ClimbingCap: Multi-Modal Dataset and Method for Rock Climbing in World Coordinate —Supplementary Material

Ming Yan[1,2,3*] Xincheng Lin[1,3*] Yuhua Luo[1,3] Shuqi Fan[1,3] Yudi Dai[7] Qixin Zhong[4]
Lincai Zhong[5] Yuexin Ma[6] Lan Xu[6] Chenglu Wen[1,3] Siqi Shen[1,3†] Cheng Wang[1,3]
[1]Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University
[2]National Institute for Data Science in Health and Medicine, Xiamen University
[3]Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, School of Informatics, Xiamen University
[4]China National Climbing Team [5]Ningbo Sports Work Training Team
[6]ShanghaiTech University [7]ETH AI Center, ETH Zürich

We would like to thank the reviewer for reading the supplementary material. Thanks for your time and effort to review this work.

In Appendix A, we describe the performance of multiple state-of-the-art methods after training from scratch on the AscendMotion dataset. For evaluation, we report multiple metrics, including Procrustes-Aligned mean per-joint position error (PMPJPE), which corresponds to the World-Aligned MPJPE (WA-MPJPE) calculated by aligning each segment to the ground truth, mean per-joint position error (MPJPE) aligned to the first two frames (W-MPJPE), percent correct keypoints (PCK) as a percentage indicator, Per Vertex Error (PVE), root translation error (RTE), motion jitter (Jitter, in m/s^3), relative global translation error (T-Error, in meters), and acceleration error (ACCEL, in mm/s^2). We analyze these experimental results in depth, following the motion evaluation protocols of [12, 13] and the global trajectory evaluation protocols of [3, 19]. We discuss the content of the experimental results in more depth. Appendix B shows additional experiments with planar motion. In Appendix C, the cross-dataset evaluation results are provided, we show that AscendMotion is a good addition to today's human motion recovery community. In Appendix E and Appendix D, we present the details of the AscendMotion dataset and the ClimbingCap algorithm.

A. Retrain results in AscendMotion

To evaluate the performance of various state-of-the-art HMR methods on the AscendMotion dataset, we categorized the scene into horizontal and vertical based on the primary direction of human motion. Vertical scenes pose more significant challenges for both climbers and HMR methods compared to horizontal scenes.

Figure 1. Rock climbing wall in the AscendMotion dataset. The top and the middle row of the scene are horizontal scenes (horizontal walls), which are not high and the major direction of motions is horizontal. The bottom row are vertical scenes (vertical walls), which are high and more challenging to climbers than the horizontal scenes. In each rock climbing wall, a SMPL model with T-pose is placed close to the wall.

In this section, GVHMR [12], LiveHPS [11], LEIR [19], are retrained from scratch based on the AscendMotion dataset. The retrained results of these methods (trained from scratch) are marked with the symbol '*', while the results

Modality	Method	Camera Coordinate					World Coordinate				
		ACCEL↓	MPJPE↓	PA-MPJPE↓	PVE↓	PCK0.3↑	WA-MPJPE↓	W-MPJPE↓	RTE↓	Jitter↓	T-Error↓
RGB	GVHMR [12]	26.22	124.60	80.30	151.10	0.71	1002.11	1442.50	7.91	32.71	2.54
	GVHMR* [12]	25.86	123.81	86.40	151.82	0.71	1112.71	1413.28	8.50	30.81	2.22
LiDAR	LiveHPS [11]	195.23	147.31	121.76	189.30	0.70	1369.89	1506.50	10.45	358.54	6.73
	LiveHPS* [11]	84.07	123.84	101.48	145.53	0.74	1064.50	1103.62	9.85	255.02	5.54
LiDAR+RGB	LEIR [19]	94.57	299.62	150.56	351.52	0.37	1313.09	1435.92	9.97	85.03	1.20
	LEIR* [19]	58.14	141.06	95.73	174.11	0.67	388.33	539.62	3.50	48.17	**1.08**
	Ours	**17.25**	**88.92**	**74.50**	**106.42**	**0.78**	**85.26**	**106.95**	**3.12**	**27.75**	1.29

Table 1. **HMR Retrain Comparison in AscendMotion Dataset(Vertical Scene). * indicates the results of this method have been retrained from scratch based on AscendMotion.**

Modality	Method	Camera Coordinate					World Coordinate				
		ACCEL↓	MPJPE↓	PA-MPJPE↓	PVE↓	PCK0.3↑	WA-MPJPE↓	W-MPJPE↓	RTE↓	Jitter↓	T-Error↓
RGB	GVHMR [12]	4.50	107.09	60.06	118.89	0.77	105.15	202.45	4.09	6.85	1.48
	GVHMR* [12]	5.17	91.82	58.86	106.31	0.82	110.46	218.23	4.16	8.01	1.38
LiDAR	LiveHPS [11]	157.87	156.5	142.19	191.87	0.64	235.4	392.34	13.94	279.96	2.1
	LiveHPS* [11]	25.65	92.42	83.29	115.37	0.82	155.81	387.57	16.48	43.48	2.52
LiDAR+RGB	LEIR [19]	110.18	297.95	187.26	340.61	0.41	266.82	282.31	9.78	73.38	1.1
	LEIR* [19]	11.04	139.90	123.56	165.15	0.65	221.65	259.91	8.56	57.15	1.08
	Ours	**5.17**	**75.45**	**61.73**	**94.89**	**0.91**	**62.95**	**78.99**	**1.57**	**8.3**	**1.07**

Table 2. **HMR Retrain Comparison in AscendMotion Dataset(Horizontal Scene). * indicates the results of this method have been retrained from scratch based on AscendMotion.**

of methods without retraining are unmarked. The experimental results for the vertical scene (vertical rock walls) and the horizontal scenes are shown in Tab. 1 and Tab. 2, respectively. The upper rows present the results of multiple state-of-the-art RGB-based methods, while the lower rows focus on LiDAR-based and LiDAR+RGB-based methods.

In vertical scenes, ClimbingCap significantly surpasses all other methods. The second-best performing method, GVHMR, a representative global HMR algorithm, struggles in vertical scenes, primarily because it estimates movement direction based on horizontal velocity predictions. However, vertical scenes in AscendMotion feature upward climbing as the dominant motion pattern, which GVHMR fails to capture effectively.

LiDAR-based and LiDAR+RGB-based methods underperform compared to ClimbingCap, as they fail to adequately account for global trajectories and the relationship between camera coordinates and global coordinates. These findings underscore the importance of integrating information from both camera and world coordinates to improve HMR performance on challenging vertical climbing scenes.

In horizontal scenes, ClimbingCap outperforms other methods in all the metrics, except the PA-MPJPE metric. It performs significantly better than others in all the world coordinate metrics such as WA-MPJPE and W-MPJPE. WHAM and GVHMR achieve good performance in camera coordinate metrics, their performance in world coordinate metrics is less competitive to ClimbingCap.

In order to intuitively understand the results of the world coordinate metrics of different methods, we visualize the trajectory of one of the horizontal and vertical rock walls in the test set, as shown in Figure 2. It can be seen that the comparison methods can hardly restore the route of the trajectory, and our proposed method is closest to the real trajectory.

B. Additional Planar Motion Experiments

To verify the capability of ClimbingCap on other planar motions, we conducted experiments on the RELI11D [19] dataset. In these experiments, which focus on sports activities on planes, various methods are compared in both camera and world coordinate systems , as shown in Tab. 3. For the RGB modality, TRACE, SLAHMR, WHAM, and GVHMR exhibit relatively higher errors in metrics such as MPJPE, PA-MPJPE, and PVE. In contrast, for the RGB+LiDAR modality, the integration of LiDAR data results in reduced error rates. Specifically, ClimbingCap achieves the lowest errors, with an MPJPE of 49.30, PA-MPJPE of 38.23, and PVE of 57.59 in the camera coordinate system, and a PCK0.3 of 0.98. Similarly, in the world coordinate system, ClimbingCap records the smallest WA-MPJPE (276.59), W-MPJPE (488.84), RTE (7.10), Jitter (10.65), and T-Error (23.90). It is important to note that none of the methods were trained on the RELI11D dataset, thereby demonstrating the generalization capability of ClimbingCap for planar motion scenarios.

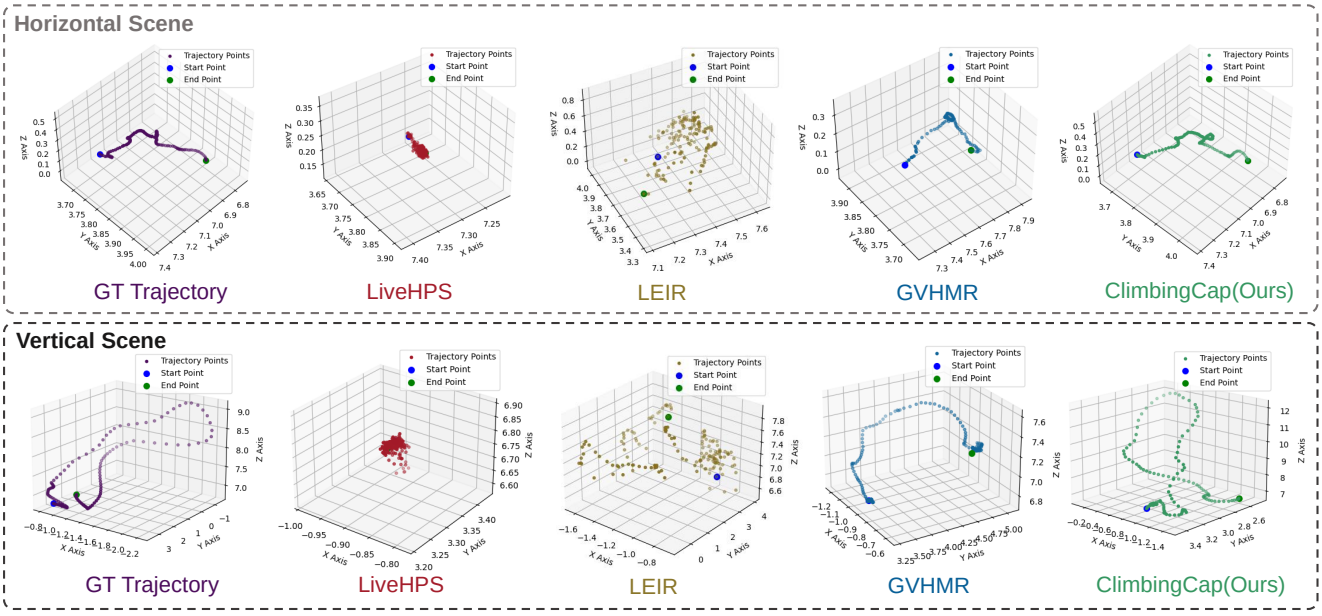

Figure 2. Global Trajectory Prediction Comparison Experiment for the AscendMotion Dataset.

Modality	Method	Camera Coordinate					World Coordinate				
		ACCEL↓	MPJPE↓	PA-MPJPE↓	PVE↓	PCK0.3↑	WA-MPJPE↓	W-MPJPE↓	RTE↓	Jitter↓	T-Error↓
RGB	TRACE [14]	-	705.25	62.49	797.86	0.08	396.75	910.01	-	-	54.64
	SLAHMR [20]	22.67	280.50	186.71	318.04	0.43	435.80	2989.62	78.47	20.88	26.15
	WHAM [13]	19.11	57.76	42.90	66.86	0.96	375.56	871.30	11.85	24.81	65.36
	GVHMR [12]	13.39	63.22	50.19	77.20	0.93	371.99	765.33	11.41	16.59	36.16
RGB+LiDAR	ImmFusion [1]	30.90	196.96	93.12	261.44	0.65	360.90	586.87	22.15	37.60	32.79
	FusionPose [2]	22.97	91.21	55.40	83.90	0.82	375.45	542.92	12.14	19.57	29.94
	ClimbingCap	**12.18**	**49.30**	**38.23**	**57.59**	**0.98**	**276.59**	**488.84**	**7.10**	**10.65**	**23.90**

Table 3. **Quantitative Comparison on the planar motion dataset RELI11D [19].** Our method demonstrates superior performance in both camera and world coordinates.

Metrix	Train / Test	AscendMotion	CIMI4D	AscendMotion+CIMI4D
ACCEL	AscendMotion	25.65	212.21	27.11
	CIMI4D	174.19	48.22	56.06
MPJPE	AscendMotion	92.42	364.26	94.17
	CIMI4D	247.18	101.80	100.08
PAMPJPE	AscendMotion	83.29	193.53	80.57
	CIMI4D	186.40	73.48	68.09
PVE	AscendMotion	115.37	404.98	116.50
	CIMI4D	294.95	120.86	119.88
PCK0.3	AscendMotion	0.82	0.25	0.82
	CIMI4D	0.45	0.79	0.85

Table 4. Cross-dataset evaluation using LiveHPS [11].

C. Cross-Dataset Evaluation

To evaluate the quality of AscendMotion, we conduct cross-dataset evaluations using the LiveHPS method [11], a state-of-the-art approach in Human Mesh Recovery (HMR) based on LiDAR point clouds. LiveHPS was trained on three datasets: CIMI4D [18], AscendMotion, and a combination of both (CIMI4D+AscendMotion). Evaluation was performed on the test sets of CIMI4D and AscendMotion, respectively, and the results are summarized in Tab. 4.

As depicted in Tab. 4, the cross-dataset evaluation revealed several insights. When LiveHPS is trained on one dataset, its performance significantly drops when tested on the other dataset. Specifically, training on CIMI4D and testing on AscendMotion results in worse performance compared to the reverse. When LiveHPS is trained on the combined dataset (CIMI4D+AscendMotion), its performance on both datasets is not always the best. This is due to the fact that the climbing motions in the two datasets are different. All the volunteers of the AscendMotion dataset are skilled climbers whose climbing motions are challenging. However, most of the volunteers of the CIMI4D dataset are casual climbers whose climbing motions are casual. A domain gap exists among these two dataset. This indicates that the AscendMotion dataset is a good addition to today's climbing motion datasets.

D. Method: ClimbingCap Details

Capturing climbing motion is challenging, as it involves poses with extreme limb extension and full-body exertion in camera coordinate. Moreover, it requires precise alignment with the rock wall in the world coordinate system as climbers ascend. The ClimbingCap pipeline consists of three parts: separate coordinate decoding, post-processing, and semi-supervised training.

An overview of the proposed pipeline is shown in the main text Fig.4. The separate coordinate decoding and post-processing parts take into account the unique challenges posted by climbing motion, which involves complex off-ground dynamics and interactions with scenes. The semi-supervised training part makes use of large-scale unlabeled climbing motion data to better learn an HMR model.

Notations. Our method utilizes several key notations to describe human pose and trajectory information. The input of the sequence consists of R_i^c and P_i^w, which respectively represent the video sequence input in the camera coordinate system c and the point cloud sequence input in the global coordinate system w, where i denotes a frame. We define the output as: the local body pose $\{\theta_i \in \mathbb{R}^{23\times3}\}_{t=0}^{T}$ and shape coefficient $\beta \in \mathbb{R}^{10}$ of the SMPL model, which capture the detailed configuration of the human body; the orientation from SMPL space to camera space, including the orientation $\{\Gamma_i^c \in \mathbb{R}^3\}_{i=0}^{T}$; the trajectory in world space, including the translation $\{\tau_i^w \in \mathbb{R}^3\}_{i=0}^{T}$, aligned with the global reference frame.

D.1. Coordinate Consistency

As can be seen from the Related Work in the main text Sec.2, most of the current global HMR methods do not have a good definition of the correct world coordinate system due to the lack of 3D point cloud modality. The dataset AscendMotion proposed in the main text Sec.3 explicitly distinguishes between these two coordinate systems through the extrinsic matrix Ω_{w2c}. For ClimbingCap, we hope that the global LiDAR point cloud information can provide more implicit information to the pixels in the camera coordinate system without affecting the estimation in the global coordinate system. We represent the global human trajectory in the world coordinate system as Γ_t^w, and obtain the human trajectory in the camera coordinate system through the extrinsic matrix Ω_{w2c}. Similarly, the human joints Υ_t^c in the camera coordinate system can also be aligned to the world coordinate system through Ω_{w2c}^{-1}. The subscript $_c$ represents the camera coordinate system, the subscript $_w$ represents the world coordinate system, and the subscript $_{w2c}$ represents the matrix from the world coordinate system to the camera coordinate system.

Given the inherent ambiguity in defining the world coordinate system, we first recover the human pose in the camera coordinate system for each frame, and then convert these poses into a consistent global trajectory. This approach takes into account the unique challenges posed by climbing motion, which involves complex off-the-ground dynamics and interactions with non-planar surfaces. Climbing motions require precise alignment with gravity as climbers adapt to available holds, often requiring extreme limb extension and full-body exertion. Our approach focuses on maintaining consistency with gravity while handling these unique motions, which are often underrepresented in existing datasets. For global translations, we predict the displacement of the body in the SMPL coordinate system from time i to $i+1$ and then transform it to the world reference frame. This process ensures that the trajectory accurately reflects the climber's movements in a variety of climbing scenarios.

The separate coordinate decoding (SCD) stage extracts features from the RGB imagery and the LiDAR point clouds, predicts the poses in camera coordinates and the positions in global coordinates.

D.2. Separate Coordinate Decoding Details

Input and Feature Extraction. The overall network structure is illustrated in the main text Fig.4. The input includes RGB images and point cloud data. First, the point cloud data is transformed from the world coordinate system to the camera coordinate system via an extrinsic matrix, represented as $\mathcal{P}_{\text{c}} = \Omega_{w2c} \cdot \mathcal{P}_{\text{w}}$. Subsequently, the RGB images and transformed point cloud data are passed through feature extraction modules, *RGB Extract* and *PC Extract*, to obtain visual and geometric features. We build the feature extraction modules based on ViT [4] and PointNet++ [10]. These features are then fed into the following two decoder modules, which regress the SMPL parameters and global motion parameters of the human body, respectively.

Camera Coordinate Decoder. This module decodes the SMPL parameters in the camera coordinate system. The RGB and point cloud features serve as inputs to the *Camera Coordinate Decoder* (denoted as $\mathcal{T}_{\text{Decoder}}$), which processes the inputs with contextual information $\mathbf{f}_{\text{backbone}}$, generating an output token $\mathbf{t}_{\text{out}}$. This output token is then used to iteratively optimize the SMPL parameters, including the pose θ, shape β, and camera translation Δc. The iterative decoding approach allows the model to gradually approximate the true pose and shape in the camera coordinate system. In each iteration, the decoder updates the current SMPL parameters θ_i, β_i, and Δc_t as follows:

$$\mathbf{t}_{\text{out}} = \mathcal{T}_{\text{Decoder}}(\mathbf{t}, \mathbf{f}_{\text{backbone}}), \tag{1}$$

where $\theta_{i+1} = \Phi_\theta \cdot \mathbf{t}_{\text{out}} + \theta_i$, $\beta_{i+1} = \Phi_\beta \cdot \mathbf{t}_{\text{out}} + \beta_i$, and $\Delta c_{i+1} = \Phi_c \cdot \mathbf{t}_{\text{out}} + \Delta c_i$ are the update equations, with Φ_θ, Φ_β, and Φ_c representing the respective weight matrices for

each parameter. Here, the input token $\mathbf{t}$ can include initialized pose, shape, and camera parameters as needed.

Global Coordinate Decoder. To fully capture the human motion trajectory in the world coordinate, we design the *Global Translation Decoder* to predict the global translation parameters Γ^{trans} of the human body. In this module, the decoder processes the features $\mathbf{f}_{\text{backbone}}$ as contextual input, iteratively updating the global translation parameters. The update formula in each iteration is given by:

$$\Gamma^{\text{trans}}_{i+1} = \Psi \cdot \mathbf{t}_{\text{out}} + \Gamma^{\text{trans}}_{i}, \tag{2}$$

where $\Gamma^{\text{trans}}_{i}$ represents the global translation parameters at time step i, and Ψ is the weight matrix for the update. This decoding process enables the model to capture a complete motion trajectory in the global coordinate system.

Loss. The total loss function not only includes the 3D keypoint loss $\mathcal{L}_{kp3d}$ and 2D keypoint loss $\mathcal{L}_{kp2d}$ but also incorporates the SMPL parameter loss $\mathcal{L}^{smpl}$ and the global trajectory loss $\mathcal{L}_{traj}$. Specifically, the 3D keypoint loss $\mathcal{L}_{kp3d}$ measures the 3D error of the predicted keypoints, while the 2D keypoint loss $\mathcal{L}_{kp2d}$ measures the 2D projection error. The orientation of the human body is strongly correlated with pose, so in the SMPL parameter loss $\mathcal{L}^{smpl}$, we concatenate the orientation + pose together to form the parameters θ and the shape parameters β, which jointly supervise the decoding. Finally, the global trajectory loss $\mathcal{L}_{traj}$ constrains the translation parameters to within close distance of ground truth positions. The specific losses are described below.

The 3D keypoint loss $\mathcal{L}_{kp3d}$ minimizes the Euclidean distance between the predicted 3D keypoints $\mathbf{k}_i^{3d,\text{pred}}$ and ground truth $\mathbf{k}_i^{3d,\text{gt}}$, calculated over all frames i as:

$$\mathcal{L}_{kp3d} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{k}_i^{3d,\text{pred}} - \mathbf{k}_i^{3d,\text{gt}}\right\|_2^2, \tag{3}$$

where N is the total number of frames. To remove global position offsets, both predicted and ground-truth 3D keypoints are pelvis-aligned before computing the loss. Similarly, the 2D keypoint loss $\mathcal{L}_{kp2d}$ supervises the projection of keypoints into the image plane, defined as:

$$\mathcal{L}_{kp2d} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{k}_i^{2d,\text{pred}} - \mathbf{k}_i^{2d,\text{gt}}\right\|_2^2, \tag{4}$$

where $\mathbf{k}_i^{2d,\text{pred}}$ and $\mathbf{k}_i^{2d,\text{gt}}$ represent the predicted and ground-truth 2D keypoints for frame i, respectively. The SMPL parameter losses $\mathcal{L}_{\theta}^{smpl}$ and $\mathcal{L}_{\beta}^{smpl}$ supervise the predicted pose parameters θ_i^{pred} and shape parameters β_i^{pred} by minimizing their squared errors with respect to ground truth, formulated as:

$$\mathcal{L}_{\theta}^{smpl} = \frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i^{\text{pred}} - \theta_i^{\text{gt}}\right\|_2^2, \tag{5}$$

$$\mathcal{L}_{\beta}^{smpl} = \frac{1}{N}\sum_{i=1}^{N}\left\|\beta_i^{\text{pred}} - \beta_i^{\text{gt}}\right\|_2^2. \tag{6}$$

In addition, the global trajectory loss $\mathcal{L}_{traj}$ ensures that the predicted global translations $\Gamma^{\text{pred}}_{\text{trans},i}$ remain close to the ground truth $\Gamma^{\text{gt}}_{\text{trans},i}$, defined as:

$$\mathcal{L}_{traj} = \frac{1}{N}\sum_{i=1}^{N}\left\|\Gamma^{\text{pred}}_{\text{trans},i} - \Gamma^{\text{gt}}_{\text{trans},i}\right\|_2^2. \tag{7}$$

The total loss is then formulated as:

$$\begin{aligned}\mathcal{L} = w_{3D}\cdot\mathcal{L}_{kp3d} + w_{2D}\cdot\mathcal{L}_{kp2d} + w_{\theta}\cdot\mathcal{L}_{\theta}^{smpl} \\ + w_{\beta}\cdot\mathcal{L}_{\beta}^{smpl} + w_{traj}\cdot\mathcal{L}_{traj},\end{aligned} \tag{8}$$

where the weights $\{w_{3D}, w_{2D}, w_{\theta}, w_{\beta}, w_{traj}\}$ are hyperparameters that balance the contributions of each loss term. These components collectively drive the optimization process by supervising the keypoints, pose, shape, and global translations to ensure accurate motion estimation.

D.3. Post-processing Details

Researches [12, 13, 20] have shown that a post-processing stage can be used to improve the output motion recovery results. Following these approaches, we employ a post-processing stage to optimize the output pose from SCD stage(the main text Sec.4.1). One distinct advantage of ClimbingCap is that the output results from the pose decoding stage can be rigidly transformed between the camera and world coordinate systems. Thanks to the LiDAR modality, the point cloud contains 3D information in the world coordinate system. The poses obtained from the SCD stage are converted from the camera coordinate system to the world coordinate system through the inverse extrinsic matrix Ω_{w2c}^{-1}.

The post-processing stage consists of three Losses: $\mathcal{L}_{LWD}$, $\mathcal{L}_{SDS}$, and $\mathcal{L}_{VLR}$. Specifically, $\mathcal{L}_{LWD}$ assigns different weights to the vertices of different parts of the climbing human SMPL model (e.g., torso, arms, hands, feet) and minimizes the weighted Chamfer Distance (CD) between the SMPL vertices and the ground-truth point cloud, formulated as:

$$\begin{aligned}\mathcal{L}_{LWD} = \frac{1}{N}\sum_{i=1}^{N}\Big(w_{\text{main}}\cdot\text{CD}(\mathbf{v}_i^{\text{main}}, \mathbf{p}_i) \\ + w_{\text{arms}}\cdot\text{CD}(\mathbf{v}_i^{\text{arms}}, \mathbf{p}_i) \\ + w_{\text{ends}}\cdot\text{CD}(\mathbf{v}_i^{\text{ends}}, \mathbf{p}_i)\Big),\end{aligned} \tag{9}$$

where $\mathrm{CD}(\mathbf{v}, \mathbf{p}) = \frac{1}{|\mathbf{v}|}\sum_{\mathbf{v}\in V}\min_{\mathbf{p}\in P}\|\mathbf{v}-\mathbf{p}\|_2^2$, and $\mathbf{v}_i^{\text{main}}, \mathbf{v}_i^{\text{arms}}, \mathbf{v}_i^{\text{ends}}$ represent the visible vertex groups for torso, arms, and limb ends (feet and hands), respectively. $w_{\text{main}}, w_{\text{arms}}, w_{\text{ends}}$ are weights assigned to these groups, reflecting their relative importance in the optimization process. For example, larger weights can be assigned to critical regions such as the torso (w_{main}) to ensure better alignment of the body with the ground-truth point cloud, while smaller weights may be used for limb ends (w_{ends}) due to their higher variability.

The second loss, $\mathcal{L}_{SDS}$, ensures smooth motion dynamics by penalizing unnatural joint accelerations on the climbing wall. For a sequence of joints $\mathbf{j}_i \in \mathbb{R}^{24\times 3}$ (where 24 is the number of joints), the loss is defined as:

$$\mathcal{L}_{SDS} = \frac{1}{N-2}\sum_{i=2}^{N-1}\max(0, \zeta - \|\mathbf{j}_{i-1} - 2\mathbf{j}_i + \mathbf{j}_{i+1}\|_2), \tag{10}$$

where ζ is the acceleration threshold, and N is the total number of frames.

The third loss, $\mathcal{L}_{VLR}$, refines the temporal smoothness of limb rotations by penalizing large rotational changes between consecutive frames. Given the axis-angle representation of pose parameters $\theta_i \in \mathbb{R}^{23\times 3}$ for 23 joints, it is first converted into a 6D representation $\boldsymbol{\phi}_i \in \mathbb{R}^{23\times 6}$ using:

$$\boldsymbol{\phi}_i = \text{Rot6D}(\theta_i), \tag{11}$$

where $\text{Rot6D}(\cdot)$ represents the conversion from axis-angle to 6D rotation representation by extracting the first two columns of the corresponding rotation matrix. The loss is then computed as:

$$\mathcal{L}_{VLR} = \frac{1}{N-1}\sum_{i=1}^{N-1}\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_{i+1}\|_1 . \tag{12}$$

Finally, the combined loss for the post-processing stage is defined as:

$$\mathcal{L}_{\text{post}} = w_{\text{LWD}}\cdot\mathcal{L}_{LWD} + w_{\text{SDS}}\cdot\mathcal{L}_{SDS} + w_{\text{VLR}}\cdot\mathcal{L}_{VLR}, \tag{13}$$

where $w_{\text{LWD}}, w_{\text{SDS}}, w_{\text{VLR}}$ are hyperparameters controlling the contribution of each loss term. The global pose optimization is performed using the Adam [8] optimizer, enhancing pose alignment, motion smoothness, and temporal consistency.

D.4. Semi-supervised Framework Details

Compared to ground-level motion datasets [5, 9, 15], the scale of labeled climbing motion data is relatively small. Solely relying on labeled climbing motion data may be insufficient to train robust models. Unlike labeled climbing motion data, collecting unlabeled climbing motion data is more cost-effective. The AscendMotion dataset contains a substantial amount of unlabeled data, derived from real climbers performing highly challenging climbing motions. These data can be used to further improve HMR models.

Research in the object detection community [16, 17, 21] has demonstrated that using a teacher-student semi-supervised training framework can effectively enhance the performance of object detection models. Drawing a parallel to AscendMotion, we hypothesize that adopting such a semi-supervised framework for HMR can effectively leverage the unlabeled data in the dataset. In this work, we define the models trained after the SCD and post-processing stages as teacher models (represented by the green box in the main text Fig.4).

During training, by inputting unlabeled dual-modal data into the teacher network, we obtain accurate outputs in the global coordinate system, including $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, $\boldsymbol{\delta}_w$, and $\boldsymbol{\Gamma}_w$. These globally post-processed pseudo-labels are referred to as high-confidence pseudo-labels. In the next step, these pseudo-labels, together with the input video, are used to train the student network.

The student model (represented by the red box in the main text Fig.4) clones the parameters of the teacher model. During semi-supervised training, the teacher model estimates pose labels from unlabeled motion data, which are then used as pseudo-labels to further train the student model. As shown in the experimental section, the performance of HMR can be further improved by utilizing the semi-supervised training framework.

E. Dataset: AscendMotion Details

AscendMotion is a multimodal dataset designed specifically for capturing the intricate movements of climbers in various climbing scenarios. To create this dataset, we recruited 22 experienced climbers to perform various climbing routes across 12 different climbing walls, including both indoor and outdoor settings. The dataset features high-resolution annotations of human poses and trajectories, ensuring a comprehensive understanding of climbing dynamics. The motions collected in AscendMotion are from skilled climbers. These participants consent to the use of their recorded data for scientific purposes. Rock climbing is highly challenging for unskilled climbers, who can fall easily from the rock, whereas skilled climbers can climb on the rock with faster speed and longer duration, and grasp more rock holds than amateur players.

As illustrated in the main text Fig.3, AscendMotion provides rich annotations, including 3D human body key points, limb trajectories, and detailed scene reconstruction, which are crucial for studying human motion in climbing scenes. The main text Table.1 highlights the unique aspects of AscendMotion in comparison to other publicly available human motion datasets, showcasing its focus on climbing activities.

E.1. Annotation Pipeline

To ensure the quality of the dataset, we implemented an automatic annotation pipeline that utilizes the motion characteristics inherent to climbing, such as spatial consistency and contact dynamics, to refine pose and trajectory annotations frame by frame. We further enhance annotation quality through manual verification and correction processes, as described in the main text Fig.3. Here are some more details.

Scene Reconstruction. Unlike actions that only interact with flat ground or stairs, rock climbing motions are complex and anti-gravity. We believe that correct and high-precision scene reconstruction is important for the annotation method. Accurately reconstructed scenes play a vital role in restoring the interaction between the human body and the scene. Fig. 1 shows the accurately reconstructed scene in AscendMotion. The upper and lower modules show part of the horizontal scene and the vertical scene respectively. We place a 5'7" human model in the wall to compare the size of scene. All scenes in AscendMotion are collected from real rock climbing venues. The gym manager agrees to use the data for scientific research. For each sequence collection, we scan the scene once to ensure the correctness of the scene in the sequence.

First, we use the static scanning device Trimble X7 to perform multi-site scans of the collected rock walls. Each scene includes 10 million centimeter-precision color point clouds. Then, we crop the rock walls used in the current sequence to ensure that each scene has a million-level refined point cloud. Finally, we perform Poisson Reconstruction [6, 7] on all scenes to obtain geometric patch models that adapt to physical contact.

Time Synchronization. The time among RGB camera and LiDAR are synchronized via Precision Time Protocol (PTP). We employ CollShark Auto 66 unit as the master clock, and its sends PTP slaves clocks to the RGB camera and LiDAR. The time of IMU MoCap is post-synchronized with the LiDAR and RGB through anchor frames. As shown in the main text Fig.2, we use the Jetson AGX development board as the basis, use Switches and Master Clocks, and design accurate multimodal trigger signals to synchronize the LiDAR and RGB Camera in hardware. The IMU human motion capture system post-synchronize with the LiDAR through the anchor frame.

Calibration. First, the LiDAR point cloud are registered with high-precision scanned-scenes. Next, the coordinate of LiDAR is treated as the world coordinate. The IMU measurements are transformed into the world coordinate through a calibration matrix. Finally, we perform frame-level calibration among RGB, LiDAR and IMU. Next, for each frame, we isolate human body point clouds and derive human poses based on them. The movement of the human body in the world coordinate system $\{W\}$ is represented as $M^W = (T^W, \theta^W, \beta)$. The IMU provides T^I and θ^I in $M^I = (T^I, \theta^I, \beta)$, and the initial pose $\theta^W = R_{WI}\theta^I$ is computed with a rough calibration matrix R_{WI} from the IMU to the world coordinates. Given the limited translation accuracy of the IMUs, we use the hip center in the point cloud as T^W. Finally, we complete frame-level temporal synchronization and spatial calibration across all modalities.

E.1.1 Multi-stage Global Optimization Detail

AscendMotion use the translation T and pose θ provided by the IMU MoCap as the initialization of annotation labels, and performs multi-stage global optimization.

To achieve accurate and natural human motion data consistent with the scene, we apply two loss functions: the Global Refit Loss $\mathcal{L}_{\text{GR}}$ and the Scene Touch Loss $\mathcal{L}_{\text{ST}}$. These losses optimize the global pose and trajectory to better align with scene constraints.

Global Refit Loss $\mathcal{L}_{\text{GR}}$: This loss, based on the Chamfer distance, calculates the geometric discrepancy between the SMPL model vertices and the human body point cloud, using distance threshold filtering and different weighting for body parts to ensure accurate matching. For each frame, given the set of visible SMPL vertices $V \subset \mathbb{R}^{6890\times 3}$ and the point cloud $P \subset \mathbb{R}^{N\times 3}$, we define the loss as follows:

$$\mathcal{L}_{\text{GR}} = \sum_{v_i \in V} \sum_{p_j \in P} \big[\mathbb{I}(\|v_i - p_j\|^2 < d^2_{\text{trunk}}) \cdot f(\|v_i - p_j\|^2) + \mathbb{I}(\|p_j - v_i\|^2 < d^2_{\text{close}}) \cdot g(\|p_j - v_i\|^2)\big], \tag{14}$$

where d_{trunk} denotes the maximum distance threshold used for filtering, d_{close} represents the minimum distance threshold for close matching, $f(x) = \frac{0.3x^2}{x^2+0.02}$ is a custom distance filtering function applied for smoothing, and $g(x)$ is the square root function applied for close-matching points.

The indicator function $\mathbb{I}(\|v_i - p_j\|^2 < d^2_{\text{trunk}})$ returns 1 if and only if $\|v_i - p_j\|^2 < d^2_{\text{trunk}}$, indicating that the distance between v_i and p_j is within the threshold d_{trunk}; otherwise, it returns 0. Similarly, $\mathbb{I}(\|p_j - v_i\|^2 < d^2_{\text{close}})$ returns 1 if $\|p_j - v_i\|^2 < d^2_{\text{close}}$, indicating that the distance between p_j and v_i is within the threshold d_{close}; otherwise, it returns 0. These indicator functions ensure that only vertex-point pairs meeting the specified distance criteria contribute to the loss. Different weights are assigned to major body parts, such as the torso and limbs, to balance their influence in the overall loss. This variation of the Chamfer distance ensures precise global alignment between the SMPL model and the point cloud, enhancing pose-fitting accuracy.

Scene Touch Loss $\mathcal{L}_{\text{ST}}$: This loss measures the penetration depth between SMPL vertices and the scene mesh, preventing unrealistic overlap between the model and the environment. Given the scene mesh vertices Q and their corresponding normal vectors N, we compute the nearest distance and penetration depth for each SMPL vertex $v_i \in V$.. If the penetration depth, $\eta(v_i)$, is negative, it is included in the loss:

$$\mathcal{L}_{\text{ST}} = -\sum_{v_i \in V} \mathbb{I}(\eta(v_i) < 0) \cdot \eta(v_i) \tag{15}$$

where $\eta(v_i) = (v_i - q_j) \cdot n_j$, $q_j \in Q$ is the closest mesh vertex to v_i, and n_j is the corresponding normal vector. When $\eta(v_i) < 0$, it indicates penetration of vertex v_i into the scene mesh. This penetration depth contributes to the scene touch loss, effectively reducing unrealistic intersections.

The indicator function $\mathbb{I}(\eta(v_i) < 0)$ returns 1 if $\eta(v_i) < 0$, indicating that the vertex v_i has penetrated the scene mesh; otherwise, it returns 0, meaning there is no contribution to the loss from that point. When the penetration depth $\eta(v_i)$ is negative, it indicates that the SMPL vertex v_i has intersected the scene mesh, and this depth contributes to the scene touch loss, effectively reducing unrealistic intersections.

References

[1] Anjun Chen, Xiangyu Wang, Kun Shi, Shaohao Zhu, Bin Fang, Yingfeng Chen, Jiming Chen, Yuchi Huo, and Qi Ye. Immfusion: Robust mmwave-rgb fusion for 3d human body reconstruction in all weather conditions. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2752–2758. IEEE, 2023. 3

[2] Peishan Cong, Yiteng Xu, Yiming Ren, Juze Zhang, Lan Xu, Jingya Wang, Jingyi Yu, and Yuexin Ma. Weakly supervised 3d multi-person pose estimation for large-scale scenes based on monocular camera and single lidar. In *AAAI*, pages 461–469. AAAI Press, 2023. 3

[3] Yudi Dai, Yitai Lin, XiPing Lin, Chenglu Wen, Lan Xu, Hongwei Yi, Siqi Shen, Yuexin Ma, and Cheng Wang. Sloper4d: A scene-aware dataset for global 4d human pose estimation in urban environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 682–692, 2023. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 4

[5] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1325–1339, 2014. 6

[6] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 7

[7] Misha Kazhdan and Hugues Hoppe. An adaptive multi-grid solver for applications in computer graphics. In *Computer graphics forum*, volume 38, pages 138–150. Wiley Online Library, 2019. 7

[8] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic gradient descent. In *ICLR: international conference on learning representations*, pages 1–15. ICLR US., 2015. 6

[9] 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. 6

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 4

[11] Yiming Ren, Xiao Han, Chengfeng Zhao, Jingya Wang, Lan Xu, Jingyi Yu, and Yuexin Ma. Livehps: Lidar-based scene-level human pose and shape estimation in free environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1291, 2024. 1, 2, 3

[12] 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 Conference Proceedings*, 2024. 1, 2, 3, 5

[13] Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2070–2080, 2024. 1, 3, 5

[14] Yu Sun, Qian Bao, Wu Liu, Tao Mei, and Michael J Black. Trace: 5d temporal regression of avatars with dynamic cameras in 3d environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8866, 2023. 3

[15] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 6

[16] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14615–14624, 2021. 6

[17] Qiming Xia, Wei Ye, Hai Wu, Shijia Zhao, Leyuan Xing, Xun Huang, Jinhao Deng, Xin Li, Chenglu Wen, and Cheng Wang. Hinted: Hard instance enhanced detector with mixed-density feature fusion for sparsely-supervised 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15321–15330, 2024. 6

[18] Ming Yan, Xin Wang, Yudi Dai, Siqi Shen, Chenglu Wen, Lan Xu, Yuexin Ma, and Cheng Wang. Cimi4d: A large multimodal climbing motion dataset under human-scene interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12977–12988, 2023. 3

[19] Ming Yan, Yan Zhang, Shuqiang Cai, Shuqi Fan, Xincheng Lin, Yudi Dai, Siqi Shen, Chenglu Wen, Lan Xu, Yuexin Ma, et al. Reli11d: A comprehensive multimodal human motion dataset and method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2250–2262, 2024. 1, 2, 3

[20] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 3, 5

[21] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14494–14503, 2021. 6