Hand-Transformer: Non-Autoregressive Structured Modeling for 3D Hand Pose Estimation

Lin Huang[1](✉), Jianchao Tan[2], Ji Liu[2], and Junsong Yuan[1]

[1] State University of New York, Buffalo, USA
{lhuang27,jsyuan}@buffalo.edu
[2] Y-tech, Kwai Inc., Beijing, China
{jianchaotan,jiliu}@kuaishou.com

Abstract. 3D hand pose estimation is still far from a well-solved problem mainly due to the highly nonlinear dynamics of hand pose and the difficulties of modeling its inherent structural dependencies. To address this issue, we connect this structured output learning problem with the structured modeling framework in sequence transduction field. Standard transduction models like Transformer adopt an autoregressive connection to capture dependencies from previously generated tokens and further correlate this information with the input sequence in order to prioritize the set of relevant input tokens for current token generation. To borrow wisdom from this structured learning framework while avoiding the sequential modeling for hand pose, taking a 3D point set as input, we propose to leverage the Transformer architecture with a novel non-autoregressive structured decoding mechanism. Specifically, instead of using previously generated results, our decoder utilizes a reference hand pose to provide equivalent dependencies among hand joints for each output joint generation. By imposing the reference structural dependencies, we can correlate the information with the input 3D points through a multi-head attention mechanism, aiming to discover informative points from different perspectives, towards each hand joint localization. We demonstrate our model's effectiveness over multiple challenging hand pose datasets, comparing with several state-of-the-art methods.

Keywords: 3D hand pose estimation · Structured learning · Attention · Non-autoregressive transformer

1 Introduction

Articulated 3D hand pose estimation has been one of the most essential topics in computer vision because of its significant role in human behavior analysis and

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58595-2_2) contains supplementary material, which is available to authorized users.

© Springer Nature Switzerland AG 2020
A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12370, pp. 17–33, 2020.
https://doi.org/10.1007/978-3-030-58595-2_2

understanding, leading to enormous practical applications in human-computer interactions, robotics, and virtual/augmented reality, etc. With the advances of deep learning algorithms as well as the emergence of consumer level depth sensors, notable progress has been brought to 3D hand pose estimation field [14–16,18,26,28,30,32,36,39,40,43,44,47,50,51,54].

Despite significant success achieved in recent years, it is still challenging to obtain precise and robust hand pose due to complex pose variations, large variability in global orientation, self-similarity between fingers, and severe self-occlusion, etc. To tackle this structured output learning problem, we argue it is vital for learning algorithms to not only explore the intrinsic dependencies from input data, but also fully exploit the structural correlations among hand joints as well as its dependencies with input data, both of which has been fewly discussed. In our work, we focus on 3D point cloud as input, a simple yet effective representation converted from depth data, aiming to take advantage of these vital information from it.

To make use of above mentioned information towards hand pose estimation, we connect the articulated pose estimation problem with the sequence transduction tasks in Natural Language Processing (NLP) field. As another type of structured output prediction problem, state-of-the-art sequence transduction algorithms [1,41,48] fully exploit these correlations, following a classic encoder-decoder framework. They utilize an autoregressive decoding strategy to model sequential correlations among output tokens while also capturing global dependencies between the input and output sequence through attention mechanism. These modeling techniques have led to drastic performance improvements in generating syntactically and semantically valid sentences, such as language translations and image captions. Thus, to borrow wisdom from these strategies, we propose to leverage the Transformer model as our fundamental building block to take advantage of all these missing pieces for robust 3D hand pose estimation.

As a structured learning task, we should first pay attention to the inherent dependencies among hand joints since human hands are highly articulated and inherently structured. For instance, pinky finger cannot be bend without bending the ring finger or all fingers cannot bend backward too much [27]. Most current works simply treat pose as a set of independent 3D joints [14,16,30,32] while a few studies have enforced pose-related constraints in the form of either pre-trained kinematic models [22,30,31,53,54] or hand-crafted priors [42]. However, due to the large variations in hand motions, there are more correlations that cannot be captured via such pre-defined constraints. Thus, learning a model that can adaptively model the structural patterns is necessary for these cases. Inspired by the autoregressive decoding mechanism used in sequence transduction tasks, we can enforce pose patterns by conditioning each joint generation on previously generated joints. However, the autoregressive factorization nature results in heavy inference latency. In addition, given a specific order of hand joints, the sequential modeling assumes each joint is mainly correlated with "previous" joints in the order. However, hand joints should be inter-correlated with both "previous" and "future" ones. Thus, if we only consider sequential

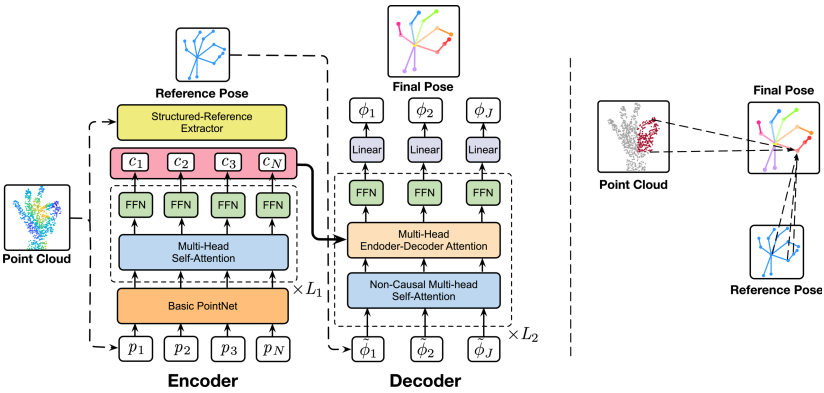

Fig. 1. Left: Overview of our proposed NARHT model composed of 3 components. The encoder computes point-wise features $\boldsymbol{c}_i$ for each input point $\boldsymbol{p}_i$. The structured-reference extractor will feed a reference pose $\tilde{\boldsymbol{\phi}}_j$ into decoder. Then decoder further models the dependencies among reference joints and correlate this information with point-wise features $\boldsymbol{c}_i$ for each joint generation $\boldsymbol{\phi}_j$. Right: An illustration of our non-autoregressive structured decoding strategy. Each joint generation is conditioned on the reference pose dependencies and relevant input points. N, J, L_1, and L_2 is the number of input points, hand joints, encoder layers, and decoder layers, respectively.

correlations, this might cause inferior and physically invalid poses due to biased modeling.

Motivated by recently proposed Non-AutoRegressive Transformer (NART) models [19,21,37,46], we propose to replace the autoregressive factorization of the Transformer with a novel non-autoregressive structured learning mechanism designed for 3D hand pose estimation. Instead of using previously generated tokens as decoder input, representative NART models directly feed a modified copy of input tokens to decoder, aiming to generate all output tokens simultaneously. Obviously, it provides drastic inference speedup but comes at the cost of performance degradation due to the removal of information from output tokens. To preserve the parallelism while feeding necessary pose-related information to the decoder, we design a structured-reference extractor, aiming to provide a reference hand pose in the form of joint-wise features and use its inherent correlations to approximate that of output pose. Thus, given the reference pose to the decoder, we adopt a non-causal self-attention layer [19] to capture its inherent dependencies towards each output joint generation. By exposing the extracted reference pose to the decoder, our model is able to generate all joints in parallel, conditioned on pose-related information.

Beyond drawing the dependencies from structured input and output data, respectively, the Transformer network further models the correlations between the input and output to explore the relevant input information. By modeling the correlations, what each output token generation can access is not only its relation with previously generated tokens but also the informative input fea-

tures. Motivated by this strategy, our Transformer-based model also correlates each output joint generation with the input points via an multi-head attention mechanism. Specifically, for certain joint estimation, we utilize the dependencies among reference hand joints as queries to attend over input points. The goal is to adaptively discover informative points that contribute towards each joint generation, from different representation subspaces. Then, we merge the attention-weighted information along with the dependencies among reference joints to localize certain output joint. This scheme is similar to current state-of-the-art voting-based techniques [18,26,44,47] which mainly take the pairwise point to point Euclidean offsets as vote scores. Nevertheless, we can easily find cases where points with small Euclidean but large Geodesic offsets can contribute less than those with small Geodesic but large Euclidean offsets. Thus, this strategy might lead to sub-optimal results. Instead, the multi-head attention generalizes the offsets-based techniques by letting the model decide itself regarding which aspects to look at towards certain joint generation. Thus, we argue our method also extends this line of work to a more adaptive version with more various aspects being examined, based on the multi-head attention mechanism.

In summary, our main contributions are shown as follows:

- We propose a novel Non-AutoRegressive Hand Transformer (NARHT) for 3D hand pose estimation from unordered point sets. To the best of our knowledge, it is the very first attempt to connect the structured hand pose estimation with the Transformer-based transduction frameworks in NLP field.
- We design a non-autoregressive structured decoding strategy specifically for articulated pose estimation to replace the autoregressive factorization of traditional Transformer, aiming to break the sequential inference bottleneck and provide necessary pose information during the decoding process.
- Using pose dependencies as queries, we further implement a fully adaptive point-wise voting scheme through a multi-head attention mechanism. This scheme correlates the captured pose dependencies for each output joint with input points from different aspects, contributing to precise joint localization.

2 Related Work

3D Hand Pose Estimation. 3D hand pose estimation has received much attention in computer vision over the last decade. The developed approaches can be categorized into three types: generative approaches, discriminative approaches, and hybrid approaches. Our method is more related with the discriminative line of works. Most discriminative approaches [6,12,32,36,40,43,44,44,47] directly feed 2D depth maps as images into 2D CNNs. Nevertheless, the mismatch between the 2.5D depth data and the 2D learning algorithms cannot guarantee a full exploration of the input 3D geometric information. Subsequent methods [15] project depth maps into multi-views and feed them into multi-view CNNs. However, the separate pipelines with multi-view fusion is non-trivial to deal with. To further address the problem, 3D voxels [11,16,28] comes into play

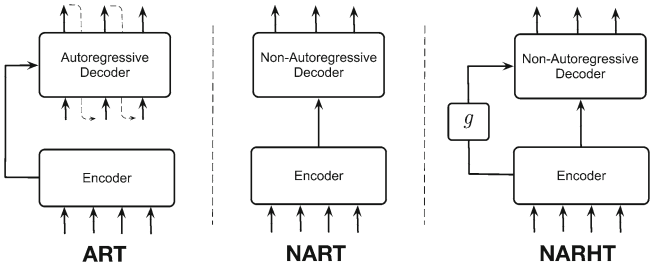

Fig. 2. Left: Classic encoder-decoder framework in sequence transduction tasks, such as AutoRegressive Transformer (ART) [41]. Middle: Recently proposed encoder-decoder framework without autoregressive connection, such as Non-AutoRegressive Transformer (NART) [19,21,37,46]. Right: Our proposed encoder-decoder framework with additional function g as the structured-reference extractor (NARHT).

for direct 3D geometric modeling. Moon et al. [28], which is still one of the state-of-the-art methods, exploits voxel-to-voxel predictions to estimate the per-voxel likelihood for each joint. But the volumteric pipeline causes high computation burden due to the need of large memory. Recent years, there is an obvious trend shifting towards RGB-based solutions [3,4,17,23,29,49,52,57] because of the convenience of data acquisition. However, the ambiguities in single RGB camera and the lack of texture features make current techniques still far more ubiquitous than depth-based methods. A lot of attentions have also been paid to the 3D point-based techniques [7,8,14,18,26]. Ge et al. [14] proposes a hand PointNet for directly mapping the unordered point sets to 3D hand poses. But as an irregular data format, more efforts can be applied in order to fully explore its geometric information. In this paper, motivated by several sequence transduction algorithms [1,19,21,37,41,46,48], we propose a novel mechanism for directly operating on input points while injecting the pose-related dependencies for robust pose estimation.

3 Methodology

Our proposed NARHT model is illustrated in Fig. 1. Given a set of unordered 3D points converted from hand depth image, our target is to infer a corresponding 3D hand pose, which is parameterized as a set of 3D joint coordinates $\mathbf{\Phi}^{cam} = \{\boldsymbol{\phi}_1^{cam}, ..., \boldsymbol{\phi}_J^{cam}\}$ in the camera Coordinate System (C.S.), where J is the number of hand joints. To obtain robust hand poses, we propose to leverage a Transformer-based architecture with a non-autoregressive structured decoding strategy. The model follows the typical encoder-decoder frameworks [2,9,38,41], with an additional structured-reference extractor for non-autoregressive decoding. Specifically, following [14,18] to make our model more robust to hand orientations, we first downsample and normalize the input point set to N points in OBB C.S., represented as $\mathcal{P} = \{\boldsymbol{p}_1, ..., \boldsymbol{p}_N\}$. The hand joints $\mathbf{\Phi}^{cam}$ is also transformed into OBB C.S., denoted as $\mathbf{\Phi} = \{\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_J\}$. Then,

we feed the normalized points $\mathcal{P}$ into the encoder to generate enhanced point-wise representations $\mathcal{C} = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_N\}$. We utilize a basic PointNet [33], followed by a permutation-invariant self-attention layer to better capture the long-range dependencies among input points. During decoding, to impose structured pose patterns for each joint generation, we first adopt a structured-reference extractor to take the input points and generate a reference hand pose in the form of joint-wise features $\tilde{\mathbf{\Phi}} = \{\tilde{\boldsymbol{\phi}}_1, ..., \tilde{\boldsymbol{\phi}}_J\}$. The reference pose is further exposed to the decoder and a non-causal self-attention layer [19] is used to capture the structural dependencies among its joints, which serves as an approximation to that of the target hand pose. Then, we correlate the captured pose dependencies with the point-wise features $\mathcal{C}$ to discover the relevant points from different representation subspaces towards certain joint generation, using the encoder-decoder attention module. Finally, we merge all the attention-weighted point-wise features along with the captured dependencies to infer each joint location in parallel. Based on this decoding strategy, our model is able to simultaneously generate all joints conditioned on pose-related information.

3.1 Tranformer Revisited

For structured output learning problems, besides extracting features from input data, we should always investigate the inherent dependencies of structured output as well as its correlations with the input data in order to generate precise and valid results, such as 3D articulated poses, language translations, and image captions. Transformer [41], established as state-of-the-art transduction model, exactly takes advantage of all these information with a solely attention-based mechanism to generate syntactically and semantically correct sentences. In particular, AutoRegressive Transformer (ART), following the classic encoder-decoder frameworks, adopts self-attention layer to first capture long-range dependencies from input structured data and previously generated output tokens, respectively. Then it further utilizes encoder-decoder attention to model dependencies between input and previously generated output. Finally, Transformer can sequentially generate token conditioned on the captured information.

The superior performance achieved by Transformer mainly comes down to the combination of autoregressive decoding with the attention mechanism for modeling the structural dependencies from data. Therefore, to translate this framework into our case, we keep the attention mechanism while extending the autoregressive decoding to a more suited strategy for 3D hand pose estimation. In the following sections, we continue revisiting both key concepts.

Multi-Head Attention. Attention mechanism [1,24] is used to adaptively aggregate the set of input values without regard to their distance, according to the attention weights that measure the compatibility of given query with a set of keys. Formally, we first assume the dimension of each query and key is d_k and dimension of value is d_v, then a scaled dot-product attention mapping [41] can be computed as:

$$Attention\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{T}}{\sqrt{d_k}}\right)\boldsymbol{V}, \tag{1}$$

where matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ denote a set of queries, keys, and values, respectively. Moreover, to extend the capacity of exploring different subspaces, the attention can be extended to cases with multi-head [41]:

$$\begin{cases} MultiHead\left(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}\right)=Concat\left(head_1,...,head_h\right)\boldsymbol{W^O}, \\ head_i=Attention\left(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}\right); \end{cases} \tag{2}$$

where linear transformations $\boldsymbol{W}_i^{\boldsymbol{Q}} \in \mathbb{R}^{d_{model}\times d_k}$, $\boldsymbol{W}_i^{\boldsymbol{K}} \in \mathbb{R}^{d_{model}\times d_k}$, $\boldsymbol{W}_i^{\boldsymbol{V}} \in \mathbb{R}^{d_{model}\times d_v}$, and $\boldsymbol{W^O} \in \mathbb{R}^{hd_v\times d_{model}}$ are parameter matrices. h is the number of subspaces and $d_k=d_v=d_{model}/h=32$ in our implementation. We mainly rely on two variants of attention in our model. The first one is self-attention for capturing long-range dependencies from input and reference hand pose, respectively. The keys and queries are from same set of elements. The second one is encoder-decoder attention to prioritize the subset of input points where relevant information is present for certain joint generation. The keys and queries are from two sets of elements.

Autoregressive Decoding. As mentioned previously, Transformer generates each token in an autoregressive manner. Thus, the decoding mechanism conditions the generation of each token on the inherent dependencies among previously generated tokens, as shown in Fig. 2. Formally, given a source sentence $\boldsymbol{X}=\{\boldsymbol{x}_1,...,\boldsymbol{x}_{T'}\}$ with length T', the autoregressive decoding factors the distribution of output sequence $\boldsymbol{Y}=\{\boldsymbol{y}_1,...,\boldsymbol{y}_T\}$ with length T into a chain of conditional probabilities with a left-to-right sequential structure [19,41]:

$$p_{art}\left(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\theta}\right)=\prod_{t=1}^{T}p\left(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{x}_{1:T'};\boldsymbol{\theta}\right), \tag{3}$$

where $\boldsymbol{\theta}$ is the model parameters. As shown in Eq. 3, despite its ability to capture inherent dependencies from output sequence, the autoregressive decoding suffers from high inference latency since the generation of tth token $\boldsymbol{y}_t$ depends on previously generated tokens $\boldsymbol{y}_{1:t-1}$. In addition, the sequential modeling might not work well towards hand motion, leading to sub-optimal results, since hand joints are inter-correlated with each other rather than constrained only in a sequential manner. Thus, we propose a non-autoregressive structured decoding mechanism to replace the slow sequential modeling process and enforce more reasonable pose dependencies into decoding process.

3.2 Non-Autoregressive Structured Decoding

Recently proposed Non-AutoRegressive Transformer (NART) models [19,21,37] remove the sequential dependence on previously generated tokens and directly

feed a modified copy of input sequence $\boldsymbol{X}' = \{\boldsymbol{x}'_1, ..., \boldsymbol{x}'_{T'}\}$ to decoder, as shown in Fig. 2. It can achieve significant inference speedup, however, at the cost of inferior accuracy compared to ART models due to the lack of information from output sequence. The decoding process can be formulated as:

$$p_{nart}\left(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\theta}\right) = \prod_{t=1}^{T} p\left(\boldsymbol{y}_t|\boldsymbol{x}'_{1:T'}, \boldsymbol{x}_{1:T'};\boldsymbol{\theta}\right). \tag{4}$$

This motivates us to come up with a similar architecture with the ability to explore the structured output patterns and run in parallel. We propose a novel non-autoregressive structured learning mechanism designed for 3D hand pose estimation. Instead of feeding previously generated results or modified copy of input, we let the decoder take a reference pose in the form of joint-wise features $\tilde{\boldsymbol{\Phi}} = \{\tilde{\boldsymbol{\phi}}_1, ..., \tilde{\boldsymbol{\phi}}_J\}$, generated by a structured-reference extractor. Exposing the reference pose to the decoder, we further employ a non-causal self-attention layer [19] for drawing the correlations among reference joints. Guided by the captured reference pose dependencies, we then adopt the encoder-decoder attention mechanism used in the Transformer to discover the informative input points from different representation subspaces. This presents a fully adaptive point-wise voting scheme, aiming to better capture correlations between output joints and input points. We aggregate the weighted point-wise features along with the pose dependencies and pass it through a Position-wise Feed-Forward Network (FFN) [41] to obtain the decoder output. Finally, the decoder output will go through a Fully Connected (FC) layer to obtain each joint coordinates. The decoding process can be formulated as:

$$p_{narht}\left(\boldsymbol{\Phi}|\mathcal{P};\boldsymbol{\theta}\right) = \prod_{j=1}^{J} p\left(\boldsymbol{\phi}_j|\tilde{\boldsymbol{\phi}}_{1:J}, \boldsymbol{p}_{1:N};\boldsymbol{\theta}\right), \tag{5}$$

where $\boldsymbol{\theta}$ is the model parameters. In this manner, our model can simultaneously generate all hand joints conditioned on the necessary pose patterns. For training, we adopt maximum likelihood estimation method with a squared L2 loss between the ground truth $\hat{\boldsymbol{\Phi}} = \{\hat{\boldsymbol{\phi}}_1, ..., \hat{\boldsymbol{\phi}}_J\}$ and the estimated joint coordinates. The loss for each training sample is defined as:

$$\mathcal{L}_1 = \sum_{j=1}^{J} \|\boldsymbol{\phi}_j - \hat{\boldsymbol{\phi}}_j\|_2^2. \tag{6}$$

Structured-Reference Extractor. In order to expose more reasonable pose-related information to the decoder, we replace the inefficient autoregressive factorization nature of Transformer model with a novel non-autoregressive structured learning mechanism. As shown in Fig. 1, we feed the normalized 3D points into a structured-reference extractor in the goal to generate a reference hand pose in the form of joint-wise features $\tilde{\boldsymbol{\Phi}} = \{\tilde{\boldsymbol{\phi}}_1, ..., \tilde{\boldsymbol{\phi}}_J\}$. By exposing the reference pose to the decoder, the decoder can capture the correlations among

reference joints as an approximation to that of the target pose and use this information to better constrain the output space, leading to more precise and physically valid hand joints.

Specifically, we adopt a PointNet++-based network [34] to map the input points $\mathcal{P}$ to a latent feature vector and transform it into J points with 64-dim features. We then pass it through a MLP network to obtain J points with d_{model}-dim features, which gives us reference pose in the form of joint-wise features.

We also apply an intermediate supervision to encourage the reference pose to include more information regarding the ground truth. Thus, we add a FC layer for regressing a hand pose $\mathbf{\Phi}' = \{\boldsymbol{\phi}'_1, ..., \boldsymbol{\phi}'_J\}$ from the joint-wise features. The second loss term is given below:

$$\mathcal{L}_2 = \sum_{j=1}^{J} \|\boldsymbol{\phi}'_j - \hat{\boldsymbol{\phi}}_j\|_2^2. \tag{7}$$

Non-Causal Self-Attention. Conventional Transformer conditions each output token generation on previously generated results without the access for the information from the future decoding steps. Given the reference pose output from the sturcutred-reference extractor, the decoder in our model could avoid this autoregressive factorization and explore the dependencies among all J reference hand joints. Therefore, we can avoid the causal mask used in the masked self-attention module of the traditional Transformer decoder. Moreover, similar to [19], we mask out each reference joint's position only from attending to itself, aiming to model the dependencies among reference joints without seeing itself.

Point-Wise Voting. Another key benefit of using the Transformer model is that it can model the global dependencies between the input and output tokens via a encoder-decoder attention mechanism. This strategy can be extended to a fully adaptive point-wise voting scheme for hand joint localization. Specifically, in our model, we utilize the captured dependencies among reference joints as queries to attend over all input points for each output joint generation. It will put strong focus on parts of the input points and help the decoding process select the informative points that can contribute to certain joint generation. Compared with the popular Euclidean offsets-based voting-scheme [18,26,44,47], the attention-based mechanism is more adaptive and comprehensive. Moreover, the multi-head self-attention mechanism enables the voting-scheme to be performed from different representation subspaces, adding more perspectives to the relevant point searching. The per-point votes will be merged with the captured dependencies to decide each 3D joint location.

3.3 Encoder

The goal for our encoder is similar to that in the Transformer, which is to draw long-range dependencies from the input data and compute point-wise representation. Various methods have been proposed for direct operation on point cloud.

Table 1. Comparison with state-of-the-art methods on ICVL [39] (Left), MSRA [36] (Middle), and NYU [40] (Right). "Error" indicates the mean joint distance error in (mm).

Methods	Error
DeepModel [53]	11.56
DeepPrior [31]	10.40
CrossingNets [43]	10.20
HBE [55]	8.62
DeepPrior++ [30]	8.10
REN ($9 \times 6 \times 6$) [20]	7.31
DenseReg [44]	7.24
SHPR-Net [7]	7.22
Hand PointNet [14]	6.94
Pose-REN [6]	6.79
CrossInfoNet [12]	6.73
A2J [47]	6.46
Point-to-Point [18]	6.33
V2V-PoseNet [28]	6.28
NARHT (Ours)	6.47

Methods	Error
Feedback [32]	15.97
CrossingNets [43]	12.20
REN (9x6x6) [20]	9.79
3D CNN [16]	9.58
DeepPrior++ [30]	9.50
Pose-REN [6]	8.65
Hand PointNet [14]	8.51
CrossInfoNet [12]	7.86
SHPR-Net [7]	7.76
Point-to-Point [18]	7.71
V2V-PoseNet [28]	7.59
DenseReg [44]	7.23
NARHT (Ours)	7.55

Methods	Error
DeepPrior [31]	20.75
DeepModel [53]	17.04
Feedback [32]	15.97
CrossingNets [43]	15.50
3D CNN [16]	14.11
REN (9x6x6) [20]	12.69
DeepPrior++ [30]	12.24
Pose-REN [6]	11.81
SHPR-Net [7]	10.78
Hand PointNet [14]	10.54
DenseReg [44]	10.21
CrossInfoNet [12]	10.08
Point-to-Point [18]	9.05
Point-to-Pose [26]	8.99
A2J [47]	8.61
V2V-PoseNet [28]	8.42
NARHT (Ours)	9.80

The classic PointNet [33] operates on each point independently without consideration of the inherent geometric correlations. Subsequent methods mainly rely on convolution-based multi-resolution hierarchy to resolve this issue. However, many recent studies [5,10,35,45,56] have shown convolution-based design is inefficient to capture long-range dependencies while also causing optimization difficulties. In addition, the transformation invariance caused by the widely used pooling operation tends to cause loss of precise localization information which is vital for articulated pose estimation. Thus, in our work, we first feed the input points into a basic PointNet-based network [33] to extract basic point-wise features and further adopt self-attention layer to enhance the representation by modeling the inherent dependencies among different points.

3.4 End-to-End Training

We utilize loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ mentioned above to jointly supervise the end-to-end learning procedure of our NARHT model, which is formulated as:

$$\mathcal{L} = \lambda \mathcal{L}_1 + \mathcal{L}_2, \tag{8}$$

where $\lambda = 10$ is the weight coefficient to balance $\mathcal{L}_1$ and $\mathcal{L}_2$.

4 Experiments

4.1 Datasets

ICVL Dataset [39]**.** It contains 22k frames for training and 1.5k frames for testing. The dataset also includes an additional 300k augmented frames with in-

Table 2. Left: Ablation study for several components of our Non-AutoRegressive Hand Transformer (NARHT) on ICVL [39]. Right: Comparison of inference time on single GPU.

Dataset	Compoent	Error
ICVL	Autoregressive transformer	8.57
	P2P as encoder	6.70
	Coordinate-based reference pose	6.67
	NARHT (Ours)	6.47

Methods	FPS (single GPU)
V2V-PoseNet[28]	3.5
DenseReg [14]	27.8
Point-to-Point [18]	41.8
NARHT (Ours)	43.2

plane rotations. The dataset provides 16 annotated joints. **MSRA Dataset** [36]**.** It consists of 76.5k depth images captured from 9 subjects. Each subject contains 17 hand gestures and each hand gesture has about 500 frames with segmented hand depth image. The ground truth annotations contains 21 joints. We adopt the common leave-one-subject-out cross-validation strategy for evaluation on this dataset. **NYU Dataset** [40]**.** It 72K training 8.2K testing frames. For each frame, the RGBD data from three Kinects is provided. Following the common protocol, we only use the frontal view with a subset of 14 ground truth joints. **HANDS 2017 Dataset** [50]**.** It consists of 957k training and 295k testing frames, which are sampled from BigHand2.2M [51] and FHAD [13] datasets. The testing set has seen subjects in training set and unseen subjects. The dataset provides 21 annotated 3D joints.

4.2 Evaluation Metrics

We adopt two most commonly used metrics in literature to evaluate the performance of 3D hand pose estimation. The first metric is the 3D per-joint Euclidean distance mean error (in mm) on all test frames as well as the overall 3D Euclidean distance mean error (in mm) of each frames' total joints across all test frames. This metric demonstrates the overall performance of each estimated joint and hand pose. The second metric is the fraction of good frames that have all joints within a specified distance to ground truth. This metric is considered more strict, which better indicates the performance of a given estimation technique.

4.3 Implementation Details

Input. We set the number of sampled points as $N = 1024$ and also concatenate each input 3D coordinate with estimated 3D surface normal. **Encoder.** We adopt a basic PointNet-based Network [33] followed by a standard Transformer encoder. The PointNet structure consists of 1 MaxPool layer and 2 MLP networks. Each MLP is composed of 3 FC layers. We use the Transformer encoder with headers h as 8, d_{model} as 256, layer number L_1 as 3 and do not use Position Encoding module. **Decoder.** We adopt PointNet++-based Network [34] as Structured-Reference Extractor and a modified standard Transformer decoder. The PointNet++ structure consists of 3 set abstraction layers followed by 1 MaxPool layer and 1 MLP for extracting joint-wise features. For the modified

Transformer decoder, we replace the Masked Multi-Head Attention layer with a Non-Causal Multi-Head Attention layer [19]. We set headers h as 8, d_{model} as 256, layer number L_2 as 6. Position Encoding module is not used. We only have 1 FC layer as the final layer to convert the decoder output to joint coordinates.
Training. For training NARHT, we use Adam [25] optimizer with initial learning rate as 1e–3, λ as 10. The learning rate is divided by 10 after 40 epochs. Following [14,18], we adopt similar strategies for data augmentation with random arm lengths and random stretch factors. All experiments were conducted on single NVIDIA TITAN Xp GPU using PyTorch framework, with the batch size of 16 for training and evaluation.

4.4 Ablation Study

We choose ICVL dataset to conduct ablation study and evaluate the results using mean joint distance error in (mm) metric. The results are shown in Table 2.

Effectiveness of Non-Autoregressive Structured Decoding. To verify the effectiveness of our proposed non-autoregressive stuctured decoding, we compare our NARHT model with Autoregressive Transformer-based (ART) model, which is to use the original Transformer [41] for 3D hand pose estimation. We implement with scheduled sampling [2] as the training strategy for ART model. With the autoregressive decoding, we can see a obvious performance drop. More importantly, the ART model runs much slower than our NARHT model due to the sequential modeling. These help demonstrate that hand joints should not be modeled only in a sequential manner and our non-autoregressive decoding process can use the reference pose dependencies for better joint localization.

Impact of the Representation of Reference Pose. We examine the impact of the representation used for reference pose. The reference pose is generated by the structured-reference extractor and fed into our decoder. Thus we could use either joint-wise coordinates or joint-wise features as decoder input. According to the results, the coordinates-based representation is inferior to the feature-based representation, which might be caused by the lack of flexibility for the coordinates-based format.

Effectiveness of Self-Attention Layer. To demonstrate the effectiveness of self-attention layer for drawing long-range dependencies from input points, we also implement a model using the Point-to-Point [18] architecture as encoder without any self-attention layers. Besides the mean distance error, the runtimes for both encoders per frame on average are compared. Although the improvement (0.23 mm) shown in Table 2 is small, our attention-based encoder (3.80 ms) has much higher running efficiency than point-to-point-based encoder (35.90 ms). This verifies our point that self-attention layer can better capture long-range dependencies in a much more efficient manner.

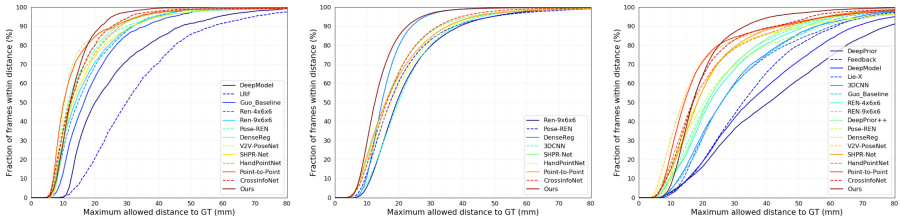

Fig. 3. Comparison with state-of-the-art methods on ICVL [39] (Left), MSRA [36] (Middle), and NYU [40] (Right) datasets. The proportions of good frames is used for comparison.

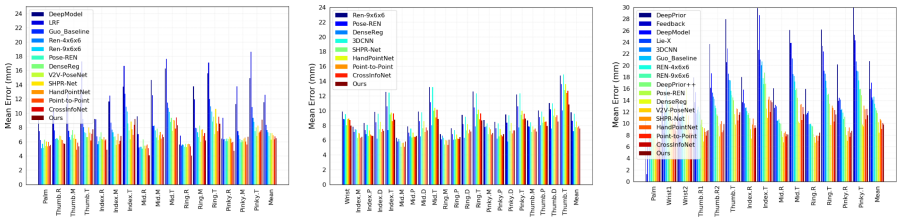

Fig. 4. Comparison with state-of-the-art methods on ICVL [39] (Left), MSRA [36] (Middle), and NYU [40] (Right) datasets. The per-joint mean error distances is used for comparison (R: root, T: tip).

4.5 Comparisons with the State-of-the-Arts

We compare the performance of the proposed NARHT on multiple public 3D hand pose datasets with most of state-of-the-art methods, including 2D and 3D-based approaches [6,7,12,14,16,18,20,28,30–32,43,44,47,53,55]. The comprehensive experimental results are given in Fig. 3 on fraction of good frames over different thresholds, Fig. 4 on per-joint mean error (mm), Table 1 on mean joint distance error (mm), and Table 2 on inference speed.

Before we perform specific analysis for each dataset. We want to point out, in terms of the percentage of good frames, when the error threshold is larger 20 mm for ICVL, 5 mm for MSRA, 30 mm for NYU, our method is superior to previous state-of-the-art methods by a certain margin. This reveals, compared with other methods, our model has the most number of good frames with all estimated joints within a certain range deviated from the ground truth joint. This also verifies the robustness of our model and meets our expectation since we explicitly enforce necessary structured pose patterns into the decoding process. Some qualitative results for three datasets are also presented in Fig. 5.

ICVL. Our method can outperform other methods except for the A2J [47], Point-to-Point [18], and V2V-PoseNet [28] on overall mean distance error shown in Table 1. However, as shown in Table 2, while our mean error is slightly infe-

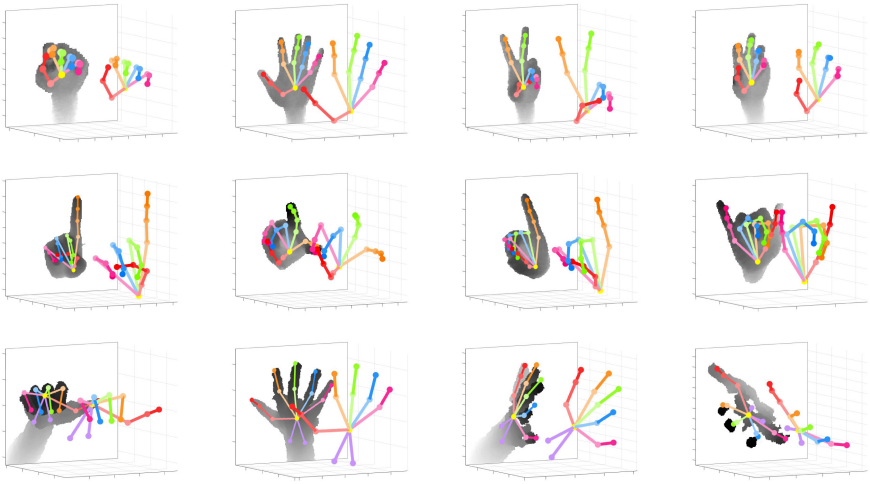

Fig. 5. Qualitative results for ICVL [39] (Top), MSRA [36] (Middle) and NYU [40] (Bottom) datasets.

rior to V2V-PoseNet, Point-to-Point, our method has higher inference efficiency, especially compared with V2V-PoseNet.

MSRA. Our method is superior to the current methods except for DenseReg [44] on the overall mean distance error. However, as mentioned above, our method has the best fraction of good frames when the threshold is larger 5 mm on MSRA and better results over almost all thresholds than DenseReg. More importantly, although our method is 3D-oriented, our model still runs much faster than DenseReg according to Table 2.

NYU. In terms of the overall mean error distances, our method in most cases outperforms current state-of-the-art models, except for A2J, Point-to-Pose [26], V2V-PoseNet, and Point-to-Point. However, our model is superior to Point-to-Point and V2V-PoseNet on fraction of good frames when the threshold is larger 30 mm by a large margin. Since we do not have the curve for A2J and Point-to-Pose, we cannot compare on this aspect.

HANDS 2017. We also compare with A2J, Point-to-Pose, V2V-PoseNet, and Hand PointNet [14] on HANDS 2017. While our mean distance error is inferior to A2J, Point-to-Pose, and V2V-PoseNet on seen cases, our model outperforms other methods except for A2J on unseen data.

5 Conclusion

In this paper, we propose to connect structured hand pose estimation problem with the sequence transduction tasks in NLP field in the goal to fully investigate

related structural information for precise pose prediction. Following the Transformer framework and proposed non-autoregressive decoding strategy, we can condition each joint generation on necessary pose dependencies as well as selective input features. Experimental results on multiple challenging datasets verify the effectiveness of our model, comparing to state-of-the-art methods in real-time performance. In the future, we plan to explore more possibilities regarding bridging the gap between the structured output learning problems in pose estimation and NLP fields, such as pose estimation from RGB images and image captioning.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: NIPS (2015)
3. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3D hand pose estimation from monocular RGB images. In: ECCV (2018)
4. Cai, Y., et al.: Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In: ICCV (2019)
5. Chaudhari, S., Polatkan, G., Ramanath, R., Mithal, V.: An attentive survey of attention models. arXiv preprint arXiv:1904.02874 (2019)
6. Chen, X., Wang, G., Guo, H., Zhang, C.: Pose guided structured region ensemble network for cascaded hand pose estimation. Neurocomputing **395**, 138–149 (2019)
7. Chen, X., Wang, G., Zhang, C., Kim, T.K., Ji, X.: Shpr-net: deep semantic hand pose regression from point clouds. IEEE Access **6**, 43425–43439 (2018)
8. Chen, Y., Tu, Z., Ge, L., Zhang, D., Chen, R., Yuan, J.: SO-HandNet: self-organizing network for 3D hand pose estimation with semi-supervised learning. In: ICCV (2019)
9. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
10. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. In: ICLR (2019)
11. Deng, X., Yang, S., Zhang, Y., Tan, P., Chang, L., Wang, H.: Hand3D: hand pose estimation using 3D neural network. arXiv preprint arXiv:1704.02224 (2017)
12. Du, K., Lin, X., Sun, Y., Ma, X.: Crossinfonet: multi-task information sharing based hand pose estimation. In: CVPR (2019)
13. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with RGB-d videos and 3D hand pose annotations. In: CVPR (2018)
14. Ge, L., Cai, Y., Weng, J., Yuan, J.: Hand pointnet: 3D hand pose estimation using point sets. In: CVPR (2018)
15. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs. In: CVPR (2016)
16. Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: CVPR (2017)
17. Ge, L., et al.: 3D hand shape and pose estimation from a single RGB image. In: CVPR (2019)

18. Ge, L., Ren, Z., Yuan, J.: Point-to-point regression pointnet for 3D hand pose estimation. In: ECCV (2018)
19. Gu, J., Bradbury, J., Xiong, C., Li, V.O., Socher, R.: Non-autoregressive neural machine translation. arXiv preprint arXiv:1711.02281 (2017)
20. Guo, H., Wang, G., Chen, X., Zhang, C., Qiao, F., Yang, H.: Region ensemble network: improving convolutional network for hand pose estimation. In: ICIP (2017)
21. Guo, J., Tan, X., He, D., Qin, T., Xu, L., Liu, T.Y.: Non-autoregressive neural machine translation with enhanced decoder input. In: AAAI (2019)
22. Hasson, Y., et al.: Learning joint reconstruction of hands and manipulated objects. In: CVPR (2019)
23. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5 d heatmap regression. In: ECCV (2018)
24. Kim, Y., Denton, C., Hoang, L., Rush, A.M.: Structured attention networks. arXiv preprint arXiv:1702.00887 (2017)
25. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
26. Li, S., Lee, D.: Point-to-pose voting based hand pose estimation using residual permutation equivariant layer. In: CVPR (2019)
27. Lin, J., Wu, Y., Huang, T.S.: Modeling the constraints of human hand motion. In: Proceedings Workshop on Human Motion (2000)
28. Moon, G., Chang, J., Lee, K.M.: V2V-PoseNet: voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In: CVPR (2018)
29. Mueller, F., et al.: Ganerated hands for real-time 3D hand tracking from monocular RGB. In: CVPR (2018)
30. Oberweger, M., Lepetit, V.: DeepPrior++: improving fast and accurate 3D hand pose estimation. In: ICCV Workshop (2017)
31. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. In: CVWW (2015)
32. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: ICCV (2015)
33. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
34. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: NIPS (2017)
35. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. In: NIPS (2019)
36. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR (2015)
37. Sun, Z., Li, Z., Wang, H., He, D., Lin, Z., Deng, Z.: Fast structured decoding for sequence models. In: NIPS (2019)
38. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
39. Tang, D., Jin Chang, H., Tejani, A., Kim, T.K.: Latent regression forest: structured estimation of 3D articulated hand posture. In: CVPR (2014)
40. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. (TOG) **33**(5), 169 (2014)
41. Vaswani, A., et al.: Attention is all you need. In: NIPS (2017)
42. Wan, C., Probst, T., Gool, L.V., Yao, A.: Self-supervised 3D hand pose estimation through training by fitting. In: CVPR (2019)

43. Wan, C., Probst, T., Van Gool, L., Yao, A.: Crossing nets: dual generative models with a shared latent space for hand pose estimation. In: CVPR (2017)
44. Wan, C., Probst, T., Van Gool, L., Yao, A.: Dense 3D regression for hand pose estimation. In: CVPR (2018)
45. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: CVPR (2018)
46. Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., Liu, T.Y.: Non-autoregressive machine translation with auxiliary regularization. In: AAAI (2019)
47. Xiong, F., et al.: A2J: anchor-to-joint regression network for 3D articulated pose estimation from a single depth image. In: ICCV (2019)
48. Xu, K., et al.: Show, attend and tell: neural image caption generation with visual attention. In: ICML (2015)
49. Yang, L., Li, S., Lee, D., Yao, A.: Aligning latent spaces for 3D hand pose estimation. In: ICCV (2019)
50. Yuan, S., Ye, Q., Garcia-Hernando, G., Kim, T.K.: The 2017 hands in the million challenge on 3D hand pose estimation. arXiv preprint arXiv:1707.02237 (2017)
51. Yuan, S., Ye, Q., Stenger, B., Jain, S., Kim, T.K.: BigHand2.2M benchmark: hand pose dataset and state of the art analysis. In: CVPR (2017)
52. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular RGB image. In: ICCV (2019)
53. Zhou, X., Sun, X., Zhang, W., Liang, S., Wei, Y.: Deep kinematic pose regression. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 186–201. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_17
54. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: IJCAI (2016)
55. Zhou, Y., Lu, J., Du, K., Lin, X., Sun, Y., Ma, X.: HBE: hand branch ensemble network for real-time 3D hand pose estimation. In: ECCV (2018)
56. Zhu, X., Cheng, D., Zhang, Z., Lin, S., Dai, J.: An empirical study of spatial attention mechanisms in deep networks. In: ICCV (2019)
57. Zimmermann, C., Brox, T.: Learning to estimate 3D hand pose from single RGB images. In: ICCV (2017)