Hamba: Single-view 3D Hand Reconstruction with Graph-guided Bi-Scanning Mamba

Haoye Dong[*,†] **Aviral Chharia**[*] **Wenbo Gou**[*]
Francisco Vicente Carrasco **Fernando De la Torre**
Carnegie Mellon University
{haoyed, achharia, wgou, fvicente, ftorre}@andrew.cmu.edu
https://humansensinglab.github.io/Hamba/

Abstract

3D Hand reconstruction from a single RGB image is challenging due to the articulated motion, self-occlusion, and interaction with objects. Existing SOTA methods employ attention-based transformers to learn the 3D hand pose and shape, yet they do not fully achieve robust and accurate performance, primarily due to inefficiently modeling spatial relations between joints. To address this problem, we propose a novel graph-guided Mamba framework, named **Hamba**, which bridges graph learning and state space modeling. Our core idea is to reformulate Mamba's scanning into graph-guided bidirectional scanning for 3D reconstruction using a few effective tokens. This enables us to efficiently learn the spatial relationships between joints for improving reconstruction performance. Specifically, we design a Graph-guided State Space (GSS) block that learns the graph-structured relations and spatial sequences of joints and uses 88.5% fewer tokens than attention-based methods. Additionally, we integrate the state space features and the global features using a fusion module. By utilizing the GSS block and the fusion module, Hamba effectively leverages the graph-guided state space features and jointly considers global and local features to improve performance. Experiments on several benchmarks and in-the-wild tests demonstrate that Hamba significantly outperforms existing SOTAs, achieving the PA-MPVPE of 5.3mm and F@15mm of 0.992 on FreiHAND. At the time of this paper's acceptance, Hamba holds the top position, **Rank 1**, in two competition leaderboards[1] on 3D hand reconstruction.

1 Introduction

3D Hand reconstruction has numerous applications across multiple fields, which include robotics, animation, human-computer interaction, and AR/VR [11, 34, 71, 18, 103]. However, reconstructing 3D hands from a single RGB image without body context or camera parameters remains a difficult challenge in computer vision. Recent works primarily explored transformers [14, 19, 51, 52, 70, 95, 73, 45, 55] for this task and achieved SOTA performance by utilizing attention mechanism. METRO [51] introduced a multi-layer transformer, using self-attention to learn vertex-vertex and vertex-joint relations. MeshGraphormer [52] integrated graph convolutions with a transformer to further enhance the reconstruction performance. Recently, HaMeR [70] designed a ViT-based model [19], using ViTPose [95] weights and large datasets to achieve better performance.

However, the above models fail to reconstruct a robust mesh in challenging in-the-wild scenarios that have occlusions, truncation, and hand-hand or hand-object interactions (See Figure 5 for visual

[*]Equal contribution. [†]Corresponding author.

[1]HO3Dv2 and HO3Dv3 Leaderboards: https://codalab.lisn.upsaclay.fr/competitions/4318#results, https://codalab.lisn.upsaclay.fr/competitions/4393#results

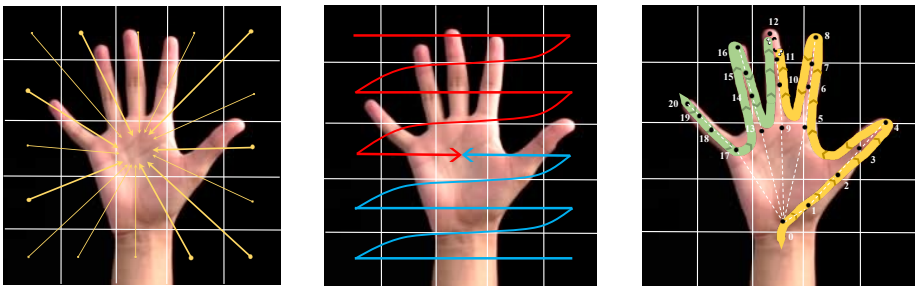
38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0069

Figure 1: **In-the-wild visual results of Hamba**. Hamba achieves significant performance in various in-the-wild scenarios, including hand interaction with objects or hands, different skin tones, different angles, challenging paintings, and vivid animations.

comparison). This is partially due to a lack of accurate modeling of spatial relations among hand joints. Secondly, transformer-based methods [19, 51, 52, 70, 95] require a large number of tokens for reconstruction, and applying attention to all image tokens does not efficiently model the joint spatial sequences (i.e., the spatial relationship between joints), which often results in an inaccurate 3D hand mesh in real-world scenarios.

To address these challenges, we propose **Hamba**, a novel Mamba-based [26] model that employs graph learning [44, 104] and state space modeling [26] for robust 3D hand mesh reconstruction. Mamba is a new state space modeling method with global receptive field capability. Most Mamba-based models [3, 26, 33, 49, 88, 89, 102] are designed for long-range data, and few studies [48, 78] have adapted Mamba for 3D vision tasks. In this work, we explore Mamba's potential for the 3D hand reconstruction task. We found that directly applying Mamba for 3D hand reconstruction results in inaccurate meshes due to its unidirectional scanning and the lack of specific design for 3D hand reconstruction. To tackle this challenge, we propose a Graph-guided Bidirectional Scan (GBS) to effectively capture the semantic and spatial relation between joints, as shown in Figure 2(c). Besides, the transformer's attention requires calculating correlation among all tokens and introduces unnecessary background correlations, while our proposed GBS uses 88.5% fewer tokens (see Section 3.2 for more details). Secondly, though Mamba-based models [3, 26, 33, 49, 88, 102] excel in modeling long-range sequences, they are not proficient at capturing the local-relation information (in our case, the 'semantics' between hand joints). Since graph learning has the capability



(a) Attention (b) Bidirectional Scan (c) Graph-guided Bidirectional Scan (Ours)

Figure 2: **Motivation**. Visual comparisons of different scanning flows. (a) Attention methods compute the correlation across all patches leading to a very high number of tokens. (b) Bidirectional scans follow two paths, resulting in less complexity. (c) The proposed graph-guided bidirectional scan (GBS) achieves effective state space modeling leveraging graph learning with a few effective tokens (illustrated as scanning by two snakes: forward and backward scanning snakes).

to effectively capture node relations, we integrate graph convolutions into state space modeling, significantly enhancing the representation by considering the intricate hand joint relations.

In particular, to effectively leverage state space modeling (SSM) and graph learning capabilities for 3D hand reconstruction, we first carefully design a Token Sampler (TS) under guidance with hand joints predicted by Joint Regressor (JR), then feed sampled token into the Graph-guided State Space block (GSS) under the Graph-guided Bidirectional Scan (GBS). Lastly, we introduce a fusion module to integrate the state space tokens and global features to further improve performance. As shown in Figure 1, Hamba achieves significant visual performance in challenging scenarios. We summarize our contributions as follows:

- We are the *first* to incorporate graph learning and state space modeling (SSM) for reconstructing robust 3D hand mesh. Our key idea is to reformulate the Mamba scanning into graph-guided bidirectional scanning for 3D reconstruction using a few effective tokens.
- We propose a simple yet effective Graph-guided State Space (GSS) block to capture structured relations between hand joints using graph convolution layers and Mamba blocks.
- We introduce a token sampler that effectively boosts performance. A fusion module is also introduced to further enhance performance by integrating state space tokens and global features.
- Extensive experiments on multiple challenging benchmarks demonstrate Hamba's superiority over current SOTAs, achieving impressive performance for in-the-wild scenarios.

2 Related Works

3D Hand Reconstruction. Multiple approaches have been proposed to reconstruct 3D hand mesh [2, 41, 62, 66, 67, 77, 80, 83, 85], with most works leveraging the MANO [75] parametric representation of the 3D hand. Zhang *et al.* [101] utilized a CNN encoder to iteratively regress the hand mesh based on heatmaps under 2D, 3D, silhouette, and geometric constraints. I2L-MeshNet [63] proposed line pixel-based 1D heatmaps for estimating joint locations and regressing MANO parameters, while HandAR [81] estimated parameters through three stages: joint, mesh, and a refining stage to combine previous features. The joint stage applies a multitask decoder to predict both hand joints and the segmentation mask. MeshGraphormer [52] introduced a graph residual block into the transformer to enhance the spatial structure. HaMeR [70] showed that a simple but large transformer-based architecture trained on a large dataset can achieve SOTA performance. SimpleHand [107] sampled tokens with UV predictions on a high-resolution feature map, cooperating with a cascade upsampling decoder. They further compare different combinations of token generation strategies are compared, including global feature, grid sampling, keypoint sampling, $4\times$ upsampling feature map, and coarse-mesh-guided point sampling. Recently, HHMR [47] proposed a graph diffusion model to learn a prior of gestures and inpaint the occluded hand portion. To further enhance performance, we propose the graph-guided state space model to leverage joint relations and capture spatial joint sequences.

State Space Models (SSMs). State space was originally elaborated in Kalman filtering [39] that described states and transitions with first-order differential equations. Structured State Space Sequence (S4) models [27, 28] have the capability to model dependencies. Recently, Mamba [26] further

https://doi.org/10.52202/079017-0069

improved the S4 models by expanding their fixed projection matrices linearly with the input sequence length. Many recent works have adapted Mamba for visual learning, leveraging its global receptive field and dynamic weights. Liu *et al.* [57] and Yang *et al.* [108] used Mamba for classification, segmentation, and object detection tasks. To effectively capture the spatial relations, they scanned the input image patches forward and backward horizontally. VMamba [57] further added two vertical directions creating a cross-scan. Zhang *et al.* [102] designed a mamba model for motion generation, scanning unidirectionally along the temporal sequence and bidirectionally along channel dimensions in a hierarchy. Behrouz *et al.* [3] and Wang *et al.* [89] designed Graph-mamba to address traditional graph representation learning tasks, enhancing long-range context learning using Mamba blocks. Hamba makes the first attempt to adapt Mamba and graph learning to solve 3D hand reconstruction.

3 Proposed Methodology

We propose a novel Mamba-based method that incorporates graph learning and state space modeling to learn the joint relations from the joint spatial sequence (Figure 3). First, we introduce the concept of state space models (SSMs). Next, we provide the detailed principle of the proposed Token Sampler (TS), Graph-guided Bidirectional Scan (GBS), and Graph-guided State Space (GSS) modules.

3.1 Preliminaries

S6 Models. Selective Scan Structured State Space Sequence (S6) models [26] is a category of sequence models that have demonstrated superior ability in handling sequences. These models are primarily an extension of the previously proposed S4 models [27], which maps an input sequence $x(t) \in \mathbb{R} \rightarrow y(t) \in \mathbb{R}$, through the latent state $h(t) \in \mathbb{R}^N$, following ordinary linear differential equations (Eq. 1), where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{B} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{C} \in \mathbb{R}^{1\times N}$ and $D \in \mathbb{R}^1$ are the weighting parameters.

$$\begin{aligned} h'(t) &= \boldsymbol{A}h(t) + \boldsymbol{B}x(t), \\ y(t) &= \boldsymbol{C}h(t) + Dx(t), \end{aligned} \tag{1}$$

$$\begin{aligned} h_t &= \overline{\boldsymbol{A}}h_{t-1} + \overline{\boldsymbol{B}}x_t, \\ y_t &= \boldsymbol{C}h(t). \end{aligned} \tag{2}$$

For practical computation, these continuous dynamical systems are discretized (Eq. 2). This is achieved by using the zero-order hold (ZOH) discretization rule (Eq. 3).

$$\overline{\boldsymbol{A}} = \exp(\Delta\boldsymbol{A}), \quad \overline{\boldsymbol{B}} = (\Delta\boldsymbol{A})^{-1}(\exp(\Delta\boldsymbol{A}) - \boldsymbol{I}) \cdot \Delta\boldsymbol{B}, \tag{3}$$

where Δ represents the discrete step size. Since both the weighting parameters and discretizing rules are fixed over time, S4 models can be viewed as linear time invariance systems. Mamba [26] further expands S4 models' projection matrices to scan the entire input sequence through a selective scan.

Mamba for Visual Representation. Since Mamba [26] is primarily designed for 1D data, it is challenging to directly apply it to image data with global spatial context and local relation information. Recent works [57, 109] have extended Mamba for learning visual representations. VMamba [57] developed a 2D selective scan (SS2D) block and integrated it into the VSS Block (Figure 4(b)). The VSS block is then stacked consecutively with convolution layers for downsampling image patches via patch merging [58]. The main difference between Mamba and VSS [57] block (Figure 4(a-b)) is replacing the S6 block with SS2D to adapt selective scanning for image data.

3.2 Hamba

Problem Formulation. Given a single hand image I, our goal is to reconstruct the 3D hand mesh. We learn the mapping function $f(I) = \{\theta, \beta, \pi\}$ that regresses MANO [75] parameters from the image I, where $\theta \in \mathbb{R}^{48}$, $\beta \in \mathbb{R}^{10}$, and $\pi \in \mathbb{R}^3$ represent the pose, shape, and camera parameters, respectively. Finally, the MANO model $\mathcal{M}(\theta, \beta)$ generates the corresponding hand mesh $M \in \mathbb{R}^{778\times 3}$.

Model Architecture. Figure 3 illustrates the Hamba model architecture. First, we feed the hand image $I \in \mathbb{R}^{H\times W\times 3}$ into a ViT [19, 95] backbone to extract tokens $T \in \mathbb{R}^{\frac{H}{16}\times\frac{W}{16}\times 1280}$ where H=256 and W=192. Tokens from the backbone are downsampled from dimensions 1280 to 512 using convolution layers (Conv2D). Second, we sample effective tokens using a Token Sampler (TS), which utilizes the 2D joint locations predicted by a Joints Regressor (JR). These tokens are fed into the Graph-guided State Space (GSS) block, which exploits the joint spatial sequence by modeling its state space using the proposed Graph-guided Bidirectional scan (GBS). Finally, we fuse the GSS tokens with the global mean feature, the sampled tokens, and the 2D joint features via a fusion module. Lastly, the MANO parameters are regressed using MLP.

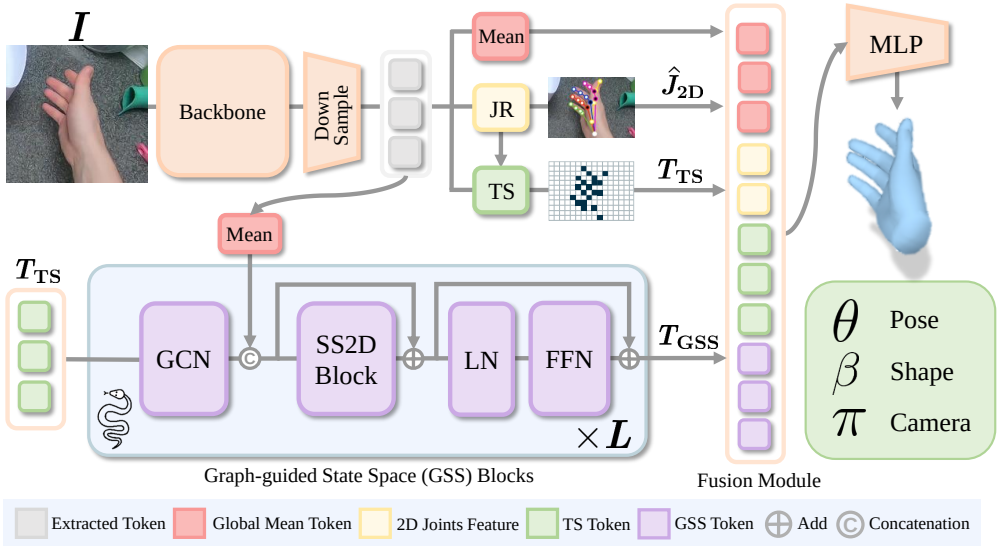


Figure 3: **Overview of Hamba's architecture**. Given a hand image I, tokens are extracted via a trainable backbone model and downsampled. We design a graph-guided SSM as a decoder to regress hand parameters. The hand joints (J_{2D}) are regressed by Joints Regressor (JR) and fed into the Token Sampler (TS) to sample tokens (T_{TS}). The joint spatial sequence tokens (T_{GSS}) are learned by the Graph-guided State Space (GSS) blocks. Inside each GSS block, the GCN network takes T_{TS} as input and its output is concatenated with the mean down-sampled tokens. GSS leverages graph learning and state space modeling to capture the joint spatial relations to achieve robust 3D reconstruction.

Token Sampler (TS) and Joints Regressor (JR). To prevent the GSS Block from being influenced by the background and unnecessary features during the early stages of the training, it is important to select effective tokens that encode the relations between hand joints. We propose a Token Sampler (TS), which selects effective tokens utilizing the initial 2D hand joint prediction from the Joints Regressor (JR). While it is possible to use off-the-shelf 2D joint estimator like OpenPose [5] or MediaPipe [60], this would increase model complexity. Previous works [74, 107] primarily used Conv-Pooling-FC schemes for initial joints regression. In our work, the JR consists of stacked SS2D blocks followed by an MLP head which regresses the initial MANO parameters $\{\hat{\theta}, \hat{\beta}, \hat{\pi}\}$. After the JR regresses 3D joints $\hat{J}_{3D} \in \mathbb{R}^{21\times3}$, these are projected back to the 2D image plane using perspective projection Π with the predicted camera translation $\hat{\pi}$ to obtain $\hat{J}_{2D} \in \mathbb{R}^{21\times2}$. We denote a predefined focal length $F_{\text{focal}} = 5000$ mm. Those are formulated as,

$$\hat{\theta}, \hat{\beta}, \hat{\pi} = \text{JR}(T), \quad \hat{J}_{3D} = \text{MANO}\,(\hat{\theta}, \hat{\beta}), \quad \hat{J}_{2D} = \Pi\,(\hat{J}_{3D}, F_{\text{focal}}, \hat{\pi}). \tag{4}$$

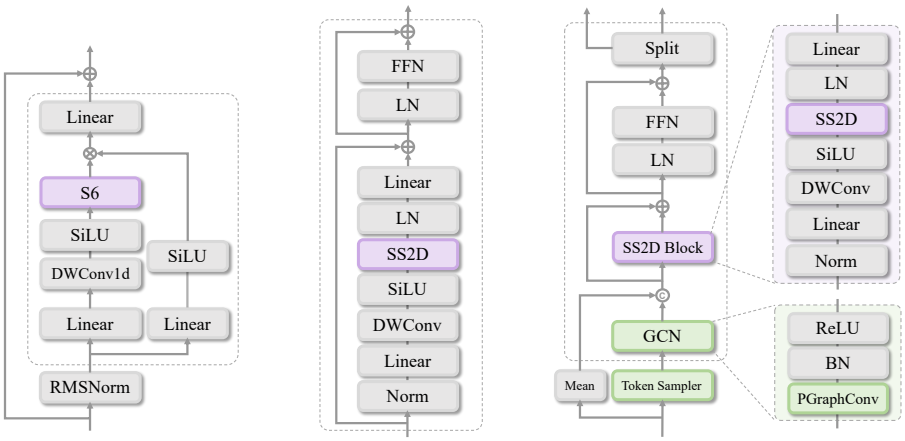
To align the sampled tokens with 2D joints, we use bilinear interpolation. The sampled token $T_{TS} \in \mathbb{R}^{C\times J}$ is formulated as,

$$T_{TS} = \text{TS}(\text{Conv2D}(T), \hat{J}_{2D}), \tag{5}$$

where J denotes the total of 21 joints, and C is the token dimension of 512.

Graph-guided Bidirectional Scan (GBS). To achieve robust reconstruction and leverage effective tokens, we reformulate Mamba's unidirectional scanning as a graph-guided bidirectional scan, thus adapting it for 3D reconstruction tasks. GBS is designed to follow a specific graph pattern, considering the spatial and topological connection of the hand joints with image features. A naive approach would be scanning all tokenized image patches (Figure 2(b)). However, this involves redundant tokens, making it challenging to learn joint spatial relations effectively. To address this, we propose two novel ideas. First, instead of scanning all tokens unidirectionally, we perform hand joint-level bidirectional scanning of sampled tokens T_{TS}. This effectively reduces the number of tokens to be scanned from 192 to 22 ($\approx$ 88.5% reduction). We adapt VMamba [57]'s SS2D block for bidirectional scanning to be suitable for our joint spatial sequence. Second, to capture the local and global joint relations, we introduce a Semantic GCN block [105]. Mamba learns long-range dependencies, but it is less effective at capturing fine-grained local information in intricate structures like the 3D mesh. The

https://doi.org/10.52202/079017-0069



(a) Mamba Block (b) VSS Block (c) GSS Block (Ours)

Figure 4: The illustration of the proposed Graph-guided State Space (GSS) block.

GCN learns input-independent weight matrix to model the edges between hand joints, reflecting how one joint influences another based on prior embedded in graph structures. Introducing graph learning makes it possible to explicitly encode the graph structure within our GSS module. Let $\mathcal{G} = \{\mathbf{V}, \mathbf{E}\}$ be the graph, $\mathbf{V}$ is the set of J nodes and $\mathbf{E}$ are the edges. T_{GCN_l} represents the output of the l-th GCN block, while the complete output of the GSS block is T_{GSS_l}. For a graph-based propagation, we multiply the token with a learnable parameter matrix $\mathbf{W} \in \mathbb{R}^{C\times C}$. Thus, the GCN operation is formulated as,

$$T_{\text{GCN}_l} = \begin{cases} \sigma(\mathbf{W}\, T_{\text{TS}}\, \mathbf{P}_i(\mathbf{M} \odot \mathbf{G})), & l = 1, \\ \sigma(\mathbf{W}\, T^{\{1,..,21\}}_{\text{GSS}_{l-1}}\, \mathbf{P}_i(\mathbf{M} \odot \mathbf{G})), & l > 1, \end{cases} \tag{6}$$

where $\mathbf{M} \in \mathbb{R}^{J\times J}$ is the learnable weighting matrix, $\mathbf{P}_i$ denotes the softmax non-linearity that is applied to normalize the input matrix for all node i choices, while $\mathbf{G} \in [0,1]^{J\times J}$ denotes the adjacency matrix of graph $\mathcal{G}$ and $\odot$ denotes element-wise multiplication. J denotes the total of 21 joints, and C is the token dimension of 512.

Graph-guided State Space (GSS) Block. Overall, our decoder consists of L GSS blocks. The GSS architecture is illustrated comparatively in Figure 4. In the first GSS block, the sampled tokens T_{TS} are passed through graph convolution (GCN) layers. The GCN layer consists of a PGraphConv [44], a Batch Norm, and a ReLU activation. For the GCN, the adjacency matrix is defined based on the hand joint skeleton in the joint order of MANO. To provide the global context, the output from the GCN is concatenated with the global mean token along the joint token sequence. This global mean token is the mean of the downsampled image tokens. This concatenated sequence $T^c_{\text{GCN}_l}$ is then fed into the SS2D block and summed with the output through a residual connection. The SS2D block is followed by a Layer Norm (LN), a Feed-Forward Network (FFN), and another residual connection. For subsequent GSS blocks $l \in \{2,..,L\}$, the input is the output from the previous block $T_{\text{GSS}_{l-1}}$. Before this sequence passes through its GCN layer, it is split, and only the first 21 tokens $T^{\{1,..,21\}}_{\text{GSS}_{l-1}}$ are fed to the GCN. The global mean token $T^{\{22\}}_{\text{GSS}_{l-1}}$ is concatenated back with the GCN's output before it enters the SS2D block as shown in Eq. 7 below:

$$T^c_{\text{GCN}_l} = \begin{cases} T_{\text{GCN}_l} \oplus \text{Mean}(\text{Conv2D}(T)), & l = 1, \\ T_{\text{GCN}_l} \oplus T^{\{22\}}_{\text{GSS}_{l-1}}, & l > 1, \end{cases} \tag{7}$$

$$T_{\text{GSS}_l} = \text{FFN}(\text{LN}(\text{SS2D}(T^c_{\text{GCN}_l}) + T^c_{\text{GCN}_l})) + \text{SS2D}(T^c_{\text{GCN}_l}) + T^c_{\text{GCN}_l}. \tag{8}$$

where $\oplus$ denotes concatenation. The GSS block not only leverages features from state space modeling and graph learning but also considers global features. This design enables Hamba to learn effective features to enhance performance by incorporating state space modeling and graph learning with few tokens, shown in our ablation study Section 4.2.

State Space Modeling for Joint Spatial Sequence. Different from the video-based Mamba models [7, 22, 46], which learns the temporal feature with the frame sequence, Hamba focuses on the joint spatial sequence per frame and reveals that modeling joint relations with Mamba [26] can significantly

improve the 3D reconstruction performance. In particular, as shown in Eq. 1, $x(t)$ represents t-th token of the joint spatial sequence, which is first sampled by the TS using the JR and then encoded with the GCN. Note that t denotes the index of the hand joint iteration. Lastly, $y(t)$ is the updated token of the t-th of the joint spatial sequence after passing through GSS Blocks. The proposed GSS block effectively enhances 3D reconstruction performance by learning the joint spatial sequence relations with graph learning and state space modeling.

Loss Functions. Following [70], we train Hamba using a combined loss which includes 2D joint loss $\mathcal{L}_{\text{2D}}$, 3D joint loss $\mathcal{L}_{\text{3D}}$, pose loss $\mathcal{L}_{\theta}$, shape loss $\mathcal{L}_{\beta}$, and an adversarial loss $\mathcal{L}_{\text{adv}}$. $\mathcal{L}_{\text{2D}}$ and $\mathcal{L}_{\text{3D}}$ are calculated using the L1 Norm, while $\mathcal{L}_{\theta}$ and $\mathcal{L}_{\beta}$ use the L2 Norm. The training loss $\mathcal{L}_{\text{total}}$ is defined as Equation 9, where λ_{2D}, λ_{3D}, λ_{θ}, λ_{β}, and λ_{adv} denote each term's weight respectively.

$$\mathcal{L}_{\text{total}} = \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}} + \lambda_{\theta}\mathcal{L}_{\theta} + \lambda_{\beta}\mathcal{L}_{\beta} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \tag{9}$$

4 Experiments

Datasets. We train Hamba on 2.7M training samples from multiple datasets (same setting as [70] for a fair comparison) that had either both 2D and 3D hand annotations or just 2D annotations. This included FreiHAND [111], HO3D [29], MTC [91], RHD [110], InterHand2.6M [64], H2O3D [29], DexYCB [6], COCO-Wholebody [36], Halpe [21], and MPII NZSL [79] datasets.

Implementation Details. We set learning rate as 10^{-5}, weight decay factor as 10^{-4}, with the 'sum' loss. Weights for each term in the loss function are $\lambda_{3D} = 0.05$ for 3D keypoint loss, $\lambda_{2D} = 0.01$ for 2D keypoint loss, $\lambda_{\theta} = 0.001$ for global orientation and hand pose loss. Weights for beta and adversarial loss, i.e., λ_{β} and λ_{adv} were set as 0.0005. Ablations were run for 60k steps due to computational limitations on 2.7M dataset. Additional details are included in the Appendix A.

Evaluation Metrics. Following the same protocols employed in previous works [52, 70, 107], we used PA-MPJPE and AUC_J as the metrics for evaluating the reconstructed 3D joints and PA-MPVPE, AUC_V, F@5mm, and F@15mm for evaluating the reconstructed 3D mesh vertices.

4.1 Main Results

3D Joints and Mesh Reconstruction Evaluation. We test Hamba on 3 widely used benchmarks: FreiHAND [111], HO3Dv2 [29], and HO3Dv3 [30]. The quantitative comparison with state-of-the-art 3D hand reconstruction models is presented in Table 1, Table 2, and Table 3 respectively. Since almost all previous methods (including the popular MobRecon [9], MeshGraphormer [52], and the recent HHMR [47], SimpleHand [107]) were trained only using the FreiHAND [111], for a fair comparison, we compared them with the Hamba version trained using only the FreiHAND [111] dataset. Meanwhile, for a fair comparison with HaMeR [70], we trained Hamba on the same datasets as HaMeR [70] for all other comparisons. Many methods, including the popular MeshGraphormer [52] and METRO [51], report their metrics using Test-Time Augmentation (TTA) which boosts the final results. We report our performances, both with and without TTA. In both scenarios, Hamba significantly achieves better results, outperforming SOTAs in all benchmarks.

In-the-wild Generalizability Evaluation. Approximately 95% of datasets used for training previous models [9, 14, 47, 51, 52, 70, 86, 107] were collected in controlled indoor environments, such as studios or multi-camera setups. This includes the FreiHAND [111], HO3Dv2 [29], and HO3Dv3 [30] benchmarks that are popularly used for both training and evaluation. However, training models on datasets collected in controlled environments often leads to decreased performance in real-world scenarios. Thus, solely evaluating performance over indoor-collected datasets might not provide a correct evaluation of the robustness of 3D hand reconstruction. We additionally evaluate Hamba's in-the-wild performance on the recently proposed HInt [70] benchmark, which has variations in visual conditions, viewpoints, and hand interactions. Since HInt-NewDays [13] and HInt-EpicKitchensVISOR [16, 17] annotations are 2D keypoints, PCK [97] computed at varying thresholds is used as the evaluation metrics. As shown in Table 5, Hamba outperforms existing models by a large margin and surpasses HaMeR [70], showing improvement in model robustness for in-the-wild scenarios. None of the models (including Hamba) have been trained on/ previously ever seen HInt dataset.

Qualitative Comparison. Figure 5 presents the qualitative comparison of Hamba's 3D hand mesh reconstruction with SOTA models on in-the-wild images from HInt-EpicKitchens. This includes models that directly regress vertices (METRO [51], MeshGraphormer [52]), and parametric methods, which regress MANO parameters (FrankMocap [76], HaMeR [70]). These images are particularly

https://doi.org/10.52202/079017-0069

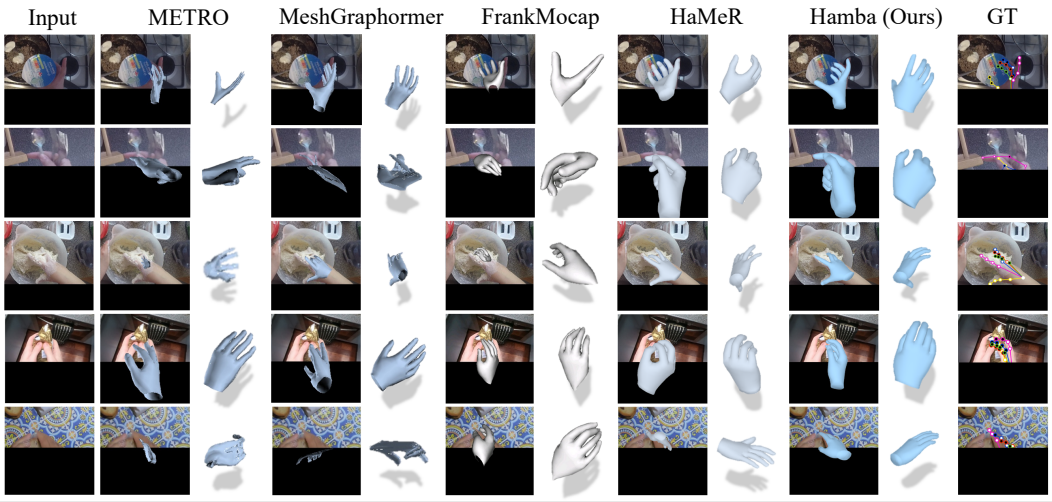


Figure 5: **Qualitative in-the-wild comparison** of the proposed Hamba with SOTAs on HInt-EpicKitchensVISOR [16, 70]. None of the models (including Hamba) have been trained on HInt.

Table 1: Comparison with SOTAs on **FreiHAND** dataset [111]. *Stacked Structure; †used Test-Time Augmentation (TTA). Best scores highlighted Green, while second best are highlighted Light Green. PA-MPJPE and PA-MPVPE are measured in mm. -: Info not reported by model.

Method	Venue	Backbone	PA-MPJPE ↓	PA-MPVPE ↓	F@5mm ↑	F@15mm ↑
Zimmermann *et al.* [111]	ICCV 19	ResNet50	-	10.7	0.529	0.935
Boukhayma *et al.* [4]	CVPR 19	ResNet50	-	13.0	0.435	0.898
ObMan [32]	CVPR 19	ResNet18	-	13.2	0.436	0.908
MobileHand [50]	ICONIP 20	MobileNet	-	13.1	0.439	0.902
YoutubeHand [43]	CVPR 20	ResNet50	8.4	8.6	0.614	0.966
Pose2Mesh [15]	ECCV 20	–	7.7	7.8	0.674	0.969
I2L-MeshNet [63]	ECCV 20	ResNet50*	7.4	7.6	0.681	0.973
HIU-DMTL [100]	ICCV 21	Custom*	7.1	7.3	0.699	0.974
CMR [10]	CVPR 21	ResNet50*	6.9	7.0	0.715	0.977
I2UV-HandNet [8]	ICCV 21	ResNet50	6.7	6.9	0.707	0.977
Tang *et al.* [82]	ICCV 21	ResNet50	6.7	6.7	0.724	0.981
METRO† [51]	CVPR 21	HRNet	6.3	6.5	0.731	0.984
MeshGraphormer†[52]	ICCV 21	HRNet	5.9	6.0	0.764	0.986
MobRecon [9]	CVPR 22	ResNet50*	**5.7**	5.8	0.784	0.986
FastMETRO [14]	ECCV 22	HRNet	6.5	7.1	0.687	0.983
FastViT [86]	ICCV 23	FastViT	6.6	6.7	0.722	0.981
AMVUR [35]	CVPR 23	ResNet50	6.2	6.1	0.767	0.987
Deformer [98]	CVPR 23	HRNet	6.2	6.4	0.743	0.984
PointHMR [42]	CVPR 23	HRNet	6.1	6.6	0.720	0.984
Zhou *et al.* [107]	CVPR 24	FastViT	**5.7**	6.0	0.772	0.986
HaMeR [70]	CVPR 24	ViTPose	6.0	5.7	0.785	0.990
HaMeR-170k [70]	CVPR 24	ViTPose	6.1	5.8	0.782	0.990
HHMR† [47]	CVPR 24	ResNet50	5.8	5.8	-	-
Hamba	**Ours**	ViTPose	5.8	5.5	0.798	0.991
Hamba†	**Ours**	ViTPose	**5.7**	**5.3**	**0.806**	**0.992**

challenging since they comprise real-world cooking videos of a person with highly occluded hands, hand-hand, and/or hand-object interactions. For visual comparison, we select images where the hand lies in the corners, causing a truncation scenario thus increasing the complexity further. Hamba consistently outperforms other models and achieves a much better reconstruction. From Figure 5, we can observe that in severe in-the-wild truncation scenarios, Hamba achieves better hand reconstruction, even though the hand is truncated or occluded. We attribute this performance to effectively learning the spatial hand joint sequence with the state space model. The same is verified in the ablation study presented in Sec. 4.2. Figure S5 presents in-the-wild results on various movies, interviews, etc., scenarios. Figure S6 and Figure S7 presents additional visual results on HInt-NewDays and HInt-EpicKitchensVISOR respectively. Hamba can robustly reconstruct 3D hands in various complicated hand gestures like grasping, holding, grabbing, finger-pointing, and flattening from different viewing directions, even in heavily occluded and truncated scenarios.

https://doi.org/10.52202/079017-0069

Table 2: Comparison with SOTAs on **HO3Dv2** [29] hand-object interaction benchmark.

Method	Venue	PA-MPJPE↓	PA-MPVPE↓	F@5mm↑	F@15mm↑	AUC_J ↑	AUC_V ↑
ObMan [32]	CVPR 19	11.0	11.2	0.464	0.939	0.780	0.777
Pose2Mesh [15]	ECCV 20	12.5	12.7	0.441	0.909	0.754	0.749
I2L-MeshNet [63]	ECCV 20	11.2	13.9	0.409	0.932	0.775	0.722
Hampali *et al*. [29]	CVPR 20	10.7	10.6	0.506	0.942	0.788	0.790
S2Hand [12]	CVPR 21	11.4	11.2	0.450	0.930	0.773	0.777
METRO [51]	CVPR 21	10.4	11.1	0.484	0.946	0.792	0.779
Liu *et al*. [56]	CVPR 21	9.9	9.5	0.528	0.956	0.803	0.810
I2UV-HandNet [8]	ICCV 21	9.9	10.1	0.500	0.943	0.804	0.799
Tse *et al*. [84]	CVPR 22	-	10.9	0.485	0.943	-	-
ArtiBoost [96]	CVPR 22	11.4	10.9	0.488	0.944	0.773	0.782
KPT-Transf [31]	CVPR 22	10.8	-	-	-	0.786	-
MobRecon [9]	CVPR 22	9.2	9.4	0.538	0.957	-	-
HandOccNet [68]	CVPR 22	9.1	8.8	0.564	0.963	0.819	0.819
HFL-Net [54]	CVPR 23	8.9	8.7	0.575	0.965	-	-
H2ONet [94]	CVPR 23	8.5	8.6	0.570	0.966	0.829	0.828
AMVUR [35]	CVPR 23	8.3	8.2	0.608	0.965	0.835	0.836
HOISDF [72]	CVPR 24	9.2	-	-	-	-	-
HandBooster [93]	CVPR 24	8.2	8.4	0.585	0.972	0.836	0.832
HaMeR [70]	CVPR 24	7.7	7.9	0.635	0.980	0.846	0.841
HaMeR-170k [70]	CVPR 24	7.6	7.9	0.639	0.981	0.848	0.843
Hamba	**Ours**	**7.5**	**7.7**	**0.648**	**0.982**	**0.850**	**0.846**

Table 3: Evaluation on **HO3Dv3** [30] benchmark. We only list SOTAs that reported on HO3Dv3.

Method	Venue	PA-MPJPE↓	PA-MPVPE↓	F@5mm↑	F@15mm↑	AUC_J ↑	AUC_V ↑
S^2HAND [12]	CVPR 21	11.5	11.1	0.448	0.932	0.769	0.778
KPT-Transf. [31]	CVPR 22	10.9	-	-	-	0.785	-
ArtiBoost [96]	CVPR 22	10.8	10.4	0.507	0.946	0.785	0.792
Yu *et al*. [99]	BMVC 22	10.8	10.4	-	-	-	-
HandGCAT [90]	ICME 23	9.3	9.1	0.552	0.956	0.814	0.818
AMVUR [35]	CVPR 23	8.7	8.3	0.593	0.964	0.826	0.834
HMP [20]	WACV 24	10.1	-	-	-	-	-
SPMHand [59]	TMM 24	8.8	8.6	0.574	0.962	-	-
Hamba	**Ours**	**6.9**	**6.8**	**0.681**	**0.982**	**0.861**	**0.864**

4.2 Ablation Studies

Effect of Branch-wise Features. We verify the effectiveness of each branch feature by excluding their respective tokens from the fusion module as shown in Table 4. First, we verify the contribution of the proposed GSS branch. When the GSS tokens are excluded (Row 3), we observe a major drop in model performance. Specifically, F@5mm (↑) drops from 0.738 → 0.717, and the PA-MPJPE (↓) and PA-MPVPE (↓) errors increase from 6.6 → 6.9 and 6.3 → 6.6. Thus, in addition to local and global contexts, incorporating structured state-space representations can be effective for 3D hand reconstruction. Moreover, it is important to note that modeling spatial joint sequence relations provides better tokens than directly using the 2D joint locations, even though the latter has a clear semantic meaning for all the hand joints. We attribute this to cases of occlusions where the 2D joints cannot be precisely predicted.

Removing the Token sampler (Row 1) or the 2D joints (Row 2) features also shows a performance drop, but is less significant than removing the GSS branch, since they only provide the local context while GSS tokens provide both local and spatial-relations information. Note that the Global Mean token (Row 4) remains important since it captures the global context, which is discarded in the ablation.

Effect of proposed Components. Since the GSS Block stands as a major contribution, we additionally evaluate the effectiveness of each

Table 4: **Ablation study** on **FreiHAND** [111] to verify proposed components. All variants are trained for same number of steps. PA-MPJPE, PA-MPVPE and without are abbreviated as PJ, PV, 'w/o'.

	Ablation	PJ ↓	PV ↓	F@5 ↑	F@15 ↑
Branch-wise					
1	w/o Token_Sampler_Branch	6.8	6.5	0.722	0.987
2	w/o 2D_Joints_Feature_Branch	6.8	6.6	0.718	0.986
3	w/o GSS_Token_Branch	6.9	6.6	0.717	0.986
4	w/o Global_Mean_Token_Branch	7.3	7.2	0.680	0.982
Component-wise					
5	w/o Token_Sampler	6.8	6.6	0.717	0.986
6	w/o Bidirectional_Scan	6.9	6.6	0.718	0.986
7	w/o GCN	7.3	7.2	0.673	0.983
8	w/o Graph-guided_Bi_Scan	7.3	7.1	0.680	0.983
9	w/o Mamba (SS2D+LN+FFN)	7.3	7.2	0.675	0.983
Hamba (Full)		**6.6**	**6.3**	**0.738**	**0.988**

https://doi.org/10.52202/079017-0069

Table 5: In-the-wild generalizability evaluation on **HInt** [70]. PCK is used as the evaluation metric.

	Method	Venue	NewDays @0.05↑	NewDays @0.1↑	NewDays @0.15↑	VISOR @0.05↑	VISOR @0.1↑	VISOR @0.15↑	Ego4D @0.05↑	Ego4D @0.1↑	Ego4D @0.15↑
All Joints	METRO [51]	CVPR 21	14.7	38.8	57.3	16.8	45.4	65.7	13.2	35.7	54.3
	FrankMocap [76]	ICCVW 21	16.1	41.4	60.2	16.8	45.6	66.2	13.1	36.9	55.8
	MeshGraphormer [52]	ICCV 21	16.8	42.0	59.7	19.1	48.5	67.4	14.6	38.2	56.0
	HandOccNet (param) [68]	CVPR 22	9.1	28.4	47.8	8.1	27.7	49.3	7.7	26.5	47.7
	HandOccNet (no param)	CVPR 22	13.7	39.1	59.3	12.4	38.7	61.8	10.9	35.1	58.9
	HaMeR [70]	CVPR 24	48.0	78.0	88.8	43.0	76.9	89.3	38.9	71.3	84.4
	HaMeR-170k [70]	CVPR 24	46.9	78.6	89.7	44.4	79.3	91.1	37.3	71.6	85.1
	Hamba	**Ours**	**48.7**	**79.2**	**90.0**	**47.2**	**80.2**	**91.2**	**41.7**	**72.9**	**85.5**
Visible Joints	METRO [51]	CVPR 21	19.2	47.6	66.0	19.7	51.9	72.0	15.8	41.7	60.3
	FrankMocap [76]	ICCVW 21	20.1	49.2	67.6	20.4	52.3	71.6	16.3	43.2	62.0
	Mesh Graphormer [52]	ICCV 21	22.3	51.6	68.8	23.6	56.4	74.7	18.4	45.6	63.2
	HandOccNet (param) [68]	CVPR 22	10.2	31.4	51.2	8.5	27.9	49.8	7.3	26.1	48.0
	HandOccNet (no param)	CVPR 22	15.7	43.4	64.0	13.1	39.9	63.2	11.2	36.2	60.3
	HaMeR [70]	CVPR 24	60.8	87.9	94.4	56.6	88.0	94.7	52.0	83.2	91.3
	HaMeR-170k [70]	CVPR 24	58.1	87.8	94.7	57.2	88.7	95.4	49.6	82.5	91.4
	Hamba	**Ours**	**61.2**	**88.4**	**94.9**	**61.4**	**89.6**	**95.6**	**56.0**	**84.3**	**91.9**
Occluded Joints	METRO [51]	CVPR 21	7.0	23.6	42.4	10.2	32.4	53.9	8.1	26.2	44.7
	FrankMocap [76]	ICCVW 21	9.2	28.0	46.9	11.0	33.0	55.0	8.4	26.9	45.1
	MeshGraphormer [52]	ICCV 21	7.9	25.7	44.3	10.9	33.3	54.1	8.3	26.9	44.6
	HandOccNet (param) [68]	CVPR 22	7.2	23.5	42.4	7.4	26.1	46.7	8.0	26.1	45.7
	HandOccNet (no param)	CVPR 22	9.8	31.2	50.8	9.9	33.7	55.4	9.6	31.1	52.7
	HaMeR [70]	CVPR 24	27.2	60.8	78.9	25.9	60.8	80.7	23.0	56.9	76.3
	HaMeR-170k [70]	CVPR 24	**28.9**	62.4	80.5	29.4	65.7	83.9	24.6	58.7	**77.7**
	Hamba	**Ours**	28.2	**62.8**	**81.1**	**29.9**	**66.6**	**84.3**	**25.2**	**59.2**	77.6

component in the proposed GSS block. The same is presented in Table 4. The GSS block models the hand-joint topological structure, learning the graph-structured relations and spatial sequences of joints via graph and state space modeling. Adopting graph learning additionally provides the local context. Excluding the GCN, i.e., when simply using a Mamba block, the structure information will be neglected from the input to the SS2D block, which leads to a large drop in performance (Row 7). This indicates that the GCN is an essential component of the GSS block and using SS2D blocks alone does not lead to accurate 3D hand mesh reconstruction. A potential counter-argument may be that the input features and the GCN alone are sufficient for 3D hand reconstruction, without much improvement from the Mamba Blocks. We removed the Mamba blocks and the GSS degenerates into simple GCN, leading to an equal performance drop (Row 9). Specifically, the PA-MPJPE (↓) and PA-MPVPE (↓) increase from 6.6 → 7.3 and 6.3 → 7.2 respectively, while the F@5mm (↑) and F@15mm (↑) drop from 0.738 → 0.675 and 0.988 → 0.983 respectively. This confirms that both the GCN and the Mamba blocks are equally important in the GSS Block. To verify the effectiveness of the bidirectional scanning, we replaced it with conventional unidirectional scanning to compare, denoted as w/o Bidirectional-scan (Row 6), and the reconstruction error increased. An even larger drop in performance is observed when the proposed GBS scan is removed from the model (Row 8). When not using the token sampler, we also see a drop in performance (Row 5). Overall, Table 4 verifies the effectiveness of each proposed component. We additionally validate this by a qualitative evaluation (in Figure S3), wherein the visual result gets worse when we remove the proposed modules.

5 Conclusion

We propose Hamba, a novel Mamba-based model for 3D hand reconstruction, which is capable of reconstructing robust 3D hand meshes with graph learning and state space modeling under bidirectional scanning. Our key insight is reformulating the Mamba scanning into graph-guided bidirectional scanning using a few effective tokens. This allows us to leverage the relations between hand joints and joint spatial sequences, addressing the occlusion and truncation problems using graph learning and state space modeling. Specifically, we designed a new GSS block to capture the relation between hand joints using graph convolution layers and Mamba blocks. Finally, we introduce a practical fusion module to boost performance by incorporating state space features and global features. Experiments on challenging benchmarks and in-the-wild tests demonstrate that Hamba outperforms all existing SOTA models.

Limitations. Although we leverage the strong representation capability from the graph-guided Mamba model and train on the large comprehensive datasets, it may still not be enough to cover all in-the-wild situations. Our current method lacks the capability to explore temporal features in videos because crawling video datasets requires extensive manual labor for 3D hand reconstruction.

Broader Impacts. Our research focuses on the Hamba model for 3D hand reconstruction, and we plan to release the pre-trained models and code. However, there is a potential risk that it could be used for unauthorized surveillance or privacy infringements.

Acknowledgments

Aviral Chharia was supported in part by the ATK-Nick G. Vlahakis Graduate Fellowship from Carnegie Mellon University, USA. The authors would like to thank Bernhard Kerbl, Ce Zheng, Cheng Zhang, Yunlu Chen, and Zhenyu Xie for providing suggestions and feedback to improve this work.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 640–653. Springer, 2012.

[3] Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. *arXiv preprint arXiv:2402.08678*, 2024.

[4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[5] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[6] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[7] Guo Chen, Yifei Huang, Jilan Xu, Baoqi Pei, Zhe Chen, Zhiqi Li, Jiahao Wang, Kunchang Li, Tong Lu, and Limin Wang. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*, 2024.

[8] Ping Chen, Yujin Chen, Dong Yang, Fangyin Wu, Qin Li, Qingpei Xia, and Yong Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12929–12938, 2021.

[9] Xingyu Chen, Yufeng Liu, Yajiao Dong, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20544–20554, 2022.

[10] Xingyu Chen, Yufeng Liu, Chongyang Ma, Jianlong Chang, Huayan Wang, Tian Chen, Xiaoyan Guo, Pengfei Wan, and Wen Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13283, 2021.

[11] Xingyu Chen, Baoyuan Wang, and Heung-Yeung Shum. Hand avatar: Free-pose hand animation and rendering from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8683–8693, 2023.

[12] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. Model-based 3d hand reconstruction via self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10451–10460, 2021.

 https://doi.org/10.52202/079017-0069

[13] Tianyi Cheng, Dandan Shan, Ayda Hassen, Richard Higgins, and David Fouhey. Towards a richer 2d understanding of hands at scale. *Advances in Neural Information Processing Systems*, 36:30453–30465, 2023.

[14] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

[15] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, pages 769–787. Springer, 2020.

[16] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.

[17] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.

[18] Haoye Dong, Tiange Xiang, Sravan Chittupalli, Jun Liu, and Dong Huang. Physical-space multi-body mesh detection achieved by local alignment and global dense learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1267–1276, 2024.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2020.

[20] Enes Duran, Muhammed Kocabas, Vasileios Choutas, Zicong Fan, and Michael J Black. Hmp: Hand motion priors for pose and shape estimation from video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6353–6363, 2024.

[21] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[22] Yu Gao, Jiancheng Huang, Xiaopeng Sun, Zequn Jie, Yujie Zhong, and Lin Ma. Matten: Video generation with mamba-attention. *arXiv preprint arXiv:2405.03025*, 2024.

[23] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023.

[24] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975.

[25] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[26] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. 2023.

[27] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. 2021.

[28] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.

[29] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020.

[30] Shreyas Hampali, Sayan Deb Sarkar, and Vincent Lepetit. Ho-3d_v3: Improving the accuracy of hand-object annotations of the ho-3d dataset. *arXiv preprint arXiv:2107.00887*, 2021.

[31] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11090–11100, 2022.

[32] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11807–11816, 2019.

[33] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Björn Ommer. Zigma: A dit-style zigzag mamba diffusion model. *arXiv preprint arXiv:2403.13802*, 2024.

[34] Rongtian Huo, Qing Gao, Jing Qi, and Zhaojie Ju. 3d human pose estimation in video for human-computer/robot interaction. In *International Conference on Intelligent Robotics and Applications*, pages 176–187. Springer, 2023.

[35] Zheheng Jiang, Hossein Rahmani, Sue Black, and Bryan M Williams. A probabilistic attention model with occlusion-aware texture regression for 3d hand reconstruction from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 758–767, 2023.

[36] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 196–214. Springer, 2020.

[37] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR 2011*, pages 1465–1472. IEEE, 2011.

[38] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[39] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[40] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018.

[41] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2540–2548, 2015.

[42] Jeonghwan Kim, Mi-Gyeong Gwon, Hyunwoo Park, Hyukmin Kwon, Gi-Mun Um, and Wonjun Kim. Sampling is matter: Point-guided 3d human mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12880–12889, 2023.

[43] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4990–5000, 2020.

 https://doi.org/10.52202/079017-0069

[44] John Boaz Lee, Ryan A. Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. Higher-order graph convolutional networks. 2018.

[45] Haoyuan Li, Haoye Dong, Hanchao Jia, Dong Huang, Michael C Kampffmeyer, Liang Lin, and Xiaodan Liang. Coordinate transformer: Achieving single-stage multi-person mesh recovery from videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8744–8753, 2023.

[46] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024.

[47] Mengcheng Li, Hongwen Zhang, Yuxiang Zhang, Ruizhi Shao, Tao Yu, and Yebin Liu. Hhmr: Holistic hand mesh recovery by enhancing the multimodal controllability of graph diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[48] Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024.

[49] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

[50] Guan Ming Lim, Prayook Jatesiktat, and Wei Tech Ang. Mobilehand: Real-time 3d hand shape and pose estimation from color image. In *International Conference on Neural Information Processing*, pages 450–459. Springer, 2020.

[51] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[52] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12939–12948, 2021.

[53] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[54] Zhifeng Lin, Changxing Ding, Huan Yao, Zengsheng Kuang, and Shaoli Huang. Harmonious feature learning for interactive hand-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12989–12998, 2023.

[55] Hongyu Liu, Xintong Han, Chengbin Jin, Lihui Qian, Huawei Wei, Zhe Lin, Faqiang Wang, Haoye Dong, Yibing Song, Jia Xu, et al. Human motionformer: Transferring human motions with vision transformers. *arXiv preprint arXiv:2302.11306*, 2023.

[56] Shaowei Liu, Hanwen Jiang, Jiarui Xu, Sifei Liu, and Xiaolong Wang. Semi-supervised 3d hand-object poses estimation with interactions in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14687–14697, 2021.

[57] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. 2024.

[58] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[59] Haofan Lu, Shuiping Gou, and Ruimin Li. Spmhand: Segmentation-guided progressive multi-path 3d hand pose and shape estimation. *IEEE Transactions on Multimedia*, 2024.

[60] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. Mediapipe: A framework for building perception pipelines. 2019.

[61] Rachel Locker McKee and David McKee. Making an online dictionary of new zealand sign language: projects. *Lexikos*, 23(1):500–531, 2013.

[62] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 184–184, 2013.

[63] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *European Conference on Computer Vision*, pages 752–768. Springer, 2020.

[64] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020.

[65] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020.

[66] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3316–3324, 2015.

[67] Iason Oikonomidis, Nikolaos Kyriazis, Antonis A Argyros, et al. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011.

[68] JoonKyu Park, Yeonguk Oh, Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1496–1505, 2022.

[69] Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Dinh-Tuan Tran, Xavier Baro, Hugo Jair Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research*, 24(198):1–6, 2023.

[70] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *CVPR*, 2024.

[71] Siyou Pei, Alexander Chen, Jaewook Lee, and Yang Zhang. Hand interfaces: Using hands to imitate objects in ar/vr for expressive interactions. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–16, 2022.

[72] Haozhe Qi, Chen Zhao, Mathieu Salzmann, and Alexander Mathis. Hoisdf: Constraining 3d hand-object pose estimation with global signed distance fields. *arXiv preprint arXiv:2402.17062*, 2024.

[73] Lihui Qian, Xintong Han, Faqiang Wang, Hongyu Liu, Haoye Dong, Zhiwen Li, Huawei Wei, Zhe Lin, and Cheng-Bin Jin. Xformer: fast and accurate monocular 3d body capture. *arXiv preprint arXiv:2305.11101*, 2023.

[74] Pengfei Ren, Chao Wen, Xiaozheng Zheng, Zhou Xue, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

 https://doi.org/10.52202/079017-0069

[75] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 2017.

[76] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *IEEE International Conference on Computer Vision Workshops*, 2021.

[77] Tanner Schmidt, Richard A Newcombe, and Dieter Fox. Dart: Dense articulated real-time tracking. In *Robotics: Science and systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[78] Qiuhong Shen, Xuanyu Yi, Zike Wu, Pan Zhou, Hanwang Zhang, Shuicheng Yan, and Xinchao Wang. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*, 2024.

[79] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[80] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE international conference on computer vision*, pages 2456–2463, 2013.

[81] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *International Conference on Computer Vision (ICCV)*, pages 11698–11707, 2021.

[82] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11698–11707, 2021.

[83] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (ToG)*, 35(6):1–11, 2016.

[84] Tze Ho Elden Tse, Kwang In Kim, Ales Leonardis, and Hyung Jin Chang. Collaborative learning for hand and object reconstruction with attention-guided graph convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1664–1674, 2022.

[85] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118:172–193, 2016.

[86] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5785–5795, 2023.

[87] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018.

[88] Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.

[89] Chloe Wang, Oleksii Tsepa, Jun Ma, and Bo Wang. Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. *arXiv preprint arXiv:2402.00789*, 2024.

[90] Shuaibing Wang, Shunli Wang, Dingkang Yang, Mingcheng Li, Ziyun Qian, Liuzhen Su, and Lihua Zhang. Handgcat: Occlusion-robust 3d hand mesh reconstruction from monocular images. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2495–2500. IEEE, 2023.

[91] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10974, 2019.

[92] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[93] Hao Xu, Haipeng Li, Yinqiao Wang, Shuaicheng Liu, and Chi-Wing Fu. Handbooster: Boosting 3d hand-mesh reconstruction by conditional synthesis and sampling of hand-object interactions. *arXiv preprint arXiv:2403.18575*, 2024.

[94] Hao Xu, Tianyu Wang, Xiao Tang, and Chi-Wing Fu. H2onet: Hand-occlusion-and-orientation-aware network for real-time 3d hand mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17048–17058, 2023.

[95] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. volume 35, pages 38571–38584, 2022.

[96] Lixin Yang, Kailin Li, Xinyu Zhan, Jun Lv, Wenqiang Xu, Jiefeng Li, and Cewu Lu. Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2750–2760, 2022.

[97] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[98] Yusuke Yoshiyasu. Deformable mesh transformer for 3d human mesh recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17006–17015, 2023.

[99] Ziwei Yu, Linlin Yang, You Xie, Ping Chen, and Angela Yao. Uv-based 3d hand-object reconstruction with grasp optimization. *arXiv preprint arXiv:2211.13429*, 2022.

[100] Xiong Zhang, Hongsheng Huang, Jianchao Tan, Hongmin Xu, Cheng Yang, Guozhu Peng, Lei Wang, and Ji Liu. Hand image understanding via deep multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11281–11292, 2021.

[101] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2354–2364, 2019.

[102] Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation with hierarchical and bidirectional selective ssm. *arXiv preprint arXiv:2403.07487*, 2024.

[103] Fuwei Zhao, Zhenyu Xie, Michael Kampffmeyer, Haoye Dong, Songfang Han, Tianxiang Zheng, Tao Zhang, and Xiaodan Liang. M3d-vton: A monocular-to-3d virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13239–13249, 2021.

[104] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3425–3435, 2019.

[105] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3425–3435, 2019.

[106] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022.

[107] Zhishan Zhou, Zhi Lv, Minqiang Zou, Yao Tang, Jiajun Liang, et al. A simple baseline for efficient hand mesh reconstruction. *arXiv preprint arXiv:2403.01813*, 2024.

[108] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

 https://doi.org/10.52202/079017-0069

[109] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

[110] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.

[111] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. 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*, pages 813–822, 2019.

A Appendix

We further elaborate on the additional model architecture details (Section A.1), and other training details A.2 including loss function weights, dataset descriptions, training schemes, test-time augmentation, and various evaluation metric definitions. We also include visual results for failure cases (Section A.3). In Section A.4 we provide a detailed explanation of the ablation experiments, Table 5 and the algorithm of the GSS block. In Section A.5 we include an additional ablation to compare our proposed model with attention-based models. Finally Section A.6 demonstrates the transferability of our GSS block over the 3D Human Mesh Recovery task.

A.1 Model Architecture Details

Hamba follows an encoder-decoder structure that first tokenizes the input image patches, and then feeds it into the decoder to predict the 3D hand reconstruction results. Architecture details of each component required to reproduce our results are included in this section. We further include the model architecture feature dimensions in Figure S1 for additional clarity.

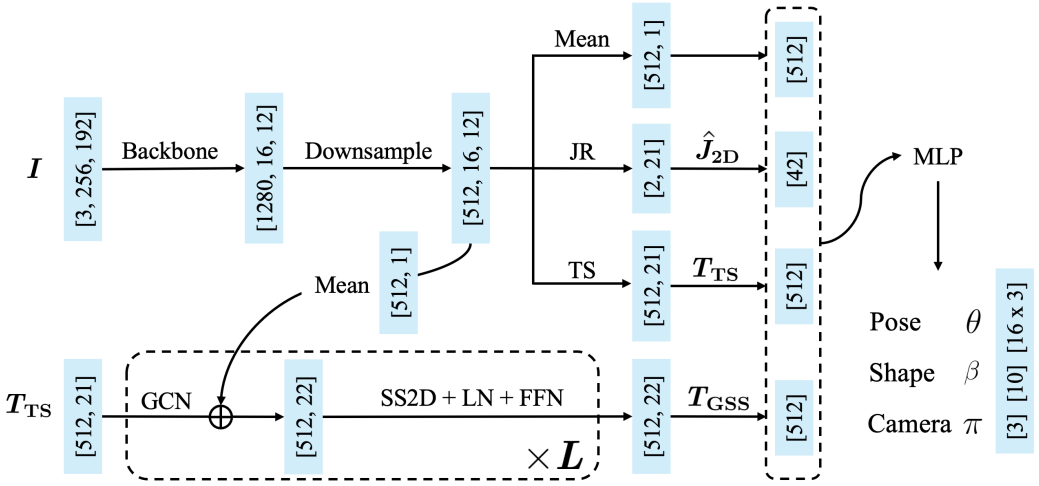


Figure S1: Illustration of model architecture feature dimensions for additional clarity.

Backbone. Following previous works [70, 86, 107], we ViT-H [19] was used as the encoder that inputs an $I \in \mathbb{R}^{256\times192\times3}$ and tokenizes the image patches to output $T \in \mathbb{R}^{16\times12\times1280}$ tokens. Specifically, the backbone contained 50 transformer layers, with a total of 630M parameters. The downsampling layers consist of 2D convolution, Batch Norm, ReLU activation, and another 2D convolution. The channel dimension is reduced from 1280-dim to 512-dim in the first convolution operation, while it is maintained as 512-dim in the second. The kernel sizes for both convolutions were set as one.

Joints Regressor (JR). The JR follows a simple structure, containing only four VSSM (SS2D) blocks followed by 3 linear layers (MLP) to predict each of the MANO parameters. The VSSM blocks are 512-dim with a depth of 2, SSM state dimension of 1, SSM ratio of 2, MLP ratio of 4, and a depth-wise convolution kernel size of 3 (without bias). The Gate control is deprecated, and the JR takes all $T \in \mathbb{R}^{16\times12\times1280}$ tokens input. The final output of the VSSM blocks maintains the same shape as the input. 'Mean' Pooling is performed over the height and width dimensions and then fed into the linear layers (MLP) to regress the MANO parameters $(\hat{\theta}, \hat{\beta}, \hat{\pi})$.

Token Sampler (TS). The TS was implemented as the `pytorch.nn.Functional.grid_sample` module of PyTorch. The TS is followed by a 1D Conv, Batch Norm, ReLU activation, and another 1D Conv operation. Bilinear Interpolation was used as the sampling mode to better adapt the float-point joint location prediction from the JR.

Graph-guided State Space (GSS) Block. The GSS Block has a simple structure. Note that each GCN layer in a GSS block consists of a single graph convolution operation [44] followed by a

https://doi.org/10.52202/079017-0069

Batch Norm and ReLU activation. For the rest of the components of the GSS blocks, each has a set dimension of 512-dim, a depth of 2, an SSM state dimension of 1, an SSM ratio of 2, an MLP ratio of 4, and a depth-wise convolution kernel size of 3 (without bias). The gate control is deprecated. We use 04 GSS blocks in our best-trained model.

A.2 Model Training and other details

Loss Functions. As described by Equation 9, the L1 Norm is used to calculate the 3D joint loss and the 2D joint reprojection loss. Let $J_{\text{2D}}^{\text{GT}}$ and $J_{\text{3D}}^{\text{GT}}$ be the 2D and 3D ground-truth (GT) joint locations, while J_{2D} and J_{3D} be the corresponding model predictions. If a training dataset additionally provides the GT MANO parameters, i.e., θ^{GT} and β^{GT} with their dataset, we also calculate a MANO parameter loss ($\mathcal{L}_\theta, \mathcal{L}_\beta$) between prediction ($\theta, \beta$) and the GT ($\theta^{\text{GT}}, \beta^{\text{GT}}$) using the L2 Norm. The loss terms are formulated as:

$$\begin{aligned} \mathcal{L}_{\text{3D}} &= ||J_{\text{3D}}^{\text{GT}} - J_{\text{3D}}||_1, \\ \mathcal{L}_{\text{2D}} &= ||J_{\text{2D}}^{\text{GT}} - J_{\text{2D}}||_1, \end{aligned} \tag{10}$$

$$\begin{aligned} \mathcal{L}_\theta &= ||\theta^{\text{GT}} - \theta||_2, \\ \mathcal{L}_\beta &= ||\beta^{\text{GT}} - \beta||_2, \end{aligned} \tag{11}$$

To prevent the model from predicting unnatural hand gestures, we cooperate with discriminators D_k for (θ, β) and each hand joint angle separately following [23, 40, 70] as: $\mathcal{L}_{adv} = \sum_k (D_k(\theta, \beta) - 1)^2$. The loss weights are kept as $\lambda_{\text{3D}} = 0.05, \lambda_{\text{2D}} = 0.01, \lambda_\theta = 0.001, \lambda_\beta = 0.0005$ and $\lambda_{\text{adv}} = 0.0005$.

Datasets. In this section, we briefly introduce the datasets used in the study. For training, we use a mixture of FreiHAND [111], HO3D [29], MTC [91], RHD [110], Interhand2.6M [64], H2O3D [29], DexYCB [6], COCO Wholebody [36], Halpe [21], and MPII NZSL [79] datasets. The final dataset contained a total of 2.7M training samples. We adopt the same dataset mixing ratios as [70] with sampling weights as 0.25 for FreiHAND and InterHand2.6M, and 0.1 for MTC and COCO-Wholebody. The rest were all set to 0.05.

- **FreiHAND** [111] is a large-scale multiview hand dataset with 3D hands annotations, popularly used by 3D hand reconstruction studies. The training set contains 33k samples collected by 08 cameras in a green-screen studio environment, which are further enhanced by replacing the backgrounds, leading to an overall 132k samples. The evaluation set contains 4K samples including both in-door and out-door in-the-wild scenarios. FreiHAND was released by Adobe Research and University of Freiburg in 2019, and is for research only, non-commercial use.
- **HO3Dv2** [29] and **HO3Dv3** [31] datasets are part of an ongoing international competition on 3D Hand Reconstruction. Both HO3Dv2 and HO3Dv3 are markerless hand-object interaction datasets containing 3D poses for both hand and object, released in 2020 and 2022 respectively. The sequences in the dataset came from the YCB dataset [92] and were collected by single or multiple RGBD cameras. Currently, the dataset has two versions: $v2$ and $v3$. The $v2$ version contains about 70k training images and around 10k test images, while the $v3$ version contains more than 103K training and 20K test images. The authors do not release the GT annotations and the results can only be tested using the Codalab [69] competition website. Hamba achieves the best performance (as of May 2024) on the leaderboard of both datasets. HO3D was released by Graz University of Technology and CNRS France, and is for research only, non-commercial use.
- **HInt-EpicKitchensVISOR** [17, 70], **HInt-NewDays** [70] and **HInt-Ego4D** [25] datasets consists of 40.4k hands with 2D keypoints, and was released in 2024. HInt stands for Hand Interactions in the Wild, which only has 2D labels and visibility labels. The three subsets of HInt are built based on the Hands23 [13], Epic-Kitchens [16], and the Ego4D [25] video datasets. In our study, HInt is not used for training but serves as a benchmark to evaluate Hamba's cross-dataset generalizability. This dataset is for research only and non-commercial use.

In addition to the above-discussed datasets, we include the descriptions of the datasets used in the Hamba training set:

- **H2O3D** [31] is first released in 2022. Similar to HO3D, it contains around 61k training images and 9k test images using a five RGBD camera multi-view setup in a controlled environment. Compared to HO3D, the content is further expanded to two hands interacting with different objects.
- **DexYCB** [6] is a hand-manipulating object dataset captured by 8 RGBD cameras in a laboratory environment. It contains 582k RGBD images for 10 subjects grasping 20 different objects. Each of the sequences lasts for 3 seconds. The dataset comes with the 2D and 3D annotation for hand.

- **MTC** [91] used the Panoptic Studio to capture both the 3D body and hand poses without using markers. The dataset recorded 40 subjects for 2.5 minutes and included 111K hand images and their 3D pose. Each of the subjects performed a large range of both body and hand motions.
- **RHD** [110] proposed a synthetic dataset for better annotation quality. The dataset utilized 20 different 3D characters performing 39 actions, containing around 44k images in total. Each image is collected under 320×320 resolution and comes with 3D joint annotation. To enhance the generalization of the dataset, the rendered hands are put on random background images.
- **InterHand2.6M** [65] is a large-scale real-captured hand interaction dataset. It contains 2.6M labeled RGB images showing single or two interacting hands. The data was collected using more than 80 cameras in a precisely calibrated multi-view studio and used a semi-automatic annotation method. The dataset involved 27 subjects and the image resolution is set to 512×334.
- **COCO Wholebody** [36] based on the COCO dataset and annotated 133 whole body keypoints including face, hand, body, and feet. It contains 200k RGB in-the-wild images, with 250k instances, and also comes with face, hand, and body bounding boxes. As for the hand pose estimation task, it includes about 100k samples with 2D keypoint annotations.
- **Halpe** [21] is an in-the-wild RGB dataset for full-body human-object interaction dataset. It contains 50k training and 5k test instances. Images are annotated with 136 full body keypoints.
- **MPII NZSL**[79] is a mixing dataset, containing in-the-wild, multiview studio collected and synthetic RGB images. Images are included with single-hand, hand-hand, and hand-object interactions. The dataset has 11k rendered synthetic samples, 2.8k manually annotated in-the-wild images from MPII [1] and NZSL [61], and 15k multiview samples from Panoptic Studio dataset [38]. All the samples come with 2D keypoint annotations.

Training schemes. The Joints Regressor (JR) was trained on a single NVIDIA A4500 GPU with a batch size of 8 for 1M steps. The training took 5 days and required around 300GB RAM. The complete Hamba model was trained on two NVIDIA A6000 GPUs which required two days on a batch size of 56. Early stopping was used after 170k steps to prevent overfitting. The mixing ratio was kept consistent for both parts of the model. `AdamW` optimizer was used with a learning rate of `1e-5`, $\beta_1 = 0.9, \beta_2 = 0.999$, and a weight decay of `1e-4`. The ViT backbone with initialized with HaMeR [70] released checkpoint and is kept unfroze during training. The same setting was used while training the JR. Note that for reproducing the results, we recommend directly loading the released Hamba weights from the GitHub repository. For training from scratch, we recommend using the same training configuration, since PyTorch Lightening might have a bug for effective batch size. Refer to the GitHub issue [2].

Test Time Augmentation (TTA). Since the FreiHAND dataset was originally part of a challenge on 3D Hand Mesh Reconstruction, many popular models used Test-time augmentation (TTA) to report their results. For a fair comparison with the models that used TTA, we report both the results, with and without TTA on the FreiHAND test set in Table 1. Following [52], we let the model infer on the same test image, which was scaled and rotated multiple times and averaged over the results. This generally results in a performance boost. The rotations are set from -90° to 90° for every 10°, and rescaled by factors of (0.7, 0.8, 0.9, 1.0, 1.1). We tested through all the combinations and averaged the error to obtain the TTA result. Note that we obtained the SOTA results both with and without TTA. Unless stated otherwise, all reported results in our study are without using TTA.

Evaluation Metrics. Here, we discuss the definitions of the evaluation metrics used in the study.

- **PA-MPJPE** means Procrustes Aligned-Mean Per Joint Position Error, which is the measure of the average joint error after Procrustes alignment. It is calculated as the Euclidean distance between the predicted joints and the GT. PA-MPJPE performs the Procrustes Analysis (PA) [24] method that aligns the predicted and GT positions using a non-rigid transformation to minimize the overall distance between them. The PA-MPJPE is measured in millimeters (mm) and widely used as the evaluation metrics in 3D hand reconstruction works.
- **PA-MPVPE** means Procrustes Aligned-Mean Per Vertex Position Error, which is similar to PA-MPJPE but measures error between mesh vertices after Procrustes alignment. It is also calculated in millimeters (mm).

[2]DDP with two GPUs doesn't give the same results as one GPU with the same effective batch size #6789 https://github.com/Lightning-AI/pytorch-lightning/issues/6789)

https://doi.org/10.52202/079017-0069

(a) Wrong Palm Orientation (b) Missed Finger due to Motion Blur (c) Incorrect Palm orientation, alignment (d) Wrong hand reconstruction (e) Missed Finger reconstruction

Figure S2: Illustration of various failure cases of Hamba: Wrong palm orientation, missed frames due to motion blur, and missed finger reconstruction.

- **F@5/ F@15mm** is the harmonic mean between recall and precision under a specified distance threshold. F@5 and F@15 represent the F-score with a threshold of 5mm and 15mm respectively.
- **PCK** stands for Percentage of Correct Keypoints. To find PCK, first, the Euclidean distances between the predicted and GT keypoint are normalized by head segment length. When the difference is under a certain threshold, the keypoint is considered to be as correct. By varying the threshold we can draw the curve for PCK that increases along with the threshold.
- **AUC** denotes the Area Under the Curve. In line with previous works [52], we specify AUC for the PCK curve with error thresholds between 0mm and 50mm. AUC for joints and mesh vertices are denoted as AUC_J and AUC_V respectively.

A.3 Failure Cases

Figure S2 presents the various failure cases encountered by Hamba during difficult in-the-wild scenarios. Here, we see that most failure cases occur when the model fails to robustly predict the palm orientation as shown in Figure S2(a, c); or when it misses the finger due to motion blur in videos captured at low FPS like Figure S2(b). Sometimes, the model cannot distinguish the direction where the palm is facing, thus making the prediction rotate for 180. Other cases include failure due to the detector predicting the wrong hand or complex finger gestures, as illustrated in Figure S2(e).

A.4 Additional details

Algorithm of GSS block. To provide a more detailed description of the proposed Graph-guided State Space (GSS) block, we present the processing steps of the algorithm, as illustrated in Algorithm 1.

Additional details of Table 5. Since HInt [70] provides annotations for occluded 2D hand keypoints, we report separate results considering: (i) all joints, (ii) only the joints annotated as visible, and lastly, (iii) only considering joints annotated as occluded in Table 5. We notice that visible joints have a relatively higher PCK compared to all joints which include both the visible and occluded joints; while the occluded joints reconstruction has the lowest PCK. It is important to note that since 3D GT poses cannot be annotated for in-the-wild images, HInt only provides 2D keypoints. Therefore, we reproject the 3D hand mesh back to 2D and use PCK to evaluate the reprojection accuracy.

Ablation Experiments. This section provides a detailed explanation of the network architecture modifications made for the ablation experiments. The ablation results discussed below are presented in Rows 1-9 of Table 4. The first four experiments involve branch-wise ablations, while the remaining five focus on architectural modifications in the component-wise ablations. Including branch-wise experiments helps investigate the contribution at the branch level, while component-wise ablations clarify the improvements brought by the proposed component.

- **w/o Token_Sampler_Branch.** We remove the TS branch's tokens from concatenation with other tokens in the fusion module, but we do not remove the token sampler. Thus the sampled tokens from the TS are still input to the GSS block.
- **w/o 2D_Joints_Feature_Branch.** We removed the 2D joints feature from the JR, which is concatenated with the other tokens in the fusion module. The joint output from the JR is still used as a strong local context for sampling effective tokens.

- **w/o GSS_Token_Branch.** To evaluate the contribution of the GSS branch tokens, we remove the GSS tokens from the fusion module. This helps to evaluate the overall contribution of the GSS block tokens. It is important to note that modeling joint spatial relations provides better tokens than directly using the 2D joint locations. It is confirmed by the larger performance drop observed when the GSS tokens are removed compared to TS tokens, and 2D joints features.
- **w/o Global_Mean_Token_Branch.** We remove the Global Mean Token branch from the model architecture and do not include their tokens in the fusion module. Note that we do not remove the global mean token, which is concatenated with the output of the GCN in the GSS block.
- **w/o Token_Sampler.** We exclude the TS component directly.
- **w/o Bidirectional_Scan.** To verify the effectiveness of the bidirectional scanning, we replaced it with unidirectional scanning.
- **w/o GCN.** We remove the GCN module to verify the effectiveness of GCN. The sampled tokens are directly concatenated with the Global Mean Token and input to the SS2D Block.
- **w/o Graph-guided_Bi_scan.** We shuffled the order of the joint sequence to simulate without graph-guided scanning.
- **w/o Mamba (SS2D+LN+FFN).** To evaluate the contribution of Mamba blocks (SS2D+LN+FFN) in the GSS. We remove the Mamba blocks from the proposed GSS blocks. This includes the SS2D, Layer Norm (LN), and the Feed-Forward Network (FFN). The GSS blocks degenerate into simple GCN blocks. We still concatenate it with the global mean token with the output of the GSS block and split it before feeding it into the next GSS block.

Algorithm 1 Graph-guided State Space (GSS) block

Input: Feature Representation T : (B, C, H, W)
Output: Enhanced Representation T_{GSS_L} : (B, C, J)

/* Token Sample */
$T_{\text{TS}} : (B, C, J') \leftarrow TS(Conv2D(T), \hat{J}_{2D})$
$m : (B, C, 1) \leftarrow Mean(T)$
/* A Set of GSS blocks */
$T_{\text{GSS}_l}^{\{1,..,21\}} \leftarrow T_{\text{TS}}$ when $l = 0$
for l in [1 to L] **do**
  /* Graph Convolutions for Hand Joints*/
  $T_{\text{GCN}_l} : (B, J', C) \leftarrow GCN(T_{\text{GSS}_{l-1}}^{\{1,..,21\}})$
  $z' : (B, C, J) \leftarrow Concat(T_{\text{GCN}_l}, m)$
  /* SS2D Block */
  $z'' : (B, C, J) \leftarrow Linear(Norm(z'))$
  $z''' : (B, C, J) \leftarrow SiLU(DWConv(z''))$
  $z_{\text{SS2D}} : (B, C, J) \leftarrow SS2D(z''')$
  $z'_{\text{SS2D}} : (B, C, J) \leftarrow Linear(LayerNorm(z_{\text{SS2D}}))$
  /* Residual Connection */
  $y' : (B, C, J) \leftarrow Sum(z', z'_{\text{SS2D}})$
  /* FFN after Layer Normalization */
  $y'' : (B, C, J) \leftarrow FFN(LayerNorm(y'))$
  /* Residual Connection */
  $T_{\text{GSS}_l} : (B, C, J) \leftarrow Sum(y', y'')$
  $T_{\text{GSS}_{l-1}}^{\{1,..,21\}}, m \leftarrow Split(T_{\text{GSS}_l})$
end for
/* Enhance feature representation */
$T_{\text{GSS}_L} : (B, C, J) \leftarrow Concat(T_{\text{GSS}_L}^{\{1,..,21\}}, m)$
return T_{GSS_L}

A.5 Ablation study with GCN + Transformer models

We performed an additional ablation study to compare the effect of SS2D block with attention-based method. The GCN + SS2D in Graph-guided state space block is replaced with GCN + Attention borrowed from Graformer [106]. The comparison results are shown in Table S1. Both models are trained with the same dataset setting on a single A6000 GPU for 60K steps, and evaluated on the FreiHand dataset [111]. Our proposed GCN + SS2D model shows improvement in all metrics

https://doi.org/10.52202/079017-0069

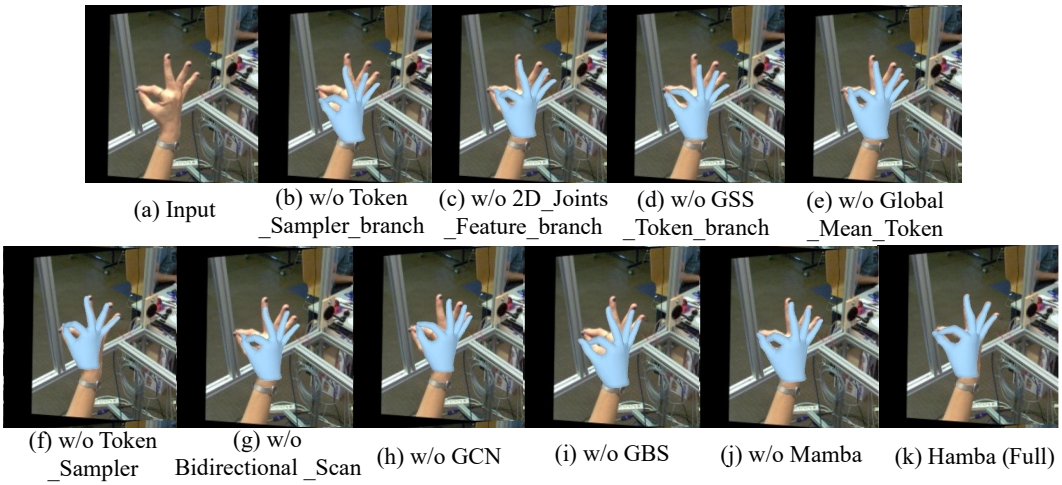


Figure S3: **Qualitative Ablation Study** on FreiHAND [111]. Our full model achieves the best result compared with all the ablation variants.

compared with the GCN + Transformer architecture. This confirms our graph-guided state-space model has better capability to learn the hand joint relationships.

Table S1: Ablation study on the FreiHand dataset [111] to verify the effectiveness of GCN + SS2D block comparing with GCN + attention mechanism. Here, 'w' denotes 'with'.

Method	PA-MPJPE ↓	PA-MPVPE ↓	F@5mm ↑	F@15mm ↑
w GCN + Attention	7.0	6.6	0.730	0.985
w GCN + SS2D (Ours)	**6.6**	**6.3**	**0.738**	**0.988**

Compared to transformer-based models that utilize a large number of tokens for 3D hand reconstruction, our proposed Hamba uses fewer tokens and is 'token-efficient'. We provide the details of Hamba method's efficiency in terms of inference time, FLOPs, and GPU memory usage in Table S2. This shows our model is more lightweight and faster comparing to GCN + Transformer-based models.

Table S2: Comparison of Token Efficiency, Parameters, FLOPS, Runtime and GPU Memory.

Method	Tokens ↓	Parameters (M) ↓				MFLOPs ↓	Runtime (ms) ↓			GPU ↓
		Backbone	JR	Decoder	All	Decoder	Backbone	JR	Decoder	Mem. (MB)
w GCN + Transformer	192	630	27.6	149	782	830	18.7	9.0	21.9	20947
w GCN + SS2D (Ours)	22	630	27.6	71.8	733	649	18.7	9.0	11.8	3413.2
	88.5%↓	-	-	**51.8%↓**	**6%↓**	**21.8%↓**	-	-	**46.1%↓**	**83.7%↓**

A.6 Transfer to 3D Human Mesh Recovery task

To test the transferability of the GSS block acting as a plug-and-play module for other downstream tasks, we adapted Hamba for the 3D human mesh recovery (HMR) task. We trained our model on the same mixing datasets as 4D-humans [23]. Our model achieved comparable performance with 4D-humans (HMR2.0b) as shown in Table S3. Hamba showed improvements on LSP-Extended [37] and COCO datasets [53], as well as achieving comparable results on the 3DPW dataset [87], even though it's trained for fewer steps on a single GPU. The performance of our model may be further improved by increasing training iterations as HMR2.0b [23] under 8 GPU settings. This confirms our proposed module is capable of serving as a plug-and-play component to solve similar or downstream tasks. The visual results for in-the-wild scenarios are shown in Figure S4.

Table S3: Results comparison on the Human Mesh Recovery task on three Benchmark datasets.

Method	Training Settings	LSP-Ext		COCO		3DPW	
		@0.05 ↑	@0.1↑	@0.05↑	@0.1 ↑	MPJPE↓	PA-MPJPE↓
HMR2.0b [23]	8 × A100s, 1M Steps	0.530	0.820	**0.860**	0.960	**81.3**	**54.3**
Hamba (Ours)	1 × A100, 300K Steps	**0.539**	**0.832**	0.856	**0.966**	81.7	54.7

Figure S4: Visual Results of Hamba for Full body Human Reconstruction

https://doi.org/10.52202/079017-0069

Figure S5: **Additional in-the-wild visual results of Hamba**. Hamba achieves significant performance in various in-the-wild scenarios, including truncations, hands interacting with objects or hands, different skin tones, viewpoints, angles, occlusion, and movie scenes.

https://doi.org/10.52202/079017-0069

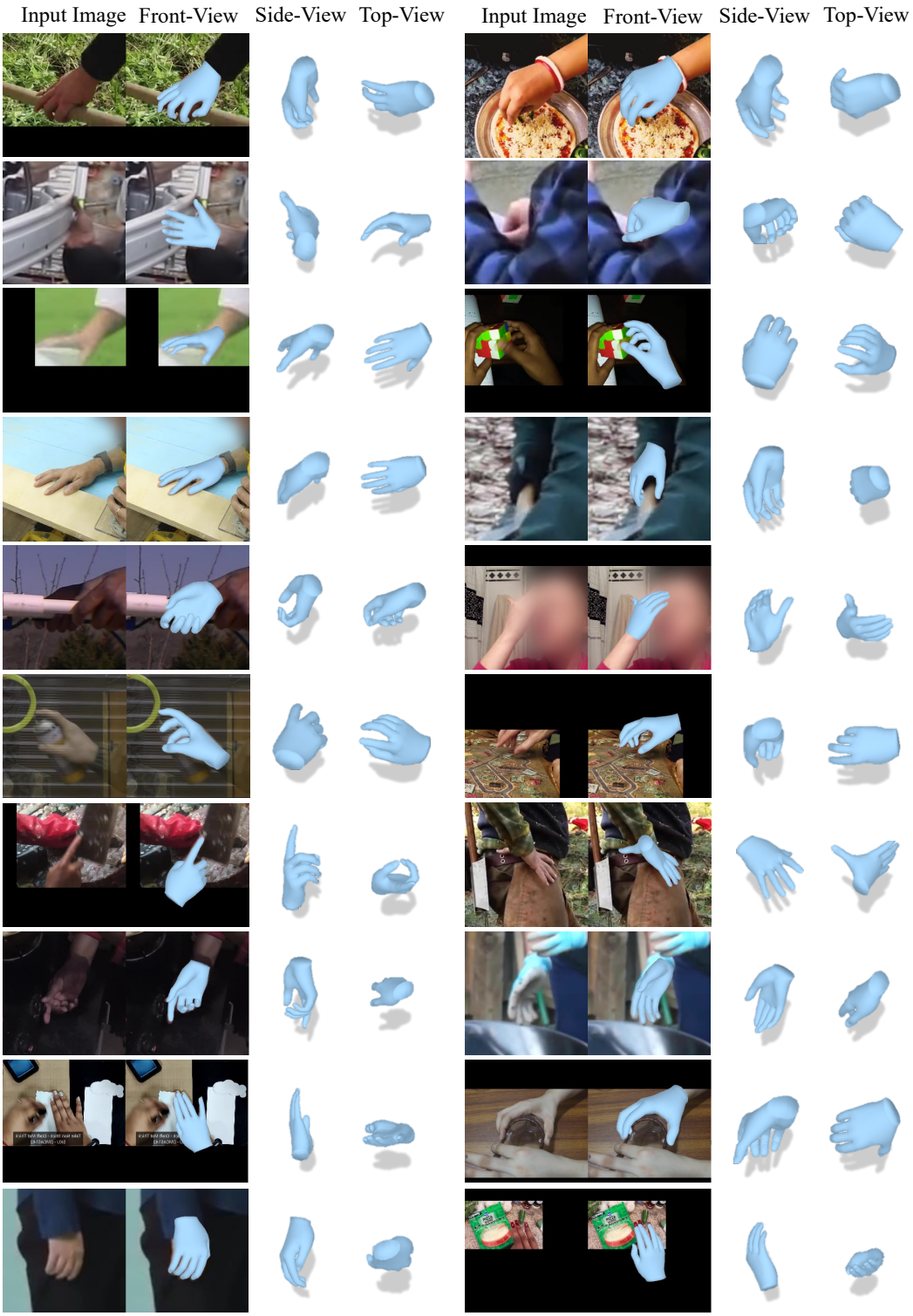


Figure S6: **Qualitative Results on HInt-NewDays** [13, 70]. In-the-wild testing results of Hamba on the HInt-NewDays, which includes highly-occluded hands, hand-hand or hand-object interactions, and truncation scenarios. We did not use the HInt dataset to train Hamba.

https://doi.org/10.52202/079017-0069

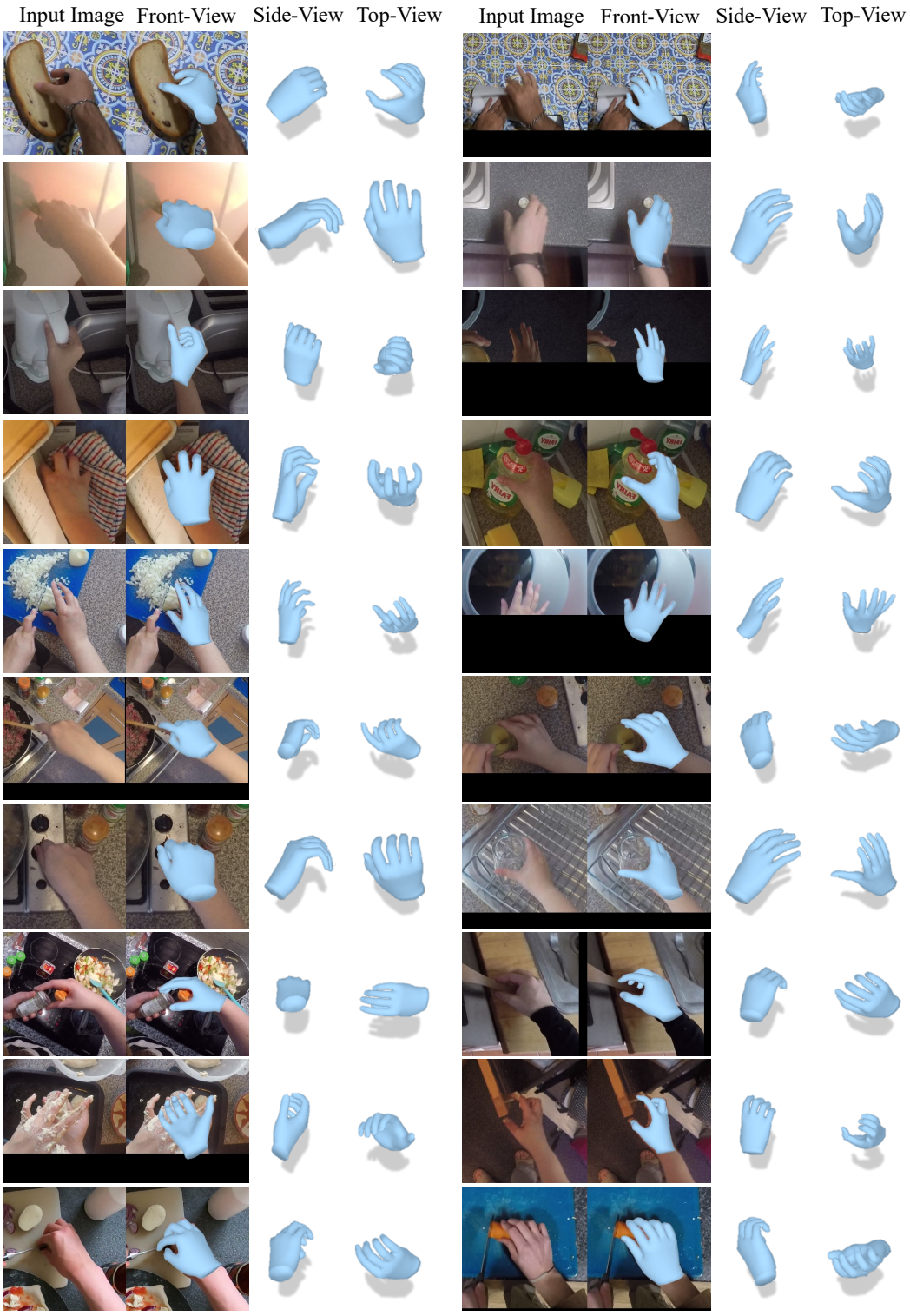


Figure S7: **Qualitative Results on HInt-EpicKitchensVISOR** [16, 70]. In-the-wild testing results of Hamba on the HInt-EpicKitchensVISOR, which includes challenging cooking videos. We did not use the HInt dataset to train Hamba.

https://doi.org/10.52202/079017-0069

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and the contribution section of the Introduction are widely substantiated with 9 ablation studies, including visual comparisons for ablations (see Section 4.2), as well as experiments on 4 popular benchmark datasets.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 5. Although we leverage the strong representation capability from the graph-guided Mamba model and train on the large comprehensive datasets, our experiments do not fully explore temporal features from videos due to the high cost of collecting video datasets for 3D hand reconstruction.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-0069

Answer: [NA]

Justification: We do not propose a theoretical proof. This question is not applicable to our work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in the appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have described our model architecture in detail in Appendix Section A.1. The training schemes have been included in Section A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: Our code was included in the Supplementary .zip file during the NeurIPS review. We will open-source it shortly with a detailed readme on the project's Github repository.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental settings including the data splits, hyperparameters, optimizers, etc., are included in the Experiments section of the manuscript under Experimental settings. Additional model settings and training details are included in the Appendix section Section A.1 and Section A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We test on large datasets. Some of them restrict access to test sets and we are only able to evaluate through their website competition, which will not report the error for each sample. This makes it impractical even impossible to conduct the error analysis. The same methodology was followed by the previous works on the same topic.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

https://doi.org/10.52202/079017-0069

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification:The type of GPU, RAM, memory, compute workers, and other compute-related parameters have been included in the Supplementary Section A.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have included both the potential positive and negative social impacts of the work along with the limitations. We will release the pre-trained models and source code. It may be used for unwarranted surveillance or privacy violations.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We only utilized public datasets and follow their Terms of use. No new data was introduced in our study. Our model does not have a high risk of being misused.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We acknowledged all papers that produced codes and datasets that we used in the paper. The licenses of datasets are provided in the Appendix Section A.2.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-0069

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our study does not involve the collection of new datasets containing human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.