4D POINT SPLATTING FOR MMWAVE RADAR MICRO-DOPPLER DIFFERENTIABLE RENDERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Millimeter-wave radar resolves the micrometer displacements of a moving body as micro-Doppler, enabling gait, gesture, activity, pose and vital-sign sensing. Each task relies on a downstream model trained on real data, yet radar datasets lag orders of magnitude behind vision and language. Brute-force collection will not close that gap, and existing synthesis methods (forward-only simulators, learned generators and static-scene inverse renderers) cannot fit a real capture of a moving body, leaving a substantial sim2real gap. We present 4D Point Splatting (4DPS), a physics-based differentiable renderer that fits an explicit moving scene to a real capture. Using the capture's synchronized camera to initialize the room and the moving body, we represent the scene as oriented points carrying material, normal and velocity, and splat each point into range and Doppler bins. Our output is the raw complex return a real radar records, so we can apply the same digital signal processing (DSP) on the rendered data, and obtain any DSP representation to run on and train any SOTA downstream model. We fit 235 sequences on HuPR and 100 on RT-Pose, two datasets with different radars. Our reconstructions set the state of the art, with median correlations of 0.940 and 0.830 on range–azimuth, 0.829 and 0.695 on range–Doppler, and 0.988 and 0.966 on micro-Doppler. Our renders also set the state of the art in downstream sim2real gap. On RT-Pose, a pose model trained on them alone reaches 21.9 cm MPJPE on real data, against 16.7 cm for real training and 127.6 cm for RF-Genesis baseline, and we show that both the rendered- and real-trained models learn nearly the same features. Finally, our custom CUDA kernels fit a sequence in 5.7 minutes on HuPR and 12.9 minutes on RT-Pose on one H100, making dataset-scale inverse rendering of mmWave radar possible for the first time. Together, this work can enable synthetic data generation for any human-centered downstream task and model, and scene editing for data augmentation.

1 INTRODUCTION

Millimeter-wave (mmWave) radar resolves radial motion to a fraction of a millimeter, works in darkness and through occlusion, and captures no identifiable imagery (Zhao et al., 2018; Guan et al., 2020; Adib et al., 2015). This makes it a natural sensor for privacy-preserving human sensing, including gait, gesture, activity, fall and pose recognition (Lien et al., 2016; Singh et al., 2019; Zhao et al., 2018; Lee et al., 2023). Each of these applications requires a downstream model trained on real micro-Doppler (Chen et al., 2006), the phase modulation a moving body imposes on the return. However, each is bottlenecked by the same data scarcity, because public radar data lags orders of magnitude behind vision and language corpora. Brute-force collection cannot close that gap: radar sensors are rarer than cameras, and where a camera is point-and-shoot, every radar capture needs a rig calibrated and synchronized to reference cameras or LiDAR.

Synthesis is the next best alternative, but every existing method leaves a sim2real gap, because none fit to a real capture of a moving body. Motion-capture-driven simulators (Sundar Ram & Ling, 2008; Vishwakarma et al., 2022) carry geometry, materials and motion explicitly but run forward only, so their parameters are set by an operator and never optimized against a real capture. Learned generators such as RF-Genesis (Chen & Zhang, 2024) produce spectrograms from a motion prior trained on the same scarce data, and the scene lives implicitly in network weights, so there is nothing to fit or edit. Radar inverse renderers do fit real captures; mmIR (Armouti et al., 2026b), 3DPS (Armouti

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.

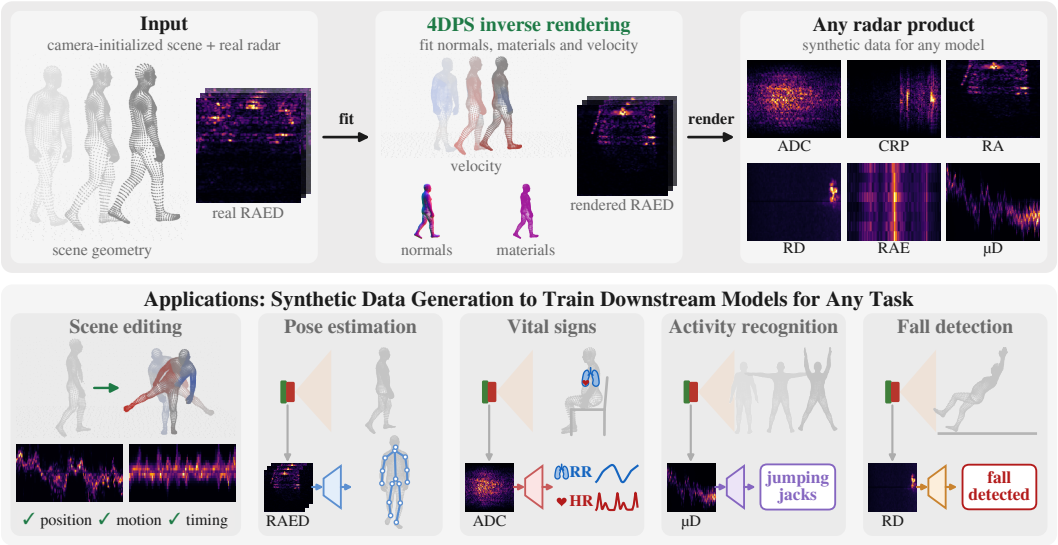

Figure 1: **4DPS fits a moving scene to a real radar capture.** Top: from a camera-initialized scene of oriented points, 4DPS fits materials, normals and velocities (color: radial velocity) until the rendered range–azimuth–elevation–Doppler tensor matches the recording. Bottom: the fitted scene can be edited (App. G) and rendered as training data for pose, vital-sign, activity and fall models.

et al., 2026a) and differentiable ray tracers (Hoydis et al., 2023; Hofmann et al., 2025; Chen et al., 2025; 2023) recover explicit materials and normals and render range, azimuth and elevation. However, all currently assume a static scene and evaluate one phase per point per frame so they have no slow-time axis (the chirp-to-chirp axis along which Doppler is measured) and nothing along which velocity could be defined. What is missing is a renderer that fits an explicit *moving* scene to a real capture.

We present 4D Point Splatting (4DPS), a physics-based differentiable renderer that fits an explicit moving scene to a real capture, with velocity as an estimated and differentiable parameter. Like prior radar renderers (Armouti et al., 2026b;a; Chen et al., 2026), we initialize geometry from another sensor, here the capture's synchronized camera. We represent the scene as oriented points, each carrying an ITU-R P.2040 material (ITU-R, 2023), a surface normal and, for the body, a velocity, and we splat each point into range and Doppler bins. The camera fixes where each surface is present but not how strongly it reflects at millimeter wavelengths. That strength depends on the surface's material and on the angle between its normal and the radar, and it sets the amplitude of every return. We therefore fit the materials, normals and velocities by gradient descent until the rendered return matches the recording (Fig. 1, top). On HuPR, fitting raises the clutter-removed range–azimuth correlation from 0.270 for the camera-initialized render to 0.666. Our output is the raw complex return that a real radar records, for every chirp, antenna pair and range bin. We therefore apply the same digital signal processing (DSP) to rendered and real data, and obtain any representation, from range profiles to micro-Doppler, for any downstream model (Fig. 1, bottom). We validate each representation on two datasets with different array geometries and downstream models.

Introducing an explicit velocity into the forward model raises three challenges. The first challenge is the cost: existing differentiable radar renderers output one chirp per frame, so rendering Doppler would repeat that render for every chirp, a cost that grows linearly with the chirp count and rules out corpus-scale fitting. 4DPS instead splats each point into Doppler bins as well as range bins (Fig. 2). A moving point's return across a frame's chirps is a single tone, so its Doppler point spread function has a closed form. Each point therefore deposits into seven Doppler bins instead of being rendered 64 times. Second, ambiguity: velocity enters the forward model only through a phase term that wraps every 2π, so velocities one Doppler ambiguity apart (9 m/s on HuPR) render identically, and gradient descent from an arbitrary start lands in the nearest minimum rather than the true minimum. To address this, we initialize the joint velocities from the video's kinematics and learn only a residual, bounded by each joint's own speed. Third, supervision: one loss must train

Table 1: **Comparison of radar rendering and synthesis methods.** Signal: CIR or raw ADC. Mat.: explicit materials; Vel.: velocity estimated; Doppler: from subject motion; Real: validated on real captures; Edit: re-renderable scene; Code: released. Radar ports of neural fields and Gaussian splatting are omitted, because they supervise post-FFT power of a static scene.

Method	Repr.	Signal	Complex	Mat.	Vel.	Doppler	Real	Edit	Code
Differentiable radar inverse rendering									
mmIR (Armouti et al., 2026b)	Mesh	ADC	✓	✓	–	–	✓	✓	✓
3DPS (Armouti et al., 2026a)	Points	ADC	✓	✓	–	–	✓	✓	✓
Human micro-Doppler simulation and synthesis									
RF-Genesis (Chen & Zhang, 2024)	Implicit	ADC	✓	–	–	✓	✓	–	✓
HybridSim (Xiong et al., 2026)	Mesh	ADC	✓	✓	–	✓	✓	✓	–
4DPS (Ours)	Points	ADC	✓	✓	✓	✓	✓	✓	✓

both the static scene and the person, who holds a small fraction of a frame's energy. A loss over the whole frame lets a large relative error on the person hide behind the scene's return, and a loss on the person alone gives the scene no gradient. We therefore compare the range–azimuth–elevation magnitude at two slow-time views: each frame's first chirp, which carries the static scene, and its Doppler planes, which carry the motion. Each view is compared over the whole frame and over a window on the person (Eq. 4). Solving these three problems lets us fit entire corpora for the first time, with no manual input. We fit every usable sequence of both datasets, 235 on HuPR and 100 on RT-Pose, at about 5.7 and 12.9 minutes a sequence on one H100.

We set the new SOTA in 4D range–azimuth–elevation–Doppler (RAED) tensor rendering and reconstruction, with median Pearson correlations of 0.940 on range–azimuth and 0.988 on micro-Doppler on HuPR, and 0.830 and 0.966 on RT-Pose (Table 2). To test whether a model can learn from these renders, we train each dataset's official pose model on real or on rendered data, four seeds each, and test it on both (Tables 3 and 11). Trained on our renders alone, the models keep 0.59 of their real-trained accuracy on HuPR and 0.76 on RT-Pose. By contrast, models trained on RF-Genesis renders of the same motion fail on real data, at 1.8 AP on HuPR and 127.6 cm MPJPE on RT-Pose. Because our rendered-trained models score far higher on renders than on recordings, the gap that remains is realism rather than missing information. A comparison of the two models' internal representations shows that they compute the same features from the backbone onward (Sec. 4.3).

Contributions. In summary, we make the following contributions: 1) 4DPS, to our knowledge the first differentiable renderer of human micro-Doppler: oriented points carrying material, normal *and* a velocity estimated per joint for the first time, splatted in closed form into range and Doppler bins, rendering the raw complex return so any representation follows; 2) a range–Doppler point splatting framework, with a closed-form Doppler point spread function and Doppler-resolved supervision, that makes corpus-scale inverse rendering tractable across multiple public corpora and different array geometries for the first time; 3) the first sim2real analysis of radar inverse rendering on downstream models: the real/rendered train–test matrix for radar human pose on two datasets with the official models, and an interpretability analysis showing that rendered- and real-trained models learn the same features; and 4) release of the renderer, the fitting chain, every fitted scene and rendered cube, the mesh-derived RT-Pose labels, the retrained models and the evaluation suite, on acceptance.

2 Related Work

Differentiable radar rendering. Radar-domain renderers fit explicit scene parameters to captures. Differentiable ray tracers (Hoydis et al., 2023; Hofmann et al., 2025; Chen et al., 2025; 2023) do so on synthetic or known-geometry scenes. mmIR (Armouti et al., 2026b) fits the raw ADC of real captures and recovers materials, normals and antenna patterns, and 3DPS (Armouti et al., 2026a) replaces mmIR's Monte Carlo sampling with closed-form point splatting. Every one of these renderers reconstructs a static scene. Its forward model evaluates one phase per primitive per frame, so it has no slow time, the axis along which Doppler and velocity are defined. 4DPS renders that axis in closed form and estimates velocity, so it fits a moving scene (Table 1).

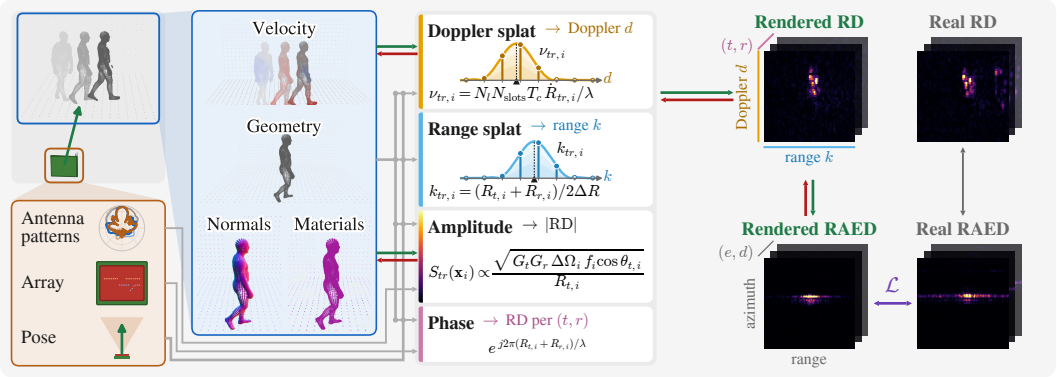

Figure 2: **4DPS overview.** Camera-initialized oriented point sets for a static scene and a moving person, carry normals, materials and, for the person, velocities. For every antenna pair, 4DPS splats each point into its Doppler and range bins with an amplitude and a carrier phase, forming the RAED tensor that $\mathcal{L}$ compares with the recording. Green: forward pass; red: gradients; gray: fixed inputs.

Radar data synthesis. Radar ports of neural fields and Gaussian splatting (Huang et al., 2024; Borts et al., 2024; Kung et al., 2025; Zhang et al., 2026; Rafidashti et al., 2025) supervise on post-FFT power and discard phase, so they cannot render the phase that encodes micro-Doppler. DART (Huang et al., 2024) is the only one with a Doppler axis, and its Doppler comes from ego-motion, not from an articulated subject. Motion-capture simulators (Sundar Ram & Ling, 2008; Vishwakarma et al., 2022) produce human micro-Doppler but run forward only, so their residual against a real capture is never quantified. mmGPE (Xue et al., 2023) narrows that residual with a learned refiner on the simulator's output, without fitting the scene. Learned generators (Chen & Zhang, 2024) leave the scene implicit in network weights, where nothing can be fitted or edited. Concurrent with this work, HybridSim (Xiong et al., 2026) ray traces a physics-learning hybrid over eight sequences from one single-chip radar, trained end to end with the downstream task it feeds. Solving both at once is a weaker requirement than ours, because a renderer supervised by the task's loss need only produce features that task finds useful. We instead fit the radar return itself and score it against the recording, so our renders must stand on their own and then transfer to a downstream model that never saw them during fitting. We do not carry it as a baseline, because it releases no code and its captures, collected for mmMesh (Xue et al., 2021), are unreleased. 4DPS fits an explicit scene, velocity included, to a real capture.

3 Method

4DPS starts from two camera-initialized point sets, a static scene and a moving person (Fig. 2; RT-Pose's scene also uses its LiDAR). Each scene point carries its own ITU-R P.2040 material and normal. Each person point carries its own normal, the whole person shares one material, and its points move with 14 joint velocities skinned onto them. 4DPS renders the complex return of a TDM MIMO FMCW radar, for every chirp, transmitter–receiver pair and range bin, by splatting both point sets in custom CUDA kernels. Static points are splatted along range only, and moving points along both range and Doppler. Sec. 3.1 states the differentiable radar rendering model, from which Sec. 3.2 derives the splatting and Sec. 3.3 the objective and its learned parameters.

3.1 Radar Forward Model

Consider an FMCW radar with N_{TX} time-division-multiplexed transmitters and N_{RX} receivers, carrier wavelength λ, range bin width ΔR, $K = 256$ range bins and $N_l = 64$ loops per frame. In each loop every transmitter fires one chirp in its own slot, so with N_{slots} slots and a chirp interval T_c the loop period is $N_{\text{slots}} T_c$ (216 μs on HuPR, 780 μs on the cascade).

The rendering equation. Following prior radar inverse renderers (Armouti et al., 2026b;a), we take the bistatic radar equation (Skolnik et al., 1980), replace the cross-section by a spatially varying BSDF f and integrate over the receiver-centered hemisphere Ω_{rx}, giving the received power as a

rendering equation,

$$P_r \;=\; \frac{P_t\,\lambda^2}{(4\pi)^2}\int_{\Omega_{rx}} \frac{G_t(\boldsymbol{\omega}_t)\,G_r(\boldsymbol{\omega}_r)\,f(\boldsymbol{\omega}_t,\mathbf{x},\boldsymbol{\omega}_r)\,\cos\theta_t\,V(\mathbf{x})}{R_t^2(\mathbf{x})}\,\mathrm{d}\omega_r, \tag{1}$$

with G_t, G_r the antenna gains, R_t the path length to the transmitter, $\cos\theta_t = |\mathbf{n}\cdot\boldsymbol{\omega}_t|$ and $V \in \{0,1\}$ visibility. We discretize the hemisphere on an oriented point set $\{(\mathbf{x}_i, \mathbf{n}_i, \Delta\Omega_i, \boldsymbol{\theta}_i, \mathbf{v}_i)\}_{i=1}^{N}$, each point subtending solid angle $\Delta\Omega_i$ at the receiver and carrying a material, with a velocity $\mathbf{v}_i$ on person points only. The Riemann sum cancels the surface Jacobian against $\Delta\Omega_i$ and leaves a sum of per-point amplitudes, carrier phase attached,

$$S_{tr}(\mathbf{x}_i) \;=\; C\,\frac{\sqrt{G_t(\boldsymbol{\omega}_{t,i})}}{R_{t,i}}\,\sqrt{G_r(\boldsymbol{\omega}_{r,i})\,\Delta\Omega_i}\,\sqrt{f_i\cos\theta_{t,i}}\;e^{\,j2\pi(R_{t,i}+R_{r,i})/\lambda}, \tag{2}$$

with C a constant that the learned gain absorbs (App. A.1). We set $\Delta\Omega_i = 1$ for every point, because a cosine solid-angle weight darkened the limbs' oblique sides, and the learned gains and materials carry the level. Nothing is sampled. Each point contributes one path per transmitter–receiver pair.

3.2 Splatting Static and Moving Points

Static scene: range splat. A renderer of one chirp can push the range FFT inside the point sum, so that each point becomes the Hann window's known response at its fractional range bin and the range profile is splatted instead of simulated (Armouti et al., 2026a), and we use it for the static scene. A static point returns the same sample in every chirp, and the real front end removes that slow-time mean before its Doppler transform, so the static return lies in no Doppler plane and is supervised through each frame's first chirp (Sec. 3.3).

Moving person: range–Doppler splat. A per-chirp renderer never forms the slow-time axis, the N_l loops of a frame along which Doppler is measured. For the person we therefore splat along slow time as well (Fig. 2), because the DFT over loops is linear in a path's slow-time samples just as the range FFT is linear in its fast-time samples. Transmitter t samples a given path once a loop, so the path's slow-time samples are a single tone. A point whose bistatic path length changes at the rate $\dot{R}_{tr,i} = \mathbf{v}_i \cdot (\mathbf{u}_{t,i} + \mathbf{u}_{r,i})$, with $\mathbf{u}_{t,i}, \mathbf{u}_{r,i}$ the unit vectors from the antennas to the point, therefore sits at the fractional Doppler bin $\nu_{tr,i} = N_l\,N_{\text{slots}}\,T_c\,\dot{R}_{tr,i}/\lambda$, because consecutive loops of one transmitter are one loop period apart. The rendered range–Doppler product of one antenna pair is a two-axis splat,

$$\mathrm{RD}_{tr}[d,k] \;=\; g\sum_{i=1}^{N} b_i\,V_i\,S_{tr}(\mathbf{x}_i)\,e^{\,j\phi_{t,i}}\,\Phi_K(k-k_{tr,i})\,\Phi_{N_l}(d-\nu_{tr,i}), \tag{3}$$

with g a learned receiver gain per radar, b_i the person's learned per-joint scattering gains skinned to point i (1 for scene points), V_i the point's visibility V in that frame, $k_{tr,i} = (R_{t,i} + R_{r,i})/2\Delta R$ the fractional range bin, $\phi_{t,i}$ the phase the transmitter's own slot contributes, and Φ_K, Φ_{N_l} the Hann window's exact response on the two axes. Static points enter the same sum with $\nu_{tr,i} = 0$. We truncate Φ_K to $L = 15$ range taps and Φ_{N_l} to 7 Doppler taps, so a person path costs 105 accumulations instead of a 64×256 cube per antenna pair, and its backward pass costs the same. App. A.3 derives this closed form from the per-chirp signal it replaces.

Material model. The BSDF f_i is a closed-form Jones BSDF (Wei et al., 2024; Armouti et al., 2026a), evaluated from a six-parameter ITU-R P.2040 material vector (ITU-R, 2023): a coherent specular lobe and a small-perturbation lobe, blended by the ITU roughness gates and resolved in a local s/p polarization basis (App. A.2). The scene's materials are per point, while the whole person shares one six-vector, because per-point materials let the fit darken the limbs (App. B.2).

Visibility and occlusion. The binary V of Eq. (1) is a per-frame ray cast that runs in both directions. A body point is switched off when its segment to the radar is blocked by the posed mesh or by the scene. A scene point is switched off for the frames in which the body stands between it and the radar (App. B.1). This matters more here than for a static renderer, because the shadow moves with the person. An unmasked scene point renders a return that stays constant over the frame's loops while the recording interrupts it, and the zero-Doppler plane is most sensitive to that error.

Noise model. A clutter-removed range–Doppler map is mostly empty, and its empty cells sit at the receiver's noise floor instead of zero. A renderer that models no noise predicts exactly zero there, so an L_2 on linear magnitude would charge the fit for every empty cell and the fit would answer by manufacturing power. This is a Doppler problem, because a range–azimuth image is filled by the static scene while most of a resolved range–Doppler product is floor. We therefore make the floor $\mathcal{N}(d, k)$ part of the forward model, a log scale and a range shape over a prior taken from the recording's own off-person power, added to the rendered power wherever magnitudes are compared. It enters the loss only through the two plane terms, which fit its level, and the unoccupied range–Doppler level of Table 2 ($L_U = 0.92/0.92$, 0.96 on the cascade) tests whether it is right.

Benefits of 4D Point Splatting. *Render any DSP representation:* Eq. (3) renders the raw complex return a real radar records, so the range, azimuth, elevation and Doppler FFTs of the real data's DSP turn one fitted scene into any representation a downstream model reads (App. A.1). *Differentiable velocity:* $\nu_{tr,i}$ depends on the point's velocity, so the Doppler axis carries gradient through the skinning to the joint velocities, and the radar, not the camera, settles what the limbs did. *Corpus-scale:* that per-path cost is independent of N_l, so a whole corpus is affordable (Sec. 3.3).

3.3 Optimization and Implementation

Our objective function evaluates different projections of the linear RAED magnitude tensor:

$$\mathcal{L} \;=\; \underbrace{\mathcal{L}_{\mathrm{RAE},0} + \mathcal{L}_{\mathrm{RAE,pl}}}_{\text{global}} \;+\; \underbrace{\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0} + \mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}}}_{\text{windowed on the person}}, \tag{4}$$

a normalized L_2 on $|\mathrm{RAE}|$ (App. B.2), taken at two slow-time views and two scopes. The first-chirp term $\mathcal{L}_{\mathrm{RAE},0}$ keeps the static clutter return, so it is the only path by which the static scene receives material gradient. Because it is a magnitude, it supervises ADC amplitude and inter-element phase without fitting absolute phase noise. The plane term $\mathcal{L}_{\mathrm{RAE,pl}}$ supervises the 63 non-DC Doppler planes, which only the moving person reaches, so a velocity error that leaves the range profile intact still moves energy between planes. The windowed pair restricts both to the person's spatial window and normalizes over it, because a frame-wide divisor lets a large relative error on the person hide behind the scene's energy. Those cells hold the occupied range–Doppler region and the micro-Doppler signature. All four terms apply FFTs sized by the radar's virtual aperture and chirp count, irrespective of the downstream model's beamformer, unlike HybridSim (Xiong et al., 2026), which trains its renderer end-to-end with that model.

What we optimize. For the static scene we optimize per-point materials and normals. For the person we optimize per-point normals, one material six-vector and 14 joint velocities, each the video's kinematic velocity times a learned per-joint scale plus a bounded, band-limited residual, skinned to the points so that a joint the video calls still stays still. For the radar we optimize its gains and the noise floor (Table 5).

Scale and runtime. A chirp is rendered once per physical transmitter–receiver pair, so fitting HuPR is $235 \times 600 \times 64 \times 24$ chirps and RT-Pose $100 \times 299 \times 64 \times 192$, for a total of 584 million chirps. Each is fitted for 400 epochs, so the fit renders 234 billion chirps forward and as many again backward, without ever forming a slow-time cube. That is 63,000 times the data 3DPS (Armouti et al., 2026a) fits, at 43.9 GPU-hours on H100s (App. H.2).

4 Evaluation

Sec. 4.1 fits 235 sequences from HuPR (Lee et al., 2023) and 100 from RT-Pose (Ho et al., 2025), two public human-sensing corpora that retain slow time and a colocated camera. HuPR records raw ADC from two orthogonal single-chip radars, and RT-Pose from one cascade (App. B.1). Sec. 4.2 measures sim2real with the pose model each dataset released, MSCSA-PRGCN (Lee et al., 2023) and HRRadarPose (Ho et al., 2025), retrained from scratch for four seeds at equal optimizer steps. Each model keeps its own metric, because the labels differ. HuPR's 2D image keypoints admit OKS AP but no metric distance, and RT-Pose's 3D joints admit MPJPE in centimeters. RT-Pose's joints come from the mesh that drives the render (App. E). No public simulator takes a real capture as input, so RF-Genesis (Chen & Zhang, 2024) is driven forward from the same registered motion (App. C.3), and our unfit render is the floor. Sec. 4.3 compares real- and rendered-trained models.

Table 2: **Rendering fidelity.** Medians over sequences (HuPR pools both radars) of the Pearson correlation with the recording: range–angle (ρ; ρ_c clutter-removed), occupied range–Doppler cells (ρ_M), and micro-Doppler centroid, spread and spectrogram (ρ_c, ρ_σ, ρ_S); L_U: rendered-to-real noise-floor level. Bold: best per section. App. D gives the other DSP representations.

	Range–angle		Range–Doppler		Micro-Doppler		
Method	$\rho \uparrow$	$\rho_c \uparrow$	$\rho_M \uparrow$	$L_U \to 1$	$\rho_c \uparrow$	$\rho_\sigma \uparrow$	$\rho_S \uparrow$
HuPR, two single-chip radars							
RF-Genesis (Chen & Zhang, 2024)	0.134	0.339	0.518	0.05	0.820	0.063	0.671
4DPS (unfit)	0.878	0.270	0.537	0.57	0.930	0.283	0.754
4DPS (Ours)	**0.940**	**0.666**	**0.829**	**0.92**	**0.993**	**0.966**	**0.988**
RT-Pose, cascade, one-person sequences							
RF-Genesis	0.119	0.221	0.324	0.09	0.750	0.042	0.394
4DPS (unfit)	0.648	0.181	0.291	0.58	0.803	0.185	0.515
4DPS (Ours)	**0.830**	**0.470**	**0.695**	**0.96**	**0.973**	**0.704**	**0.966**

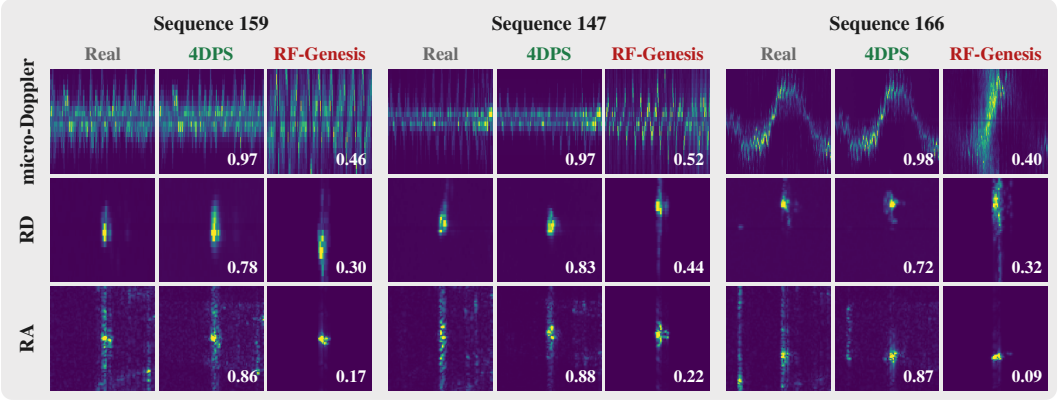

Figure 3: **Real, 4DPS and RF-Genesis on three RT-Pose test sequences.** Rows: micro-Doppler for the full sequence; typical range–Doppler frame; and typical range–azimuth frame; the number is the render's sequence-level correlation with the recording. Activities: 159 stand and lift a leg, 147 stand and wave, 166 walk and wave. See App. J for more details.

4.1 Rendering Fidelity

Does 4DPS reproduce the real capture, on every DSP representation? The fitted render reaches 0.940 range–azimuth and 0.988 micro-Doppler correlation on HuPR, and 0.830 and 0.966 on RT-Pose (Table 2). Fig. 3 shows three RT-Pose sequences beside the recording and RF-Genesis, and Tables 7–10 give every column. RF-Genesis, driven by the same motion, reaches only 0.134 and 0.671 on HuPR and 0.119 and 0.394 on RT-Pose. It renders the person alone, with no room, so it misses the static scene that fills a real range–azimuth map, and it keeps the body's bulk motion (micro-Doppler centroid 0.820 on HuPR) but loses the limbs' spread around it (0.063 against 0.966 fitted), below even our unfit render, because its 800 body scatterers are reassigned between ray casts and nothing fits their motion to the recording. Our control is the unfit render, the same chain with the velocity field trained for zero epochs, which reaches 0.878, 0.754, 0.648 and 0.515. In a paired, one-sided comparison, the control scores lower on every sequence, for both representations and datasets. The one exception is range–azimuth on one HuPR sequence's vertical board. Camera motion alone already tracks the body's bulk radial movement, so the control keeps a Doppler-centroid correlation of 0.930 on HuPR and 0.803 on RT-Pose. What it loses is the spread around that centroid, where the limbs' micro-motion sits. Its spread correlation falls to 0.283 on HuPR and 0.185 on RT-Pose, against 0.966 and 0.704 for the fitted render. The radar fit therefore recovers the motion the video gets wrong. The cascade is the harder case, because it resolves elevation, the axis the scaffold reconstructs least well. The still recordings score lowest.

Table 3: **Pose estimation downstream model training: real vs rendered data.** On RT-Pose we train HRRadarPose (Ho et al., 2025), and on HuPR MSCSA-PRGCN (Lee et al., 2023). Every row is tested on real data, so the rendered-trained rows measure sim-to-real transfer. Mean (std) over four seeds trained from scratch. Table 11 gives the full train–test matrix.

		RT-Pose (cm ↓)			**HuPR** (AP ↑)		
Train	**Test**	MPJPE	MRPE	Abs-MPJPE	AP	AP_{50}	AP_{75}
Real	Real	16.67 (0.42)	13.14 (0.87)	19.07 (0.52)	65.6 (1.7)	96.9 (0.2)	78.0 (3.1)
RF-Genesis	Real	127.59 (48.32)	106.48 (45.00)	148.88 (24.86)	1.8 (3.5)	8.5 (16.1)	0.0 (0.1)
4DPS	Real	21.86 (1.33)	17.06 (0.98)	25.45 (1.69)	38.4 (2.9)	80.6 (3.2)	30.1 (4.3)

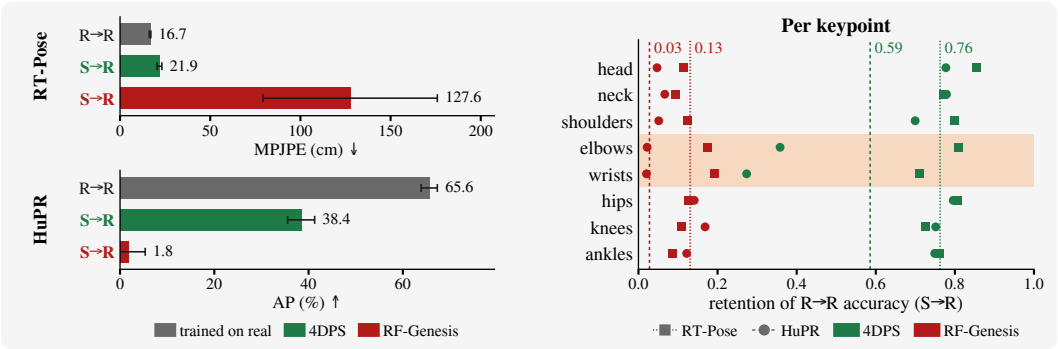

Figure 4: **The sim-to-real gap.** Left: real-data MPJPE on RT-Pose (↓) and AP on HuPR (↑) after training on real data, 4DPS or RF-Genesis (four seeds, ±1 std; bars on their true scale). Right: retention of real-trained accuracy per keypoint group (S→R over R→R, the inverse for MPJPE); on HuPR the 4DPS deficit sits in the arms (shaded). Table 3 gives every metric.

4.2 Training Pose Models on Rendered Data

How much of real-trained accuracy survives training on renders? Every row of Table 3 is tested on real data. Trained on 4DPS renders alone, each model keeps 0.59 of its real-trained accuracy on HuPR (38.4 (2.9) against 65.6 (1.7) AP) and 0.76 on RT-Pose (21.86 (1.33) against 16.67 (0.42) cm). A model trained on RF-Genesis renders of the same motion transfers almost nothing. It reaches 1.8 (3.5) AP on HuPR and 127.59 (48.32) cm MPJPE on RT-Pose (Fig. 4, left). Both renderers see the same motion, so the difference lies in what each makes of it. An RF-Genesis render holds the person alone and carries little of the limbs' micro-motion (Sec. 4.1), so a model trained on it never learns to separate a moving body from the room around it. The 4DPS renders do carry the pose information a model needs. Tested on held-out renders, a 4DPS-trained model scores 59.6 (1.4) AP and 15.71 (1.38) cm, close to the real-trained model on recordings (Table 11). Its drop on recordings therefore comes from the remaining rendered–real differences, which the model first meets at test time. Per keypoint, the drop concentrates in the arms (Fig. 4, right). On HuPR the wrists fall from 22.2 and 25.1 AP to 8.0 and 4.7, while hips, knees and the head keep 0.73 to 0.80 of their real-trained AP. The arms are the body's smallest and fastest-moving parts, and their rendered motion comes from the video's kinematics, corrected only within a bounded residual (Sec. 3.3). HuPR's coarse azimuth resolution also gives the fit little signal to correct them, so they depart most from the recording.

HuPR is the harder downstream task of the two, although its renders score higher on fidelity (Table 2), for three reasons. i) One sequence gives the renderer 921,600 chirps against RT-Pose's 3,674,112, a factor of 0.25. ii) Its radars also resolve the lateral axes far more coarsely. Eight azimuth elements on a single chip put the broadside resolution near 14.3°, against 1.3° for the cascade's 86 slots. Each HuPR board resolves one axis only, azimuth on the horizontal and elevation on the vertical, and carries 2 elevation bins where the cascade carries 8. The labels, meanwhile, are keypoints in a camera image whose pixels subtend about 0.2°. The model must place joints on a grid some seventy times finer than its own sensor resolves laterally. Coarse angles make the camera-

Table 4: **Do rendered- and real-trained models learn the same features?** Each cell is the real–rendered distance over the real–real distance between seeds (1.0: as close as two real-trained models), with its 95% interval; the readout columns are error ratios. Bold: the interval lies above the $1.2\times$ margin fixed before the analysis. †: uninformative, as an untrained network is about as close. HuPR's middle layers are undefined (dead temporal merges); App. F gives every layer and the error-consistency analysis.

	Feature similarity					*Linear readout*	
	CKA first	CKA middle	CKA backbone	CKA logits	Procrustes every layer	Rendered-trained	Untrained backbone
RT-Pose	**1.35**, 1.17	0.98–1.10	0.98†	0.99†	0.98–1.05	1.16 [1.09, 1.28]	1.25
HuPR	1.15 1.37	undefined	1.08† 1.10†	1.37 [1.08, 1.70]	1.05–1.17	1.30 [1.20, 1.38]	2.50

to-radar extrinsic harder to estimate as well, and our fit is sensitive to that registration, as prior radar inverse renderers are (Armouti et al., 2026b;a) (App. I).

4.3 Comparing Real- and Rendered-Trained Models

Does a model trained on renders learn the same features as one trained on real data? We compare the two models' activations layer by layer with two standard similarity measures, orthogonal Procrustes distance (Williams et al., 2021) and debiased linear CKA (Kornblith et al., 2019; Nguyen et al., 2021; Murphy et al., 2024), because no single measure wins across settings (Klabunde et al., 2025). Each score is divided by the same score between two of five real-trained models, which differ only in seed, so 1.0 means as similar as two real-trained models, and we fixed 1.2 as the margin before running the analysis (Table 4). The Procrustes ratio stays between 0.98 and 1.05 at every layer on RT-Pose and between 1.05 and 1.17 on HuPR, and CKA clears the margin with confidence only at RT-Pose's first layer (1.35), where the rendered and real inputs themselves differ most. Training on renders therefore teaches the same features as training on real data, consistent with Sec. 4.2, where the rendered-trained model's drop comes from rendered–real differences in its input. Features that look alike can still serve pose estimation differently, so we also freeze each model's feature extractor and fit a new linear pose readout on real data. On HuPR, features learned from renders give 1.30 times the keypoint error of real-trained features, against 2.50 times for an untrained network. Therefore, they carry most but not all of the pose information. At the deepest layers even an untrained network scores about as close (†), so similarity there is no evidence either way. App. F gives every layer, the error-consistency analysis and the protocol.

5 Conclusion

We presented 4DPS, a physics-based differentiable renderer that fits an explicit moving scene to a real radar capture. Camera-initialized oriented points carry an ITU-R P.2040 material, a normal and a velocity, and are splatted in closed form into range and Doppler bins on the TDM timing of a MIMO array. The closed-form Doppler splat makes the fit cheap enough to run unattended over all 235 HuPR and 100 RT-Pose sequences, and because 4DPS renders the raw complex return, the real data's DSP turns it into any representation. 4DPS sets the state of the art on range–azimuth, range–Doppler and micro-Doppler. On HuPR it reaches median correlations of 0.666 on clutter-removed range–azimuth and 0.988 on micro-Doppler, against 0.339 and 0.671 for RF-Genesis driven by the same motion. Pose models trained on our renders alone keep 0.59 of their real-trained accuracy on HuPR and 0.76 on RT-Pose, while models trained on RF-Genesis renders do not transfer. The rendered- and real-trained models also compute the same features, so the remaining gap comes from rendered–real differences in their input. **Limitations and future work.** Velocity stays within a bounded residual of the video kinematics. The background is static and rendered along single-bounce paths, and we model one person per sequence. RT-Pose's labels come from the body mesh that drives the render, and no public simulator takes a capture as input, so our baselines run forward (App. I). Scoring edited scenes against real recordings is the next step (App. G).

Reproducibility Statement

Code, configurations, the fitted checkpoints of every sequence and the scripts behind every table will be released. Both datasets are public (HuPR (Lee et al., 2023), RT-Pose (Ho et al., 2025)). The forward model is stated in Sec. 3 with every learned quantity in Table 5; the preprocessing chain, the shipped runtime design and the downstream protocol are in the supplement (Secs. B.1, H.2, B.2, F).

Use of AI Statement

We used large language models to implement and edit code, clean and reformat the datasets, assist in writing, and give feedback on the research methodology and experiments. The authors reviewed all AI-assisted work and take full responsibility for the content of this paper.

References

Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C. Miller. Smart homes that monitor breathing and heart rate. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 837–846, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702200. URL https://doi.org/10.1145/2702123.2702200.

Sizhe An and Umit Y. Ogras. Mars: mmwave-based assistive rehabilitation system for smart healthcare. *ACM Trans. Embed. Comput. Syst.*, 20(5s), September 2021. ISSN 1539-9087. doi: 10.1145/3477003. URL https://doi.org/10.1145/3477003.

Sizhe An, Yin Li, and Umit Ogras. mri: Multi-modal 3d human pose estimation dataset using mmwave, rgb-d, and inertial sensors. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27414–27426. Curran Associates, Inc., 2022. doi: 10.52202/068431-1988. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/af9c9c6d2da701da5a0acf91ec217815-Paper-Datasets_and_Benchmarks.pdf.

Adnan Armouti, Yixuan Gao, and Rajalakshmi Nandakumar. 3d point splatting for mmwave radar novel view synthesis, 2026a. URL https://arxiv.org/abs/2609.11894.

Adnan Armouti, Yixuan Gao, and Rajalakshmi Nandakumar. mmir: Frequency-space inverse rendering for 3d millimeter-wave radar adc synthesis. In Paolo Favaro, Zuzana Kukelova, Atsuto Maki, Anna Rohrbach, Konrad Schindler, and Federico Tombari (eds.), *Computer Vision – ECCV 2026*, pp. 148–166, Cham, 2026b. Springer Nature Switzerland. ISBN 978-3-032-37359-5.

David Borts, Erich Liang, Tim Broedermann, Andrea Ramazzina, Stefanie Walz, Edoardo Palladin, Jipeng Sun, David Brueggemann, Christos Sakaridis, Luc Van Gool, Mario Bijelic, and Felix Heide. Radar fields: Frequency-space neural scene representations for fmcw radar. In *ACM SIGGRAPH 2024 Conference Papers*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705250. doi: 10.1145/3641519.3657510. URL https://doi.org/10.1145/3641519.3657510.

Anjun Chen, Xiangyu Wang, Shaohao Zhu, Yanxu Li, Jiming Chen, and Qi Ye. mmbody benchmark: 3d body reconstruction dataset and analysis for millimeter wave radar. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 3501–3510, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548262. URL https://doi.org/10.1145/3503161.3548262.

Chuhan Chen, Tianshu Huang, Akarsh Prabhakara, Chaithanya Kumar Mummadi, Zhongxiao Cong, Anthony Rowe, Matthew O'Toole, and Deva Ramanan. Radarsim: Simulating single-chip radar via multimodal neural fields. In *2026 International Conference on 3D Vision (3DV)*, pp. 1258–1268, 2026. doi: 10.1109/3DV69130.2026.00124.

V.C. Chen, F. Li, S.-S. Ho, and H. Wechsler. Micro-doppler effect in radar: phenomenon, model, and simulation study. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):2–21, 2006. doi: 10.1109/TAES.2006.1603402.

Xingyu Chen and Xinyu Zhang. Rf genesis: Zero-shot generalization of mmwave sensing through simulation-based data synthesis and generative diffusion models. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, SenSys '23, pp. 28–42, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704147. doi: 10.1145/3625687.3625798. URL https://doi.org/10.1145/3625687.3625798.

Xingyu Chen, Xinyu Zhang, Qiyue Xia, Xinmin Fang, Chris Xiaoxuan Lu, and Zhengxiong Li. Differentiable radio frequency ray tracing for millimeter-wave sensing, 2023. URL https://arxiv.org/abs/2311.13182.

Xingyu Chen, Jianrong Ding, Kai Zheng, Xinmin Fang, Xinyu Zhang, Chris Xiaoxuan Lu, and Zhengxiong Li. Invertwin: Solving inverse problems via differentiable radio frequency digital twin, 2025. URL https://arxiv.org/abs/2508.14204.

Han Cui, Shu Zhong, Jiacheng Wu, Zichao Shen, Naim Dahnoun, and Yiren Zhao. Milipoint: A point cloud dataset for mmwave radar. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 62713–62726. Curran Associates, Inc., 2023. doi: 10.52202/075280-2739. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c60468eca9cd0b0083f0ff9d0aeb171a-Paper-Datasets_and_Benchmarks.pdf.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1556–1568. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/0c0bf917c7942b5a08df71f9da626f97-Paper.pdf.

Junqiao Fan, Yunjiao Zhou, Yizhuo Yang, Xinyuan Cui, Jiarui Zhang, Lihua Xie, Jianfei Yang, Chris Xiaoxuan Lu, and Fangqiang Ding. M4human: A large-scale multimodal mmwave radar benchmark for human mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42836–42846, June 2026.

Xiangyu Gao, Guanbin Xing, Sumit Roy, and Hui Liu. Ramp-cnn: A novel neural network for enhanced automotive radar object recognition. *IEEE Sensors Journal*, 21(4):5119–5132, 2021. doi: 10.1109/JSEN.2020.3036047.

Robert Geirhos, Kristof Meding, and Felix A. Wichmann. Beyond accuracy: quantifying trial-by-trial behaviour of cnns and humans by measuring error consistency. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 13890–13902. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/9f6992966d4c363ea0162a056cb45fe5-Paper.pdf.

Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. Through fog high-resolution imaging using millimeter wave radar. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11461–11470, 2020. doi: 10.1109/CVPR42600.2020.01148.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151, 2022. doi: 10.1109/CVPR52688.2022.00509.

Rongxiao Guo and Qingchao Chen. Dghmesh: A large-scale dual-radar mmwave dataset and generalization-focused benchmark for human mesh reconstruction, 2026. URL https://arxiv.org/abs/2604.22827.

Yuan-Hao Ho, Jen-Hao Cheng, Sheng Yao Kuan, Zhongyu Jiang, Wenhao Chai, Hsiang-Wei Huang, Chih-Lung Lin, and Jenq-Neng Hwang. Rt-pose: A 4d radar tensor-based 3d human pose estimation and localization benchmark. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 107–125, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73036-8.

Nikolai Hofmann, Vanessa Wirth, Johanna Bräunig, Ingrid Ullmann, Martin Vossiek, Tim Weyrich, and Marc Stamminger. Inverse rendering of near-field mmwave mimo radar for material reconstruction. *IEEE Journal of Microwaves*, 5(2):356–372, 2025. doi: 10.1109/JMW.2025.3535077.

Jakob Hoydis, Faycal Ait Aoudia, Sebastian Cammerer, Merlin Nimier-David, Nikolaus Binder, Guillermo Marcus, and Alexander Keller. Sionna rt: Differentiable ray tracing for radio propagation modeling. In *2023 IEEE Globecom Workshops (GC Wkshps)*, pp. 317–321, 2023. doi: 10.1109/GCWkshps58843.2023.10465179.

Tianshu Huang, John Miller, Akarsh Prabhakara, Tao Jin, Tarana Laroia, Zico Kolter, and Anthony Rowe. Dart: Implicit doppler tomography for radar novel view synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24118–24129. IEEE, 2024.

Tianshu Huang, Akarsh Prabhakara, Chuhan Chen, Jay Karhade, Deva Ramanan, Matthew O'Toole, and Anthony Rowe. Towards foundational models for single-chip radar. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24655–24665, 2025. doi: 10.1109/ICCV51701.2025.02286.

ITU-R. Effects of building materials and structures on radiowave propagation above about 100 MHz. Recommendation P.2040-3, International Telecommunication Union, aug 2023. URL https://www.itu.int/rec/R-REC-P.2040/en.

Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. Resi: A comprehensive benchmark for representational similarity measures. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 18701–18740, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2eef1f75516b0cfd3e944345e5f88c08-Paper-Conference.pdf.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Andrew Kramer, Kyle Harlow, Christopher Williams, and Christoffer Heckman. Coloradar: The direct 3d millimeter wave radar dataset. *Int. J. Rob. Res.*, 41(4):351–360, April 2022. ISSN 0278-3649. doi: 10.1177/02783649211068535. URL https://doi.org/10.1177/02783649211068535.

Pou-Chun Kung, Skanda Harisha, Ram Vasudevan, Aline Eid, and Katherine A. Skinner. Radarsplat: Radar gaussian splatting for high-fidelity data synthesis and 3d reconstruction of autonomous driving scenes. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27596–27606, 2025. doi: 10.1109/ICCV51701.2025.02562.

Shih-Po Lee, Niraj Prakash Kini, Wen-Hsiao Peng, Ching-Wen Ma, and Jenq-Neng Hwang. Hupr: A benchmark for human pose estimation using millimeter wave radar. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5704–5713, 2023. doi: 10.1109/WACV56688.2023.00567.

Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: ubiquitous gesture sensing with millimeter wave radar. *ACM Trans. Graph.*, 35(4), July 2016. ISSN 0730-0301. doi: 10.1145/2897824.2925953. URL https://doi.org/10.1145/2897824.2925953.

Teck-Yian Lim, Spencer A. Markowitz, and Minh N. Do. Radical: A synchronized fmcw radar, depth, imu and rgb camera data dataset with low-level fmcw radar signals. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):941–953, 2021. doi: 10.1109/JSTSP.2021.3061270.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), November 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Alex Murphy, Joel Zylberberg, and Alona Fyshe. Correcting biased centered kernel alignment measures in biological and artificial neural networks, 2024. URL https://arxiv.org/abs/2405.01012.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Mahan Rafidashti, Ji Lan, Maryam Fatemi, Junsheng Fu, Lars Hammarstrand, and Lennart Svensson. Neuradar: Neural radiance fields for automotive radar point clouds. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2479–2489, 2025. doi: 10.1109/CVPRW67362.2025.00234.

M. Mahbubur Rahman, Ryoma Yataka, Sorachi Kato, Pu Wang, Peizhao Li, Adriano Cardace, and Petros Boufounos. Mmvr: Millimeter-wave multi-view radar dataset and benchmark for indoor perception. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 306–322, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72986-7.

Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Pérez. Raw high-definition radar for multi-task learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17000–17009, 2022. doi: 10.1109/CVPR52688.2022.01651.

Ignacio Roldan, Andras Palffy, Julian F. P. Kooij, Dariu M. Gavrila, Francesco Fioranelli, and Alexander Yarovoy. A deep automotive radar detector using the radelft dataset. *IEEE Transactions on Radar Systems*, 2:1062–1075, 2024. doi: 10.1109/TRS.2024.3485578.

Argha Sen, Anirban Das, Swadhin Pradhan, and Sandip Chakraborty. A dataset for multi-intensity continuous human activity recognition through passive sensing, 2024. URL https://arxiv.org/abs/2407.21125.

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 *SIGGRAPH Asia 2024 Conference Papers*, SA '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711312. doi: 10.1145/3680528.3687565. URL https://doi.org/10.1145/3680528.3687565.

Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, mmNets '19, pp. 51–56, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369329. doi: 10.1145/3349624.3356768. URL https://doi.org/10.1145/3349624.3356768.

Merrill Ivan Skolnik et al. *Introduction to radar systems*, volume 3. McGraw-hill New York, 1980.

Tieshuai Song, Jiandong Ye, Ao Guo, Guidong He, and Bin Yang. Radarrgbd a multi-sensor fusion dataset for perception with rgb-d and mmwave radar, 2025. URL https://arxiv.org/abs/2505.15860.

Shobha Sundar Ram and Hao Ling. Simulation of human microdopplers using computer animation data. In *2008 IEEE Radar Conference*, pp. 1–6, 2008. doi: 10.1109/RADAR.2008.4720816.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

Alexander Vilesov, Pradyumna Chari, Adnan Armouti, Anirudh Bindiganavale Harish, Kimaya Kulkarni, Ananya Deoghare, Laleh Jalilian, and Achuta Kadambi. Blending camera and 77 ghz radar sensing for equitable, robust plethysmography. *ACM Trans. Graph.*, 41(4), July 2022. ISSN 0730-0301. doi: 10.1145/3528223.3530161. URL https://doi.org/10.1145/3528223.3530161.

Shelly Vishwakarma, Wenda Li, Chong Tang, Karl Woodbridge, Raviraj Adve, and Kevin Chetty. Simhumalator: An open-source end-to-end radar simulator for human activity recognition. *IEEE Aerospace and Electronic Systems Magazine*, 37(3):6–22, 2022. doi: 10.1109/MAES.2021.3138948.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pp. 195–206, Goslar, DEU, 2007. Eurographics Association. ISBN 9783905673524.

Fei Wang, Yizhe Lv, Mengdie Zhu, Han Ding, and Jinsong Han. Xrf55: A radio frequency dataset for human indoor action analysis. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 8(1), March 2024. doi: 10.1145/3643543. URL https://doi.org/10.1145/3643543.

Jiangtao Wei, Yixiang Luomei, Xu Zhang, and Feng Xu. Learning surface scattering parameters from sar images using differentiable ray tracing. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024. doi: 10.1109/TGRS.2024.3459620.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4738–4750. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf.

Zhi Wu, Dongheng Zhang, Chunyang Xie, Cong Yu, Jinbo Chen, Yang Hu, and Yan Chen. Rfmask: A simple baseline for human silhouette segmentation with radio signals. *IEEE Transactions on Multimedia*, 25:4730–4741, 2023. doi: 10.1109/TMM.2022.3181455.

Weitao Xiong, Tianyu Liu, Peng Li, Kok Chung Chua, Toa Chean Khim, Pu Wang, and Hongfei Xue. Hybridsim: A physics-learning hybrid digital twin for mmwave human sensing, 2026. URL https://arxiv.org/abs/2607.15806.

Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. mmmesh: towards 3d real-time dynamic human mesh construction using millimeter-wave. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, pp. 269–282, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384438. doi: 10.1145/3458864.3467679. URL https://doi.org/10.1145/3458864.3467679.

Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. Towards generalized mmwave-based human pose estimation through signal augmentation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ACM MobiCom '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399906. doi: 10.1145/3570361.3613302. URL https://doi.org/10.1145/3570361.3613302.

Jianfei Yang, He Huang, Yunjiao Zhou, Xinyan Chen, Yuecong Xu, Shenghai Yuan, Han Zou, Chris Xiaoxuan Lu, and Lihua Xie. Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing. In A. Oh, T. Naumann, A. Globerson, K. Saenko,

M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 18756–18768. Curran Associates, Inc., 2023. doi: 10.52202/075280-0822. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3baf7a39d07e9f4f1e258a412df94521-Paper-Datasets_and_Benchmarks.pdf.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 21875–21911. Curran Associates, Inc., 2024. doi: 10.52202/079017-0688. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/26cfdcd8fe6fd75cc53e92963a656c58-Paper-Conference.pdf.

Dongsheng Yuan, Xie Zhang, Weiying Hou, Sheng Lyu, Yuemin Yu, Luca Jiang-Tao Yu, Chengxiao Li, and Chenshu Wu. Octonet: A large-scale multi-modal dataset for human activity understanding grounded in motion-captured 3d pose labels. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-0469. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/14950adea25005f89545688fe97fe5ea-Paper-Datasets_and_Benchmarks_Track.pdf.

Bo Zhang, Zimeng Zhou, Boyu Jiang, and Rong Zheng. Super: Seated upper body pose estimation using mmwave radars. In *2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 181–191, 2024. doi: 10.1109/IoTDI61053.2024.00020.

Jiarui Zhang, Zhihao Li, Chong Wang, and Bihan Wen. Rf4d:neural radar fields for novel view synthesis in outdoor dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15387–15397, June 2026.

Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, and Andrew Markham. Cubelearn: End-to-end learning for human motion recognition from raw mmwave radar signals. *IEEE Internet of Things Journal*, 10(12):10236–10249, 2023. doi: 10.1109/JIOT.2023.3237494.

TABLE OF CONTENTS

A Radar and Rendering Theory **17**

A.1 From the Radar Equation to the Rendering Equation 17

A.2 The BSDF as Evaluated 18

A.3 Closed-Form Doppler-Plane Amplitudes 18

B Implementation Details **20**

B.1 Preprocessing 20

B.2 Training 21

C Datasets and Baseline **24**

C.1 Dataset survey 24

C.2 Baseline survey 25

C.3 RF-Genesis as a baseline 25

D Diagnostic Fidelity Metrics **26**

E Downstream Results in Full **29**

F Representation Analysis Protocol **32**

G Scene Editing **35**

G.1 Separability of the scene and the person 35

G.2 Novel motions 35

H Ablations, Runtime, and Peak Memory **40**

H.1 Ablations 40

H.2 Runtime and Peak Memory 43

I Limitations **46**

J Additional Qualitative Results **46**

A RADAR AND RENDERING THEORY

A.1 FROM THE RADAR EQUATION TO THE RENDERING EQUATION

Radar equation. The power received over the single-bounce path TX $\rightarrow$ $\mathbf{x}$ $\rightarrow$ RX is the bistatic radar equation (Skolnik et al., 1980),

$$P_r \;=\; \frac{P_t\, G_t(\boldsymbol{\omega}_t)\, G_r(\boldsymbol{\omega}_r)\, \lambda^2\, \sigma}{(4\pi)^3\, R_t^2\, R_r^2}, \tag{5}$$

with P_t the transmit power, G_t, G_r the antenna gain patterns and σ the bistatic radar cross-section. As in prior radar inverse rendering (Armouti et al., 2026b;a), we replace σ by a spatially varying BSDF $f(\boldsymbol{\omega}_t, \mathbf{x}, \boldsymbol{\omega}_r)$ and integrates over the receiver-centered hemisphere Ω_{rx}, converting the surface element to receiver solid angle through $\mathrm{d}\omega_r = |\mathbf{n}\cdot\boldsymbol{\omega}_r|\,\mathrm{d}A/R_r^2$, which gives Eq. (1) of the main text. Transmitter t sits at $\mathbf{p}_t$ and receiver r at $\mathbf{p}_r$; for a surface point $\mathbf{x}$ with unit normal $\mathbf{n}$, $R_t(\mathbf{x}) = \|\mathbf{x}-\mathbf{p}_t\|$, $R_r(\mathbf{x}) = \|\mathbf{x}-\mathbf{p}_r\|$, and $\boldsymbol{\omega}_t, \boldsymbol{\omega}_r$ are the unit directions from $\mathbf{x}$ toward the transmitter and the receiver.

Surface to solid angle. A distributed surface replaces the point target's cross-section by a BSDF f through $\sigma = 4\pi \int f\, \cos\theta_t \cos\theta_r\, \mathrm{d}A$, with $\cos\theta_r = |\mathbf{n}\cdot\boldsymbol{\omega}_r|$, so that the power received from the surface element $\mathrm{d}A$ at $\mathbf{x}$ is $V\, P_t\, G_t G_r \lambda^2 f \cos\theta_t \cos\theta_r\, \mathrm{d}A/((4\pi)^2 R_t^2 R_r^2)$. That element subtends $\mathrm{d}\omega_r = \cos\theta_r\, \mathrm{d}A/R_r^2$ at the receiver, which absorbs both $\cos\theta_r$ and R_r^2 and leaves Eq. (1), the form mmIR integrates by Monte Carlo and 3DPS discretizes deterministically.

Riemann sum on the oriented point set. Each point i of the set would be assigned the solid angle its surface patch subtends at the receiver. In the configuration behind every number in this paper the slot is unit for every point. For the scene, the per-point materials absorb the level. For the person, the SMPL mesh's sampling density is not the view's, and a cosine weight darkens every oblique surface, the limbs' sides among them, so the level is left to the area scale and the learned gains. The sum of per-point powers is then $P_r = \sum_i P_t \lambda^2 G_t G_r f_i \cos\theta_{t,i}\, \Delta\Omega_i\, V_i/((4\pi)^2 R_{t,i}^2)$, whose amplitude form, the quantity Eq. (3) of the main text calls $S_{tr}(\mathbf{x}_i)$, is

$$\begin{aligned} S_{tr}(\mathbf{x}_i) \;&= C\, \frac{\sqrt{G_t(\boldsymbol{\omega}_{t,i})}}{R_{t,i}}\, \sqrt{G_r(\boldsymbol{\omega}_{r,i})\, \Delta\Omega_i}\, \sqrt{f_i(\boldsymbol{\omega}_{t,i}, \boldsymbol{\omega}_{r,i}) \cos\theta_{t,i}} \\ &\quad\times\; e^{\,j\,2\pi\,(R_{t,i}+R_{r,i})/\lambda}, \end{aligned} \tag{6}$$

with the carrier phase attached and C a constant that is immaterial because the absolute level is learned. The solid angle $\Delta\Omega_i$ is unit for every point, because the scene's per-point materials and the person's area scale α and learned joint gains carry the level instead. The renderer evaluates exactly this product: the per-(point, TX) factor $\sqrt{G_t}/R_t$, the per-(point, RX) factor $\sqrt{G_r\,\Delta\Omega_i}$, the per-path factor $\sqrt{f_i \cos\theta_{t,i}}$ and the constant $C = \sqrt{P_t\, L\, \lambda^2}/(4\pi)$ times the ADC full-scale conversion, with L the fixed antenna loss ($P_t = 20\,\mathrm{mW}$ and $L = 0.40$ in the configuration, both immaterial under the learned gain g). Back-facing paths ($\cos\theta_t$ or $\cos\theta_r$ at the numerical floor) and paths switched off by the visibility masks are given exactly zero weight; the renderer is double-sided, so a normal's sign flips rather than culls.

The splat. The Hann window's exact response, the Φ of Eq. (3), is

$$\Phi_K(\delta) \;=\; \tfrac{1}{2}\, D_K(\delta) - \tfrac{1}{4}\, D_K(\delta-1) - \tfrac{1}{4}\, D_K(\delta+1), \tag{7}$$

with $D_K(\delta) = \sum_{n=0}^{K-1} e^{-j2\pi\delta n/K}$ the K-point Dirichlet kernel. Pushing the range FFT inside the point sum (Armouti et al., 2026a) makes the complex range profile of one physical pair the coherent sum over the visible points

$$S_{tr}[k] \;=\; g \sum_{i=1}^{N} b_i V_i\, S_{tr}(\mathbf{x}_i)\, \Phi_K(k - k_{tr,i}) + C_{tr}[k], \tag{8}$$

where $C_{tr}[k]$ is the antenna-coupling profile estimated from the recording, non-zero on the first 15 bins only, and b_i is 1 for scene points and $\sqrt{\alpha\gamma_i}$ for person points, with α the body's area scale and γ_i a per-point scattering gain skinned from 14 learned joint gains. The rendered profile has the shape of the recording's demultiplexed ADC after the range FFT, $(N_{\mathrm{TX}}, N_{\mathrm{RX}}, K)$ with $K = 256$, and no

channel or bin is dropped. The range FFT of a Hann-windowed chirp is, per point, the window's response at the point's fractional bin, Eq. (7) with $K = 256$, so the ADC-then-FFT pipeline and the direct scatter-add of Eq. (8) are the same computation up to the truncation to $L = 15$ taps. The bin index $k_{tr,i} = (R_{t,i} + R_{r,i})/2\Delta R$ with $\Delta R = cf_s/(2SK)$ (4.29 cm on HuPR, 4.51 cm on the cascade) is rounded only to choose the 15 receiving bins; the carrier phase and the PSF's fractional argument use the unrounded path length, and the PSF is read from a 1024-entry table over the fractional bin by linear interpolation. Gate: one point target through the real and the rendered chain agrees to 1.5×10^{-7} (float32 rounding) on range profiles, loop range–azimuth, the Doppler cube and the Doppler planes; the 15-tap truncation is the one approximation left, 7.3×10^{-4} of the peak, the same on both datasets (`qa window-check`).

Products. Every product is a Fourier transform of the rendered return, taken with the shared windows of Sec. 4 on the real and the rendered side alike. The range–Doppler map of one antenna pair is the windowed N_l-point DFT over loops after removing the slow-time mean,

$$\mathrm{RD}_{tr}[d,k] \;=\; \sum_{l=0}^{N_l-1} w_D[l]\,\big(S[l,t,r,k] - \bar{S}_{tr}[k]\big)\,e^{-j2\pi d\,l/N_l}, \tag{9}$$

with $\bar{S}_{tr}[k]$ the mean over l and w_D the Doppler window, shifted so that $d = N_l/2$ is DC; Eq. (3) evaluates this directly, without forming $S[l,t,r,k]$. Range–azimuth is the angle FFT over the azimuth virtual array formed from the same cube, $\mathrm{RA}[l,k,a] = \mathcal{F}_v\{w_A[v]\,S[l,t(v),r(v),k]\}$; range–azimuth–elevation adds an elevation FFT over the array's elevation rows, two on HuPR and seven on the cascade, so every transmitter contributes; and the range–azimuth–Doppler cube $\mathrm{RAD}[d,k,a]$ is Eq. (9) applied along l of RA. The micro-Doppler spectrogram column of a frame is the person-masked power marginal of that cube,

$$\mathrm{MD}[d] \;=\; \sum_{(k,a)\in\mathcal{M}} \big|\mathrm{RAD}[d,k,a]\big|^2, \tag{10}$$

with $\mathcal{M}$ the person's (range, angle) support projected from its point cloud. The angle FFT sizes are the lossless ones for the array (16 azimuth bins on HuPR's 8-slot row, 86 on the cascade; 2 and 8 elevation bins), and the angle axes are untapered inside the trainer and the downstream models, because the downstream models beamform untapered: a rendered wall that read once the recording's in the tapered view read ten times it untapered.

A.2 The BSDF as Evaluated

The per-path BSDF is the closed-form BSDF of Armouti et al. (2026a), unchanged, in the coherent spatially varying BSDF (CSVBSDF) formulation of Wei et al. (2024). With $\boldsymbol{\theta}_i = (\varepsilon'_r, \varepsilon''_r, \sigma_h, \ell_c, \tau, d)$ mapped to their physical ranges by sigmoid (τ) and shifted softplus (the rest), the value the kernel returns for the path (i,t,r) is

$$f_i \cos\theta_{t,i} \;=\; \Big[\,\tau_{\text{eff}}\,|\Gamma_{\mathbf{h}}|^2\,K_{\text{KA}} + (1-\tau_{\text{eff}})\,|\Gamma_{\mathbf{n}}|^2\,K_{\text{SPM}}\Big]\cos\theta_{t,i}, \tag{11}$$

where $K_{\text{KA}} = D\,G/(4\cos\theta_t\cos\theta_r)$ is a GGX microfacet lobe (Walter et al., 2007) with Smith masking and roughness $\alpha = \sqrt{4\pi\sigma_h/\lambda}$ (clamped to $[0.05, 0.95]$); $K_{\text{SPM}} = \frac{\kappa}{4\pi\sinh\kappa}\,e^{\kappa(\cos\Delta-1)}\,\eta_\varepsilon$ is a von Mises–Fisher lobe about the mirror direction with concentration $\kappa = \sqrt{\ell_c/\lambda}/(1+3\sigma_h/\ell_c)$ clamped to $[0.5, 10]$ and a permittivity-contrast factor $\eta_\varepsilon = \mathrm{clamp}((|\varepsilon'_r - 1| + \varepsilon''_r)/5,\ 0.2,\ 1)$; $\Gamma_{\mathbf{h}}$ and $\Gamma_{\mathbf{n}}$ are the multilayer-slab Fresnel coefficients (ITU-R, 2023) of thickness d and permittivity $\varepsilon_r = \varepsilon'_r - j\varepsilon''_r$, evaluated at the half-vector and at the macro normal respectively, each projected from the transmit polarization through the local s/p Jones basis onto the receive polarization and squared; and τ_{eff} is the learned blend τ gated by the ITU validity sigmoids in $2\pi\ell_c/\lambda$, $2\pi\sigma_h/\lambda$ and the incidence angle. The scene's normals are the third column of a per-point unit quaternion; the person's are a free unit vector per visible point per frame.

A.3 Closed-Form Doppler-Plane Amplitudes

The slow-time cube the trainer never forms. Under TDM, transmitter t fires in slot s_t of every loop, so loop l of transmitter t is acquired at

$$t_{l,t} \;=\; \big(l\,N_{\text{slots}} + s_t\big)\,T_c, \tag{12}$$

with T_c the chirp interval and N_{slots} the slots per loop (HuPR: $T_c = 72\,\mu$s, 3 slots, a 216 μs loop; the cascade: 65 μs, 12 slots, 780 μs), and a person point moving with velocity $\mathbf{v}_i$ changes its bistatic path length to (t, r) at the rate

$$\dot{R}_{tr,i} \;=\; \frac{\mathrm{d}}{\mathrm{d}t}\big(R_{t,i} + R_{r,i}\big) \;=\; \mathbf{v}_i \cdot \mathbf{u}_{t,i} + \mathbf{v}_i \cdot \mathbf{u}_{r,i}, \tag{13}$$

with $\mathbf{u}_{t,i} = (\mathbf{x}_i - \mathbf{p}_t)/R_{t,i}$ and $\mathbf{u}_{r,i} = (\mathbf{x}_i - \mathbf{p}_r)/R_{r,i}$, positive when the point recedes, so its contribution to loop l carries the phase ramp $e^{\,j2\pi \dot{R}_{tr,i} t_{l,t}/\lambda}$. The rendered product of a frame is therefore the complex slow-time cube

$$\begin{aligned} S[l,t,r,k] \;&= g \sum_{i \in \text{scene}} b_i V_i S_{tr}(\mathbf{x}_i)\, \Phi_K(k - k_{tr,i}) \;+\; C_{tr}[k] \\ &\quad +\; g \sum_{i \in \text{person}} b_i V_i S_{tr}(\mathbf{x}_i)\, \Phi_K(k - k_{tr,i})\, e^{\,j2\pi \dot{R}_{tr,i}\, t_{l,t}/\lambda}, \end{aligned} \tag{14}$$

of shape $(N_l, N_{\text{TX}}, N_{\text{RX}}, K)$, exactly the shape of the recording's demultiplexed, range-transformed ADC, for every physical TX–RX pair. Because Eq. (12) places each transmitter at its own intra-loop offset, a moving point accumulates the inter-TX phase progression that a real TDM capture must compensate before angle estimation, and Doppler is coupled to whichever lateral axes the array resolves. Range migration and the change of the BSDF within the frame's slow-time span (13.8 ms on HuPR, 49.9 ms on the cascade) are neglected; the carrier phase uses every physical transmitter time.

The trainer never forms the cube of Eq. (14). For a person path the slow-time samples are one tone: with Eq. (12), $t_{l,t} = t_{0,t} + l\,T_{\text{loop}}$, so

$$e^{\,j2\pi \dot{R}_{tr,i} t_{l,t}/\lambda} \;=\; e^{\,j\phi_0}\, e^{\,j\omega l}, \quad \phi_0 = \tfrac{2\pi}{\lambda} \dot{R}_{tr,i}\, t_{0,t}, \quad \omega = \tfrac{2\pi}{\lambda} \dot{R}_{tr,i}\, T_{\text{loop}}, \tag{15}$$

and, with $\nu = N_l\, \omega/2\pi$ the path's fractional Doppler bin, the windowed, mean-removed DFT of Eq. (9) over l evaluates at unshifted bin d' to

$$\begin{aligned} A_{d'}(\nu) \;=\; e^{\,j\phi_0} \Big[&\tfrac{1}{2} D_{N_l}(d' - \nu) - \tfrac{1}{4} D_{N_l}(d' - 1 - \nu) \\ &- \tfrac{1}{4} D_{N_l}(d' + 1 - \nu) - m_{d'}\, D_{N_l}(-\nu) \Big], \end{aligned} \tag{16}$$

the Hann Doppler PSF (Eq. (7) with $K = N_l$) plus the mean removal folded into the row, $m_{d'} = \frac{1}{2}$ at $d' = 0$, $-\frac{1}{4}$ at $d' = \pm 1$ and 0 elsewhere, which is what makes the rendered planes agree with a real chain that removes the mean *before* it windows. The loop-0 anchor plane the range–azimuth term reads is $A_0 = e^{\,j\phi_0}$. The rate gradient $\partial A_{d'}/\partial \dot{R}$ is the same expression with the Dirichlet kernel's derivative, and both are evaluated in float32 with ν and ϕ_0 formed in float64 and reduced to an integer bin and a fraction first. Gate (`qa doppler-check`, both datasets, windowed and not): against a float64 reference that is itself 4×10^{-14} from the 64-loop sum and whose gradient is 8×10^{-9} from finite differences, the closed-form kernels are at 7.3×10^{-7} (amplitudes) and 5.4×10^{-7} (rate gradient).

Truncation. With the window the Doppler response is compact (the first sidelobe is 2.8×10^{-2} of the peak in amplitude at 2.5 bins and the envelope falls by a factor of eight per octave, in bin units whatever N_l is), so each path deposits into the L_d planes around its own fractional bin, as the range PSF is cut at 15. Measured on the point-target gate, the truncation of the planes is 7.3×10^{-4} at 15 taps, 3.5×10^{-3} at 9, 7.7×10^{-3} at 7 and 2.3×10^{-2} at 5; $L_d = 7$ is the default, and at 400 epochs on sequence 183 the 7-tap and exact arms agree (spectrogram $+0.941$ / $+0.933$, energy $+0.884$ / $+0.861$, range–Doppler occupied $+0.399$ / $+0.394$, the same learned floor). A path whose 7 planes fall outside a view's rendered support deposits nothing there.

B Implementation Details

B.1 Preprocessing

HuPR scaffold. Frames 0, 150, 300 and 450 of the sequence are read from the released RGB, passed in one batch through Depth-Anything-V2 (Large, metric) (Yang et al., 2024), the annotated person's bounding box (padded by 12%) is cut out of each depth map, and the maps are back-projected under an assumed 40° horizontal field of view at a pixel stride of 2, voxel-downsampled at 4 cm per view and 5 cm after merging, and given PCA normals (0.18 m radius, 50 neighbors) oriented toward the radar. Two person-free real maps are formed per board: the initial material fit trains on the pixelwise median of the four sampled frames' loop-mean $|\mathrm{RA}|$, and the range alignment scores against the per-cell 25th percentile of loop-mean $|\mathrm{RA}|$ over every fourth frame of the sequence (the subject is the only mover and occupies any one cell in a minority of frames, so a low quantile is the static scene with the person gone). The range translation is searched on a 2 cm grid over ± 2.4 m and refined at 2 mm, every candidate scored at once on the GPU (0.3 s): the score is the min–max-normalized Pearson correlation of the shifted cloud's (range, angle) occupancy against the person-free map, mean over both boards, restricted to the cloud's own angle bins (the camera sees $\pm 20^\circ$ of the radar's $\pm 75^\circ$) and to ranges beyond 0.5 m. Lateral offset and height are not fitted because the map does not identify them: competing optima 0.6–1.4 m apart lie within 0.03 of the best score on every sequence, while the range optimum is unique: sequence 256: -0.34 m (r 0.25 to 0.47); 38: -0.82 m (0.16 to 0.48); 102: -0.73 m (0.16 to 0.50). The point-cloud initialization of Armouti et al. (2026a) then reduces the aligned cloud to $N = 20{,}000$ points by a field-of-view restriction, cosine-weighted resampling toward boresight and farthest-point sampling, and fits per-point materials and normals to the person-free static $|\mathrm{RA}|$ of both boards for 75 iterations (2–3 s), the trainer refining both for its 400 epochs.

RT-Pose scaffold. Eight annotated radar frames spread over the sequence supply Livox LiDAR frames, each densified by the left camera's Depth-Anything depth rescaled to that frame's LiDAR by the median LiDAR/predicted depth ratio at the projected LiDAR pixels; the person is cut from each frame by the registered body mesh of that frame (points within 0.25 m of a vertex, the floor kept), so that what the person touches, such as a chair, survives. The clouds are taken to the radar frame by the rig extrinsic, merged at a 3 cm voxel and given PCA normals over a 0.3 m radius (0.5 m is needed for the LiDAR alone, whose scan lines are 0.4 m apart on the far wall). The extrinsic is the identity: the radar sits at the LiDAR to within 0.03 m laterally by two independent person measurements, and a ± 0.5 m translation search against the real static map cannot identify a lateral shift (correlation 0.189 to 0.235 while walking the height to the edge of its range), so it is not refitted; the annotation frame's origin is 0.2 m to the left of the LiDAR and the board is rotated 180° about boresight, both verified from the data. The cascade frames are decoded with TI's phase and amplitude calibration.

Antenna coupling. Per radar, the coherent mean over every third frame and over loops of the range-windowed profiles at the first 15 range bins, per azimuth virtual channel. Nothing else lives within those bins (0.5 m on the cascade), so the coherent mean is the board's own static near-field return: it explains 99.98% of the near-bin power on HuPR sequence 256 and 99.95% on cascade sequence 183 (the fraction of the near-bin energy the mean explains). It is added to the rendered profiles in training and rendering, and the absolute-gain initialization leaves those bins out.

Mesh recovery. GVHMR (Shen et al., 2024) runs in static-camera mode in its own environment, started with the scene stage so that its $\sim$30 s overlap: the frames held in memory, YOLO tracking of the largest persistent track, ViTPose and the HMR2 features in half precision on half-resolution crops, giving the $6{,}890$ SMPL vertices (Loper et al., 2015) per camera frame. RT-Pose's true focal length (25.65 mm full-frame equivalent) is passed; HuPR's is estimated from the image. The mesh is a deterministic function of the frames (the same video twice is bitwise; a lossy re-encode moves it 3–5 mm median, a tenth of a range bin).

Camera-to-radar placement. RT-Pose: camera to LiDAR by the shipped stereo calibration, LiDAR to radar by the extrinsic; the camera and LiDAR file belonging to radar frame f has stem $3f - 4$ on 85 sequences and $2f - 2$ on 17, a rig clock fitted on the annotation's pairs and exact on every annotated frame (a constant offset fed the person stage a camera frame up to 20 s off on the sequences it did not hold). HuPR: the camera at the rig, co-boresighted, camera (right, down, forward) to radar $(x, -z, y)$, with the assumed 40° intrinsics at 512 px for the joints' image projection.

Per-frame radar registration. For every frame, the clutter-removed range–azimuth cube (slow-time mean removed) is averaged over loops and squared-summed over angle into a range profile of moving power; its peak between bins 10 and 200, refined by a parabola over the two neighbors, is the radar's range to the person, and the fraction of the profile's energy within 5 bins of the peak is the detection confidence (a rig of two boards contributes the confidence-weighted mean). The per-frame ratio r_f = radar range / GVHMR centroid range is interpolated across frames whose confidence is below 0.35, median-filtered over ± 7 frames, clamped to $[0.5, 2]$, flagged where it leaves the confident frames' interquartile range widened three times (GVHMR's track is wrong there, and the radar's range is followed), and every unflagged run is polished by a quadratic Savitzky–Golay filter 51 frames wide, because the median filter's steps would otherwise add spurious radial motion to the cloud. r_f moves the body along its centroid's camera ray to r_f times its range; the body keeps GVHMR's size, since GVHMR's error is its depth, not the body's size. The range search is confined to the camera's angular sector, on both mirrors of the beamformer's angle axis, and to a range window bounded by the scale's own range. Azimuth: one registration per sequence maps the mesh's azimuth onto the radar's moving return, $\sin\theta_{\text{radar}} = a + b\sin\theta_{\text{mesh}}$, the scale b fitted only where the mesh spans azimuth (sine std ≥ 0.10, $r \geq 0.90$) and a yaw alone otherwise; the offsets are 3 to 5 degrees on the RT-Pose walking sequences, and the labels inherit the session's lateral scale. The median scale is 0.707 (IQR 0.699–0.717) on HuPR sequence 256, i.e. GVHMR 30% long; RT-Pose's GVHMR runs about 7% short.

Joints, velocities, motion frames. The 14 joints in HuPR's order come from the SMPL joint regressor; their camera-derived velocities are a first-derivative Savitzky–Golay filter, 11 frames (1.1 s at 10 Hz), quadratic. Motion frames for evaluation are those whose pose speed, the 90th percentile over the 14 smoothed joint speeds, is at least 0.15 m/s (1.5 cm per frame), the same threshold on every sequence, pre-registered. The person's material is initialized as one shared six-parameter vector fitted for one epoch on all frames against the loop-0 $|\mathrm{RA}|$ with the scene model fixed, together with the body's area scale; the trainer refines the shared vector (per-vertex materials let the fit darken the limbs), and the level calibration of Sec. 3.3 sets the area scale after training.

Visibility. For each frame the body's 13,776 SMPL triangles are binned by their (azimuth, elevation) box as seen from the radar on a 128×128 grid, padded one cell either side, so a ray asks only the triangles of its cell; the hit test is unchanged, so the result is the brute-force one bit for bit. Person mask: field of view (a boresight cosine above 0.1761, range in $(0.5, 9.4)$ m), then a Möller–Trumbore test of the open segment from the vertex (nudged 1 mm toward the radar, past its own faces) to the radar center against the body's triangles, and against the scaffold triangles on the radar side of the body. Scene shadow mask: every scaffold point inside the body's angular silhouette box and beyond its nearest vertex is tested against the body's triangles; the rest are visible without a test. Both masks are computed for the whole sequence on the device and enter the renderer as a per-point gain of 0 or 1, never as a change of the point set. With the shadow mask the wall behind the subject dims on all six products as it does in the real capture.

Windowing. Range and Doppler take the periodic Hann $w[n] = \frac{1}{2} - \frac{1}{2}\cos(2\pi n/N)$, whose DFT is exactly three bins and whose response is Eq. (7); the symmetric $\mathrm{hanning}(N)$ is not this window and would leave a $1/N$ mismatch between the two sides. The array axis takes $\mathrm{hanning}(N+2)[1:-1]$, because an aperture of 8 or 86 slots cannot afford to zero an element. Before the windows only the rendered range response was Hann-shaped, an inconsistency of 73% on a point target. The cascade's 144 azimuth virtual channels (9 TX $\times$ 16 RX) land on 86 half-wavelength grid slots, 58 of them holding two channels which are averaged, then windowed and zero-padded to a 128-bin angle FFT; HuPR's 8 fill 8 slots, padded to 64. Preprocessing runs in 10–18 s per sequence: the ray casting, voxel downsampling, PCA normals, person ranges and masks batched on the GPU, the ADC decoded once into memory and shared by the stages.

B.2 Training

Level calibration. The area scale α is not learned by the objective. It is set once, after training, so that the rendered person has the recording's level in the transform the downstream pose model of Sec. 4 reads: on every fourth training frame the render's Doppler-summed magnitude is formed in that model's own beamformer, and α is the value at which the median ratio of rendered to recorded peak inside the pelvis's window (± 0.5 m, $\pm 8°$) is one. This uses no labels and no downstream model

Table 5: **Learned and fixed quantities of the forward model.** Materials and normals are parameterized as in Armouti et al. (2026a). Learning rates: velocity coefficients 10^{-2}, person normals 10^{-3}, scene materials 10^{-2}, scene normals $5 \cdot 10^{-3}$, person material 10^{-3}, gains 10^{-2}, noise floor $3 \cdot 10^{-2}$ (Adam).

Quantity	Shape	Note
Learned		
joint velocity scales s_j, residual c_{jk}	(14), (bands, 14, 3)	residual bounded by $0.75\|\mathbf{v}_j^{\text{kin}}\|$; 3.5 Hz cap
person normals $\mathbf{n}_i$	(visible points, 3) per frame	one fused parameter over the training frames
person material $\boldsymbol{\theta}$	(6)	one six-vector for the body, initialized by the person stage
scene materials $\boldsymbol{\theta}_i$, normals	(N,6), (N,4)	normals as quaternions; $N = 20{,}000$
receiver gain g, joint gains γ_j	(1) per radar, (14)	log-parameterized; $\prod_j \gamma_j = 1$
noise floor $\mathcal{N}(d, k)$	log scale + 8 range-DCT coeff. per radar	on the recording's own off-person prior
Fixed		
positions (registered per frame), skinning weights W, the area scale α (level calibration, after training), antenna patterns, array geometry, TDM schedule, windows, PSF widths ($L = 15$ range, 7 Doppler), visibility masks v_i, coupling C_{tr}		

weights, it is fitted per sequence on training frames only, and it is the operation a radar engineer calls gain calibration.

The velocity field. The 14 joint velocities $\mathbf{v}_j(f)$ of frame f are the video-derived kinematic velocities $\mathbf{v}_j^{\text{kin}}(f)$ (Sec. 4) times a learned per-joint scale, plus a bounded residual,

$$\mathbf{v}_j(f) \;=\; s_j\,\mathbf{v}_j^{\text{kin}}(f) + b_j(f)\tanh\big(\mathbf{r}_j(f)/b_j(f)\big), \qquad b_j(f) = 0.75\,\|\mathbf{v}_j^{\text{kin}}(f)\|. \tag{17}$$

The residual $\mathbf{r}_j(f) = \sum_k c_{jk} B_k(f)$ lives in an orthonormal temporal DCT basis truncated at 420 of the sequence's 600 bands (209 of 299 on RT-Pose), a 3.5 Hz cap at the 10 Hz frame rate. Each point's velocity $\mathbf{v}_i = \sum_j W_{ij}\mathbf{v}_j$ is skinned from the joints by soft weights W (normalized Gaussian weights, $\sigma = 0.25$ m, over the three nearest joints).

The normalized L_2 form. Every term of Eq. (4) has the same shape, the normalized L_2 that prior radar inverse renderers apply to $|\mathrm{RA}|$ (Armouti et al., 2026b;a),

$$\mathcal{L}_\bullet \;=\; \frac{\mathrm{mean}_c\big(\,\hat{y}_\bullet[c] - y_\bullet[c]\,\big)^2}{\mathrm{mean}_c\; y_\bullet[c]^2}, \tag{18}$$

the divisor a detached constant per frame and radar. It never contains the prediction, so the terms stay plain L_2 and the absolute level remains supervised, and all terms exclude the near-field range bins below 0.5 m. The targets are $y_{\mathrm{RAE},0} = |\mathrm{RAE}[0, k, a, e]|$, the first loop with the static clutter kept, and $y_{\mathrm{RAE,pl}} = |\mathrm{RAED}[d, k, a, e]|$ for the 63 planes $d \neq N_l/2$, each with its own divisor and with the learned floor entering the rendered side under the square root, $\hat{y} = \sqrt{|\widehat{\mathrm{RAED}}|^2 + \mathcal{N}(d, k)}$, so the plane terms are what fit the floor. Both are formed from every transmitter. The windowed pair are the same two terms restricted to the registered person's box, the pelvis ± 0.6 m in range and ± 1.2 m across the line of sight at every elevation bin, and normalized over that box, because a frame-wide divisor lets a large relative error on the person hide behind the scene's energy. A frame renders only the Doppler bins whose real power at the person's range rows exceeds the frame's median ($\kappa = 1$) plus the motion band (± 1.1 m/s), and every rendered plane is supervised, including the empty bins whose zero targets keep the rendered person compact in Doppler.

Both datasets use the same flags: 400 epochs on all 600 (HuPR) or 299 (RT-Pose) frames, a constant learning rate for the first half and a cosine decay to 10^{-2} of it over the second, the plane targets in float16, the Doppler support at $\kappa = 1$, every term at unit weight, Adam with the rates of Table 5, the trainer's angle axes untapered, the level calibration after training in the consumer's view, the velocity residual bounded at 0.75 of the joint's kinematic speed, and the objective of Eq. (4). What differs is set by the recording, not tuned: HuPR renders 24 frames a step (two boards of 3×4

channels, 600 training frames, 420 of 600 DCT bands), the cascade 4 frames a step ($12 \times 16 = 192$ channels, the count set for a 24 GB card, 299 frames, 209 of 299 bands, the same 3.5 Hz cap); the motion band is the same ± 1.1 m/s on both, 13 bins on HuPR's 216 μs loop (0.1408 m/s per bin, ± 4.5 m/s unambiguous) and 53 on the cascade's 780 μs loop (0.0390 m/s per bin, ± 1.25 m/s). The plane targets are $63 \times 244 \times N_a$ per frame and radar (3.9 MB in float32 on HuPR), so they are stored in float16. Every fourth frame is scored every 10 epochs and the best-scoring snapshot is reported. Peak allocated device memory is 18.7 GiB on the cascade and 5.7 GiB on HuPR (median over sequences), and the fit's time per sequence is in App. H.2.

C Datasets and Baseline

C.1 Dataset survey

Table 6 records the mmWave human-sensing datasets we considered and why each was included or excluded, against six eligibility criteria that its caption defines.

Two failure modes dominate. The first is the on-chip CFAR and beamforming pipeline discarding slow time before release, which removes every dataset in the table's second section. The second is withholding camera imagery for privacy, a practice that is tightening rather than relaxing. Only RT-Pose (Ho et al., 2025) and HuPR (Lee et al., 2023) satisfy all six criteria.

One terminology caveat carries into the rest of the paper. "Raw" is used inconsistently in this literature, and HIBER (Wu et al., 2023), MMVR (Rahman et al., 2025), and M4Human (Fan et al., 2026) all describe beamformed tensors as raw radar data. Only ADC or I/Q samples preserve the per-chirp phase history that micro-Doppler is computed from.

Table 6: **Dataset survey. ST**: slow time retained, at least two chirps per frame. **ADC**: raw ADC or I/Q samples, not heatmaps or point clouds. **Cam**: colocated camera imagery released, not merely used to generate labels. **Pose**: human pose labels (keypoints or a body mesh). **Pub**: openly downloadable. **Code**: reference downstream model code released. ✓ satisfied, × not satisfied, ○ partially satisfied (access by request or agreement, a withheld or blurred modality, an expired mirror, chirps per frame undocumented, a released cube already transformed along range, angle and Doppler, or model code without training scripts or only from later work). Only RT-Pose and HuPR satisfy all six. Pretrained weights are not a criterion: RT-Pose publishes none, so we train its reference model ourselves (App. E).

Dataset	ST	ADC	Cam	Pose	Pub	Code
Selected						
RT-Pose (Ho et al., 2025)	✓	✓	✓	✓	✓	✓
HuPR (Lee et al., 2023)	✓	✓	✓	✓	✓	✓
No slow time (human sensing)						
MMVR (Rahman et al., 2025)	×	×	×	✓	✓	○
HIBER (Wu et al., 2023)	×	×	✓	✓	○	○
mmBody (Chen et al., 2022)	×	×	✓	✓	○	✓
M4Human (Fan et al., 2026)	×	×	✓	✓	×	✓
mm-Fi (Yang et al., 2023)	×	×	○	✓	✓	✓
mRI (An et al., 2022)	×	×	○	✓	✓	✓
OctoNet (Yuan et al., 2025)	×	×	✓	✓	✓	✓
MiliPoint (Cui et al., 2023)	×	×	×	✓	✓	✓
MARS (An & Ogras, 2021)	×	×	×	✓	✓	✓
RadHAR (Singh et al., 2019)	×	×	×	×	✓	✓
mmDoppler (Sen et al., 2024)	×	×	×	×	✓	✓
No camera released						
CubeLearn (Zhao et al., 2023)	✓	✓	×	×	✓	✓
ColoRadar (Kramer et al., 2022)	✓	✓	×	×	✓	×
Not (fully) public						
DGHMesh (Guo & Chen, 2026)	✓	✓	✓	✓	×	×
XRF55 (Wang et al., 2024)	○	○	✓	×	○	✓
mmMesh (Xue et al., 2021)	✓	✓	✓	✓	×	✓
SUPER (Zhang et al., 2024)	✓	✓	×	✓	×	×
EquiPleth (Vilesov et al., 2022)	○	✓	✓	×	○	✓
Not human sensing						
RaDICaL (Lim et al., 2021)	✓	✓	✓	×	✓	×
IQ-1M (Huang et al., 2025)	✓	✓	✓	×	✓	✓
RAMP-CNN (Gao et al., 2021)	✓	✓	✓	×	✓	✓
RadarRGBD (Song et al., 2025)	✓	○	✓	×	✓	×
RADIal (Rebut et al., 2022)	✓	✓	✓	×	✓	✓
RaDelft (Roldan et al., 2024)	✓	✓	✓	×	○	✓

C.2 BASELINE SURVEY

A row of Table 2 needs a raw complex signal for our array layout and chirp, driven by the recorded motion rather than by a class label or text, from code that runs. Of the published radar simulators, only RF-Genesis (Chen & Zhang, 2024) meets all of these requirements. mmGPE (Xue et al., 2023) and HybridSim (Xiong et al., 2026) release no code, and SimHumalator (Vishwakarma et al., 2022) renders one receive channel from a fixed library of eleven motion-capture activity classes, so it cannot take our motions or our arrays.

C.3 RF-GENESIS AS A BASELINE

What was run. We ran RF-Genesis's released code (upstream commit `3393af6`) on all 335 sequences that 4DPS renders, on both datasets, and no sequence failed. Each sequence is driven by the registered SMPL mesh that drives the 4DPS render, in the same radar frame, and its output is scored by the same code on the same pose-speed frames (App. D), so the two rows of Table 2 differ only in the renderer.

Changes to the released code. Each change is needed for the output to describe our radar and the recorded motion, and none adds a fitted parameter: (i) the camera and the spot light that stands for the transmitter sit at the radar board, along its boresight; (ii) the registered mesh's vertices replace the text-generated motion; (iii) the ray intersection is evaluated before the render call, which changes no value but lets the pipeline link on our driver; (iv) each chirp is timed at its slot in the recording's TDM schedule and fires its own transmitter, where upstream spread the chirps over the whole frame and so scaled every Doppler frequency by $3.6\times$; and (v) the interpolator reads the depth channel where it read intensity as depth.

Adapter choices. RF-Genesis interpolates motion at 30 fps, so we resample the registered mesh from the radar's 10 Hz frames by linear interpolation of its vertices. We give it no environment, because its only environment model is a text-prompted image that is not registered with the body.

Downstream training. The RF-Genesis arms of Table 3 use the models, protocol, splits and labels of the 4DPS arms. On RT-Pose one difference is needed: RF-Genesis's units are arbitrary and HRRadarPose normalizes its input with fixed constants, so without a gain its inputs reach the network as zeros. We therefore scale them by one gain, $G = 4.57 \times 10^8$, fixed before any training as the median, over 8 frames of each of the 44 training sequences, of the recording's moving-return level (the 99.9th percentile of the mean-removed cube) over RF-Genesis's (interquartile range 2.43×10^8 to 8.55×10^8). HuPR needs no gain, because its preprocessing normalizes each map from the data.

Limitations of the comparison. First, RF-Genesis renders no static scene, so its range–angle products hold the person alone while the recording holds the room. The range–azimuth ρ therefore also measures the missing room, and the clutter-removed columns (range–azimuth ρ_c, range–Doppler ρ_M and micro-Doppler) are the fair comparison, because they remove the static scene on both sides. Second, the released code is single-bounce diffuse ray casting, not the full system the RF-Genesis paper describes, and we run it as released apart from the changes above. Third, RF-Genesis has no absolute scale and no receiver noise, so for the columns that compare levels we multiply its render by one constant per radar and sequence, $g = (\sum |\mathrm{RA}_{\mathrm{real}}|^2 / \sum |\mathrm{RA}_{\mathrm{RF\text{-}Genesis}}|^2)^{1/2}$ over the loop-0 range–azimuth magnitude on the retained range bins of every frame, the rule that initializes 4DPS's own gain. The constant leaves every correlation and shape column unchanged, and because no noise is simulated, RF-Genesis's L_U stays far below 1. Fourth, it runs open loop: nothing is fitted to the capture, so its row measures a forward simulator driven by the recorded motion, not a competing fitted renderer (App. I).

D DIAGNOSTIC FIDELITY METRICS

The main-paper tables carry the metrics that separate methods. These are the ones that diagnose *how* a render fails, and they are reported for 4DPS, its unfit render and, where a column is defined for it, RF-Genesis.

Protocol details behind Sec. 4. The micro-Doppler trajectory statistics (centroid, spread, envelopes) are taken on frames whose pose speed is at least 0.15 m/s, because a frame with no motion has no trajectory to score and its centroid is set by the residual clutter rather than by the subject.

The raw chain: ADC, complex range profile and range–azimuth–elevation. The same fitted scene yields the raw ADC samples, the complex range profile and the range–azimuth–elevation magnitude with no retraining, because each is a fixed transform of the per-path quantity Eq. (3) already deposits. Tables 7 and 8 score the ADC and the CRP, and Table 9 scores the RAE next to range–angle, on the same sequences, frames and row sections as the other representations, following the CRP and ADC protocol of Armouti et al. (2026a): the first loop's azimuth virtual array, near-field bins zeroed. The magnitude columns of the CRP and the ADC compare each side min-max normalized on its own, so they do not depend on absolute scale, and the ADC envelope is compared after the rank-1 per-channel and per-range phase correction, because a magnitude-only objective leaves that phase free. The relative-phase columns need no correction: each scores the phase step between neighboring samples (fast time t, range r, virtual channel v) against the recording's, as the $|y|$-weighted mean of $\cos \Delta\phi$, and a per-bin rotation cancels in that step.

Table 7: **Rendering fidelity on the ADC samples.** Table 2 of the main text carries the seven columns its prose reads; this table and Tables 8–10 carry all of them, grouped by representation, over the same sequences and the same renders, with H, V and C as there. ρ_{env} is the envelope correlation after the rank-1 phase correction; $\Delta\phi_t$ is the relative-phase coherence along fast time (1 is the recording's phase step, 0 random), with its weighted RMSE in degrees. The unfit render predates this family and is not scored. Every column here is scale-free, so RF-Genesis is scored as rendered. The ADC phase depends on the scene: on RT-Pose its median $\Delta\phi_t$ is 0.51 on the clean recordings and 0.23 on the cluttered ones. Bold: the better of the two methods in each column of a row section.

Method	$\rho_{\text{env}}\uparrow$	PSNR↑	SSIM↑	RMSE↓	$\Delta\phi_t\uparrow$	$\Delta\phi_t$ (°)↓
HuPR, horizontal single-chip radar (H)						
RF-Genesis (Chen & Zhang, 2024)	0.649	14.0	0.37	0.205	-0.21	109
4DPS (Ours)	**0.857**	**21.2**	**0.58**	**0.088**	**0.55**	**63**
HuPR, vertical single-chip radar (V)						
RF-Genesis	0.629	14.0	0.34	0.203	0.11	90
4DPS (Ours)	**0.772**	**19.0**	**0.48**	**0.114**	**0.44**	**72**
RT-Pose, cascade (C), one-person sequences						
RF-Genesis	0.630	15.1	0.28	0.179	-0.05	103
4DPS (Ours)	**0.667**	**20.0**	**0.32**	**0.100**	**0.24**	**88**

Table 8: **Rendering fidelity on the complex range profile (CRP).** Sequences, renders and row sections as in Table 7. $\Delta\phi_r$ and $\Delta\phi_v$ are the relative-phase coherences along range and virtual channel (1 is the recording's phase step, 0 random), each with its weighted RMSE in degrees. The unfit render predates this family and is not scored. Every column here is scale-free, so RF-Genesis is scored as rendered. Its negative $\Delta\phi_r$ is a sampling convention: its range profiles step by π per bin against the recording's, and multiplying them by $(-1)^k$ makes the coherence positive and changes no other column; we report it as run. Bold: the better of the two methods in each column of a row section.

Method	$\rho\uparrow$	PSNR↑	SSIM↑	RMSE↓	$\Delta\phi_r\uparrow$	$\Delta\phi_r$ (°)↓	$\Delta\phi_v\uparrow$	$\Delta\phi_v$ (°)↓
HuPR, horizontal single-chip radar (H)								
RF-Genesis	0.246	20.0	0.28	0.102	-0.63	147	0.52	67
4DPS (Ours)	**0.856**	**26.0**	**0.65**	**0.052**	**0.66**	**56**	**0.75**	**48**
HuPR, vertical single-chip radar (V)								
RF-Genesis	0.274	20.0	0.19	0.102	-0.52	139	0.31	82
4DPS (Ours)	**0.785**	**24.8**	**0.56**	**0.061**	**0.55**	**65**	**0.47**	**71**
RT-Pose, cascade (C), one-person sequences								
RF-Genesis	0.211	22.8	0.43	0.073	-0.54	137	0.16	94
4DPS (Ours)	**0.735**	**28.2**	**0.76**	**0.039**	**0.50**	**68**	**0.33**	**82**

Table 9: **Rendering fidelity on range–angle and range–azimuth–elevation (RAE).** Sequences, renders and row sections as in Table 7. PSNR is $20\log_{10}(\text{peak}/\text{RMSE})$, the only logarithm and applied to a scalar ratio. The unfit render predates the RAE family and is not scored (–). Restricted to the 57 sequences of the RT-Pose split the fitted row reads 0.835 / 0.451 on range–angle ρ and ρ_c, so the fitted and unfit sequence sets are comparable. RF-Genesis has no absolute scale, so for SSIM, NRMSE and PSNR its render is multiplied before scoring by one constant per radar that matches its loop-0 range–azimuth energy to the recording's, the rule that initializes 4DPS's own gain (App. C.3); ρ and ρ_c are unchanged by it. Bold: the best value in each column of a row section, by the column's arrow.

	Range–angle					**RAE**			
Method	$\rho\uparrow$	$\rho_c\uparrow$	SSIM↑	NRMSE↓	PSNR↑	$\rho\uparrow$	SSIM↑	NRMSE↓	PSNR↑
HuPR, horizontal single-chip radar (H)									
RF-Genesis (Chen & Zhang, 2024)	0.139	0.374	0.51	1.08	22.5	0.165	0.48	1.11	22.7
4DPS (unfit)	0.901	0.353	0.68	0.44	30.3	–	–	–	–
4DPS (Ours)	**0.960**	**0.699**	**0.79**	**0.28**	**34.0**	**0.942**	**0.80**	**0.33**	**33.3**
HuPR, vertical single-chip radar (V)									
RF-Genesis	0.129	0.317	0.35	1.10	21.0	0.153	0.36	1.13	21.8
4DPS (unfit)	0.863	0.234	0.53	0.52	27.6	–	–	–	–
4DPS (Ours)	**0.925**	**0.622**	**0.67**	**0.38**	**30.2**	**0.904**	**0.72**	**0.40**	**30.7**
RT-Pose, cascade (C), one-person sequences									
RF-Genesis	0.119	0.221	0.46	1.13	25.4	0.117	0.51	1.11	26.5
4DPS (unfit)	0.648	0.181	0.72	0.90	27.5	–	–	–	–
4DPS (Ours)	**0.830**	**0.470**	**0.78**	**0.53**	**31.8**	**0.779**	**0.81**	**0.58**	**31.9**

Table 10: **Rendering fidelity on range–Doppler (RD) and micro-Doppler.** Sequences, renders and row sections as in Table 7. P_M is the rendered-over-real power ratio on the CFAR-occupied cells, L_U and σ_U the level and spread ratios on the unoccupied ones, ρ_{env} the correlation of the Doppler envelopes and BW their separation, and clDice scores connectivity after independent segmentation of each signature. Restricted to the 57 sequences of the RT-Pose split the fitted row reads 0.695 / 0.968 on RD ρ_M and micro-Doppler ρ_S, so the fitted and unfit sequence sets are comparable. RF-Genesis has no absolute scale, so for P_M, L_U and σ_U its render is multiplied before scoring by one constant per radar that matches its loop-0 range–azimuth energy to the recording's, the rule that initializes 4DPS's own gain (App. C.3); every other column is unchanged by it. Bold: the best value in each column of a row section, by the column's arrow.

	RD occupied		**RD floor**		**Micro-Doppler**					
Method	$\rho_M \uparrow$	$P_M \to 1$	$L_U \to 1$	$\sigma_U \to 1$	$\rho_c \uparrow$	$\rho_\sigma \uparrow$	$\rho_S \uparrow$	$\rho_{\mathrm{env}} \uparrow$	BW$\to 1$	clDice$\uparrow$
HuPR, horizontal single-chip radar (H)										
RF-Genesis (Chen & Zhang, 2024)	0.564	1.51	0.06	1.30	0.82	0.08	0.670	0.28	4.22	0.34
4DPS (unfit)	0.584	0.60	0.57	0.66	0.94	0.31	0.771	0.79	**1.07**	0.89
4DPS (Ours)	**0.891**	**0.71**	**0.92**	**0.88**	**1.00**	**0.98**	**0.993**	**0.87**	0.87	**0.91**
HuPR, vertical single-chip radar (V)										
RF-Genesis	0.480	**1.12**	0.05	1.24	0.82	0.03	0.672	0.28	3.79	0.37
4DPS (unfit)	0.513	0.37	0.56	0.60	0.93	0.25	0.740	0.77	**0.91**	**0.83**
4DPS (Ours)	**0.795**	0.54	**0.92**	**0.88**	**0.99**	**0.94**	**0.984**	**0.80**	0.79	**0.83**
RT-Pose, cascade (C), one-person sequences										
RF-Genesis	0.324	0.61	0.09	1.51	0.75	0.04	0.394	0.07	1.53	0.52
4DPS (unfit)	0.291	0.28	0.58	0.75	0.80	0.19	0.515	0.46	0.60	0.59
4DPS (Ours)	**0.695**	**0.63**	**0.96**	**0.94**	**0.97**	**0.70**	**0.966**	**0.65**	**0.71**	**0.71**

E DOWNSTREAM RESULTS IN FULL

Table 11 gives the full real/rendered train–test matrix, of which Table 3 keeps the rows tested on real data. Tables 12, 13 and 14 carry the generated downstream tables behind Table 3. Table 12 gives the retention and gap summary (retention is S→R over R→R; the information gap is R→R minus S→S, the realism gap R→R minus S→R, the reverse gap R→R minus R→S, in the dataset's own unit) beside the real-trained model on a 50/50 real/rendered test set, Table 13 the per-keypoint and per-joint breakdowns with the PCK scores, and Table 14 the RT-Pose scores by activity, location, occlusion and scenario. Read the HuPR per-keypoint rows against the arms' AP: the arms lose most of what the rendered arm loses (wrists 22.2 and 25.1 real-to-real against 8.0 and 4.7 rendered-to-real), while hips, knees and the head retain 0.73 to 0.80 of their real-to-real value against the arms' 0.19 to 0.39.

The RT-Pose baseline and its labels. HRRadarPose (Ho et al., 2025) publishes no checkpoint; our reproduction on real data reaches 16.7 cm against the 9.9 cm the paper reports, and every RT-Pose number in this paper is measured under that reproduction rather than against the published figure. The released 3D labels stop before the end of each sequence, carry a per-session scale and offset against the radar, and label one sequence with another's radar, which is why we derive the 15 joints on every radar frame from the registered body mesh that drives the render and exclude sequence 13. HuPR trains at the released sampling ratio for 200 epochs, and RT-Pose's remaining 63 one-person sequences are split 44/6/13. Its 37 two-person sequences are fitted against the one person who carries pose annotations and are not scored downstream.

Protocol details behind Sec. 4. RT-Pose's 44/6/13 split is stratified by location, scenario, activity and occlusion, so that no split is carried by one room or one action. On rendered test data the released decoder's confidence threshold is lowered from its released value to -1, so that every joint is kept and a rendered frame is never scored on fewer joints than its real twin; the threshold never triggers on real data, so no real number in the paper depends on it.

Table 11: **Real and rendered as train and test domains, the full matrix.** Table 3 keeps the rows tested on real data. All six metrics are pose-estimation metrics, and every cell is retrained from scratch for four seeds with the standard deviation in parentheses. The first block is the real-only reference, the second is RF-Genesis and the third the same four arms under 4DPS. Rows tested on a renderer measure how much pose information that render carries, not accuracy on real data. RF-Genesis's RT-Pose inputs carry one level gain fixed before training, because its units are arbitrary (App. C.3).

		RT-Pose (cm ↓)			**HuPR** (AP ↑)		
Train	**Test**	MPJPE	MRPE	Abs-MPJPE	AP	AP_{50}	AP_{75}
Real	Real	16.67 (0.42)	13.14 (0.87)	19.07 (0.52)	65.6 (1.7)	96.9 (0.2)	78.0 (3.1)
Real	RF-Genesis	28.46 (3.55)	28.56 (3.54)	38.15 (6.14)	45.9 (3.9)	97.4 (0.7)	28.5 (9.2)
RF-Genesis	Real	127.59 (48.32)	106.48 (45.00)	148.88 (24.86)	1.8 (3.5)	8.5 (16.1)	0.0 (0.1)
RF-Genesis	RF-Genesis	12.44 (0.34)	8.97 (0.74)	12.72 (0.37)	77.0 (0.4)	98.8 (0.1)	93.8 (1.0)
Real + RF-Genesis	Real	16.82 (0.56)	13.71 (0.88)	19.41 (0.58)	64.1 (0.9)	96.9 (0.0)	76.5 (1.6)
Real	4DPS	20.05 (1.50)	14.32 (1.07)	22.70 (1.76)	49.2 (0.7)	93.9 (0.8)	45.7 (1.3)
4DPS	Real	21.86 (1.33)	17.06 (0.98)	25.45 (1.69)	38.4 (2.9)	80.6 (3.2)	30.1 (4.3)
4DPS	4DPS	15.71 (1.38)	10.08 (1.00)	15.81 (1.30)	59.6 (1.4)	98.5 (0.1)	67.9 (4.1)
Real + 4DPS	Real	16.52 (0.80)	14.39 (0.93)	19.43 (0.35)	61.6 (0.4)	96.4 (0.6)	72.3 (0.2)

Table 12: **Downstream transfer summary**, generated by the results script (protocol hashes hupr 38ea71937b32, rtpose 989ccef2230e). Retention and the three gaps are defined in the text and read in the dataset's own unit (AP $\times 100$ on HuPR, MPJPE in cm on RT-Pose); the last two columns score the real-trained model on a 50/50 real/rendered test set and on the real test set, mean (std) over four seeds.

					R→R model	
Dataset	Retention	Information gap	Realism gap	Reverse gap	50/50 test	Real test
HuPR (AP, pts)	0.59	6.0	21.2	16.4	55.8 (1.1)	65.6 (1.7)
RT-Pose (MPJPE, cm)	0.76	-1.0	6.1	3.4	18.93 (1.44)	16.67 (0.42)

Table 13: **Per-keypoint and per-joint downstream accuracy**, per arm (train→test; R real, S rendered by 4DPS), same protocol as Table 12: HuPR AP $\times 100$ per keypoint and PCKh@0.2 (top), RT-Pose MPJPE in cm per joint and PCK at 10 and 15 cm (bottom). MPJPE is taken relative to the pelvis, so the pelvis reads 0; PCK on RT-Pose is scored on real test data only (–). Mean (std) over four seeds.

HuPR keypoint (AP)	R→R	S→R	R→S	S→S	R+S→R
R_Hip	90.1 (0.6)	72.1 (2.9)	82.7 (1.7)	91.5 (0.8)	89.4 (0.2)
R_Knee	84.5 (0.7)	65.3 (2.8)	76.0 (1.3)	87.7 (0.6)	83.8 (0.5)
R_Ankle	73.3 (1.7)	50.2 (2.6)	73.9 (2.2)	73.6 (1.3)	71.8 (1.6)
L_Hip	90.6 (1.3)	71.7 (3.3)	83.0 (1.4)	90.8 (1.3)	88.9 (0.6)
L_Knee	83.7 (2.2)	61.1 (3.3)	75.7 (1.0)	85.2 (2.6)	82.1 (0.9)
L_Ankle	77.5 (1.3)	63.0 (2.0)	66.2 (1.3)	80.8 (1.0)	75.0 (1.0)
Neck	83.1 (1.8)	64.7 (4.1)	73.2 (1.7)	82.0 (1.2)	80.4 (1.7)
Head	81.4 (1.1)	63.3 (4.0)	70.1 (1.0)	79.9 (1.1)	78.9 (1.0)
L_Shoulder	74.7 (1.2)	53.3 (4.5)	58.0 (2.0)	71.6 (1.2)	72.4 (0.4)
L_Elbow	41.1 (1.4)	16.2 (2.6)	19.5 (1.0)	27.6 (1.4)	37.1 (1.3)
L_Wrist	22.2 (1.8)	8.0 (1.1)	7.1 (0.5)	13.3 (0.8)	18.1 (0.7)
R_Shoulder	76.8 (2.3)	52.7 (3.6)	63.7 (3.3)	73.7 (1.7)	74.1 (0.9)
R_Elbow	50.3 (4.1)	16.1 (2.7)	25.2 (0.9)	36.8 (2.1)	43.6 (2.4)
R_Wrist	25.1 (3.0)	4.7 (0.8)	9.8 (0.7)	16.1 (2.3)	19.3 (1.1)
PCKh@0.2	43.6 (1.7)	25.0 (1.8)	27.3 (2.2)	39.2 (2.2)	41.7 (1.6)

RT-Pose joint (MPJPE, cm)	R→R	S→R	R→S	S→S	R+S→R
Pelvis	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
Right_Hip	9.5 (1.0)	11.4 (0.9)	11.4 (1.5)	9.0 (0.4)	9.2 (0.8)
Right_Knee	14.2 (1.4)	18.6 (0.8)	17.2 (2.4)	20.4 (14.9)	15.5 (1.1)
Right_Ankle	17.3 (0.9)	23.0 (1.1)	19.9 (1.3)	15.2 (0.5)	18.9 (1.0)
Left_Hip	9.5 (0.9)	12.1 (0.9)	11.1 (1.7)	9.4 (0.7)	9.0 (0.8)
Left_Knee	13.6 (0.5)	19.7 (2.1)	16.3 (1.4)	13.3 (0.7)	15.1 (1.2)
Left_Ankle	17.0 (0.6)	22.0 (1.2)	19.4 (1.2)	15.8 (1.2)	18.1 (1.2)
Thorax	10.6 (0.8)	13.8 (0.7)	12.5 (1.2)	10.0 (0.8)	10.0 (1.0)
Head	14.0 (0.8)	16.4 (1.6)	15.4 (1.4)	13.1 (0.4)	13.5 (0.2)
Left_Shoulder	13.2 (1.1)	16.6 (0.8)	15.2 (1.5)	12.7 (0.8)	13.1 (0.6)
Left_Elbow	23.4 (0.4)	28.5 (1.2)	26.6 (2.0)	21.4 (0.5)	23.1 (0.8)
Left_Wrist	32.6 (2.0)	46.7 (6.5)	37.5 (4.5)	29.7 (2.4)	30.7 (0.8)
Right_Shoulder	13.1 (0.9)	16.3 (0.6)	15.6 (0.9)	12.6 (0.3)	13.0 (0.9)
Right_Elbow	24.0 (0.8)	30.0 (2.1)	27.6 (1.8)	21.6 (0.4)	23.9 (1.5)
Right_Wrist	38.1 (4.2)	52.7 (7.7)	54.9 (18.8)	31.6 (3.5)	34.7 (1.3)
PCK@10cm	45.8 (2.0)	35.9 (1.5)	–	–	44.8 (2.3)
PCK@15cm	64.3 (1.9)	54.0 (1.7)	–	–	62.1 (2.7)

Table 14: **RT-Pose MPJPE (cm) by group**: the test sequences' activity, location, occlusion and scenario, for the three arms scored on real test data, same protocol as Table 12; mean (std) over four seeds.

RT-Pose MPJPE (cm) by group		R→R	S→R	R+S→R
activity	Stand and Lift Leg	9.81 (0.20)	14.45 (1.60)	9.77 (0.27)
activity	Stand and Wave Hand	17.08 (1.08)	22.03 (0.81)	18.66 (1.70)
activity	Walk	21.67 (2.14)	25.69 (2.54)	19.23 (0.43)
activity	Walk and Sit	10.43 (0.51)	25.28 (1.00)	10.77 (1.20)
activity	Walk and Wave Hand	14.58 (0.28)	17.05 (1.31)	14.18 (0.41)
location	Indoor	19.40 (0.88)	23.89 (1.63)	17.92 (1.09)
location	Outdoor	12.30 (0.63)	18.60 (0.89)	14.27 (0.67)
occlusion	Cardboard	25.66 (1.99)	24.13 (1.74)	25.31 (0.77)
occlusion	Nothing	14.62 (1.15)	20.72 (1.24)	13.53 (0.73)
occlusion	Plastic pad	16.87 (1.86)	24.68 (2.81)	21.16 (1.65)
scenario	Clean	16.55 (0.39)	22.08 (1.32)	15.30 (0.69)
scenario	Cluttered	21.11 (1.29)	24.98 (2.07)	19.49 (1.33)
scenario	Normal	12.61 (0.77)	19.67 (1.17)	15.17 (0.59)
scenario	Rain	11.06 (0.69)	14.36 (0.26)	10.69 (1.14)

F REPRESENTATION ANALYSIS PROTOCOL

We fixed this protocol, including every measure and the $1.2\times$ margin, before running any comparison, so the analysis is pre-registered; where a measurement departs from it, we say so. HuPR follows RT-Pose's design and changes only what the dataset forces. Every number comes from a JSON that records the protocol hash (`989ccef2230e` for RT-Pose, `38ea71937b32` for HuPR) and the code commit (`3505210`).

Models and comparisons. We compare 13 checkpoints per dataset: five real-trained seeds and the four rendered- and four mixed-trained seeds (0–3) of Table 3. All arms of a seed start from the same weights, and a seed's real and rendered arms see their batches in the same order. Table 15 lists the comparisons and what differs within each; every ratio is taken against the real–real comparison of the same row. The different-initialization row is complete and carries the claim. The same-initialization row has no real–real reference of its own, because we trained no second real arm per seed, so it is read against the different-initialization real–real mean, which bounds its ratio from above. An untrained network is the only control.

Table 15: **The cells compared**, with what differs inside each. Ratios are taken against the real–real cell of the matched row. No real arm was trained twice from one initialization, so the first row has no real–real reference and its cross pairs are read against the different-initialization real–real mean, which bounds their ratio from above.

Row	Cell	Pairs	Differs in	Reference
same init	real_k vs rendered_k	4	data	(none trained)
diff. init	real_i vs real_j	10	init, order	itself
	real_i vs rendered_j	16	init, order, data	real–real, diff. init
beside the design	mixed vs real	4, 16	data (and init)	real–real, diff. init
	rendered vs rendered	6	init, order	real–real, diff. init
	untrained vs real	5	training itself	real–real, diff. init

Frames and layers. On RT-Pose we use every annotated frame of the 13 real test sequences (3,887 frames) and the same frames of the rendered test split. The layers (taps) are the input (T0), HRNet layer 1 (T1), the parallel branches of stages 2 to 4 (T2a–b, T3a–c, T4a–d; T4a is also the backbone output) and the head's pre-activation maps (T6). On HuPR a frame stacks eight frames of eight chirps on two boards, so error consistency and paired inputs use all 12,600 test frames, while similarity uses every tenth (1,260 frames, every sequence represented) and the readout is fitted on every tenth frame of the real training split. Its layers are each board's input (H0), the chirp network's output (H1), the three encoder levels (H2–H4; H4 feeds the decoder) and the decoder's pre-sigmoid heatmap (H5); the two boards are never averaged. At every layer we subsample 1,024 spatial positions (fewer where the grid is smaller), drawn once and shared by every frame and model, and treat each position as a row of channels.

Measures. Error consistency (Geirhos et al., 2020) is the correlation of per-frame error between two models and their per-joint agreement, where each model's per-frame error at a joint is split at that model's own median. On HuPR a keypoint is a hit at a per-keypoint OKS of at least 0.5, and Cohen's κ over (frame, keypoint) pairs is the primary statistic. Similarity uses debiased linear CKA (Nguyen et al., 2021), computed exactly, and the angular orthogonal Procrustes distance (Williams et al., 2021), because Procrustes catches functional changes CKA misses (Ding et al., 2021). The fully whitened Procrustes stays inside the margin at every layer (1.00 to 1.02) but barely separates an untrained network, so we do not use it. Intervals come from 1,000 bootstrap replicates that resample frames and model pairs.

Checks. Seed 0's real-trained checkpoint re-scores to 16.652 cm against 16.651 on the machine that produced Table 3 on RT-Pose, and to 0.66871 against 0.66871 AP on HuPR, under the same protocol hashes. A model against itself gives CKA 1 and Procrustes 0 at every layer, the untrained network lies above every real–real band after the input, and the rendered test split repeats the similarity picture layer for layer.

Table 16: **Representational similarity by tap** (RT-Pose, real probe, positions convention). CKA is a similarity and the Procrustes distance is in radians; *ratio* is the real–rendered different-init distance over the real–real different-init distance, with its 95% interval, against the pre-registered margin of 1.2. *Floor* is an untrained network. T0 is the input tensor, identical for every model. Values from `similarity_rtpose.json`, generated into `tables/repr_macros.tex`.

	debiased CKA (similarity ↑)				**Procrustes** (distance, rad ↓)			
Tap	real–real	real–rend.	floor	ratio	real–real	real–rend.	floor	ratio
T0 input	1.000	1.000	1.000	–	0.000	0.000	0.000	–
T1 layer 1	0.713	0.613	0.417	1.35 [1.21, 1.49]	0.816	0.857	1.013	1.05 [1.02, 1.07]
T2a	0.537	0.459	0.188	1.17 [1.01, 1.35]	0.993	1.038	1.213	1.04 [1.01, 1.08]
T2b	0.350	0.366	0.169	0.98 [0.89, 1.07]	1.095	1.087	1.261	0.99 [0.96, 1.02]
T3a	0.228	0.200	0.071	1.04 [0.94, 1.15]	1.223	1.238	1.345	1.01 [0.97, 1.05]
T3b	0.300	0.270	0.085	1.04 [0.92, 1.19]	1.182	1.202	1.338	1.02 [0.98, 1.06]
T3c	0.251	0.194	0.133	1.08 [1.05, 1.11]	1.236	1.275	1.340	1.03 [1.02, 1.04]
T4a (= T5)	0.069	0.085	0.039	0.98 [0.92, 1.05]	1.357	1.359	1.397	1.00 [0.95, 1.07]
T4b	0.267	0.247	0.078	1.03 [0.96, 1.10]	1.200	1.222	1.353	1.02 [0.99, 1.05]
T4c	0.368	0.343	0.140	1.04 [0.97, 1.11]	1.240	1.266	1.343	1.02 [1.01, 1.04]
T4d	0.332	0.266	0.111	1.10 [1.03, 1.16]	1.254	1.302	1.355	1.04 [1.02, 1.05]
T6 head	0.048	0.055	0.014	0.99 [0.92, 1.07]	1.410	1.385	1.478	0.98 [0.90, 1.09]

Error consistency. The two models do not fail on the same frames. Relative to two real-trained models of different seeds, they disagree about which joints they get right at 1.23 on RT-Pose and 2.94 on HuPR, and their per-frame errors correlate less, at 1.13 and 2.20 (Table 17). Two real-trained models also disagree on frames, which is why each value is a ratio to that disagreement. Behind the ratios, two real-trained seeds correlate at 0.320 in per-frame MPJPE and agree on 0.690 of the per-joint indicators, against 0.234 and 0.618 across domains; on HuPR they agree on 0.936 of the keypoint hits against 0.812, and κ falls from 0.589 to 0.335. On the rendered test split the RT-Pose ratios fall to 1.01 and 1.11, and the mixed arm is indistinguishable from a real seed (ratios 0.88 to 0.99). We report this here rather than in Sec. 4.3 because it concerns which inputs each model gets wrong, not what the models learned.

Table 17: **Do rendered- and real-trained models fail on the same frames?** Error consistency as the real–rendered value over the real–real value between seeds, with its 95% interval (1.0: as consistent as two real-trained models). Bold: the interval lies above the $1.2\times$ margin. The cells behind each ratio follow.

	Frame error corr.	Joint error agr.
RT-Pose	1.13 [0.99, 1.31]	**1.23** [1.21, 1.26]
HuPR	**2.20** [2.07, 2.34]	**2.94** [2.78, 3.11]

Table 18: **Representational similarity by tap** (HuPR, real probe, positions convention), as Table 16 for RT-Pose. H1 is the chirp network's output, H4 the tensor the decoder reads and H5 its pre-sigmoid heatmap; h and v are the horizontal and vertical boards. The level-1 and level-2 encoder taps are undefined, because they carry no signal in some checkpoints (App. F, dead temporal merges). The rendered probe reproduces every computable cell to within 0.03.

	debiased CKA (similarity ↑)				**Procrustes**
Tap	real–real	real–rend.	floor	ratio	ratio
H1h chirp output	0.976	0.973	0.858	1.15 [0.97, 1.35]	1.08
H2h, H3h, H2v, H3v	*undefined: the tap is dead in some checkpoints*				–
H4h encoder level 3	0.287	0.229	0.227	1.08 [0.93, 1.27]	1.05
H4v encoder level 3	0.164	0.084	0.044	1.10 [0.96, 1.30]	1.15
H5 decoder logits	0.674	0.554	0.040	1.37 [1.08, 1.70]	1.17

Mixed arm and shared initialization. The mixed arm sits between 0.92 and 1.24 and passes 1.2 only in the point estimate at T2a (1.24, interval [1.12, 1.38]). The shared initialization explains little of the agreement: the same-initialization pairs are no closer than the different-initialization ones from stage 2 onward (0.88 to 1.13).

Linear readout. We freeze each backbone and fit a closed-form ridge readout on real training data to the joint coordinates (RT-Pose: T5 pooled to $4 \times 8 \times 16$; HuPR: H4h and H4v pooled to 4×4), with one penalty chosen on held-out real frames and applied to every backbone, and score it with the released evaluator. The registered form, a $1 \times 1 \times 1$ convolution trained under the released losses, converged on no backbone, the real ones included, so we use the plan's alternative. On RT-Pose the rendered-trained models lose 31% through their own decoders and 16% through the readout (15.62 against 13.46 cm, ratio 1.16, interval [1.09, 1.28]), but an untrained backbone probes at only 1.25, so the test barely discriminates there. On HuPR the real, rendered, mixed and untrained backbones reach 8.98, 11.66, 9.64 and 22.43 px of keypoint error, and the untrained ratio of 2.50 is what gives this test its power on HuPR.

Paired inputs. Every rendered frame has a real twin at the same timestamp, so we measure the feature distance between the twins inside one model, in units of the distance between neighboring real frames. To a real-trained model the render's departure from the recording is worth 2.87 to 3.29 frames of real motion, and it does not shrink with depth. A rendered-trained model sees a smaller gap (2.54 to 3.05), so training on renders makes a model less sensitive to the domain difference without closing it. On HuPR we use all 12,600 test frames rather than the registered tenth, so that a neighbor is one frame away as on RT-Pose; there the twin distance at the chirp network's output is 0.97 for a real-trained and 0.97 for a rendered-trained model, so the two part only after the encoder.

Dead layers on HuPR. MSCSA-PRGCN merges its frames with bias-free convolutions, and training drives some of them to denormal magnitude, so the layer after such a merge carries no signal. Of the 13 checkpoints, H2h and H3h are dead in 1 and 3, and H2v and H3v in 5 and 3. A pair containing a dead layer is undefined, so we report those layers as undefined rather than substitute another.

G SCENE EDITING

4DPS fits the static scene and the person as separate explicit components. We can therefore place a fitted person in another fitted scene, or give a fitted scene a motion that was never recorded, and render again without refitting. We test the first edit against real recordings (App. G.1). The second edit has no real recording to compare against, so we show it without a score (App. G.2).

G.1 SEPARABILITY OF THE SCENE AND THE PERSON

The test. We ask whether a fitted person keeps its fidelity when the scene around it changes. For two recordings A and B of the same environment, we render B's complete fit inside A's fitted static scene. B's fit is its registered mesh, person material, per-frame velocity field and normals, and joint gains; A's scene is its points, their orientations and their materials. We then score the render against B's real recording, on B's frames, with the evaluation of Table 2. B's receiver gain, noise floor and antenna coupling stay with B, because they describe B's capture and not the room. The only other change is the person's shadow on the scene, which we cast again for A's scene points with B's mesh, and nothing is refitted. Rendering B's fit in B's own scene through the same code reproduces B's own render.

The pairs. Two recordings count as the same environment when the time-averaged zero-Doppler range–azimuth maps of their real data correlate at 0.75 or more. We correlate the maps in dB over the retained range bins, and on HuPR we take the lower of the two boards. We set this threshold on RT-Pose, where the largest correlation between two recordings of different labeled scenes is 0.74. Each pair comes from one recording session (one subject in one set of clothes), with one sequence standing and the other moving. We take 24 such pairs on RT-Pose and 20 on HuPR, evenly spaced over the candidates, whose static returns correlate at 0.77 and 0.79 or more. Rendering every pair in both directions gives 48 and 40 renders.

Result. Table 19 shows that the render in the other scene keeps all of the person's fidelity and most of the scene's. The range–Doppler and micro-Doppler columns and L_U equal B's own fit: the median paired difference in micro-Doppler ρ_S is 0.000 on RT-Pose and 0.000 on HuPR. These columns are not evidence that the scene transfers, however, because they would match even if it did not. The range–Doppler and micro-Doppler products remove the zero-Doppler return, and our renderer is single-bounce, so the static scene reaches them only through the person's shadow; the noise floor behind L_U stays with B. The evidence is in the range–azimuth columns, which the scene dominates. There, A's scene explains B's static return at 90 % of B's own fit on ρ on RT-Pose and 92 % on HuPR (median paired differences -0.055 and -0.079). On the clutter-removed ρ_c, the share is 90 % on RT-Pose and 62 % on HuPR, and HuPR's is the largest loss in the table. The render in the other scene still scores above B's unfit render and RF-Genesis on most renders in every column. The one exception is HuPR's range–azimuth ρ against the unfit render, where it is higher on 48 % of renders. Fig. 5 shows three pairs per dataset.

G.2 NOVEL MOTIONS

Source and placement. Data augmentation for new activities and for falls needs motions that neither dataset contains, so we render such motions in fitted scenes. We take jumping jacks, squats, kicks and one real fall, a rug pulled from under the subject, from the CMU motion-capture set in AMASS (Mahmood et al., 2019). CMU has no clean forward or backward fall, so we generate those two with MDM (Tevet et al., 2023), a text-to-motion model trained on HumanML3D (Guo et al., 2022). MDM's samples often end in a different posture from their prompt. We therefore drew three samples per prompt and kept the one whose joints do what the prompt says. We judge that by the pelvis's travel relative to the radar and by where the toes point at the end. Each clip is labeled by that measured outcome. We pose each clip with the body shape fitted to the scene's own subject and resample it to the radar's 10 Hz frame rate. We then place it on the scene's floor at the subject's real position, facing the radar. No scene point more than 15 cm above the floor may come within 5 cm of the body in any frame. Where the subject's own position fails that check, we move the start to the nearest point within 1 m that passes, which moved 7 of the 6×5 renders by at most 0.9 m. The render then follows App. G.1: the scene, its materials, noise floor and gain, and the subject's person material come from the fit. The velocities are the clip's joint kinematics scaled by the subject's fitted

Table 19: **Separability of the scene and the person.** B's complete fit rendered in A's fitted static scene ("Separated"), scored against B's real recording with the evaluation of Table 2, beside B's own fit, B's unfit render and RF-Genesis driven by B's registered motion, on the same renders and frames. Medians over renders; HuPR pools its two boards, as Table 2 does, so n counts (render, board) products. Retention is the median of the separated render over the median of B's own fit, on the same renders. RF-Genesis's L_U comes from its level-matched scoring, as in Table 2. A column undefined on a render (no frame of the evaluation's motion subset) has a smaller n than the row's, which the column shows; the generated macros record every column's n.

	range–azimuth	range–Doppler		micro-Doppler			
	ρ	ρ_c	ρ_M	L_U	ρ_c	ρ_σ	ρ_S	n
RT-Pose, 24 pairs, both directions								
Separated (A's scene, B's fit)	0.756	0.484	0.696	0.97	0.962	0.679	0.962	48
B's own fit	0.843	0.540	0.697	0.97	0.962	0.679	0.962	48
Retention (%)	90	90	100	100	100	100	100	
B's unfit render	0.660	0.120	0.282	0.58	0.581	0.168	0.497	44
RF-Genesis	0.131	0.221	0.335	0.09	0.711	0.068	0.401	48
HuPR, 20 pairs, both directions								
Separated (A's scene, B's fit)	0.863	0.382	0.802	0.92	0.970	0.976	0.985	80
B's own fit	0.942	0.617	0.802	0.92	0.970	0.976	0.985	80
Retention (%)	92	62	100	100	100	100	100	
B's unfit render	0.884	0.136	0.414	0.57	0.823	0.265	0.733	80
RF-Genesis	0.131	0.161	0.410	0.05	0.689	0.019	0.648	80

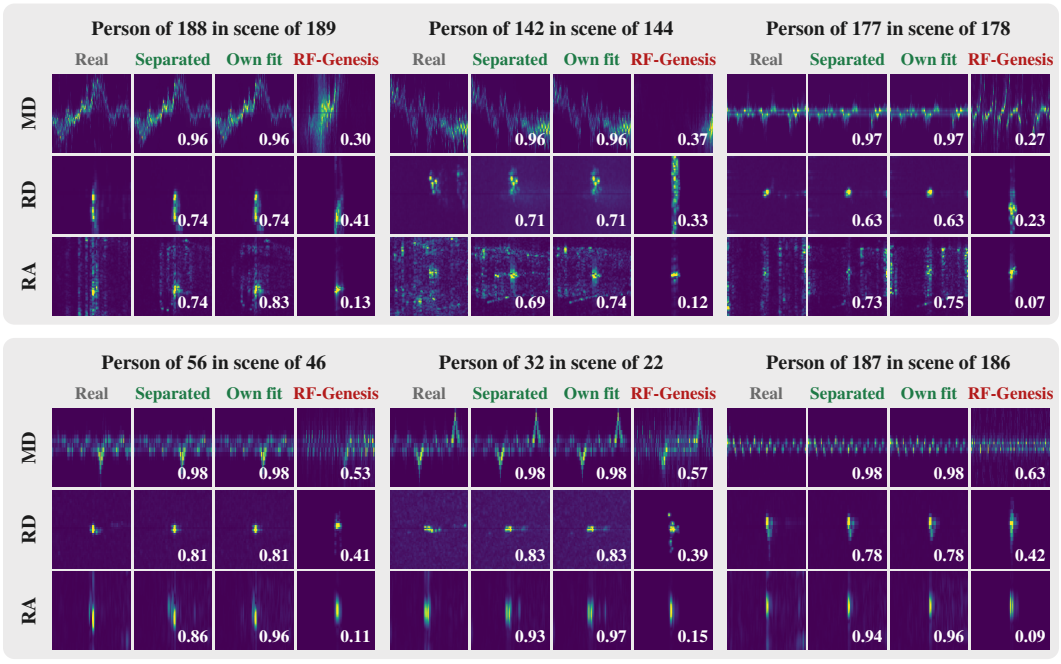

Figure 5: **Separability, RT-Pose (top) and HuPR's horizontal board (bottom).** Each block renders the person of one recording in the fitted scene of another recording of the same environment. Columns: the recording (Real), the person's complete fit rendered in the other scene (Separated), the person's own fit, and RF-Genesis driven by the same registered motion. Rows, frame, normalization and scores are those of Fig. 3. We chose the pairs by a rule: the renders at the 25th, 50th and 75th percentile of the mean ratio of the separated render's three panel scores to the own fit's.

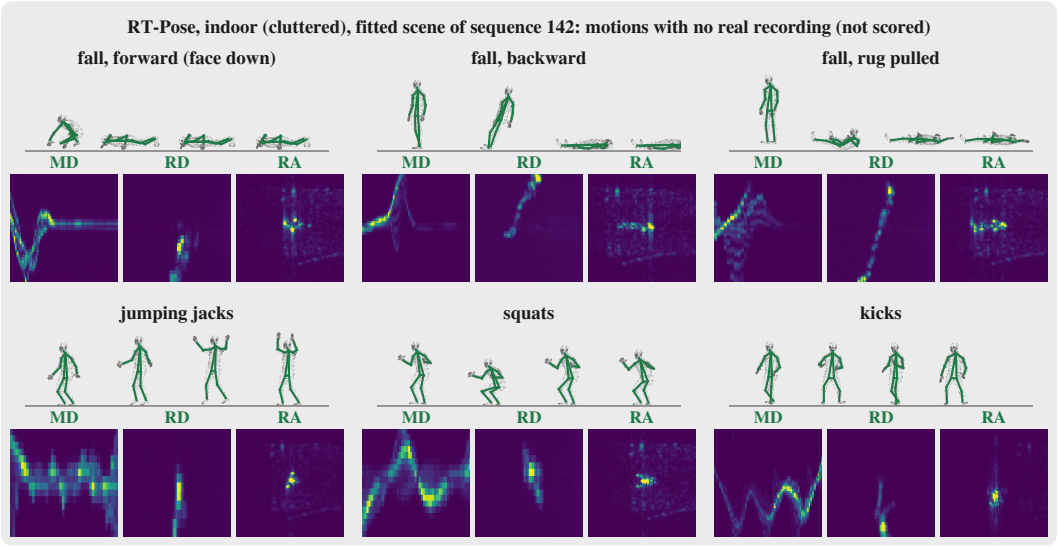

Figure 6: **Novel motions in the fitted scene of RT-Pose sequence 142 (indoor, cluttered) (not scored).** Per motion: the posed mesh at four evenly spaced frames, seen from 45° between the radar's view and the side, above the rendered micro-Doppler over the whole clip (time across, radial velocity up) and the rendered range–Doppler and range–azimuth at the clip's frame of largest pelvis speed (range across, over the person's range ± 2 m). Panels are linear magnitude normalized to their own range, as in Fig. 3. Falls: forward and backward generated by MDM (Tevet et al., 2023), the rug-pulled fall from CMU (Mahmood et al., 2019); exercises from CMU.

per-joint factors. A motion without a recording has no per-frame velocity correction to fit, so these renders carry the motion fidelity of a render without the velocity fit.

What the figures show. Figs. 6–10 render 6 motions in each of 5 fitted scenes. These motions have no real recording, so we cannot score them. We show them as a capability for data augmentation, not as a result.

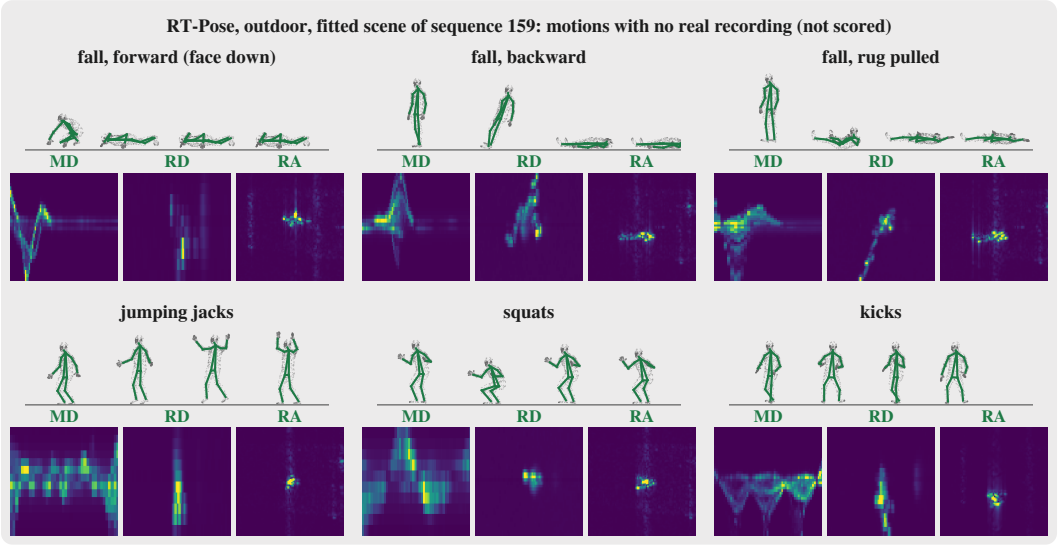

Figure 7: **Novel motions in the fitted scene of RT-Pose sequence 159 (outdoor) (not scored).** Per motion: the posed mesh at four evenly spaced frames, seen from 45° between the radar's view and the side, above the rendered micro-Doppler over the whole clip (time across, radial velocity up) and the rendered range–Doppler and range–azimuth at the clip's frame of largest pelvis speed (range across, over the person's range ±2 m). Panels are linear magnitude normalized to their own range, as in Fig. 3. Falls: forward and backward generated by MDM (Tevet et al., 2023), the rug-pulled fall from CMU (Mahmood et al., 2019); exercises from CMU.

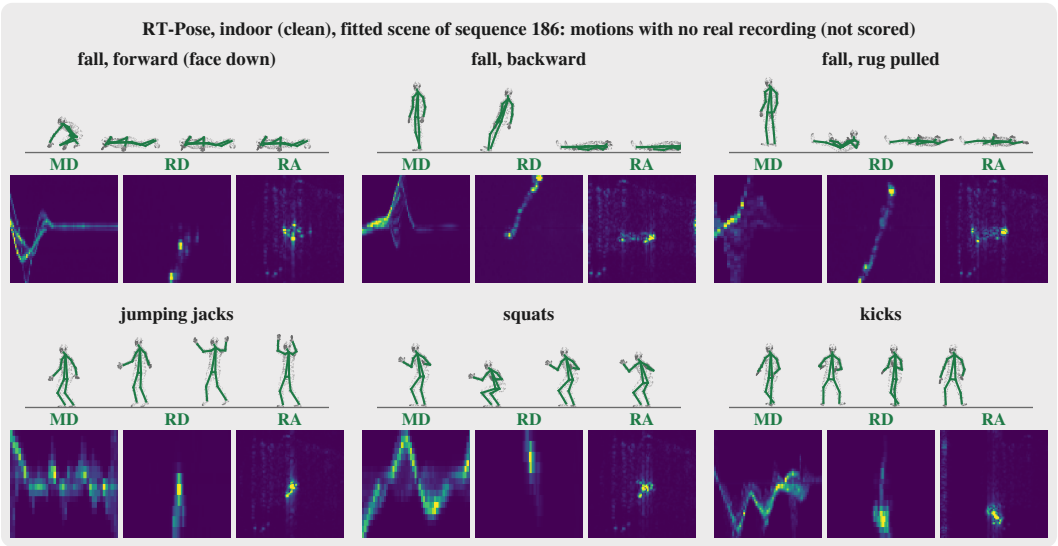

Figure 8: **Novel motions in the fitted scene of RT-Pose sequence 186 (indoor, clean) (not scored).** Per motion: the posed mesh at four evenly spaced frames, seen from 45° between the radar's view and the side, above the rendered micro-Doppler over the whole clip (time across, radial velocity up) and the rendered range–Doppler and range–azimuth at the clip's frame of largest pelvis speed (range across, over the person's range ±2 m). Panels are linear magnitude normalized to their own range, as in Fig. 3. Falls: forward and backward generated by MDM (Tevet et al., 2023), the rug-pulled fall from CMU (Mahmood et al., 2019); exercises from CMU.

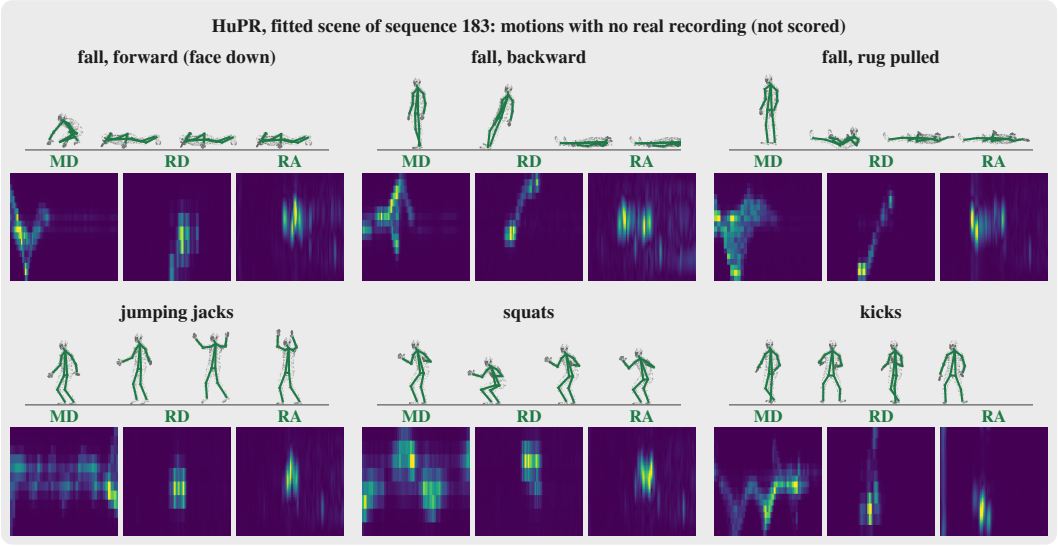

Figure 9: **Novel motions in the fitted scene of HuPR sequence 183 (horizontal board) (not scored).** Per motion: the posed mesh at four evenly spaced frames, seen from 45° between the radar's view and the side, above the rendered micro-Doppler over the whole clip (time across, radial velocity up) and the rendered range–Doppler and range–azimuth at the clip's frame of largest pelvis speed (range across, over the person's range ±2 m). Panels are linear magnitude normalized to their own range, as in Fig. 3. Falls: forward and backward generated by MDM (Tevet et al., 2023), the rug-pulled fall from CMU (Mahmood et al., 2019); exercises from CMU.

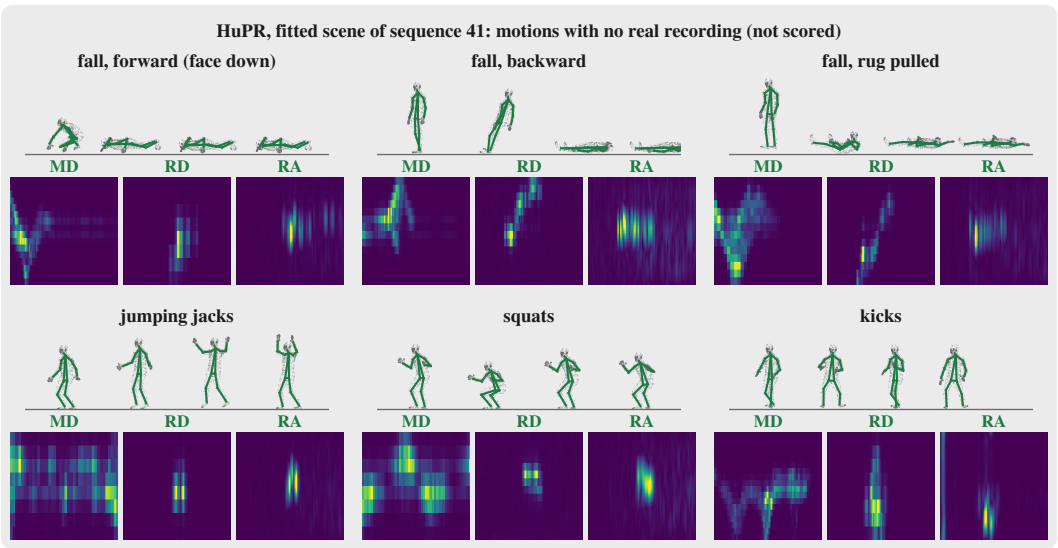

Figure 10: **Novel motions in the fitted scene of HuPR sequence 41 (horizontal board) (not scored).** Per motion: the posed mesh at four evenly spaced frames, seen from 45° between the radar's view and the side, above the rendered micro-Doppler over the whole clip (time across, radial velocity up) and the rendered range–Doppler and range–azimuth at the clip's frame of largest pelvis speed (range across, over the person's range ±2 m). Panels are linear magnitude normalized to their own range, as in Fig. 3. Falls: forward and backward generated by MDM (Tevet et al., 2023), the rug-pulled fall from CMU (Mahmood et al., 2019); exercises from CMU.

H Ablations, Runtime, and Peak Memory

H.1 Ablations

Is every term of the objective necessary? Four terms at unit weight invite the question, and the failures Sec. 3.3 reports are evidence that supervising the person's cells at their own scale does something, not that four is the right number. We therefore remove each global term alone and the two windowed terms together. The windowed pair shares one window and one scope, the person's box, so we remove it as one unit. We expect the global loop-0 term to matter most. The plane terms exclude DC and the windowed pair reaches only the person's box, so the global loop-0 term is the only path by which the static scene receives gradient at all. Each variant multiplies the removed terms by zero inside the loss and re-fits the renderer once per sequence, with everything else at the shipped recipe (the same epochs, person inputs, scaffolds, code path and seed). The "None" row is not a re-fit. It is the render Table 2 reports, and it reproduces Table 2 and Tables 7–10 cell for cell. We judge each variant twice. Reconstruction is scored over Table 2's sequences (HuPR 235 per board, RT-Pose 63 one-person). The gap to real, rendered-to-real minus real-to-real paired per sequence, is read through the four real-trained seeds that Table 3 reports, and only on the test splits (HuPR 21, RT-Pose 13 sequences), because the real-trained models are in-sample on every other sequence. We also count, per sequence and radar, how often a variant scores worse or better than the "None" row by more than a threshold. The threshold is 0.01 on bounded scores (correlations, SSIM, clDice and the phase coherences), 2% of the "None" value on the others, and 1 cm or one AP point on the gap. We fit one renderer seed, because sequences rather than seeds are the axis we generalize over. Only the medians and the paired counts are therefore interpretable, never a single sequence.

Every term carries part of the fit, but not the same part. Table 20 summarizes the three variants, and Tables 21–24 report them on every metric of Tables 7–10. The global loop-0 term carries the static scene, the global per-plane term carries the noise floor and the Doppler structure, and the windowed pair carries a smaller share on the person's cells. None of the three can be removed without a measurable cost somewhere in that metric set.

The global loop-0 term carries the static scene. Without it, the median of every magnitude column of every representation falls on every radar. Range–azimuth ρ falls from 0.960 to 0.646 on H and from 0.830 to 0.141 on C, worse on 235 of 235 and 63 of 63 sequences. The CRP ρ falls from 0.856 to 0.667 on H and from 0.735 to 0.227 on C, again worse on 235 of 235 and 63 of 63. The columns scored after static clutter removal do not move. RD ρ_M reads 0.697 against 0.695 on C and micro-Doppler ρ_S 0.968 against 0.966, and the floor ratios match the "None" row. The expectation held. The two gaps differ. RT-Pose's gap grows from 2.00 to 174.26 cm, worse on 13 of 13 test sequences, while HuPR's reads 16.5 against 16.4 AP (9 worse, 5 better). HuPR's released preprocessing removes static clutter over loops before its model reads the heatmaps, so the static scene this term fits never reaches the HuPR model, while HRRadarPose reads it.

The global per-plane term carries the noise floor and the Doppler structure. Without it, the floor level L_U falls on every radar, from 0.92 to 0.79 on H (worse on 213 of 235), from 0.92 to 0.80 on V and from 0.96 to 0.86 on C. The floor spread σ_U on C moves away from 1, from 0.94 to 1.33. RD ρ_M falls on every radar, worse on 91 of 235 sequences on H and 53 of 63 on C, and RT-Pose's micro-Doppler ρ_S falls from 0.966 to 0.957 (worse on 29 of 61). SSIM on the CRP and range–azimuth maps also falls on both HuPR boards, worse on 188 and 177 of 235 sequences on H. The geometry does not move. Range–azimuth ρ reads 0.838 on C, and neither gap separates from the "None" row (HuPR 5 worse and 9 better, RT-Pose 4 and 2). The term therefore fits the floor and the Doppler planes, not the geometry.

The windowed pair has the smallest effect, on the person's cells in the scene. Without it, RT-Pose's clutter-removed range–azimuth ρ_c falls from 0.470 to 0.444, worse on 51 of 61 sequences, and its RAE ρ from 0.779 to 0.762. HuPR's gap widens from 16.4 to 17.4 AP (12 test sequences worse, 2 better), while its reconstruction stays at the "None" row, with range–azimuth ρ at 0.960 on H. RT-Pose's gap does not separate (2 worse, 2 better). The effect is the smallest of the three, and it rests on paired counts from one renderer seed.

The phase-coherence columns ($\Delta\phi_t$, $\Delta\phi_r$, $\Delta\phi_v$) do not separate the variants in a consistent direction, so we do not read them as evidence for or against any term.

Table 20: **Objective ablation, summary.** Each row removes the named terms of Eq. (4) and re-fits the renderer, with everything else at the shipped recipe (setup in App. H.1). "None" is the render Table 2 reports, not a re-fit. ρ is the median range–azimuth correlation and ρ_S the median micro-Doppler spectrogram correlation, over Table 2's sequences (HuPR both boards pooled). Δ is the median per-sequence gap to real on the test splits, read through the real-trained seeds of Table 3. On HuPR it is real-to-real minus rendered-to-real AP, and on RT-Pose rendered-to-real minus real-to-real MPJPE and MRPE in cm, so a larger Δ is a worse render. W/B counts the test sequences whose Δ is worse / better than the "None" row's by more than one AP point or 1 cm (MPJPE). One renderer seed. Tables 21–24 give every metric per radar.

	HuPR				**RT-Pose**				
Term removed	$\rho\uparrow$	$\rho_S\uparrow$	Δ (AP)$\downarrow$	W/B	$\rho\uparrow$	$\rho_S\uparrow$	Δ MPJPE$\downarrow$	Δ MRPE$\downarrow$	W/B
None (shipped recipe)	0.940	0.988	16.4	–	0.830	0.966	2.00	0.47	–
$-\mathcal{L}_{\mathrm{RAE},0}$ (global, loop 0)	0.485	0.989	16.5	9/5	0.141	0.968	174.26	262.43	13/0
$-\mathcal{L}_{\mathrm{RAE,pl}}$ (global, per plane)	0.941	0.986	16.7	5/9	0.838	0.957	2.30	1.21	4/2
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$ (windowed)	0.941	0.989	17.4	12/2	0.828	0.970	1.08	0.88	2/2

Table 21: **Objective ablation on the ADC samples.** Columns as Table 7 (ρ_{env} the envelope correlation after the rank-1 phase correction, $\Delta\phi_t$ the fast-time relative-phase coherence with its weighted RMSE in degrees); rows as Table 20, over the same sequences (HuPR 235 per board, RT-Pose 63 one-person). Bold: the best value in each column of a row section, by the column's arrow.

Term removed	$\rho_{\mathrm{env}}\uparrow$	PSNR↑	SSIM↑	RMSE↓	$\Delta\phi_t\uparrow$	$\Delta\phi_t$ (°)↓
HuPR, horizontal single-chip radar (H)						
None (shipped recipe)	**0.857**	**21.2**	**0.58**	**0.088**	0.55	63
$-\mathcal{L}_{\mathrm{RAE},0}$	0.742	18.4	0.45	0.121	**0.63**	**56**
$-\mathcal{L}_{\mathrm{RAE,pl}}$	0.856	**21.2**	**0.58**	**0.088**	0.54	64
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.856	**21.2**	**0.58**	**0.088**	0.55	64
HuPR, vertical single-chip radar (V)						
None (shipped recipe)	0.772	**19.0**	**0.48**	0.114	0.44	72
$-\mathcal{L}_{\mathrm{RAE},0}$	0.668	16.1	0.35	0.157	**0.56**	**62**
$-\mathcal{L}_{\mathrm{RAE,pl}}$	0.773	**19.0**	**0.48**	**0.113**	0.45	72
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	**0.776**	**19.0**	**0.48**	0.114	0.45	72
RT-Pose, cascade (C), one-person sequences						
None (shipped recipe)	0.667	20.0	0.32	0.100	0.24	88
$-\mathcal{L}_{\mathrm{RAE},0}$	0.596	18.2	0.26	0.123	**0.27**	**85**
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.668**	**20.2**	**0.33**	**0.098**	0.24	87
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.658	20.0	0.31	0.101	0.23	89

Table 22: **Objective ablation on the complex range profile (CRP).** Columns as Table 8 ($\Delta\phi_r$ and $\Delta\phi_v$ the relative-phase coherences along range and virtual channel, each with its weighted RMSE in degrees); rows, sequences and bold as Table 21.

Term removed	$\rho\uparrow$	PSNR↑	SSIM↑	RMSE↓	$\Delta\phi_r\uparrow$	$\Delta\phi_r$ (°)↓	$\Delta\phi_v\uparrow$	$\Delta\phi_v$ (°)↓
HuPR, horizontal single-chip radar (H)								
None (shipped recipe)	**0.856**	**26.0**	**0.65**	**0.052**	**0.66**	**56**	**0.75**	**48**
$-\mathcal{L}_{\mathrm{RAE},0}$	0.667	22.4	0.44	0.076	0.64	58	0.66	57
$-\mathcal{L}_{\mathrm{RAE,pl}}$	0.850	25.9	0.62	**0.052**	0.65	**56**	**0.75**	**48**
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.854	**26.0**	**0.65**	**0.052**	0.64	57	**0.75**	**48**
HuPR, vertical single-chip radar (V)								
None (shipped recipe)	0.785	**24.8**	**0.56**	0.061	0.55	65	0.47	71
$-\mathcal{L}_{\mathrm{RAE},0}$	0.491	20.6	0.26	0.095	**0.57**	**63**	**0.49**	**69**
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.789**	**24.8**	0.52	0.061	0.55	65	0.48	71
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.788	**24.8**	**0.56**	**0.060**	0.55	65	0.48	71
RT-Pose, cascade (C), one-person sequences								
None (shipped recipe)	0.735	28.2	**0.76**	0.039	0.50	68	**0.33**	**82**
$-\mathcal{L}_{\mathrm{RAE},0}$	0.227	23.7	0.52	0.066	**0.53**	**66**	0.27	85
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.739**	**28.4**	0.75	**0.038**	0.51	68	**0.33**	**82**
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.727	27.9	0.73	0.041	0.51	69	**0.33**	**82**

Table 23: **Objective ablation on range–angle and range–azimuth–elevation (RAE).** Columns as Table 9; rows, sequences and bold as Table 21. Every row is a 4DPS render, which carries an absolute scale, so SSIM, NRMSE and PSNR are scored directly, without the level matching that RF-Genesis needs.

	Range–angle					**RAE**			
Term removed	$\rho\uparrow$	$\rho_c\uparrow$	SSIM↑	NRMSE↓	PSNR↑	$\rho\uparrow$	SSIM↑	NRMSE↓	PSNR↑
HuPR, horizontal single-chip radar (H)									
None (shipped recipe)	**0.960**	**0.699**	**0.79**	**0.28**	34.0	**0.942**	**0.80**	0.33	**33.3**
$-\mathcal{L}_{\mathrm{RAE},0}$	0.646	0.456	0.67	1.60	19.0	0.628	0.66	1.87	18.7
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.960**	0.696	0.77	**0.28**	34.0	0.941	0.78	0.33	33.2
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	**0.960**	0.690	**0.79**	**0.28**	**34.1**	**0.942**	**0.80**	**0.32**	**33.3**
HuPR, vertical single-chip radar (V)									
None (shipped recipe)	0.925	**0.622**	**0.67**	**0.38**	**30.2**	0.904	**0.72**	**0.40**	30.7
$-\mathcal{L}_{\mathrm{RAE},0}$	0.337	0.126	0.56	10.23	1.6	0.349	0.56	11.71	1.7
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.926**	0.618	0.65	**0.38**	**30.2**	**0.905**	0.70	**0.40**	30.7
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.925	0.615	**0.67**	**0.38**	**30.2**	0.904	**0.72**	**0.40**	**30.8**
RT-Pose, cascade (C), one-person sequences									
None (shipped recipe)	0.830	0.470	**0.78**	0.53	31.8	0.779	**0.81**	**0.58**	**31.9**
$-\mathcal{L}_{\mathrm{RAE},0}$	0.141	0.166	0.65	7.79	9.9	0.125	0.65	7.08	11.7
$-\mathcal{L}_{\mathrm{RAE,pl}}$	**0.838**	**0.473**	0.77	**0.52**	**31.9**	**0.785**	0.80	**0.58**	**31.9**
$-(\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE},0}+\mathcal{L}^{\mathrm{win}}_{\mathrm{RAE,pl}})$	0.828	0.444	0.77	0.53	31.7	0.762	**0.81**	0.59	**31.9**

Table 24: **Objective ablation on range–Doppler (RD) and micro-Doppler.** Columns as Table 10 (P_M the rendered-over-real power ratio on the CFAR-occupied cells, L_U and σ_U the level and spread ratios on the unoccupied ones, ρ_{env} the Doppler-envelope correlation, BW the envelope separation, clDice the connectivity of the segmented signatures); rows, sequences and bold as Table 21. Every row is a 4DPS render, which carries an absolute scale, so P_M, L_U and σ_U are scored directly, without the level matching that RF-Genesis needs.

	RD occupied		**RD floor**		**Micro-Doppler**					
Term removed	$\rho_M \uparrow$	$P_M \to 1$	$L_U \to 1$	$\sigma_U \to 1$	$\rho_c \uparrow$	$\rho_\sigma \uparrow$	$\rho_S \uparrow$	$\rho_{\text{env}} \uparrow$	BW$\to 1$	clDice$\uparrow$
HuPR, horizontal single-chip radar (H)										
None (shipped recipe)	0.891	**0.71**	**0.92**	0.88	**1.00**	**0.98**	0.993	0.87	0.87	**0.91**
$-\mathcal{L}_{\text{RAE},0}$	0.892	0.70	**0.92**	0.88	**1.00**	**0.98**	0.993	0.87	0.87	**0.91**
$-\mathcal{L}_{\text{RAE,pl}}$	0.878	0.69	0.79	**0.94**	**1.00**	**0.98**	0.993	0.87	0.87	0.90
$-(\mathcal{L}^{\text{win}}_{\text{RAE},0} + \mathcal{L}^{\text{win}}_{\text{RAE,pl}})$	**0.894**	**0.71**	**0.92**	0.88	**1.00**	**0.98**	**0.994**	**0.88**	**0.88**	**0.91**
HuPR, vertical single-chip radar (V)										
None (shipped recipe)	0.795	**0.54**	**0.92**	0.88	**0.99**	0.94	0.984	0.80	**0.79**	**0.83**
$-\mathcal{L}_{\text{RAE},0}$	0.796	**0.54**	**0.92**	0.88	**0.99**	0.94	0.984	**0.81**	**0.79**	**0.83**
$-\mathcal{L}_{\text{RAE,pl}}$	0.781	**0.54**	0.80	**1.07**	**0.99**	0.94	0.982	0.80	0.78	**0.83**
$-(\mathcal{L}^{\text{win}}_{\text{RAE},0} + \mathcal{L}^{\text{win}}_{\text{RAE,pl}})$	**0.800**	**0.54**	**0.92**	0.88	**0.99**	**0.95**	**0.985**	**0.81**	**0.79**	**0.83**
RT-Pose, cascade (C), one-person sequences										
None (shipped recipe)	0.695	0.63	**0.96**	**0.94**	0.97	0.70	0.966	0.65	0.71	**0.71**
$-\mathcal{L}_{\text{RAE},0}$	**0.697**	0.63	**0.96**	**0.94**	0.97	0.69	0.968	0.63	0.71	**0.71**
$-\mathcal{L}_{\text{RAE,pl}}$	0.677	**0.64**	0.86	1.33	0.96	0.69	0.957	**0.67**	**0.72**	0.70
$-(\mathcal{L}^{\text{win}}_{\text{RAE},0} + \mathcal{L}^{\text{win}}_{\text{RAE,pl}})$	0.695	0.60	0.95	0.93	**0.98**	**0.71**	**0.970**	0.63	0.71	**0.71**

H.2 Runtime and Peak Memory

This subsection reports what the two experiments cost in time and in GPU memory: reconstructing every sequence (Sec. 4.1) with 4DPS, its unfit render and RF-Genesis, and training the downstream models (Sec. 4.2). Every timing in Tables 25 and 26 is a wall time from the run that produced the paper's numbers, one sequence or one training per GPU, for 4DPS and for RF-Genesis alike. The 4DPS fits, the unfit renders and every downstream training, including those on RF-Genesis's renders, ran on NVIDIA H100 GPUs. A 4DPS fit's time runs from the person stage through the 400-epoch velocity and material fit to the render and its evaluation, and excludes the scene preprocessing it inherits (App. B.1). The unfit render runs the same stages with no fit. RF-Genesis's time covers its mesh export, its ray-cast simulation, the representations computed from the simulated cube and their scoring against the recording.

Table 25: **Reconstruction cost** (Sec. 4.1): the median wall time per sequence and the total over the corpus. RT-Pose's unfit render covers the 57 split sequences it is scored on. The 4DPS step is the median over sequences of the training step's median. RF-Genesis ran on RTX 4090s and 4DPS on H100s, so the two methods' times are not a like-for-like speed comparison.

Data	**Method**	GPU	Sequences	Per sequence (min)	Total (GPU-h)	Step (ms)
HuPR	4DPS	H100	235	5.7	22.3	19.4
	4DPS (unfit)	H100	235	1.5	5.9	–
	RF-Genesis	RTX 4090	235	5.3	21.0	–
RT-Pose	4DPS	H100	100	12.9	21.6	19.3
	4DPS (unfit)	H100	57	1.9	1.9	–
	RF-Genesis	RTX 4090	100	6.0	10.6	–

Downstream training (Sec. 4.2). Table 26 gives the cost of one training per H100 at the protocol's fixed schedule, for every arm of Table 3. The real arm is shared: the RF-Genesis arms were compared against the same four real trainings, which were not retrained. A training's time is the sum over epochs of its training time, plus the per-epoch validation on HuPR.

Peak memory. Table 27 gives the peak GPU memory of each stage, and its caption says which counter each cell reads. The largest 4DPS fit allocates 19.6 GiB at its peak.

Table 26: **Downstream training cost** (Sec. 4.2): hours per training, mean (std) over the four seeds, and the total over the four seeds. Every training ran on one H100. Evaluating Table 3's five cells over the four seeds adds 0.5 (4DPS) and 0.5 (RF-Genesis) GPU-h on HuPR, and 0.7 and 0.7 on RT-Pose.

			HuPR			**RT-Pose**		
Arm	**Renderer**	GPU	h / training	Epochs	GPU-h	h / training	Epochs	GPU-h
Real	none (real)	H100	6.8 (0.1)	200	27.1	6.5 (0.1)	50	26.2
Rendered	4DPS	H100	6.7 (0.0)	200	26.8	8.4 (0.0)	50	33.7
Mixed	4DPS	H100	6.7 (0.0)	200	26.9	8.2 (0.2)	50	32.9
Rendered	RF-Genesis	H100	6.7 (0.0)	200	26.7	7.7 (0.1)	50	31.0
Mixed	RF-Genesis	H100	6.7 (0.0)	200	26.7	7.9 (0.3)	50	31.5

Table 27: **Peak GPU memory** (GiB): the median and the maximum over sequences for reconstruction, and the maximum over the four seeds for training. Every 4DPS cell and every training cell is PyTorch's allocator peak (`max_memory_allocated`), as the 4DPS runs record it per sequence and RT-Pose's trainer logs it; the reserved pool is larger. The RF-Genesis reconstruction cells are instead the card-level peak from nvidia-smi, because its ray tracer allocates memory outside PyTorch (PyTorch alone records 1.1 and 1.6 GiB), so they are not the same quantity as the 4DPS cells. The cells marked † were measured separately from the runs that produced the paper's numbers, with the same code, protocol and data: RF-Genesis as the maximum over the shortest, median and longest sequence of each dataset on an RTX 4090, and HuPR training over 300 steps and one validation pass per arm on an H100; the 5 HuPR arms are identical (21.9 GiB at card level). The unfit render ran at an earlier commit of the renderer, so its memory is not directly comparable with the fit's; on RT-Pose it peaks above it.

			HuPR		**RT-Pose**	
Stage	**Method / arm**	GPU	median	max	median	max
Reconstruction	4DPS	H100	5.7	6.7	18.7	19.6
(Sec. 4.1)	4DPS (unfit)	H100	3.6	3.8	20.2	20.3
	RF-Genesis	RTX 4090	–	3.3†	–	10.8†
Training	real	H100	–	14.9†	–	11.2
(Sec. 4.2)	4DPS rendered	H100	–	14.9†	–	11.2
	4DPS mixed	H100	–	14.9†	–	11.2
	RF-Genesis rendered	H100	–	14.9†	–	11.2
	RF-Genesis mixed	H100	–	14.9†	–	11.2

The shipped step. Three decisions make a corpus-scale fit affordable. First, the step has no data-dependent shape and no host synchronization: the path count is fixed per step and the loss chain and frames are batched, so the whole training step is captured once as a CUDA graph and replayed. Second, the forward model runs in fused CUDA kernels with hand-written backward passes, and the Doppler-plane amplitudes are evaluated in closed form (Eq. (16)). Third, everything in Eq. (3) that does not read a normal or a material is cached per frame, for the scene and for the person's paths, and the person's paths are sorted once per frame by their target cell, so each accumulator cell is one contiguous run summed without atomics.

I LIMITATIONS

Single-bounce transport, and one mover. As in prior radar splatting (Armouti et al., 2026a; Kung et al., 2025), we render the single-bounce path from each transmitter to a point and back, so the returns that reach a receiver only after reflecting off a second surface are absent from every product. Only the subject moves: the scene is fitted once per sequence and held static, and one person is modeled, because RT-Pose annotates the pose of only one person in its multi-person sequences. Its 37 two-person sequences are therefore fitted against the annotated person alone, and the paper reports results on the one-person sequences.

Motion and labels come from the video's body model. Each point's velocity stays within a bounded residual of the kinematics the monocular body model supplies, and the skeleton comes from that model, never from a radar label. The fit therefore corrects the video's motion rather than recovering it independently: a limb the body model misplaces beyond the bound is a limb the radar term cannot invent, and the bound is what keeps the Doppler fit from absorbing scaffold error as spurious velocity. RT-Pose's labels are derived on every radar frame from the same registered mesh that drives the render, with the released-label result reported beside them (Sec. 4).

Elevation is the weakest axis. The scaffold reconstructs elevation least well of the three spatial axes, because it is built from a monocular depth or LiDAR scene and a body model rather than measured by the radar. The products that resolve elevation are therefore the harder ones, which is why HuPR's vertical board and the RT-Pose cascade score below HuPR's horizontal board throughout Sec. 4.1. Two of HuPR's encoder taps are also undefined in the representation analysis, because the temporal merges feeding them are denormal in some trained checkpoints (App. F).

No fitted baseline exists to compare against. No public radar simulator takes a capture as input, so RF-Genesis is driven forward from the registered motion rather than fitted to the recording we score against. The comparison that carries Sec. 4.1 is therefore against our own unfit render, which isolates what the radar fit adds over the camera but cannot show what a competing fitted renderer would achieve. The survey in App. C.2 records that this is a property of the field rather than of our selection.

What the equivalence measurements cannot settle. An equivalence claim needs a control that detects a difference of a known size, and the only one we trained is the untrained network. It separates cleanly at the shallowest taps and barely at all at the backbone output and the head, so the pre-registered margin means less the deeper the tap, and we read the two deepest taps as degenerate rather than as passes. We also did not train a second real arm per seed from a shared initialization, so the same-initialization row is read against the different-initialization real–real mean rather than against a reference of its own (App. F).

Rendered features transfer on one dataset and not the other. Freezing each backbone and fitting a fresh linear head on real data recovers most of the rendered arm's deficit on RT-Pose and none of it on HuPR (Sec. 4.3). We report the asymmetry rather than explain it: the two datasets differ in radar geometry, in downstream architecture and in label provenance at once, so this evaluation cannot attribute the difference to any one of them.

Most edits are not scored. Fitting an explicit scene is what makes translation, retiming and recomposition possible. One edit, rendering a fitted person in another fitted scene of the same environment, is evaluated in App. G.1; translation, retiming and new motions are not scored, because no real recording exists to compare them against. A corpus of edited sequences is the next step.

J ADDITIONAL QUALITATIVE RESULTS

Figs. 11–16 show 18 sequences per dataset, six per figure, chosen by a rule fixed before rendering so that they are typical rather than the best: we rank the corpus of Table 2 (HuPR on its horizontal board) by 4DPS fidelity (the mean range–azimuth, range–Doppler and micro-Doppler correlation), keep Fig. 3's three RT-Pose sequences and the three HuPR test sequences ranked the same way, and add 15 at evenly spaced quantiles of the rest. Columns, frames, normalization, scores and the first three rows follow Fig. 3; the others are range–azimuth–elevation (azimuth $\times$ elevation, maximum over the person's range ± 10 bins), the complex range profile (range $\times$ virtual channel) and ADC samples (fast-time sample $\times$ channel). An undefined score prints as –.

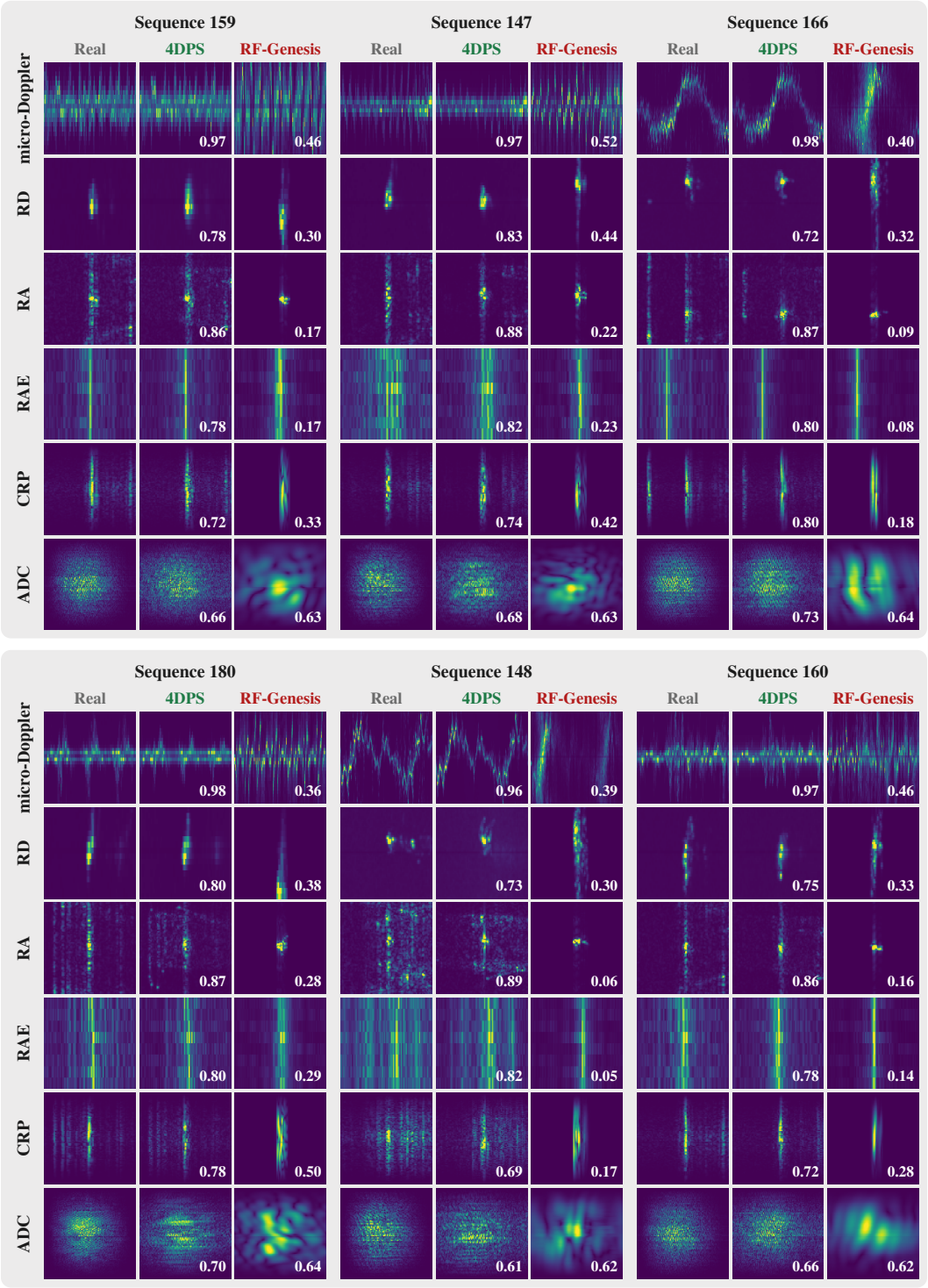

Figure 11: **Qualitative gallery, RT-Pose (1 of 3): sequences 159, 147 and 166 (top) and 180, 148 and 160 (bottom)** (4DPS-fidelity ranks 8, 1, 16, 4, 9 and 13 of 63). The top three are the sequences of Fig. 3. Selection rule, rows, columns, frames and scores are given at the start of App. J.

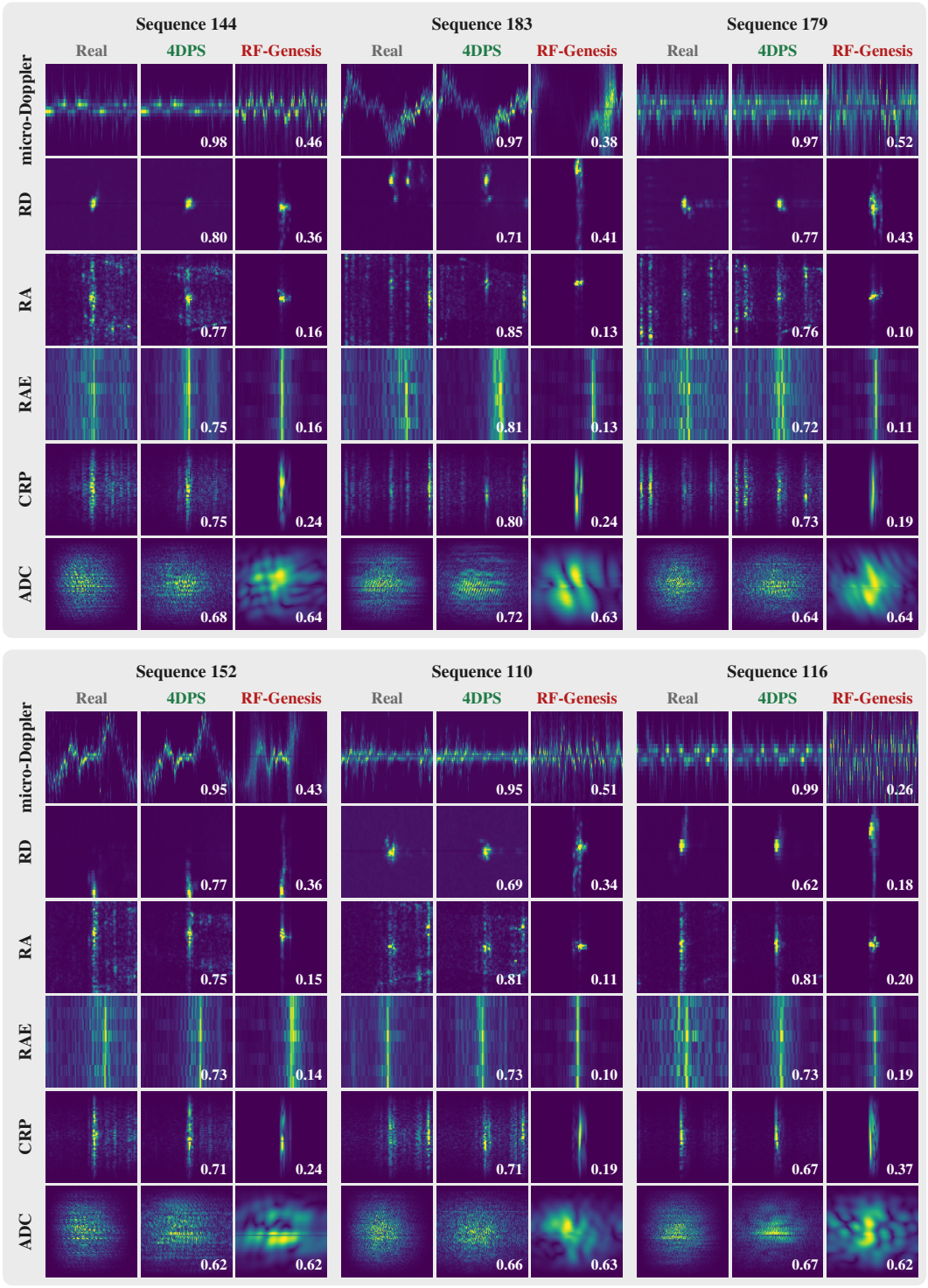

Figure 12: **Qualitative gallery, RT-Pose (2 of 3): sequences 144, 183 and 179 (top) and 152, 110 and 116 (bottom)** (4DPS-fidelity ranks 18, 22, 26, 30, 34 and 38 of 63). As Fig. 11.

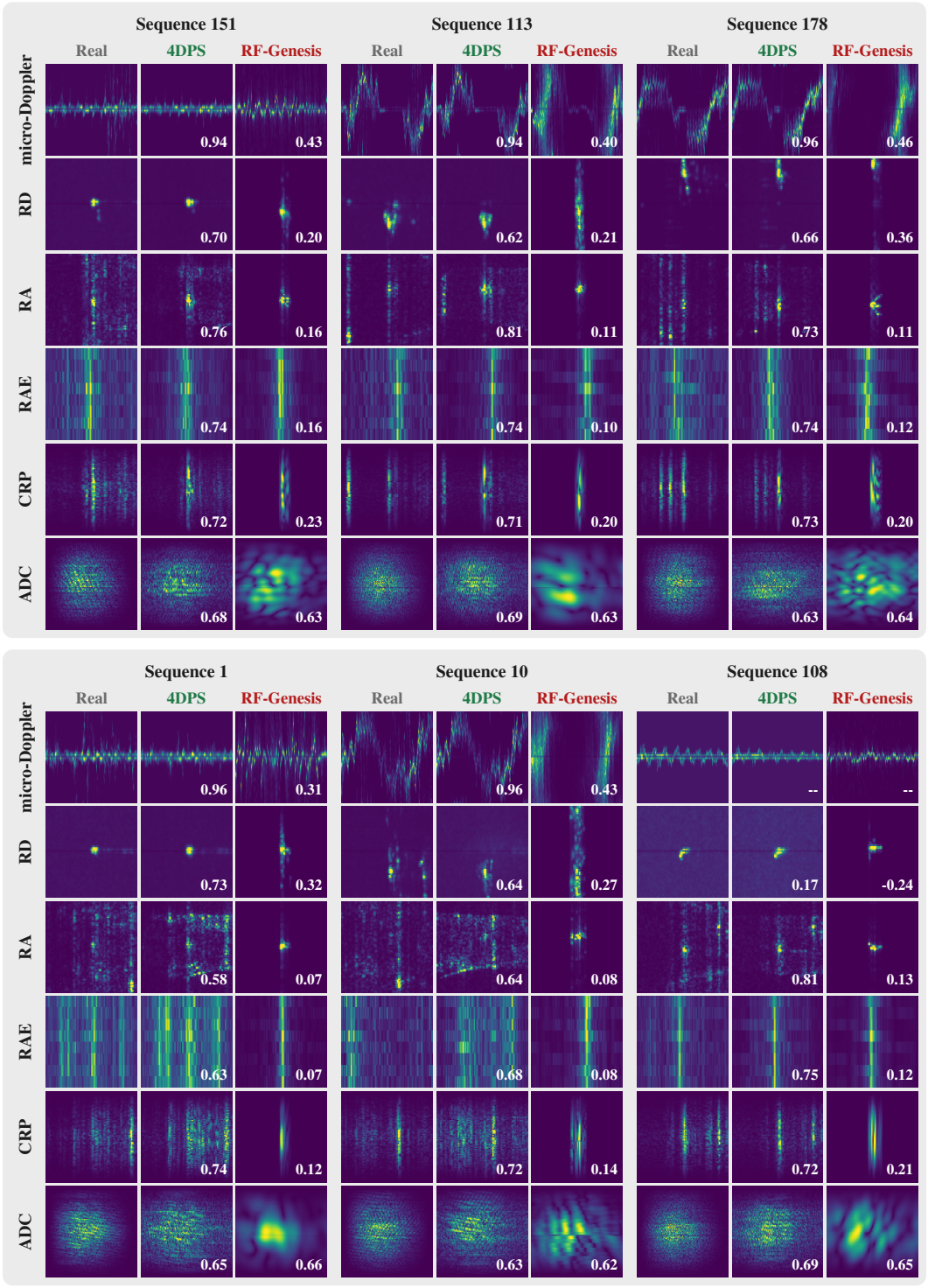

Figure 13: **Qualitative gallery, RT-Pose (3 of 3): sequences 151, 113 and 178 (top) and 1, 10 and 108 (bottom)** (4DPS-fidelity ranks 42, 46, 50, 54, 58 and 62 of 63). As Fig. 11.

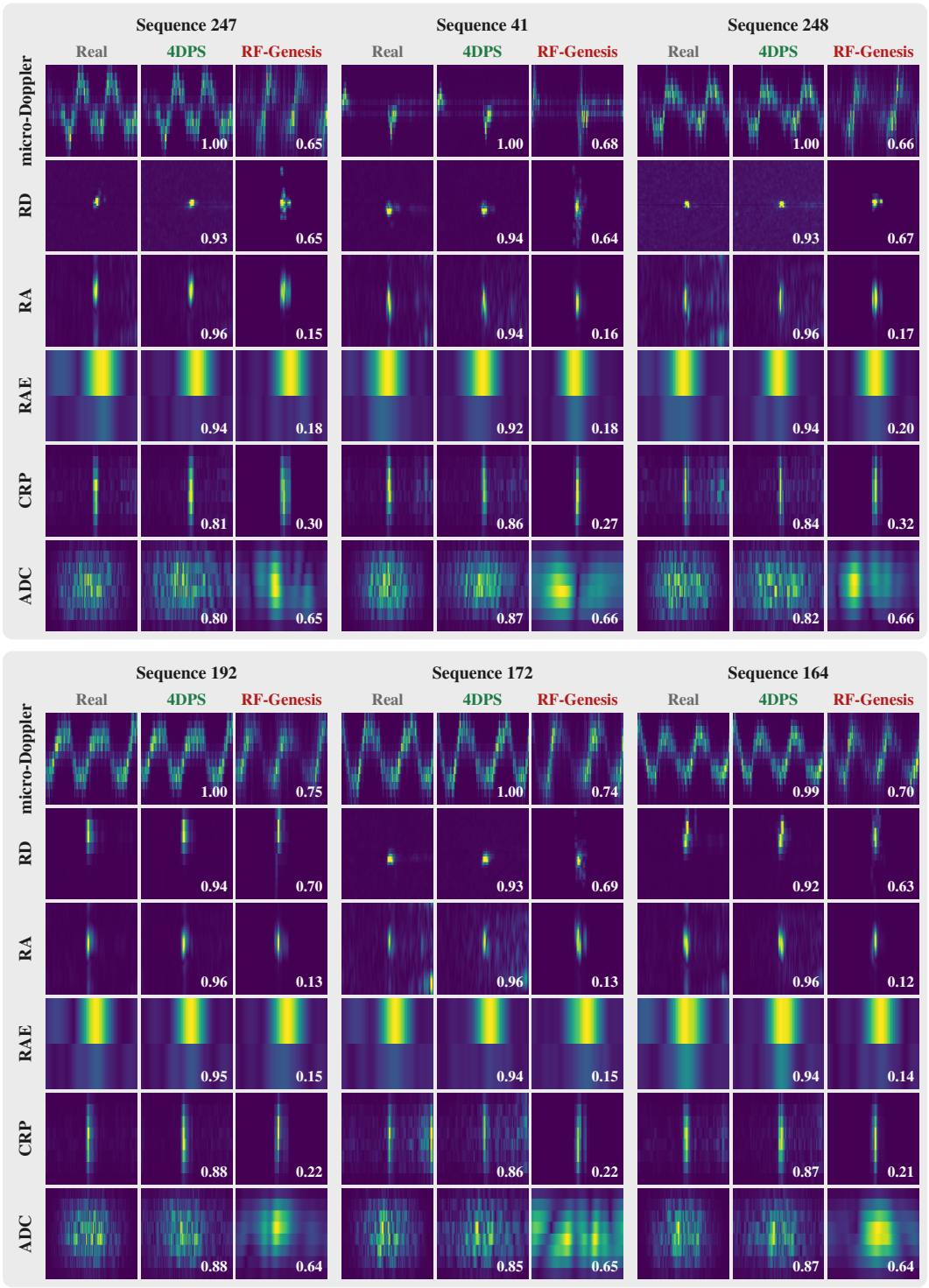

Figure 14: **Qualitative gallery, HuPR, horizontal board (1 of 3): sequences 247, 41 and 248 (top) and 192, 172 and 164 (bottom)** (4DPS-fidelity ranks 11, 57, 13, 8, 26 and 41 of 235). The top three are the HuPR test sequences ranked like those of Fig. 3 (4DPS fidelity plus rendered-to-real OKS). A single board resolves elevation poorly, so the range–azimuth–elevation rows are coarse along that axis.

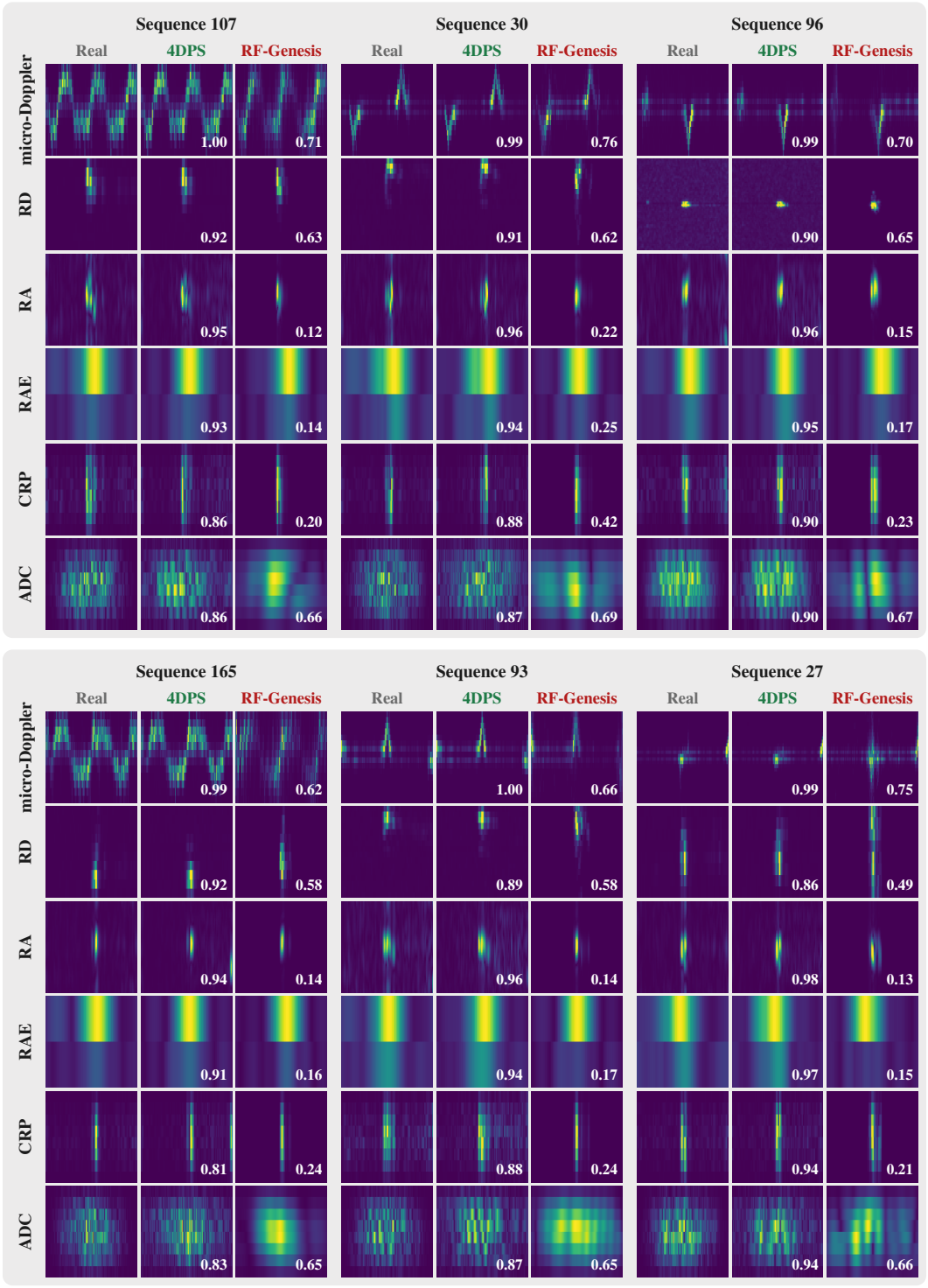

Figure 15: **Qualitative gallery, HuPR, horizontal board (2 of 3): sequences 107, 30 and 96 (top) and 165, 93 and 27 (bottom)** (4DPS-fidelity ranks 58, 73, 89, 104, 120 and 135 of 235). As Fig. 14.

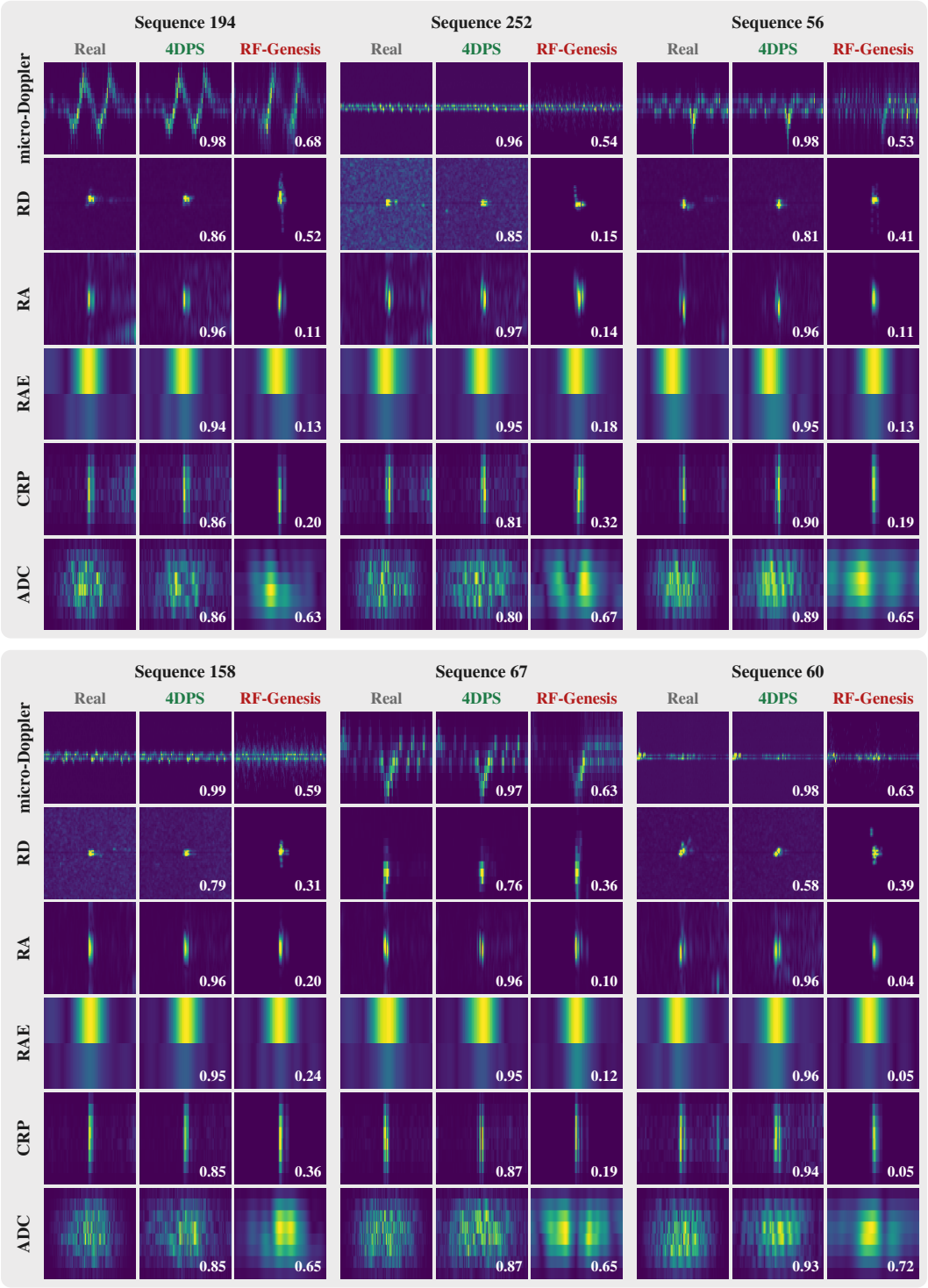

Figure 16: **Qualitative gallery, HuPR, horizontal board (3 of 3): sequences 194, 252 and 56 (top) and 158, 67 and 60 (bottom)** (4DPS-fidelity ranks 150, 166, 181, 197, 212 and 228 of 235). As Fig. 14.