OpenMoCap: Rethinking Optical Motion Capture under Real-world Occlusion

Chen Qian
Tsinghua University
Beijing, China
chen.cronus.qian@gmail.com

Danyang Li*
Tsinghua University
Beijing, China
lidanyang1919@gmail.com

Xinran Yu
Tsinghua University
Beijing, China
yuxinran0929@126.com

Zheng Yang
Tsinghua University
Beijing, China
hmilyyz@gmail.com

Qiang Ma
Tsinghua University
Beijing, China
tsinghuamq@gmail.com

Abstract

Optical motion capture is a foundational technology driving advancements in cutting-edge fields such as virtual reality and film production. However, system performance suffers severely under large-scale marker occlusions common in real-world applications. An in-depth analysis identifies two primary limitations of current models: (i) the lack of training datasets accurately reflecting realistic marker occlusion patterns, and (ii) the absence of training strategies designed to capture long-range dependencies among markers. To tackle these challenges, we introduce the CMU-Occlu dataset, which incorporates ray tracing techniques to realistically simulate practical marker occlusion patterns. Furthermore, we propose OpenMoCap, a novel motion-solving model designed specifically for robust motion capture in environments with significant occlusions. Leveraging a marker-joint chain inference mechanism, OpenMoCap enables simultaneous optimization and construction of deep constraints between markers and joints. Extensive comparative experiments demonstrate that OpenMoCap consistently outperforms competing methods across diverse scenarios, while the CMU-Occlu dataset opens the door for future studies in robust motion solving. The proposed OpenMoCap is integrated into the MoSen MoCap system for practical deployment. The code is released at: https://github.com/qianchen214/OpenMoCap.

CCS Concepts

- **Computing methodologies → Motion capture**.

Keywords

Motion Capture; Motion Processing; Optical Motion Capture; MoCap Solving

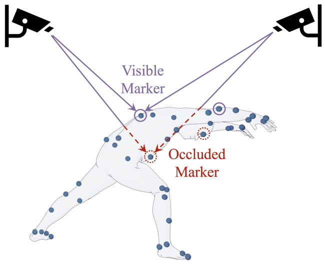


Figure 1: Marker occlusion. The marker placed on the back is captured by two infrared cameras, enabling accurate localization via triangulation. Meanwhile, the marker on the abdomen is occluded in either view.

ACM Reference Format:

Chen Qian, Danyang Li, Xinran Yu, Zheng Yang, and Qiang Ma. 2025. OpenMoCap: Rethinking Optical Motion Capture under Real-world Occlusion. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3746027.3754932

1 Introduction

Optical motion capture (MoCap) technology plays a critical role in digitally recording and reconstructing human motion patterns with high precision. This technology has become indispensable across diverse fields such as film production, game development, and embodied intelligence [5, 17, 30–32]. During the MoCap process, multiple infrared cameras synchronously emit infrared light at specific wavelengths (e.g., 850 nm) and capture reflected signals from markers attached to the human body. These signals enable precise positional reconstruction for markers through triangulation methods [12]. The subsequent stage, known as MoCap data solving [1, 15, 19, 26], involves deriving skeletal movements from noisy marker point clouds, thus facilitating reliable motion analysis.

Deep learning-based approaches to MoCap data solving [6, 15, 19, 26, 27] are emerging as a focal point of research in this field. MoCap-Solver [6] decomposes the task into three components—template skeletons, marker layout, and motion, and jointly decodes their latent representations. LocalMoCap [26] utilizes spatially and temporally adjacent markers for mutual completion and designs graph neural networks to reconstruct human skeletons. Building upon LocalMoCap, RoMo [27] further reduces the complexity of motion solving by decomposing joint rotations into directional components, improving both efficiency and accuracy.

[*] Danyang Li is the corresponding author.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3754932

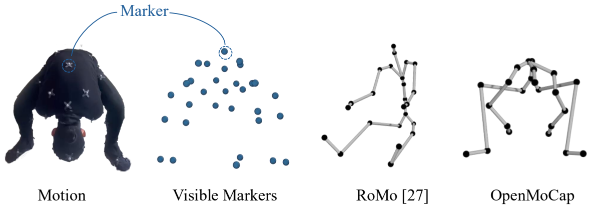


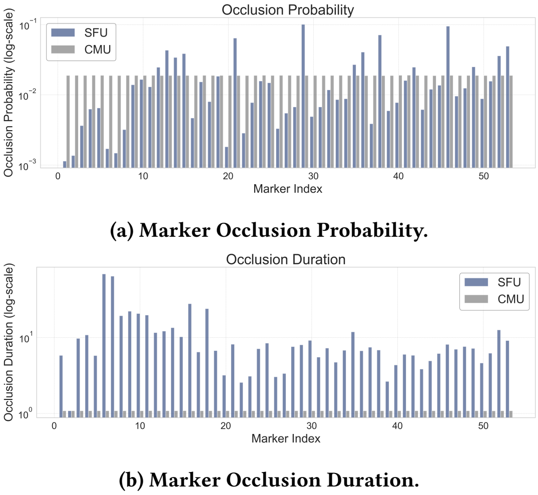
Figure 2: Performance of MoCap in real occlusion scenario. A MoCap actor performing a squatting action with arms extended backward. Significant occlusions occur on markers placed on the abdomen, chest, and forehead. OpenMoCap provides a comparatively reliable solution.

Albeit inspiring, we observe significant performance degradation when deploying state-of-the-art (SOTA) system [27] in real-world production environments due to marker occlusion. Marker occlusion in MoCap occurs when markers are blocked by body parts, obstacles, or environmental structures, which prevents cameras from capturing reflected infrared signals and determining the spatial locations of markers. As illustrated in Fig.1, the markers on the back of the body are visible, whereas those on the abdomen and chest are occluded by the body. Existing method [27] fails to reconstruct the human skeleton due to significant occlusions affecting markers placed on regions like the chest as depicted in Fig.2.

Although existing methods exhibit impressive performance on synthetic test sets, they struggle to achieve satisfactory results in solving real MoCap data, like the SFU dataset [25] . A thorough analysis of this issue has led us to the following two conclusions:

• **Marker occlusions in real MoCap settings often exhibit high variability and long durations.** Existing datasets typically use random marker occlusions for data augmentation to boost model robustness. We have analyzed occlusion scenarios for markers in both synthetic dataset CMU [9] with random occlusions and the real MoCap dataset SFU as shown in Fig.3. Markers are indexed by their proximity relationships. The results reveal that occlusions tend to exhibit certain patterns rather than occurring at random, underscoring a significant mismatch between the occlusion modes in existing synthetic datasets and those observed in real scenarios. This discrepancy becomes problematic as model deployment heavily relies on the assumption that training and testing datasets are independently and identically distributed (i.i.d.). The divergence in data distribution due to marker occlusions leads to a deterioration in model performance, constraining its deployment in real-world settings. This gap highlights the need for more realistic, large-scale training datasets that better reflect real-world marker occlusion patterns.

• **Existing models lack mechanisms to effectively handle the real characteristics of occlusions.** To address the issue of marker occlusion, recent initiatives, including LocalMoCap [26] and RoMo [27], have introduced methods that utilize visible markers to infer the positions of spatially adjacent occluded markers. However, their underlying assumption does not hold in practice. For example, complex motions often lead to occlusions of multiple adjacent markers simultaneously. This invalid assumption ultimately leads to degraded performance of methods in real-world occlusion scenarios.



(a) Marker Occlusion Probability.

(b) Marker Occlusion Duration.

Figure 3: Distribution comparison of marker occlusions in the CMU and SFU MoCap datasets.

To overcome the limitations outlined above, this paper introduces the **CMU-Occlu dataset**, which better reflects the distribution of marker occlusions in the real world, along with **OpenMoCap**, a motion solving model designed to address authentic occlusion patterns. Specifically, our work has made targeted innovations in two key dimensions:

• **Training Dataset.** We release the CMU-Occlu, a large-scale motion capture dataset that accurately conforms to the characteristics of real marker occlusions. To mitigate the issue of distribution shift between synthetic datasets and real-world data, this study introduces ray tracing algorithms into the generation process of the MoCap dataset for the first time. By simulating various spatial arrangements of infrared cameras in a virtual environment, we have modeled the occlusions caused by obstacles or body parts. This approach significantly enhances the consistency between synthetic data and actual occlusion patterns.

• **Solving Mechanism.** This paper proposes a marker-joint chain inference mechanism aimed at accurately reconstructing the positions of both markers and joints. This mechanism incorporates joints and occluded markers as learnable parameters, employing bidirectional chain inference between markers and joints. With the joints serving as intermediaries, this approach establishes long-distance spatial constraints among markers. Furthermore, it enables simultaneous optimization of marker and joint positions, where progressively refined marker estimates contribute to the improved accuracy of joint reconstruction.

To validate the effectiveness of the proposed work, we conducted extensive experiments on both the synthetic MoCap dataset CMU-Occlu and the real MoCap dataset SFU. The experiments demonstrate that CMU-Occlu provides a consistent performance improvement for existing methods [26, 27], compared to the CMU dataset. Additionally, OpenMoCap surpasses SOTA method, with joint position and joint rotation errors reduced by more than 27%.

In summary, this paper makes following contributions.

- For the first time, a systematic analysis of marker occlusion patterns in real MoCap scenarios is conducted, revealing two fundamental reasons underlying the performance bottlenecks of current SOTA methods: (*i*) Inconsistencies between the occlusion data distributions in the training sets and the real-world test sets; (*ii*) The inability of models to effectively establish long-distance spatial constraints.
- We introduce the CMU-Occlu dataset, which incorporates realistic marker occlusion patterns and overcomes the limitations of existing optical MoCap datasets with overly simplistic and unrealistic occlusion assumptions. This dataset can serve as a benchmark for evaluation in the field of optical motion capture solving.
- We propose a robust MoCap solving model, OpenMoCap, which innovatively implements a marker-joint chain inference mechanism to enhance reconstruction capabilities in scenarios with marker occlusions. Extensive experiments demonstrate that the proposed method surpasses prior state-of-the-art techniques.
- Based on the OpenMoCap algorithm, we develop a low-cost MoCap system, MoSen, which eliminates the need for labor-intensive post-processing commonly required in mainstream commercial solutions (e.g., OptiTrack, VICON). By fundamentally transforming the workflow of MoCap repair specialists, our system significantly reduces the overall cost of MoCap and paves the way for its broader adoption.

2 Related Work

2.1 Motion Data Synthesis

The CMU Motion Capture (MoCap) dataset [9] is a widely used benchmark in human motion analysis, offering over 2000 high-quality sequences captured via a Vicon system, covering diverse actions like walking, jumping, and dancing. It has served as a foundation for developing and evaluating data-driven MoCap algorithms. SMPL [18] enables accurate motion reconstruction by parameterizing body shape and pose. AMASS [9, 20, 23, 24] unifies data from multiple MoCap datasets and refines surface representations to provide high-quality motion data.

To improve model robustness, synthetic datasets often apply frame-wise random occlusion [6, 14] or long-term occlusion of a single marker [26]. In real-world settings, however, occlusions are typically sustained and affect multiple neighboring markers, creating a distribution gap that limits model applicability in practical MoCap scenarios.

2.2 Motion Capture Solving

Recently, MoCap has become a research hotspot. UUO-Mocap [21] tackles motion capture in unstructured, unlabeled video settings by leveraging body priors and handling partial-body observations. SportsCap [7] addresses challenging sports scenarios with a framework that jointly captures 3D motion and fine-grained actions using structured priors and a multi-stream spatio-temporal GCN.

Optical motion capture, the most precise and widely adopted method, estimates body pose from marker trajectories. Traditional marker-based solving methods [1, 2, 10, 16] rely on geometric constraints and optimization algorithms under specific assumptions. Recently, deep learning has driven advances in marker tracking and data reconstruction. MoSh++ [19] estimates body pose via frame-wise parameter optimization. MoCap-Solver [6] encodes skeletal structure, marker layout, and motion separately, then jointly decodes them. LocalMoCap [26] exploits local marker dependencies, completing occluded positions from neighbors and applying GCNs for motion inference. Damo [15] improves generalization across marker layouts, while RoMo [27] addresses marker mislabeling and positional noise. Although data-driven methods are more robust, most are not explicitly designed for real-world occlusion, leading to performance degradation when assumptions about clean inputs are violated.

In other fields [3, 8], methods like MAE [13] enhance learning by masking parts of the input and training on the visible data. OpenMoCap adopts this idea to recover occluded information.

3 CMU-Occlu Dataset Synthesis

Marker occlusion is inevitable in optical motion capture. Existing synthetic datasets rely solely on random occlusion methods for data augmentation. However, this approach does not accurately reflect occlusion patterns observed in real-world motion capture scenarios, thus limiting the effectiveness and efficiency of current pre-trained models when deployed in production environments.

3.1 Preliminary: Marker Capture

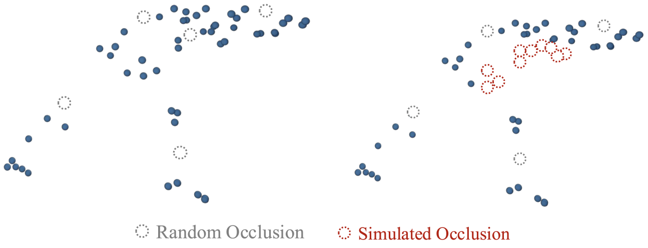
Here, we briefly describe the working principle of optical motion capture systems. In such systems, calculating the three-dimensional coordinates of markers relies on satisfying a co-visibility constraint: spatial position of one marker can be reconstructed using epipolar geometry [11] only if it is simultaneously captured by at least two infrared cameras. As illustrated in Fig.1, two stationary infrared cameras are positioned at the top, and the blue spheres attached to the body represent markers. In this example, when the subject performs a forward-bending movement:

- **Visible Marker**: The marker on the back remains visible to both cameras, satisfying the co-visibility condition, allowing its accurate 3D coordinates to be calculated through triangulation.
- **Occluded Marker**: Markers on the abdomen and chest become obscured by body, resulting in fewer than two cameras capturing their reflected signals. Consequently, the system cannot form valid observation equations for their reconstruction, and these markers are classified as occluded.

Considering the working principles of MoCap systems, marker occlusion depends on various factors, including the number of cameras, their spatial arrangement, and the complexity of movements performed. Consequently, the resulting occlusion patterns often involve extensive occlusions and prolonged occlusion periods for individual markers.

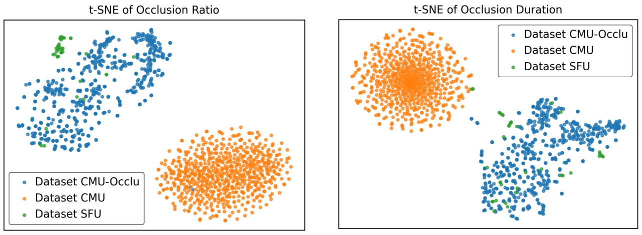
3.2 Dataset Construction

To incorporate realistic marker occlusion patterns into optical motion capture datasets, we propose an improved version of the original CMU dataset [9], termed CMU-Occlu. This dataset leverages a parallel implementation of the Möller–Trumbore algorithm



(a) CMU Dataset. (b) CMU-Occlu Dataset.

Figure 4: Comparison of marker occlusion patterns of different datasets.



(a) Marker Occlusion Ratio. (b) Marker Occlusion Duration.

Figure 5: Comparison of marker occlusion distribution characteristics across datasets.

[22], integrating real MoCap scenario configurations. As a result, CMU-Occlu effectively bridges the gap between existing synthetic datasets and real-world occlusion patterns.

Specifically, an infrared ray is defined by its origin $\mathbf{r}_0$ and direction vector $\mathbf{d}$:

$$\mathbf{R}(t) = \mathbf{r}_0 + t\mathbf{d}. \tag{1}$$

At the same time, the triangular mesh of the human body is defined by its vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, and the marker $\mathbf{p}$ located on a mesh triangle can be expressed as:

$$\mathbf{p} = (1 - \beta - \gamma)\mathbf{v}_1 + \beta\mathbf{v}_2 + \gamma\mathbf{v}_3. \tag{2}$$

To determine whether the human body occludes the infrared ray from capturing a marker, it is necessary to check whether the mesh intersects with the ray, and whether the intersection point lies closer to the camera than the marker itself.

$$\mathbf{r}_0 + t\mathbf{d} = (1 - \beta - \gamma)\mathbf{v}_1 + \beta\mathbf{v}_2 + \gamma\mathbf{v}_3, \tag{3}$$

$$\textbf{is\_occluded} = (t_{\text{closest}} < \|\mathbf{p} - \mathbf{r}_0\|), \tag{4}$$

where t_{closest} is the value of t for the nearest valid intersection point.

Through this approach, we incorporate simulated marker occlusion into the CMU-Occlu dataset. As illustrated in Fig.4, the corresponding frames of markers from both the CMU dataset and the CMU-Occlu dataset are visualized.

To facilitate training more robust models, the proposed CMU-Occlu dataset encompasses both random and simulated occlusion patterns. Specifically, the simulated occlusion patterns are generated by selecting four cameras through comparing the Kullback–Leibler (KL) divergence of occlusion distributions with the real-world SFU dataset. Additionally, the four camera layouts with the highest divergence are chosen, and oversampling [4] is employed to address the imbalance problem arising from an insufficient number of occluded MoCap frames.

As shown in Fig.5, we conduct a comparative analysis of marker occlusion distributions across the CMU, CMU-Occlu, and the real-world MoCap dataset SFU [25]. The comparison encompasses both the occlusion probability of individual markers and the duration of occlusion for each marker. Results show that the occlusion distribution of CMU-Occlu aligns more closely with that of SFU, whereas CMU exhibits a distinctly different pattern.

To ensure versatility and ease of use across various research and production environments, we will publicly release our dataset generation method as open-source, along with interfaces supporting different parameter configurations to accommodate diverse application scenarios.

4 Architecture

Given a captured point cloud of visible markers in a single frame, our goal is to accurately estimate joint positions and rotations, while simultaneously reconstructing the positions of markers that may be occluded or displaced. To address this challenge, we propose a multi-stage framework that decouples position estimation from rotation estimation.

4.1 Decoupled MoCap Architecture

Different joints play distinct roles in human motion, with waist-region joints (e.g., pelvis) being critical for determining global pose. To ensure accurate modeling, most methods assume partial visibility of key markers. For instance, to aid model convergence, training data is typically aligned to a standard pose using eight waist markers. This alignment introduces two key limitations:

- **Occlusion Sensitivity**. Alignment methods such as Singular Value Decomposition (SVD) require at least three pairs of corresponding visible points between two point clouds. Consequently, if one of key markers remain continuously occluded, models like MoCap-Solver[6] cannot function properly.
- **Irreversible Error Propagation**. Directly performing spatial alignment using partially missing critical markers [26, 27] introduces alignment errors at the preprocessing stage. These initial errors propagate iteratively through the global skeletal model, ultimately causing substantial distortions in the reconstructed motion.

To address critical data loss caused by marker occlusions, we propose a decoupled MoCap solving architecture that postpones the spatial alignment process. This architectural design stems from a fundamental insight: solving for joint and marker positions constitutes a linear problem that does not heavily depend on spatial alignment outcomes. Conversely, joint rotation estimation is inherently more challenging, characterized as a nonlinear problem that benefits significantly from spatial alignment to facilitate training convergence.

Following this design principle, the overall architecture of our approach, OpenMoCap, is illustrated in Fig.6. The architecture consists of two primary components: a Position Solver and a Rotation Solver. The Position Solver takes the raw marker positions as input to compute joint positions. It also reconstructs positions of

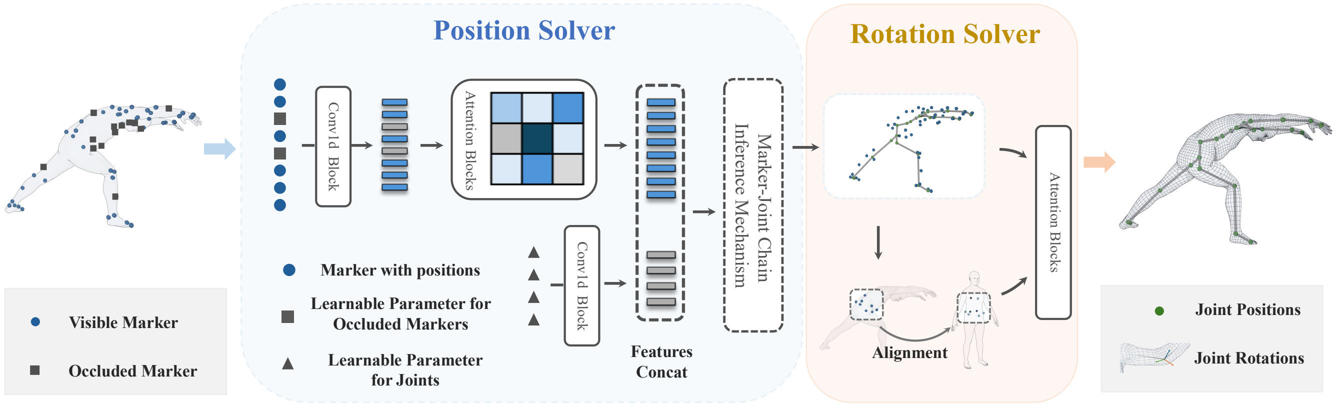


Figure 6: Overview of the proposed multi-stage framework, OpenMoCap. The framework takes raw markers with occlusions as input. A Position Solver first estimates the positions of all markers and joints through the Marker-Joint Chain Inference Mechanism. These positions are then used to align the input for a Rotation Solver, which predicts joint rotations using a stacked attention-based architecture.

all markers, filling in occluded markers and correcting positions of displaced markers. Subsequently, critical reference markers are spatially aligned with their corresponding markers in a T-pose configuration to perform a transformation. The whole aligned markers are then used in the Rotation Solver, which computes the joint rotations of the human body.

4.1.1 Position Solver. To handle varying levels of marker occlusion, we represent occluded markers as shared learnable parameters. This allows the network to learn their feature distributions, avoiding distortion from fixed placeholders like zeros. Convolutional and attention modules extract features from visible markers, which are then propagated to occluded ones.

Joint position solving is formulated as a generative task, where the input includes extracted features and N shared parameters that guide the generation of joint representations. The decoder, based on the Marker–Joint Chain Inference Mechanism, computes attention distributions. The loss function for position prediction is defined as follows:

$$L_P = \lambda_1 L_{M_{occ}} + \lambda_2 L_{M_{shift}} + \lambda_3 L_J. \tag{5}$$

Since the number of occluded markers is relatively small compared to the total number of markers, we emphasize the importance of marker prediction by separating the marker completions and marker refinements into two distinct losses. $L_{M_{occ}}$ represents the error in solving the positions of the occluded markers. $L_{M_{shift}}$ represents the error in correcting the positions of the shifted markers, and L_J represents the error in solving the positions of the human joints. All three errors are computed using Euclidean distance. The parameters λ_1, λ_2 and λ_3 will be discussed in the experimental section.

4.1.2 Rotation Solver. After processing through the Position Solver, we obtain accurate joint and marker positions. Leveraging this precise and detailed input, we construct the Rotation Solver to estimate joint rotations.

As noted earlier, the positions of key reference markers are used to eliminate global transformations. Considering the forward kinematics of the human body, joint rotations can be inferred by fitting the skeletal model to the observed marker, indicating a strong correlation between marker locations and joint rotations. To capture this relationship, the rotation solver incorporates stacked attention modules.

In terms of loss function design, instead of using hierarchical weighting, we compute rotation errors for each joint independently. This design offers two key advantages: (i) it ensures equal weighting of rotation loss across all joints, thereby preventing error accumulation and propagation; (ii) hierarchical schemes impose fixed parent–child dependencies in joint rotations. We relax these constraints to allow more flexible motion modeling.

Finally, to ensure continuity during rotation regression training, we adopt the 6D representation [33] for computing rotation errors. Notably, the network operates without relying on temporal correlations, reducing preprocessing overhead and enabling real-time motion MoCap with deep learning.

4.2 Marker-Joint Chain Inference Mechanism

In real-world motion capture environments, occlusion of a marker by the body or external obstacles often leads to the simultaneous occlusion of neighboring markers. To address the challenges posed by such occlusion patterns, we conduct an in-depth investigation into the relationship between markers and joints, and accordingly design a marker-joint chain inference mechanism. This mechanism is driven by the key insight that markers and joints are mutually constrained.

Bidirectional Inference represents the mutual relationship between the positions of markers and joints. Joint features extracted from imperfect marker positions help integrate contextual information, thereby contributing to more accurate marker position refinement.

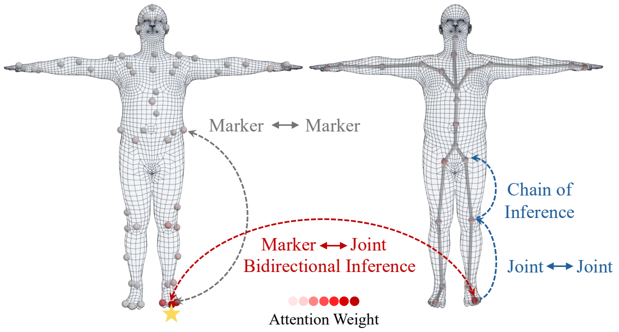


Figure 7: Marker-Joint Chain Inference Mechanism. The figure highlights a marker on the front of the foot (indicated by a star) and visualizes attention weights across other markers and joints, with red intensity indicating attention strength. Long-range dependencies are captured through bidirectional marker–joint reasoning and joint chain inference.

This insight is validated in our experiments. As illustrated in Fig.7, the red color gradient indicates the weights from one attention layer, with lighter shades corresponding to lower attention values. The yellow pentagon marks the target marker of interest. When estimating its position, the model relies not only on information from other markers but also on that from the joints. By learning the bidirectional correlations between markers and joints, the network iteratively integrates information from both sources, enabling mutual refinement and unified optimization of the final result.

Chain of Inference represents the formation of information pathways. As shown in Fig.7, a marker exhibits limited dependency on distant markers. This sheds light on the limitations of previous approaches that focus exclusively on mutual completion among markers, as they fail to effectively leverage all available information for accurate completion. The marker on the right toe is closely associated with the corresponding toe joint and even establishes a connection to the left hip joint through inter-joint relationships. When the foot markers are heavily occluded, the markers near the hip can help constrain the possible positions of the foot markers by predicting joint positions and performing chain-like reasoning. As a result, long-range dependencies are successfully established.

Previous methods typically infer joint positions and rotations from initially corrected marker positions. In contrast, the Marker-joint chain inference mechanism introduces a key distinction: it treats joints as intermediate nodes to establish long-range dependencies among markers. Specifically, marker information can be propagated through the close connections between nearby joints and other related joints. Since joint positions are tightly coupled with marker positions, they can in turn help refine or complete missing or distant marker observations. This simultaneous optimization of markers and joints leads to more accurate reconstruction results. In particular, precise and complete estimation of markers around the waist is crucial for subsequent spatial alignment procedures.

We formalize the modeling process as follows. The entire mechanism can be interpreted as an information diffusion process over a weighted directed graph. Specifically, let M denote the set of all markers, J the set of joints to be predicted, and V the complete set of nodes, where:

$$M = \{m_1, ..., m_M\}, J = \left\{j_1, ..., j_J\right\}, V = M \cup J. \tag{6}$$

We define the initial state of the directed graph as $h^{(0)}$, and the weighted propagation matrix at step t as $P^{(t)}$. Accordingly, the representation of the k-th marker after L steps can be expressed as:

$$h_k^{(L)} = \sum_{v=1}^{M+J} \left[P^{(L-1)} P^{(L-2)} \cdots P^{(0)}\right]_{k,v} h_v^{(0)}. \tag{7}$$

Similarly, the state of the k-th joint after L steps can be computed as:

$$h_{M+k}^{(L)} = \sum_{v=1}^{M+J} \left[P^{(L-1)} P^{(L-2)} \cdots P^{(0)}\right]_{M+k,v} h_v^{(0)}. \tag{8}$$

As shown in Fig.7, the chain reasoning between different joint positions across multiple steps can thus be expressed as:

$$[P^{(s)} P^{(s-1)}]_{j_c,j_a} = \sum_{j_b} P^{(s)}_{j_c,j_b} P^{(s-1)}_{j_b,j_a}. \tag{9}$$

5 Experiments

5.1 Experimental Details

5.1.1 Parameter Settings and Training Environment. In our experiment, the input markers are centralized by subtracting the centroid position. The occluded markers and input joints are set to different shared parameters respectively.

As for the weight of loss in position solving network, we use

$$L_P = L_{M_{occ}} + L_{M_{shift}} + 2 * L_J. \tag{10}$$

Since accurate positions of markers are very important for the operation of rotation network, the position solving loss of marker is set to the same weight as the joint position.

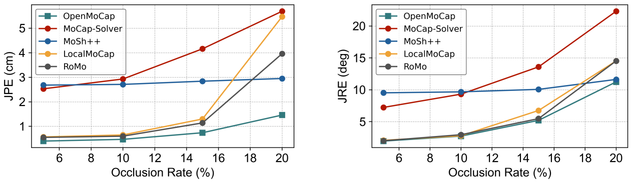
The whole network is trained on 1 GeForce RTX 4090 with 24GB memory, and the batch size is set to 256.

5.1.2 Dataset. A dataset of superior quality, encompassing a wide variety of motion types and body sizes, is crucial for enhancing the generalization capabilities of models. Three datasets are used in our experiments. In addition, we collected a set of real-world MoCap sequences to qualitatively evaluate and compare the performance of different approaches.

The first dataset is generated by driving the SMPL model using pose parameters from the CMU MoCap dataset [9] and shape parameters from the CAESAR dataset [28]. We refer to this dataset as the CMU dataset. We apply the corruption function proposed by Holden [14]. The dataset consists of 5k synthetic MoCap sequences, totaling 8m frames, with 1,700 characters and 5,100 marker configurations. As described in Sec.3, the second dataset we use is the CMU-Occlu Dataset. The third dataset is SFU Motion Capture Database [25]. This real MoCap dataset includes motion capture recordings from 8 actors, with a total of 44 manually annotated sequences.

		MoSh++	MoCap-Solver	Local MoCap	RoMo	Open MoCap
CMU	JPE (cm)	2.58	2.56	0.94	0.89	**0.41**
	JOE (°)	9.40	6.51	3.59	3.43	**2.52**
CMU-Occlu	JPE (cm)	2.72	2.95	1.23	1.16	**0.46**
	JOE (°)	9.68	6.83	3.80	3.54	**2.60**

Table 1: Comparison with other methods on different datasets.



(a) Joint Position Error (JPE). **(b) Joint Rotation Error (JRE).**

Figure 8: Error variation of different methods under varying marker occlusion ratios.

5.1.3 Evaluation Metrics. Given the positions of markers as input, the model outputs the positions and rotations of joints. In the following experiments, we use Joint Position Error (JPE) to represent the Euclidean distance between the predicted results and ground truth (GT). We also use Joint Orientation Error (JOE) to measure the discrepancy between the predicted and actual joint rotation angles.

5.2 Approach Comparisons

Each method is independently trained and tested on the CMU and CMU-Occlu datasets, respectively. Both datasets contain marker occlusions and positional perturbations. As shown in Tab.1, the increased level of occlusion in CMU-Occlu leads to a slight performance degradation across all methods.

MoSh++ [19], as a parameter optimization method, relies heavily on temporal continuity and struggles with frame-wise corrupted data. MoCap-Solver [6] depends on consistently visible key markers, limiting its robustness in real-world settings. LocalMoCap [26] and its successor RoMo [27] address occlusions by interpolating missing markers and predicting their positions using neural networks. This approach achieves favorable results on the CMU dataset with randomly simulated occlusions, where occlusion durations are short and adjacent markers can compensate for each other. Among these methods, OpenMoCap achieves the best performance on both the CMU and CMU-Occlu datasets.

To further evaluate the robustness of different methods under varying levels of occlusion, we divide the CMU-Occlu test set into four subsets based on occlusion severity: 5%, 10%, 15%, and 20%. For a given sequence, if more than half of its frames contain over 20% of occluded markers, it is assigned to the 20% occlusion group. Models trained on the CMU-Occlu dataset are then tested under each occlusion condition, and the results are shown in Fig.8.

		w/o marker-joint chain	w/o decoupled architecture	Open MoCap
CMU	JPE (cm)	0.75	0.61	**0.41**
	JOE (°)	3.34	3.16	**2.52**
CMU-Occlu	JPE (cm)	0.87	1.09	**0.46**
	JOE (°)	3.55	5.13	**2.60**

Table 2: Ablation studies of our method.

		MoCap-Solver	Local MoCap	RoMo	Open MoCap
CMU	JPE (cm)	5.50	1.93	1.46	**0.40**
	JOE (°)	10.03	4.86	4.78	**4.47**
CMU-Occlu	JPE (cm)	5.73	1.41	1.38	**0.39**
	JOE (°)	10.21	4.25	4.22	**4.10**

Table 3: Comparison of methods trained on CMU and CMU-Occlu datasets, with evaluation conducted on SFU.

As occlusion increases, all methods show rising error. MoSh++ struggles with intra-sequence marker shifts, leading to high overall error. LocalMoCap and RoMo perform well under low occlusion but degrade sharply on the CMU-Occlu dataset due to their limited recovery strategies. In contrast, OpenMoCap models occluded markers as learnable parameters and employs a marker–joint chain inference mechanism to capture long-range dependencies, maintaining strong performance under realistic, high-occlusion conditions.

Fig. 9 shows visualization results. When key markers are missing, MoCap-Solver fails to align accurately, leading to large discrepancies from the ground truth. In contrast, OpenMoCap achieves more accurate reconstructions under severe occlusions. For example, in the bottom-right yoga pose with occluded abdominal and chest markers, it produces the closest reconstruction to the ground truth.

5.3 Ablation Studies

To evaluate component effectiveness, we conducted two ablation studies. In the first, we removed the marker–joint chain and directly estimated joint positions from visible markers, predicting occluded ones afterward. As discussed in Sec.4.2, this weakens spatial reasoning and leads to inaccurate marker recovery, which further degrades rotation estimation. As shown in Tab.2, both joint position and rotation errors increase under this setting.

In the second experiment, we merged the position solver and rotation solver into a single network while proportionally increasing its depth. In the CMU-Occlu test set, certain sequences contain fewer than three visible key markers after occlusion. While the method performs satisfactorily on the CMU dataset, it requires the alignment process to be performed during data preprocessing, which can result in a notable degradation in performance on the CMU-Occlu dataset.

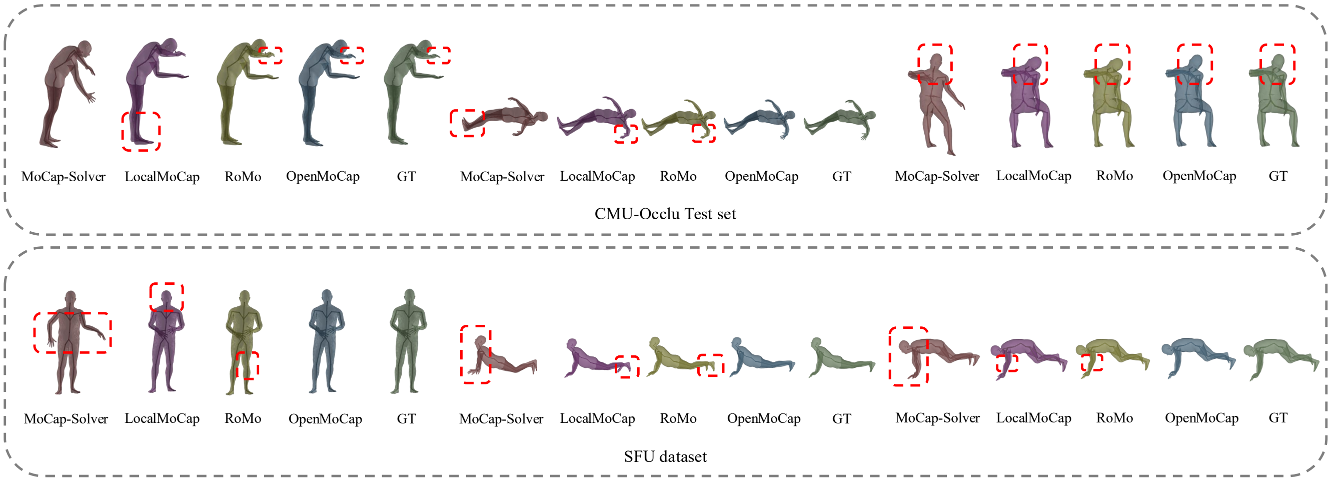


Figure 9: Qualitative evaluation of different models on CMU-Occlu test set and SFU dataset.

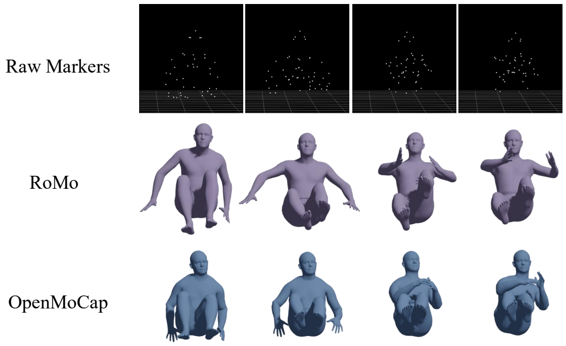


Figure 10: Qualitative evaluation of different models on real MoCap of a Russian twist.

	Mocap-Solver	LocalMoCap	RoMo	OpenMoCap
JPE (cm)	7.68	4.92	4.73	**1.01**
JOE (°)	28.83	16.73	15.98	**12.68**

Table 4: Comparison of different models on the processed MOYO dataset.

5.4 Dataset Analysis

To evaluate the effectiveness of the dataset in supporting real-world deployment of pre-trained models, we train each method separately on the CMU and CMU-Occlu datasets and test them on the real-world MoCap dataset SFU.

In the CMU dataset, complex motions often involve realistic occlusions around the waist and abdomen. As shown in Tab. 3, models like MoCap-Solver, which rely on marker visibility, struggle under such conditions, as misaligned inputs degrade training. In contrast, models with inherent occlusion robustness benefit from training on CMU-Occlu and perform better in real-world scenarios.

5.5 Application

We conducted the real-world experiment using MoSen MoCap system and tested the performance of different models. The actor performed a Russian twist, which involves four stages: sitting down, raising the legs, lifting the arms, and twisting the torso. As shown in Fig.10, the raw markers represent the captured visible markers. Significant occlusions occurred in the abdomen, leg, and hip regions. We compared the inference results of the SOTA method and OpenMoCap. In contrast to RoMo, OpenMoCap robustly reconstructed the motion and successfully completed the solving process.

5.6 Generalization Study

To further explore the generalization ability of OpenMoCap, we processed the MOYO dataset [29], which contains diverse and challenging yoga poses and strong self-occlusion, fine-tuned the models on this new domain and report the performance in Tab.4. The results indicate that OpenMoCap achieves a significant performance advantage compared to existing approaches.

6 Conclusion

In this paper, we conduct an in-depth analysis of two key factors that lead to performance degradation when deploying existing models in real-world MoCap environments, and propose targeted solutions for each. First, we introduce the CMU-Occlu dataset, which incorporates more realistic marker occlusion patterns, thereby improving the distributional alignment between synthetic training data and real-world test scenarios. Second, we propose the OpenMoCap solver, which establishes strong long-range dependencies between markers through a marker-joint inference mechanism. Experimental results demonstrate that CMU-Occlu enhances model generalization, while OpenMoCap achieves robust motion solving under diverse occlusion conditions, surpassing SOTA performance.

Acknowledgments

We sincerely thank the MobiSense group and anonymous reviewers for their insightful comments. This work is supported in part by the National Key Research Plan under grant No. 2021YFB2900100, the NSFC under grant No. 62372265, No. 62302254, and No. 62402276.

References

[1] Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, and Ariel Shamir. 2018. Self-similarity analysis for motion capture cleaning. In *Computer graphics forum*.

[2] Andreas Aristidou and Joan Lasenby. 2013. Real-time marker prediction and CoR estimation in optical motion capture. *The Visual Computer* (2013).

[3] Yiming Bao, Xu Zhao, and Dahong Qian. 2022. FusePose: IMU-vision sensor fusion in kinematic space for parametric human pose estimation. *IEEE Transactions on Multimedia* 25 (2022), 7736–7746.

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks* 106 (2018), 249–259.

[5] Anargyros Chatzitofis, Dimitrios Zarpalas, Petros Daras, and Stefanos Kollias. 2021. DeMoCap: Low-cost marker-based motion capture. *International Journal of Computer Vision* 129, 12 (2021), 3338–3366.

[6] Kang Chen, Yupan Wang, Song-Hai Zhang, Sen-Zhe Xu, Weidong Zhang, and Shi-Min Hu. 2021. Mocap-solver: A neural solver for optical motion capture data. *ACM Transactions on Graphics (TOG)* (2021).

[7] Xin Chen, Anqi Pang, Wei Yang, Yuexin Ma, Lan Xu, and Jingyi Yu. 2021. Sportscap: Monocular 3d human motion capture and fine-grained understanding in challenging sports videos. *International Journal of Computer Vision* 129, 10 (2021), 2846–2864.

[8] Zhuo Chen, Xu Zhao, and Xiaoyue Wan. 2022. Structural triangulation: A closed-form solution to constrained 3d human pose estimation. In *European Conference on Computer Vision*. Springer, 695–711.

[9] CMU. 2000. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/.

[10] Yinfu Feng, Jun Xiao, Yueting Zhuang, Xiaosong Yang, Jian J Zhang, and Rong Song. 2014. Exploiting temporal stability and low-rank structure for motion capture data refinement. *Information Sciences* (2014).

[11] Richard Hartley and Andrew Zisserman. 2003. *Multiple view geometry in computer vision*. Cambridge university press.

[12] Richard I Hartley and Peter Sturm. 1997. Triangulation. *Computer vision and image understanding* 68, 2 (1997), 146–157.

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 16000–16009.

[14] Daniel Holden. 2018. Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

[15] KyeongMin Kim, SeungWon Seo, DongHeun Han, and HyeongYeop Kang. 2024. DAMO: A Deep Solver for Arbitrary Marker Configuration in Optical Motion Capture. *ACM Transactions on Graphics* 44, 1 (2024), 1–14.

[16] Xin Liu, Yiu-ming Cheung, Shu-Juan Peng, Zhen Cui, Bineng Zhong, and Ji-Xiang Du. 2014. Automatic motion capture data denoising via filtered subspace clustering and low rank matrix approximation. *Signal processing* (2014).

[17] Umile Giuseppe Longo, Sergio De Salvatore, Arianna Carnevale, Salvatore Maria Tecce, Benedetta Bandini, Alberto Lalli, Emiliano Schena, and Vincenzo Denaro. 2022. Optical motion capture systems for 3D kinematic analysis in patients with shoulder disorders. *International Journal of Environmental Research and Public Health* 19, 19 (2022), 12033.

[18] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2023. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. Association for Computing Machinery.

[19] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*.

[20] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. 2015. The KIT whole-body human motion database. In *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, 329–336.

[21] Nicholas Milef, John Keyser, and Shu Kong. 2024. Towards Unstructured Unlabeled Optical Mocap: A Video Helps!. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

[22] Tomas Möller and Ben Trumbore. 1997. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools* 2, 1 (1997), 21–28.

[23] Meinard Müller, Andreas Baak, and Hans-Peter Seidel. 2009. Efficient and robust annotation of motion capture data. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 17–26.

[24] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. 2007. Mocap database hdm05. *Institut für Informatik II, Universität Bonn* 2, 7 (2007).

[25] Simon Fraser University National University of Singapore. [n. d.]. SFU Motion Capture Database. https://mocap.cs.sfu.ca/. Accessed: 2025-04-05.

[26] Xiaoyu Pan, Bowen Zheng, Xinwei Jiang, Guanglong Xu, Xianli Gu, Jingxiang Li, Qilong Kou, He Wang, Tianjia Shao, Kun Zhou, et al. 2023. A Locality-based Neural Solver for Optical Motion Capture. In *SIGGRAPH Asia 2023 Conference Papers*.

[27] Xiaoyu Pan, Bowen Zheng, Xinwei Jiang, Zijiao Zeng, Qilong Kou, He Wang, and Xiaogang Jin. 2024. RoMo: A Robust Solver for Full-body Unlabeled Optical Motion Capture. *arXiv preprint arXiv:2410.02788* (2024).

[28] Kathleen M Robinette, Sherri Blackwell, Hein Daanen, Mark Boehmer, Scott Fleming, Tina Brill, David Hoeferlin, and Dennis Burnsides. 2002. Civilian American and European surface anthropometry resource (CAESAR) final report. *DTIC Document* 1 (2002).

[29] Shashank Tripathi, Lea Müller, Chun-Hao P. Huang, Taheri Omid, Michael J. Black, and Dimitrios Tzionas. [n. d.]. 3D Human Pose Estimation via Intuitive Physics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[30] Eline Van der Kruk and Marco M Reijne. 2018. Accuracy of human motion capture systems for sport applications; state-of-the-art review. *European journal of sport science* 18, 6 (2018), 806–819.

[31] Junjie Wang, Zhenbo Yu, Zhengyan Tong, Hang Wang, Jinxian Liu, Wenjun Zhang, and Xiaoyan Wu. 2022. Ocr-pose: Occlusion-aware contrastive representation for unsupervised 3d human pose estimation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5477–5485.

[32] Tao Wang, Lei Jin, Zhang Wang, Xiaojin Fan, Yu Cheng, Yinglei Teng, Junliang Xing, and Jian Zhao. 2023. Decenternet: Bottom-up human pose estimation via decentralized pose representation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 1798–1808.

[33] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5745–5753.