

TokenHand: Discrete Token Representation for Efficient Hand Mesh Reconstruction

Xinguo He^{1,2} Yixin Shen¹ Rahul Chaudhari^{1,2}

¹Chair of Media Technology, Technical University of Munich

²Munich Institute of Robotics and Machine Intelligence, Technical University of Munich

{xinguo.he, yixin.shen, rahul.chaudhari}@tum.de

Abstract

Hand mesh reconstruction has attracted growing attention in recent years. Despite significant progress, existing methods often struggle to balance reconstruction quality and inference efficiency. In this work, we propose TokenHand, a novel framework for single-view 3D hand mesh reconstruction that achieves both high accuracy and real-time inference. Our method represents a 3D hand model using M discrete tokens, each describing a specific sub-structure of the hand. This compositional representation enables efficient modeling with minimal reconstruction error. Furthermore, we reformulate hand mesh reconstruction as a classification problem rather than a regression task. Specifically, a classifier predicts the categories of the M tokens from an input image, and a pre-trained decoder network subsequently reconstructs the 3D hand mesh from the predicted tokens without any post-processing. Extensive experiments demonstrate that TokenHand achieves comparable or superior performance to existing methods across standard benchmarks, while maintaining high efficiency in practical scenarios.

1. Introduction

Single-view hand mesh reconstruction has been extensively studied due to its broad applicability in areas such as augmented reality [16, 40], behavior understanding [22, 35], imitation-based robotic learning [17, 26], etc. These applications require the reconstruction to be accurate, robust, and computationally efficient for real-time operation. Although recent approaches have achieved remarkable progress, it remains challenging to satisfy all these requirements simultaneously, especially in the presence of occlusions.

Existing image-based 3D hand reconstruction methods can be broadly categorized into two groups. The first group leverages a parametric hand model, such as MANO [38], as a strong prior [30, 36, 43, 53, 54]. These methods extract

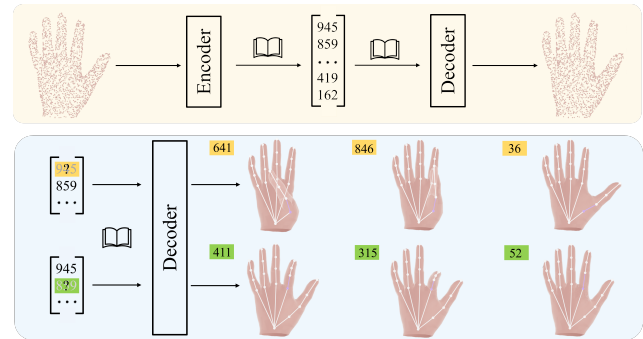


Figure 1. Our approach represents a hand model using M discrete tokens, each corresponding to an index in the learned codebook (top). Each token encodes a specific sub-structure of the hand. In each row, changing the value of a single token consistently modifies the same sub-structure (bottom).

image features through deep neural networks and regress the corresponding shape and pose parameters of the MANO model. Benefiting from the incorporated prior, this category can generate visually plausible 3D hand meshes. However, due to the kinematic chain structure of MANO, errors in pose parameters tend to accumulate hierarchically along the chain. As a result, even a minor inaccuracy in the wrist joint may lead to a significant displacement at the fingertips, making the overall reconstruction highly sensitive to local errors. The second group directly regresses the 3D hand mesh or point cloud from the input image [8, 9, 14, 25]. Typically, these methods first extract localized image features guided by hand joints or keypoints and then perform 2D-to-3D lifting to progressively recover the mesh from a coarse representation to the original resolution. Such direct regression approaches are computationally efficient and allow each vertex to be predicted independently, avoiding inter-joint dependency. Nevertheless, the absence of a parametric prior often results in incomplete geometries or artifacts such as finger fusion and surface discontinuities.

We aim to combine the advantages of the two aforemen-

tioned paradigms by designing a framework that incorporates a strong prior to ensure a plausible hand shape, while maintaining structural independence among local components. To this end, we propose TokenHand, whose core idea is to learn the dependency among hand point cloud distributions in the representation stage.

A straightforward approach would be to collect a large set of realistic 3D hand model prototypes and regress the hand mesh by matching the most similar prototype. Although this can guarantee the realism of the reconstructed hands, it requires an impractically large number of prototypes to cover the entire pose space, which makes it computationally infeasible. Inspired by VQ-VAE [42], we instead introduce a discrete representation to model the space of 3D hand configurations.

Our representation consists of two stages. In the first stage, we employ a Point Transformer as the encoder to map a 3D hand point cloud into M token features, each encoding a specific sub-structure of the hand model. These tokens are then quantized using a shared codebook, enabling each hand instance to be represented compactly by M discrete indices. Finally, the quantized features are decoded to reconstruct the 3D hand point cloud. The space spanned by the learned codebook is sufficiently expressive to represent diverse hand geometries and poses. During this stage, the encoder, codebook, and decoder are jointly optimized by minimizing the reconstruction loss, allowing the model to effectively learn a compact and structured latent representation of 3D hands.

In the second stage, we cast 3D hand reconstruction as a classification problem. Given an input image, the network predicts the category of each of the M tokens, which are then decoded to reconstruct the 3D hand. The decoder can be directly adopted from the first stage without additional training, making the overall reconstruction process lightweight and suitable for real-time applications. By representing hand models with discrete tokens, our approach introduces a strong prior over the full pose space, which effectively prevents unrealistic or incomplete reconstructions commonly observed in direct regression methods. Moreover, since each token corresponds to a specific sub-structure of the hand, the reconstruction maintains local independence without affecting the global consistency. In addition, this discrete representation has the potential to facilitate integration with other discrete modalities, such as text-based hand modeling, although this direction is beyond the scope of this work.

Extensive experiments verify the effectiveness of our approach across multiple benchmarks. On both the FreiHAND and DexYCB datasets, our method consistently achieves superior reconstruction accuracy compared with existing techniques. In addition, the proposed framework runs in real time at 65 FPS and delivers a 14% accuracy

improvement over the best real-time baselines, highlighting its efficiency and scalability for practical deployment. Our main contributions can be summarized as follows:

- We introduce a token-based representation for 3D hand modeling, which effectively captures structural dependencies among local hand components through discrete tokens.
- We propose a new reconstruction pipeline that reformulates the conventional regression-based 3D hand reconstruction task as a token classification problem, enabling real-time inference while preserving high reconstruction fidelity.
- Our method achieves state-of-the-art performance on multiple public benchmarks, demonstrating both strong accuracy and efficiency.

2. Related Work

Hand Mesh Reconstruction Hand reconstruction from monocular images can be broadly divided into two major categories. The first category integrates parametric hand models [2–5, 10, 18, 21, 30, 31, 36, 44, 46, 50–54], typically employing MANO [38], and uses neural networks to regress shape and pose coefficients to infer the hand mesh. The pioneering work of [18] first incorporated MANO into a deep network for end-to-end parameter estimation. Other approaches explore an alternative direction by estimating 2.5D heatmaps to localize joints and subsequently regressing the corresponding MANO parameters [20, 32, 33, 45].

The second category follows a vertex regression paradigm, directly estimating the 3D coordinates of mesh vertices [8, 9, 14, 25]. With the rapid development of transformer architectures, several works [13, 23, 28, 29, 37, 48, 55] employ transformer-based designs to enhance the representation capacity and reconstruction accuracy. METRO [51] extracts a global image feature using a CNN. It constructs input tokens by repeatedly concatenating this feature with the 3D coordinates of a template mesh. A multi-layer transformer encoder with progressive dimensionality reduction then regresses the mesh vertex positions. Graph Convolutional Neural Networks (GCNNs) [24] are well-suited for modeling local vertex interactions, making them highly effective for mesh reconstruction. Pose2Mesh [12] employs a cascaded GCNN architecture to regress 3D vertices from 2D pose inputs. MeshGraphormer [29] combines transformer and GCNN modules to jointly capture global dependencies and local structural relationships. PointHMR [23] argues that sampling backbone features guided by vertex-relevant points enables better correspondence between features and spatial positions.

Our approach differs from prior work by introducing a discrete token representation that captures local hand sub-structures within a quantized latent space. This formulation enables us to transform mesh reconstruction into a classifi-

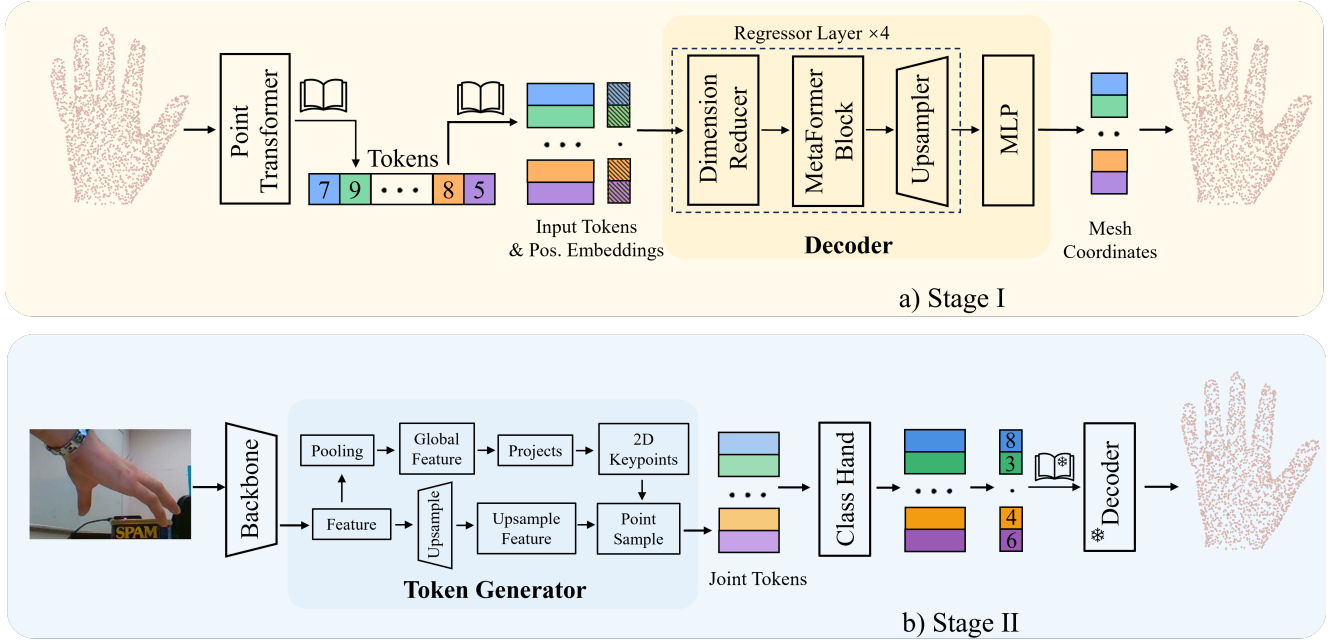


Figure 2. Two stages of the TokenHand representation (a, b). In Stage I, a compositional encoder maps the hand point cloud to M tokens, which are quantized using a codebook. Thus, each hand model is represented as a set of discrete codebook indices. In Stage II, hand reconstruction is formulated as a classification task by predicting the categories of the M tokens, corresponding to the codebook entries. The predicted tokens are then decoded by a decoder network to produce the final hand model.

cation task, supported by a lightweight decoder, resulting in an efficient and scalable solution.

Lightweight Networks Achieving real-time hand mesh reconstruction has motivated the development of various lightweight architectures. FastViT [1] introduces a hybrid vision transformer that attains a strong balance between latency and accuracy by combining structural reparameterization with train-time overparameterization techniques. MoReCon [9] improves efficiency through several specialized components, including a stacked two-dimensional encoding pipeline, a map-based position regression module for two-dimensional to three-dimensional lifting, and a graph operator built on spiral sampling [27]. FastMETRO [11] identifies that the main performance bottleneck in encoder-based transformers arises from the design of input tokens. It adopts an encoder–decoder architecture that separates token interactions, which reduces the overall parameter count. Our method approaches real-time reconstruction from a distinct perspective by reformulating hand mesh estimation as a classification problem over discrete tokens, enabling fast inference and a lightweight architecture.

3. Method

Our method, TokenHand, takes a single-view RGB image as input and reconstructs the corresponding 3D hand mesh. The reconstructed mesh consists of 778 vertices in

3D space, and its topology follows the MANO model. As illustrated in Fig. 2, the proposed framework is composed of two stages. The first stage constructs a token-based representation of 3D hand models, serving as a compact and structured latent prior. The second stage utilizes this learned prior to predict the token categories from the input image, which are then decoded to generate the 3D hand mesh.

3.1. Tokenization

We represent the original 3D hand model as a point cloud $\mathbf{H} \in \mathbb{R}^{778 \times 3}$. An encoder is then trained to transform $\mathbf{H}$ into M token features using a Point Transformer architecture.

$$\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_M) = f_e(\mathbf{H}) \quad (1)$$

where each token feature $\mathbf{t}_i$ approximately corresponds to a specific sub-structure of the 3D hand model. Figure 1 illustrates several examples of the learned token representations. It is worth noting that the learned tokens exhibit a certain degree of redundancy, as different tokens may correspond to overlapping sub-structures. Such redundancy, however, enhances the representational capacity of the token space, allowing for a more comprehensive coverage of diverse hand geometries.

Following [42], we define a latent embedding space represented by a codebook

$$\mathbf{C} = (\mathbf{c}_1, \dots, \mathbf{c}_K)^T \in \mathbb{R}^{K \times D}, \quad (2)$$

where K denotes the number of discrete codebook entries (*i.e.*, the size of the categorical latent space) and D is the dimensionality of each embedding vector $\mathbf{c}_i$. Each token feature $\mathbf{t}_i$ is quantized by performing a nearest-neighbor lookup within this embedding space, as formulated below:

$$q(\mathbf{t}_i = k | \mathbf{H}) = \begin{cases} 1 & \text{if } k = \arg \min_j \|\mathbf{t}_i - \mathbf{c}_j\|_2, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Note that all tokens share a common embedding space $\mathbf{C}$, which facilitates more stable and efficient training.

We slightly abuse the notation $q(\mathbf{t}_i)$ to denote the index of the corresponding entry in the codebook. The decoder then takes the quantized token embeddings $\mathbf{Z} = (\mathbf{c}_{q(t_1)}, \mathbf{c}_{q(t_2)}, \dots, \mathbf{c}_{q(t_M)})$ as input to reconstruct the 3D hand model:

$$\hat{\mathbf{H}} = f_d(\mathbf{Z}). \quad (4)$$

Previous regression-based approaches adopt a multi-stage approximation strategy to model the topological relationship between hand joints and vertices [9, 25]. Inspired by these methods, we construct a cascading upsampling mesh decoder composed of a sequence of regressor layers $\mathbf{R}_n$. Each regressor layer consists of a dimension reduction layer, a MetaFormer block [49], and an upsampling layer. The upsampling layers $\mathbf{U}_n$ and dimension reduction layers $\mathbf{P}_n$ are both implemented as single-layer MLPs. The upsampling layer progressively increases the number of tokens until it matches the total number of hand mesh vertices, while the dimension reduction layer adjusts the feature channel size to the 3D spatial dimension. The MetaFormer block $\mathbf{MF}_n$ serves as the core transformer unit within each stage, where the token mixer is realized using a multi-head self-attention mechanism.

The encoder, codebook, and decoder are jointly optimized by minimizing the following objective function over the training dataset:

$$\ell = w_{rec} \mathcal{L}_1(\hat{\mathbf{H}}, \mathbf{H}) + \beta \sum_{i=1}^M \|\mathbf{t}_i - \text{sg}[\mathbf{c}_{q(t_i)}]\|_2^2, \quad (5)$$

where sg denotes the stop-gradient operation. Here, w_{rec} is the weight for the reconstruction loss, and the hyperparameters are set to $w_{rec} = 10$ and $\beta = 5$. We adopt the optimization strategy proposed in [42] to address the broken gradient problem that arises during the quantization process. Nevertheless, this approach still suffers from codebook collapse, where only a limited subset of codebook entries is effectively utilized. To mitigate this issue, we employ exponential moving average (EMA) updates and a codebook reset (Code Reset) mechanism to ensure more stable training and better codebook utilization.

3.2. 3D HAND Reconstruction

With the learned codebook and decoder, we formulate 3D hand reconstruction as a classification task. As illustrated in Fig. 2, given a cropped input image $\mathbf{I}$, our model first employs a backbone network to extract image features $\mathbf{X}_b$. A token generator then takes $\mathbf{X}_b$ as input and produces tokenized mesh features $\mathbf{X}_m \in \mathbb{R}^{N \times C}$, where N denotes the number of sampled points and C is the feature dimension. These tokenized features are subsequently passed to a classification head, which predicts the discrete categories of the M tokens. The predicted tokens are finally decoded to reconstruct the 3D hand mesh.

Token Generator. Inspired by MobRecon [9], we adopt a keypoint-guided point sampling strategy to extract spatially aligned features from the backbone outputs. Specifically, the backbone network first extracts image features, which are fed into a global pooling layer to predict the 2D hand joint locations. For token generation, the backbone feature maps are upsampled by a factor of 4, and the predicted 2D joint coordinates are used to sample the corresponding spatial positions in the upsampled feature space. The sampled features serve as token features $\mathbf{X}_m$ for the subsequent classification and 3D hand reconstruction stages.

Classification head. We begin by applying two residual convolutional blocks [19] to refine the token features. The resulting features are then flattened and linearly projected to adjust their dimensionality:

$$\mathbf{X}_f = \mathcal{L}(\text{Flatten}(\mathcal{C}(\mathbf{X}_m))), \quad (6)$$

where $\mathcal{C}$ and $\mathcal{L}$ denote the feature refinement module and the linear projection layer, respectively. The one-dimensional feature vector is subsequently reshaped into a matrix $\mathbf{X}_f \in \mathbb{R}^{M \times D}$, which is processed by four MLP-Mixer blocks [41] to capture inter-token dependencies and output the classification logits:

$$\hat{\mathbf{L}} = \mathcal{M}(\mathbf{X}_f), \quad (7)$$

where $\hat{\mathbf{L}} \in \mathbb{R}^{M \times K}$ represents the predicted token logits.

Training. We employ two loss terms to optimize the classification head. The first is the cross-entropy loss, formulated as

$$\ell_{cls} = \text{CE}(\hat{\mathbf{L}}, \mathbf{L}), \quad (8)$$

where $\mathbf{L}$ represents the ground-truth token labels, which are derived by encoding the ground-truth 3D hand model using the pretrained encoder.

We further introduce a hand reconstruction loss to minimize the discrepancy between the predicted and ground-truth hand models. To enable gradient propagation from the decoder back to the classification head, we replace the hard token selection with a soft inference scheme defined as

$$\mathbf{S} = \hat{\mathbf{L}} \times \mathbf{C}, \quad (9)$$

where $\mathbf{S} \in \mathbb{R}^{M \times D}$ denotes the linearly interpolated token features. These features are subsequently fed into the pre-trained decoder to obtain the predicted hand mesh $\hat{\mathbf{H}}$.

The reconstruction loss ℓ_{rec} supervises the prediction of vertices, 3D joints, and 2D joints. Both the 2D joint coordinates, denoted as $\mathbf{J}_{2d}$, and the 3D vertices, denoted as $\mathbf{V}_{3d}$, are directly obtained from the model outputs. The 3D joint locations $\mathbf{J}_{3d}$ are computed from the vertices using a fixed regression matrix $\mathbf{J}$:

$$\mathbf{J}_{3d} = \mathbf{J} \times \mathbf{V}_{3d}. \quad (10)$$

Each of these components is optimized using the L1 loss to measure the discrepancy between predictions and ground truth. The individual loss terms for the vertices, 3D joints, and 2D joints, denoted as L_{vert} , $L_{J_{3d}}$, and $L_{J_{2d}}$, are defined as follows:

$$L_{J_{3d}} = \mathcal{L}_1(\hat{\mathbf{J}}_{3d}, \mathbf{J}_{3d}), \quad (11)$$

$$L_{J_{2d}} = \mathcal{L}_1(\hat{\mathbf{J}}_{2d}, \mathbf{J}_{2d}), \quad (12)$$

$$L_{vert} = \mathcal{L}_1(\hat{\mathbf{V}}_{3d}, \mathbf{V}_{3d}). \quad (13)$$

Here, the variables without hats represent the ground-truth values, while those with hats denote the predicted results. The overall reconstruction objective is formulated as

$$\ell_{rec} = w_{3d}L_{J_{3d}} + w_{2d}L_{J_{2d}} + w_{vert}L_{vert}, \quad (14)$$

where the weights w_{3d} , w_{2d} , and w_{vert} are empirically set to 10, 1, and 10, respectively. Since the main focus of our framework is mesh reconstruction, the 2D keypoints primarily serve as auxiliary supervision for point sampling and thus do not require high precision. The final objective function is defined as

$$\ell_{all} = \ell_{cls} + \ell_{rec}, \quad (15)$$

Note that the decoder network is kept frozen during training.

4. Experiments

4.1. Implementation Details

Our method is implemented in PyTorch and trained on a single NVIDIA RTX A5000 GPU. For the tokenization network, the encoder adopts a Point Transformer module, while the decoder consists of four regressor layers. Each regressor layer contains a MetaFormer block with attention-based token mixers. The corresponding numbers of input tokens are [48, 97, 194, 389], and the output token numbers are [97, 194, 389, 778]. The feature dimensions for these layers are set to [256, 128, 64, 32], where the output from the final layer is processed by a fully connected layer to map the features into 3D spatial coordinates. The codebook used for quantization has a size of 512×512 . The network

is trained using the AdamW optimizer with a base learning rate of 6e-3, a weight decay of 0.05, and a batch size of 128 for 200 epochs.

For the 3D hand reconstruction network, we employ FastViT-MA36 [1] as the backbone, initialized with ImageNet-pretrained weights. The model is trained for 300 epochs using the AdamW optimizer with a batch size of 64. The learning rate is initialized to 5e-4 and reduced to 5e-5 after 50 epochs. Data augmentation includes random noise, flipping, rotation, and scaling to improve generalization.

4.2. Datasets

For tokenization training, we utilize multiple datasets, including FreiHAND [56], DexYCB [6], InterHand2.6M [33], OakInk [47], HO3D [15], and ReInterHand [34]. For each dataset, we use the provided MANO parameters to generate corresponding 3D hand point clouds. Since many datasets contain multi-view images of the same hand pose, we retain only one view of each pose for training to avoid redundancy.

For 3D hand reconstruction, our main experiments and analyses are conducted on the FreiHAND dataset. The FreiHAND dataset is used under its official training–testing split, containing 130,240 training samples and 3,960 testing samples. To further evaluate the generalization ability of our method, we also perform experiments on the large-scale DexYCB dataset, following its official S0 splitting protocol, which includes 406,888 training samples and 78,768 testing samples.

4.3. Evaluation Metrics

To evaluate the accuracy of 3D hand reconstruction, we adopt five commonly used metrics: Procrustes-Aligned Mean Per Joint Position Error (PA-MPJPE), Procrustes-Aligned Mean Per Vertex Position Error (PA-MPVPE), Mean Per Joint Position Error (MPJPE), Mean Per Vertex Position Error (MPVPE), and F-Score. The PA-MPJPE and PA-MPVPE are obtained after applying Procrustes alignment between the predicted and ground-truth hand meshes.

4.4. Results

Evaluation on FreiHAND To validate the effectiveness of our proposed method, we follow the experimental settings of previous works [11, 29]. For a fair comparison, we report all performance metrics without Test-Time Augmentation (TTA) and measure the inference speed (FPS) without TensorRT optimization. As shown in Tab. 1, compared with methods that are not designed for real-time performance, such as METRO [51] and MeshGraphormer [29], our approach achieves a substantial improvement in inference speed, with at least a 35 FPS increase, while simultaneously delivering higher accuracy, improving PA-MPJPE by 0.5 mm. For methods that emphasize real-time execu-

Table 1. Comparison with state-of-the-art methods on the FreiHAND dataset. “–” indicates results not reported. The best performance is highlighted in bold. Second best performance is underlined.

Method	Backbone	PA-MPJPE ↓	PA-MPVPE ↓	F@05 ↑	F@15 ↑	FPS
I2L-MeshNet [32]	ResNet50	7.4	7.6	0.681	0.973	72
CMR [8]	ResNet50	6.9	7.0	0.715	0.977	-
I2UV [7]	ResNet50	7.2	7.4	0.682	0.973	-
Tang et al. [40]	ResNet50	6.7	6.7	0.724	0.981	47
MobRecon [9]	DenseStack	6.9	7.2	0.694	0.979	<u>80</u>
METRO [51]	HRNet	6.7	6.8	0.717	0.981	27
MeshGraphomer [29]	HRNet	6.3	6.5	0.738	0.983	24
FastMETRO [11]	HRNet	6.5	7.1	0.687	0.983	28
Deformer [48]	HRNet	6.2	6.4	0.743	0.984	33
PointHMR [23]	HRNet	6.1	6.6	0.720	<u>0.984</u>	33
FastViT [1]	FastViT-MA36	6.6	6.7	0.722	0.981	84
Zhou et al. [55]	HRNet	<u>5.8</u>	<u>6.1</u>	<u>0.766</u>	0.986	33
Ours	FastViT-MA36	5.7	5.9	0.768	0.986	65

Table 2. Comparison of transformer-based approaches.

Method	#Params	PA-MPJPE ↓	PA-MPVPE ↓
METRO [51]	102M	6.7	6.8
MeshGraphormer [29]	98M	6.3	6.5
FastMETRO [11]	25M	6.5	7.1
Ours	3.0M	5.7	5.9

Table 3. Comparison with state-of-the-art methods on the DexYCB dataset. “–” indicates results not reported. The best performance is highlighted in bold. Second best performance is underlined.

Method	PA-MPJPE ↓	PA-MPVPE ↓	MPJPE ↓	MPVPE ↓
METRO [51]	7.0	-	-	-
Spurr et al. [39]	6.8	-	-	-
Liu et al. [31]	6.6	-	-	-
HandOccNet [36]	5.8	5.5	14.0	13.1
MobRecon [9]	6.4	<u>5.6</u>	14.2	13.1
H2ONet [43]	5.7	5.5	14.0	13.0
Zhou et al. [55]	<u>5.5</u>	5.5	<u>12.4</u>	<u>12.1</u>
Ours	5.4	5.5	12.3	12.0

tion, our model attains a stronger accuracy–efficiency balance. Although slightly slower than FastViT [1], TokenHand achieves a notably better accuracy, surpassing it by 0.9 mm in PA-MPJPE. Furthermore, Fig. 3 provides a visual comparison of inference speed. All methods are evaluated on an NVIDIA 2080 Ti GPU with a batch size of one to ensure fairness. Across both real-time and non-real-time approaches, our method consistently delivers the best overall performance. These results demonstrate that our method provides a favorable trade-off between precision and speed.

We also compare the parameter efficiency of our approach with transformer-based methods, as summarized in Tab. 2. The reported parameter counts exclude the backbone components for a fair comparison. Existing transformer-

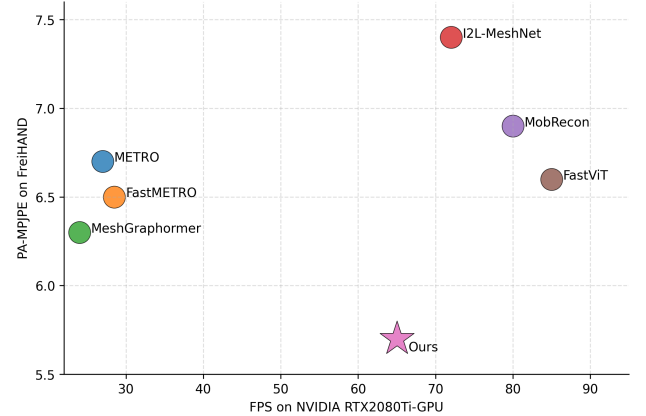


Figure 3. Trade-off between accuracy and inference speed. Our method exceeds non-real-time approaches (≤ 40 FPS) in both speed and precision. Compared with real-time methods (≥ 60 FPS), it achieves noticeably higher accuracy while maintaining comparable inference speed.

based pipelines typically rely on large decoder modules to recover detailed hand geometry. In contrast, our method leverages a learned codebook that provides a strong structural prior, which allows the decoder to remain lightweight. As a result, TokenHand uses only about 10% of the parameters required by prior transformer-based designs, while simultaneously improving accuracy and inference efficiency. The qualitative comparison is presented in Fig. 4. Compared with previous approaches, our method generates more accurate and visually consistent 3D hand models.

Evaluation on DexYCB We evaluate our method on the large-scale hand–object interaction dataset DexYCB to assess its effectiveness and generalization capability. As shown in Tab. 3, our model consistently outperforms ex-

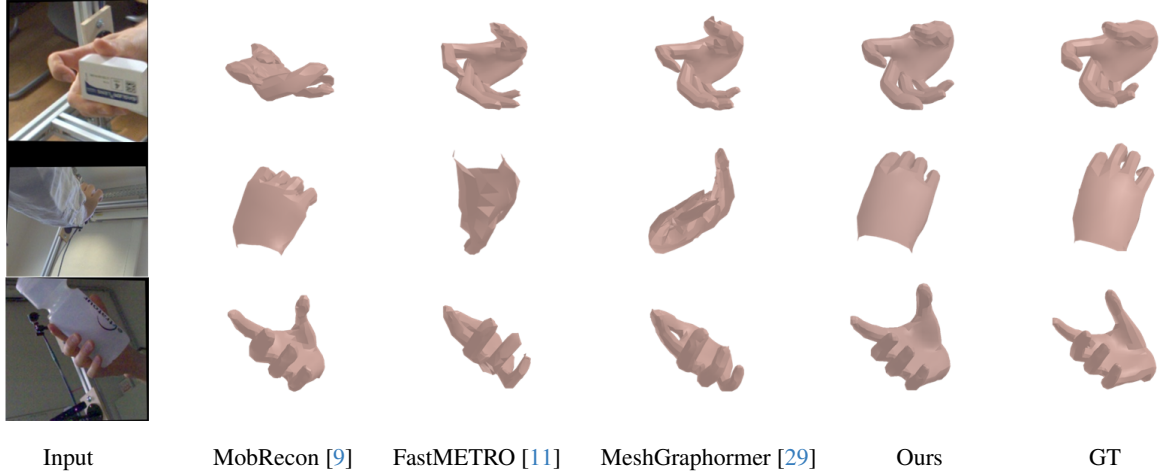


Figure 4. Qualitative comparison between our method and other state-of-the-art approaches on the FreiHAND dataset.

Table 4. Effect of codebook size on reconstruction accuracy.

Codebook	PA-MPJPE ↓	PA-MPVPE ↓
384×512	4.9	5.2
512×512	4.9	5.1
640×512	5.1	5.4

isting single-image-based approaches across all evaluation metrics. In particular, it achieves improvements of 1.7 mm in MPJPE and 1.0 mm in MPVPE over H2ONet [43], establishing a new benchmark and demonstrating the strong generalizability of our approach.

4.5. Ablation Study

To comprehensively evaluate the impact of different design choices, we conduct ablation studies here. Each component of the proposed framework is systematically examined under multiple parameter configurations to assess its contribution to overall performance. All ablation experiments are performed on the FreiHAND test set to ensure consistency and comparability across settings.

Ablation on Codebook size. The effect of codebook size is evaluated in Stage I of TokenHand, as summarized in Tab. 4. For the selected codebook sizes, the reconstruction results remain consistently stable across all configurations. This indicates that the chosen codebook sizes are sufficiently expressive to capture the range of hand shapes and poses present in the datasets.

Ablation on the Core Structure of the Mesh Regressor. To assess the impact of the decoder architecture, we evaluate several variants of the regressor layer, including different numbers of regressor blocks, varying token counts within each block, and changes in feature dimensionality.

As shown in Tab. 5, using a single regressor layer pro-

Table 5. Effect of regressor layers and token counts.

Layer Nums	Token Nums	PA-MPJPE ↓	PA-MPVPE ↓
1	[48]	6.4	6.8
2	[48, 194]	6.1	6.4
2	[48, 389]	6.1	6.3
3	[48, 97, 389]	6.0	6.3
3	[48, 194, 389]	5.9	6.2
4	[48, 97, 194, 389]	5.7	5.9
4	[48, 128, 256, 512]	5.9	6.1

Table 6. Ablation on feature dimensions and the number of MetaFormer blocks in the regressor layer.

Dimensions	PA-MPJPE ↓	PA-MPVPE ↓
256, 128, 64, 32	5.7	5.9
256, 128, 64, 16	5.9	6.2
256, 128, 32, 16	5.9	6.3
256, 64, 32, 16	6.0	6.4
Block Nums	PA-MPJPE ↓	PA-MPVPE ↓
1, 1, 1, 1	5.7	5.9
2, 2, 2, 2	5.8	6.0
3, 3, 3, 3	5.8	6.1

vides limited modeling capacity, and adding even one additional layer leads to a noticeable improvement, reducing PA-MPJPE by approximately 0.3 mm. A consistent trend can be observed as the number of regressor layers increases: performance gradually improves with deeper regressor stacks. The optimal configuration consists of four regressor layers, where the token counts increase progressively by a factor of two at each subsequent layer. We further verify alternative token-number configurations and find that deviating from the progressive $\times 2$ scaling results in slightly degraded performance, indicating that this hierarchical expansion strategy provides a more effective structure for feature aggregation and mesh refinement.

Table 7. Ablation on backbone selection and its impact on reconstruction performance.

Backbone	PA-MPJPE ↓	PA-MPVPE ↓
ResNet50	6.7	6.8
DenseStack	6.7	7.0
HRNet	6.1	6.4
FastViT-MA36	5.7	5.9

The influence of feature dimensionality is evaluated by fixing the token numbers and varying the feature dimensions across regressor layers. As shown in Tab. 6, the configuration that progressively reduces the feature dimensions to 32 yields the best reconstruction performance, while deviations from this setting lead to noticeable degradation. With respect to the number of MetaFormer blocks in each regressor layer, increasing the block count or the computational complexity has minimal impact on performance and can even result in slight deterioration.

Ablation on Backbone Selection. Table 7 summarizes the results obtained with different backbone architectures. Across all tested options, the codebook-based token representation remains consistently effective, producing stable reconstruction quality. When compared with prior methods using the same backbones, TokenHand delivers comparable or better performance, with FastViT-MA36 achieving the strongest overall results.

Ablation on the Core Structure of the Token Generator. To investigate how to most effectively extract token features from the backbone, we evaluate several sampling strategies within the token generator, as illustrated in Fig. 5. Method A applies spatial pooling directly on the backbone feature map to obtain a single global descriptor used as the token feature. Method B partitions the backbone features using grid sampling and assigns each partition as a token feature. Method C adopts the strategy of MobRecon [9] by sampling features at the predicted 2D keypoint locations. Method D increases the spatial resolution of the backbone features through a four-fold upsampling step before performing keypoint-guided sampling. Method E additionally enhances the upsampled features using convolutional layers prior to keypoint-guided sampling. Method F follows the coarse-mesh sampling strategy introduced in FastMETRO [11], sampling features from the upsampled backbone map according to the coarse mesh topology.

The results are summarized in Tab. 8. Using only a global feature yields the lowest performance. At low resolutions, grid sampling and keypoint-based sampling produce comparable results. Increasing the feature-map resolution to 28×28 through a single four-fold upsampling step leads to a noticeable improvement. However, adding additional convolutional layers to the upsampled features does not provide further gains. Similarly, replacing keypoint-

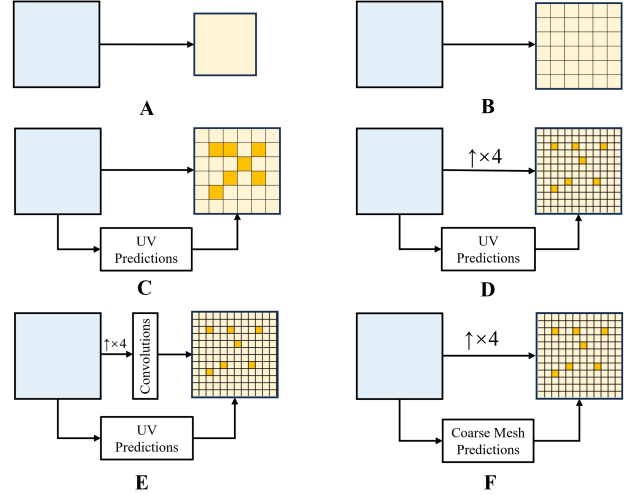


Figure 5. An illustration demonstrates various designs of token generators.

Table 8. Ablation study of sampling strategies and resolution settings used in the token generator.

Sample Method	Resolution	PA-MPJPE ↓	PA-MPVPE ↓
A	1×1	6.2	6.5
B	7×7	6.0	6.3
C	7×7	6.0	6.3
D	28×28	5.7	5.9
E	28×28 enhanced	5.7	5.9
F	28×28	5.8	5.9

based sampling with coarse-mesh sampling offers no improvement. Based on these observations, we adopt the configuration in Method D, which applies upsampling followed by keypoint-guided sampling, as our default design.

5. Conclusion

In this work, we introduced TokenHand, a novel framework that reformulates single-view 3D hand mesh reconstruction through a discrete token representation. By encoding a hand mesh into a compact set of structural tokens learned in a quantized latent space, TokenHand provides an efficient and expressive representation of hand geometry. This compositional design allows reconstruction to be posed as a classification task, with a pre-trained decoder recovering high-quality meshes without post-processing. The proposed design offers a principled way to combine strong geometric priors with efficient inference, substantially reducing model complexity while preserving detailed structural cues. This token-based perspective offers a promising direction for scalable hand modeling and supports real-time AR/VR and future multimodal applications.

Acknowledgements

This work was supported by the China Scholarship Council (CSC) under Grant No. 202206150032.

References

- [1] 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 (ICCV)*, 2023. [3](#), [5](#), [6](#)
- [2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the Envelope for RGB-Based Dense 3D Hand Pose Estimation via Neural Rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)
- [3] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [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 (CVPR)*, 2019.
- [5] Zhe Cao, Ilija Radosavovic, Angjoo Kanazawa, and Jitendra Malik. Reconstructing Hand-Object Interactions in the Wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [6] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A Benchmark for Capturing Hand Grasping of Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [5](#)
- [7] 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 (ICCV)*, 2021. [6](#)
- [8] 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 (CVPR)*, 2021. [1](#), [2](#), [6](#)
- [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 (CVPR)*, 2022. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [10] 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 (CVPR)*, 2021. [2](#)
- [11] 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. [3](#), [5](#), [6](#), [7](#), [8](#)
- [12] 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 (ECCV)*, 2020. [2](#)
- [13] Haoye Dong, Aviral Chharia, Wenbo Gou, Francisco Vicente Carrasco, and Fernando D De la Torre. Hamba: Single-view 3d hand reconstruction with graph-guided bi-scanning mamba. *Advances in Neural Information Processing Systems*, 37:2127–2160, 2024. [2](#)
- [14] Lihao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D Hand Shape and Pose Estimation From a Single RGB Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [1](#), [2](#)
- [15] 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 (CVPR)*, 2020. [5](#)
- [16] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D. Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, Asaf Nitzan, Gang Dong, Yuting Ye, Lingling Tao, Chengde Wan, and Robert Wang. MEgATrack: Monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)*, 39(4):87–1, 2020. [1](#)
- [17] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. DexPilot: Vision-Based Teleoperation of Dexterous Robotic Hand-Arm System. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. [1](#)
- [18] Yana Hasson, Gül 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 (CVPR)*, 2019. [2](#)
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [4](#)
- [20] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand Pose Estimation via Latent 2.5D Heatmap Regression. In *European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [21] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-Object Contact Consistency Reasoning for Human Grasps Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [22] Hanbyul Joo, Tomas Simon, Mina Cikara, and Yaser Sheikh. Towards Social Artificial Intelligence: Nonverbal Social Signal Prediction in a Triadic Interaction. In *Proceedings of*

- the *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [23] 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 (CVPR)*, 2023. 2, 6
- [24] Thomas N Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*, 2016. 2
- [25] Dominik Kulon, Riza Alp Güler, 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 (CVPR)*, 2020. 1, 2, 4
- [26] Rui Li, Hongyu Wang, and Zhenyu Liu. Survey on Mapping Human Hand Motion to Robotic Hands for Teleoperation. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 32(5):2647–2665, 2021. 1
- [27] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A Simple Approach to Intrinsic Correspondence Learning on Unstructured 3D Meshes. In *European Conference on Computer Vision (ECCV)*, 2018. 3
- [28] 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 (CVPR)*, 2021. 2
- [29] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh Graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 5, 6, 7
- [30] 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 (CVPR)*, 2023. 1, 2
- [31] 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 (CVPR)*, 2021. 2, 6
- [32] 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 (ECCV)*, 2020. 2, 6
- [33] Gyeongsik Moon, Shou-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. 2, 5
- [34] Gyeongsik Moon, Shunsuke Saito, Weipeng Xu, Rohan Joshi, Julia Buffalini, Harley Bellan, Nicholas Rosen, Jesse Richardson, Mallorie Mize, Philippe De Bree, Tomas Simon, Bo Peng, Shubham Garg, Kevyn McPhail, and Takaaki Shiratori. A Dataset of Relighted 3D Interacting Hands. *Advances in Neural Information Processing Systems*, 36:17689–17701, 2023. 5
- [35] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2Hands: Learning To Infer 3D Hands From Conversational Gesture Body Dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1
- [36] 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 (CVPR)*, 2022. 1, 2, 6
- [37] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing Hands in 3D with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2
- [38] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics (TOG)*, 36(6):1–17, 2017. 1, 2
- [39] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *European Conference on Computer Vision (ECCV)*, 2020. 6
- [40] 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 (ICCV)*, 2021. 1, 6
- [41] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP Architecture for Vision. *Advances in Neural Information Processing Systems*, 34:24261–24272, 2021. 4
- [42] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 30:6309–6318, 2017. 2, 3, 4
- [43] 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 (CVPR)*, 2023. 1, 6, 7
- [44] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. BiHand: Recovering Hand Mesh with Multi-stage Bisected Hourglass Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020. 2
- [45] Linlin Yang, Shicheng Chen, and Angela Yao. SemiHand: Semi-supervised Hand Pose Estimation with Consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2
- [46] Lixin Yang, Xinyu Zhan, Kailin Li, Wenqiang Xu, Jiefeng Li, and Cewu Lu. CPF: Learning a Contact Potential Field to Model the Hand-Object Interaction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2
- [47] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. OakInk: A Large-Scale Knowledge

- Repository for Understanding Hand-Object Interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [5](#)
- [48] Yusuke Yoshiyasu. Deformable Mesh Transformer for 3D Human Mesh Recovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#), [6](#)
- [49] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. MetaFormer Is Actually What You Need for Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [4](#)
- [50] Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting Two-Hand 3D Pose and Shape Reconstruction from Single Color Image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [2](#)
- [51] 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 (ICCV)*, 2019-10. [2](#), [5](#), [6](#)
- [52] 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 (ICCV)*, 2021.
- [53] Zimeng Zhao, Xi Zhao, and Yangang Wang. TravelNet: Self-supervised Physically Plausible Hand Motion Learning from Monocular Color Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. [1](#)
- [54] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular Real-Time Hand Shape and Motion Capture Using Multi-Modal Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#), [2](#)
- [55] Zhishan Zhou, Shihao Zhou, Zhi Lv, Minqiang Zou, Yao Tang, and Jiajun Liang. A Simple Baseline for Efficient Hand Mesh Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [2](#), [6](#)
- [56] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max J. 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 (ICCV)*, 2019. [5](#)