A. Overview

The supplementary material includes the subsequent components.

- **Relative Concept Explanation**
  - Explanation of the up-to-scale estimation of the SLAM system (Sec. B – Sec. 3.3).
  - Implementation details and numerical quantification regarding spectrogram analysis (Sec. C – Sec. 3.4).
- **Supplementary Experiments and Analysis.**
  - Details of the main experiment (Sec. D – Sec. 3.2 & Sec. 4.1).
  - Ablation on sliding window size shown by camera coordinate human reconstruction metrics (Sec. E – Sec. 5).
  - Ablation on different masking strategies for world HMR precision. (Sec. F – Sec. 5).
- **Additional Visualization**
  - World HMR visualization on custom videos (Sec. G – Sec. 4.2).
  - World HMR visualization on multi-individual and diverse scene cases (Sec. H – Sec. 4.2).
  - Visualization on ablation results (Sec. I – Sec. 5).
  - Visualization on failure cases. (Sec. J – Sec. 6).
- **Supplementary Video (on website)**

B. Up-to-Scale SLAM System

In Sec. 3.3, we adopt MASt3R-SLAM [4] to estimate the world-coordinate camera trajectory and mentioned its up-to-scale reconstruction of camera transformation.

Built upon the two-view 3D reconstruction model MASt3R [3], MASt3R-SLAM [4] is among the first to integrate strong 3D priors into an incremental SLAM framework, achieving fast inference of up to 15 FPS on a single RTX 4090 GPU. Although MASt3R [3] is trained with a metric regression loss, where the normalization factor used in its predecessor DUSt3R [5] is removed to enable metric-scale reconstruction, MASt3R-SLAM still produces up-to-scale results.

This is because MASt3R-SLAM does not explicitly constrain the incremental camera pose as $\mathbf{T}_{cw} \in \mathbf{SE}(3)$ during training. Instead, it defines all camera poses as $\mathbf{T}_{cw} \in \mathbf{Sim}(3)$, allowing an additional scale component. Consequently, while the scale of the estimated scene and camera trajectory are temporally consistent, they remain up-to-scale rather than metrically scaled. We estimate the scaler s from metric depth. In practice, we find that the scaler is around 1 in most of the testset sequences, indicating an inheritated metric scale reconstruction ability from MASt3R [3]. But for texture-less and highly dynamic sequences, the scaler factor drifts away from 1 and plays an important role in metric scale recovery.

C. Details about Frequency Domain Metrics

In Section 3.4, we propose a frequency-domain representation of jittering effects. The mathematical formulation and implementation details are as follows.

As mentioned in the main paper, we obtain the motion spectrogram using the Short Time Fourier Transform (STFT) to preserve both the time-axis and frequency-axis. The reason for directly flattening the 3D joint position to 1D is that we care about jittering among all dimensions of the space and all joints. So there is no subsample of joints or separation of xyz axes.

We use a Hann window with length $N_w = n_{\text{fft}} = 128$, hop length $L = 32$. After transforming the flattened signal using STFT, we apply $\text{abs}(\cdot)$ to get the magnitude, and interpolate it to the original sequence length. After these operations, the spectrogram $|\mathbf{S}(i, f)|$ has its y-axis representing frequency bins, and x-axis denoting temporal frame index.

Following prior work on signal processing [2], we calculate the root mean square error (RMSE) and correlation (Corr) based on the spectrogram statistics, and derive metrics based on that:

$$\text{MSE} = \frac{1}{N} \sum_{i,f} \left(S^{\text{gt}}_{i,f} - S^{\text{pred}}_{i,f} \right)^2,$$

$$\text{RMSE} = \sqrt{\text{MSE}}, \qquad \text{RMSE}_{\text{norm}} = 100 \times \frac{\text{RMSE}}{\sigma_{\text{gt}} + \epsilon}.$$

where σ_{gt} is the standard deviation of $S^{\text{gt}}_{i,f}$. For Corr, we first calculate its mean:

$$\mu_{\text{gt}} = \frac{1}{N} \sum_{i,f} S^{\text{gt}}_{i,f}, \qquad \mu_{\text{pred}} = \frac{1}{N} \sum_{i,f} S^{\text{pred}}_{i,f}.$$

Then the correlation coefficient is:

$$\text{Corr} = \frac{\sum_{i,f} \left(S^{\text{gt}}_{i,f} - \mu_{\text{gt}} \right) \left(S^{\text{pred}}_{i,f} - \mu_{\text{pred}} \right)}{\sqrt{\sum_{i,f} \left(S^{\text{gt}}_{i,f} - \mu_{\text{gt}} \right)^2 \sum_{i,f} \left(S^{\text{pred}}_{i,f} - \mu_{\text{pred}} \right)^2}}$$

$$\text{Corr}_{\text{norm}} = 100 \times \frac{1 - \text{Corr}}{2}.$$

We use the $\text{RMSE}_{\text{norm}}$ and the $\text{Corr}_{\text{norm}}$ as dependent metrics and show an example of our OnlineHMR checkpoints trained by 1K iters and 52K iters. Both metrics are the lower the better, as shown in Fig. 1.

Table 1. RMSE↓ / Corr ↓ results.

Dataset	GVHMR	TRAM	Ours
3DPW	3.59 / 0.02	17.01 / 0.64	19.91 / 0.82
EMDB-1	75.52 / 0.12	25.82 / 1.24	24.98 / 1.41

We report camera coordinate results on recent methods and ours, as shown in Tab. 1. Ours (focus online) is comparable to the offline baseline TRAM in frequency amplitude

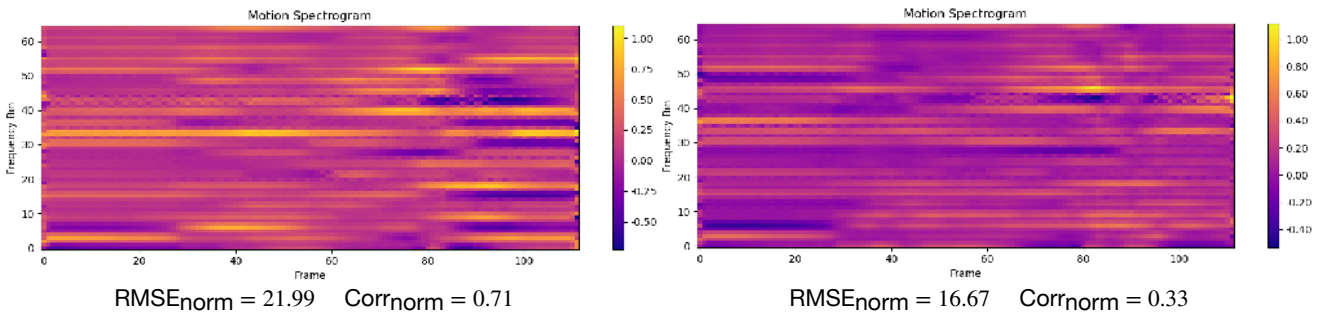


Figure 1. Difference spectrograms computed as GT-Pred, visualizing discrepancies in the time–frequency domain. The left figure corresponds to a model trained for 1K iterations, and the right for 52K iterations. Lower values (darker/closer to zero) indicate better alignment with the ground truth. The converged model (52K) exhibits substantially reduced differences.

(RMSE) and pattern (Corr) distribution similarity w.r.t GT. Interestingly, GVHMR shows better pattern reconstruction but slightly larger deviation in amplitude on EMDB-1.

D. Details about the Main Experiment

Loss functions. The standard per-frame HMR loss function in Sec.3.2 includes 3D keypoints, 2D keypoints, SMPL parameters, and 3D vertices components, shown as.

$$\mathcal{L}_f = \lambda_1\mathcal{L}_{2D} + \lambda_2\mathcal{L}_{3D} + \lambda_3\mathcal{L}_{\text{SMPL}} + \lambda_4\mathcal{L}_V \tag{1}$$

We denote $\mathbf{J}_{3D}$ as the integration of 3D position for all the joints $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_j, ..., \mathbf{p}_J\}$. Similarly, $\boldsymbol{\Theta}$ indicates the SMPL pose parameters, $\boldsymbol{V}$ denotes 3D vertices for all the joints. Then each component in (1) is:

$$\mathcal{L}_{2D} = \left\| \hat{\mathbf{J}}_{2D} - \Pi(\mathbf{J}_{3D}) \right\|_F^2, \tag{2}$$

$$\mathcal{L}_{3D} = \left\| \hat{\mathbf{J}}_{3D} - \mathbf{J}_{3D} \right\|_F^2, \tag{3}$$

$$\mathcal{L}_{\text{SMPL}} = \left\| \hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta} \right\|_2^2, \tag{4}$$

$$\mathcal{L}_V = \left\| \hat{\mathbf{V}} - \mathbf{V} \right\|_F^2, \tag{5}$$

where $\hat{}$ indicates the ground truth, F is the total frame number, and Π is the projection function. The loss weights $\lambda_1 = 5.0$, $\lambda_2 = 5.0$, $\lambda_3 = 1.0$, $\lambda_4 = 1.0$. And the velocity regularization weights mentioned in the main paper are $\lambda_5 = 10.0$, $\lambda_6 = 5.0$.

Parameters setup. In training, we first process the input video feature sequence into 16-frame chunks, with training batch size=24, as used in TRAM [6]. Then apply a window slicing using length $N = 3$, $stepsize = 1$. Within each window, we estimate the result of frame $N-1$ only, while taking frame $N-3$ and $N-2$ as conditions, which we name as ***intra-window information fusion***. This is supervised by frame-level HMR losses. Then, for ***inter-window temporal modeling***, we stack the output of all windows together into a 14-frame chunk, and add additional supervision of velocity regularizations. The model is trained on an Nvidia 80GB H100 GPU.

E. More Ablation on Sliding Window Size.

We ablate on the sliding window size 3-6 as reported in Tab. 2. Results show the accuracy reach optimal at window size=4, but Accel increases along with the window size getting larger. We infer this is due to the temporal sensitivity of Accel Metric. Larger windows fuse more past frames, introducing a delay effect that amplifies the acceleration deviations from GT.

Table 2. Ablation study on sliding window size (SWS).

	3DPW				EMDB-1			
SWS	PA-MPJPE↓	MPJPE↓	PVE↓	ACCEL↓	PA-MPJPE↓	MPJPE↓	PVE↓	ACCEL↓
3	43.7	69.9	83.7	**6.4**	**46.0**	74.0	86.1	**9.0**
4	**41.7**	**64.9**	**78.6**	**6.4**	46.8	**73.8**	85.3	9.1
5	43.2	70.8	85.7	6.6	46.7	74.5	86.5	9.4
6	42.5	65.5	80.1	6.6	46.9	74.3	**84.4**	9.6

F. More Ablation on Masking Strategy

Table 3. Comparison of different masking strategies on world coordinate human reconstruction upon MASt3R-SLAM [4]

Strategy	WA-MPJPE	W-MPJPE	RTE	ERVE
Vanilla	119.6	412.9	4.1	14.4
Hard Mask	112.6	386.8	3.2	13.3
Soft Mask	93.5	310.4	2.2	12.4

In Sec. 5, we provided a comparison of the camera trajectory accuracy on different masking strategies. Here, we additionally present the world coordinates human reconstruction metrics that are also affected by the camera trajectory estimation results, as shown in Tab. 3.

G. Additional Visualization on Custom Video

As shown by Fig. 5 and Fig. 7, our OnlineHMR reconstructs a more faithful world coordinate camera trajectory and human mesh compared to concurrent work Human3R [1]. Fig. 7 additionally shows that Human3R sometimes loses tracking of the person (blue) in the video and mixes up with a second person (red). Dynamic results are in the supplementary video.

H. Visualization on Multi-Individual and Diverse Scene

As stated in the main paper, our main focus is on world coordinate human mesh recovery. However, we can also obtain the scene point cloud from the SLAM part. We illustrate an example of a human mesh and camera trajectory with scene geometry.

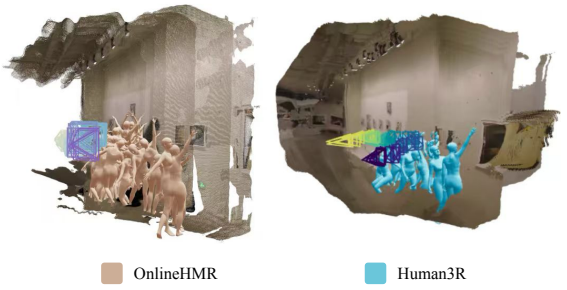


Figure 2. Using the same video input with Fig. 5, we show the global human mesh, camera trajectory, and the final updated scene.

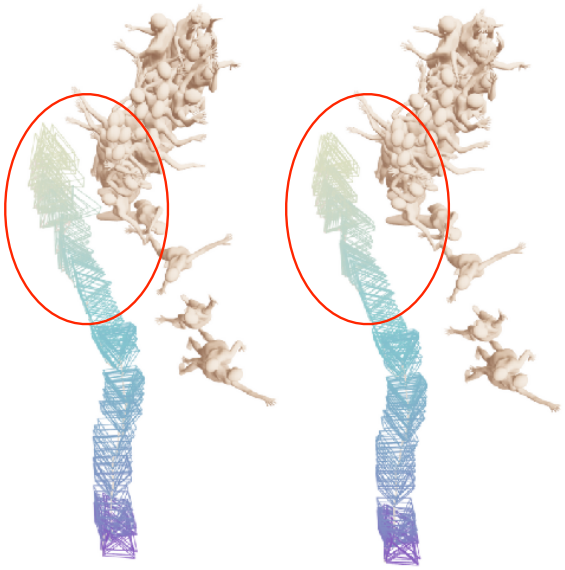
Our method is also able to generalize to multi-individual and complex scene scenarios. Examples are shown in Fig. 6.

I. Visualization on Ablation Results

EMA Correction. As stated in Sec. 3.3, EMA correction on camera translation and rotation indirectly imposes a smoothness for world coordinate human motion. We show a qualitative comparison of custom video in Fig. 3. The results demonstrate less jittering effect on the camera and world coordinate human translation with EMA correction.

J. Visualization on Failure Cases

As shown in Fig. 4, our method fails on inputs with repetitive textures and dynamic environments, such as shadow. Also, the current design is not suitable for camera switching cases.



w/o EMA Correction w/ EMA Correction

Figure 3. Qualitative results w/ and w/o EMA correction on custom videos.

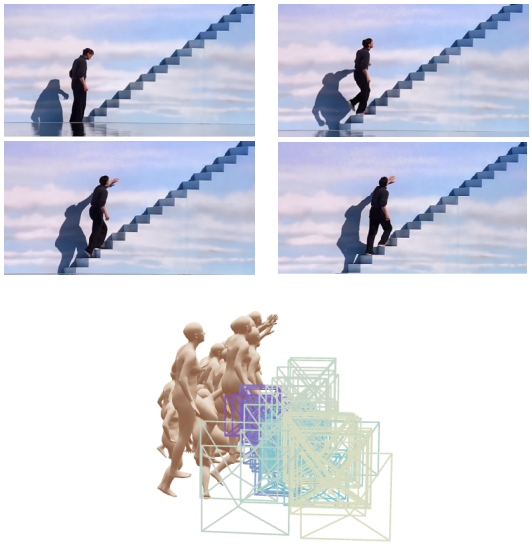


Figure 4. Example of failure cases.

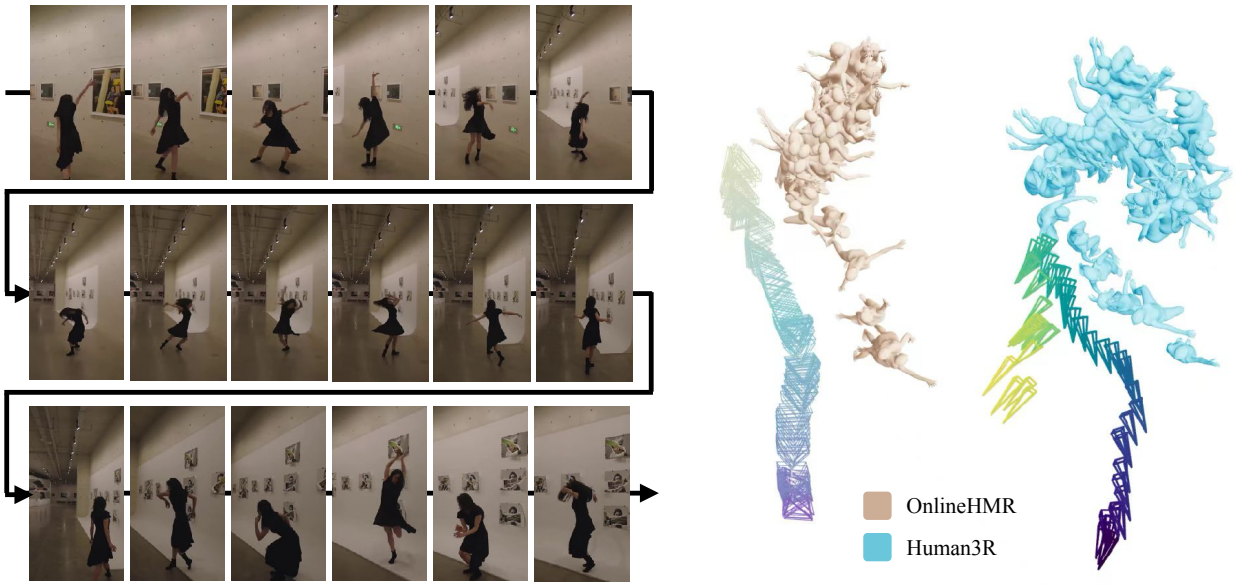


Figure 5. Quantitative comparison of OnlineHMR and Human3R on a custom video of a famous dancer. OnlineHMR produces a faithful reconstruction of the human trajectory, whereas Human3R yields trajectories that appear compressed and crowded together.

Figure 6. More examples with multiple individuals and a diverse scene. Dynamic results can be found on the demo page.

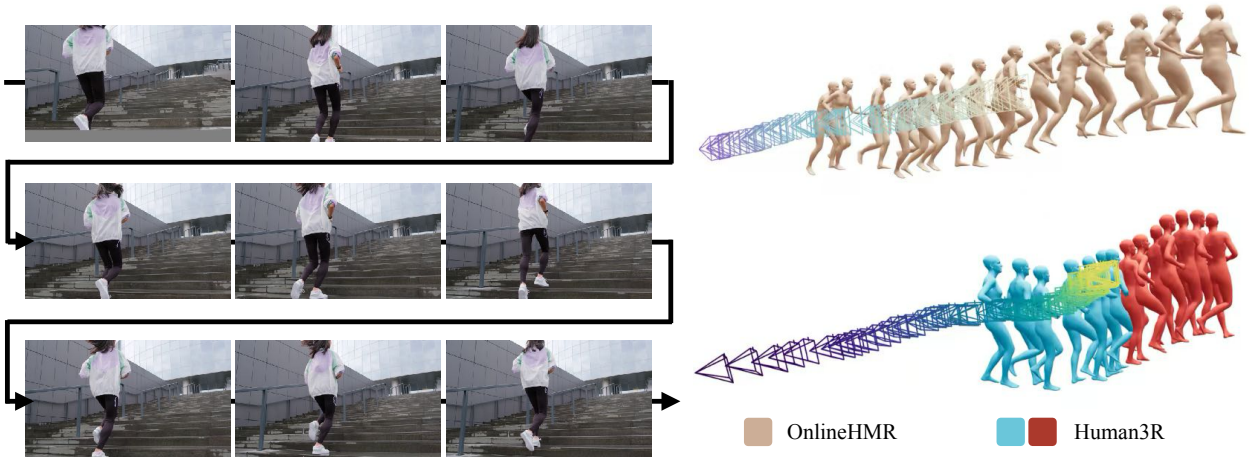


Figure 7. Quantitative comparison of OnlineHMR and Human3R on a custom video of running up stairs.

References

[1] Yue Chen, Xingyu Chen, Yuxuan Xue, Anpei Chen, Yuliang Xiu, and Gerard Pons-Moll. Human3r: Everyone everywhere all at once. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[2] R. Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, pages 125–128 vol.1, 1993. 1

[3] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXII*, 2024. 1

[4] Riku Murai, Eric Dexheimer, and Andrew J Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16695–16705, 2025. 1, 2

[5] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 1

[6] Yufu Wang, Ziyun Wang, Lingjie Liu, and Kostas Daniilidis. Tram: Global trajectory and motion of 3d humans from in-the-wild videos. In *European Conference on Computer Vision*, pages 467–487. Springer, 2024. 2