Handformer2T: A Lightweight Regression-based Model for Interacting Hands Pose Estimation from A Single RGB Image

Pengfei Zhang
University of California, Irvine
Irvine, CA 92617, USA
pengfz5@uci.edu

Deying Kong
Google
USA
deyingk@google.com

Abstract

Despite its extensive range of potential applications in virtual reality and augmented reality, 3D interacting hand pose estimation from RGB image remains a very challenging problem, due to appearance confusions between keypoints of the two hands, and severe hand-hand occlusion. Due to their ability to capture long range relationships between keypoints, transformer-based methods have gained popularity in the research community. However, the existing methods usually deploy tokens at keypoint level, which inevitably results in high computational and memory complexity. In this paper, we propose a simple yet novel mechanism, i.e., hand-level tokenization, in our transformer based model, where we deploy only one token for each hand. With this novel design, we also propose a pose query enhancer module, which can refine the pose prediction iteratively, by focusing on features guided by previous coarse pose predictions. As a result, our proposed model, Handformer2T, can achieve high performance while maintaining lightweight. Extensive experiments on public benchmarks demonstrate that our model can achieve state-of-the-art performance on interacting-hand pose estimation with higher throughput, less memory and faster speed.

1. Introduction

3D hand pose estimation has significant applications in various fields, including human-computer interaction, virtual reality, and robotics [8, 9, 19, 27, 38–40]. It facilitates more natural interactions between humans and technology [10, 34]. However, accurately estimating hand poses from monocular RGB images remains challenging due to appearance confusion between the two hands and their keypoints, along with frequent occurrences of hand-hand occlusion and self-occlusion.

Heatmap-based methods have been one of the main streams of solving the 3D hand pose estimation problem.

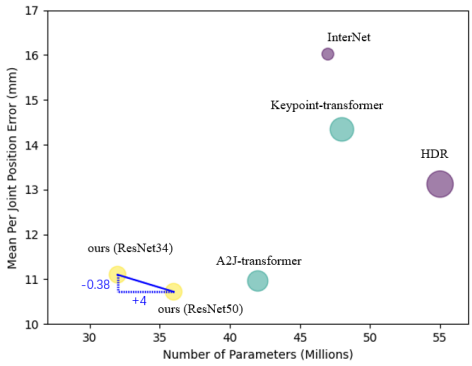

Figure 1. Performance comparison between our method and existing methods. The x-axis represents the model size while the y-axis depicts mean per joint position error in millimeter. Heatmap-based methods are draw in purple circles and regression-based methods are in blue circles. The radius of the circle indicates the inference speed, the smaller the faster. Two variants of our model is shown in yellow, with different backbones.

These methods predict likelihood heatmaps for each keypoints and then obtain the final 3D keypoints positions via argmax or soft argmax operations. However, due to the high 3D dimension, the heatmap-based methods suffer from computational complexity.

Other researchers have proposed to predict 3D keypoints coordinates directly, instead of relying on intermediate heatmaps. Previously, while having higher efficiency, regression-based methods often underperform the heatmap-based methods. With the emergence of Transformers in the computer vision [1, 23, 48], regression-based methods have prospered and started to achieve state-of-the-art performance [9, 25]. Existing transformer-based methods usually assign one dedicated token for each keypoint. Since the attention mechanism has a quadratic complexity in terms of the number of tokens, existing methods would face substantial computational challenges, especially in the scenario of interacting hands, where the number of the keypoints doubles compared to single hand scenario.

To solve this issue, for the first time, we propose the use of *hand-level* tokens instead of keypoint-level tokens. Specifically, we deploy only one token for each hand, instead of one token for each keypoint. With this novel design, we propose our lightweight transformer-based model, Handformer2T, for interacting hand pose estimation from a single RGB image. Since our model only utilizes two tokens, the computational complexity can significantly reduced. As shown in Fig. 1, compared with state-of-the-art methods, our model can achieve best performance in terms of mean per joint position, while maintaining smallest size.

In detail, our network is composed of a feature extractor, a coarse predictor and a refiner, as illustrated in Fig. 2. The feature extractor can extract both global and local features from the input image. With these rich features, the coarse predictor generates an initial prediction by employ several layers of multi-head self-attention. Then, the refiner takes as input the image features and the initial prediction, and improves the prediction iteratively. In particular, we propose a novel pose query enhancement mechanism for the refiner. The input query in the refiner is obtained from a keypoint guided feature fusion module, where the features from the backbone are sampled with the guidance of the previously predicted keypoint positions. In this way, the refiner can focus on features near the keypoints, thus improving the keypoint predictions progressively. In summary, our main contributions are listed as following

- To the best of our knowledge, our work is the *first* to propose the concept of *hand-level* tokenization. With this novel design, we have proposed a lightweight transformer based model for 3D interacting hand pose estimation using a single RGB image.
- To capture the local information around the keypoints of the two hands, we have proposed a novel pose query enhancement mechanism, which can iteratively improve the keypoint predictions.
- State-of-the-art performance of interacting hand pose estimation has been achieved on two large public hand datasets, with smaller model size and faster inference speed.

2. Related Works

Heatmap-based vs Regression-based Pose estimation. There exist numerous heatmap-based methods in the field of human/hand pose estimation [16–18, 24, 28, 40, 43], where likelihood heatmaps are employed to represent joint locations. Previous studies [12, 22, 27] employ differentiable soft-argmax [37] operations to retrieve joint locations from heatmaps in a differentiable manner, enabling end-to-end training. Furthermore, [4, 26] utilize hand or finger segmentations as additional supervision for joint likelihood heatmaps. While achieving excellent performance, these models are required to generate high-resolution features and heatmaps, significantly increasing computational costs and reducing throughput.

In contrast to heatmap-based methods that calculate likelihood heatmaps, regression-based methods compute probability distributions for joint coordinates. In the context of human or hand pose estimation, few approaches are regression-based. In human pose estimation, RLE [15, 20, 34, 42] introduces a regression-based approach using maximum likelihood estimation, which a significant advancement in regression techniques. Building upon this, Poseur [25] presents a two-stage regression model designed to refine RLE results. Additionally, anchor-based methods [31, 32] classify poses into a set of K anchor poses for human pose estimation, followed by a regression module that refines the anchor to obtain the final prediction. For hand pose estimation, there are fewer regression-based approaches. Two-stage methods like [4, 12] elevate 2D poses to 3D space through regression, relying on heatmap-based models for predicting 2D poses. A2J [9, 45] extend anchor-based techniques to the interacting hand domain.

Lightweight Pose Estimation Models. Lightweight models for hand pose estimation play a crucial role in applications involving human-computer interaction and gesture recognition, but this domain remains relatively unexplored. In this context, Santavas et al. [33] emphasize vision-based human pose estimation for Human-Computer Interaction (HCI) [29], while Wu et al. [44] concentrate on enhancing real-time hand pose estimation. However, these models do not delve into the intricacies of the interacting hand pose challenge.

Transformer in Hand Pose Estimation. The self-attention mechanism and the transformer model [41], have been applied in various fields, including pose estimation. Researchers have made significant strides [6, 9, 21, 26] in incorporating transformers into this problem. However, the self-attention mechanism involves pairwise computations between tokens, resulting in an $O(n^2)$ complexity in terms of time and memory [11], where n is the number of tokens. Therefore, existing architectures face performance limitations due to token number constraints.

In this context, our Handformer2T falls into the lightweight regression-based category, utilizing the Transformer module. Different from prior works, we only employ two tokens as input to the transformer, significantly reducing computational costs. Moreover, we don't rely on the assumptions of two hands prior, unlike previous work [21, 47] which attempt to regress two hands, even if the input image contains just a single hand. When the input image contains only a single hand, the ground truth of the other hand becomes pure noise. In such cases, if we input "both hands" into QEM, our model simplifies to Poseur

[25], which enhances coarse regression by manually generating noise and feeding it to the Transformer. The main differences are: 1) we use Keypoint-guided features and lower dimensions for tokens, and 2) the attention modules utilizes two tokens instead of $2 \times J$ tokens (J is the number of keypoints for each hand), making our model more memory-efficient.

3. Our Method

In this section, we present our model, Handformer2T, for interacting hand pose estimation from a single RGB image. The proposed Handformer2T model is a transformer-based model, which directly outputs keypoint locations instead of heatmaps. Importantly, for the first time, we propose to use *hand-level* tokens instead of keypoint-level tokens in the transformer-based architecture, resulting in only *two* tokens needed in our model. With this novel hand-level token design, our model mainly consists of three parts, namely, a feature extractor, a coarse predictor that provides an initial prediction, and an iterative refiner which benefits from a novel pose query enhancing mechanism to improve the coarse prediction.

Concisely, our whole model can be formulated as the following function

$$\phi : I \mapsto \{p^{(i)}, u^{(i)}\}_{i=1,\cdots,N}, \tag{1}$$

where $I \in \mathbb{R}^{w \times h \times 3}$ is the input RGB image, N represents the number of multiple coarse-to-fine stages in our model, $p^{(i)} \in \mathbb{R}^{2J \times 3}$ and $u^{(i)} \in \mathbb{R}^{2J \times 1}$ stand for 3D positions of the keypoints and their corresponding uncertainties at the i-th stage. Note that J denotes the number of keypoints of each hand. In the following subsections, we will discuss each component of our model in details.

3.1. Feature Extractor

The feature extractor aims to extract features that can both represent the global and local information from the input image. In our model, the feature extractor is composed by a convolutional neural network backbone and a few subsequent operators.

Given an input image $I \in \mathbb{R}^{h \times w \times 3}$, the backbone outputs multi-scale features $\{F_i \in \mathbb{R}^{h_i \times w_i \times c_i}\}_{i=1,\cdots,n}$, where $h_i \times w_i$ is the resolution and c_i is the number of the channels of the feature map extracted from the i-th layer of the backbone. Inspired by [6], we apply several additional operations on the feature maps $\{F_i\}_{i=1,\cdots,n}$ to get the final features. For each feature map F_i, we first apply an average pooling on it, obtaining

$$f_{i,1} = \text{Average\_Pool}(F_i) \in \mathbb{R}^{c_i}, \tag{2}$$

which encodes global information of the input image. To capture rich local information, we construct another feature vector by sampling and flattening the channels of F_i. Specifically, we randomly sample $\tilde{c}_i$ channels from the entire c_i channels, which would result in a sampled feature map $\tilde{F}_i \in \mathbb{R}^{h_i \times w_i \times \tilde{c}_i}$. Then we flatten the sampled feature as

$$f_{i,2} = \text{Flatten}(\tilde{F}_i) \in \mathbb{R}^{h_i \cdot w_i \cdot \tilde{c}_i}, \tag{3}$$

which contains rich local information. In order to obtain a final feature vector of length c_i^*, we randomly sample the remaining elements from the flattened feature map as

$$f_{i,3} = \text{Sample}(\text{Flatten}(F_i)) \in \mathbb{R}^{c_i^* - c_i - h_i \cdot w_i \cdot \tilde{c}_i}. \tag{4}$$

Finally, the feature obtained from the i-th layer of the backbone is given by

$$f_i = \text{Concat}([f_{i,1},\ f_{i,2},\ f_{i,3}]) \in \mathbb{R}^{c_i^*}. \tag{5}$$

The extracted features $\{f_i\}_{i=1,\cdots,n}$ would be sent into the subsequent coarse-to-fine stages of our model.

3.2. Hand-level Tokenization

Existing transformer-based methods for pose estimation usually deploy one token for each keypoint, which would result in $2 \cdot J = 42$ tokens in the scenario of interacting hands, with the assumption that each hand contains $J = 21$ keypoints. However, since the attention operation has a quadratic complexity of time and memory proportional to the number of input tokens, a large amount of tokens would render the transformer computationally demanding, leading to a decline in its efficiency.

To solve the above issue, for the first time, we propose the concept of *hand-level* tokenization. Instead of using one token for each keypoint, we only assign one token to each hand, reducing the number of tokens from $2 \times J$ to two. The proposed concept is simple yet very effective as validated by extensive experiments in Section. 4. As shown in Fig. 2, given the extracted image features $\{f_i\}_{i=1,\cdots,n}$, embeddings of the two hand tokens are obtained by

$$\begin{aligned} E_i^{\text{left}} &= \text{MLP}_i^{\text{left}}(f_i) \in \mathbb{R}^{c_e}, \\ E_i^{\text{right}} &= \text{MLP}_i^{\text{right}}(f_i) \in \mathbb{R}^{c_e}, \end{aligned} \tag{6}$$

where $\text{MLP}(\cdot)$ stands for a shallow multi-layer perceptron and c_e is the dimension of the embedding.

3.3. Coarse Predictor

Utilizing the proposed *hand-level* tokens, the coarse predictor of our model is mainly composed of multiple self-attention layers, as illustrated in Fig. 2. The goal of this module is to give a coarse prediction based on the feature f_1, as summarized in the following function

$$\psi^{(1)} : f_1 \mapsto \{p^{(1)}, u^{(1)}\}, \tag{7}$$

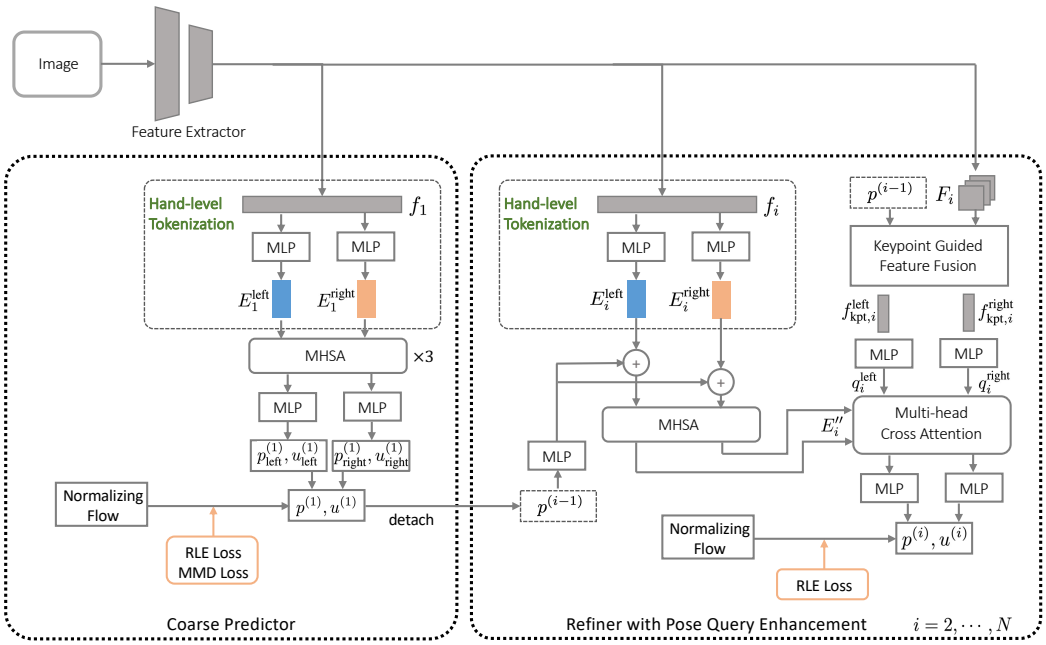

Figure 2. Overview of our proposed Handformer2T model, which mainly consists of three parts, namely, a feature extractor, a coarse predictor and a refiner. Our model utilizes a transformer-based architecture, but with a key difference from existing methods. We propose to use novel *hand-level* tokens instead of keypoint-level tokens. With rich features extracted by the feature extractor, the coarse predictor gives an initial pose prediction, which is then improved iteratively by the refiner. In both the coarse predictor and the refiner, learnable positional encodings are utilized, however they are omitted in the diagram for clearer visualization.

where $p^{(1)}$ is the coarse 3D keypoint predictions and $u^{(1)}$ is their associated uncertainties. Note that only f_1 is used in this module.

In detail, we first obtain embeddings of left-hand and right-hand tokens $\{E_1^{\text{left}}, E_1^{\text{right}}\}$ from the image feature f_1 via two shallow MLPs following Eq. (8). Then we add learnable positional encoding to the embedding as

$$\begin{aligned} E_{1,0}^{\text{left}} &= E_1^{\text{left}} + P_1^{\text{left}}, \\ E_{1,0}^{\text{right}} &= E_1^{\text{right}} + P_1^{\text{right}}. \end{aligned} \tag{8}$$

Afterwards, the embeddings would go through L layers Multi-Head Self-Attention (MHSA) layers,

$$\begin{aligned} E_{1,l} &= \text{MHSA}(E_{1,l-1}) \\ &\text{for } l = 1, \cdots, L, \end{aligned} \tag{9}$$

where embedding $E_{1,l}$ is the stacked array of $\{E_{1,l}^{\text{left}}, E_{1,l}^{\text{right}}\}$, and the MHSA is defined on an input embedding E as

$$\text{MHSA}(E) = \text{MultiHead}(E, E, E), \tag{10}$$

where MHSA is defined as

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \cdots, \text{head}_H) \cdot W^O, \\ \text{head}_h &= \text{Attention}(Q \cdot W_h^Q, K \cdot W_h^K, V \cdot W_h^V) \end{aligned} \tag{11}$$

in which W_h^Q, W_h^K, W_h^V are learnable weighting matrices for calculating query, key and value at the h-th head, respectively.

Finally, we apply two separate MLPs to the last MHSA output $E_{1,L_{\text{en}}} = \{E_{1,L_{\text{en}}}^{\text{left}}, E_{1,L_{\text{en}}}^{\text{right}}\}$ to get the coarse predictions as

$$\begin{aligned} \{p_{\text{left}}^{(1)}, u_{\text{left}}^{(1)}\} &= \text{MLP}_{\text{out}}(E_{1,L_{\text{en}}}^{\text{left}}), \\ \{p_{\text{right}}^{(1)}, u_{\text{right}}^{(1)}\} &= \text{MLP}_{\text{out}}(E_{1,L_{\text{en}}}^{\text{right}}), \\ p^{(1)} &= \text{Concat}(p_{\text{left}}^{(1)}, p_{\text{right}}^{(1)}), \\ u^{(1)} &= \text{Concat}(u_{\text{left}}^{(1)}, u_{\text{right}}^{(1)}). \end{aligned} \tag{12}$$

The coarse prediction $p^{(1)}$ and $u^{(1)}$ will be further refined by the refiner that would be discussed in the following subsection.

3.4. Refiner with Pose Query Enhancement

With the initial 3D keypoint positions from the coarse predictor and the image features from the feature extractor, the refiner module $\psi^{(i)} : F_{i-1}, p^{(i-1)} \mapsto \{p^{(i)}, u^{(i)}\}, i \in \{2, ..., N\}$ aims to iteratively improve the final pose estimation. The refiner mainly contains two key parts, the keypoint guided feature fusion module and a query enhancement module designed with a transformer-based architecture. We want to emphasize that in this transformer-based architecture, *hand-level* tokens are again utilized instead of traditional keypoint-level tokens.

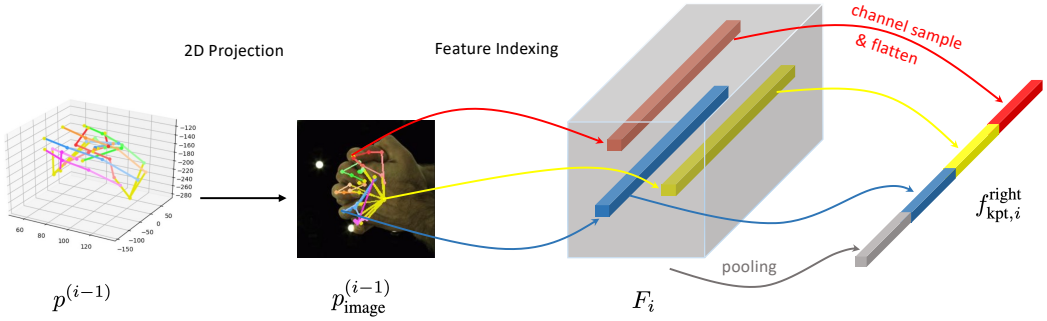

Figure 3. Illustration of keypoint-guided feature fusion. By reprojecting the 3D keypoint positions to 2D space, for each keypoint, we crop a small 3x3 patch from the feature map around the projected 2D location. After performing channel sampling for each selected patch, we flatten them and then concatenate them with the feature directly obtained by average pooling.

Keypoint guided feature fusion. To better capture the image features around the keypoints, inspired by [13, 30], we propose to use the keypoint guided feature fusion mechanism, where we sample features around the keypoints to obtain rich local information. Specifically, given an initial keypoints prediction $p^{(i-1)} \in \mathbb{R}^{2J\times 3}$ and a feature map $F_i \in \mathbb{R}^{h_i\times w_i\times c_i}$, $i \in \{2,\cdots,N\}$, we first reproject the 3D keypoints $p^{(i-1)}$ to the 2D image space,

$$p^{(i-1)}_{\text{image}} = \Pi(p^{(i-1)}) \in \mathbb{R}^{2J\times 2}, \tag{13}$$

where $p^{(i-1)}_{\text{image}}$ is the image coordinates of the hand keypoints. Meanwhile, we randomly sample $\tilde{c}_i$ channels from the feature map, obtaining a sub-sampled feature map $\hat{F}_i \in \mathbb{R}^{h_i\times w_i\times \tilde{c}_i}$. Then we crop $2J$ patches centered at $p^{(i-1)}_{\text{image}}$ from the sampled feature map $\hat{F}_i$. Through our experiments, the patch size is fixed as 3×3. As illustrated in Fig. 3, for each hand, we flatten the corresponding J patches and contatenate them together with the average pooled feature obtained in Eq. (2), resulting to the keypoint guided features $f^{\text{left}}_{\text{kpt},i} \in \mathbb{R}^{9\cdot J\cdot \tilde{c}_i + c_i}$ and $f^{\text{right}}_{\text{kpt},i} \in \mathbb{R}^{9\cdot J\cdot \tilde{c}_i + c_i}$.

Pose Query Enhancer. As shown in Fig. 2, the pose query enhancer is primarily composed with a self-attention and a cross-attention module. The input to the self-attention module is constructed by

$$E'_i = E_i + \text{MLP}(p^{(i-1)}) + P_i, \tag{14}$$

where $E_i = \{E^{\text{left}}_i, E^{\text{right}}_i\}$ is the left hand and right hand token embeddings obtained from the extracted feature f_i via Eq. (8), $p^{(i-1)}$ is keypoint predictions from the previous stage and P_i is a learnable positional encoding. By feeding E'_i to the MHSA, we obtain

$$E''_i = \text{MHSA}(E'_i). \tag{15}$$

For the cross-attention module, the queries are obtained from the keypoint guided features, $f^{\text{left}}_{\text{kpt},i}$ and $f^{\text{right}}_{\text{kpt},i}$, via two separate MLPs as following

$$\begin{aligned} q^{\text{left}}_i &= \text{MLP}(f^{\text{left}}_{\text{kpt},i}), \\ q^{\text{right}}_i &= \text{MLP}(f^{\text{right}}_{\text{kpt},i}). \end{aligned} \tag{16}$$

We denote the two queries compactly in one matrix $Q_i = \{q^{\text{left}}_i, q^{\text{right}}_i\}$. Then, the output of the cross-attention module is given by

$$E^{\text{out}}_i = \text{MultiHead}(Q_i, E''_i, E''_i). \tag{17}$$

Finally, similarly to that in the coarse predictor (Eq. (12)), two MLPs are applied to the output E^{out}_i to get the refined predictions for left and right hands, resulting in $p^{(i)}$ and $u^{(i)}$ for $i = 2,\cdots,N$.

3.5. Loss Functions

Residual Log-likelihood Estimation (RLE) Loss. To calculate RLE Loss, following [20], we calculate a probability distribution $P_{\psi,\varphi}(x|\mathcal{I})$ that reflects the probability of the ground truth appearing in the location x conditioning on the input image $\mathcal{I}$, where ψ is the parameters of regression model and φ is the parameters of the flow model. The flow model φ calculates the deviation of the output from the ground truth p^g: Firstly, φ maps a initial distribution $\bar{z} \sim N(0, I)$ to a zero-mean complex distribution $\bar{x} = \varphi(\bar{z}) \sim G_\varphi(\bar{x})$. Then, by adding a zero-mean Laplace distribution $L(\bar{x})$ to $G_\varphi(\bar{x})$, $P_\varphi(\bar{x})$ is obtained to represent the normalized density function for the underlying probability distribution $P_{\psi,\varphi}(x|\mathcal{I})$. Finally, $P_{\psi,\varphi}(x|\mathcal{I})$ is built upon $P_\varphi(\bar{x})$ by shifting and rescaling $\bar{x}$ into x: $x = u\cdot\bar{x} + p$, where p, u are predicted by regression model ψ conditioned on the input image $\mathcal{I}$. Then, the RLE loss can be calculated as

$$l_{rle} = -\log P_{\psi,\varphi}(x|\mathcal{I})|_{x=p^g} = -\log L - \log G_\varphi + \log u, \tag{18}$$

which optimizes the model parameters to make the observed ground truth p^g most probable. For more details, please refer to [20].

As our regression model consists of N parts $\psi^{(i)}, i \in \{1,...,N\}$ and each part outputs a "$p^{(i)}, u^{(i)}$" pair, we choose to supervise them with the same ground truth p^g, u^g

and different flow models $\varphi^{(i)}$. The RLE loss is given by

$$Loss_{RLE} = -\sum_{i=1}^{N} \log P_{\psi^{(i)},\varphi^{(i)}}(p^{(i)}|\mathcal{I})|_{p^{(i)}=p^g}. \quad (19)$$

Maximum Mean Discrepancy (MMD) Loss. Inspired by [3], the MMD term [5] is used to measure the distance between the distribution of predicted $p^{(1)} \sim P(p^{(1)})$ and ground truth $p^g \sim P(p^g)$ across all images, thus accelerates the convergence of coarse predictor $\psi^{(1)}$. The MDD loss term is given by

$$\begin{aligned} &Loss_{MMD} = MMD^2(k, \mathbf{P}(p^{(1)}), \mathbf{P}(p^g)) \\ &= \mathop{\mathbb{E}}_{p_i^{(1)}, p_{i'}^{(1)} \sim \mathbf{P}(p^{(1)})} [k(p_i^{(1)}, p_{i'}^{(1)})] + \mathop{\mathbb{E}}_{p_j^g, p_{j'}^g \sim \mathbf{P}(p^g)} [k(p_j^g, p_{j'}^g)] \\ &\quad - 2 \cdot \mathop{\mathbb{E}}_{p_i^{(1)} \sim \mathbf{P}(p^{(1)}), p_j^g \sim \mathbf{P}(p^g)} [k(p_i^{(1)}, p_j^g)], \end{aligned} \quad (20)$$

where k is the kernel function [3]. We choose Gaussian Kernel [36] by default.

Total Loss. Finally, the total loss is given by:

$$Loss_{total} = Loss_{RLE} + \lambda_1 Loss_{MMD}, \quad (21)$$

where hyper-parameter $\lambda_1 \geq 0$ balances the loss terms. In all our experiments, we set $\lambda_1 = 10$.

4. Experiment

We evaluate the performance of our model in terms of accuracy, inference speed, and memory consumption.

4.1. Experimental Settings

Datasets. Our evaluation is conducted on two public hand datasets.

InterHand2.6M [27] is a challenging RGB image dataset specifically designed for capturing two-hand interactions, even in cases of high occlusion. The dataset consists of 1.36 million training images and 849,000 test images. The ground-truth data provides semi-automatically annotated 3D coordinates for 42 hand keypoints (21 keypoints per hand).

RHP [49] is a synthetically generated dataset simulating various hand actions in everyday scenarios. It includes 41,000 training samples and 2,700 testing samples. Notably, RHP images incorporate background elements depicting outdoor scenes, allowing us to evaluate our model's performance in real-world environments. The dataset also offers corresponding depth images, though we exclude them from our experiments.

For the data preprocessing step, we apply the same methods as [9, 27, 47], including direct RGB image cropping and resizing to 256×256 pixels, along with the data augmentations proposed by InterNet [27].

Evaluation metrics. We employ well-established evaluation metrics. For the InterHand2.6M dataset, we utilize the Mean Per Joint Position Error (MJPJE). This metric measures the Euclidean distance (mm) between predicted and ground-truth 3D joint locations after aligning them with the root (wrist). We calculate the MJPJE on single-hand images and interacting-hand images, denoted as MJPJE-S and MJPJE-I respectively. For the RHP dataset, we utilize the End Point Error (EPE). This metric calculates the mean Euclidean distance (mm) between predicted and ground-truth 3D hand poses, considering root joint alignment for each left and right hand individually.

Furthermore, we assess the inference speed in terms of Frames Per Second (FPS) and quantify model parameters using the *thop* package[1]. The FPS is assessed by processing data with a batch size of 1. Throughput is denoted as the maximum number of images that can be processed per second. It can be calculated by: $Throughput = \frac{maximum\ batchsize}{average\ time\ per\ batch(s)}$. All model evaluations are conducted on a single NVIDIA RTX 3090ti GPU with 24GB memory.

Implementation details. Handformer2T is compatible with various convolutional backbones. We tested two lightweight backbones: ResNet34 [7] and ResNet50 [7]. The detailed model sizes and performances of the different backbones are shown in Table 5. Handformer2T is implemented by PyTorch on CUDA 11.1. Training is conducted using the Adam optimizer [14] with a learning rate of 0.001 and step decay (step=80, decrease_factor=4). A total of 270 epochs are executed for both InterHand2.6M and RHP datasets. During the inference phase, the refined mean $\hat{\mu_f}$ serves as the regressed output. Consequently, there is no need to execute the flow model during inference. This characteristic enhances computational efficiency and ease of deployment.

As shown in Figure 2, our model architecture does not include an excessive number of networks. Furthermore, due to the absence of a high-dimensional requirement for multiple MLPs, our model has significantly fewer parameters compared to other methods [4, 6, 9, 12, 22]. A detailed analysis will be presented in Section 4.2. Additionally, the dimension of Transformer tokens (query/key/value) plays a pivotal role in both model size and performance, a topic we will discuss in Section 4.3.

4.2. Quantitative Results

InterHand2.6M dataset: We compare Handformer2T against state-of-the-art methods on InterHand2.6M, as shown in Table 1. Handformer2T outperforms previous methods, the results of which were presented in previous

[1] https://github.com/Lyken17/pytorch-OpCounter

Table 1. Comparison with state-of-the-art model-based and model-free methods on InterHand2.6M.

Method	MJPJE-S (mm)	MJPJE-I (mm) ↓	FPS ↑	Model Size(M) ↓
Moon *et al*. [27]	12.16	16.02	107.08	47
Fan *et al*. [47]	11.32	15.57	-	-
Hampali *et al*. [6]	10.99	14.34	19.66	48
Zhang *et al*. [47]	-	13.48	17.02	143
Meng *et al*. [26]	8.51	13.12	15.47	55
Li *et al*. [21]	-	12.40	18.05	39
Jiang *et al*. [9]	8.10	10.96	25.65	42
Ours	8.28	10.72	66.34	36

Table 2. Throughput comparison with Moon *et al*. [27]. Although Moon *et al*. [27] achieves a high FPS, our model outperforms it in terms of throughput, number of images processed per second.

Methods	FPS↑	Max Batchsize↑	Throughput ↑
Moon *et al*. [27]	107.08	48	137
Ours-Resnet50	66.34	128	298

publications [4,6,9,21,26,27,47]. Analysis of Table 1 yields the following insights:

1) Notably, Handformer2T exhibits an improvement of 5.50 mm in interacting hand pose estimation and 3.92 mm in single hand pose estimation compared to the baseline [27]. In comparison with the state-of-the-art regression-based method [9], our method achieves an improvement of 0.24 mm in scenarios of interacting hands.

2) Handformer2T showcases substantial advancements in interacting hand pose estimation over the baseline [27], with a minor trade-off in frames per second (FPS). In contrast to the other methods, Handformer2T attains superior performance while maintaining much higher inference speeds.

3) Despite Handformer2T's lower FPS compared to InterNet [27], our larger batch size compensates, ultimately elevating Handformer2T's throughput (Table 2). This holds true especially when compared to other methods, as 128 substantially surpasses their reported batch sizes given the fixed GPU memory. Therefore, due to its higher FPS, Handformer2T attains the greatest throughput, proving values for data processing and deployment needs.

4) Handformer2T stands out as the most lightweight model across all alternatives, rendering it more suitable for mobile device deployment. This is attributed to two factors: 1) Our adoption of a regression-based approach that obviates the need for heatmap computations, and 2) a significant reduction in token count from 2J to 2, resulting in decreased computational complexity of the transformer.

RHP dataset: We present the comparison with the state-of-the-art methods on RHP in Table 3. Handformer2T outperforms previous methods [4, 6, 9, 21, 26, 27, 47] without relying on ground-truth information during inference time. This experiment shows the generalization ability of Handformer2T by demonstrating its effectiveness on in-the-wild two hand images.

Table 3. Comparison with state-of-the-art methods on RHP, in terms of end point error. GT-S and GT-H denote ground truth scale and handness, respectively.

Methods	GT-S	GT-H	End point error (mm) ↓
Zimm. *et al*. [49]	✓	✓	30.42
Chen *et al*. [2]	✓	✓	24.20
Moon *et al*. [27]	✗	✗	20.89
Yang *et al*. [46]	✓	✓	19.95
Spurr *et al*. [35]	✓	✓	19.73
A2J-Transformer [9]	✗	✗	17.75
Ours	✗	✗	17.20

4.3. Ablation study

4.3.1 Component Effectiveness Analysis

We explore the effectiveness of 1) MMD Loss, 2) Pose Query Enhancement and 3) Keypoint-guided feature fusion by conductiog experiments on Interhand2.6M. The specific implementation details are respectively set as: 1) remove MMD Loss; 2) remove the whole Pose Query Enhancement module and take coarse prediction $p^{(1)}$ as final prediction; 3) use original feature extractor as stated in section 3.1. As Keypoint-guided feature fusion can only be implemented inside the Pose Query Enhancement, we need to remove the Keypoint-guided feature fusion if we remove the whole Pose Query Enhancement module for the second setting. The other parameters can be fixed based on these configs to try the best performance under these conditions. We show the results in Table 4.

We can see that all components are effective to Handformer2T. Removing any one of them will result in performance decrease. Firstly, the RLE loss is effective in improving interacting hand pose estimation. Secondly, add MMD Loss will increase the performance. Thirdly, if we remove the Keypoint-guided feature fusion, the Pose Query Enhancement won't work and might affect the function of original coarse joint regressor.

4.3.2 Model Size and performance Tradeoff

We continue to explore the influence of dimension on our model size and performance, and try to find a balance between the memory usage and performance. As Handformer2T mainly consists of MLPs and Transformer modules, we focus on the dimension of Transformer tokens in Table 5. As shown from the table, higher dimension generally increase the performance. For the backbone Resnet50, the performance saturates at dimension 256.

We also investigate the effectiveness of the proposed *hand-level* token compared with traditional keypoint level token. As shown in Table 6, utilizing *hand-level* token can

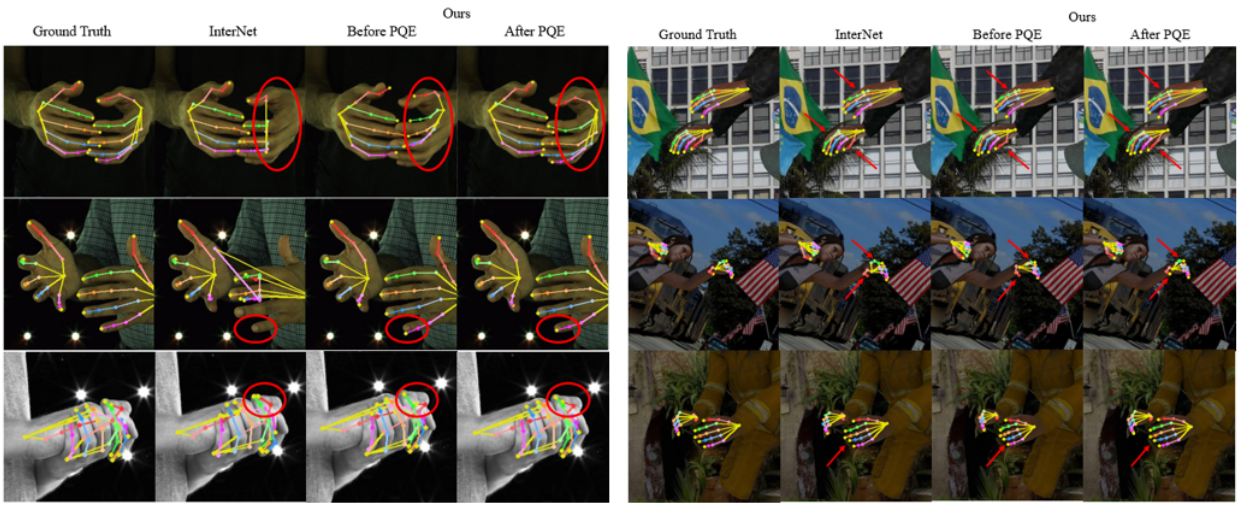

Figure 4. The qualitative results of Handformer2T. Left: results on InterHand2.6M; right: results on RHP. PQE: Pose Query Enhancement. We display three images from InterHand2.6M and three images from RHP, representing different hand pose and interaction paradigm. The first column displays the ground truth p^g, while the second column presents results obtained using InterNet [27]. The third and fourth columns showcase the outcomes produced by our module, depicting results prior to PQE (coarse prediction $p^{(1)}$) and after PQE (refined prediction $p^{(N)}$), respectively.

significantly reduce the model size by 3x, and interestingly achieve better performance.

Table 4. Component effectiveness analysis of Handformer2T. MMD Loss: Maximum Mean Discrepancy Loss. PQE: Pose Query Enhancement. KGFF: Keypoint-guided feature fusion as described in section 3.4.

MMD Loss	PQE	KGFF	MJPJE (mm) ↓
✗	✗	✗	12.49
✓	✗	✗	12.32
✓	✓	✗	16.23
✗	✓	✓	11.20
✓	✓	✓	10.72

Table 5. Ablation study on backbone and dimension of the transformer.

Backbone	Transformer Dimension	Model Size (M) ↓	MJPJE (mm) ↓
Resnet34	128	24.9	11.47
Resnet34	256	31.7	11.10
Resnet34	512	46.9	11.02
Resnet50	128	29.1	11.14
Resnet50	256	35.9	10.72
Resnet50	512	51.1	10.76

Table 6. Ablation study on the proposed *hand-level* tokenization. With a fixed embedding dimension of 256, we conduct experiments where hand-level or keypoint-level tokens are utilized on a small subset of InterHand2.6M dataset.

Token Level	Backbone	Model Size (M) ↓	MJPJE (mm) ↓
Hand Level	Resnet34	31.7	15.19
Hand Level	Resnet50	35.9	14.28
Joint Level	Resnet34	95.1	16.77
Joint Level	Resnet50	99.4	16.05

4.4. Qualititative Results

We show the qualitative results in Fig 4. We can see that Handformer2T can obtain accurate pose estimation than InterNet [27] for certain images, largely due to the refinement of PQE. However, certain limitations persist, notably our reliance on the ground-truth bounding box, which impedes the direct applicability of our model to real-world data.

4.5. Limitations

Since the model only works on single image, if extremely severe occlusion occurs, the model might fail. A possible solution is to extend the current method to a multi-view setting. Additionally, prior knowledge of camera matrix is required when performing the 3D to 2D projection in the keypoint guided feature fusion module. A future work might investigate the possibility to learn a projection matrix.

5. Conclusions

In this paper, we have proposed a novel lightweight transformer-based model, Handformer2T, for the task of interacting hand pose estimation from RGB images. By utilizing a novel *hand-level* tokenization mechanism, our model can achieve the state-of-the-art performance, while keeping the model size small and reaching fast inference speed. Extensive experiments have been conducted on two large public dataset to validate the efficacy of our proposed model.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-

end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 1

[2] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Hui Tang, Yufan Xue, Xiaohui Xie, Yen-Yu Lin, and Wei Fan. Generating realistic training images based on tonality-alignment generative adversarial networks for hand pose estimation. *arXiv preprint arXiv:1811.09916*, 2018. 7

[3] Rasool Fakoor, Pratik Chaudhari, Jonas Mueller, and Alexander J Smola. Trade: Transformers for density estimation. *arXiv preprint arXiv:2004.02441*, 2020. 6

[4] Zicong Fan, Adrian Spurr, Muhammed Kocabas, Siyu Tang, Michael J Black, and Otmar Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation. In *2021 International Conference on 3D Vision (3DV)*, pages 1–10. IEEE, 2021. 2, 6, 7

[5] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 6

[6] Shreyas Hampali, Sayan Deb Sarkar, Mahdi Rad, and Vincent Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11090–11100, 2022. 2, 3, 6, 7

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[8] Adnan Hussain, Sareer Ul Amin, Muhammad Fayaz, and Sanghyun Seo. An efficient and robust hand gesture recognition system of sign language employing finetuned inception-v3 and efficientnet-b0 network. *Computer Systems Science & Engineering*, 46(3), 2023. 1

[9] Changlong Jiang, Yang Xiao, Cunlin Wu, Mingyang Zhang, Jinghong Zheng, Zhiguo Cao, and Joey Tianyi Zhou. A2j-transformer: Anchor-to-joint transformer network for 3d interacting hand pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8846–8855, 2023. 1, 2, 6, 7

[10] Korrawe Karunratanakul, Adrian Spurr, Zicong Fan, Otmar Hilliges, and Siyu Tang. A skeleton-driven neural occupancy representation for articulated hands. In *2021 International Conference on 3D Vision (3DV)*, pages 11–21. IEEE, 2021. 1

[11] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International Conference on Algorithmic Learning Theory*, pages 597–619. PMLR, 2023. 2

[12] Dong Uk Kim, Kwang In Kim, and Seungryul Baek. End-to-end detection and pose estimation of two interacting hands. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11189–11198, 2021. 2, 6

[13] Jeonghwan Kim, Mi-Gyeong Gwon, Hyunwoo Park, Hyukmin Kwon, Gi-Mun Um, and Wonjun Kim. Sampling is matter: Point-guided 3d human mesh reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12880–12889, 2023. 5

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[15] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11605–11614, 2021. 2

[16] Deying Kong, Yifei Chen, Haoyu Ma, Xiangyi Yan, and Xiaohui Xie. Adaptive graphical model network for 2d handpose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 2

[17] Deying Kong, Haoyu Ma, Yifei Chen, and Xiaohui Xie. Rotation-invariant mixed graphical model network for 2d hand pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1546–1555, 2020. 2

[18] Deying Kong, Haoyu Ma, and Xiaohui Xie. Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2020. 2

[19] Deying Kong, Linguang Zhang, Liangjian Chen, Haoyu Ma, Xiangyi Yan, Shanlin Sun, Xingwei Liu, Kun Han, and Xiaohui Xie. Identity-aware hand mesh estimation and personalization from rgb images. In *European Conference on Computer Vision*, pages 536–553. Springer, 2022. 1

[20] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11025–11034, 2021. 2, 5

[21] Mengcheng Li, Liang An, Hongwen Zhang, Lianpeng Wu, Feng Chen, Tao Yu, and Yebin Liu. Interacting attention graph for single image two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2761–2770, 2022. 2, 7

[22] Fanqing Lin, Connor Wilhelm, and Tony Martinez. Two-hand global 3d pose estimation using monocular rgb. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2373–2381, 2021. 2, 6

[23] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1954–1963, 2021. 1

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[25] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 72–88. Springer, 2022. 1, 2, 3

[26] Hao Meng, Sheng Jin, Wentao Liu, Chen Qian, Mengxiang Lin, Wanli Ouyang, and Ping Luo. 3d interacting hand pose estimation by hand de-occlusion and removal. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 380–397. Springer, 2022. 2, 7

[27] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 548–564. Springer, 2020. 1, 2, 6, 7, 8

[28] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 483–499. Springer, 2016. 2

[29] Wenrao Pang, Qing Gao, Yinan Zhao, Zhaojie Ju, and Junjie Hu. Basicnet: Lightweight 3d hand pose estimation network based on biomechanical structure information for dexterous manipulator teleoperation. *IEEE Transactions on Cognitive and Developmental Systems*, 2022. 2

[30] Pengfei Ren, Chao Wen, Xiaozheng Zheng, Zhou Xue, Haifeng Sun, Qi Qi, Jingyu Wang, and Jianxin Liao. Decoupled iterative refinement framework for interacting hands reconstruction from a single rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8014–8025, 2023. 5

[31] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net: Localization-classification-regression for human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3433–3441, 2017. 2

[32] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1146–1161, 2019. 2

[33] Nicholas Santavas, Ioannis Kansizoglou, Loukas Bampis, Evangelos Karakasis, and Antonios Gasteratos. Attention! a lightweight 2d hand pose estimation approach. *IEEE Sensors Journal*, 21(10):11488–11496, 2020. 2

[34] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Humaniflow: Ancestor-conditioned normalising flows on so (3) manifolds for human pose and shape distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4779–4789, 2023. 1, 2

[35] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–98, 2018. 7

[36] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of machine learning research*, 2(Nov):67–93, 2001. 6

[37] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European conference on computer vision (ECCV)*, pages 529–545, 2018. 2

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[39] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 1

[40] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *Advances in neural information processing systems*, 27, 2014. 1, 2

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[42] Tom Wehrbein, Marco Rudolph, Bodo Rosenhahn, and Bastian Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11199–11208, 2021. 2

[43] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 2

[44] Yufei Wu, Xiaofei Ruan, Yu Zhang, Huang Zhou, Shengyu Du, and Gang Wu. Lightweight architecture for real-time hand pose estimation with deep supervision. *Symmetry*, 11(4):585, 2019. 2

[45] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019. 2

[46] Linlin Yang and Angela Yao. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9877–9886, 2019. 7

[47] Baowen Zhang, Yangang Wang, Xiaoming Deng, Yinda Zhang, Ping Tan, Cuixia Ma, and Hongan Wang. Interacting two-hand 3d pose and shape reconstruction from single color image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11354–11363, 2021. 2, 6, 7

[48] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1

[49] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017. 6, 7