Fast SAM 3D Body: Accelerating SAM 3D Body for Real-Time Full-Body Human Mesh Recovery

Timing Yang[1], Sicheng He[1], Hongyi Jing[1], Jiawei Yang[1], Zhijian Liu[2,3], Chuhang Zou[4†], and Yue Wang[1,3†]

[1] USC Physical Superintelligence (PSI) Lab
[2] University of California, San Diego
[3] NVIDIA
[4] Meta Reality Labs

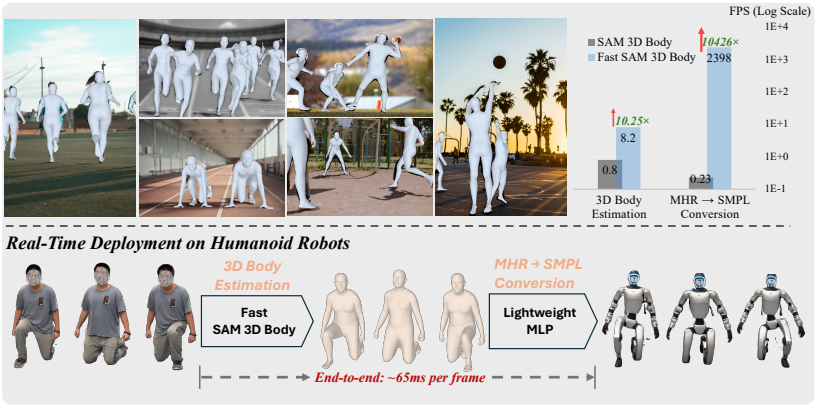

Fig. 1: Speed–accuracy overview of Fast SAM 3D Body. Top left: Qualitative results on in-the-wild images show our framework preserves high-fidelity reconstruction. Top right: Our method achieves up to a 10.25× end-to-end speedup over SAM 3D Body [43] and replaces the iterative MHR-to-SMPL bottleneck [8] with a 10,000× faster neural mapping (log scale for visual clarity). Bottom: Our system enables real-time humanoid robot control from a single RGB stream at ∼65 ms per frame on an NVIDIA RTX 5090.

Abstract. SAM 3D Body (3DB) achieves state-of-the-art accuracy in monocular 3D human mesh recovery, yet its inference latency of several seconds per image precludes real-time application. We present **Fast SAM 3D Body**, a training-free acceleration framework that reformulates the 3DB inference pathway to achieve interactive rates. By decoupling serial spatial dependencies and applying architecture-aware pruning, we enable parallelized multi-crop feature extraction and streamlined transformer decoding. Moreover, to extract the joint-level kinematics

† Joint corresponding authors.
Code is available at: `https://github.com/yangtiming/Fast-SAM-3D-Body`

(SMPL) compatible with existing humanoid control and policy learning frameworks, we replace the iterative mesh fitting with a direct feed-forward mapping, accelerating this specific conversion by over 10,000×. Overall, our framework delivers up to a 10.25× end-to-end speedup while maintaining largely on-par reconstruction fidelity, even surpassing 3DB on benchmarks such as LSPET. We demonstrate its utility by deploying Fast SAM 3D Body in a vision-only teleoperation system that—unlike methods reliant on wearable IMUs—enables **real-time humanoid control** and the direct collection of manipulation policies from a single RGB stream.

Keywords: Human Mesh Recovery · Acceleration · Humanoid Control

1 Introduction

Monocular 3D human mesh recovery (HMR)—estimating the 3D pose and shape of a person from a single RGB image—is a foundational capability for augmented reality, biomechanics, and human-robot interaction. For these applications, real-time responsiveness is as critical as accuracy; high-fidelity reconstructions are of limited utility if they cannot keep pace with human motion or provide immediate feedback for control loops. While the recent adoption of large vision backbones and expressive body models [1,3,5,31] has significantly advanced reconstruction accuracy, it has done so at the cost of immense computational overhead.

A prime exemplar of this trade-off is SAM 3D Body (3DB) [43]. While 3DB achieves state-of-the-art performance, the computational intensity of its multi-stage pipeline—robust detection [22], sequential encoding [36], and iterative MHR-to-SMPL conversion [8]—bounds inference to sub-FPS speeds. Prior acceleration efforts [4, 6, 11, 29, 37, 46, 47] primarily redesign ViT architectures. Because these methods optimize only isolated neural components, they are not designed to resolve the cross-stage dependencies and intensive post-processing required by comprehensive pipelines like 3DB, thus falling short of real-time end-to-end speeds. Furthermore, applying such architectural modifications requires extensive retraining, which risks degrading the exceptional generalization ability provided by 3DB's pre-trained backbones. Bridging this gap demands a holistic reformulation of the inference pathway—one that streamlines these compound dependencies to unlock real-time performance through a training-free approach that keeps 3DB's pretrained weights frozen, training only a lightweight MHR-to-SMPL MLP and pose denoiser.

To this end, we present **Fast SAM 3D Body**, a training-free acceleration framework that holistically reformulates the 3DB pipeline. As shown in Fig. 1, rather than redesigning the model architecture, we preserve 3DB's robust generalization ability while unlocking real-time performance through three methodological shifts: (i) **Spatial Dependency Decoupling:** We circumvent the baseline's serial body-to-hand decoding bottleneck by introducing a lightweight 2D pose prior [16]. By analytically deriving extremity bounding boxes from coarse keypoints, we decouple spatial localization from the main decoder, directly un-

locking parallelized, multi-crop feature extraction in a single batched forward pass. (ii) **Compute-Aware Decoding:** We prune redundant prompt queries and bypass iterative self-refinement. This yields a deterministic execution graph, unlocking low-level hardware compilation (via torch.compile and TensorRT) without sacrificing dynamic expressivity. (iii) **Neural Kinematic Projection:** Because standard humanoid control, teleoperation pipelines, and policy learning frameworks widely adopt the SMPL representation, projecting expressive MHR surfaces into this kinematic manifold is a prerequisite for downstream actuation. To bridge this topological gap, we replace the iterative MHR-to-SMPL conversion with a lightweight feedforward mapping, directly projecting MHR features into the actuatable joint space.

In summary, our key contributions are as follows:

- A training-free acceleration framework that holistically reformulates the multi-stage SAM 3D Body pipeline, achieving up to a 10.9× end-to-end speedup while maintaining largely on-par reconstruction fidelity, even surpassing 3DB on benchmarks such as LSPET.
- A learned feedforward MHR-to-SMPL projection module that replaces hundreds of iterative optimization steps, accelerating cross-topology mesh conversion by over 10,000× without compromising millimeter-level precision.
- We demonstrate the successful deployment of this framework in a vision-only, single-RGB teleoperation system that—unlike methods reliant on cumbersome wearable IMUs—enables real-time humanoid robot control and the direct collection of deployable whole-body manipulation policies.

2 Related Work

Human Mesh Recovery. Monocular HMR has transitioned from body-only regression [7, 9, 18, 23, 30, 40] to high-fidelity full-body estimation including hands and facial expression [1, 3, 5, 34]. While part-specific methods [32, 33] offer specialized accuracy for the extremities, they often lack the holistic context required for global trajectory estimation and temporal coherence [21, 35, 41]. This evolution toward expressive, unified models has necessitated large vision backbones, which—while significantly improving reconstruction quality—has made real-time deployment computationally expensive. 3DB [43] represents the current state-of-the-art in this trajectory. By leveraging a promptable encoder–decoder architecture and the expressive MHR [8] representation, it achieves unprecedented recovery. However, the rigor of its multi-stage design—incorporating dense detection, sequential multi-crop feature extraction, and iterative kinematic fitting—accumulates latency. Our work resolves these bottlenecks through holistic reformulations that enable real-time execution while retaining 3DB's robust generalization in a training-free manner.

Parametric Human Body Models. SMPL [26] remains a widely adopted representation for HMR, parameterizing pose and shape via learned blend shapes and

linear blend skinning. Recently, MHR [8] introduced a decoupled parameterization that separates skeletal structure from soft-tissue deformation. However, because standard evaluation benchmarks and existing robotic control frameworks (e.g., humanoid teleoperation pipelines) are built specifically around the SMPL skeletal topology, models predicting MHR must topologically translate their outputs for practical use. In 3DB, this cross-topology translation is performed via a computationally intensive hierarchical iterative optimization (Eq. (1)) that constitutes the primary pipeline bottleneck. We obviate this costly optimization by introducing a lightweight feedforward mapping (Eq. (4)) that achieves real-time inter-model projection with accuracy on par with iterative fitting.

Efficient Human Mesh Recovery. Efforts to reduce HMR inference costs have largely focused on two independent directions. The first is architectural redesign, including lighter encoder–decoders [4], efficient attention [11,46,47], and token/layer pruning [2, 6, 29, 37]. While effective, these methods typically address only the vision backbone and require extensive retraining. The second direction is system-level optimization—such as graph compilation and pipeline parallelism [10, 15, 24]—which has proven successful in 2D pose estimation but remains under-explored for multi-stage 3D pipelines. High-fidelity mesh recovery presents a unique "compound latency" profile, where detection, multi-crop encoding, and topology conversion all contribute to sub-real-time rates. Our work unifies these directions, applying holistic optimizations across all inference stages without retraining, thus preserving the "in-the-wild" robustness of the underlying state-of-the-art model.

3 Method

3.1 Preliminaries: SAM 3D Body (3DB) Pipeline

3DB [43] recovers full-body 3D human meshes from a single RGB image using a promptable encoder–decoder built on the expressive MHR [8] representation. While highly accurate, 3DB's inference pipeline suffers from structural latencies, which we outline below as the primary targets for our acceleration framework.

Detection and Encoding. Given an image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, a detector extracts body bounding boxes $\{b_i\}_{i=1}^{N}$, while a field-of-view estimator predicts camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times 3}$. Each cropped region $\mathbf{I}_{\text{body}} \in \mathbb{R}^{S\times S\times 3}$ passes through a vision backbone to extract dense features:

$$\mathbf{F} = \text{Enc}(\mathbf{I}_{\text{body}}) \in \mathbb{R}^{h\times w\times D}$$

where $h{=}w{=}S/p$, p is the patch size, and D is the embedding dimension. These features are optionally fused with Fourier-encoded pixel rays derived from $\mathbf{K}$.

Tokenized Transformer Decoding. The body decoder processes a concatenated sequence of learnable query tokens representing the initial MHR state, spatial prompts, and keypoints:

$$\mathbf{T} = [\,\mathbf{t}_{\text{mhr}},\ \mathbf{T}_{\text{prompt}},\ \mathbf{T}_{\text{kp2d}},\ \mathbf{T}_{\text{kp3d}},\ \mathbf{T}_{\text{hand}}\,] \in \mathbb{R}^{M\times D}$$

During the L-layer decoding, these tokens cross-attend to the image features $\mathbf{F}$. A defining characteristic of 3DB is its intermediate prediction mechanism: after every layer ℓ, the output token $\mathbf{t}_{\text{mhr}}^{(\ell)}$ decodes intermediate MHR parameters and camera estimates. Forward kinematics yields 3D joints $\mathbf{J}^{(\ell)}$ and projected 2D keypoints $\hat{\mathbf{J}}_{\text{2d}}^{(\ell)}$, which dynamically update the positional encodings for the subsequent layer:

$$\mathbf{P}_{\text{kp2d}}^{(\ell+1)} = \phi_{\text{2d}}\big(\hat{\mathbf{J}}_{\text{2d}}^{(\ell)}\big), \qquad \mathbf{P}_{\text{kp3d}}^{(\ell+1)} = \phi_{\text{3d}}\big(\mathbf{J}^{(\ell)} - \bar{\mathbf{J}}^{(\ell)}\big)$$

where $\phi_{\text{2d}}, \phi_{\text{3d}}$ are learned linear projections and $\bar{\mathbf{J}}^{(\ell)}$ is the pelvis location.

Sequential Hand Decoding and Refinement. Following body decoding, hand bounding boxes are predicted by a dedicated head. Each hand crop is independently encoded and decoded to yield refined hand MHR parameters, which are merged with the body. To ensure joint alignment, the merged 2D keypoints are fed back as spatial prompts for a complete second forward pass through the body decoder.

Iterative MHR-to-SMPL Conversion. Because standard benchmarks evaluate in the SMPL [26] topology, the predicted MHR mesh undergoes a hierarchical iterative optimization [8]:

$$\hat{\boldsymbol{\Theta}}_{\text{smpl}} = \arg\min_{\boldsymbol{\Theta}} \big\|\mathbf{V}_{\text{mhr}} - \mathbf{V}_{\text{smpl}}(\boldsymbol{\Theta})\big\|^2 + \mathcal{R}(\boldsymbol{\Theta}), \tag{1}$$

where $\mathbf{V}_{\text{mhr}}$ is the MHR mesh predicted by the decoder, $\mathbf{V}_{\text{smpl}}(\boldsymbol{\Theta})$ is the SMPL mesh, and $\mathcal{R}$ comprises anatomical regularizers. This cross-topology fitting requires hundreds of steps per person.

3.2 Acceleration

We address the compound latencies of the 3DB pipeline through a series of algorithmic reformulations. Rather than treating pipeline stages in isolation, we holistically resolve cross-stage dependencies, transform dynamic execution into deterministic graphs, and bypass optimization bottlenecks. The reformulations targeting the 3DB inference pathway itself require no weight changes; the MHR-to-SMPL conversion [8]—independent of 3DB itself and needed only for downstream kinematic control—is instead replaced with a learned topological projection, leaving all 3DB weights frozen. Fig. 2 illustrates our streamlined pipeline, and Fig. 3 provides a detailed stage-by-stage comparison with the original 3DB.

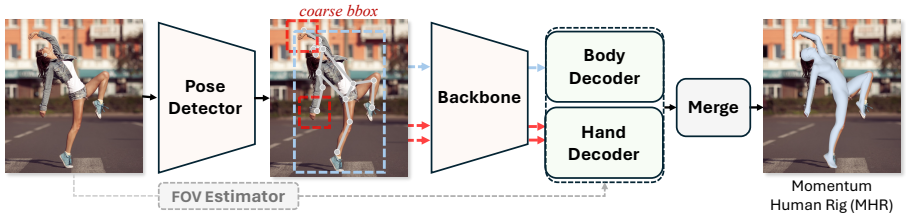

Fig. 2: ***Overview of the Fast SAM 3D Body pipeline.*** A pose detector predicts 2D body keypoints from which body and hand coarse bboxes are derived simultaneously, enabling all crops to be encoded in a single batched backbone pass. A lightweight FOV estimator predicts camera intrinsics. The body and hand decoders process the resulting features, and their outputs are merged to produce the final MHR mesh.

Decoupled Spatial Priors In the original 3DB architecture, hand detection is intrinsically entangled with the body decoder, creating a major serial dependency: hand crops cannot be spatially resolved until the body decoding concludes. We circumvent this bottleneck by introducing a lightweight spatial prior. By predicting coarse 2D body keypoints via a single-stage pose detector in an initial pass, we analytically derive the bounding boxes for extremities prior to any transformer execution. Specifically, the hand bounding box b_{hand} is deterministically computed as:

$$b_{\text{hand}} = \left[x_w - \tfrac{s}{2},\ y_w - \tfrac{s}{2},\ x_w + \tfrac{s}{2},\ y_w + \tfrac{s}{2} \right],$$

where (x_w, y_w) is the predicted wrist keypoint location and $s = \min(w_{\text{body}}, h_{\text{body}})/\alpha$ is a scale factor derived from the global body dimensions. Crucially, because the high-capacity downstream decoder is robust to minor spatial shifts, these initial keypoints do not require sub-pixel precision; they merely need to establish a reliable bounding region. This decoupling entirely abstracts the spatial preparation of extremities away from the main decoder, directly enabling batched multi-crop feature extraction while drastically reducing initial detection latency.

Static Graph Reformulation A critical bottleneck in expressive HMR pipelines is the dynamic, data-dependent nature of their execution. 3DB's reliance on asynchronous crop generation and dynamic token updating creates an unpredictable computation graph, incurring severe kernel launch and memory allocation overheads. We restructure the inference pathway into a static execution graph. Depending on the target hardware ecosystem, we utilize two parallel compilation strategies: (i) compiling the backbone into a TensorRT FP16 engine with dynamic batching, which fuses operations and executes asynchronously; or (ii) leveraging native static compilation to capture the forward pass as a CUDA Graph, allowing for deterministic replay with near-zero launch overhead. We apply an identical compilation strategy to the FOV estimator, purposely selecting its most compact model variant at the lowest resolution, as our empirical ablations indicate the FOV task saturates early at this capacity.

Compute-Aware Decoder Pruning The transformer decoder is the most complex component to optimize, involving the interplay of cross-attention layers, intermediate predictions, and feedback loops. We apply several complementary theoretical and structural strategies.

Intermediate prediction pruning. The original decoder executes a full intermediate prediction (IntermPred) after every layer $\ell < L$, requiring costly forward kinematics, skinning, and camera projection each time. However, we posit that early transformer layers primarily capture low-level semantic correlations, making full kinematic projection mathematically redundant at these initial stages. To exploit this, we introduce a configurable layer selection set $\mathcal{S} \subset \{0, \ldots, L-1\}$ that gates this execution. For any layer $\ell \notin \mathcal{S}$, the expensive IntermPred is bypassed entirely; the keypoint token positional encodings $\mathbf{P}_{\text{kp2d}}^{(\ell+1)}$ and $\mathbf{P}_{\text{kp3d}}^{(\ell+1)}$ simply retain their cached values from the most recent update, preserving the feedback signal without redundant recomputation.

Disabling keypoint-prompted refinement. The self-prompting refinement pass, in which predicted 2D keypoints are fed back as prompts for a second decoder forward pass, is removed at inference. Because our spatial priors are already accurately decoupled, eliminating this entire additional decoder evaluation results in no measurable degradation in reconstruction accuracy.

Pipeline Restructuring The original pipeline processes body and hand crops sequentially through three independent backbone forward passes, with hand crops prepared on the CPU. We consolidate this into a single batched pass.

GPU-native hand crop preprocessing. We eliminate the GPU–CPU–GPU round trip per hand by constructing differentiable sampling grids directly from the analytically derived bounding box coordinates. Both crops are extracted in a single bilinear interpolation pass on the GPU, removing data transfer overhead.

Merged body–hand batching. Since the spatial prior provides body and hand bounding boxes simultaneously, all three crops can be prepared in parallel without waiting for the body decoder output. We concatenate them into a single batch and execute one backbone forward pass:

$$[\,\mathbf{F}_{\text{body}},\, \mathbf{F}_{\text{L}},\, \mathbf{F}_{\text{R}}\,] = \text{Enc}\big([\,\mathbf{I}_{\text{body}},\, \mathbf{I}_{\text{L}},\, \mathbf{I}_{\text{R}}\,]\big). \tag{2}$$

The resulting feature maps are split and fed to the body and hand decoders, respectively, reducing backbone evaluations from three to one per person.

Operator-level optimizations. We apply a series of micro-optimizations throughout the pipeline, including replacing generic library calls with graph-traceable specialized operators, vectorizing per-joint loops, and caching frequently accessed parameters to eliminate redundant CPU–GPU synchronizations.

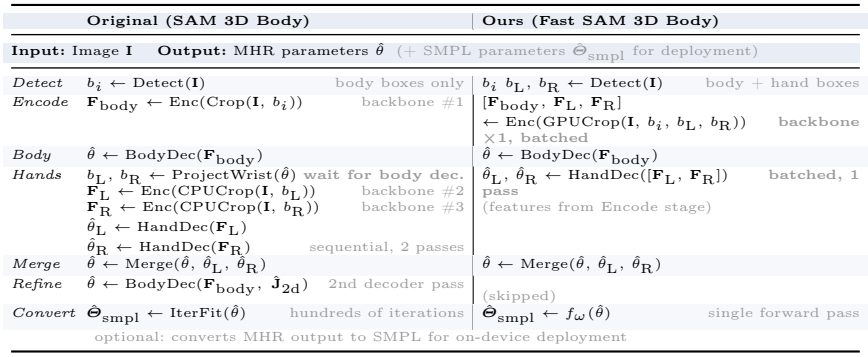

Fig. 3: ***Inference pipeline comparison.*** Original 3DB pipeline with serial execution (left) and our accelerated variant with batched execution (right).

Neural Kinematic Projection The iterative MHR-to-SMPL fitting in Eq. (1) is the slowest single stage, requiring hundreds of optimization iterations per person. We obviate this cross-topology bottleneck with a lightweight feedforward network f_ω that directly regresses the actuatable SMPL kinematic manifold from the expressive MHR surface in a single pass.

Topology bridging. Since MHR and SMPL use different mesh topologies ($N_v^{\text{mhr}} =$ 18,439 vs. $N_v^{\text{smpl}} = 6{,}890$ vertices), we first project the predicted MHR vertices onto the SMPL surface using precomputed barycentric coordinates:

$$\tilde{\mathbf{V}} = \mathcal{B}(\mathbf{V}_{\text{mhr}}) \in \mathbb{R}^{N_v^{\text{smpl}} \times 3}, \tag{3}$$

where $\mathcal{B}$ maps each SMPL vertex to its corresponding MHR triangle via stored face indices and barycentric weights. This operation is a single batched matrix multiply with no iterative optimization.

Network architecture and input representation. We subsample V_{sub} vertices from $\tilde{\mathbf{V}}$, subtract the centroid, and flatten the result into a vector $\mathbf{x} \in \mathbb{R}^{3V_{\text{sub}}}$. The network f_ω is a three-layer MLP ($3V_{\text{sub}} \to 512 \to 256 \to d_\Theta$) with ReLU activations. We set $V_{\text{sub}} = 1{,}500$, yielding an input dimension of 4,500 and approximately 2.5 M parameters. The output $d_\Theta = 76$ decomposes into global orientation (3), body pose (63, covering 21 body joints; the two hand joints are handled by the hand decoder and zeroed here), and shape coefficients (10):

$$\hat{\boldsymbol{\Theta}}_{\text{smpl}} = f_\omega(\mathbf{x}). \tag{4}$$

Training. We generate training pairs $\{(\mathbf{x}_i, \boldsymbol{\Theta}_i^*)\}$ by running the original iterative fitting (Eq. (1)) on a large set of 3DB predictions, where $\boldsymbol{\Theta}_i^*$ denotes the converged SMPL parameters. We apply SMPL forward kinematics to $\hat{\boldsymbol{\Theta}}_{\text{smpl}}$ to obtain the predicted mesh $\hat{\mathbf{V}}_{\text{smpl}}$, and train the MLP with:

$$\mathcal{L}_{\text{convert}} = \lambda_v \|\hat{\mathbf{V}}_{\text{smpl}} - \tilde{\mathbf{V}}\|_1 + \lambda_{\text{reg}} \|\hat{\boldsymbol{\Theta}}_{\text{smpl}} - \boldsymbol{\Theta}^*\|_2^2. \tag{5}$$

Kinematic prior refinement. The per-frame regression output may contain anatomically implausible poses (e.g., kinematic artifacts). We train a lightweight denoising MLP on clean AMASS [28] motion-capture sequences by adding synthetic noise to ground-truth poses and supervising the network to recover the original. At inference, the MLP takes the predicted SMPL body pose as input and projects it onto the learned manifold of natural human poses. This single-frame refinement adds negligible latency ($\sim$0.1 ms) and improves the plausibility of the converted output without affecting benchmark metrics.

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics. We adopt the standard HMR evaluation suite comprising three 3D error metrics—PA-MPJPE(PA) [44], MPJPE [14], and PVE [31], all reported in millimeters—together with PCK@0.05 [44] for 2D alignment. These are evaluated on six benchmarks that span a wide range of capture conditions: 3DPW [38], EMDB [19], RICH [13], Harmony4D [20], COCO [25], and LSPET [17]. Since 3DB natively predicts MHR parameters, we convert the predicted mesh to the SMPL [26] topology using the procedure described in [8] before computing all 3D metrics on SMPL-annotated benchmarks. For hand-specific evaluation, we additionally report PA-MPVPE and F-scores (F@5, F@15) on FreiHAND [48].

Evaluation Protocols. We report results under two complementary settings. In the *Oracle* protocol, ground-truth bounding boxes isolate encoder-decoder accuracy from detection errors, following [43] for controlled comparison. In the *Automatic* protocol, bounding boxes from a detector reflect end-to-end performance. Specifically, we evaluate the baseline 3DB with its original detector [22, 36] and our method with our proposed decoupled pose priors [16]. We report per-crop FPS under *Oracle* and per-frame FPS under *Automatic* to detail speed-accuracy trade-offs. For Harmony4D, we follow the leave-one-out split of [43].

Implementation Details. All experiments are conducted on a single NVIDIA RTX 6000 Ada Generation GPU with a 16-core AMD Ryzen Threadripper PRO 3955WX CPU (3.9 GHz, 32 threads), using PyTorch 2.5.1 and CUDA 11.5. All throughput numbers are reported with a batch size of one (single image at a time) to reflect the latency-critical setting. The acceleration techniques described in Sec. 3.2 are applied throughout; specific design choices are validated in the ablation studies below.

4.2 Comparison on Common Datasets

To comprehensively assess our pipeline, we report quantitative and qualitative results on both body reconstruction and hand-specific reconstruction.

Body reconstruction Table 1 compares our Fast SAM 3D Body pipeline against the original SAM 3D Body (3DB) [43] and a comprehensive set of recent

Fig. 4: ***Qualitative comparison.*** The original SAM 3D Body (left) and our Fast variant (right) yield visually comparable mesh reconstructions across diverse poses and multi-person scenes on 3DPW [38] and EMDB [19].

Models	3DPW (14)			EMDB (24)			RICH (24)			Harmony4D (24)		COCO	LSPET
	PA↓	MPJPE↓	PVE↓	PA↓	MPJPE↓	PVE↓	PA↓	MPJPE↓	PVE↓	PVE↓	MPJPE↓	PCK↑	PCK↑
HMR2.0b	54.3	81.3	93.1	79.2	118.5	140.6	48.1†	96.0†	110.9†	—	—	86.1	53.3
CameraHMR	35.1	56.0	65.9	43.3	70.3	81.7	34.0	55.7	64.4	84.6	70.8	80.5†	49.1†
PromptHMR	36.1	58.7	69.4	41.0	71.7	84.5	37.3	56.6	65.5	91.9	78.0	79.2†	55.6†
SMPLerX-H	46.6†	76.7†	91.8†	64.5†	92.7†	112.0†	37.4†	62.5†	69.5†	—	—	—	—
NLF-L+fit*	33.6	54.9	63.7	40.9	68.4	80.6	**28.7**†	**51.0**†	**58.2**†	97.3	84.9	74.9†	54.9†
WHAM	35.9	57.8	68.7	50.4	79.7	94.4	—	—	—	—	—	—	—
TRAM	35.6	59.3	69.6	45.7	74.4	86.6	—	—	—	—	—	—	—
GENMO	34.6	**53.9**	65.8	42.5	73.0	84.8	39.1	66.8	75.4	—	—	—	—
Oracle: evaluated with annotated bounding boxes													
3DB	33.8	54.8	**63.6**	38.2	**61.7**	**72.5**	30.9	53.7	60.3	**41.0**	**33.9**	**86.5**	67.8
Ours	**30.4**	59.5	68.9	**37.3**	64.3	74.4	33.8	58.8	64.2	46.9	41.9	**86.5**	**70.2**
Automatic: evaluated with detected bounding boxes													
3DB	**26.8**	57.6	68.5	36.9	67.9	**71.5**	33.9	**55.4**	**60.6**	44.1	37.3	85.1	66.5
Ours	29.7	**58.9**	**68.0**	**36.2**	**65.3**	75.6	**33.8**	58.7	64.1	**44.0**	**37.2**	**85.2**	**67.1**
*Throughput: 3DB → **Fast** (**speedup**×)*													
Oracle per crop	2.0→**9.0** (**4.5**×)			1.9→**8.4** (**4.4**×)			2.0→**8.3** (**4.2**×)			1.4→**8.7** (**6.2**×)		1.9→**9.7** (**5.1**×)	1.9→**9.7** (**5.1**×)
Automatic per frame	0.8→**6.6** (**8.3**×)			1.0→**8.3** (**8.3**×)			0.8→**8.2** (**10.3**×)			0.6→**6.5** (**10.9**×)		0.8→**6.3** (**7.9**×)	0.8→**7.6** (**9.5**×)

Table 1: ***Comparison on common benchmarks.*** Oracle uses ground-truth boxes; Automatic uses detected boxes. **Bold**: best, underline: second-best. †: public checkpoint. ∗: trained on RICH. Harmony4D: leave-one-out split [43]. PA = PA-MPJPE.

HMR methods, including both image-based approaches (HMR2.0b [9], CameraHMR [30], PromptHMR [40], SMPLer-X-H [3], NLF [34]) and video-based methods (WHAM [35], TRAM [41], GenMo [21]).

Oracle evaluation (GT bbox provided). Our method largely preserves the reconstruction quality of the original 3DB while achieving 4–6× higher throughput per crop. For instance, on 3DPW, our Fast variant accelerates the per-crop inference by over 4.5× (reaching 9.0 FPS), while incurring an MPJPE increase of under 5 mm; across datasets this trade-off grows with scene complexity, reaching +8.0 mm on the dense-contact Harmony4D benchmark. We note that the PA-MPJPE metric, which factors out global alignment, often shows smaller degradation than MPJPE. This indicates that the simplified model maintains accurate

local pose estimation, trading off only slightly in global positioning. Qualitative visualizations (Fig. 4) further corroborate this preserved spatial accuracy.

Automatic evaluation (use predicted bbox). In the Automatic setting, our method frequently matches or surpasses the 3DB baseline (e.g., on EMDB and Harmony4D). This confirms that extracting hand crops from coarse YOLO11-Pose [16] wrist keypoints provides a sufficient spatial prior for the robust downstream decoder, requiring no sub-pixel precision. Crucially, breaking the baseline's serial dependency unlocks batched TensorRT execution, driving an 8 to 11× end-to-end throughput boost while largely preserving geometric fidelity.

2D alignment. On COCO and LSPET (PCK@0.05), our method achieves highly competitive results under Oracle evaluation, matching or exceeding the original 3DB and all other baselines. The consistent improvement on LSPET may be attributed to the simplified decoder encouraging more robust 2D projections on out-of-domain data.

Failure and challenging cases. Reconstruction degrades in four scenarios: (i) close two-person contact, with error roughly proportional to inter-person overlap; (ii) deep knee/hip flexion under self-occlusion; (iii) extreme limb extensions, where distal joints degrade while the trunk is preserved; and (iv) mild global root drift on long outdoor sequences, where local pose accuracy (PA-MPJPE) is nonetheless preserved.

Metric	3DB	Ours
PA-MPJPE↓	5.50	5.67
PA-MPVPE↓	6.20	6.46
F@5↑	0.737	0.714
F@15↑	0.988	0.987
*Throughput: 3DB → **Ours** (**speedup**×)*		
FPS	9.8→**45.5** (**4.6**×)	

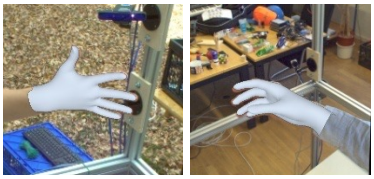

Fig. 5: ***Hand reconstruction.*** (Left) Quantitative comparison on FreiHAND [48]: our lightweight hand decoder maintains competitive accuracy with 4.6× faster inference. (Right) Qualitative results.

Hand reconstruction. Figure 5 evaluates our lightweight hand decoder on FreiHAND [48]. The simplified decoder maintains competitive accuracy across all metrics, with differences within 0.3 mm on PA-MPJPE/PA-MPVPE and near-identical F@15 scores, while achieving a 4.6× throughput improvement. Qualitatively, our method recovers high-fidelity hand meshes robust to diverse poses and background clutter.

4.3 Real-World Deployment

Deploying on humanoid platforms requires accurate SMPL output at real-time throughput. We address both, enabling vision-only teleoperation of the Unitree G1 from a single RGB stream.

Method	3DPW			EMDB		
	PA↓	MPJPE↓	PVE↓	PA↓	MPJPE↓	PVE↓
(3DB) Iterative fitting	**30.4**	59.5	68.9	**37.3**	**64.3**	74.4
(Ours) Feedforward MLP	31.1	**59.4**	**68.6**	**37.3**	64.8	**68.8**
*Throughput: Iterative → **MLP** (**speedup**×)*						
FPS	0.24→**1992** (**8300**×)			0.23→**2398** (**10426**×)		

Table 2: ***Lightweight MHR-to-SMPL conversion.*** Iterative fitting vs. feedforward MLP, Oracle protocol. Near-identical accuracy at $>10^4\times$ speedup.

Lightweight MHR-to-SMPL conversion. Table 2 compares the iterative fitting (Eq. 1) against our feedforward MLP (Eq. 4) (excludes the AMASS denoiser). The MLP achieves near-identical joint accuracy to the iterative baseline across both benchmarks, and even noticeably improves the vertex-level alignment (PVE) on EMDB. Crucially, this feedforward approach yields a speedup of roughly four orders of magnitude ($\sim 10^4\times$), effectively eliminating the final latency bottleneck in our pipeline.

Humanoid teleoperation. We demonstrate a real-time, vision-only teleoperation system for the Unitree G1 humanoid robot using a single RGB camera. Operating at an end-to-end latency of just 65 ms on an NVIDIA RTX 5090 (145/180 ms on consumer RTX 4090/4060 GPUs), our framework recovers full-body MHR parameters and instantly translates them via our neural kinematic projection (Sec. 3.2). In our multi-threaded pipeline, Fast 3DB inference throughput is 15.2 Hz, roughly half the 30 Hz camera rate (52.4% frame drop), composed of 65.5 ms inference, 0.5 ms MHR-to-SMPL conversion, and a 17.2 ms inter-thread queue. These derived SMPL kinematics are directly fed as reference motion into SONIC [27] to drive the G1. Qualitative results are visualized in Fig. 6. We further validate our pipeline by finetuning Ψ_0 [42]—a VLA model pre-trained on diverse human [12] and humanoid [45] priors—using 40 demonstrations collected via our system. Evaluated on a whole-body manipulation task (bimanual grasping, squatting, and lateral stepping), the policy achieves an 80% success rate (Fig. 7). This confirms our framework generates high-quality motion data suitable for deployable policy learning.

4.4 Ablation Studies

We conduct extensive ablations to justify each design choice, evaluating on 3DPW with oracle GT bounding boxes unless otherwise noted.

Decoder layer selection. Table 3 evaluates the number of transformer layers in the body decoder. The multi-layer configuration $\{0, 1, 2\}$ achieves the best accuracy–speed trade-off, effectively matching the full model's reconstruction quality while delivering a noticeable boost in inference speed. Further reducing the number of layers leads to clear precision loss. While single-layer options offer higher throughput, they fail to maintain the robust performance of our selected multi-layer setting.

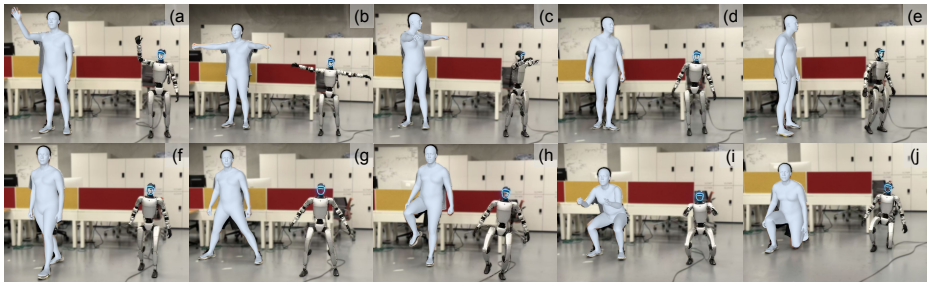

Fig. 6: ***Qualitative Results of Humanoid Teleoperation.*** The system tracks diverse whole-body motions including upper-body gestures (a), body rotations (b–e), walking (f), wide stance (g), single-leg standing (h), squatting (i), and kneeling (j).

Fig. 7: ***Humanoid Policy Rollout.*** The robot grasps a box on the table with both hands, squats down, and steps to the right.

Multi-layer					*Single-layer*				
Layers	MPJPE↓	PA↓	PVE↓	FPS↑	Layer	MPJPE↓	PA↓	PVE↓	FPS↑
{0,1,2}	58.96	31.33	69.28	7.18	L0	60.17	33.76	**70.26**	7.52
{0,1,2,3,4}	**58.89**	**31.21**	**69.21**	6.88	L1	60.03	33.41	71.27	7.65
{0,2,4}	59.00	31.43	69.53	7.18	L2	**59.15**	**32.62**	71.21	7.71
{0,4}	60.23	33.67	70.22	7.31	L3	59.89	33.04	72.59	7.88
∅	60.73	33.38	72.41	**8.19**	L4	60.16	33.03	71.86	**7.95**

Table 3: ***Ablation on intermediate prediction layers for the body decoder.*** Multi-layer (left) retains a subset of the five original layers; single-layer (right) keeps only one. Hand decoder layers fixed at {0,1}. Shaded rows indicate selected configuration.

Size	MPJPE↓	PA↓	PVE↓	FPS↑
128	86.78	62.33	129.10	**11.69**
256	61.70	33.45	75.33	10.73
384	59.01	**31.00**	70.22	8.96
448	**58.95**	31.02	69.50	8.25
512	58.96	31.33	**69.28**	7.18

Table 4: ***Input resolution ablation.*** Smaller crops improve throughput at the cost of reconstruction accuracy. Shaded row indicates selected configuration.

Input resolution. Table 4 demonstrates that input resolution significantly impacts performance. While accuracy saturates at higher resolutions, the 384/448-pixel variants provides a highly efficient alternative with minimal quality loss. We maintain the maximum resolution as the default to remain consistent with the original 3DB baseline, but note that the mid-range option is ideal for latency-sensitive applications.

Pose-dependent correctives and keypoint prompts. Table 6 evaluates the removal of the 3DB baseline's heavy refinement modules. Disabling pose-dependent correctives and keypoint prompts significantly accelerates inference, as these components primarily govern high-frequency surface details rather than

Model	Setting	MPJPE↓	PA↓	PVE↓	FPS↑
(a) Model size					
small (TRT)	35M	59.08	31.33	69.33	7.34
small	35M	59.04	31.34	69.32	**7.38**
base	104M	59.06	31.35	69.35	7.31
large	331M	**58.95**	**31.32**	**69.23**	7.24
(b) Resolution level					
0	∼1200	59.08	31.34	69.33	**7.45**
5	∼2400	58.95	31.33	**69.18**	7.42
9	∼3600	**58.95**	**31.32**	69.23	7.38

Table 5: *FOV estimator ablation.* (a) Model size has negligible impact on accuracy (params). (b) Resolution has diminishing returns (tokens). Shaded rows indicate selected configuration.

Component	Setting	MPJPE↓	PA↓	PVE↓	FPS↑
(a) Pose-dependent Blend Shapes					
Correctives	OFF	58.96	31.33	69.28	**7.18**
Correctives	ON	**58.04**	**31.00**	**69.26**	6.89
(b) Keypoint Prompt					
KP Prompt	OFF	58.96	**31.33**	**69.28**	**7.18**
KP Prompt	ON	**58.92**	31.34	70.96	6.07
(c) Corr. + KP combined					
Corr.+KP	OFF	58.96	31.33	**69.28**	**7.18**
Corr.+KP	ON	**58.00**	**31.01**	70.94	5.78

Table 6: *Component ablations.* Each section toggles one component from the Fast baseline. (c) shows the cumulative effect of all simplifications. Shaded rows indicate selected configuration.

joint-level kinematics. These results confirm that our streamlined architecture — relying on decoupled spatial priors derived from a lightweight pose detector rather than heavy iterative prompting—delivers a highly efficient alternative with negligible impact on overall reconstruction quality.

FOV estimator. Table 5 ablates the MoGe-2 [39] field-of-view estimator across model sizes and input resolutions. We find that accuracy remains nearly constant across all model scales, suggesting that the FOV estimation task is well-saturated and the smallest model is sufficient. Similarly, increasing the input resolution yields negligible accuracy gains while slightly reducing throughput. Consequently, we adopt the small model at the lowest resolution to maximize speed without sacrificing performance.

Detector	Setting	Recall	MPJPE↓	PA↓	PVE↓	FPS↑
(a) Detector type						
YOLO11m-Pose	body + hand	99.2%	**58.49**	**30.71**	**67.36**	**3.50**
YOLO11m (det only)	body only	99.7%	58.81	30.97	67.71	2.60
ViTDet-H	body only	**100%**	58.79	30.98	67.70	1.05
(b) YOLO-Pose model size						
YOLO11n-Pose	3M	98.3%	**58.30**	**30.54**	**67.06**	**4.15**
YOLO11s-Pose	10M	99.0%	58.47	30.75	67.31	3.56
YOLO11m-Pose	20M	**99.3%**	58.49	30.71	67.36	3.50
YOLO11l-Pose	26M	**99.3%**	58.57	30.77	67.41	3.47
YOLO11x-Pose	59M	**99.3%**	58.63	30.85	67.47	3.38
(c) TRT vs PyTorch						
YOLO11m-Pose	PyTorch	99.2%	58.49	**30.71**	67.36	3.50
YOLO11m-Pose	TensorRT FP16	99.2%	**58.48**	30.72	**67.32**	**3.58**

Table 7: *Person detector ablation (automatic mode).* (a) YOLO11-Pose replaces ViTDet-H with comparable accuracy at higher throughput. (b) Larger model sizes yield diminishing returns beyond medium. (c) TensorRT conversion provides a modest additional speedup. Shaded rows indicate the selected configurations.

Person detector. Table 7 evaluates detector choices under the Automatic protocol. YOLO11m-Pose [16] achieves high recall while natively providing both body and hand bounding boxes, eliminating the need for a separate hand detector. Compared to the heavyweight ViTDet-H [22], our selected detector offers a substantial speedup with negligible impact on recall and downstream accuracy. We also find that larger model variants offer diminishing returns, with increased parameter counts failing to improve performance. Finally, TensorRT conversion provides a modest additional throughput gain with no loss in accuracy.

Batch Mode	Description	MPJPE↓	PA↓	PVE↓	FPS↑
full_batch	Body + Hand batched	57.81	30.97	70.85	**5.28**
hand_batch	Hand only batched	57.75	30.97	70.79	5.20
no_batch	No batching	**57.67**	**30.96**	**70.73**	4.39

Table 8: ***Batch mode ablation.*** Full batching (body + hand crops in a single forward pass) improves throughput with negligible accuracy overhead. Shaded row indicates the selected configuration.

Batching strategy. Table 8 compares different batching strategies. Full batching, which processes body and hand crops in a single forward pass, yields a significant throughput improvement compared to processing them individually. The accuracy overhead introduced by full batching is negligible, confirming that shared backbone computation across different crops does not introduce significant interference. Therefore, we adopt full batching as our default strategy to maximize inference speed.

5 Conclusion

We present Fast SAM 3D Body, a training-free framework that holistically reformulates the SAM 3D Body pipeline for near-real-time human mesh recovery. Our optimizations deliver up to a 10.25× end-to-end speedup while preserving the baseline's high-fidelity reconstruction. We further introduce a lightweight neural kinematic projection that replaces the iterative MHR-to-SMPL bottleneck, achieving over 10,000× faster topological translation. We validate the practical impact of these latency gains by demonstrating near-real-time, vision-only teleoperation of the humanoid robot directly from a single RGB stream.

Acknowledgment

The USC Physical Superintelligence Lab acknowledges generous support from Toyota Research Institute, Dolby, Google DeepMind, Capital One, Nvidia, Bosch, NSF, and Qualcomm. Yue Wang is also supported by a Powell Research Award. We thank Songlin Wei and Boqian Li for their assistance with real-world policy learning and evaluation.

References

1. Baradel, F., Armando, M., Galaaoui, S., Brégier, R., Weinzaepfel, P., Rogez, G., Lucas, T.: Multi-hmr: Multi-person whole-body human mesh recovery in a single shot. In: European Conference on Computer Vision. pp. 202–218. Springer (2024)
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster (2022)
3. Cai, Z., Yin, W., Zeng, A., Wei, C., Sun, Q., Yanjun, W., Pang, H.E., Mei, H., Zhang, M., Zhang, L., et al.: Smpler-x: Scaling up expressive human pose and shape estimation. vol. 36, pp. 11454–11468 (2023)
4. Cho, J., Youwang, K., Oh, T.H.: Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In: European Conference on Computer Vision. pp. 342–359. Springer (2022)
5. Choutas, V., Pavlakos, G., Bolkart, T., Tzionas, D., Black, M.J.: Monocular expressive body regression through body-driven attention. In: European Conference on Computer Vision. pp. 20–40. Springer (2020)
6. Dou, Z., Wu, Q., Lin, C., Cao, Z., Wu, Q., Wan, W., Komura, T., Wang, W.: Tore: Token reduction for efficient human mesh recovery with transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15143–15155 (2023)
7. Dwivedi, S.K., Sun, Y., Patel, P., Feng, Y., Black, M.J.: Tokenhmr: Advancing human mesh recovery with a tokenized pose representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1323–1333 (2024)
8. Ferguson, A., Osman, A.A.A., Bescos, B., Stoll, C., Twigg, C., Lassner, C., Otte, D., Vignola, E., Prada, F., Bogo, F., Santesteban, I., Romero, J., Zarate, J., Lee, J., Park, J., Yang, J., Doublestein, J., Venkateshan, K., Kitani, K., Kavan, L., Farra, M.D., Hu, M., Cioffi, M., Fabris, M., Ranieri, M., Modarres, M., Kadlecek, P., Khirodkar, R., Abdrashitov, R., Prévost, R., Rajbhandari, R., Mallet, R., Pearsall, R., Kao, S., Kumar, S., Parrish, S., Yu, S.I., Saito, S., Shiratori, T., Wang, T.L., Tung, T., Xu, Y., Dong, Y., Chen, Y., Xu, Y., Ye, Y., Jiang, Z.: Mhr: Momentum human rig (2025), https://arxiv.org/abs/2511.15586
9. Goel, S., Pavlakos, G., Rajasegaran, J., Kanazawa, A., Malik, J.: Humans in 4d: Reconstructing and tracking humans with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14783–14794 (2023)
10. Guo, Y., Liu, J., Li, G., Mai, L., Dong, H.: Fast and flexible human pose estimation with hyperpose. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3763–3766 (2021)
11. Heo, J., Hu, G., Wang, Z., Yeung-Levy, S.: Deforhmr: Vision transformer with deformable cross-attention for 3d human mesh recovery. In: 2025 International Conference on 3D Vision (3DV). pp. 1594–1604. IEEE (2025)
12. Hoque, R., Huang, P., Yoon, D.J., Sivapurapu, M., Zhang, J.: Egodex: Learning dexterous manipulation from large-scale egocentric video. arXiv preprint arXiv:2505.11709 (2025)
13. Huang, C.H.P., Yi, H., Höschle, M., Safroshkin, M., Alexiadis, T., Polikovsky, S., Scharstein, D., Black, M.J.: Capturing and inferring dense full-body human-scene contact. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13274–13285 (2022)
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. vol. 36, pp. 1325–1339. IEEE (2013)

15. Jiang, T., Lu, P., Zhang, L., Ma, N., Han, R., Lyu, C., Li, Y., Chen, K.: Rtmpose: Real-time multi-person pose estimation based on mmpose. arXiv preprint arXiv:2303.07399 (2023)
16. Jocher, G., Qiu, J.: Ultralytics yolo11 (2024), https://github.com/ultralytics/ultralytics
17. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: CVPR 2011. pp. 1465–1472. IEEE (2011)
18. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7122–7131 (2018)
19. Kaufmann, M., Song, J., Guo, C., Shen, K., Jiang, T., Tang, C., Zárate, J.J., Hilliges, O.: EMDB: The Electromagnetic Database of Global 3D Human Pose and Shape in the Wild. In: International Conference on Computer Vision (ICCV) (2023)
20. Khirodkar, R., Song, J.T., Cao, J., Luo, Z., Kitani, K.: Harmony4d: A video dataset for in-the-wild close human interactions. vol. 37, pp. 107270–107285 (2024)
21. Li, J., Cao, J., Zhang, H., Rempe, D., Kautz, J., Iqbal, U., Yuan, Y.: Genmo: A generalist model for human motion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11766–11776 (2025)
22. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European conference on computer vision. pp. 280–296. Springer (2022)
23. Li, Z., Liu, J., Zhang, Z., Xu, S., Yan, Y.: Cliff: Carrying location information in full frames into human pose and shape estimation. In: European Conference on Computer Vision. pp. 590–606. Springer (2022)
24. Liang, W., Yuan, Y., Ding, H., Luo, X., Lin, W., Jia, D., Zhang, Z., Zhang, C., Hu, H.: Expediting large-scale vision transformer for dense prediction without fine-tuning. vol. 35, pp. 35462–35477 (2022)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
26. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2. pp. 851–866 (2023)
27. Luo, Z., Yuan, Y., Wang, T., Li, C., Chen, S., Castañeda, F., Cao, Z.A., Li, J., Minor, D., Ben, Q., Da, X., Ding, R., Hogg, C., Song, L., Lim, E., Jeong, E., He, T., Xue, H., Xiao, W., Wang, Z., Yuen, S., Kautz, J., Chang, Y., Iqbal, U., Fan, L., Zhu, Y.: Sonic: Supersizing motion tracking for natural humanoid whole-body control. arXiv preprint arXiv:2511.07820 (2025)
28. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: International Conference on Computer Vision. pp. 5442–5451 (Oct 2019)
29. Mehraban, S., Iaboni, A., Taati, B.: Fasthmr: Accelerating human mesh recovery via token and layer merging with diffusion decoding. arXiv preprint arXiv:2510.10868 (2025)
30. Patel, P., Black, M.J.: Camerahmr: Aligning people with perspective. In: 2025 International Conference on 3D Vision (3DV). pp. 1562–1571. IEEE (2025)
31. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10975–10985 (2019)

32. Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., Malik, J.: Reconstructing hands in 3d with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9826–9836 (2024)
33. Potamias, R.A., Zhang, J., Deng, J., Zafeiriou, S.: Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 12242–12254 (2025)
34. Sárándi, I., Pons-Moll, G.: Neural localizer fields for continuous 3d human pose and shape estimation. vol. 37, pp. 140032–140065 (2024)
35. Shin, S., Kim, J., Halilaj, E., Black, M.J.: Wham: Reconstructing world-grounded humans with accurate 3d motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2070–2080 (2024)
36. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv preprint arXiv:2508.10104 (2025)
37. Tian, S., Szafranski, C., Zheng, C., Yao, F., Louri, A., Chen, C., Zheng, H.: Vita: Vit acceleration for efficient 3d human mesh recovery via hardware-algorithm co-design. In: Proceedings of the 61st ACM/IEEE Design Automation Conference. pp. 1–6 (2024)
38. Von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: Proceedings of the European conference on computer vision (ECCV). pp. 601–617 (2018)
39. Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J.: Moge-2: Accurate monocular geometry with metric scale and sharp details. arXiv preprint arXiv:2507.02546 (2025)
40. Wang, Y., Sun, Y., Patel, P., Daniilidis, K., Black, M.J., Kocabas, M.: Prompthmr: Promptable human mesh recovery. In: Proceedings of the computer vision and pattern recognition conference. pp. 1148–1159 (2025)
41. Wang, Y., Wang, Z., Liu, L., Daniilidis, K.: Tram: Global trajectory and motion of 3d humans from in-the-wild videos. In: European Conference on Computer Vision. pp. 467–487. Springer (2024)
42. Wei, S., Jing, H., Li, B., Zhao, Z., Mao, J., Ni, Z., He, S., Liu, J., Liu, X., Kang, K., Zang, S., Yuan, W., Pavone, M., Huang, D., Wang, Y.: ψ_0: An open foundation model towards universal humanoid loco-manipulation (2026), `https://arxiv.org/abs/2603.12263`
43. Yang, X., Kukreja, D., Pinkus, D., Sagar, A., Fan, T., Park, J., Shin, S., Cao, J., Liu, J., Ugrinovic, N., et al.: Sam 3d body: Robust full-body human mesh recovery. arXiv preprint arXiv:2602.15989 (2026)
44. Zhang, J., Nie, X., Feng, J.: Inference stage optimization for cross-scenario 3d human pose estimation. vol. 33, pp. 2408–2419 (2020)
45. Zhao, Z., Jing, H., Liu, X., Mao, J., Jha, A., Yang, H., Xue, R., Zakharor, S., Guizilini, V., Wang, Y.: Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation (2025), `https://arxiv.org/abs/2510.08807`
46. Zheng, C., Liu, X., Qi, G.J., Chen, C.: Potter: Pooling attention transformer for efficient human mesh recovery. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1611–1620 (2023)
47. Zheng, C., Mendieta, M., Yang, T., Qi, G.J., Chen, C.: Feater: An efficient network for human reconstruction via feature map-based transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13945–13954 (2023)

48. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., Brox, T.: Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 813–822 (2019)