frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 29 January 2025
DOI 10.3389/fphy.2024.1515842

OPEN ACCESS

EDITED BY
Peican Zhu,
Northwestern Polytechnical University, China

REVIEWED BY
Anas Bilal,
Hainan Normal University, China
Abdelkarim Ben Sada,
University College Cork, Ireland

*CORRESPONDENCE
Huan Wang,
wanghuan6@email.szu.edu.cn

RECEIVED 23 October 2024
ACCEPTED 18 December 2024
PUBLISHED 29 January 2025

CITATION
Li Y, Wang C and Wang H (2025) Toward accurate hand mesh estimation via masked image modeling.
Front. Phys. 12:1515842.
doi: 10.3389/fphy.2024.1515842

Toward accurate hand mesh estimation via masked image modeling

Yanli Li[1], Congyi Wang[2] and Huan Wang[3]*

[1]Fuzhou Medical College of Nanchang University, Fuzhou, China, [2]Financial Technology Research Institute of the Industrial Bank, Fuzhou, China, [3]Industrial Technology Research Center, Guangdong Institute of Scientific and Technical Information, Guangzhou, China

Introduction: With an enormous number of hand images generated over time, leveraging unlabeled images for pose estimation is an emerging yet challenging topic. While some semi-supervised and self-supervised methods have emerged, they are constrained by their reliance on high-quality keypoint detection models or complicated network architectures.

Methods: We propose a novel selfsupervised pretraining strategy for 3D hand mesh regression. Our approach integrates a multi-granularity strategy with pseudo-keypoint alignment in a teacher–student framework, employing self-distillation and masked image modeling for comprehensive representation learning. We pair this with a robust pose estimation baseline, combining a standard vision transformer backbone with a pyramidal mesh alignment feedback head.

Results: Extensive experiments demonstrate HandMIM's competitive performance across diverse datasets, notably achieving an 8.00 mm Procrustes alignment vertex-point-error on the challenging HO3Dv2 test set, which features severe hand occlusions, surpassing many specially optimized architectures.

KEYWORDS

3D hand mesh estimation, multi-granularity representation, self-supervised learning, masked image modeling, vision transformer

1 Introduction

Image-based 3D hand reconstruction technology has widespread applications in the smart film industry, such as motion capture, special effects synthesis, virtual production, post-production animation modification, and interactive film production. Meanwhile, 3D hand mesh estimation from monocular RGB images has drawn great attention in computer vision research [1, 2] driven by its potential in various applications, such as action recognition [3, 4], digital human modeling, simultaneous localization and mapping (SLAM) [5–10], and AR/VR. However, training a high-quality hand estimation model is challenging due to complex backgrounds and severe self-occlusion. Furthermore, it is laborious and costly to collect high-quality training pairs, especially in the format of 3D mesh. A limited amount of image-mesh training data are available, making it difficult to train effective and generalizable models. Weakly supervised methods detecting 2D keypoints or measuring noisy depth maps [11] or kinematic priors [12] from off-the-shelf models have been proposed to improve the accuracy of supervised-trained models. However, these methods

heavily rely on fine-grained keypoint detectors, such as MediaPipe [13], which struggle with the wide variety of wild images encountered in practice and may produce many noisy labels.

Self-supervised learning is a promising technique for addressing the above problem by exploiting the large quantity of unlabeled image data generated over time. Masked image modeling (MIM) pretraining has emerged as a new paradigm in self-supervised learning based on the vision transformer [14] architecture that divides images into individual patches. In MIM pretraining, we randomly mask a specified ratio of image patches and set the self-supervised learning target to reconstruct the masked patches. Previous works [15, 16] have demonstrated that MIM-based methods can learn better local and global representation than conventional self-supervised methods based on contrastive learning [17]. In contrast to traditional self-supervised methods based on contrastive learning, which focus on high-level feature representation suitable for image classification, MIM-based methods can learn better local and global representations. This is especially critical for low-level, fine-grained regression tasks such as 3D hand estimation, where capturing the equivalence of geometric transformations is essential. The potential ability of MIM to reconstruct masked patches allows the model to understand the spatial relationships within an image at a finer granularity, making it more adept at handling detailed structures like the human hand.

However, most existing self-supervised work focuses on recognition tasks and aims to learn features appropriate for high-level image classification tasks. In low-level regression tasks, mainstream methods cannot capture the equivalence of geometric transformation, a critical characteristic of human/hand pose or mesh regression. Therefore, most state-of-the-art MIM self-supervised pretraining approaches must be adapted for regression tasks such as 3D hand estimation. Figure 1 exhibits the difference between our MIM approach and the previous ones. MIM's extension to regression tasks like 3D hand mesh estimation offers significant advantages. It leverages the strengths of MIM—such as detailed feature capture and understanding of spatial relationships—while introducing mechanisms specifically tailored for the challenges of regression tasks. We confirmed the abovementioned findings through experiments in Section 4.3.

In this paper, we conduct the first attempt to apply the effective masked image modeling (MIM) self-supervised technique to 3D hand estimation tasks. We propose HandMIM, a unified and multi-granularity self-supervised pretraining strategy optimized for pose regression tasks. During the pretraining period, we use a teacher-student self-distillation approach, where input hand images are augmented into two views that vary in sizes, rotations, colors, and other factors. The student network is then tasked with reconstructing masked tokens under the guidance of the teacher network. To ensure that the class tokens are semantic with pose-aware knowledge, we introduce the pseudo-keypoint alignment operation in the latent feature space. This operation allows us to undo the geometric transformation in the format of 2D pseudo-keypoints, enabling the network to learn pose equivalence between cross-view tokens. To facilitate high-level and low-level recognition, we adopt token-level recovery between parallel-view masked tokens and pixel-level reconstruction between masked input images and recovered images, respectively. It is important to note that the token recovery is conducted in the *same* latent space as the pose-aware alignment. We sketch our method in Figure 2 and compare it with related self-supervised works [18, 19] for hand pose/shape estimation. PeCLR [18] is the current state-of-the-art hand pose estimation work using a self-supervised training approach. Our model differs from PeCLR [18] in the following aspects: First, we learn global features using a self-distillation manner rather than the contrastive learning paradigm. Second, we designed the *pose-aware keypoint alignment* mechanism, making HandMIM exploit the pose knowledge, which as originally coupled with task-irrelated information (such as color, affine transformation, etc.) from the image. Last, *token-level self-distillation* and *pixel-level reconstruction* are imposed to learn the local or low-level features, which are vital for regression tasks like 3D mesh estimation. Accordingly, HandMIM overcomes the limitations of contrastive learning and other self-supervised approaches by incorporating multi-granularity feature learning and pose-aware mechanisms in a unified self-distillation-based MIM framework. This combination results in superior performance on 3D hand mesh estimation tasks. In the supervised fine-tuning period, most existing pose estimation methods rely on a combination of grid convolution, transformer structure, and a dedicated and complicated prediction head for better results. We designed a simple yet effective pose estimation pipeline with a standard vision transformer as the backbone, attached by a PyMAF [20] decoder head to promote mesh-image alignment and use the MANO [21] parameters to represent the estimated hand mesh. We loaded the self-supervised, pre-trained weights to transformer blocks and fine-tuned the whole network for hand pose estimation. Extensive experiments demonstrated that our HandMIM can learn better features to improve 3D hand pose estimation precision than alternative self-supervised and fully supervised methods under the same amount of labeled training data. We conducted our main experiments on two mainstream and challenging 3D hand mesh estimation datasets, FreiHAND [22] and HO3Dv2 [23]. We implemented HandMIM on three different sizes of vision transformers, namely ViT-Small, ViT-Base, and ViT-Large, respectively, which show strong scalability. After HandMIM pretraining, we achieved a performance boost of 9.7%/11.9%/16.5% in Procrustes Alignment Joint Position Error (PAJPE) on the FreiHAND [22] test set and 7.1%/8.4%/9.0% on the HO3Dv2 [23] test set. Notably, after pretraining HandMIM on ViT-Large, we achieve competitive results on 3D hand mesh estimation through the simple standard vision transformer architecture rather than complex graph architectures adopted by fully supervised methods such as I2L-MeshNet [24].

Conclusively, the main contributions of our work are in four folds:

1. We adopted a new self-distillation method for 3D hand mesh estimation. This method markedly enhanced the efficiency of learning from potentially unlimited unlabeled hand image data.
2. We designed the *pose-aware keypoint alignment* mechanism for the MIM paradigm, making HandMIM exploit the

10.3389/fphy.2024.1515842

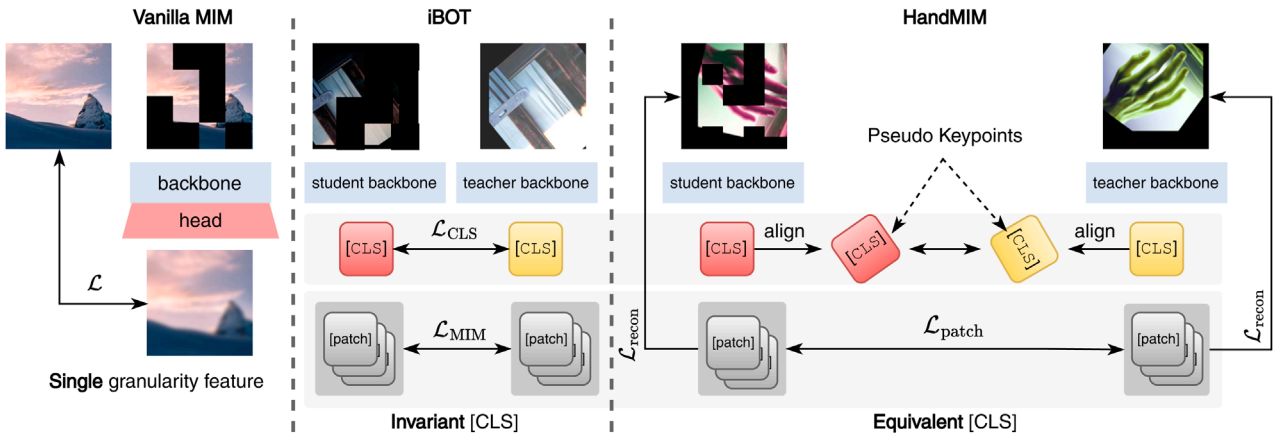


FIGURE 1
Comparison with other MIM self-supervised frameworks in vision tasks. Left: MIM techniques such as [16] encourage the model to recover the masked patch of images. Middle: iBOT [15] utilized a vision transformer [14] to extract multi-level features for image details and semantics. Self-distillation mechanics is introduced to learn the semantic [CLS] feature, which is invariant [15] under task-specific transformation. Right: our HandMIM pruned the MIM-training ViT architecture to fit the properties of the regression task. The pose-aware alignment mechanics are designed to enforce the transformation equivalence [18] of [CLS] and the patch features, given the masked pose image, which boosts the regression task. Note that our framework adds the geometric equivalence property to the [CLS] token via pseudo-keypoints and simultaneously learns global, patch, and pixel-level features, which are specially tailored for the fine-grained regression tasks.

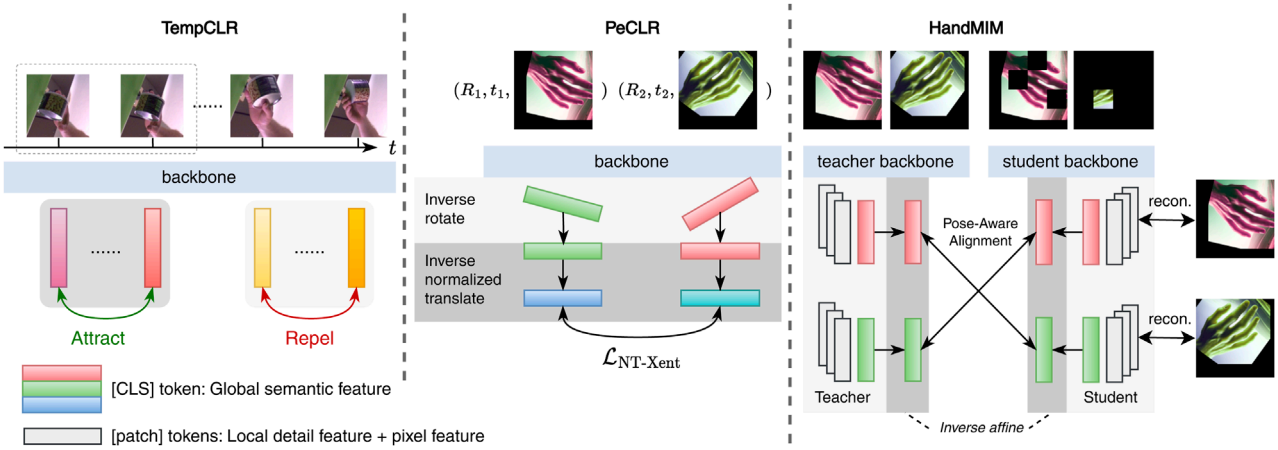


FIGURE 2
Comparison with other self-supervised frameworks for hand shape/pose estimation. Left: TempCLR [19] proposes exploiting the temporal relations to generate positive and negative samples for contrastive learning. Middle: PeCLR [18] is noted for implementing a transformation equivalence constraint on extracted global features. It is important to note that both TempCLR [19] and PeCLR [18] are contrastive learning methods that focus solely on global features, neglecting multi-level features that could significantly refine the predicted hand mesh vertices. Right: our contribution stands out with the introduction of *pose-aware alignment*mechanics and multi-granularity feature learning, which is the key difference between HandMIM and competitive methods. We first align pseudo-keypoints (represented by the [CLS] token) in the latent space. Concurrently, the [patch] tokens capture detailed features essential for mesh estimation and refinement. For clarity, the token-level self-distillation mechanics are omitted in this comparison. Unlike contrastive learning approaches such as PeCLR [18], which focus on global high-level features, HandMIM's pixel-level reconstruction enhances the model's ability to rebuild fine-grained geometric details in hand mesh vertex prediction. This integration helps the model recover from occlusions and challenging hand poses. By predicting the original pixel values from masked input, HandMIM is more effective at handling transformations and occluded regions, which is evident from the improved performance on datasets with hand–object interaction (e.g., HO3Dv2).

pose knowledge, which was originally coupled with task-irrelated information (such as color and affine transformation) from images.

3. The integration of *token-level self-distillation* and *pixel-level reconstruction* in our framework allowed the effective learning of both high- and low-level features. These features are crucial for fine-grained regression tasks, including hand mesh estimation.
4. To our knowledge, HandMIM represented the inaugural model pre-trained with masked image modeling mechanics, specifically in the field of hand mesh estimation.

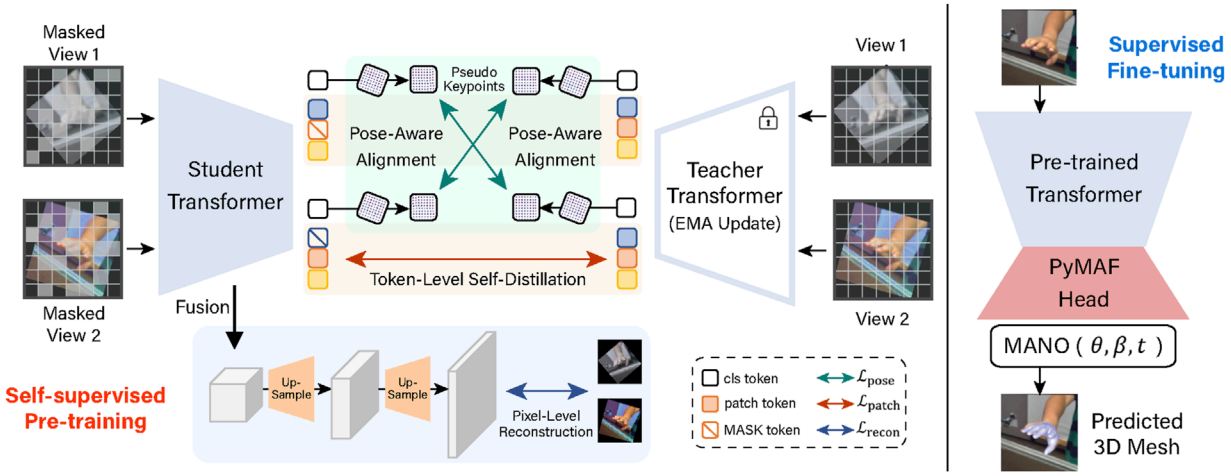


FIGURE 3
Overall framework of HandMIM. During the self-supervised pretraining phase, we design multi-granularity tasks to acquire pose-aware knowledge, high-level token recovery, and low-level pixel reconstruction. We propose a simple baseline based on the standard vision transformer architecture and the PyMAF [20] decoder in the fine-tuning period. We load the pre-trained weights onto the model, which allows us to estimate 3D hand poses by leveraging the network's ability to learn pose-sensitive information in hand images.

2 Related work

2.1 Hand pose estimation

Estimating hand poses aims to predict hand information from a monocular RGB/depth image and can be broadly classified into parametric and non-parametric methods. Parameter methods [25] use statistical priors from parametric hand models like MANO [21] to constrain the regression space and make the prediction more robust in cases of severe occlusions. Except for fully supervised manners, pioneer works [12] predict the MANO parameters with weak supervision, such as hand masks, depth maps, or 2D annotations. Non-parametric methods [26, 27] aim to predict the entire mesh vertices directly using either graph convolutional networks or transformer blocks. Although these methods can generate results that align better with the input image, they are more prone to failure in cases of occlusions and truncations. More recent work has focused on explicitly modeling hand–hand [28], complicated hand–object interactions [29], high inference speed [30], and increased robustness to occlusions [29] that pose new and more complex challenges. Instead of designing dedicated and resource-intensive heads, we proposed a lightweight head that regresses MANO parameters from a pre-trained standard ViT for both single-hand estimation and hand–object interaction predictions.

2.2 Vision transformer (ViT)

ViT [14] first introduced vision transformers to the visual field by patching images for transformer blocks. This approach has led to significant progress in image recognition and has also shown promising results in human and hand estimation tasks [1, 2, 27]. For example, Mesh Graphormer [1] designs a transformer-based head fused with graph convolution layers. HaMeR [31] directly utilizes ViT with a transformation head to predict MANO parameters and camera extrinsic with several mixed-label labeled datasets. Keypoint Transformer [2] first collects candidate 2D keypoints and utilizes a transformer encoder-decoder for the mesh predictions. AMVUR [26] further proposes a probabilistic attention-based mesh vertices model to estimate the prior probability distribution of joints and mesh vertices to improve their feature representation.

Most prior works have designed complex structures on top of the transformer or attention blocks. Accordingly, standard transformers cannot easily achieve competitive performance. Our approach attempts to leverage large quantities of unlabeled hand images and surpass existing methods solely based on the standard ViT backbone without any delicate domain-related architecture, demonstrating the effectiveness of our self-supervised regression learning algorithm.

2.3 Self-supervised learning

Self-supervised learning is an approach to learning effective feature representation from abundant unlabeled images. Contrastive learning techniques [17] aim to learn by constraining positive pairs to become close in feature space while pushing negative pairs apart and have been employed in the hand pose estimation tasks for improved performance [18, 19, 32]. Masked image modeling (MIM) [15, 16, 33] methods are new paradigms of self-supervised learning that randomly mask a portion of the input image and reconstruct the masked parts via reasoning other unmasked parts. The knowledge of masked images can be learned in alterable manners, including dVAE codebooks in

BeiT [33], raw RGB pixels in MAE [16], etc. Previous MIM studies have focused on learning representative features for image classification tasks but have neglected the specificity of pose or mesh regression tasks. To our knowledge, this is the first time that MIM techniques have been extended to such 3D regression tasks. MIM contributes to the proposed HandMIM in two aspects. First, it allows the model to learn local and global features better than traditional contrastive learning methods. It captures spatial relationships within images at a finer granularity, making it more adept at handling detailed structures like human hands. Second, it constitutes an important part of our designed multi-granularity loss functions, which involve both token-level recovery between parallel-view masked tokens and pixel-level reconstruction between masked inputs and recovered images. This dual-level loss facilitates high-level and low-level recognition, ensuring comprehensive representation learning.

3 Methods

In this section, we will discuss the detailed architecture of HandMIM. The pipeline of HandMIM can be found in Figure 3. We start with preliminaries, including basic knowledge of vision transformers, masked image modeling, and self-distillation techniques in Section 3.1. Then, we introduce the detailed design of HandMIM, including pose-aware keypoint alignment in Section 3.2, token-level self-distillation in Section 3.3, and pixel-level reconstruction in Section 3.4. Finally, we illustrate how to apply pre-trained features after self-supervised learning for 3D hand mesh estimation tasks in Section 3.5. The PyTorch-like pseudocode of HandMIM is listed in Algorithms 1–3.

3.1 Preliminaries

3.1.1 Vision transformers

Given input images $I \in \mathbb{R}^{3\times H\times W}$, a vision transformer [14] applies a patch embedding layer to divide the images into patch tokens $z \in \mathbb{R}^{n^2\times c}$, where n is determined by a pre-defined patch length, and c is the channel dimension after the first convolution layer. A learnable class token is appended to the patch tokens to generate the input feature $z \in \mathbb{R}^{(n^2+1)\times c}$ of transformer blocks. Each transformer block comprises a multi-head self-attention (MHSA) and feed-forward network (FFN). The input tokens are projected into query, key, and value triplets $(\mathbf{Q},\mathbf{K},\mathbf{V})$ through linear layers, and the forward process of MHSA can be formulated as Equation 1:

$$\mathrm{MHSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}. \quad (1)$$

The output of the self-attention module is then passed through an inverted bottleneck multi-layer perceptron (MLP), also known as the feed-forward network. In practice, vision transformers are assembled by stacking a series of transformer blocks. We can obtain models of varying sizes by varying the channel width and layer depth of vision transformers.

Input:
batch size N, constant softmax temperature $\tau_s^{\text{Pose}}, \tau_s^{\text{Patch}}, \tau_t^{\text{Pose}}, \tau_t^{\text{Patch}}$, student and teacher network f_s, f_t, logit center $C^{\text{Pose}}, C^{\text{Patch}}$.
Pseudo code:
for sampled minibatch images $\{x_k\}_{k=1}^N$ **do**
 for all $k \in \{1,\ldots,N\}$ **do**
  draw two random augmentation functions $t \sim \Gamma, t' \sim \Gamma$
  $\tilde{x}_{2k-1}, R_{2k-1}, [i,j,h,w]_{2k-1} = t(x_k)$ ▷ random data augmentation for image, $[i,j,h,w]$ is the image crop parameters (left, top, height, and width), and R is the 2 × 2 rotation matrix.
  $\tilde{x}_{2k}, R_{2k}, [i,j,h,w]_{2k} = t'(x_k)$
  $\tilde{x}_{2k-1}, M_{2k-1} = \mathrm{mask}(\tilde{x}_{2k-1})$ ▷ randomly mask the image
  $\tilde{x}_{2k}, M_{2k} = \mathrm{mask}(\tilde{x}_{2k})$
  $u_{2k-1} = f_s(\tilde{x}_{2k-1}), u_{2k} = f_s(\tilde{x}_{2k})$ ▷ tokens encoded by student network
  $v_{2k-1} = f_t(\tilde{x}_{2k-1}), v_{2k} = f_t(\tilde{x}_{2k})$ ▷ tokens encoded by teacher network
  $h_{2k-1} = \mathrm{decoder}(u_{2k-1}), h_{2k} = \mathrm{decoder}(u_{2k})$ ▷ reconstruct the image from tokens
  $u_{2k-1}^{\text{CLS}}, u_{2k-1}^{\text{Patch}} = split(u_{2k-1})$
  $v_{2k-1}^{\text{CLS}}, v_{2k-1}^{\text{Patch}} = split(v_{2k-1})$
  $u_{2k}^{\text{CLS}}, u_{2k}^{\text{Patch}} = split(u_{2k})$
  $v_{2k}^{\text{CLS}}, v_{2k}^{\text{Patch}} = split(v_{2k})$
  $u_{2k-1}^{\text{CLS}}, v_{2k-1}^{\text{CLS}} = \mathrm{PAA}(u_{2k-1}^{\text{CLS}}, v_{2k-1}^{\text{CLS}}, M_{2k-1}, [i,j,h,w]_{2k-1})$ ▷ *Pose-aware alignment*
  $u_{2k}^{\text{CLS}}, v_{2k}^{\text{CLS}} = \mathrm{PAA}(u_{2k}^{\text{CLS}}, v_{2k}^{\text{CLS}}, M_{2k}, [i,j,h,w]_{2k})$ ▷ *Pose-aware alignment*
  $\mathcal{L}_{\text{Pose}} = \text{S-D}(u_{2k-1}^{\text{CLS}}, v_{2k}^{\text{CLS}}, C^{\text{Pose}}, \tau_s^{\text{Pose}}, \tau_t^{\text{Pose}}) + \text{S-D}(u_{2k}^{\text{CLS}}, v_{2k-1}^{\text{CLS}}, C^{\text{Pose}}, \tau_s^{\text{Pose}}, \tau_t^{\text{Pose}})$ ▷ [CLS] token loss
  $\mathcal{L}_{\text{MIM}} = \text{S-D}u_{2k-1}^{\text{Patch}}, v_{2k-1}^{\text{Patch}}, C^{\text{Patch}}, \tau_s^{\text{Patch}}, \tau_t^{\text{Patch}} \cdot M_{2k-1}.\mathrm{mean}() + \text{S-D}u_{2k}^{\text{Patch}}, v_{2k}^{\text{Patch}}, C^{\text{Patch}}, \tau_s^{\text{Patch}}, \tau_t^{\text{Patch}} \cdot M_{2k}.\mathrm{mean}()$ ▷ [Patch] token loss
  $\mathcal{L}_{\text{Recon}} = |h_{2k-1} - \tilde{x}_{2k-1}| \cdot M_{2k-1}.\mathrm{mean}() + |h_{2k} - \tilde{x}_{2k}| \cdot M_{2k}.\mathrm{mean}()$ ▷ image reconstruction loss
  $\mathcal{L} = \mathcal{L}_{\text{Pose}} + \mathcal{L}_{\text{MIM}} + \mathcal{L}_{\text{Recon}}$
 end for
 Update network f_s to minimize $\mathcal{L}$
 Update network f_t using exponentially moving average (EMA)
 Update logit center $C^{\text{Pose}}, C^{\text{Patch}}$ by moving average
end for
Return student network f.

Algorithm 1. HandMIM PyTorch-like Style Pseudocode.

Input: s, t, c, τ_s, τ_t.
Pseudo code:
$s = \mathrm{softmax}(s/\tau_s)$
$t = \mathrm{softmax}((t-c)/\tau_t)$
return $-(t \cdot \log(s)).\mathrm{sum}(\mathrm{dim} = -1)$

Algorithm 2. Self-distillation (S-D).

Input: $x^{CLS}, R, [i, j, h, w]$.
Pseudo code:
x^{CLS} = MLP (x^{CLS})
x^{CLS} = x^{CLS}.reshape (-1, 2) · R
x^{CLS} = (x^{CLS*} $[h, w] + [i, j]$)/img_size
x^{CLS} = x^{CLS}.reshape (-1)
return x^{CLS}

Algorithm 3. Pose-aware alignment (PAA).

3.1.2 Masked image modeling

Masked image modeling (MIM) is a self-supervised learning technique that has been demonstrated to be a general method for image recognition tasks in many recent works [33]. Given input tokens $z \in \mathbb{R}^{n^2 \times c}$, randomly create a binary mask $\mathcal{M} \in \{0,1\}^{n \times n}$. When $\mathcal{M} = 1$, the origin image tokens are passed through the neural network backbone, and when $\mathcal{M} = 0$, the input tokens are replaced with a special mask token p_{mask}. By doing so, we obtain both the original tokens z and the masked tokens $\hat{z}$, which are calculated as $\hat{z} = \mathcal{M} \odot p_{mask} + (1 - \mathcal{M}) \odot z$. The goal of the masked image modeling task is to train the backbone function $f(\cdot)$ and minimize the following loss function Equation 2 to recognize and recover the original tokens z from the masked tokens $\hat{z}$:

$$\mathcal{L}_{MIM} = \mathcal{M} \cdot \|f(\hat{z}) - z\|^2. \tag{2}$$

MIM encourages the model to learn robust local and global image representations, which is especially important for tasks requiring fine-grained understanding, such as 3D hand mesh estimation.

3.1.3 Self-distillation

Self-distillation is a common technique adopted in recent self-supervised learning frameworks [34, 35]. Given an input image I, we apply two random data augmentations to the image, denoted as I_1 and I_2, respectively. During training, we treat the backbone function $f(\cdot)$ as the student network. The teacher network shares the same architecture as the student network, but its weights are updated using the exponential moving average of the student weights rather than through gradient updates. The goal of self-distillation is to minimize the following consistency loss function Equation 3, which enforces consistency between the output features from the student and teacher networks using I_1 and I_2, respectively, where $\mathcal{D}$ is the distance metric, such as Kullback–Leibler divergence or L1/L2 loss functions:

$$\mathcal{L}_{SD} = \mathcal{D}\left(f_{\text{student}}(I_1), f_{\text{teacher}}(I_2)\right). \tag{3}$$

3.2 Pose-aware keypoint alignment

We observe that the 2D pose of hands in input images remains equivalent after some spatial data augmentation, such as random rotation and resizing operations, while the positional information is altered. As justified in our experiments, existing mainstream self-supervised learning methods fail to capture the knowledge of "poses." In this work, we propose the idea of pose-aware keypoint alignment to extract the pose-relevant knowledge. This is critical for 3D hand mesh estimation tasks, where understanding and preserving the geometric relationships between keypoints (or joints) is essential for accurate reconstruction. Moreover, we choose this method because it efficiently and effectively captures and utilizes pose-relevant knowledge, integrates seamlessly with the self-distillation and multi-granularity learning paradigms, and enhances the overall performance and robustness of 3D hand mesh estimation.

Consider a point $P = (x_0, y_0)$ in input image I. After the augmentation process, the point is transformed to Equation 4:

$$(x_0', y_0') = R\left((x_0, y_0) \cdot \gamma - (a, b)\right), \tag{4}$$

where R denotes 2D rotation matrix, γ denotes scale factor, and (a, b) denotes the upper left coordinate of the resized image. After the last transformer layer, we obtain the output class token $\tau \in \mathbb{R}^{1 \times c}$ in latent space, where we can regard it as a set of *pseudo points* and reshape τ into the format of point $(\tau_x, \tau_y) \in \mathbb{R}^{2 \times c/2}$. We can then recover the linear transformation $\Lambda(\cdot)$ to get the original latent feature τ' before any spatial augmentation as the folloiwng functions Equations 5, 6:

$$\left(\tau_x', \tau_y'\right) = 1/\gamma \cdot \left(R^{-1}\left(\tau_x, \tau_y\right) + (a, b)\right), \tag{5}$$

$$\tau' = \Lambda(\tau). \tag{6}$$

Then, we apply the softmax function to τ' and obtain class token features $U_{[cls]}, V_{[cls]}, \hat{U}_{[cls]}, \hat{V}_{[cls]}$ to compute the cross-entropy self-distillation losses as depicted in the following subsection. After the pose alignment in latent space, image features after different augmentations exhibit a unified "hand pose," facilitating the extraction of pose-sensitive knowledge by vision backbones. In the following subsection, we will elaborate on how to learn the pose-aware task.

3.3 Token-level self-distillation

The knowledge of masked image modeling can be acquired through a self-distillation approach proposed by DINO [36]. We treat self-supervised learning as a discriminative task involving two backbones with identical architecture, which play the roles of a teacher network f_t and a student network f_s. Specifically, we train the student network to comprehend corrupted input tokens $\hat{z}$ under the guidance of the teacher network, which receives complete input tokens z.

To fully recognize the images, we use two random image augmentations, denoted as μ and ν; thus, we get augmented tokens $u = \mu(z), v = \nu(z)$ for the teacher network f_t. We then apply a randomly generated mask $\mathcal{M}$ to the augmented tokens after the patch embedding layer, resulting in corrupted tokens $\hat{u}$ and $\hat{v}$ for the student network f_s. The process in the student and teacher networks can be formulated as the following Equation 7:

$$\hat{U} = \text{softmax}\left(f_s(\hat{u})\right), \hat{V} = \text{softmax}\left(f_s(\hat{v})\right), U = \text{softmax}\left(f_t(u)\right),$$
$$V = \text{softmax}\left(f_t(v)\right). \tag{7}$$

Note that the softmax function is applied to the channel dimension. We use uppercase letters, that is, $U = (U_{[cls]}, U_{[patch]}) \in$

TABLE 1 Details of the vision transformer architecture, as well as the pretraining and inference time in HandMIM.

Model	Layer depth	Embed dim	MLP size	Number of heads	Params (M)	Pretraining time (hours)	Inference FPS
ViT-Small	12	384	1,536	6	22	12	40
ViT-Base	12	768	3,072	12	86	23	15
ViT-Large	24	1,024	4,096	16	307	53	4

TABLE 2 Results on the FreiHAND [22] dataset. We perform our results before (fine-tuned from ImageNet pre-trained weights) and after HandMIM pretraining and list the lifting ratio compared with the ViT-S baseline.

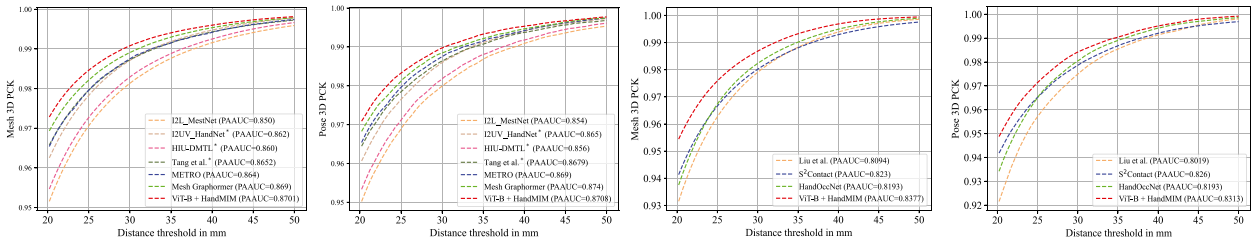
Method	Params(M)	PAVPE↓	PAJPE↓	F@5↑	F@15↑
Kulon et al.[c] [11]	-	8.6	8.4	0.614	0.966
HaMeR/ViT-Base [31]	86 M	-	10.72	-	-
I2L-MestNet [24]	135 M	7.6	7.4	0.681	0.973
I2UV-HandNet [39]	-	7.4	7.2	0.707	0.977
HIU-DMTL[b] [40]	-	7.3	7.1	0.699	0.974
Tang et al. [41]	149 M	7.1	7.1	0.706	0.977
PeCLR-Res50[b] [18]	26 M	-	7.1	-	-
TempCLR-Res50[b] [19]	26 M	10.2	-	0.541	0.941
Mesh Graphormer[a] [1]	204 M	6.8	6.6	0.732	0.982
MobRecon[a] [30]	**22M**	7.2	6.9	0.694	0.979
FastViT [27]	-	6.6	6.7	0.722	0.981
ViT-Small-ImageNet + PyMAF	**22M**	7.1	7.2	0.697	0.978
ViT-Large-ImageNet + PyMAF	307 M	6.6	6.6	0.727	0.983
HandMIM-Small[b]	**22M**	$6.57_{-8.1\%}$	$6.57_{-9.7\%}$	0.725	0.984
HandMIM-Base[b]	86 M	$6.4_{-9.7\%}$	$6.4_{-11.9\%}$	0.731	0.985
HandMIM-Large[b]	307 M	$\mathbf{6.2}_{-12.9\%}$	$\mathbf{6.2}_{-16.5\%}$	**0.744**	**0.986**

[a]denotes non-ensemble evaluation results for a fair comparison.
[b]denotes self-supervised training approaches.
[c]denotes weakly supervised training approaches.
For a fair comparison, we re-trained HaMeR [31] with its official implementation on the FreiHand dataset alone using the same ViT-Base architecture. HandMIM outperforms various existing methods by a certain margin and achieves scalable results with larger ViT models. Notably, HandMIM achieves an impressive joint error of 6.6 mm with 22 M parameters. In comparison, MobRecon [42] cannot attain this level of joint error with the same number of parameters. The bold values means the optimal performance metric in each colum.

$\mathcal{R}^{(n^2+1)\times d}$ and $V = (V_{[cls]}, V_{[patch]}) \in \mathcal{R}^{(n^2+1)\times d}$, where d is the last latent dimension, to denote the output probability distribution [15] of the backbones f. The class tokens, denoted by $U_{[cls]} \in \mathcal{R}^{1\times d}$ and $V_{[cls]} \in \mathcal{R}^{1\times d}$, contain high-level semantic knowledge, while the patch tokens, denoted by $U_{[patch]} \in \mathcal{R}^{n^2\times d}$ and $V_{[patch]} \in \mathcal{R}^{n^2\times d}$, contain middle and low-level local knowledge of the input images.

We design specific tasks of self-supervised learning for the class tokens, considering their semantic meanings. For the class tokens, we aim to extract the pose of the original images, which is equivalent after the inverse operation of spatial data augmentations, implemented as pose-aware keypoint alignment in Section 3.2. Because we expect images under different augmentations to have the same pose expression, we adopt a cross-entropy loss between the *cross-view* images and apply the self-distillation approach in Section 3.1 to measure the discrepancy between teacher and student distribution. Specifically, we obtain the $\mathcal{L}_{\text{pose}}$ loss, which can be formulated as the following function Equation 8:

$$\mathcal{L}_{\text{pose}} = -U_{[cls]} \log \hat{V}_{[cls]} - V_{[cls]} \log \hat{U}_{[cls]}. \tag{8}$$



(a) FreiHAND Mesh AUC (b) FreiHAND Pose AUC (c) HO3D v2 Mesh AUC (d) HO3D v2 Pose AUC

FIGURE 4
Pose and mesh AUC comparison with some competitive methods on the FreiHAND and HO3D datasets.*indicates the method is supervised and trained with extra 2D/3D labeled data. It can be observed from the plot that our method achieves the best performance on both datasets for both meshes and poses AUC values with ViT-B as the backbone. **(A)** FreiHAND Mesh AUC, **(B)** FreiHAND Pose AUC, **(C)** HO3D v2 Mesh AUC, and **(D)** HO3D v2 Pose AUC.

TABLE 3 Results on the HO3D v2 [23] dataset. Compared with current methods specially designed for hand–object interactions, we achieve better results under a standard backbone with no special operation. All the listed results use the same labeled dataset for supervised learning.

Method	Params(M)	PAVPE↓	PAJPE↓	MPJPE↓	F@5↑	F@15↑
Liu et al[a]. [43]	34 M	9.5	9.9	31.7	0.528	0.956
HandOccNet [29]	38 M	8.8	9.1	24.0	0.564	0.963
AMVUR [26]	195 M	8.2	8.3	-	0.608	0.965
Keypoint Trans [2]	48 M	-	10.8	-	-	-
ViT-Small-ImageNet + PyMAF	**22M**	8.78	9.18	26.37	0.567	0.963
ViT-Large-ImageNet + PyMAF	307 M	8.43	8.73	23.57	0.588	0.970
HandMIM-Small	**22M**	$8.22_{-6.8\%}$	$8.57_{-7.1\%}$	24.00	0.597	0.970
HandMIM-Base	86 M	$8.08_{-8.0\%}$	$8.41_{-8.4\%}$	22.01	0.610	0.971
HandMIM-Large	307 M	$\mathbf{8.00}_{-8.9\%}$	$\mathbf{8.3}_{-9.0\%}$	**21.94**	**0.617**	**0.972**

[a]denotes the self-supervised training approach. Note that our HandMIM-Base already achieves competitive without any complicated designs for hand occlusion issues, such as AMVUR [26] or HandOccNet [29]. Moreover, the table demonstrates scalable results with larger ViT models.
The bold values means the optimal performance metric in each colum.

During the backward period, only the student network requires gradient backpropagation, as we treat the output of the teacher network as ground truth. Subsequently, we update the teacher network through an exponentially moving average (EMA) using the student network.

Given the patch output of the transformer backbone, which represents the spatial knowledge of input images, we can define the patch loss $\mathcal{L}_{\text{patch}}$. This loss measures the discrepancy between the *parallel-view* tokens, which share the same spatial position after the augmentations. Specifically, we aim to train our module to recover the corrupted patch tokens. We learn the knowledge using a similar self-distillation approach as in Equation 8 using Equation 9:

$$\mathcal{L}_{\text{patch}} = -\sum_{i=1}^{n^2} U_{i[patch]} \log \hat{U}_{i[patch]} + V_{i[patch]} \log \hat{V}_{i[patch]}. \quad (9)$$

3.4 Pixel-level reconstruction

Hand pose estimation is a low-level task that involves directly analyzing image pixels, in contrast to image classification. Although token-level self-distillation may be effective for higher-level knowledge, it may lack the necessary low-level understanding. To address this, we propose a pixel-level reconstruction module. Because transformer tokens are applied in a patch-based manner, we integrate a pyramid fusion layer following certain intermediate transformer layers and gradually up-sample using transposed convolution (T-Conv). The convolution stride is set to 2. The resulting pyramid fusion output feature maps (T^j) are concatenated with each transformer block output (L^j) and fused using a linear layer. Mathematically, this can be represented as the following Equation 10:

$$T^{j+1} = \text{T} - \text{Conv}\left(\text{Linear}\left(\text{Concat}\left[T^j, L^j\right]\right)\right). \quad (10)$$

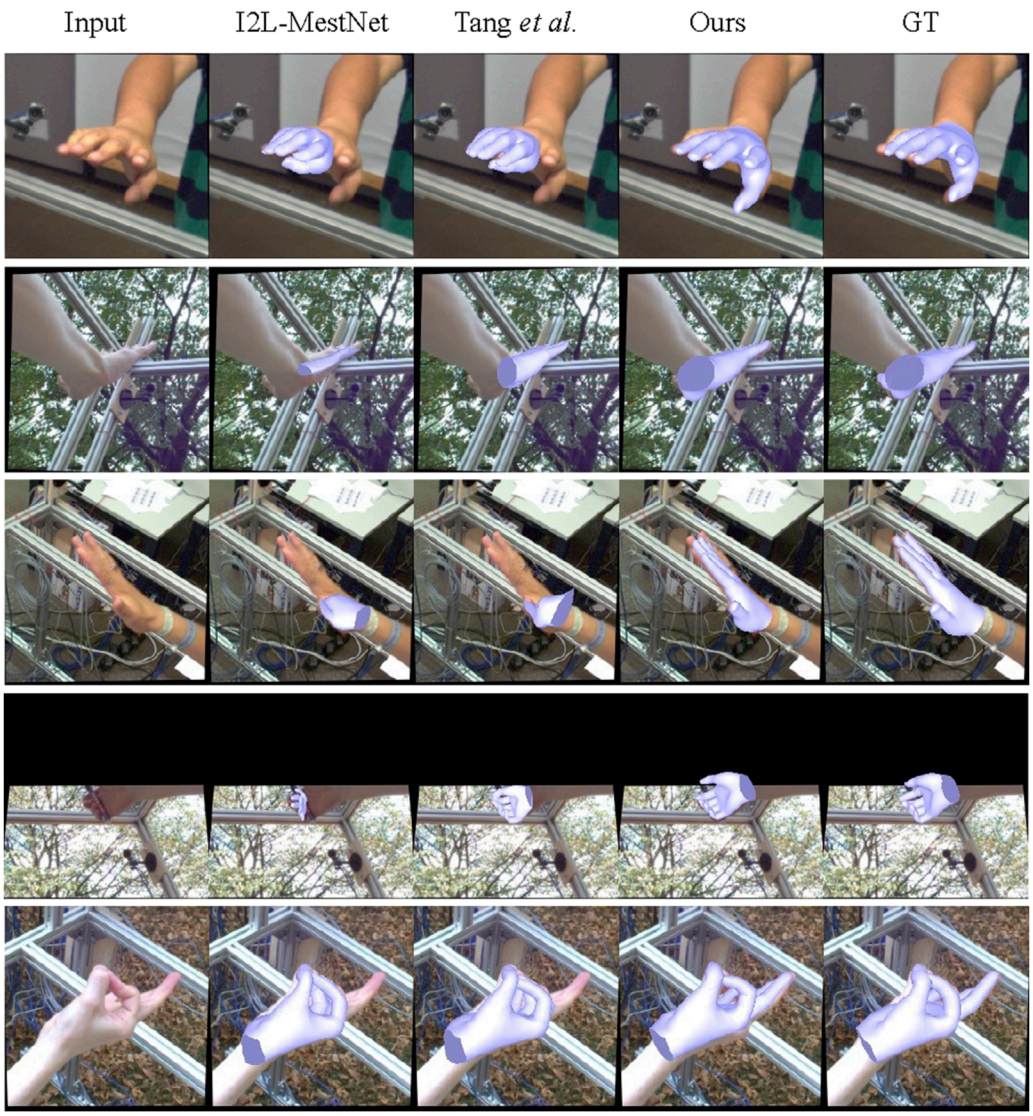


FIGURE 5
Visualizations on the FreiHAND [22] test set. From left to right, we show the input images, the predictions from I2L-MeshNet [24], Tang et al. [41], our HandMIM-Small, and the ground truth. Our method is more robust for hard viewpoints, occlusion, and complicated hand gestures.

In common practice, vision transformers use a patch size of 16; therefore, four iterations of transposed convolution are adopted to recover the original shape of input image I. We can adopt L1-Loss between input images and reconstruction results using Equation 11:

$$\mathcal{L}_{\text{recon}} = \mathcal{M} \odot \| (T^4 - I) \|_1, \tag{11}$$

where $\mathcal{M}$ denotes the token mask, and $\odot$ denotes the Hadamard product. Note that only the student network requires a gradient; therefore, we only adopt $\mathcal{L}_{\text{recon}}$ at the student network with masked input.

The final loss function Equation 12 is the sum of the losses mentioned above:

$$\mathcal{L} = \mathcal{L}_{\text{pose}} + \mathcal{L}_{\text{patch}} + \mathcal{L}_{\text{recon}}. \tag{12}$$

The above loss function indicates that HandMIM can capture both local detail features and global geometric context via a vision transformer backbone. The transformer architecture naturally handles multi-scale information, but HandMIM goes further by introducing a mechanism that specifically targets different levels of granularity. More specifically, the [Patch] tokens represent local regions of the image and are used to capture fine-grained geometric features essential for mesh estimation and refinement. Pseudo keypoints are aligned in the latent space using the [CLS] token, which acts as a global representation of the entire image. By aligning these keypoints, the model can better understand the pose equivalence between different views of the hand after applying spatial augmentations. Finally, the combination of pixel-level reconstruction and multi-granularity feature learning allows HandMIM to learn how to recover pixels from occlusions and handle complex hand–object interactions more effectively, which is particularly beneficial on datasets like HO3Dv2, which feature severe hand occlusions.

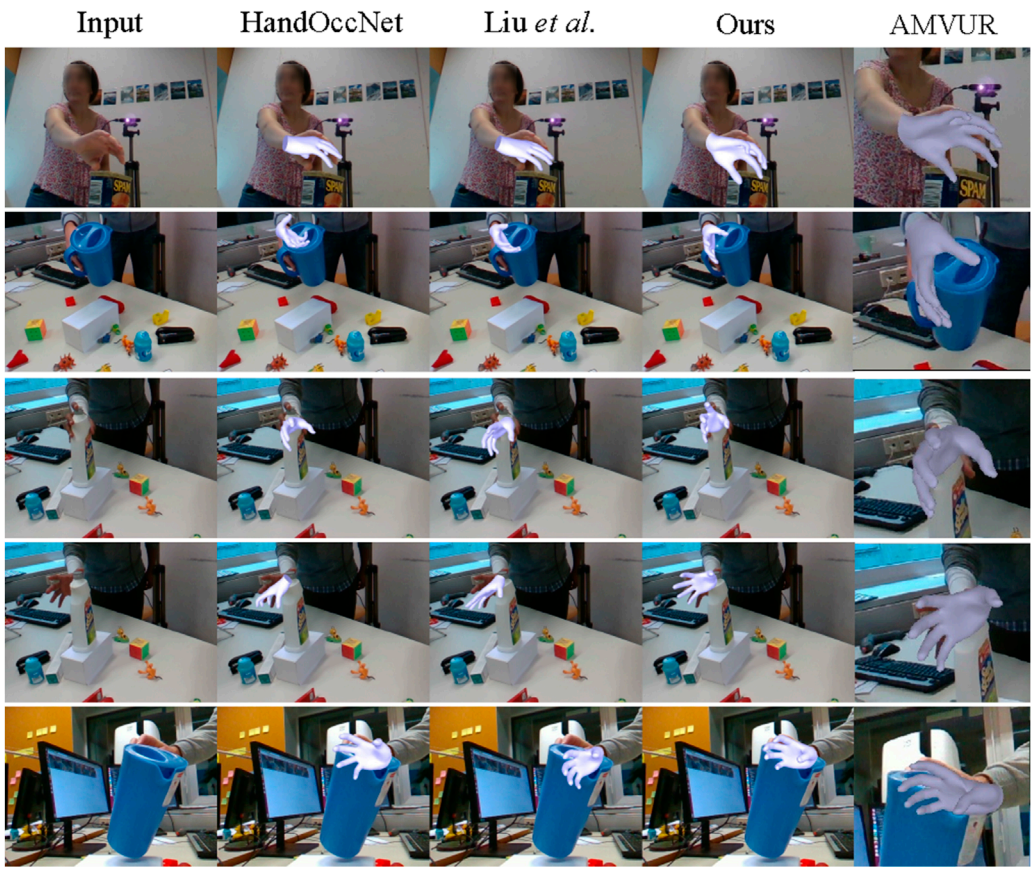


FIGURE 6
Visualization of the HO3D v2 [23] test set. We show the input images, the predictions from HandOccNet [29], Liu et al. [43], and our HandMIM-Small and AMVUR [26]. Our method can capture more precise poses even under the corruption and occlusion of complex objects, achieving results comparable to the strong baseline method AMVUR [26] while using many fewer parameters (195 M versus 22 M).

3.5 3D hand mesh estimation via ViT

To evaluate the effectiveness and benefits of HandMIM self-supervised pretraining, we fine-tune the pre-trained vision transformer backbone on a supervised 3D hand mesh estimation task. Specifically, we incorporate a keypoint feedback loop after the backbone, similar to the approach used in PyMAF [20], to predict MANO [21] parameters, including joint rotation (θ), shape coefficient (β), and global translation (δ). This keypoint feedback loop comprises three cascaded rectifier layers that extract local features based on the current keypoint-image alignment status and feed them back for rectification. Notably, the rectifier layers in PyMAF [20] utilize multi-level feature maps for its **coarse-to-fine** mesh refinement, which aligns exactly with the multi-level self-supervised losses (i.e., global, local patch distillation, and pixel reconstruction for fine-detail features) designed by HandMIM. Consequently, we adopt the PyMAF [20] structure as our mesh regressor. To train our method, we use a combination of MANO parameter loss ($\mathcal{L}_{\text{MANO}}$), vertex loss ($\mathcal{L}_{\text{vert}}$), and keypoint loss ($\mathcal{L}_{\text{kpt}}$), which are described as the following Equation 13.

$$\mathcal{L}_{\text{FT}} = \mathcal{L}_{\text{MANO}} + \mathcal{L}_{\text{vert}} + \mathcal{L}_{\text{kpt}}. \tag{13}$$

The MANO parameter loss ($\mathcal{L}_{\text{MANO}}$) is calculated as the L2 distance between the predicted MANO parameters and the ground truth. Given MANO parameters θ, β, δ, the 3D mesh vertices can be obtained using the MANO model $M(\theta,\beta) \rightarrow \mathbb{R}^{778}$, which can be used to calculate the vertex loss ($\mathcal{L}_{\text{vert}}$) as a more direct form of supervision. Furthermore, the 3D keypoints $J_{3D} \in \mathbb{R}^{21\times3}$ can be generated by mapping the 3D mesh using a pre-trained linear regressor. By projecting the 3D keypoints J_{3D} onto the image coordinate system, we can obtain 2D keypoints, which can be used to supervise the training process with 2D keypoint ground truth $\hat{J}_{2D}$. Overall, the keypoint loss $\mathcal{L}_{\text{kpt}}$ is composed of the 3D keypoint loss and the projected 2D keypoint loss as the following Equation 14:

$$\mathcal{L}_{\text{kpt}} = \|J_{3D} - \hat{J}_{3D}\|_1 + \|(K(J_{3D} + \delta) - \hat{J}_{2D}\|_1, \tag{14}$$

where K indicates the ground-truth camera intrinsic matrix following common practice. Together, the pre-trained ViT backbone and the pyramidal mesh alignment feedback head contribute significantly to the superior performance of HandMIM. The ViT backbone's capacity to learn detailed and hierarchical features and the PyMAF head's ability to refine the mesh through iterative alignment and direct parameter supervision results in competitive performance across various datasets, especially in challenging scenarios involving severe occlusions.

FIGURE 7
Qualitative comparison with several methods on the HO3D v2 test set. From left to right, it shows the input images, the overlaid results by HandOccNet [29], Liu et al. [43], and our best HandMIM-Large model.

4 Results

In this section, we conducted extensive experiments to evaluate the proposed self-supervised pretraining framework HandMIM. We first introduce our settings on HandMIM pretraining in Section 4.1. Then, we show the results of our pre-trained model on 3D hand mesh estimation tasks in Section 4.2. Finally, we present in-depth analysis and ablation studies in Section 4.3.

TABLE 4 Results of linear probe regression. We compared our method with the mainstream self-supervised learning method, and our HandMIM outperforms existing methods by a large margin.

Method	2D-error (px)↓	3D joint error (cm)↓
Random	21.69	80.57
iBOT [15]	9.74	23.67
PeCLR [18]	8.94	19.41
Ours/ViT-S	5.75	10.83
Ours/ViT-B	5.19	10.59

4.1 HandMIM pretraining

4.1.1 Pretraining settings

We employ vision transformers [14] as our backbone in different sizes, including ViT-Small (ViT-S), ViT-Base (ViT-B), and ViT-Large (ViT-L). Details of the architectures can be found in the supplementary materials. We collect the multi-level features from layers [3,6,9,12] for pixel reconstruction with a decoder consisting of linear layers for feature fusing and transposed convolutions for up-sampling. Input images are augmented through random resizing within the range $(0.08, 1)$, rotating within the range $(0, 150°)$, followed by color jitter, grayscale, Gaussian blur, and solarization. After the backbone, a shared MLP is used to project the tokens into latent space. We then resize the class token into high-level pseudo-keypoints with size $[128, 2]$, which indicates the latent dimension is 256. During HandMIM pretraining, we use AdamW as the optimizer with a batch size of 1,024. We pre-train ViT-S and ViT-B for 400 epochs, and ViT-L for 250 epochs. The learning rate is set to 2e-3, and the masked ratio r is randomly sampled within the range $[0.1, 0.5]$.

4.1.2 Pretraining datasets

As there are currently no standardized datasets for hand pose self-supervised learning, we collect hand images across a variety of datasets for sufficient hand pose and background distributions, including the FreiHAND [22] training set (FreiHAND is a 3D hand pose dataset that records different hand actions performed by 32 people. MANO-based 3D hand pose annotations are provided for each hand image. This training set provides a large number of hand images with green screen or composite backgrounds, offering a wide range of hand poses.), Youtube3DHands [11] (The dataset contains various in-the-wild images, with automatically acquired 3D annotations via key point detection and MANO fitting. It has 47,125 effective frames.), and COCO-WholeBody train and unlabeled images [37] (It is a large-scale dataset with keypoint and bounding box annotations. Approximately 130 K faces and left/right-hand boxes are labeled, resulting in more than 800 K hand keypoints and 4 M face keypoints in total.) For datasets with hand annotations, we directly enlarge the bounding boxes of the hand annotations by a ratio of 2.0 and then crop the hand image. For datasets that do not come with hand annotations, such as some parts of COCO-WholeBody or other unlabeled image collections, we utilized MediaPipe [13], an open-source framework developed by Google. MediaPipe is specifically chosen due to its robust and superfast performance in detecting hands within images. By applying MediaPipe's hand detection capabilities, the researchers were able to identify and crop out regions of interest (ROIs) containing hands, even in the absence of explicit annotations. This step was crucial because it allowed the inclusion of a vast amount of unlabeled data into the training process, thereby increasing the diversity and quantity of training samples.

4.2 3D hand mesh estimation

We evaluated the performance of HandMIM models against several competitive methods in 3D hand mesh estimation. Our experiments demonstrate that pretraining HandMIM models significantly enhances the accuracy and quality of visualizations in 3D hand mesh estimation tasks and achieves competitive performance in multiple datasets and metrics.

4.2.1 Setups

For evaluation, we use two challenging publicly available hand pose estimation datasets, FreiHAND [22] and HO3D v2 [23], in our experiments. The FreiHAND dataset comprises 130,240 training images with a green screen or composite background and 3,960 test images with a real background. HO3Dv2 is a hand–object interaction dataset with complex occlusion that contains 77,558 hand–object 3D pose-annotated RGB images and their corresponding depth maps, 10 different human subjects (three female and seven male individuals), and 10 different objects from the YCB [38] dataset, and its evaluation process is conducted online. Note that the HO3Dv2 dataset is particularly challenging for 3D hand mesh estimation due to its focus on real-world scenarios that introduce a variety of difficulties not commonly found in more controlled datasets. The characteristics of the dataset include severe hand–object occlusions, complex interactions with objects, high-quality annotations, and the online evaluation process.

During training, we set the batch size to 128 and then crop and resize the hand image to 224×224. Random scale, translation, rotation, and color jitter are applied for data augmentation. We fine-tune our model using the Adam optimizer for 100 epochs, with a learning rate of $4e^{-5}$. Our ViT-S model achieves a real-time inference speed of 40 frames per second on a single NVIDIA V100 GPU. The detailed architecture, pretraining, and inference time are listed in Table 1.

4.2.2 Evaluation metrics

We incorporate multiple evaluation metrics for comprehensive analysis and comparison. We use *joint-point-error (JPE)* and *vertex-point-error (VPE)* to denote the average L2 distance between the ground truth and predicted keypoints and mesh vertices, respectively. We prefix the metrics with *PA* and *MP* to denote Procrustes alignment and scale-and-translation alignment. *F-scores* are defined as the harmonic means between recall and precision between two meshes given a distance threshold. We also report the *area under curve (AUC)* following common practice, which denotes the area under the percentage-of-correct-keypoints (PCK) curve for

TABLE 5 Comparisons with self-supervised methods. We train HandMIM with baselines under the same backbone and pretraining data. Results are evaluated on the FreiHAND [22] test set using the same regression head. PeCLR [18] shows an accuracy drop based on stronger vision transformers. Our HandMIM outperforms existing self-supervised methods by a large margin.

Method	Dataset	PAVPE↓	PAJPE↓	F@5↑	F@15↑
ViT-S	FH	7.10	7.21	0.697	0.978
ViT-S + iBOT [15]	FH	6.98	6.98	0.704	0.979
ViT-S + PeCLR [18]	FH	8.51	8.76	0.629	0.961
ViT-S + HandMIM	FH	**6.57**	**6.57**	**0.725**	**0.984**
ViT-S	HO3Dv2	8.71	9.05	0.571	0.965
ViT-S + iBOT [15]	HO3Dv2	8.56	8.84	0.581	0.966
ViT-S + PeCLR [18]	HO3Dv2	8.81	9.14	0.565	0.963
ViT-S + HandMIM	HO3Dv2	**8.22**	**8.57**	**0.597**	**0.970**

The bold values means the optimal performance metric in each colum.

TABLE 6 Cross-dataset analysis on HO3D and FreiHAND. Methods are trained on FreiHAND and tested on HO3D and *vice versa*.

Method	Train FH/Test HO3D		Train HO3D/Test FH	
	PAJPE↓	MPJPE↓	PAJPE↓	MPJPE↓
Hasson et al[a]. [44]	11.0	31.8	-	-
Hampali et al[a]. [23]	10.7	30.4	-	-
PeCLR [18]	13.6	-	17.8	-
TempCLR [19]	13.6	-	17.0	-
HandMIM/ViT-S	**9.9**	**30.4**	**14.1**	**29.74**

[a]Indicates the methods are trained and tested on the same dataset. Performances of [43], [19, 23], and [18] are acquired from [18,23], respectively. Note that we use the same pre-train dataset as [23] and [18] for a fair comparison.
The bold values means the optimal performance metric in each colum.

threshold values between 0 mm and 50 mm in 100 equally spaced increments. We report our evaluation results in *mm* units by default.

4.2.3 Results on FreiHAND

We compare our approach with existing methods [1, 18, 24, 27, 30, 39–41] on the mainstream FreiHAND dataset. We conduct self-supervised pretraining with HandMIM using ViT-Small (ViT-S), ViT-Base (ViT-B), and ViT-Large (ViT-L). As shown in Table 2, fine-tuning our approach using HandMIM pre-trained weights consistently improves the performance on both datasets compared to the commonly used ImageNet pre-trained weights (ViT-Small/Large-ImageNet + PyMAF), confirming the effectiveness of HandMIM pretraining. We plot the mesh and pose AUC in Figure 4. Notably, even with the lightweight ViT-Small with 22 M parameters, our approach achieves a competitive Procrustes alignment vertex-point error (PAVPE) of *6.6 mm*, which further improves to the best PAVPE of *6.2 mm* when we employ ViT-L as the backbone.

4.2.4 Results on HO3Dv2

For HO3Dv2, existing methods [29, 43]; [26] design various complex strategies via hand–object interaction information to improve the estimation accuracy. For example, HandOccNet [29] carefully designs a network to tackle severe hand occlusion. S^2Contact [43] learns hand–object contact clues to refine inaccurate pose estimations. AMVUR [26] also designs an occlusion-aware mechanism in their algorithm. In contrast, our proposed HandMIM approach trains a simple and lightweight hand-to-mesh regression model that achieves superior results without relying on complex strategies. As shown in Table 3, even when using a ViT-S model with only 22 M parameters, HandMIM achieves a PAVPE score of *8.22 mm*, surpassing other existing methods by a significant margin. We also plot the mesh and pose AUC in Figure 4. This demonstrates the robustness of our model, particularly in handling severe hand–object occlusion.

4.2.5 Visualizations

We visualized and compared the hand mesh predictions of our proposed method with some competitive methods on the test sets of FreiHAND [22] and HO3Dv2 [23] in Figures 5, 6 respectively. Compared to existing methods, our method achieved better estimation accuracy for challenging viewpoints, severe occlusion, and difficult gestures. For images in the HO3Dv2 dataset under severe hand–object occlusion, our method can capture local finger clues and infer the overall wrist pose and plausible finger positions, demonstrating its superior robustness to alternative methods, which often adopt complex mechanisms to handle those occlusions. More visualization examples are shown in Figure 7.

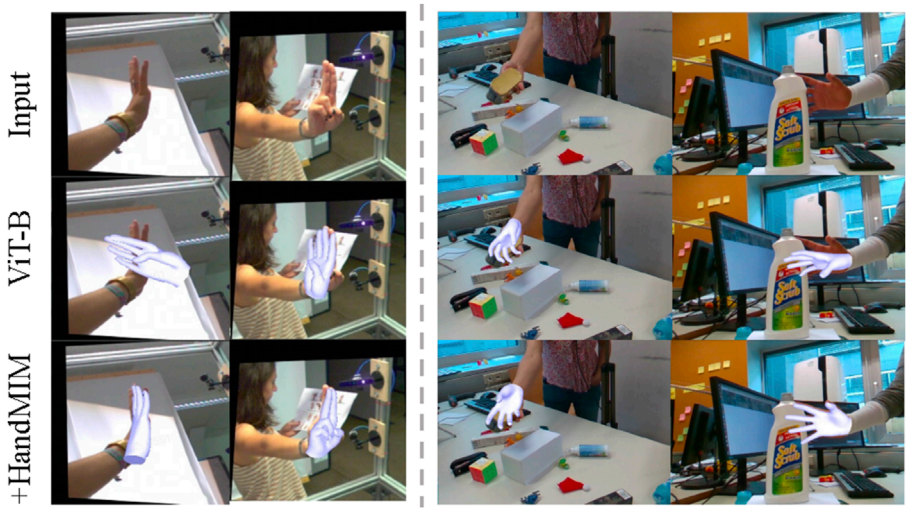


FIGURE 8
Visualizations of HandMIM pretraining. Images in the left column are from the FreiHAND [22] test set, while the images in the right column are from HO3D v2 [23]. Using ViT-Base as our backbone, we visualize the predicted mesh before (ViT-B) and after (+HandMIM) self-supervised training. We obtain more precise predictions after HandMIM pretraining.

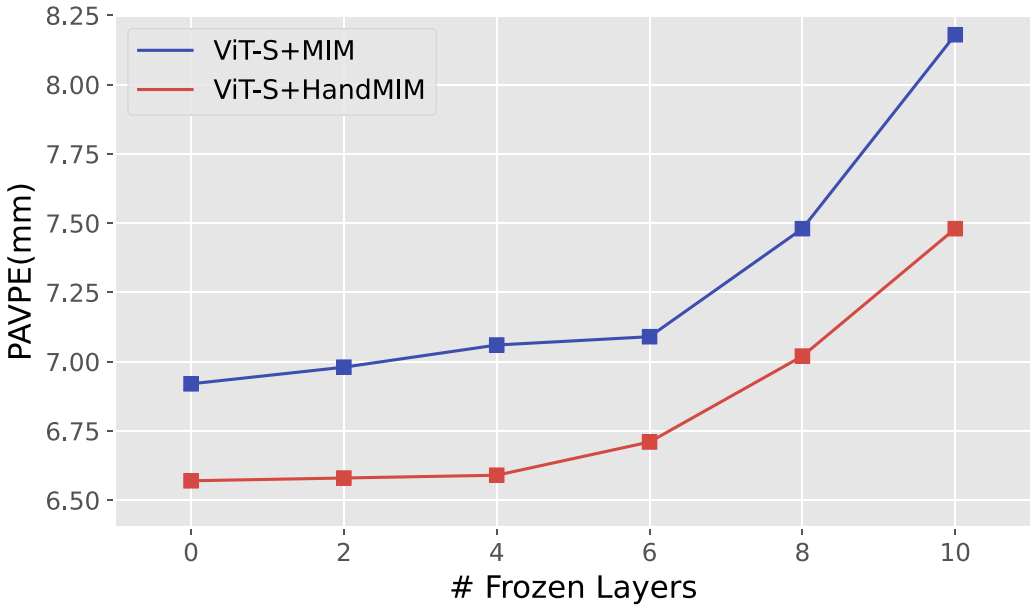


FIGURE 9
Partial fine-tuning performance comparison between pre-trained weight from mainstream masked image modeling methods and our HandMIM with ViT-Small as the backbone. We use the FreiHAND [22] test set as a metric and adopt iBOT [15] as baselines of MIM methods. We gradually freeze different numbers of blocks to reveal the feature generalizability learned from pretraining.

4.3 Ablation study

In this subsection, we presented a series of convincing analysis experiments and ablations to evaluate the effectiveness of HandMIM. We demonstrated the superiority of our method against existing self-supervised methods through comprehensive comparisons. To assess the generalizability of our method, we perform linear prob, cross-dataset, partial fine-tuning analysis, and visualizations of HandMIM.

4.3.1 Linear probe for keypoint regression

As we enforce the pose-sensitive knowledge in our latent feature, we can adopt the linear prob strategy to validate their effectiveness. Linear probing is an intuitionistic method for a self-supervised-trained model to show the quality of representation learning by *freezing* the pre-trained backbone and using a simple MLP layer to predict the output. We use the 2.5D joint representation to regress 2D and 3D keypoints jointly. Concretely, we learned two 3-layer multilayer perceptrons (MLPs) to predict 2D keypoints

TABLE 7 Ablation studies. We perform ablations on the loss design of HandMIM. Specifically, we remove all three critical losses $\mathcal{L}_{pose}$, $\mathcal{L}_{patch}$, and $\mathcal{L}_{recon}$, in order. We conduct experiments based on the ViT-Small backbone and FreiHAND [22] datasets under the same setting as the main experiments. We can conclude that every self-supervised learning target by our design is effective.

$\mathcal{L}_{pose}$	$\mathcal{L}_{patch}$	$\mathcal{L}_{recon}$	PAVPE↓	PAJPE↓	F@5↑	F@15↑
✓	✓	✓	**6.57**	**6.57**	**0.725**	**0.984**
✗	✓	✓	6.89	6.87	0.707	0.981
✓	✗	✓	6.90	6.85	0.708	0.981
✓	✓	✗	6.76	6.74	0.715	0.982

The bold values means the optimal performance metric in each colum.

TABLE 8 Evaluation results on the scalability of HandMIM on unlabeled images.

Dataset proportion	25%	50%	100%
PAVPE	7.1	6.78	**6.57**

The bold values means the optimal performance metric in each colum.

and 1D relative depth, respectively. The resulting 3D keypoints are calculated according to the camera's intrinsic parameters. We trained our MLP layer on FreiHAND [22] and split the training and validation set afterward [18]. Note that we only train for 10 epochs with the AdamW optimizer and set the initial lr as 0.01. We report the predicted 2D keypoint error and 3D joint error, as shown in Table 4, which demonstrates: 1) our HandMIM can learn better features than previous self-supervised methods such as PeCLR [18]. 2) The geometric equivalence ingrained in our features plays an important role in predicting accurate 2D/3D keypoints when compared against iBOT [15], which does not encode any pose-aware mechanism in their self-supervised framework and thus obtains a much higher prediction error rate using their features.

4.3.2 Comparisons with alternative self-supervised learning methods

As shown in Table 5, we compared the performance of our proposed pose-aware method for 3D hand mesh estimation with two representative self-supervised learning methods, the mainstream masked image modeling method iBOT [15] and the contrastive-learning-based method PeCLR [18]. We conducted these comparisons using *the same* ViT-Small backbone and *the same* amount of training data. Our results indicate that HandMIM outperforms iBOT, which is a representative MIM method used for visual recognition tasks. Furthermore, we observed that the contrastive-learning-based method is not suitable for stronger vision transformer architectures, resulting in a significant accuracy drop. These findings demonstrate the superiority of our proposed method over existing self-supervised learning methods for hand estimation tasks.

4.3.3 Cross-dataset validation

To evaluate the generalizability of our proposed method, we conducted a cross-data validation on 3D hand mesh estimation tasks. Specifically, we fine-tuned our model on the training set of FreiHAND and evaluated its performance on the test set of HO3D v2 and *vice versa*. Our results, presented in Table 6, demonstrate significant improvements compared to existing self-supervised methods such as PeCLR [18] or TempCLR [19], which indicates the superiority of our approach. Notably, our method even outperforms some recent fully supervised methods [44]; [23] when evaluated on the HO3D v2 test set.

4.3.4 Visualization of HandMIM pretraining

We are curious about the effects of hand pose estimation after self-supervised pretraining and visualize the results before and after pre-train in Figure 8. The findings demonstrate that HandMIM pretraining enhances the resilience of 3D hand mesh estimation tasks, indicating the beneficial effects of pretraining. Specifically, the results highlight the positive influence of pretraining on the robustness of hand pose estimation. More visualization examples are shown in the supplementary document.

4.3.5 Partial fine-tuning

To further explore the efficacy of the learned features, we employ a partial fine-tuning method based on the protocol proposed in [16]. We sequentially freeze the first several layers while fine-tuning the remaining transformer blocks. The results are presented in Figure 9, which indicates that when we freeze around half of the layers (i.e., 4 of 12), the HandMIM approach shows only a minor decrease in accuracy compared to mainstream masked image modeling methods. Moreover, when we freeze eight or more layers, the performance gap between our method and most fully supervised methods becomes more pronounced. These findings suggest that our approach can effectively learn multi-level hand representations via our multi-level learning approach.

4.3.6 Ablations on self-supervised loss designs

The pose-aware $\mathcal{L}_{pose}$, token-level $\mathcal{L}_{patch}$, and pixel-level $\mathcal{L}_{recon}$ losses in our HandMIM framework collaborate to capture distinct levels of representations from input images in a self-supervised manner. To verify the effectiveness of our design, we conduct experiments by removing one of the losses from our framework, as shown in Table 7. The results demonstrate that the removal of any one of the losses results in a decrease in overall precision, justifying the importance of our multi-level loss design. Therefore, our approach can effectively leverage various levels of information to enhance the robustness and accuracy of hand mesh estimation tasks.

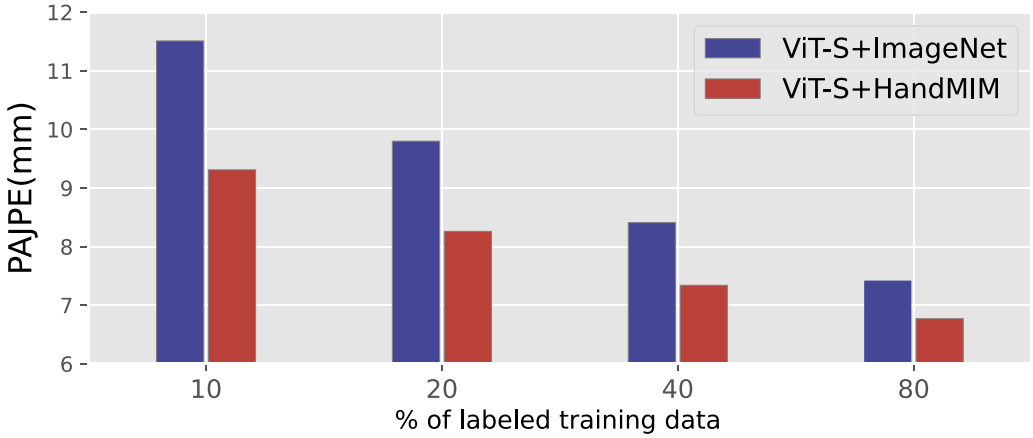


FIGURE 10
Scalability of HandMIM pretraining. We pre-trained the backbone ViT-S with two strategies: (i) ViT-S + ImageNet: training ViT-S in a supervised approach with labeled data on ImageNet. (ii) ViT-S + HandMIM: training ViT-S in the self-supervised approach described above with unlabeled data. Both models are connected with PyMAF [20] and fine-tuned for mesh estimation. The PAJPE metric is evaluated for both.

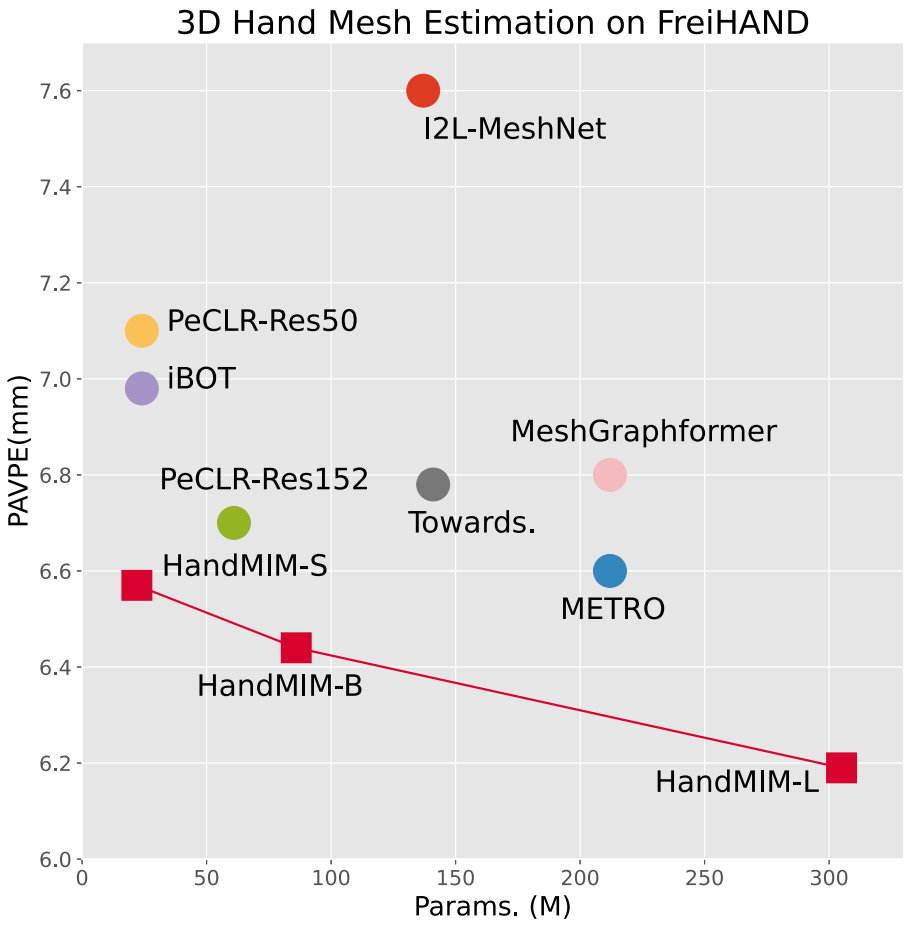


FIGURE 11
Performance–parameter trade-off of mainstream 3D hand mesh estimation methods on the FreiHAND [22] test set. We perform vertex-point-error after Procrustes alignment (lower is better). Our proposed HandMIM achieves better trade-offs in various model sizes under the standard ViT backbone.

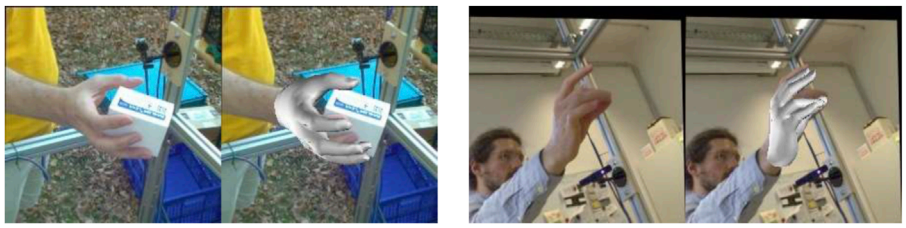


FIGURE 12
The figure demonstrates some common failure cases. **(A)** Complex hand–object interactions. **(B)** Extreme occlusion between fingers.

4.3.7 Scalability of HandMIM pretraining

To justify the scalability of HandMIM on unlabeled images, we conduct pretrain experiments with a certain proportion of the full dataset. Table 8 shows our evaluation results. We obtain better performance with more unlabeled data (6.78*mm* on 50% pretraining data and 6.57*mm* on the full dataset), indicating that HandMIM holds the potential to further boost performance with abundant unlabeled hand images. We then assessed the accuracy of our model's estimations by varying the proportion of labeled data used for fine-tuning, specifically at ratios of 10%, 20%, 40%, and 80%. As indicated by the red bins in Figure 10, HandMIM demonstrates remarkable scalability: performance improves with an increase in the amount of labeled data. The model's estimation error decreases exponentially, aligning with the scaling law as outlined in Tan and Le [45]. Additionally, we also evaluated a model that was supervised and pre-trained with labeled data. The results of this evaluation, represented by blue bins in Figure 10, show that our self-supervised training approach outperforms the traditional method, reducing the error by approximately 40%~50%. This finding underscores the significant advantages of our approach in regression tasks related to hand pose estimation and highlights its reduced reliance on labeled training data.

Furthermore, we evaluated the performance of HandMIM across different scales of parameters, specifically using vision transformer small (ViT-S), base (ViT-B), and large (ViT-L) configurations. The results, depicted in Figure 11, demonstrate two key insights: (i) the performance of HandMIM is enhanced with the increase in parameter size, and (ii) HandMIM consistently outperforms other methods when matched for parameter scale.

5 Limitations

HandMIM has demonstrated competitive performance across various datasets. Nevertheless, there are still situations where the model might struggle, as shown in Figure 12.

(1) Complex hand–object interactions. When hands are engaged in complex interactions with objects, the model must infer the occluded parts of the hand based on limited visual cues. Although HandMIM shows promise in these scenarios, there is room for improvement, especially when the interaction involves intricate movements or unusual poses that the model has not encountered during training.

(2) Extreme occlusions. Despite advancements in handling occlusions, extremely occluded hands—where large portions of the hand are hidden or covered by other fingers—remain challenging. In these cases, the model may lack sufficient visible information to accurately reconstruct the hand mesh, leading to increased prediction errors.

(3) Dataset variability. The effectiveness of HandMIM depends on the diversity and quality of the pretraining datasets. If the datasets used for pretraining do not adequately cover certain types of hand poses or backgrounds, the model's ability to generalize to unseen data may be compromised.

Accordingly, while HandMIM excels in many aspects of 3D hand mesh estimation, it faces challenges related to the quality of pseudo-keypoint generation and potential failures in extreme occlusion scenarios. Addressing these limitations will be essential for further enhancing the robustness and applicability of our model.

6 Conclusion

In this study, we have introduced HandMIM, a novel self-supervised pretraining strategy specifically designed for 3D hand mesh regression from monocular RGB images. Our approach leverages masked image modeling in conjunction with a multi-granularity strategy and pseudo-keypoint alignment within a teacher–student framework, utilizing self-distillation to learn comprehensive representations. By integrating these components, HandMIM achieves significant improvements over traditional supervised methods, reducing errors by approximately 40%–50%. This underscores the effectiveness of our method in requiring less reliance on labeled training data. The experiments conducted across various datasets highlight HandMIM's robustness and adaptability, particularly under challenging conditions such as severe occlusions. Notably, it achieved an 8.00 mm PAVPE on the HO3Dv2 test set, outperforming many specialized architectures. Furthermore, scalability tests on unlabeled images demonstrated that increasing the dataset proportion from 25% to 100% progressively decreased the PAVPE from 7.1 mm to 6.57 mm, indicating improved performance with more data. Additionally, evaluating HandMIM using different parameter scales revealed that its performance is enhanced with larger models, and it consistently outperforms other methods when matched for parameter scale. These results suggest that HandMIM not only benefits from deeper networks but also

maintains superior performance relative to alternative approaches at similar model sizes. For future work, we propose several directions:

- **Exploring the integration of temporal information.** Current research focuses on single-image-based estimation. Expanding HandMIM to incorporate sequential video frames could enhance pose estimation accuracy and stability.
- **Addressing dual-hand interactions.** The current scope is limited to single-hand poses. Future efforts should consider extending the model to handle scenarios involving two interacting hands.
- **Generalizing to related tasks.** Investigating how the principles behind HandMIM can be applied to other human-centric regression and estimation tasks could broaden its impact.

Overall, HandMIM represents a significant advancement in self-supervised learning for 3D hand pose estimation, setting a new benchmark and opening avenues for further exploration.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material; further inquiries can be directed to the corresponding author.

Author contributions

YL: Writing–original draft, Funding acquisition, Formal analysis, Investigation. CW: Writing–original draft, Conceptualization, Data curation, Methodology. HW: Project administration, Supervision, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. This work was supported by Science and Technology Research Project of Jiangxi Provincial Department of Education (No. GJJ2203419).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

1. Lin K, Wang L, Liu Z. *Mesh graphormer*. IEEE International Conference on Computer Vision ICCV (2021). p. 12939–48.

2. Hampali S, Sarkar SD, Rad M, Lepetit V. 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* (2022). p. 11090–100.

3. Cai G, Zheng X, Guo J, Gao W. Real-time identification of borehole rescue environment situation in underground disaster areas based on multi-source heterogeneous data fusion. *Saf Sci* (2025) 181:106690. doi:10.1016/j.ssci.2024.106690

4. Jin W, Tian X, Shi B, Zhao B, Duan H, Wu H. Enhanced uav pursuit-evasion using boids modelling: a synergistic integration of bird swarm intelligence and drl. *Comput Mater & Continua* (2024) 80:3523–53. doi:10.32604/cmc.2024.055125

5. Hu Z, Qi W, Ding K, Liu G, Zhao Y. An adaptive lighting indoor vslam with limited on-device resources. *IEEE Internet Things J* (2024) 11:28863–75. doi:10.1109/JIOT.2024.3406816

6. Chen J, Li T, Zhang Y, You T, Lu Y, Tiwari P, et al. Global-and-local attention-based reinforcement learning for cooperative behaviour control of multiple uavs. *IEEE Trans Vehicular Technology* (2024) 73:4194–206. doi:10.1109/TVT.2023.3327571

7. Chen J, Du C, Zhang Y, Han P, Wei W. A clustering-based coverage path planning method for autonomous heterogeneous uavs. *IEEE Trans Intell Transportation Syst* (2021) 23:25546–56. doi:10.1109/tits.2021.3066240

8. Zhu P, Pan Z, Liu Y, Tian J, Tang K, Wang Z. A general black-box adversarial attack on graph-based fake news detectors. In: *International joint conference on artificial intelligence (IJACI 2024)* (2024).

9. Zhu P, Fan Z, Guo S, Tang K, Li X. Improving adversarial transferability through hybrid augmentation. *Comput & Security* (2024) 139:103674. doi:10.1016/j.cose.2023.103674

10. Guo S, Li X, Zhu P, Mu Z. Ads-detector: an attention-based dual stream adversarial example detection method. *Knowledge-Based Syst* (2023) 265:110388. doi:10.1016/j.knosys.2023.110388

11. Kulon D, Guler RA, Kokkinos I, Bronstein MM, Zafeiriou S. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020). p. 4990–5000.

12. Spurr A, Iqbal U, Molchanov P, Hilliges O, Kautz J. Weakly supervised 3d hand pose estimation via biomechanical constraints. In: *Proceedings of the European conference on computer vision*. Springer (2020). p. 211–28.

13. Lugaresi C, Tang J, Nash H, McClanahan C, Uboweja E, Hays M, et al. (2019). Mediapipe: a framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*

14. Kolesnikov A, Dosovitskiy A, Weissenborn D, Heigold G, Uszkoreit J, Beyer L, et al. (2021). *An image is worth 16x16 words: transformers for image recognition at scale*

15. Zhou J, Wei C, Wang H, Shen W, Xie C, Yuille A, et al. ibot: image bert pre-training with online tokenizer. In: *International conference on learning representations (ICLR)* (2022).

16. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. Masked autoencoders are scalable vision learners. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). p. 16000–9.

17. Oord Avd, Li Y, Vinyals O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018). doi:10.48550/arXiv.1807.03748

18. Spurr A, Dahiya A, Wang X, Zhang X, Hilliges O. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In: *IEEE international conference on computer vision (ICCV)* (2021). p. 11230–9.

19. Ziani A, Fan Z, Kocabas M, Christen S, Hilliges O. Tempclr: reconstructing hands via time-coherent contrastive learning. In: *International conference on 3D vision (3DV)* (2022).

20. Zhang H, Tian Y, Zhou X, Ouyang W, Liu Y, Wang L, et al. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In: *2021 IEEE/CVF international conference on computer vision (ICCV)* (2021). p. 11426–36. doi:10.1109/ICCV48922.2021.01125

21. Romero J, Tzionas D, Black MJ. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans Graph* (2017) 36:1–17. doi:10.1145/3130800.3130883

22. Zimmermann C, Ceylan D, Yang J, Russell B, Argus M, Brox T. *Freihand: a dataset for markerless capture of hand pose and shape from single rgb images*. IEEE International Conference on Computer Vision ICCV (2019). p. 813–22.

23. Hampali S, Rad M, Oberweger M, Lepetit V. Honnotate: a method for 3d annotation of hand and object poses. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020). p. 3196–206.

24. Moon G, Lee KM. I2l-meshnet: image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In: *Proceedings of the European conference on computer vision* (2020).

25. Xie P, Xu W, Tang T, Yu Z, Lu C. Ms-mano: enabling hand pose tracking with biomechanical constraints. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2024). p. 2382–92.

26. Jiang Z, Rahmani H, Black S, Williams BM. 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* (2023). p. 758–67.

27. Vasu PKA, Gabriel J, Zhu J, Tuzel O, Ranjan A. Fastvit: a fast hybrid vision transformer using structural reparameterization. In: *Proceedings of the IEEE/CVF international conference on computer vision* (2023). p. 5785–95.

28. Wang C, Zhu F, Wen S. Memahand: exploiting mesh-mano interaction for single image two-hand reconstruction. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023). p. 564–73.

29. Park J, Oh Y, Moon G, Choi H, Lee KM. Handoccnet: occlusion-robust 3d hand mesh estimation network. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). p. 14682–92.

30. Chen X, Liu Y, Dong Y, Zhang X, Ma C, Xiong Y, et al. Mobrecon: mobile-friendly hand mesh reconstruction from monocular image. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022). p. 20544–54.

31. Pavlakos G, Shan D, Radosavovic I, Kanazawa A, Fouhey D, Malik J. Reconstructing hands in 3D with transformers. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2024).

32. Zimmermann C, Argus M, Brox T. Contrastive representation learning for hand shape estimation. In: *DAGM German conference on pattern recognition*. Springer (2021). p. 250–64.

33. Bao H, Dong L, Piao S, Wei F. BEiT: BERT pre-training of image transformers. In: *International conference on learning representations* (2022).

34. He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020). p. 9729–38.

35. Grill J-B, Strub F, Altché F, Tallec C, Richemond P, Buchatskaya E, et al. Bootstrap your own latent-a new approach to self-supervised learning. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)* (2020) 33:21271–84. doi:10.5555/3495724.3496201

36. Caron M, Touvron H, Misra I, Jégou H, Mairal J, Bojanowski P, et al. *Emerging properties in self-supervised vision transformers*. IEEE International Conference on Computer Vision ICCV (2021). p. 9650–60.

37. Jin S, Xu L, Xu J, Wang C, Liu W, Qian C, et al. Whole-body human pose estimation in the wild. In: *Proceedings of the European conference on computer vision*. Springer (2020). p. 196–214.

38. Xiang Y, Schmidt T, Narayanan V, Fox D (2018). *Posecnn: a convolutional neural network for 6d object pose estimation in cluttered scenes*

39. Chen P, Chen Y, Yang D, Wu F, Li Q, Xia Q, et al. I2uv-handnet: image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In: *IEEE international conference on computer vision (ICCV)* (2021). p. 12929–38.

40. Zhang X, Huang H, Tan J, Xu H, Yang C, Peng G, et al. Hand image understanding via deep multi-task learning. In: *IEEE international conference on computer vision (ICCV)* (2021). p. 11281–92.

41. Tang X, Wang T, Fu C-W. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In: *IEEE international conference on computer vision (ICCV)* (2021). p. 11698–707.

42. Lim GM, Jatesiktat P, Ang WT. Mobilehand: real-time 3d hand shape and pose estimation from color image. In: *International conference on neural information processing*. Springer (2020). p. 450–9.

43. Liu S, Jiang H, Xu J, Liu S, Wang X. Semi-supervised 3d hand-object poses estimation with interactions in time. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021).

44. Hasson Y, Varol G, Tzionas D, Kalevatykh I, Black MJ, Laptev I, et al. Learning joint reconstruction of hands and manipulated objects. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (United States: Computer Vision Foundation (CVF) and IEEE) (2019). p. 11807–16.

45. Tan M, Le Q. EfficientNet: rethinking model scaling for convolutional neural networks. In: K Chaudhuri, R Salakhutdinov, editors. *Proceedings of the 36th international conference on machine learning*. PMLR, vol. 97 of Proceedings of Machine Learning Research (2019). p. 6105–14.