Fast-HaMeR: Boosting Hand Mesh Reconstruction using Knowledge Distillation

Hunain Ahmed Jillani[1] [a], Ahmed Tawfik Aboukhadra[1,2] [b], Ahmed Elhayek[3] [c], Jameel Malik[4] [d], Nadia Robertini[2] [e], Didier Stricker[1,2] [f]

[1] *RPTU, Kaiserslautern, Germany;* [2] *DFKI-AV, Kaiserslautern, Germany;* [3] *UPM, Saudi Arabia;* [4] *NUST-SEECS, Islamabad, Pakistan;*

Keywords: 3D Hand Reconstruction, Pose Estimation, Knowledge Distillation, Real-Time Inference.

Abstract: Fast and accurate 3D hand reconstruction is essential for real-time applications in VR/AR, human-computer interaction, robotics, and healthcare. Most state-of-the-art methods rely on heavy models, limiting their use on resource-constrained devices like headsets, smartphones, and embedded systems. In this paper, we investigate how the use of lightweight neural networks, combined with Knowledge Distillation, can accelerate complex 3D hand reconstruction models by making them faster and lighter, while maintaining comparable reconstruction accuracy. While our approach is suited for various hand reconstruction frameworks, we focus primarily on boosting the HaMeR model, currently the leading method in terms of reconstruction accuracy. We replace its original ViT-H backbone with lighter alternatives, including MobileNet, MobileViT, ConvNeXt, and ResNet, and evaluate three knowledge distillation strategies: output-level, feature-level, and a hybrid of both. Our experiments show that using lightweight backbones that are only 35% the size of the original achieves $1.5x$ faster inference speed while preserving similar performance quality with only a minimal accuracy difference of $0.4mm$. More specifically, we show how output-level distillation notably improves student performance, while feature-level distillation proves more effective for higher-capacity students. Overall, the findings pave the way for efficient real-world applications on low-power devices. The code and models are publicly available under https://github.com/hunainahmedj/Fast-HaMeR.

1 INTRODUCTION

A key challenge in computer vision and graphics is estimating the 3D pose and shape of the human hand from visual inputs like images, videos, or depth maps. Accurate and real-time 3D hand reconstruction is essential for many computer vision and graphics applications such as augmented and virtual reality, human-computer interaction, sign language recognition, and human behavior analysis. These applications require precise hand modeling to interpret hand articulations and understand gestures. This problem is challenging because human hands are highly articulated, self-occluding, and vary significantly across individuals. In recent years, there have been significant advances in hand reconstruction (Chatzis et al., 2020), which have

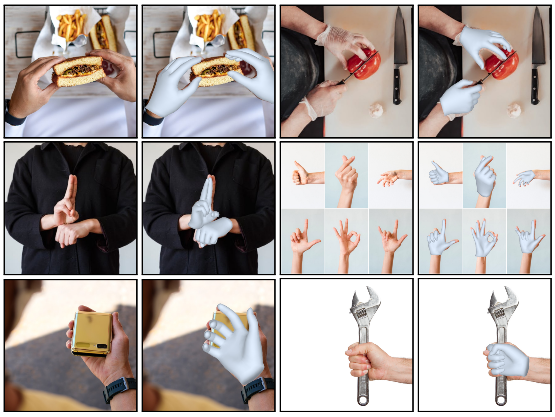

Figure 1: **Real-time monocular 3D hand mesh reconstruction using lightweight student networks.** We present a Knowledge Distillation approach for training lightweight networks that accelerate 3D hand reconstruction without compromising quality. Our best-performing network achieves real-time inference with only a modest drop in reconstruction quality. The Figure showcases qualitative results across challenging scenarios, including mutual occlusion, complex hand poses, and interactions with various objects.

[a] https://orcid.org/0009-0000-9286-898X

[b] https://orcid.org/0009-0005-6111-1366

[c] https://orcid.org/0000-0002-5919-7202

[d] https://orcid.org/0000-0002-7528-7240

[e] https://orcid.org/0009-0007-2861-3326

[f] https://orcid.org/0000-0002-5708-6023

been made possible with the use of modern Deep Learning techniques (Pavlakos et al., 2024), (Li et al., 2024), (Ge et al., 2019), (Lin et al., 2021), (Malik et al., 2020), (Aboukhadra et al., 2023), (Dong et al., 2024), (Potamias et al., 2025). However, many state-of-the-art hand mesh recovery methods remain computationally expensive (Pavlakos et al., 2024; Potamias et al., 2025), as they rely on large-scale networks with transformer-based architectures, making real-time performance difficult to achieve.

To accelerate state-of-the-art 3D hand mesh reconstruction, we propose a Knowledge Distillation (KD) framework for hand mesh reconstruction that transfers knowledge from a large, accurate teacher model to a lightweight student network. While applicable to various models, we focus on HaMeR, a transformer-based architecture with a ViT-H backbone. Our approach replaces this backbone with more efficient alternatives—such as MobileNet, MobileViT, ConvNeXt, and ResNet—while preserving reconstruction quality. During training, the student learns to mimic the teacher at multiple levels: final outputs, intermediate features, or both. This distillation strategy enables the student model to achieve near-teacher performance at a fraction of the computational cost, making it suitable for real-time applications on resource-limited devices.

Through practical experiments, we demonstrate that applying Knowledge Distillation, along with using lightweight backbones, leads to improvements in runtime performance with comparable reconstruction accuracy. This paper makes the following key contributions to the domain of 3D hand reconstruction:

- Demonstrating the effectiveness of common distillation losses (output-level and feature-level) for the task of hand mesh reconstruction.
- Proposing a lightweight alternative to the state-of-the-art HaMeR model, achieving 35% of its size, 1.5× faster runtime, and with an accuracy difference of just 0.4*mm*.
- Analyzing how student architectures respond to different distillation losses, showing that feature-level supervision benefits higher-capacity models, while output-level losses are more effective for smaller or structurally aligned ones.

2 RELATED WORKS

In recent years, several works have been proposed for monocular 3D hand reconstruction that utilize depth scans (Wu et al., 2018; Ge et al., 2018; Aboukhadra et al., 2024), RGB frames (Rong et al., 2021; Lin et al., 2021; Park et al., 2022; Pavlakos et al., 2024), or a hybrid RGB-D approach (Makris and Argyros, 2015; Sridhar et al., 2016; Malik et al., 2020). These works can be broadly divided into two categories: parametric model-driven approaches and direct mesh recovery approaches. Arguably, the primary distinction between these approaches lies in the use of parametric hand models (Romero et al., 2017; Loper et al., 2015; Anguelov et al., 2023; Li et al., 2022). These hand models provide prior anatomical constraints to facilitate the reconstruction of the hand. Parametric model-driven approaches learn a low-dimensional mapping from image features to hand model parameters, which allow them to generate a 3D mesh, while the direct mesh recovery approaches choose to regress the mesh directly without the use of parametric models.

Previous works, such as HAMR (Zhang et al., 2019b), introduced a weakly-supervised approach to recover hand mesh and pose from a monocular hand image using silhouette consistency loss. FrankMocap (Rong et al., 2021) followed with a modular whole-body pose estimation system, incorporating separate modules for face, hand, and body. To address occlusion, HandOccNet (Park et al., 2022) leveraged transformers, utilizing an attention mechanism to handle occlusions via the correlations between visible and occluded regions. Supervised learning methods, due to data constraints (i.e., lack of diverse, real-world data), can often fail to generalize well to unconstrained, real-world images. To address this, (Hasson et al., 2021) introduced a learning-free optimization-based approach to reconstructing 3D hand-object interactions, using only RGB video frames.

Direct mesh recovery methods tend to bypass parametric models by directly regressing 3D hand mesh vertices. METRO (Lin et al., 2020) is a fully transformer-based human pose estimation and mesh reconstruction method, which features a multi-layer transformer encoder. The core mechanism is Self-attention (Vaswani et al., 2017), which enables joint modeling of vertex-vertex and vertex-joint interactions, and it can learn both short and long-range interactions. Mesh Graphormer (Lin et al., 2021) combines the strengths of self-attention (Vaswani et al., 2017), and graph convolutions for human body mesh reconstruction. THOR-Net (Aboukhadra et al., 2023) introduced the first framework for estimating 3D pose and shape of two hands interacting with an object, along with the texture of the vertices in the resulting hand meshes, through combining the strengths of Graph Convolutional Networks (GCNs) and Transformers with self-supervision. Recently, Hamba

(Dong et al., 2024) introduced a graph-guided bi-scanning (GBS) Mamba-based framework (Gu and Dao, 2023) for single-view 3D hand mesh reconstruction.

HaMeR (Pavlakos et al., 2024), the framework we built upon, represents a new state-of-the-art approach for hand mesh reconstruction from a single RGB image. It combines a ViT-H (Dosovitskiy et al., 2021) backbone with a transformer decoder to regress MANO (Romero et al., 2017) parameters. The success of the method can be attributed to scaling the architecture and training data, leading to improved generalization. Despite outperforming all previous RGB-based approaches, scaling the HaMeR network came at the cost of runtime and resource efficiency, which makes it unsuitable for real-time applications. Therefore, in our work, we aim to improve runtime and resource efficiency by utilizing HaMeR as a reference for training lighter networks.

2.1 Knowledge Distillation for Acceleration

Knowledge Distillation (KD) (Hinton et al., 2015) is a famous technique for reducing network complexity that transfers the knowledge from large teacher networks to smaller student networks. Distillation was used recently to compress and accelerate Large Language Models (LLMs) (Ko et al., 2024; Gu et al., 2024; Hsieh et al., 2023). On the other hand, KD has been only marginally explored in the context of 3D reconstruction and pose estimation problems. Works such as (Zhang et al., 2019a) and (Hwang et al., 2020) focus on full-body pose tasks, distilling 2D or 3D joint information without addressing dense geometric structures. (Zhang et al., 2020) and, more recently (Capistrano et al., 2025), apply KD to hand pose estimation, transferring pose-level knowledge into compact models. In contrast, our method applies both feature and output distillation to compress a full mesh reconstruction model, preserving high-quality geometry beyond joint estimation.

3 METHOD

Our goal is to develop a fast, lightweight, and efficient method for parametric 3D hand reconstruction that matches the performance of slower, resource-intensive models. While our framework is designed to generalize across different reconstruction algorithms, we focus on accelerating the state-of-the-art HaMeR (Pavlakos et al., 2024) model without loss of generality. We achieve this task by training smaller and lighter student networks that learn from the original high-capacity teacher network (i.e., HaMeR) through the use of Knowledge Distillation. At a high level, the framework consists of the original HaMeR as a teacher network $\boldsymbol{T}$ with frozen weights, and a smaller and lighter network that acts as a student $\boldsymbol{S}$.

3.1 Teacher Model (HaMeR)

HaMeR (Pavlakos et al., 2024) features a fully transformer-based design, in particular, it adopts a Huge Vision Transformer (ViT-H) (Dosovitskiy et al., 2021) as its backbone. The output token sequence (i.e. image features) is fed into a transformer-decoder head responsible for regressing the MANO parameters (θ, β) and camera parameters π, where $\theta \in \mathbb{R}^{48}, \beta \in \mathbb{R}^{10}$. The transformer head utilizes self-attention and processes a single token while cross-attending to the backbone output tokens (Pavlakos et al., 2024). The regressed parameters are fed into the MANO head to obtain a 3D mesh of the hand along with the 3D joint locations K_T^{3D} for 21 keypoints, where $K_T^{3D} \in \mathbb{R}^{3\times N}$, and $N = 21$. The camera parameters includes a 3D translation vector $(t \in \mathbb{R}^3)$ necessary to project the 3D mesh and the 3D joint locations onto the image plane and in obtaining the 2D joint locations $K_T^{2D} \in \mathbb{R}^{2\times N}$.

In order to utilize datasets with no 3D annotations, HaMeR applies a re-projection loss L_{2D} between the 2D projection of the 3D predicted joint locations and 2D ground-truth joint locations. The Keypoint 3D Loss L_{3D} measures the distance between the predicted 3D joint locations and the ground-truth joint locations, in order to encourage the model to reconstruct the correct 3D hand pose. For datasets that provide MANO (Romero et al., 2017) annotations, the MANO parameter loss L_{MANO} penalizes the difference between the predicted MANO parameters and the ground-truth parameters, ensuring the model learns to predict the correct parameters. The combined loss of L_{2D}, L_{3D}, and L_{MANO} will be referred to as $\mathcal{L}_{GT}$.

3.2 Student Model

The student network retains the overall architecture of the teacher model; however, in our work, we replace the larger ViT-Huge backbone (Dosovitskiy et al., 2021) with smaller networks, namely, *MobileNet* (Howard et al., 2017), *MobileViT* (Mehta and Rastegari, 2021), *ConvNeXt* (Liu et al., 2022), and *ResNet* (He et al., 2015). We utilize these networks as backbones in our student network S, aiming to evaluate the trade-offs between reconstruction accu-

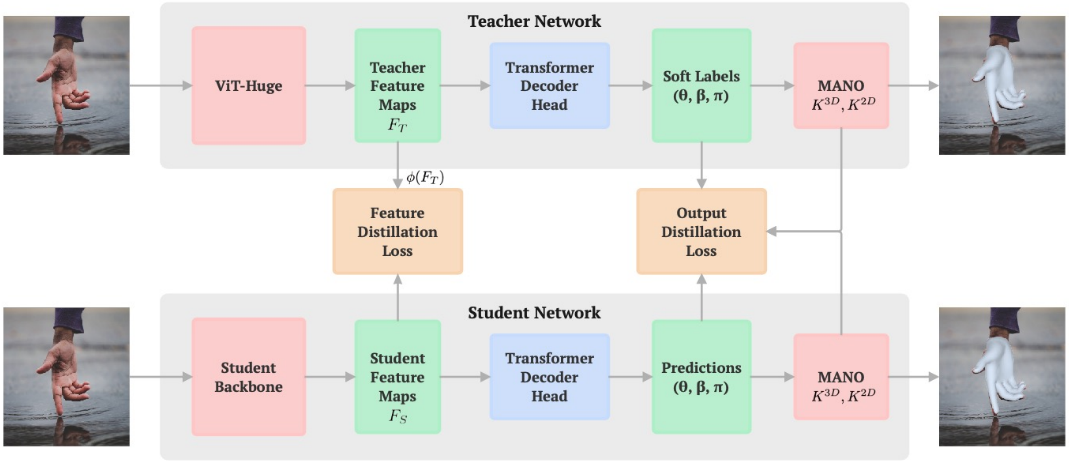

Figure 2: High-level overview of the teacher-student architecture, only relevant distillation losses are used depending on the KD level. $\phi(F_T)$ refers to a 1x1 convolution to project the dimensions of the teacher's features to match those of the student's for feature-level distillation. The teacher network is used only during training; at inference time, only the trained student network is used.

racy and inference speed. These choices represent a few design philosophies, such as lower parameter counts, depthwise convolutions (Howard et al., 2017), lightweight hybrid transformer-based models (Mehta and Rastegari, 2021), and residual connections (He et al., 2015). Furthermore, the criteria for choosing this subset of backbones is their relative performance on the ImageNet-1K dataset (Deng et al., 2009). Additionally, we considered their number of parameters stated in the torchvision library, and their runtime efficiency.

3.3 Knowledge Distillation

During the training process, an input image is passed through both the student $\boldsymbol{S}$ and the teacher network $\boldsymbol{T}$. Both $\boldsymbol{T}$ and $\boldsymbol{S}$ produce intermediate feature maps (i.e, F_T and F_S) and final predictions (i.e, $\hat{Y}_T$ and $\hat{Y}_S$) where $\hat{Y} = \{K^{3D}, K^{2D}, \theta, \beta, \pi\}$. We enforce similarity between the predictions on three different levels: output-level, feature-level, and combined distillation. We then evaluate each approach to determine which performs best.

In the first approach, we calculate distillation losses between $\hat{Y}_T$ and $\hat{Y}_S$ (i.e., *output-level distillation*). In the second approach we calculate distillation losses between F_T and F_S (i.e, *feature-level distillation*), and finally, between both F_T and F_S, and $\hat{Y}_T$ and $\hat{Y}_S$ (i.e., *combined distillation*). Figure 2 shows the different distillation methods. By enforcing these similarities, we allow the smaller student model to learn how the bigger teacher model learns and represents the inputs internally, at the same time decreasing output latency due to the smaller size of $\boldsymbol{S}$. We finally evaluate each distillation scheme to select the best approach.

Output-level distillation In the case of *output-level distillation*, to enforce similarity between $\hat{Y}_T$ and $\hat{Y}_S$, we use the same loss terms from the original network (specified in Section 3.1) but substitute the ground truth values for the teacher's predictions. The total loss for output-level distillation includes the original losses $\mathcal{L}_{GT}$ on the ground truths in addition to the losses between the student's predictions $\hat{Y}_S$ and the teacher's predictions $\hat{Y}_T$. We also use λ_{KD} to control how much distillation losses contribute to the total loss.

$$\mathcal{L}_{total,out} = \mathcal{L}_{GT} + \lambda_{KD}\mathcal{L}_{KD,out} \tag{1}$$

$$\mathcal{L}_{KD,out} = \|\hat{K}_S^{3D} - \hat{K}_T^{3D}\|_2^2 + \|\hat{K}_S^{2D} - \hat{K}_T^{2D}\|_2^2 + \|\hat{\Theta}_S - \hat{\Theta}_T\|_2^2 \tag{2}$$

Feature-level distillation Knowledge Distillation on the feature maps (i.e., *feature-level distillation*) enables the student $\boldsymbol{S}$ to learn feature representations similar to those of the teacher network $\boldsymbol{T}$. Since the output feature maps F_S and F_T may differ in channel dimensionality, we apply a learnable 1x1 convolution ϕ to project the feature dimensions of F_T to match those of F_S. Additionally, we align the spatial dimensions using bilinear interpolation when they differ. The total loss for the feature-level distillation includes losses on the ground-truth data $\mathcal{L}_{\mathcal{GT}}$, and between teacher and student features (i.e., F_T and F_S respectively): $\mathcal{L}_{KD,feat} = ||F_S - \phi(F_T)||_2^2$. γ_{feat} is used to scale the feature-distillation loss, while λ_{KD} controls the contribution of the distillation loss to the total loss.

$$\mathcal{L}_{total,feat} = \mathcal{L}_{GT} + \lambda_{KD}(\gamma_{feat} \times \mathcal{L}_{KD,feat}) \quad (3)$$

Combined distillation Finally, the motivation behind *combined distillation* is to guide not just the final representations but also the intermediate representations simultaneously, in hopes of achieving better results by combining the merits of output distillation and feature distillation.

$$\mathcal{L}_{total,comb} = \mathcal{L}_{GT} + \lambda_{KD}(\mathcal{L}_{KD,out} + \gamma_{feat} \times \mathcal{L}_{KD,feat}) \quad (4)$$

3.4 Datasets

The reason behind the state-of-the-art performance of HaMeR lies in its scaling of training data and the learning capacity of the network architecture. The dataset used for training HaMeR consists of approximately 2.7 million annotated samples, combined from different sources. FreiHAND (Zimmermann et al., 2019), HO3D (Hampali et al., 2020), H2O3D (Hampali et al., 2020), InterHand2.6M (Moon et al., 2020), MTC (Xiang et al., 2018), DexYCB (Chao et al., 2021) provide 3D annotations along with RHD (Zimmermann and Brox, 2017) which is a synthetic dataset, while COCO WholeBody (Jin et al., 2020), Haple-FullBody (Fang et al., 2022), MPII + NZSL (Simon et al., 2017) contribute with 2D annotations. The unification of various hand pose datasets into a consistent format ready to be used is a crucial contribution from the HaMeR team, allowing large-scale supervised training with minimal pre-processing. Accordingly, we adopt the same combined dataset and sampling distributions used in HaMeR for training all of our networks. We evaluate and provide quantitative results on the HO3D-v2 (Hampali et al., 2020) dataset, allowing us to compare our results directly with HaMeR and other baselines. The dataset serves as an ideal evaluation set as it offers challenging and realistic hand-object interaction scenarios.

4 EXPERIMENTS AND RESULTS

In this section, we present a quantitative evaluation of our proposed KD approach on the HO3D-v2 dataset (Hampali et al., 2020). We use standard evaluation metrics, including *PA-MPJPE* ($\mathbf{J}_{err}$), and *PA-MPVPE* ($\mathbf{V}_{err}$) both being in *mm*, and *F@5mm*, and *F@15mm*. The official HO3D-v2 competition website provides more details on the metrics used for evaluation[1]. We also comment on the performance of the model with respect to its number of parameters and FPS, as shown in Table 1 and Figure 3. All experiments and FPS calculations are evaluated on a consumer-grade *RTX-4060 Ti* GPU.

4.1 Ablation Study

We start by establishing performance baselines using lightweight backbones without any distillation. Subsequently, we compare the different distillation methods to assess their impact on performance.

4.1.1 Baseline Performance without Distillation

Prior to applying any distillation, we establish baselines by replacing HaMeR's original ViT-H backbone (Dosovitskiy et al., 2021; Pavlakos et al., 2024) with a range of lightweight models (Howard et al., 2017; Mehta and Rastegari, 2021; He et al., 2015; Liu et al., 2022). Each model is initialized with ImageNet-pretrained weights and trained only using ground-truth supervision, without the teacher model's guidance.

The baseline experiments, shown in Table 2, highlight a clear trade-off between model complexity and performance. The original HaMeR (671M parameters) produces the best results across all metrics, but runs at only 27 FPS. In contrast, lighter models like MobileViT-S (Mehta and Rastegari, 2021) reduce parameter count by *6x* and improve the inference speed by *1.55x*, but with a *1.6mm* drop in accuracy. ResNet-50 (He et al., 2015) offers promising results with only 10% of the parameter count, and *1.85x* speed-up, at just a *1mm* difference in accuracy. ConvNeXt-L (Liu et al., 2022), arguably, offers the best trade-off between compression and accuracy, with a *1.48x* increase in FPS and about *64%* reduction in complexity, performs only *0.6mm* worse than HaMeR. These results demonstrate promising trade-offs and indicate that data scaling may be more beneficial than increasing model complexity alone.

[1] https://codalab.lisn.upsaclay.fr/competitions/4318#learn_the_details

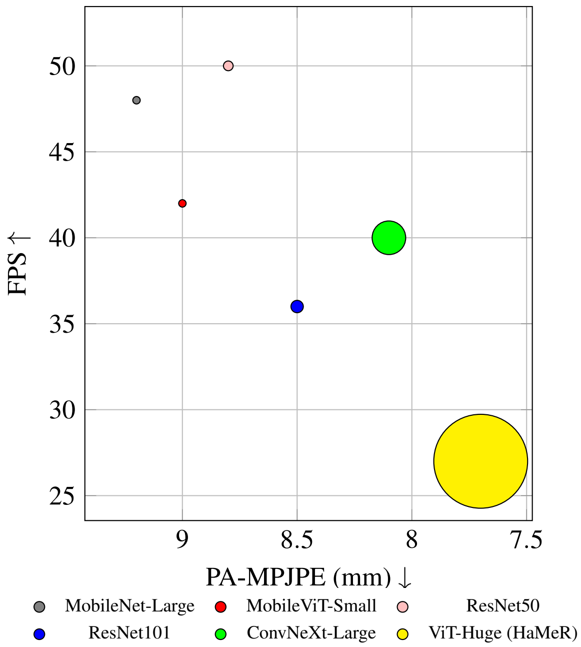

Figure 3: Trade-off between model accuracy (PA-MPJPE), speed (FPS), and parameter size (circle size). The Figure shows that the most accurate model is HaMeR; however, it shows that other alternatives give close performance with fewer resources and better runtime. ConvNeXt-L with feature distillation falls right behind HaMeR in our experiments with 1.48x FPS boost and 64% reduction in size.

4.1.2 Output-Level Knowledge Distillation

Next, we evaluate output-level distillation, where the objective is to allow the student network to learn from the predictions of the teacher network along with the ground-truth. The formulation of all the relevant loss terms is detailed in Section 3.3. All the results in this study are summarized in Table 3. Applying teacher supervision on the outputs of the student model produces interesting results across different λ_{KD} (KD weight) values. For MobileNet-L (Howard et al., 2017) configurations, we observe no improvements in any of the metrics; instead, we observe a degradation of up to *0.3mm* for PA-MPVPE, along with similar degradation for all other metrics. On the other hand, MobileViT-S (Mehta and Rastegari, 2021) configurations offer noticeable improvements at $\lambda_{KD} = 0.5$, with *0.4mm* improvement in PA-MPVPE. This gain could indicate the compatibility between the similar architecture shared by both the ViT-Huge (Dosovitskiy et al., 2021) teacher backbone and the MobileVit-S student backbone (Mehta and Rastegari, 2021). The improvement, however, starts to degrade at $\lambda_{KD} = 0.8$. ResNet-50 (He et al., 2015) does not yield noticeable improvements with output-level distillation; the baseline without any distillation remains the best-performing configuration. ResNet-101 (He et al., 2015) with a deeper CNN architecture suggests some improvements at $\lambda_{KD} = 0.8$ with approximately *0.3mm* improvement in PA-MPVPE. The configuration with ConvNeXt-L (Liu et al., 2022) and $\lambda_{KD} = 0.5$ shows a small improvement by *0.1mm* in PA-MPVPE.

Table 1: The number of networks' parameters in millions, along with the runtime of different backbone configurations in Frames-per-Second (FPS). All configurations were tested on a consumer-grade RTX 4060 Ti GPU. We also report the number of floating point operations (GFLOPs) as stated in the torchvision library.

Backbone	**Total Params (M)**	**FPS ↑**	**GFLOPs**
MobileNet-L	42.0	48	0.2
MobileViT-S	42.0	42	0.5
ResNet-50	69.3	50	4.1
ResNet-101	88.3	36	7.8
ConvNeXt-L	240	40	34.4
ViT-H	671	27	167.3

Table 2: Performance evaluation after training different backbones without KD to set baselines. The experiments are sorted in ascending order of the total parameter size of the network.

Backbone	**J_{err} ↓**	**V_{err} ↓**	**F@5.0 / F@15.0 ↑**
MobileNet-L	9.2	9.3	0.532 / 0.967
MobileViT-S	9.5	9.5	0.515 / 0.966
ResNet-50	8.8	8.9	0.563 / 0.971
ResNet-101	8.8	9.0	0.558 / 0.971
ConvNext-L	8.3	8.5	0.599 / 0.976
ViT-H (HaMeR)	7.7	7.9	0.635 / 0.980

Table 3: **Output-level Distillation:** Evaluating different backbones trained with KD loss applied on the outputs of the student and teacher models. λ_{KD} refers to the weight of the KD loss.

Backbone	**λ_{KD}**	**J_{err} ↓**	**V_{err} ↓**	**F@5.0 / F@15.0 ↑**
MobileNet-L	0.3	9.5	9.6	0.525 / 0.964
MobileNet-L	0.5	9.3	9.4	0.528 / 0.964
MobileNet-L	0.8	9.3	9.4	0.525 / 0.964
MobileViT-S	0.3	9.1	9.2	0.536 / 0.968
MobileViT-S	0.5	9.0	9.1	0.543 / 0.970
MobileViT-S	0.8	9.5	9.6	0.516 / 0.965
ResNet-50	0.3	9.0	9.2	0.547 / 0.969
ResNet-50	0.5	8.9	9.0	0.562 / 0.968
ResNet-50	0.8	9.0	9.1	0.556 / 0.970
ResNet-101	0.3	8.9	9.1	0.552 / 0.970
ResNet-101	0.5	8.8	8.9	0.569 / 0.970
ResNet-101	0.8	8.5	8.7	0.583 / 0.974
ConvNext-L	0.3	8.3	8.5	0.588 / 0.976
ConvNext-L	0.5	8.2	8.4	0.596 / 0.978
ConvNext-L	0.8	8.4	8.7	0.582 / 0.974

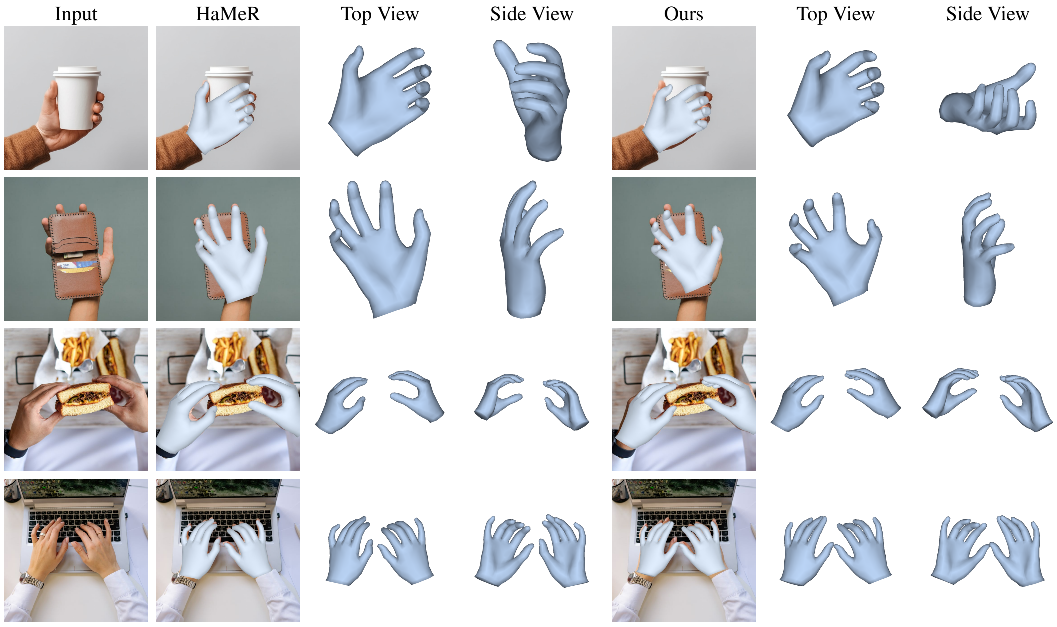

Figure 4: Qualitative results on images from the internet, the scenes represent hands interacting with the environment. We compare our results of the best configuration (ConvNext-L with feature-level distillation) with HaMeR both in 2D and 3D.

4.1.3 Feature-Level Knowledge Distillation

In this section, we further explore the performance of the models with distillation applied on the feature-map level. More specifically, in these configurations, the backbone in the student network attempts to learn a similar feature representation as the larger teacher model. The evaluation results are presented in Table 4.

We observe that MobileNet-L (Howard et al., 2017), MobileViT-S (Mehta and Rastegari, 2021), ResNet-50, and ResNet-100 (He et al., 2015) all show no improvement with feature-level distillation, with some even performing worse than output-level distillation. Interestingly, ConvNeXt-L (Liu et al., 2022) performs best under feature-level distillation, with $\lambda_{KD} = 0.8$ and $\gamma_{FD} = 12$, it was able to achieve the best results across all the metrics, showing a *0.2mm* improvement in PA-MPVPE compared to its baseline. The results for ConvNeXt suggest that feature-level distillation might be more suitable for relatively high-capacity student networks. A network like ConvNeXt with deep and modern architecture and rich hierarchical features might be better suited to learn the spatial and semantic cues in the teacher's feature maps.

4.1.4 Combined Distillation: Feature + Output

Finally, we evaluate the performance of the student networks trained with a combined distillation loss, where knowledge is transferred at both feature maps and output levels. This setting aims to leverage the information from both loss types and mimic the teacher more holistically. The quantitative results for this strategy are reported in Table 5.

We observe that for most backbones, combined distillation achieves performance between that of individual output-level and feature-level strategies. The combination of both distillation methods (i.e., output-level and feature-level) generally does not surpass the individual methods. For instance, ConvNeXt-L configurations do not show the same improvements gained by feature-level distillation. We can also observe this for ResNet-101, where combined distillation offers only a marginal increase in performance. The results from these experiments show that while it is possible to obtain some improvements over non-distilled models, specifically for relatively larger models, a combined distillation approach may dilute the stronger signal of either individual method.

4.2 Comparison with the State-of-the-art

In Table 6, we compare our best-performing model with recent state-of-the-art methods for 3D hand reconstruction. Using a ConvNeXt-L (Liu et al., 2022) backbone trained using feature-level distillation ($\lambda_{KD} = 0.8$, $\gamma_{FD} = 12$), our approach outperforms strong baselines including HandOccNet (Park et al., 2022), METRO (Lin et al., 2020), ArtiBoost

Table 4: **Feature-level Distillation:** Evaluating different backbones trained with KD loss applied on the feature maps produced by the backbones of student and teacher models. λ_{KD} refers to the weight of the total KD loss, while γ_{FD} refers to a scalar used to scale only the feature-distillation loss.

Backbone	λ_{KD}	γ_{FD}	$\mathbf{J}_{err} \downarrow$	$\mathbf{V}_{err} \downarrow$	**F@5.0 / F@15.0** $\uparrow$
MobileNet-L	0.3	6	9.6	9.7	0.514 / 0.961
MobileNet-L	0.5	6	10.4	10.5	0.460 / 0.952
MobileNet-L	0.8	12	10.0	10.2	0.477 / 0.958
MobileViT-S	0.3	6	9.5	9.4	0.525 / 0.965
MobileViT-S	0.5	6	9.8	9.9	0.491 / 0.961
MobileViT-S	0.8	12	9.7	9.7	0.509 / 0.959
ResNet-50	0.3	6	9.1	9.2	0.541 / 0.969
ResNet-50	0.5	6	9.2	9.3	0.531 / 0.968
ResNet-50	0.8	12	9.0	9.0	0.546 / 0.971
ResNet-101	0.3	6	9.0	9.1	0.547 / 0.970
ResNet-101	0.5	6	8.9	9.0	0.546 / 0.970
ResNet-101	0.8	12	9.0	9.0	0.546 / 0.971
ConvNext-L	0.3	6	8.5	8.6	0.579 / 0.978
ConvNext-L	0.5	6	8.2	8.4	0.591 / 0.977
ConvNext-L	0.8	12	8.1	8.3	0.599 / 0.979

Table 5: **Combined Distillation:** Evaluating different backbones trained with KD loss applied on the outputs and the feature maps produced by the backbones of student and teacher models. λ_{KD} refers to the weight of the total KD loss, while γ_{FD} refers to a scalar used to scale only the feature-distillation loss.

Backbone	λ_{KD}	γ_{FD}	$\mathbf{J}_{err} \downarrow$	$\mathbf{V}_{err} \downarrow$	**F@5.0 / F@15.0** $\uparrow$
MobileNet-L	0.3	6	9.6	9.7	0.509 / 0.960
MobileNet-L	0.5	6	9.8	9.9	0.499 / 0.961
MobileNet-L	0.8	12	10.3	10.4	0.472 / 0.951
MobileViT-S	0.3	6	9.8	9.8	0.498 / 0.960
MobileViT-S	0.5	6	9.7	9.8	0.498 / 0.959
MobileViT-S	0.8	12	9.9	10.0	0.489 / 0.958
ResNet-50	0.3	6	9.1	9.2	0.540 / 0.969
ResNet-50	0.5	6	9.2	9.2	0.538 / 0.969
ResNet-50	0.8	12	9.2	9.3	0.533 / 0.968
ResNet-101	0.3	6	9.0	9.1	0.551 / 0.970
ResNet-101	0.5	6	8.9	9.1	0.553 / 0.970
ResNet-101	0.8	12	9.0	9.2	0.550 / 0.969
ConvNext-L	0.3	6	8.6	8.8	0.573 / 0.975
ConvNext-L	0.5	6	8.5	8.7	0.581 / 0.976
ConvNext-L	0.8	12	8.4	8.5	0.587 / 0.978

(Li et al., 2021) and I2L-MeshNet (Moon and Lee, 2020). Our method achieves comparable performance to the recent methods, while also improving upon the inference time as discussed in Section 4.1.1. In Figure 4, we present qualitative results for our best-performing network configuration (i.e., ConvNeXt-L with feature-level Distillation) on challenging images collected from the internet. We also present results from HaMeR (teacher network) to demonstrate the difference in reconstruction accuracy. Although HaMeR's qualitative results align more closely with the hands in the images, KD-trained ConvNeXt achieves similarly strong performance with only minor inaccuracies.

Table 6: Comparison with the state-of-the-art using our best network configuration. The selected model contains a ConvNeXt-L (Liu et al., 2022) as the backbone and was trained using feature-level distillation with $\lambda_{KD} = 0.8, \gamma_{FD} = 12$.

Method	$\mathbf{J}_{err} \downarrow$	$\mathbf{V}_{err} \downarrow$	**F@5.0 / @15.0** $\uparrow$
Pose2Mesh (Choi et al., 2020)	12.5	12.7	0.441 / 0.909
THOR-Net (Aboukhadra et al., 2023)	11.3	10.7	- / -
I2L-MeshNet (Moon and Lee, 2020)	11.2	13.9	0.409 / 0.932
ArtiBoost (Li et al., 2021)	11.1	10.9	0.488 / 0.944
METRO (Lin et al., 2020)	10.4	11.1	0.484 / 0.946
I2UV-HandNet (Chen et al., 2021)	9.9	9.5	0.956 / 0.803
HandOccNet (Park et al., 2022)	9.1	8.8	0.564 / 0.968
HaMeR (Pavlakos et al., 2024)	7.7	7.9	0.635 / 0.980
Hamba (Dong et al., 2024)	7.5	7.7	0.648 / 0.982
WiLoR (Potamias et al., 2025)	7.5	7.7	0.646 / 0.983
Ours	8.1	8.3	0.599 / 0.979

5 CONCLUSION

In this work, we systematically investigated the use of smaller and lighter networks combined with Knowledge Distillation strategies—specifically output-level, feature-level, and their combination—to accelerate 3D hand mesh reconstruction. Our experiments revealed key insights into the relationship between model size, architectural complexity, and the effectiveness of teacher-student supervision. We replaced HaMeR's original ViT-H backbone with more efficient alternatives such as MobileViT and ConvNeXt, and evaluated the impact of different distillation strategies. Our results show that output-level distillation consistently yields strong performance across student models, while feature-level distillation provides benefits primarily for higher-capacity networks. Overall, this work demonstrates the potential of KD to enable fast and accurate hand reconstruction suitable for real-time applications on resource-constrained devices. Our approach offers a practical alternative to large transformer-based models without a significant sacrifice in accuracy.

Future research can explore and investigate more expressive ways of enhancing feature-level distillation, such as using attention-based distillation, spatial or channel-wise distillation, or relational knowledge distillation. These approaches may better help the student networks to learn the complex feature representation of the teacher network.

Acknowledgments This work was partially funded by the European Union under Horizon Europe under the project *SHARESPACE* (101092889) and by the Federal Ministry of Education and Research Germany under the project *COPPER* (16IW24009).

REFERENCES

Aboukhadra, A. T., Malik, J., Elhayek, A., Robertini, N., and Stricker, D. (2023). Thor-net: End-to-end graformer-based realistic two hands and object reconstruction with self-supervision. In *WACV*, pages 1001–1010.

Aboukhadra, A. T., Malik, J., Robertini, N., Elhayek, A., and Stricker, D. (2024). Shapegraformer: Graformer-based network for hand-object reconstruction from a single depth map. *IEEE Access*, 12:124021–124031.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2023). *SCAPE: Shape Completion and Animation of People*. Association for Computing Machinery, New York, NY, USA, 1 edition.

Capistrano, J., Orillaza, N., and Naval, P. (2025). *Lightweight 3D Hand Pose Estimation Using Knowledge Distillation with Hand Kinematics*, pages 331–344.

Chao, Y.-W., Yang, W., Xiang, Y., Molchanov, P., Handa, A., Tremblay, J., Narang, Y. S., Van Wyk, K., Iqbal, U., Birchfield, S., Kautz, J., and Fox, D. (2021). Dexycb: A benchmark for capturing hand grasping of objects. In *CVPR*, pages 9044–9053.

Chatzis, T., Stergioulas, A., Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2020). A comprehensive study on deep learning-based 3d hand pose estimation methods. *Applied Sciences*, 10(19):6850.

Chen, P., Chen, Y., Yang, D., Wu, F., Li, Q., Xia, Q., and Tan, Y. (2021). I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling. In *CVPR*, pages 12929–12938.

Choi, H., Moon, G., and Lee, K. M. (2020). Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, pages 769–787. Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee.

Dong, H., Chharia, A., Gou, W., Vicente Carrasco, F., and De la Torre, F. D. (2024). Hamba: Single-view 3d hand reconstruction with graph-guided bi-scanning mamba. *NeurIPS*, 37:2127–2160.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Fang, H.-S., Li, J., Tang, H., Xu, C., Zhu, H., Xiu, Y., Li, Y.-L., and Lu, C. (2022). Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time.

Ge, L., Cai, Y., Weng, J., and Yuan, J. (2018). Hand pointnet: 3d hand pose estimation using point sets. In *CVPR*, pages 8417–8426.

Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., and Yuan, J. (2019). 3d hand shape and pose estimation from a single rgb image.

Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *ArXiv*, abs/2312.00752.

Gu, Y., Dong, L., Wei, F., and Huang, M. (2024). MiniLLM: Knowledge distillation of large language models. In *ICLR*.

Hampali, S., Rad, M., Oberweger, M., and Lepetit, V. (2020). Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*.

Hasson, Y., Varol, G., Schmid, C., and Laptev, I. (2021). Towards unconstrained joint hand-object reconstruction from rgb videos. In *3DV*, pages 659–668. IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CVPR*, pages 770–778.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *ArXiv*, abs/1704.04861.

Hsieh, C.-Y., Li, C.-L., YEH, C.-K., Nakhost, H., Fujii, Y., Ratner, A. J., Krishna, R., Lee, C.-Y., and Pfister, T. (2023). Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Hwang, D.-H., Kim, S., Monet, N., Koike, H., and Bae, S. (2020). Lightweight 3d human pose estimation network training using teacher-student learning. In *WACV*, pages 479–488.

Jin, S., Xu, L., Xu, J., Wang, C., Liu, W., Qian, C., Ouyang, W., and Luo, P. (2020). Whole-body human pose estimation in the wild. In *ECCV*, pages 196–214. Springer.

Ko, J., Kim, S., Chen, T., and Yun, S.-Y. (2024). Distillm: Towards streamlined distillation for large language models. In *ICML*, pages 24872–24895. PMLR.

Li, K., Yang, L., Zhan, X., Lv, J., Xu, W., Li, J., and Lu, C. (2021). Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis. *CVPR*, pages 2740–2750.

Li, M., Zhang, H., Zhang, Y., Shao, R., Yu, T., and Liu, Y. (2024). Hhmr: Holistic hand mesh recovery by enhancing the multimodal controllability of graph diffusion models. In *CVPR*, pages 645–654.

Li, Y., Zhang, L., Qiu, Z., Jiang, Y., Li, N., Ma, Y., Zhang, Y., Xu, L., and Yu, J. (2022). Nimble: a non-rigid hand model with bones and muscles. *ACM Transactions on Graphics (TOG)*, 41(4):1–16.

Lin, K., Wang, L., and Liu, Z. (2020). End-to-end human pose and mesh reconstruction with transformers. *CVPR*, pages 1954–1963.

Lin, K., Wang, L., and Liu, Z. (2021). Mesh graphormer. In *ICCV*, pages 12939–12948.

Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). A convnet for the 2020s. *CVPR*, pages 11966–11976.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6).

Makris, A. and Argyros, A. (2015). Model-based 3d hand tracking with on-line shape adaptation. In *BMVC*, pages 77.1–77.12. BMVA Press.

Malik, J., Abdelaziz, I., Elhayek, A., Shimada, S., Ali, S. A., Golyanik, V., Theobalt, C., and Stricker, D. (2020). Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In *CVPR*.

Mehta, S. and Rastegari, M. (2021). Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. *ArXiv*, abs/2110.02178.

Moon, G. and Lee, K. M. (2020). I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. *ArXiv*, abs/2008.03713.

Moon, G., Yu, S.-I., Wen, H., Shiratori, T., and Lee, K. M. (2020). Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *ECCV*.

Park, J., Oh, Y., Moon, G., Choi, H., and Lee, K. M. (2022). Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *CVPR*, pages 1496–1505.

Pavlakos, G., Shan, D., Radosavovic, I., Kanazawa, A., Fouhey, D., and Malik, J. (2024). Reconstructing hands in 3d with transformers. In *CVPR*, pages 9826–9836.

Potamias, R. A., Zhang, J., Deng, J., and Zafeiriou, S. (2025). Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *CVPR*, pages 12242–12254.

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 36(6):1–17.

Rong, Y., Shiratori, T., and Joo, H. (2021). Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration.

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, pages 1145–1153.

Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D., Oulasvirta, A., and Theobalt, C. (2016). Real-time joint tracking of a hand manipulating an object from rgb-d input. *ArXiv*, abs/1610.04889.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wu, X., Finnegan, D., O'Neill, E., and Yang, Y.-L. (2018). Handmap: Robust hand pose estimation via intermediate dense guidance map supervision. In *ECCV*.

Xiang, D., Joo, H., and Sheikh, Y. (2018). Monocular total capture: Posing face, body, and hands in the wild. *CVPR*, pages 10957–10966.

Zhang, F., Zhu, X., and Ye, M. (2019a). Fast human pose estimation. In *CVPR*, pages 3517–3526.

Zhang, X., Li, Q., Mo, H., Zhang, W., and Zheng, W. (2019b). End-to-end hand mesh recovery from a monocular rgb image.

Zhang, Y., Chen, L., Liu, Y., Zheng, W., and Yong, J. (2020). Explicit knowledge distillation for 3d hand pose estimation from monocular rgb.

Zimmermann, C. and Brox, T. (2017). Learning to estimate 3d hand pose from single rgb images. In *ICCV*.

Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., and Brox, T. (2019). Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*.