware monocular 4D crowd reconstruction. In *Proc. International Conference on Machine Learning*, 2026.

Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. GENMO: A GENeralist model for human MOtion. In *Proc. IEEE/CVF International Conference on Computer Vision*, pp. 11766–11776, October 2025.

Mengfei Li, Peng Li, Zheng Zhang, Jiahao Lu, Chengfeng Zhao, Wei Xue, Qifeng Liu, Sida Peng, Wenxiao Zhang, Wenhan Luo, Yuan Liu, and Yike Guo. UniSH: Unifying scene and human reconstruction in a feed-forward pass. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying location information in full frames into human pose and shape estimation. In *Proc. European Conference on Computer Vision*, 2022.

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 *Proc. European Conference on Computer Vision*, pp. 740–755, 2014.

Shaohui Liu, Yifan Yu, Rémi Pautrat, Marc Pollefeys, and Viktor Larsson. 3D line mapping revisited. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21445–21455, 2023.

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.

Zehong Shen, Huaijin Pi, Yan Xia, Zhi Cen, Sida Peng, Zechen Hu, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. World-grounded human motion recovery via gravity-view coordinates. In *Proc. ACM SIGGRAPH Asia*, pp. 1–11, 2024.

Chi Su, Xiaoxuan Ma, Jiajun Su, and Yizhou Wang. SAT-HMR: Real-time multi-person 3D mesh estimation via scale-adaptive tokens. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Iago Suárez, José M. Buenaposada, and Luis Baumela. ELSED: Enhanced line SEgment drawing. *Pattern Recognition*, 127:108619, 2022.

Minghao Sun, Chongyang Xu, Yitao Xie, Buzhen Huang, and Kun Li. Contrastive multi-modal hypergraph reasoning for 3D crowd mesh recovery. In *Proc. IEEE International Conference on Multimedia and Expo*, 2026.

Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3D people. In *Proc. IEEE/CVF International Conference on Computer Vision*, pp. 11179–11188, 2021.

Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3D people in depth. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13233–13242, 2022.

Joachim Tesch, Giorgio Becherini, Prerana Achar, Anastasios Yiannakidis, Muhammed Kocabas, Priyanka Patel, and Michael J. Black. BEDLAM2.0: Synthetic humans and cameras in motion. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025a.

Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. TRAM: Global trajectory and motion of 3D humans from in-the-wild videos. In *Proc. European Conference on Computer Vision*, pp. 467–487. Springer Nature Switzerland, 2024.

Yufu Wang, Yu Sun, Priyanka Patel, Kostas Daniilidis, Michael J. Black, and Muhammed Kocabas. PromptHMR: Promptable human mesh recovery. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1148–1159, June 2025b.

Hao Wen, Jing Huang, Huili Cui, Haozhe Lin, Yu-Kun Lai, Lu Fang, and Kun Li. Crowd3D: Towards hundreds of people reconstruction from a single image. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8937–8946, 2023.

Hao Wen, Hongbo Kang, Jian Ma, Jing Huang, Yuanwang Yang, Haozhe Lin, Yu-Kun Lai, and Kun Li. DyCrowd: Towards dynamic crowd reconstruction from a large-scene video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *Proc. European Conference on Computer Vision*, pp. 1–21, 2022.

A CCN CONSTRUCTION DETAILS

Sec. 3.1 defines the canonical camera by two conditions on its rotation and one free parameter, the focal length. This section constructs both from the tracked 2D detections.

The rotation. In the source camera's coordinates, let $\tilde{\mathbf{a}}^t$ be the target anchor in homogeneous pixel coordinates, the center of the person's bounding box, and $\mathbf{n}$ the unit ground normal, pointing up. The first condition puts the canonical principal axis through the anchor, $\mathbf{e}_z = K_{\text{src}}^{-1}\tilde{\mathbf{a}}^t / \|K_{\text{src}}^{-1}\tilde{\mathbf{a}}^t\|$; the second zeroes the roll, which makes the canonical x axis parallel to the ground, $\mathbf{e}_x = \mathbf{e}_z \times \mathbf{n}/\|\mathbf{e}_z \times \mathbf{n}\|$, and $\mathbf{e}_y = \mathbf{e}_z \times \mathbf{e}_x$ completes the frame, so that $R^t_{\text{ccn}\leftarrow\text{src}}$ has rows $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$. The two conditions consume the three degrees of freedom of $\mathrm{SO}(3)$: the first fixes the direction of the principal axis, two of them, and the second the rotation about it, the third. What they do not remove is the angle between the principal axis and the ground, the residual pitch ρ^t, $\sin\rho^t = |\mathbf{n}\cdot\mathbf{e}_z|$ with $\mathbf{n}\cdot\mathbf{e}_z < 0$ for a camera that looks down, which is handed to the network of Sec. 3.2 together with f^t. In the canonical camera's coordinates it becomes $\mathbf{n}^t$, which lies in the y–z plane and is the form Sec. B uses.

The focal length. The canonical intrinsics are $K_{\text{ccn}}(f^t) = \mathrm{diag}(f^t, f^t, 1)$ on a CCN image of side $S = 256$ pixels with the principal point at its center, so aspect ratio, principal-point offset and skew are removed by construction and f^t is the only intrinsic degree of freedom. It is set by a human-scale rule from the person's bounding box, one square side per track from its largest box and a per-frame scale smoothed along time. The source focal length cannot be kept instead by moving the camera toward or away from the person. The warp is exact only because the optical center stays put, so this one degree of freedom in the focal length is necessary.

B MHL DERIVATION

Sec. 3.2 regresses, for each joint j of frame t, a metric height h^t_j above the ground plane G, and lifts it, with the joint's 2D observation $\mathbf{u}^t_j$, to a 3D position $\mathbf{x}^t_j = \Lambda^t(\mathbf{u}^t_j, h^t_j)$. This section derives MHL, Λ^t of Eq. 3, and its conditioning.

Everything below is in the coordinates of the canonical camera of frame t, which CCN (Sec. 3.1) constructs: the origin is its optical center, it looks along $+z$, and its principal point is the image center, so with image coordinates measured from there its intrinsics are $K_{\text{ccn}}(f^t) = \mathrm{diag}(f^t, f^t, 1)$. The ground G has unit normal $\mathbf{n}^t$, pointing up, and lies H^t below the origin, so a point $\mathbf{x}$ has height $H^t + \mathbf{n}^t\cdot\mathbf{x}$ above G; CCN has zeroed the camera's roll, so $\mathbf{n}^t$ lies in the y–z plane and is fixed by the residual pitch ρ^t.

A 2D observation fixes the joint only up to its viewing ray, leaving one unknown: how far along that ray the joint lies. A height removes exactly that one, because it is measured from the ground along the ground normal, and the scene fixes that ground independently of the image. One scalar per joint is therefore the exact amount of information the image leaves to be supplied. Above a plane the constraint is linear and the solution is closed form.

A ground plane. The 2D observation, in the CCN image and in homogeneous coordinates $\tilde{\mathbf{u}}^t_j = (u^t_{j,x}, u^t_{j,y}, 1)^\top$, back-projects to the unit direction

$$\mathbf{d}^t_j \;=\; \frac{K_{\text{ccn}}(f^t)^{-1}\,\tilde{\mathbf{u}}^t_j}{\left\|K_{\text{ccn}}(f^t)^{-1}\,\tilde{\mathbf{u}}^t_j\right\|}, \tag{6}$$

so that μ measures distance from the camera along the ray $\{\mu\,\mathbf{d}^t_j : \mu > 0\}$. A point of the ray has height $H^t + \mu\,\mathbf{n}^t\cdot\mathbf{d}^t_j$, so $H^t + \mu\,\mathbf{n}^t\cdot\mathbf{d}^t_j = h^t_j$ fixes $\mu^t_j = (h^t_j - H^t)/\mathbf{n}^t\cdot\mathbf{d}^t_j$ and $\mathbf{x}^t_j = \mu^t_j\,\mathbf{d}^t_j$, the first branch of Eq. 3. The point is unique when $\mathbf{n}^t\cdot\mathbf{d}^t_j \neq 0$ — a ray parallel to G meets the plane at height h^t_j nowhere, or everywhere if the camera sits at that height — and lies in front of the camera when $\mu^t_j > 0$, which Eq. 3 requires and which an elevated view gives for every joint below it.

Differentiating Eq. 3 at a fixed observation,

$$\left\| \frac{\partial \mathbf{x}_j^t}{\partial h_j^t} \right\| = \frac{1}{|\mathbf{n}^t \cdot \mathbf{d}_j^t|} = \frac{1}{\sin \psi_j^t}, \tag{7}$$

with $\psi_j^t \in [0, \pi/2]$ the acute angle between the ray and G. The derivative has component along $\mathbf{n}^t$ exactly one, the height error reproducing itself, and horizontal magnitude $\cot \psi_j^t$, which is the part the camera geometry decides. It does not depend on h_j^t, so the one threshold ϵ of Eq. 3 serves every joint at every height; Sec. C gives its value. Across joints the angle does vary, since a joint at another height is seen elsewhere: for joints of one person at horizontal distance ℓ from the camera, $\tan \psi_j^t = |H^t - h_j^t|/\ell$, so under a camera above them all a lower joint is seen at a larger ψ_j^t, and is located more accurately provided the regressed heights carry comparable error.

Arbitrary terrain. Over arbitrary terrain the ground beneath a candidate point is no longer the same everywhere, so the constraint is not linear and no closed form exists in general. It remains solvable, and for the same reasons.

- The observation still leaves one unknown along the ray, and the height is still one constraint on it.
- That constraint can be evaluated anywhere on the ray. The scene is asked one question — the height of a point above the ground beneath it — and any representation that answers it will serve.
- A solution exists. Along the ray that height runs continuously from the camera's own height above the ground, at the camera, down to zero where the ray meets the ground, so every height between zero and the camera's own is attained.
- The first such point, counted from the camera, is the one taken, which makes the output well defined.

The degeneracy carries over: where that height varies slowly along the ray, the crossing is determined as poorly as a shallow intersection is, and the same threshold, applied to that rate, rejects it. The plane is this argument with the ground held constant, where the crossing is unique and has the closed form above. The scene is given as a ground-only depth map rendered for the view, which answers that question directly. Nothing else about the scene is required, and nothing in the argument above depends on this choice.

Sec. L shows the recovered motion on such terrain.

C Implementation Details

Height network. Image features come from a frozen HMR2.0 (Goel et al., 2023) ViT-H encoder applied to each 256×256 CCN image, one 1024-d token per frame. The multi-task decoder is a 12-layer transformer of width 512 with 8 heads, rotary position embedding, pre-layer normalization and zero-initialized gates on its attention and MLP branches. Its conditions are added to the frame token after a linear embedding each: the focal length as the field of view, the residual pitch ρ^t, and the 15 COCO (Lin et al., 2014) body keypoints with their detection scores, where an undetected keypoint is replaced by a learned per-joint embedding. The heads regress the local pose as 6D rotations, a shape pooled over the sequence, the orthographic camera as a log scale and a 2D translation, and the ten heights through a linear layer and a softplus, so a height is never negative. Training sequences are 120 frames; at inference the whole sequence is one forward pass, with attention restricted to a band of 120 frames when the sequence is longer, and a single frame is a valid input.

Trajectory decoder. The ten numbers of a joint and the three of a frame are embedded by two two-layer MLPs to 128 each and concatenated, with dropout 0.1, to a 256-d context per frame. A stack of four refinement blocks follows. Each adds to that context a linear embedding of the current position and, from the second round on, of the previous round's trajectory, passes it through two temporal convolutions of kernel 5 with a residual connection and layer normalization, and projects it to an increment on the position. The same stack serves all three rounds, and the windows of a

round are joined by linear blending across their seams. The previous round's trajectory is detached, so no gradient crosses rounds.

Training. The height network is trained on BEDLAM and BEDLAM2 in 120-frame sequences, AdamW with learning rate 5×10^{-4} and weight decay 0.01, gradient clipping at 0.5, 50 epochs, on one GPU; each condition is dropped with probability 0.1 per frame, and there is no image augmentation. The weights of Eq. 4 are $\lambda_t = 1$, $\lambda_J = 0.25$, $\lambda_V = 1$, $\lambda_\beta = 1$, $\lambda_{\text{2D}} = 5/256$ and $\lambda_c = 1$; $\mathcal{L}_h$ is the L1 error over the ten joints, $\mathcal{L}_t$ a piecewise-linear penalty on the second difference of each height whose breakpoints are the 90th and 95th percentiles of that joint on the training set, $\mathcal{L}_\beta$ a hinge on $\|\boldsymbol{\beta}\|$ above 3, and $\mathcal{L}_c$ the squared error of the three camera parameters. The threshold of Eq. 3 is $\epsilon = \sin 2^\circ$. The trajectory decoder is trained over its three rounds on the per-joint 3D estimates of the frozen height network. Windows of 120 frames come from JTA, H36M, RICH and BEDLAM, keeping runs of at least 240 frames whose root moves at least 2.5 m; AdamW with learning rate 3×10^{-4} and weight decay 10^{-4}, gradient clipping at 0.5, 60 epochs, on one GPU. The height network keeps the checkpoint with the best validation T-MPJPE and the trajectory decoder its final epoch. The weights of Eq. 5 are $\lambda_a = 0.517$ and, for the position, velocity and acceleration terms of $\mathcal{L}_{\text{shift}}$, 50, 406.1 and 845.5; $\mathcal{L}_{\text{pos}}$ is also applied to the three intermediate blocks at weight $1/3$, $\mathcal{L}_a$ is zero below $2.5\,\text{m/s}^2$, quadratic to $10\,\text{m/s}^2$ and Huber beyond, and $\mathcal{L}_{\text{shift}}$ weights the five edge frames $1, 0.8, 0.6, 0.4, 0.2$ from the boundary inward, with $\mathcal{L}_{\text{pos}}$ faded by the complement there.

Inference of the trajectory. Windows of 120 frames are laid at a phase of 60, and every window is regressed three times: in the first round without a prior, in the second with the two first-round neighbors as its prior, and once more with the second-round neighbors, the first and last windows falling back to the first round on their unsupported side. A sequence is padded to the next multiple of 120 frames by repeating its last frame, and only its real frames are returned. The window's coordinate origin is the valid frame nearest its center that is supported by its neighbors; a window with no valid frame takes the origin at zero.

Ablated versions. Apart from the component named in its row, each ablated version keeps the network of the full model and is trained on the same data. The single-point representations regress one point per person and frame, so there is nothing to fuse. HVIP takes the torso center, the HVIP and its lifting to 3D as Crowd3D defines them, and Root, height regresses the height of the body root. The softmax fusion weights the per-joint 3D estimates of MJH in each frame by $\text{softmax}_j(-h_j^t/\tau)$ with $\tau = 0.15$ m, so that a lower joint receives a larger weight. GTF, whole seq. regresses the whole sequence as one window of any length, with rotary position embedding, in place of phase-shifted coverage; everything else is GTF's, except the shift-consistency loss and the training sequences, of 120, 240 and 360 frames. No extra input replaces CCN with the plain crop and tells the network nothing about where the crop lies in the image, and Box encoding adds that location as CLIFF (Li et al., 2022) does.

D Evaluation Dataset Population

The evaluation population is built from each dataset's own annotation by one rule with the same parameters on both datasets. A frame is valid when at least 9 of the COCO-17 ground-truth joints project visibly. Every continuous ground-truth track run, grouped by scene and track identity, is trimmed to its first and last valid frame; the trimmed run is kept when it spans at least 60 frames and at most one third of its frames are invalid. Invalid frames inside a kept run are retained, runs separated by a gap are not stitched, and time is not resampled.

VirtualCrowd carries two populations, which differ only in the set they start from. The one used under the standard protocol starts from every released ground-truth person-frame occurrence and retains 140,344 person-frames in 727 runs over the six evaluated videos. The one used under the Crowd4D-compatible protocol starts from the intersection of that set with the occurrences for which both Crowd4D and DyCrowd provide output that maps to the released ground-truth track, and retains 134,750 person-frames in 701 runs; this is what lets that table evaluate every method on the same people. WorldPose uses a single population over the full release.

E Details of Metrics

The ACC error is computed on the root joint rather than on all joints, as the norm of the difference between the predicted and the ground-truth accelerations, each taken as $x_{t+1} - 2x_t + x_{t-1}$. The ACC norm ratio instead takes the acceleration as $(x_{t+2} - 2x_t + x_{t-2})/4$, drops three frames at each end of a track, and divides the summed predicted norm by the summed ground-truth norm. One evaluator computes both for every method, Crowd4D and DyCrowd included, from the world-coordinate results their authors provided (Sec. G).

T-MPJPE aligns the root joint by translation in every frame independently.

SEQ-RT-MPJPE fits a full rigid transform in SO(3), not a rotation about the gravity axis alone, so a constant tilt of the whole reconstruction is absorbed by the alignment and does not appear in the number.

F Camera and Ground Estimation

We estimate the ground, and the camera when none is given.

We do so once per video, from its first frame. The ground normal is the vertical vanishing point of line segments detected by ELSED (Suárez et al., 2022) and grouped into vanishing points by LIMAP (Liu et al., 2023). When no camera is given, the focal length is solved jointly with the normal, assuming square pixels and a centered principal point. Building on the idea of Crowd3D (Wen et al., 2023), we obtain the ground offset from the shoulder and ankle midpoints of the tracked people under a fixed body-height prior. We add the following filtering, which together with the prior gives three parameters:

- **Keypoint confidence**: the lowest score among the two shoulders and two ankles, which is not normalized to $[0, 1]$, must exceed a threshold, searched from 2.0 to 5.0 in steps of 0.1 and set to 2.1.
- **Ankle spread**: the distance between the two ankles relative to the box width must stay below a threshold, searched over 0.05, 0.10 and 0.20 and set to 0.10.
- **Height prior**: the shoulder midpoint is placed at a fixed height above the ankle midpoint, searched from 1.20 to 1.35 m in steps of 0.01 m and set to 1.22 m.

We select the three parameters by the ground error, the distance between the points where the same image ray meets the estimated ground and the reference ground. For each person-frame, the camera ray through the point below the person meets the reference ground, fitted to the ground-truth ankles, at $\mathbf{X}$ and the estimated ground at $\hat{\mathbf{X}}$, and the error is $\|\hat{\mathbf{X}} - \mathbf{X}\|$. The parameters are selected on the two views of Scene 1. Scene 1 is not used in any evaluation.

G Details of the Compared Methods

Every method below is re-run from its official code and released weights, except Crowd4D and DyCrowd. Each is given the same protocol inputs, and whatever detection, tracking or camera estimation it brings of its own is left out. Unless noted, its output corresponds one to one with the tracks it was given. Their JTA fine-tuning is reported in Sec. H.

G.1 Crowd3D

Beyond the per-frame input Crowd3D needs two further things, a scene-level crop configuration and a per-scene self-supervised adaptation, and both are fitted once per viewpoint. We derive the crop configuration once per scene from its ground-truth tracks, and the boxes passed in the per-frame request are themselves unchanged. Each scene is then adapted once on a single frame. It discovers people on its own, so its output carries no correspondence to the tracks it was given and has to be associated before scoring, with the matching rate given in the captions of Tables 7 and 8. Apart from the correction below we use the released code as it is.



Figure 8: Crowd3D on two adjacent **WorldPose** frames, adapted once for both. Red boxes mark people it misses in one of the two frames; blue boxes mark duplicates.

Fig. 8 runs Crowd3D with its released code and weights on two adjacent frames of WorldPose, after its standard per-scene adaptation, done once for both. It misses people (red boxes) and duplicates others (blue boxes). By design, its adaptation requires the sizes and positions of the largest and the smallest person as external input, and has to be redone whenever they change. On the moving camera of WorldPose they change on every frame. One adaptation takes over two minutes, and WorldPose has 152,392 frames, so we do not test Crowd3D there. VirtualCrowd already shows that its local pose falls short: its T-MPJPE is the highest of all methods under both protocols (Tables 2 and 8).

Correction. The released implementation scans the crop grid row by row and keeps a survivor buffer for the current and the previous row. While a candidate is being compared it writes to that committed state at once, and if a later comparison then rejects the candidate, the write has already taken effect and there is no pending state that could undo it. An accepted person can therefore be deleted by a candidate that is itself never kept. On a frozen set of observations, a person reconstructed in a valid input patch is missing from the merged scene. We hold the writes a candidate proposes until it has passed every comparison the released code makes, and discard them if it has not. Traversal, matching rule, threshold, boundary preference, tie handling and the prediction payload are unchanged, and the corrected merge restores the missing person on the same frozen observations. Every Crowd3D result reported in this paper uses the corrected merge.

G.2 CROWD4D AND DYCROWD

No public implementation of DyCrowd or Crowd4D is available. Their code is announced as forthcoming. The authors provided their results in world coordinates, together with an evaluation that reproduces the numbers in their papers. We evaluate the results they provided, which cover the evaluation population of this paper, because that population was itself filtered by what they cover.

G.3 GVHMR AND GEM

GVHMR and GEM each return two sets of results, one in the camera frame and one in the world. Each receives person runs derived from the ground-truth tracks, observations computed in advance, and the camera intrinsics. The ground is passed as well, and neither consumes it.

Modification. A shared protocol has to choose one of the two sets. We take the world one, because it is the one that keeps the continuous global trajectory this family defines, while the camera one anchors each position by a per-frame regression in image space and does not form a trajectory. Its gauge is each run's own first frame rather than an anchor shared with the dataset, so we add one rigid transform per run, fixed from that first frame and held for the whole run. It fits no scale, levels no ground and re-aligns no frame, so whatever drift follows the first frame stays in the error. This alignment is a concession we make on the family's behalf rather than a capability it claims, and it holds exactly only at the first frame.

G.4 TRAM

The released code of TRAM has explicit support for a static camera, which is the mode we use on VirtualCrowd, and on WorldPose we give the per-frame intrinsics and extrinsics. Its output lies in

Table 6: Each method on the JTA validation subset at epoch 0 and at the epoch selected for it, and on the training set over the selected epoch. Values are rounded to millimeters.

Method	Selected epoch	Validation				Training	
		MPJPE (mm) ↓		T-MPJPE (mm) ↓		MPJPE (mm)	T-MPJPE (mm)
		epoch 0	selected	epoch 0	selected	selected	selected
Crowd3D	8	899	443	160	123	292	114
GVHMR	8	1933	1005	158	116	304	48
GEM	1	6407	1675	138	129	686	89
TRAM	6	4470	954	121	47	836	43
GroupRec	7	4578	1469	171	106	1017	100
CoMHR	10	3504	1885	175	102	1577	95

the camera frame shared by the whole image, so the conversion is exact per-frame geometry with the given extrinsics and no alignment is needed.

G.5 GroupRec and CoMHR

GroupRec and CoMHR normally run their own detector on every image, and we enter from the given boxes instead. The model is given the width and height of the image, a single scalar focal length, one box per person and an ordering by confidence, and it uses the relation among at most eight people within a frame. The remaining modalities CoMHR uses are produced by its own released preprocessing and are not an additional ground-truth input. Its output lies in the camera frame shared by the whole image, so per-frame conversion with the given extrinsics suffices.

Grouping. The original authors confirmed that their LargeCrowd experiments order people by the YOLOX (Ge et al., 2021) detection order and then cut that order into groups of at most eight. We keep that setting, running YOLOX only to give each ground-truth box a confidence, leaving the boxes themselves unchanged and never letting the detector replace them.

H Fine-tuning Effectiveness of the Compared Methods

Only JTA among our training sets is a large scene, so every trainable baseline was fine-tuned on it under one rule. Each method starts from its own released weights and runs for ten full epochs, with a checkpoint kept at every epoch. Epoch 0 is the released initialization, evaluated once on a fixed validation subset before training begins, and the checkpoints of epochs 1 to 10 are each evaluated once on that same subset. The subset fixes both the span that is inferred and the population that is scored, and it does not change across the eleven evaluations. We select the checkpoint with the lowest validation absolute MPJPE, taking the earlier epoch when two are equal. The selection uses absolute MPJPE because this paper is about where a person is placed. No test metric is read before the selection, and once it is made a single full test is run on the selected checkpoint alone.

Table 6 reports each method on the validation subset at epoch 0 and at the epoch selected for it, together with its training error at that epoch. The training error is averaged over all training steps of the epoch. Every method is compared against its own initialization, so the rows are not meant to be read across. All six improve on both metrics. Most of them reach their lowest validation error at a late epoch, while GEM reaches it at epoch 1.

I Additional Quantitative Results

These tables extend the comparison tables of the main text along two axes: they add two further single-image methods, and they give both checkpoints of every baseline we fine-tuned rather than only the one the main text reports. They also add VirtualCrowd under the standard protocol. Protocol, evaluation population and metrics are unchanged, so the rows carried over from the main text report the same runs.

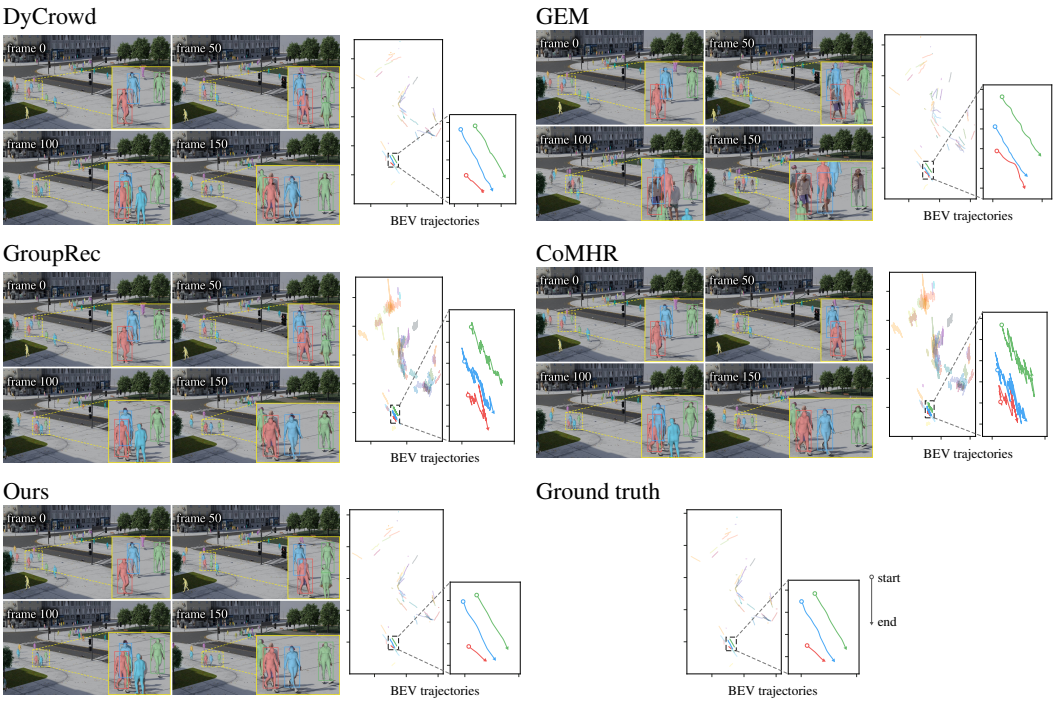

Figure 9: Qualitative comparison on **VirtualCrowd**, Crowd4D-compatible protocol: the methods left out of the main text.

Additional methods. Like Crowd3D, GroupRec and CoMHR reconstruct a whole crowd from a single image. Being anchored to the image, they localize better than GVHMR and GEM, but they carry no temporal model, so their acceleration error is of the same order as TRAM's and their sequence-level error is close to two meters or above under every protocol. Their PPDS (pair-wise percentual distance similarity) stays below 0.86 everywhere, against 0.975 and above for ours, so recovering a crowd in one pass does not by itself give its layout. Fine-tuning on JTA does not change any of this.

J Additional Qualitative Results

Figs. 9 and 10 show the compared methods that the qualitative figures of the main text leave out, on the same frames and people, each figure with ours and the ground truth.

K In-the-Wild Comparison

Fig. 11 compares every method on real footage from a fixed camera, the setting we target and one that neither benchmark covers. Human3R (Chen et al., 2026), SAT-HMR (Su et al., 2025) and Crowd3D find people across the whole image on their own, so their reprojections show everyone they find, while the other methods reconstruct the same given person tracks. SAT-HMR is shown by its reprojection only. The whole-image methods Human3R and SAT-HMR do not recover the crowd: Human3R finds only the few people nearest the camera, and SAT-HMR misses many. As on the benchmarks, Crowd3D, TRAM, GroupRec and CoMHR keep roughly the right layout but jitter from frame to frame, while GVHMR and GEM are smooth but spatially inconsistent, as their BEVs and GEM's side view in Fig. 1 show. Crowd3D's jitter is also marked in Fig. 1 (red boxes), and its reprojections match the people clearly worse than ours. UniSH (Li et al., 2026) places every person within about five meters of the camera, far closer than they stand, as in the close-range clips it is trained on: in Fig. 11 its people are reprojected far too large, and its BEV collapses the crowd into a single cluster. Ours is consistent in both space and time.

Table 7: **VirtualCrowd**, Crowd4D-compatible protocol, all checkpoints. * Crowd3D is scored only on the ground-truth people it matches (0.951 released, 0.928 fine-tuned), which favors it. Within a matched track, *matched* keeps the frames it finds the person in, and *SMPL-fill* fills the rest by linear interpolation (pose on SO(3)) and by repetition at the ends. † marks the checkpoint reported in the main text.

Method	Checkpoint	MPJPE (m) ↓	T-MPJPE (mm) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓	ACC norm ratio (→ 1)	PPDS ↑	PA-PPDS ↑
Crowd3D* (matched)	released	2.67	119.18	266.8	187.94	36.525	0.976	0.978
	JTA-ft†	2.61	115.36	232.7	169.70	32.575	**0.977**	**0.979**
Crowd3D* (SMPL-fill)	released	2.66	120.33	283.4	182.45	37.713	0.975	0.977
	JTA-ft	2.62	116.92	257.5	161.53	33.383	0.975	0.977
Crowd4D	authors'†	14.46	64.98	135.2	7.07	1.674	0.951	0.965
DyCrowd	authors'†	22.08	76.22	308.5	10.56	2.473	0.824	0.964
TRAM	released	5.79	66.79	2278.6	1488.77	299.730	0.825	0.827
	JTA-ft†	5.13	63.79	1670.6	960.57	207.309	0.842	0.844
GVHMR	released	17.34	60.25	460.7	6.99	2.255	0.813	0.841
	JTA-ft†	16.38	61.51	354.5	7.23	2.420	0.817	0.844
GEM	released†	18.17	71.48	386.1	**2.29**	**0.953**	0.810	0.842
	JTA-ft	19.05	86.96	526.8	3.49	1.564	0.802	0.839
GroupRec	released	6.16	92.69	1763.6	1457.82	279.792	0.823	0.827
	JTA-ft	11.22	89.38	1875.8	1540.06	289.812	0.801	0.841
CoMHR	released	8.27	95.39	2649.9	2301.61	429.706	0.774	0.792
	JTA-ft	8.63	89.60	2429.2	2641.22	460.924	0.803	0.833
Ours	–†	**2.19**	**58.58**	**133.5**	2.56	0.789	**0.977**	0.978

Table 8: **VirtualCrowd**, standard protocol, all checkpoints. * as in Table 7. Crowd3D matches 0.930 released and 0.908 fine-tuned.

Method	Checkpoint	MPJPE (m) ↓	T-MPJPE (mm) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓	ACC norm ratio (→ 1)	PPDS ↑	PA-PPDS ↑
Crowd3D* (matched)	released	1.85	120.36	275.2	193.64	37.595	0.974	**0.979**
	JTA-ft	1.79	116.62	241.4	174.78	33.436	0.974	**0.979**
Crowd3D* (SMPL-fill)	released	1.88	122.04	298.4	185.85	38.774	0.972	0.977
	JTA-ft	1.84	118.59	273.7	164.31	34.143	0.972	0.977
TRAM	released	6.31	69.08	2516.8	1562.75	317.390	0.812	0.814
	JTA-ft	5.44	65.52	1887.8	1031.46	223.556	0.834	0.837
GVHMR	released	16.67	63.84	481.7	7.14	2.301	0.808	0.833
	JTA-ft	15.65	64.71	370.0	7.24	2.439	0.815	0.839
GEM	released	18.83	75.39	411.0	**2.43**	**0.990**	0.796	0.820
	JTA-ft	18.42	89.09	545.4	3.58	1.592	0.800	0.833
GroupRec	released	6.49	95.10	2080.4	1633.63	312.955	0.809	0.814
	JTA-ft	12.51	90.62	2070.4	1723.13	323.803	0.790	0.834
CoMHR	released	9.08	97.77	2982.2	2467.87	464.491	0.761	0.780
	JTA-ft	10.02	90.93	2770.4	3027.74	527.375	0.786	0.821
Ours	–	**1.51**	**61.80**	**156.1**	2.65	0.852	**0.975**	0.978

Table 9: **WorldPose**, standard protocol, all checkpoints. † as in Table 7.

Method	Checkpoint	MPJPE (m) ↓	T-MPJPE (mm) ↓	SEQ-RT-MPJPE (mm) ↓	ACC error (mm/f^2) ↓	ACC norm ratio (→ 1)	PPDS ↑	PA-PPDS ↑
TRAM	released	4.23	71.70	2346.7	895.93	507.568	0.819	0.827
	JTA-ft†	2.71	72.65	1928.0	628.77	397.648	0.841	0.845
GVHMR	released†	17.26	90.44	3952.3	3.44	5.068	0.680	0.755
	JTA-ft	24.67	99.97	6053.9	**2.22**	3.319	0.543	0.568
GEM	released†	19.64	105.89	4876.2	3.04	4.442	0.640	0.700
	JTA-ft	25.96	151.72	6544.7	3.55	4.978	0.561	0.573
GroupRec	released	4.83	78.76	2116.6	1126.38	601.613	0.836	0.842
	JTA-ft	5.01	98.70	2306.2	881.92	496.733	0.850	0.861
CoMHR	released	3.51	81.14	2725.6	1543.24	821.353	0.798	0.816
	JTA-ft	3.40	98.91	2630.8	1438.98	753.338	0.835	0.839
Ours	–†	**0.33**	**59.95**	**199.2**	2.26	**3.142**	**0.984**	**0.985**

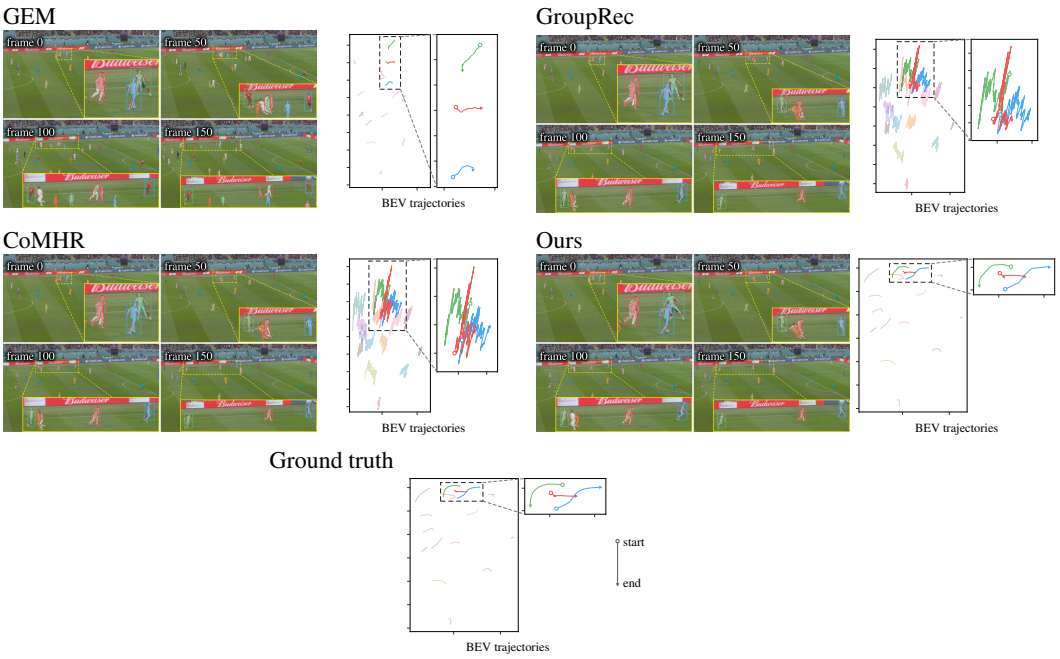

Figure 10: Qualitative comparison on **WorldPose**, standard protocol: the methods left out of the main text.

People in VC and WP take up a markedly smaller part of the image than in this footage, so we do not compare Human3R, SAT-HMR and UniSH quantitatively.

L More Results of StableCrowd

Fig. 13 shows ours on eight more scenes. These scenes all look onto one site, which we capture with a drone and reconstruct in 3D. We filter the clutter off the ground automatically, keep the ground as a single mesh, and render its depth from the camera of each video, which gives the ground-only depth map of Sec. B. The ground is non-planar in rows 1, 2 and 5, where ours still gives a plausible reconstruction. Fig. 12 shows ours with the ground estimated as a plane instead of reconstructed, the camera being kept as calibrated.

M Discussion

M.1 From Optimization to Feed-Forward

The first difficulty of a large scene is one of resolution. Regressing the whole image in a single pass leaves too few pixels per person, so the image has to be cropped and the crops merged back into one scene. Per-sequence optimization (Kang et al., 2026; Wen et al., 2025) never takes that step, and none of this stands in its way. It reaches a consistent scene by other means, solving each sequence on its own, visiting the whole observation repeatedly, and choosing a parameterization per instance. Crowd4D claims its contribution in the scene, taking an estimated point cloud as the support surface and adding a temporal regularizer. DyCrowd claims its own in robustness to occlusion, tying together the motions of people grouped by trajectory similarity. DyCrowd uses a feed-forward HVIP regression to initialize and still leaves the absolute position to the optimizer, and Crowd4D names a feed-forward approximation as future work.

Removing that machinery and replacing it with a feed-forward method is not obvious. What the optimizer used to solve now has to be carried by the representation itself. Crowd3D (Wen et al., 2023) gives one for a single frame, anchored to the shared scene but not stable over time. Three things become hard to learn under regression.

- **The principal-point shift.** Cropping pushes the principal point away from the center of the image, which is the same as changing the camera. An energy term is written in the original camera model, where a crop is only the way an observation is taken and the geometry is untouched. Crowd3D passes that offset to the network as part of its camera input. However, close-range corpora hold no samples at those offsets, and the reason the paper gives for its per-scene fine-tuning is generalization to camera and ground parameters it has not seen. Every new camera view therefore needs a fit of its own. What that costs is measurable in local accuracy, where its T-MPJPE is the highest of any method on both VirtualCrowd tables (Tables 2 and 8). StableCrowd's CCN removes the offset from the input with an exact warp, and that step is no longer needed.
- **A spatio-temporally consistent regression target for absolute position.** World depth spans a range that close-range corpora do not reach, and a regressor has to learn that distribution. Cropping also makes the regression target depend on the crop, so one physical position maps to different target values. Neither is a problem for an optimizer, which learns no distribution and treats global consistency as a free variable. Feed-forward work has already solved the absolute half of it. Crowd3D's network predicts the 2D position of the HVIP, and once the scene camera and ground are given the absolute position follows in closed form. However, a single anchor does not carry the whole of it, since lifting a lower joint yields a more accurate position while low joints are the most easily occluded. StableCrowd therefore regresses the height of every joint with MJH and lifts each of them in closed form with MHL. Those per-joint 3D estimates disagree systematically and averaging does not remove the disagreement, so they have to be fused.
- **Absolute position versus arbitrary length.** It is a sub-problem of the previous item, and again a question about the representation. The fusion is itself a regression, from the per-joint 3D estimates to a trajectory, and it has to satisfy two things at once. Its output has to lie in the space its input lies in, and it has to hold for a sequence of any length. Meeting both at once is a dilemma. Absolute position has to be stated in the frame the whole scene shares, while a regressor generalizes only over a range of values it has seen, and the longer the sequence the further outside that range the positions go. StableCrowd's GTF settles this with phase-shifted coverage, regressing fixed-length windows each in its own coordinates, related to the shared scene by a known rigid transform so that the values stay bounded, and shifting the windows by half their length in successive rounds to bridge the boundaries.

An optimizer pays once for every sequence and goes on paying, while the training cost here is paid once and amortized over every inference that follows. The videos that we target are long and their crowds are dense, so that efficiency there decides whether a method can be used at all. Crowd4D reports about four hours for a scene of 100 people over 200 frames on an NVIDIA 4090D, and DyCrowd about four hours for the same setting on an NVIDIA 3090. On the same 4090D we reconstruct 103 person-frames per second, measured from the given tracks to the reconstructed 3D crowd and excluding bounding-box and keypoint detection; a scene of that size takes about 195 seconds. What the conversion leaves behind is a representation that carries global consistency, and it can be used and studied outside this pipeline.

Per-sequence optimization and a feed-forward method need not be compared at all, since their input contracts and their budgets differ. There is little else on this line to compare against, so we give the closest comparison we could arrange. Our results are better than those of Crowd4D and DyCrowd in every column of Table 2. This should not be read as a feed-forward method defeating optimization, since each method runs on a scene it estimated for itself and the budgets are not of one order.

M.2 Limitations

Our method depends on the upstream person tracks: a person who is not tracked is not reconstructed, and an identity switch within a track carries into the trajectory.

Interactions between people, such as contact and collision avoidance, are not modeled.

Where the view runs nearly parallel to the ground, a height constrains the position only weakly. The multi-joint representation largely mitigates this, since such a view usually leaves some joints well conditioned, and simple engineering remedies, not used in this paper, can cover the rest.

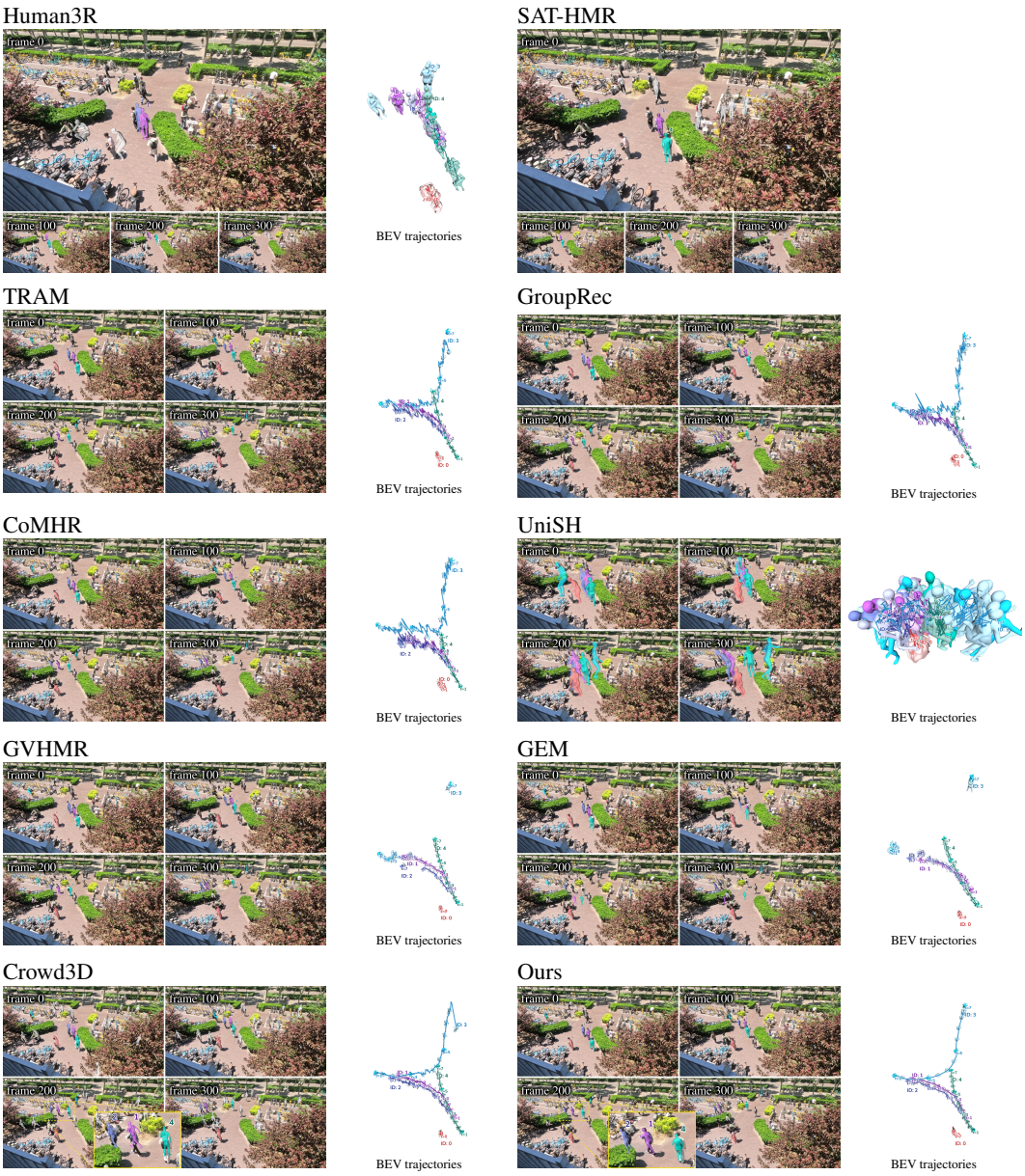

Figure 11: Qualitative comparison **in the wild**.

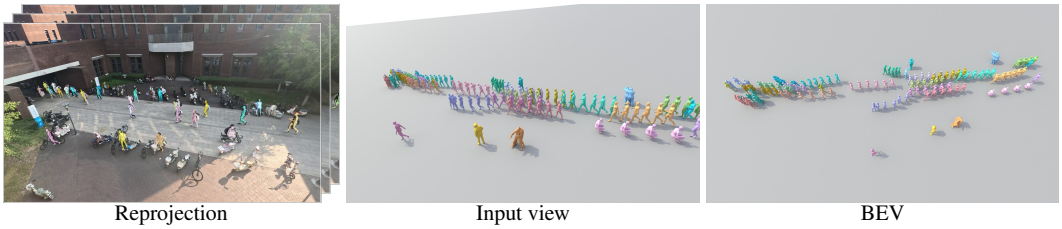

Figure 12: Ours **in the wild**, on an estimated ground plane.

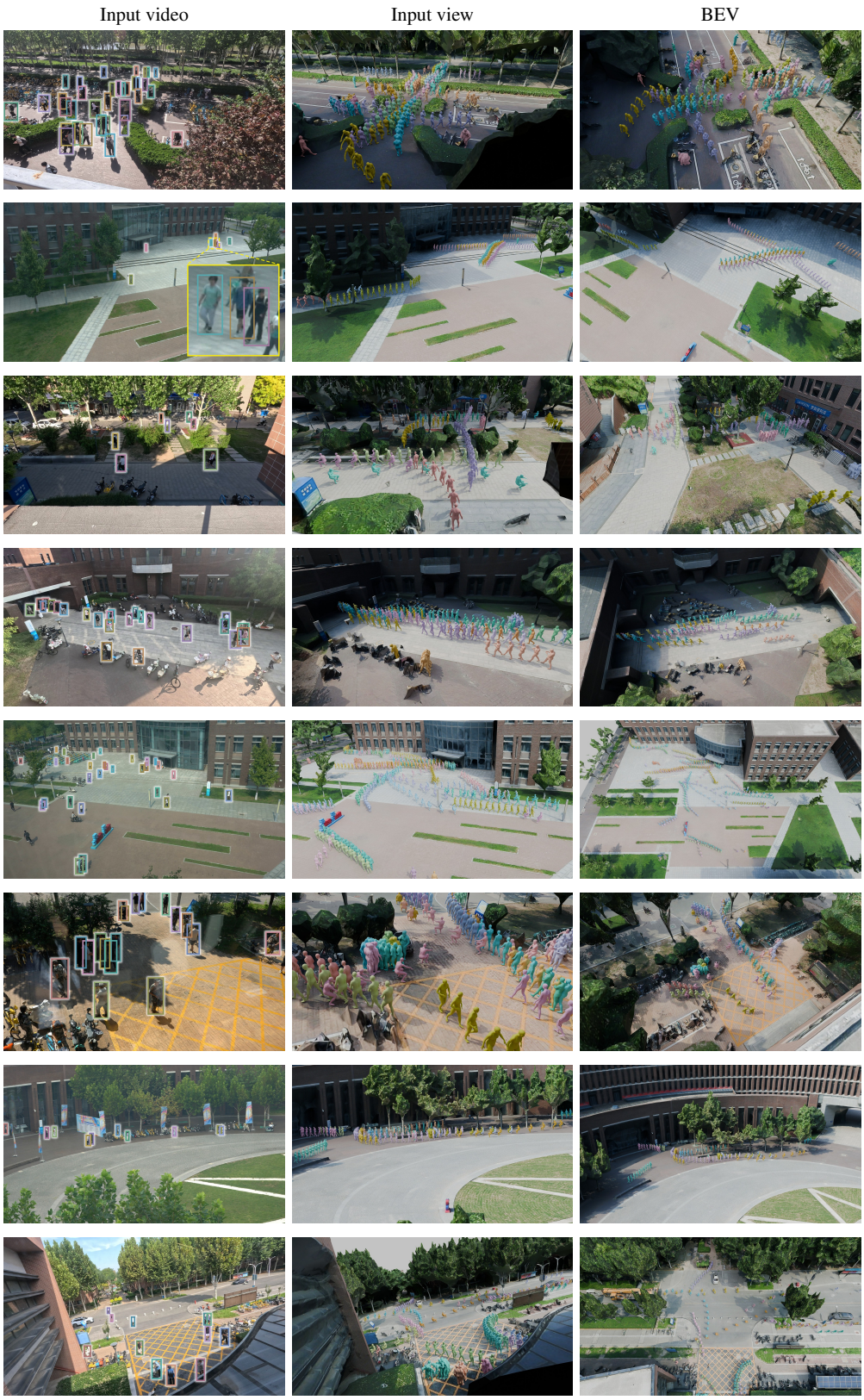

Figure 13: Ours **in the wild**, on more scenes.