Adaptive Computationally Efficient Network for Monocular 3D Hand Pose Estimation

Zhipeng Fan[1], Jun Liu[2⋆], and Yao Wang[1]

[1] Tandon School of Engineering, New York University, Brooklyn NY, USA
{zf606, yw523}@nyu.edu
[2] Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore
jun_liu@sutd.edu.sg

Abstract. 3D hand pose estimation is an important task for a wide range of real-world applications. Existing works in this domain mainly focus on designing advanced algorithms to achieve high pose estimation accuracy. However, besides accuracy, the computation efficiency that affects the computation speed and power consumption is also crucial for real-world applications. In this paper, we investigate the problem of reducing the overall computation cost yet maintaining the high accuracy for 3D hand pose estimation from video sequences. A novel model, called Adaptive Computationally Efficient (ACE) network, is proposed, which takes advantage of a Gaussian kernel based Gate Module to dynamically switch the computation between a light model and a heavy network for feature extraction. Our model employs the light model to compute efficient features for most of the frames and invokes the heavy model only when necessary. Combined with the temporal context, the proposed model accurately estimates the 3D hand pose. We evaluate our model on two publicly available datasets, and achieve state-of-the-art performance at 22% of the computation cost compared to traditional temporal models.

Keywords: 3D Hand Pose Estimation, Computation Efficiency, Dynamic Adaption, Gaussian Gate

1 Introduction

Understanding human hand poses is a long lasting problem in computer vision community, due to the great amount of potential applications in action recognition, AR/VR [28], robotics and human computer interactions (HCI) [11]. The problem of inferring 3D configurations of human hands from images and videos is inherently challenging because of the frequent self-occlusion and the large variance of hand poses. A large body of existing works address the problem of hand pose estimation from depth data [7, 37], as it reduces ambiguities in the

⋆ Corresponding author.

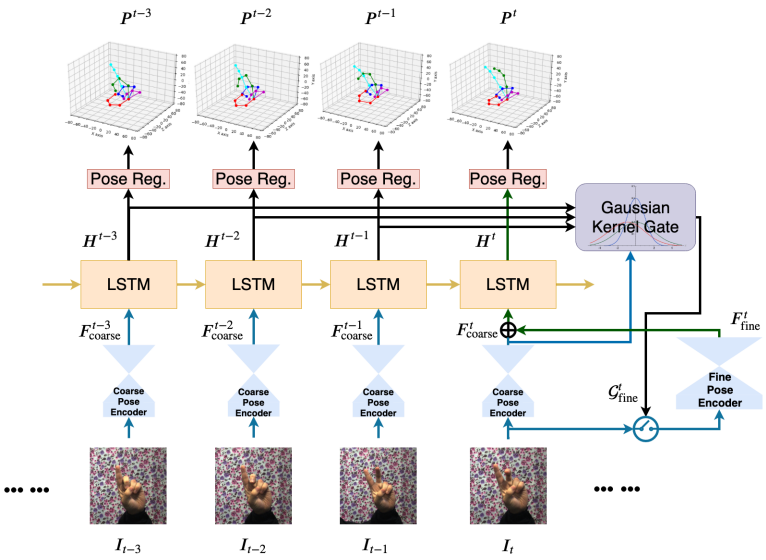


Fig. 1: Illustration of our Adaptive Computationally Efficient (ACE) network. In most of the time, the LSTM takes features from the coarse pose encoder and refines the predicted pose. Occasionally, when the pose varies a lot or severely occluded, the Gaussian Kernel Gate opts to compute fine features with the computationally heavy model to inject more accurate features to the LSTM.

depth dimension and makes it easier to acquire the 3D poses of the corresponding hand. However, depth cameras, such as Kinect, are not always available and are prone to measurement errors if deployed in outdoor settings. Therefore, in this work we address the problem of 3D hand pose estimation with a monocular RGB commercial camera.

Recent successes in 3D hand pose estimation [2, 3, 22, 26, 46] mainly focus on employing the same computation framework for all video frames, without considering the redundancy that exists across adjacent frames and the variation of the pose estimation difficulties over frames. The moving speed and occlusion status of the human hands vary when performing different actions, which inspires us to design a new scheme to dynamically allocate the computation resources based on the ambiguity determined by the current input frame and temporal context status. This kind of adaption mechanism is useful for both online and offline applications. For offline pose estimation from videos, being able to use a simpler computation module in most of the frames saves the amount of the resource usage and reduces the total inference time for the entire video. For online pose estimation applications (e.g. HCI and robots), multiple tasks often run concurrently under a total computation resource constraint, thus the saved resources at most of the frames could be released for other important tasks at those time steps, which meanwhile also reduces the amount of energy consumed by the pose estimation task.

Motivated by our idea of reducing computation consumption, and given the fact that the information among video frames could be redundant and the pose estimation difficulty varies over frames, we propose a novel Adaptive Computationally Efficient (ACE) network using a recurrent 3D hand pose estimator with adaptive input. In our method, we design two base pose encoders based on the hourglass(HG) [27] architecture with different computational costs. A Long Short-Term Memory (LSTM) [14] model was introduced to refine the predicted pose and features from the single-frame base pose encoder, by considering the temporal consistency. We propose a new **Gaussian Gate Module** to automatically determine whether the low complexity coarse encoder output alone is sufficient for the LSTM, or the high complexity fine encoder is needed. The fine encoder is only invoked when necessary and its output is combined with the output of the coarse encoder to generate the input for the LSTM. The proposed network architecture is illustrated in Fig. 1. To facilitate the training of our switch module, which is naturally a discrete operation, an effective Gumbel-SoftMax strategy, as an approximation of sampling from discrete distributions, is introduced.

To summarize, a novel end-to-end ACE network is proposed for 3D hand pose estimation from monocular video. It dynamically switches between using coarse v.s. fine features at each time step, which eliminates the computational cost of the fine encoder when the prediction from the coarse encoder is deemed sufficient. We evaluate our network on two broadly used datasets, First-Person Hand Action (FPHA) and Stereo Tracking Benchmark (STB), and obtain state-of-the-art pose estimation accuracy, while greatly reducing the overall computation cost (around 78% on STB dataset), compared to baseline models that constantly use the fine encoder for all time steps.

2 Related Work

Most of the existing works focus on the accuracy of 3D hand pose estimation without explicitly considering the important computation cost issue. We will briefly review the recent works in both the 3D hand pose estimation domain as well as the recent endeavor in designing computationally efficient architectures for image and video understanding.

3D hand pose estimation. 3D hand pose estimation is a long-standing problem in computer vision domain, and various methods have been proposed. We restrict ourselves to the more recent deep learning based approaches since they are more related to our work.

A large body of the works on hand pose estimation operate on the depth input, which greatly reduces the depth ambiguity of the task. Deephand proposes a ConvNet model with an additional matrix completion algorithm to retrieve the actual poses [34]. Volumetric representation was adopted to better encode the depth image recently [7, 8]. The volumetric representation is projected to multiple views and then processed by several 2D ConvNets followed by fusion in [7]. Rather than tedious projections to multiple views, a 3D ConvNet is directly in-

troduced to infer the 3D position from the volumetric representations [8]. This line of work is further summarized in [9], in which the completeness of the 3D hand surface is leveraged as additional supervision. Rather than volumetric representations, the skeleton annotation could be represented as dense pixel-wise labels [37]. The predicted dense estimations are then converted back to 3D coordinates with a vote casting mechanism. Recently, self-supervised methods are also explored on a mixture of synthetic and unlabelled dataset by exploring the approximate depth and the kinematic feasibility as the weak supervision [36].

Rather than performing pose estimation on depth data, we lay more focus on the works with RGB inputs, which are often less restricted in real-world applications. Zimmermann and Brox proposed a multi-stage network, which performs hand segmentation, localization, 2D and 3D pose estimations one by one [46]. Similar to the depth based method, depth regularization was employed to enable weakly supervised learning [2]. Instead of regressing the joint positions independently, kinematic model could be naturally integrated into the model to yield anatomically plausible results [26] . A latent 2.5D representation is introduced in [16], where the ConvNet also learns the implicit depth map of the entire palm. Numerous graphic models are also proposed to better handle the joint relationships [3, 22]. Spatial dependencies and temporal consistencies could be modeled explicitly with graph neural net [3] and could further boost the quality of estimated features [22] from hourglass models [27]. Another line of works reconstruct the shape and the pose of hands at the same time [1, 10, 25, 42, 45], in which either a hand mesh model [25, 33] or a generative GNN [45] is leveraged to map the low-dimensional hand pose & shape manifold to the full 3D meshes.

Despite all the success in accurate hand pose estimation, we argue that the efficiency problem is also of vital importance, especially for AR/VR [28] and mobile devices [11], where resources are often limited. To harvest the redundancy present in the consecutive frames, we propose an adaptive dynamic gate to efficiently switch between an efficient light pose estimator and a computationally heavy pose estimator for 3D hand pose estimation from sequences of frames.

Computationally efficient architectures. Recent progresses have shown that the computation efficiency of neural net models could be improved in various ways. Neural network pruning was first realized using second-order derivative [13, 19] and then evolved into pruning weights with relatively small magnitude [12]. Different from the pruning technique operated on fully trained models [12, 13, 19], recent developments reveal that pruning while training often results in better performance. This was achieved by enforcing additional loss ($L1$ norm [23], Group LASSO [39] or $L0$ norm approximations [24].) during training. Other innovative ideas include specially designed architectures for high-efficiency computing [15, 44] and network quantization [4, 5, 20, 31].

In videos, consecutive frames are often quite similar and strongly co-dependent, which leave lots of space for efficiency optimization. Recently, various works have been developed to improve the computation efficiency for video classification [18,29,38,40]. Leveraging the fact that most of the computational expansive layers (w/o activation) are linear and sparse feature updates are more efficient,

a recurrent residual model was introduced [29] to incur minimum amount of feature updates between consecutive frames. Hierarchical coarse-to-fine architectures are also introduced for more efficient video inference [40]. Recently, RL frameworks are adopted to learn an efficient sampling agent to filter out salient parts/frames from videos for fast recognition [18] [41].

In this work, we address the problem of dense hand pose estimation from video sequences, where we need to derive corresponding poses for each individual frame. We take advantage of the fact that at most of the time, when the motion of the hand is not extreme or the hand pose is not severely occluded, the 3D hand pose could be safely derived from the temporal context. We thus propose a novel Gaussian Kernel-based Adaptive Dynamic Gate module that explicitly measures the necessity to compute fine features with a costly model, which significantly reduces the total amount of computation in general. Our scheme is also orthogonal to many of the aforementioned methods, such as the pruning methods, which leaves the potential to further boost the efficiency.

3 Method

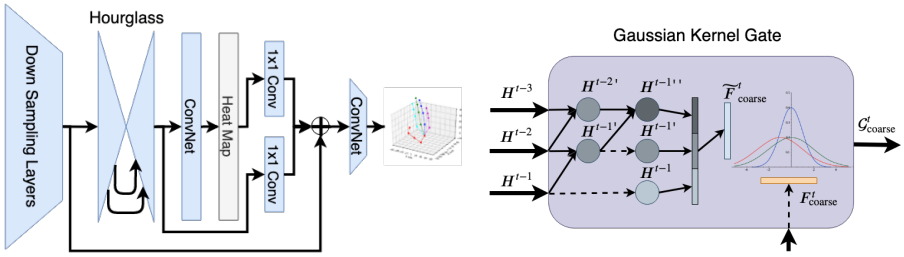
3.1 Overview

Given a sequence of video frames $\{I^t\}_{t=1}^T$, our task is to infer the 3D pose $\mathbf{P}^t = \{P_k^t\}_{k=1}^K$ of the hand at each frame t, where K denotes the number of hand joints, and P_k^t denotes the 3D position of the joint k at frame t.

The overall pipeline of our proposed ACE network is illustrated in Fig. 1. In our method, at each time step, both a less accurate yet computationally light model and an accurate but computationally heavy model can be selected as the pose encoder for the RGB input. The features from either models could be fed into a LSTM to refine the inferred features and the estimated pose based on the temporal coherence. To reduce the computation cost, inspired by the idea that temporal context can provide sufficient information when the motion of the target hand is slow or the pose is less challenging, we propose a novel Gaussian Kernel-based gate module as the key component of our ACE network, which compares the temporal context information provided by the LSTM model with the coarse features computed by the light encoder to assess the necessity of extracting fine features with the heavier encoder for the current time step. Below we introduce each component in more detail.

3.2 Single Frame Hand Pose Estimator

We first introduce two base pose encoders: coarse pose encoder and fine pose encoder, which have significantly different computation profiles for a single frame. Both models are constructed with the state-of-the-art hourglass (HG) network [27]. Furthermore, as illustrated in Fig. 2a, we augment it to directly regress the hand joint coordinates $\mathbf{P}^t$ via a ConvNet from the heat map of joint probabilities $\mathbf{H}^t = \{H_k^t\}_{k=1}^K$, output feature map from HG as well as feature maps from early



(a) Architecture of pose encoder (b) Schema of the Gaussian Kernel Gate.

Fig. 2: Graphical illustration of the Single frame pose encoder and the Gaussian Kernel Based Gate.

downsampling layers. The complexity of the models is adjusted by changing the number of convolutional layers and the size of the inputs of the hourglass module. We denote the light-weight coarse pose encoder model as $\mathbf{M}_{\text{Coarse-Enc}}$, and the heavy model as $\mathbf{M}_{\text{Fine-Enc}}$. These encoders extract pose related features, $\mathbf{F}^t_{\text{coarse}}$ and $\mathbf{F}^t_{\text{fine}}$, based on the input frame I^t, as follows:

$$\mathbf{F}^t_{\text{coarse}} = \mathbf{M}_{\text{Coarse-Enc}}(I^t) \tag{1}$$

$$\mathbf{F}^t_{\text{fine}} = \mathbf{M}_{\text{Fine-Enc}}(I^t) \tag{2}$$

Note that in our final ACE network with the gate mechanism, we compute the coarse features ($\mathbf{F}^t_{\text{coarse}}$) for each frame, while the fine features ($\mathbf{F}^t_{\text{fine}}$) are computed for a fraction of time only, thus reducing the overall computation cost.

3.3 Pose Refinement Recurrent Model

In pose estimation from videos, a natural idea is to exploit the temporal context information for more smooth and accurate estimations, i.e., instead of solely relying on the information of the current frame, historical context can also be incorporated to reduce the ambiguities in pose estimation [21,30,32,35]. Thus we introduce a LSTM model to refine the estimations from the hourglass modules. The LSTM module, denoted as M_{LSTM}, takes the sequential features from pose encoder as inputs, and refine these input features using the temporal context.

More formally, at the t-th time step, the LSTM takes the pose-related features from the current frame as the input, and infer the 3D pose ($\mathbf{P}^t$) for the current step based on the hidden state, as follows:

$$h^t, c^t = \text{M}_{\text{LSTM}}(\mathbf{F}^t_{\text{frame}}, (h^{t-1}, c^{t-1})) \tag{3}$$

$$\mathbf{P}^t = W^\top_{\text{pose}}\, h^t + b_{\text{pose}} \tag{4}$$

where h^t and c^t are the hidden state and cell state of the LSTM module respectively. W_{pose} and b_{pose} are the parameters of the output linear layer for regressing the final 3D hand joint coordinates. Here we denote the features from the single frame pose estimator as $\mathbf{F}^t_{\text{frame}}$, which is controlled by our adaptive dynamic gate model (introduced next) and could be either the coarse features $\mathbf{F}^t_{\text{coarse}}$ or the weighted combination of the coarse features $\mathbf{F}^t_{\text{coarse}}$ and fine features $\mathbf{F}^t_{\text{Fine}}$.

3.4 Adaptive Dynamic Gate Model

Recall that when humans perform activities with their hands, the motion speed and the self-occlusion status of the hands vary across different activities and different frames. In some of the actions like "high five", the palm is often less occluded and the pose pattern is relatively static and simple, while in some other actions, like "open soda can" and "handshake", the human hand is often under severe occlusions and presents rich and delicate movements of the fingers.

This inspires us to rely more on the temporal context information (and only use a brief glimpse over the current frame with the coarse pose encoder) for pose inference when the pose pattern is simple, stable and could be safely derived from the temporal context. However, if the temporal context is not consistent with the current frame information, this means either the current frame could be challenging for pose inference (i.e. pose inaccurately estimated by coarse pose encoder but temporal context is reliable) or significantly differs from previous frames due to large motions (i.e. temporal context becomes unstable), and thus the network needs to take a more careful examination for the current frame by using the fine pose encoder. Therefore, we propose an adaptive dynamic gate model in our ACE framework to dynamically determine the granularity of the features needed for pose estimation with our LSTM model.

Assuming the motion of the hand is smooth, the first and second-order statistics of the hand's status over different frames provide useful context information for estimating the evolution of the hand pose over time. Accordingly, we compute the first-order difference ($h^{t'}$) and second-order difference ($h^{t''}$) over the history hidden states of the LSTM to estimate the motion status information of the hand pose as:

$$h^{t'} = h^t - h^{t-1} \tag{5}$$

$$h^{t''} = (h^t - h^{t-1}) - (h^{t-1} - h^{t-2}) \tag{6}$$

At the time step t, we feed the hidden state of the previous frame(h^{t-1}), as well as its first and second-order information ($h^{t-1'}$ and $h^{t-1''}$) as the history context information, to our gate module, which then estimates the pose feature information of current frame (t) with a sub-network, as follows:

$$\widetilde{\mathbf{F}}^t = W_g^\top [h^{t-1}, h^{t-1'}, h^{t-1''}] + b_g \tag{7}$$

We then measure the similarity of the predicted pose feature information ($\widetilde{\mathbf{F}}^t$) that is completely estimated from the temporal context of previous frames, with the pose features ($\mathbf{F}^t_{\text{coarse}}$) that are extracted with the coarse pose encoder solely based on current frame I^t, via a Gaussian Kernel with a fixed spread ω as follows:

$$\mathcal{G}^t_{\text{coarse}} = \left[\exp\left(-\frac{(\widetilde{\mathbf{F}}^t - \mathbf{F}^t_{\text{coarse}})^2}{\omega^2} \right) \right]_{\text{Mean}} \tag{8}$$

This Gaussian Kernel based gate outputs mean value ($\mathcal{G}^t_{coarse}$) between 0 and 1, which provides an explicit measurement of the consistency and similarity between $\widetilde{\mathbf{F}}^t$ and $\mathbf{F}^t_{\text{coarse}}$, implying the pose estimation difficulty of current frame, i.e., higher $\mathcal{G}^t_{coarse}$ value indicates simple pose and stable movements of the hand.

If the hand pose status at this step changes a lot and pose feature becomes unpredictable from the temporal context, or the pose at current frame becomes challenging, leading to the pose features ($\mathbf{F}^t_{\text{coarse}}$) extracted by the coarse pose encoder not reliable and therefore inconsistent with the temporal context, the discrepancy between $\widetilde{\mathbf{F}}^t$ and $\mathbf{F}^t_{\text{coarse}}$ grows larger, and thus our Gaussian gate will output a relatively small value close to 0.

With an estimation of the difficulty of current frame, we then decide if we need to employ the more powerful fine pose encoder to carefully examine the input frame of current time step. Specifically, we can use the $\mathcal{G}^t_{\text{coarse}}$ from our Gaussian gate as the confidence score of staying with the coarse pose encoder for current time step, and naturally $\mathcal{G}^t_{\text{fine}} = 1 - \mathcal{G}^t_{\text{coarse}}$ becomes the score that we need to use the more powerful fine pose encoder.

A straight-forward switching mechanism would be to directly follow the one with a larger confidence score, i.e., if $\mathcal{G}^t_{\text{fine}} > \mathcal{G}^t_{\text{coarse}}$, we need to involve the fine pose encoder for the current frame. This switching operation is however a discrete operation that is not differentiable. To facilitate the network training, following the recent work on reparameterization for the categorical distribution [17], we reparameterize the Bernoulli distribution with the Gumbel-Softmax trick, which introduces a simple yet efficient way to draw samples z from a categorical distribution parameterized by the unnormalized probability π. Specifically, we can approximately sample from π_i following:

$$z_i = \underset{i \in \mathcal{M}}{\operatorname{argmax}} \left[g_i + \log \pi_i \right] \qquad \mathcal{M} = \{\text{coarse}, \text{fine}\} \tag{9}$$

where at each time step t, we set $\pi^t_i = -\log(1 - \mathcal{G}^t_i)$, which is the unnormalized version of the predict probability $\mathcal{G}^t_i$ in Bernoulli distribution $\mathcal{G}^t_i \in \{\mathcal{G}^t_{\text{coarse}}, \mathcal{G}^t_{\text{fine}}\}$. g_i is the Gumbel noise. Here we draw samples from the Gumbel distribution following $g_i = -\log(-\log(u_i))$, where u_i is the i.i.d. samples drawn from Uniform(0,1). We further relax the non-differentiable operation argmax with softmax to facilitate back propagation. The final sampled probability is obtained with:

$$z^t_i = \frac{\exp\Big((g_i + \log \pi^t_i)/\tau\Big)}{\sum_j \exp\Big((g_j + \log \pi^t_j)/\tau\Big)} \qquad \text{for } i, j \in \{\text{coarse}, \text{fine}\} \tag{10}$$

where τ is the hyper-parameter of temperature, which controls the discreteness of the sampling mechanism. When $\tau \to \infty$, the sample approximates the uniform sampling, and when $\tau \to 0$, it yields the argmax operation while allows the gradient to be back-propagated.

During training, we obtain the confidence scores of using rough glimpse for the input frame via the coarse pose encoder or using careful derived features

with the fine encoder, via the Gumbel-SoftMax trick following Eq. (10), and then we combine the coarse features $\mathbf{F}^t_{\text{coarse}}$ and fine features $\mathbf{F}^t_{\text{fine}}$ as:

$$\mathbf{F}^t_{\text{weighted}} = z^t_{\text{coarse}}\mathbf{F}^t_{\text{coarse}} + z^t_{\text{fine}}\mathbf{F}^t_{\text{fine}} \tag{11}$$

During evaluation, we omit the sampling process and directly use the coarse features when $\mathcal{G}^t_{\text{fine}} \leq \lambda$, and use the weighted average features when $\mathcal{G}^t_{\text{fine}} > \lambda$ with weight $\mathcal{G}^t_{\text{fine}}$ and $\mathcal{G}^t_{\text{coarse}}$. In general, λ is set to 0.5, which essentially follows the larger probability. This threshold λ could also be tweaked to balance between accuracy and efficiency during inference.

3.5 Training Strategy and Losses

We employ a two-step training strategy, in which we separately train the single-frame coarse pose encoder and fine pose encoder first, and then fine-tune them during the training of the LSTM pose refinement module and the adaptive gate module. To train the single frame pose encoder, we use the combination of 2D heat map regression loss and 3D coordinate regression loss:

$$\mathcal{L}_{\text{single}} = \frac{1}{K}\sum_{k=1}^{K}(\widetilde{H}^k - H^k)^2 + \beta \cdot \text{Smooth L1}(\widetilde{\mathbf{P}}, \mathbf{P}) \tag{12}$$

where H^k corresponds to the 2D heat map of joint k and $\mathbf{P}$ is the 3D joint coordinates. We use the mean squared loss for the heat map and Smooth L1 loss for the 3D coordinates, which has a squared term when the absolute element-wise difference is below 1 (otherwise it is essentially a L1 term).

The single frame pose estimator is then fine-tuned when training the pose refinement LSTM and the gate module. To prevent the gate module from constantly using the fine features, we set an expected activation frequency (γ_g) for the gate, and optimize the mean square error between mean probability of using fine encoder and the expected activation frequency. Specifically, we define the loss mathematically given the expected activate rate, γ_g as:

$$\mathcal{L}_{\text{whole}} = \sum_{d\in\mathcal{S}} \text{Smooth L1}(\widetilde{\mathbf{P}}_{\text{d}}, \mathbf{P}_{\text{d}}) + \delta \cdot \mathbb{E}_{z^t\sim\text{Bernoulli}(\mathcal{G}^t|\theta_g)}(\frac{1}{T}\sum_{t=1}^{T} z^t_{\text{fine}} - \gamma_g)^2 \tag{13}$$

where $\mathcal{S} = \{\text{coarse}, \text{fine}, \text{LSTM}\}$ and z^t_{fine} is the sample probability based on the prediction $\mathcal{G}^t$ given by the adaptive dynamic gate model. θ_g denotes the parameter of the gate and δ balances the accuracy and the efficiency.

4 Experiments

4.1 Datasets and Metrics

We evaluate our ACE network on two publicly available datasets, namely the Stereo Tracking Benchmark (STB) [43] dataset and the First-Person Hand Action (FPHA) [6] dataset.

Stereo Tracking Benchmark (STB) provides 2D and 3D pose annotations of the 21 hand keypoints for 12 stereo video sequences. Each sequence consists of 1500 RGB frames for both the left camera and right camera. In total, this dataset consists of 18000 frames and the resolution of the frame is 640×480. Within the dataset, 6 set of different backgrounds are captured, with each background appears in two video sequences. Following the setting of [46], we separate the dataset into a training set of 10 videos (15000 frames) and a evaluation set of 2 video sequences (3000 frames).

First Person Hand Action (FPHA) contains video sequences for 45 different daily actions from 6 different subjects in egocentric views. In total, FPHA contains more than 100k frames with a resolution of 1920×1080. The ground truth is provided via a mo-cap system and derived with inverse kinematics. Similar to the STB dataset, 21 keypoints on the human hand are annotated. Object interaction with 26 different objects is involved, which introduces additional challenges to hand pose estimation. We follow the official split of the dataset.

Metrics. We report the Percentage of Correct Keypoints (PCKs) under 20 mm and the Area Under Curve (AUC) of PCKs under the error thresholds from 20 mm to 50 mm for STB dataset following [46], and from 0 mm to 50 mm for FPHA dataset. We report average GFLOPs [1] per frame for speed comparison, which does not rely on the hardware configurations and thus provides more objective evaluations.

4.2 Implementation Details

Although the proposed ACE module is theoretically compatible with different pose encoder architectures, we mainly evaluate it with the hourglass (HG) architecture [27] as it is widely used and works well in many existing works [22, 45]. Compared to the FPHA dataset, STB is less challenging as no hand-object interaction is involved. Therefore, different HG architectures are employed for different datasets. For the STB dataset, the coarse pose encoder contains one hourglass module with 32 feature channels, while for the fine pose encoder, we employ 64 channels. In addition to the different configurations of the module, the input images to the coarse and fine modules are set to 64×64 and 256×256 respectively, which greatly reduce the amount of computation. For the more challenging FPHA dataset, we keep the configurations of the fine pose encoder as STB, while for the coarse pose encoder, we double the size of input to 128×128. Please see the supplementary materials for more details of the pose encoder.

For the LSTM refinement module, we use one layer of LSTM with hidden state dimension of 256. The hidden states and its order statistics are first mapped to a fixed dimension of 256 and then concatenated as the input to our adaptive Gaussian gate. During training, we set $\gamma_g = 0.05$ for STB and $\gamma_g = 0.01$ for FPHA and $\omega = 0.1$.

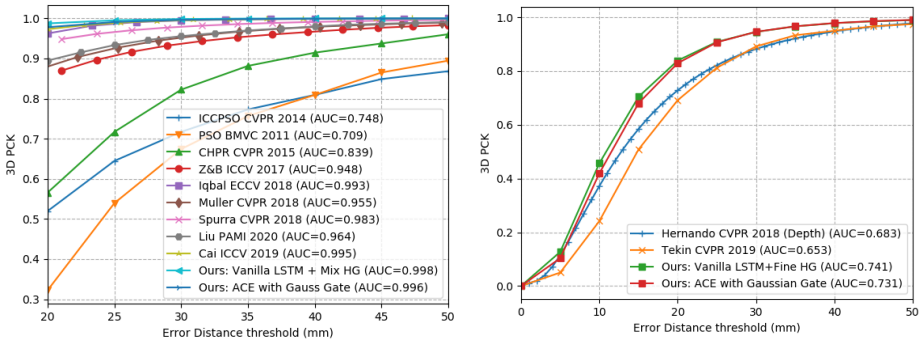
[1] Computed based on the public toolbox: PyTorch-OpCounter.

Table 1: Results of various models (vanilla single frame coarse/fine models and their variants considering temporal dynamics) for 3D hand pose estimation. Our adaptive model uses much less computation with minor accuracy drops.

	STB			FPHA		
Method	3D PCK20	AUC(20-50)	GFLOPs	3D PCK20	AUC(0-50)	GFLOPs
Coarse-HG	85.1%	0.946	0.28	72.6%	0.674	1.10
Fine-HG	96.3%	0.994	6.96	79.7%	0.714	6.96
Vanilla-LSTM-Coarse-HG	92.1%	0.973	0.28	78.9%	0.707	1.10
Vanilla-LSTM-Fine-HG	98.7%	0.997	6.96	83.9%	0.740	6.96
Vanilla-LSTM-Mix-HG	98.7%	0.997	7.24	83.1%	0.734	8.06
Adaptive-LSTM-Mix-HG	97.9%	0.996	1.56	82.9%	0.731	1.37

4.3 Main Results

We conduct extensive experiments to show the advantages of our proposed ACE framework for hand pose estimation from videos. We compare the accuracy and computation efficiency among different models and further visualize the prediction results of our model. To facilitate the understanding of the gate behaviour, we also present the frames selected for fine feature computation.



(a) STB dataset (b) FPHA dataset

Fig. 3: Quantiatitative evaluations. We achieve state-of-the-art performance on STB, and outperform the existing methods on FPHA by a large margin.

Quantitative comparison. We present the comparison among our adaptive dynamic gate model and various baselines in Table 1, where Coarse-HG/fine-HG indicates that the baseline pose encoder (hourglass structure) is employed to predict 3D joint coordinates frame by frame. For the Vanilla-LSTM variants, we take features from either coarse pose encoder, fine pose encoder, or average features from coarse and fine pose encoders, and then feed them into an ordinary LSTM model without gate module. The detailed results are in Table. 1.

Table 2: Comparison of the computation cost with state-of-the arts on STB. Our method achieves higher AUC yet consumes significantly less computation.

Method	3D PCK20	AUC	GFLOPs
Z&B [46]	0.870	0.948	78.2
Liu et al. [22]	0.895	0.964	16.0
HAMR [45]	0.982	0.995	8.0
Cai et al. [3]	0.973	0.995	6.2
Ours	0.979	0.996	1.6

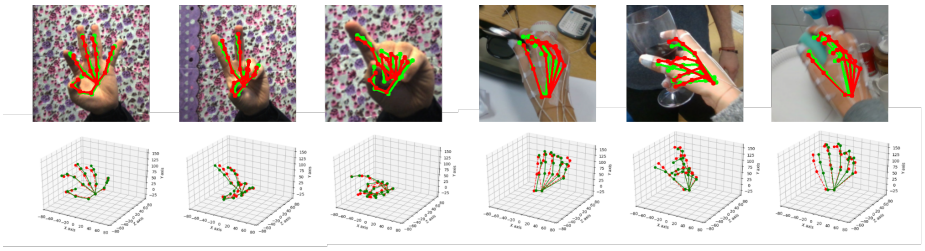


Fig. 4: Visualization of pose estimation. The top row shows input frames and the bottom row visualizes the predicted poses (red) and ground-truth poses (green).

As shown in Table 1, our adaptive model obtains comparable performance to our designed baseline model "Vanilla-LSTM-Fine-HG" that constantly takes the fine features for pose estimation with less than 1/4 computation cost by computing the fine features only on selected frames. Besides, our proposed method obtains state-of-the-art performance on both benchmarks, which is presented in Fig. 3a and 3b, where we plot the area under the curve (AUC) on the percentage of the correct key points (PCK) with various thresholds.

In addition to the comparison in terms of the accuracy, we further evaluate the speed of our model compared to the existing art. The detailed comparison are illustrated in Table 2. As FPHA dataset is relatively new and fewer works report their performance, we mainly conduct the evaluation on the STB dataset.

Visualization To verify our model works well in terms of accurately deriving poses from the RGB images. We visualize a few predictions by our network in Fig. 4. Our model is capable of inferring precise poses from RGB input images even under severe occlusion and challenging lightning conditions.

We further look into the mechanism of the Gaussian kernel based gate. We visualize a few test sequences as in Fig. 5. In (a), the fine pose encoder activates less often for the straightforward poses while more densely used for the challenging poses close to the end of the sequence. For (b), the gate tends to invoke fine pose encoder more often when occlusion presents (1st half v.s. 2nd half), while in (c) and (d), when large motion presents (see the rightmost blurry frames from both sequences), the gate chooses to examine the frame more closely with the fine pose encoder. Those observations are in par with our motivations that to only invoke the computationally heavy pose encoders when necessary.

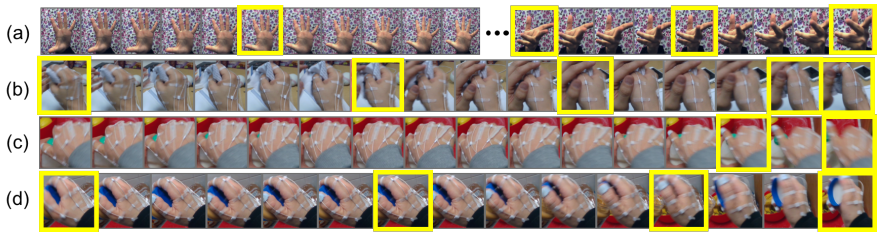


Fig. 5: Visualization of frames selected (marked with yellow boxes) to adopt the fine pose encoder. The fine encoder activates sparsely when the pose is straightforward while is frequently used when the pose becomes challenging (left part v.s. right part of (a)). When the hand pose status becomes less stable (see rightmost part of (c) and (d)) or occlusions become more severe (see rightmost part of (b)), our model tends to use the fine encoder more frequently. The frequency of invoking fine pose encoder is much lower when the poses are relatively stable.

Table 3: Evaluation of using different gate architectures on STB dataset. P_{fine} denotes the frequency of using the fine pose encoder. Our Gaussian kernel gate achieves highest accuracy yet at lowest computation cost.

Gate	γ_g	3D PCK20 AUC	GFLOPs	P_{fine}
Neural Gate	0.1	0.981	0.995	2.54	0.32
Neural Temporal Gate	0.1	0.977	0.996	2.20	0.43
Gaussian Kernel Gate	0.1	0.983	0.997	2.09	0.26

4.4 Ablation study

We first study on the design choice of the Gaussian kernel based adaptive gate. Instead of explicitly parameterize the difference with Gaussian function, one straight forward way would be to directly predict the probability via a linear module. The linear module takes the hidden state, 1st and 2nd order statistics and coarse feature as the input and yields the probability of introducing the fine module. This model is referred as **Neural Gate**. Going one step further, although the coarse pose encoder is relatively light, we could still obtain performance gains by avoiding it and derive probability solely based on the temporal context. Therefore, we also evaluate the model that make decisions based on the temporal context only, which is referred as **Neural Temporal Gate.** The detailed results are in Table. 3.

As shown in Table. 3, different gates offer similar performance while the Gaussian Kernel is slightly more accurate and more efficient. We further investigate the impact of a few hyper parameters on the overall performance. Specifically, we look into the γ_g in Table 4 and λ in Table 5, which could be tweaked to adjust the rate of computing fine features before and after training.

When varying γ_g from 0.3 to 0.01, the accuracy of the models does not vary much while the frequency of using fine features drops from 0.43 to 0.15, which suggests the large amount redundancy in consecutive frames are exploited by

Table 4: Evaluation of different γ_g values for network training on STB dataset. As the expected usage of fine pose encoder drops, the computation cost falls significantly, while the accuracy decreases marginally

γ_g	3D PCK20	AUC	GFLOPs	P_{fine}
1	0.987	0.9978	6.96	1
0.3	0.984	0.9972	3.30	0.43
0.2	0.985	0.9973	2.34	0.29
0.1	0.983	0.9970	2.09	0.26
0.05	0.979	0.9962	1.56	0.18
0.01	0.977	0.9956	1.37	0.15
0.001	0.955	0.9897	1.43	0.16

Table 5: Evaluation of different λ ($\gamma_g = 0.1$) during testing on the STB dataset. For the same trained model, with higher λ, fine encoder is used less often, i.e., we can configure λ to balance the trade-off between the efficiency and accuracy.

λ	3D PCK20	AUC	GFLOPs	P_{fine}
0.1	0.987	0.9977	7.01	0.97
0.3	0.986	0.9976	3.31	0.43
0.5	0.983	0.9970	2.09	0.26
0.7	0.943	0.9894	0.88	0.08
0.9	0.505	0.8277	0.30	0

the ACE model. While for λ, with a larger threshold, we greatly reduce the frequency of using fine encoders at the cost of accuracy. λ could be adjusted during inference to balance the trade off between efficiency and accuracy.

5 Conclusion

We present the ACE framework, an adaptive dynamic model for efficient hand pose estimation from monocular videos. At the core of the ACE model is the Gaussian kernel based gate, which determines whether to carefully examine the current frame using a computationally heavy pose encoder based on a quick glimpse of the current frame with a light pose encoder and the temporal context. We further introduce the Gumbel-SoftMax trick to enable the learning of the discrete decision gate. As a result, we obtain state of the art performance on 2 widely used datasets, STB and FPHA, while with less than 1/4 of the computation compared to the baseline models. The proposed ACE model is general and could be built upon any single frame pose encoder, which indicates the efficiency could be further improved by harvesting more efficient structures as single frame pose encoder.

Acknowledgements This work is partially supported by the National Institutes of Health under Grant R01CA214085 as well as SUTD Projects PIE-SGP-Al-2020-02 and SRG-ISTD-2020-153.

References

1. Boukhayma, A., Bem, R.d., Torr, P.H.: 3d hand shape and pose from images in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10843–10852 (2019)
2. Cai, Y., Ge, L., Cai, J., Yuan, J.: Weakly-supervised 3d hand pose estimation from monocular rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 666–682 (2018)
3. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.J., Yuan, J., Thalmann, N.M.: Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2272–2281 (2019)
4. Courbariaux, M., Bengio, Y., David, J.P.: Binaryconnect: Training deep neural networks with binary weights during propagations. In: Advances in neural information processing systems. pp. 3123–3131 (2015)
5. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830 (2016)
6. 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: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 409–419 (2018)
7. 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3593–3601 (2016)
8. Ge, L., Liang, H., Yuan, J., Thalmann, D.: 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1991–2000 (2017)
9. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Real-time 3d hand pose estimation with 3d convolutional neural networks. IEEE transactions on pattern analysis and machine intelligence **41**(4), 956–970 (2018)
10. Ge, L., Ren, Z., Li, Y., Xue, Z., Wang, Y., Cai, J., Yuan, J.: 3d hand shape and pose estimation from a single rgb image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 10833–10842 (2019)
11. Gouidis, F., Panteleris, P., Oikonomidis, I., Argyros, A.: Accurate hand keypoint localization on mobile devices. In: 2019 16th International Conference on Machine Vision Applications (MVA). pp. 1–6. IEEE (2019)
12. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in neural information processing systems. pp. 1135–1143 (2015)
13. Hassibi, B., Stork, D.G.: Second order derivatives for network pruning: Optimal brain surgeon. In: Advances in neural information processing systems. pp. 164–171 (1993)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

16. Iqbal, U., Molchanov, P., Breuel Juergen Gall, T., Kautz, J.: Hand pose estimation via latent 2.5 d heatmap regression. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 118–134 (2018)
17. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
18. Korbar, B., Tran, D., Torresani, L.: Scsampler: Sampling salient clips from video for efficient action recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6232–6242 (2019)
19. LeCun, Y., Denker, J.S., Solla, S.A.: Optimal brain damage. In: Advances in neural information processing systems. pp. 598–605 (1990)
20. Li, Z., Ni, B., Zhang, W., Yang, X., Gao, W.: Performance guaranteed network acceleration via high-order residual quantization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2584–2592 (2017)
21. Lin, M., Lin, L., Liang, X., Wang, K., Cheng, H.: Recurrent 3d pose sequence machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 810–819 (2017)
22. Liu, J., Ding, H., Shahroudy, A., Duan, L.Y., Jiang, X., Wang, G., Kot, A.C.: Feature boosting network for 3d pose estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 494–501 (2020)
23. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2736–2744 (2017)
24. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through l_0 regularization. arXiv preprint arXiv:1712.01312 (2017)
25. Malik, J., Elhayek, A., Nunnari, F., Varanasi, K., Tamaddon, K., Heloir, A., Stricker, D.: Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In: 2018 International Conference on 3D Vision (3DV). pp. 110–119. IEEE (2018)
26. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: Ganerated hands for real-time 3d hand tracking from monocular rgb. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 49–59 (2018)
27. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
28. Oculus: Hand tracking sdk for oculus quest available with v12 release, https://developer.oculus.com/blog/hand-tracking-sdk-for-oculus-quest-available
29. Pan, B., Lin, W., Fang, X., Huang, C., Zhou, B., Lu, C.: Recurrent residual module for fast inference in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1536–1545 (2018)
30. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7753–7762 (2019)
31. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016)
32. Rayat Imtiaz Hossain, M., Little, J.J.: Exploiting temporal information for 3d human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 68–84 (2018)

33. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics (ToG) **36**(6), 245 (2017)
34. Sinha, A., Choi, C., Ramani, K.: Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4150–4158 (2016)
35. Tekin, B., Rozantsev, A., Lepetit, V., Fua, P.: Direct prediction of 3d body poses from motion compensated sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 991–1000 (2016)
36. Wan, C., Probst, T., Gool, L.V., Yao, A.: Self-supervised 3d hand pose estimation through training by fitting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10853–10862 (2019)
37. Wan, C., Probst, T., Van Gool, L., Yao, A.: Dense 3d regression for hand pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5147–5156 (2018)
38. Wang, F., Wang, G., Huang, Y., Chu, H.: Sast: Learning semantic action-aware spatial-temporal features for efficient action recognition. IEEE Access **7**, 164876–164886 (2019)
39. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in neural information processing systems. pp. 2074–2082 (2016)
40. Wu, Z., Xiong, C., Jiang, Y.G., Davis, L.S.: Liteeval: A coarse-to-fine framework for resource efficient video recognition. In: Advances in Neural Information Processing Systems. pp. 7778–7787 (2019)
41. Wu, Z., Xiong, C., Ma, C.Y., Socher, R., Davis, L.S.: Adaframe: Adaptive frame selection for fast video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1278–1287 (2019)
42. Xiang, D., Joo, H., Sheikh, Y.: Monocular total capture: Posing face, body, and hands in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10965–10974 (2019)
43. Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., Yang, Q.: 3d hand pose tracking and estimation using stereo matching. arXiv preprint arXiv:1610.07214 (2016)
44. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
45. Zhang, X., Li, Q., Mo, H., Zhang, W., Zheng, W.: End-to-end hand mesh recovery from a monocular rgb image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2354–2364 (2019)
46. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4903–4911 (2017)