SDPose: Tokenized Pose Estimation via Circulation-Guide Self-Distillation

Sichen Chen[1*] Yingyi Zhang[2*] Siming Huang[2*] Ran Yi[1] Ke Fan[1]
Ruixin Zhang[2] Peixian Chen[2] Jun Wang[3] Shouhong Ding[2†] Lizhuang Ma[1,4†]

[1]Shanghai Jiao Tong University, [2]Tencent Youtu Lab, [3]Tencent WeChat Pay Lab33
[4]MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University

Abstract

Recently, transformer-based methods have achieved state-of-the-art prediction quality on human pose estimation(HPE). Nonetheless, most of these top-performing transformer-based models are too computation-consuming and storage-demanding to deploy on edge computing platforms. Those transformer-based models that require fewer resources are prone to under-fitting due to their smaller scale and thus perform notably worse than their larger counterparts. Given this conundrum, we introduce SDPose, a new self-distillation method for improving the performance of small transformer-based models. To mitigate the problem of under-fitting, we design a transformer module named Multi-Cycled Transformer(MCT) based on multiple-cycled forwards to more fully exploit the potential of small model parameters. Further, in order to prevent the additional inference compute-consuming brought by MCT, we introduce a self-distillation scheme, extracting the knowledge from the MCT module to a naive forward model. Specifically, on the MSCOCO validation dataset, SDPose-T obtains 69.7% mAP with 4.4M parameters and 1.8 GFLOPs. Furthermore, SDPose-S-V2 obtains 73.5% mAP on the MSCOCO validation dataset with 6.2M parameters and 4.7 GFLOPs, achieving a new state-of-the-art among predominant tiny neural network methods.

1. Introduction

Human Pose Estimation (HPE) aims to estimate the position of each joint point of the human body in a given image. HPE tasks support a wide range of downstream tasks such as activity recognition[1], motion capture[2], etc. Recently with the ViT model being proven effective on many visual tasks, many transformer-based methods[3–5] have achieved excellent performance on HPE tasks. Compared with past CNN-based methods[6], transformer-based models are much more powerful in capturing the relationship between visual elements. However, most of them are large and computationally expensive. The state-of-the-art(SOTA) transformer-based model[5] has 632 million parameters and requires 122.9 billion floating-point operations. Such a large-scale model is difficult to deploy on edge computing devices and cannot accommodate the growing development of embodied intelligence. However, when the CNN or ViT used as a backbone is not of sufficient scale, transformer-based models are not able to learn the relationship between keypoints and visual elements well resulting in poor performance. Stacking more transformer layers is a viable approach[4], but this also increases the scale of the network resulting in larger parameters and the difficulty of edge deployment.

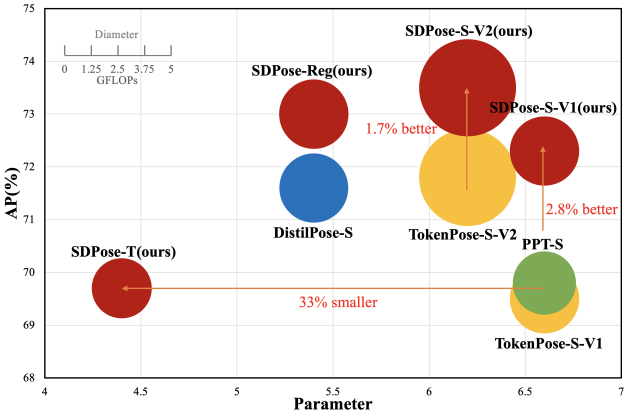

Figure 1. Comparsions between other small models and our methods on MSCOCO validation dataset. Compared to other methods, our approach can significantly reduce the scale while maintaining the same performance, or greatly improve performance under the same scale.

To help small models learn better, one possible way is to distill knowledge from big model to small model[7, 8]. However, previous distillation methods have the following drawbacks: (1) To align the vector space, an additional manipulation is required during feature distillation[9] and leads to a potential performance decrease. (2) A huge extra

*Equal Contribution.
†Corresponding Authors.

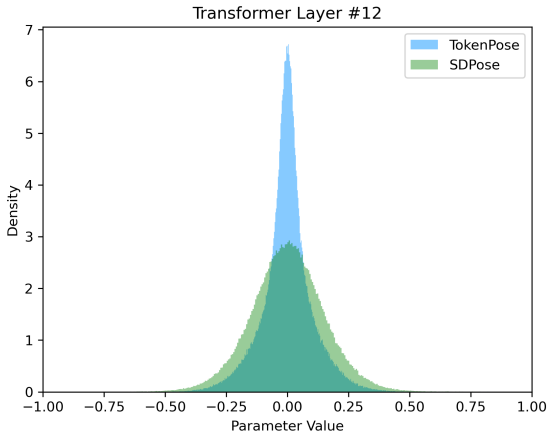

Figure 2. Visualization of parameter distributions for transformer layer # 12. The blue represents TokenPose-S-V1[3] and the green represents SDPose-S-V1. There are fewer parameters close to 0 in our method, which proves that the parameters are more fully learned.

training cost is required to train the powerful teacher network.

In this paper, we introduce a cyclic forwarding scheme, for which we further design a self-distillation method. This framework, termed SDPose, mitigated the conflicts between scale and performance for HPE. The key insight guiding our designs is that for a deep HPE method, its performance can improve proportionally to what we define as the model's *latent depth*. *Latent depth* is the transformer layers depth involved in the complete inference process. Adding layers to the model is the straightforward way to increase *latent depth*, but it also incurs extra parameters. To increase the *latent depth* without adding extra parameters, we design the Multi-Cycled Transformer(MCT) module, which passes the tokenized features through the transformer layers in multiple cycles during inference and uses the last output as the result. As shown in Fig. 2, compared to the transformer-based models, the parameters of the MCT-based models have higher variance and lower density near zero, which proves that it has been better trained. In this way, utilizing the MCT module can help small transformer-based models to be considered as a transformer-based model with greater *latent depth*, and break free from under-fitting to achieve a better performance.

Nevertheless, the MCT module still adds extra computational effort. In order to avoid additional computational consumption, we come up with a quite simple but effective self-distillation paradigm. Specifically, during the training phase, we send the tokenized features into the MCT module, and because the input and output are in the same vector space for each cycle in the MCT module, previous results can be distilled from the latter outputs without any additional operations. At inference time, we perform one single pass to maintain the original computation consumption. With this design, we extract the knowledge of the MCT module into a naive forward model in one training, resulting in a better-trained model. Overall, our method achieved improved performance while maintaining the computational consumption.

We designed several SDPose models based on TokenPose[3] and DistilPose[8]: SDPose-T, SDPose-S-V1, SDPose-S-V2, SDPose-B and SDPose-Reg. As evident in Fig. 1, our MCT-based models achieved improved performance under the same compute consumption as their base models. They also achieved similar performance compared to other much larger models.

Our contribution can be summarized as follows:

- We are the first to find that looping the token through the transformer layers can increase the *latent depth* of the transformer layers without adding extra parameters. Based on this finding, we design Multi-Cycled Transformer(MCT) module.
- We design a self-distillation paradigm SDPose that extracts the knowledge in the MCT module into one single pass model, achieving a balance between performance and resource consumption. To the best of our knowledge, we are the first to explore how self-distillation can be applied to the transformer-based HPE task.
- We have conducted extensive experiments and analyses to demonstrate the effectiveness and broad applicability of our approach on multiple tasks.

2. Related Works

2.1. Human Pose Estimation

Deep learning based methods dominate the HPE tasks. Deep learning based human pose estimators can be classified into regression-based and heatmap-based.

Regression-based methods directly estimate the coordinates for each keypoint. Toshev *et al.* [10] first leveraged a convolutional network to predict the image coordinates of 2D human joints, followed by numerous innovations in network architecture. Notably, transformer-based networks such as Poseur [11] have achieved good prediction quality. Besides seeking better network architectures, works like RLE [12] improved the regressor learning framework by quantifying the uncertainty of the regressed coordinates.

Heatmap-based methods estimate a 2D image or 3D volume of likelihood and decode it into coordinates. Since the seminal work by Tompson *et al.*[13], heatmap has become the predominant output representation, as its dimensionality better aligns with the input image space and thus reduces the learning complexity for neural networks.

For both the output representations, transformer has proven to be an effective module for feature extraction and processing for human pose estimators. Yang *et al.* [4] uti-

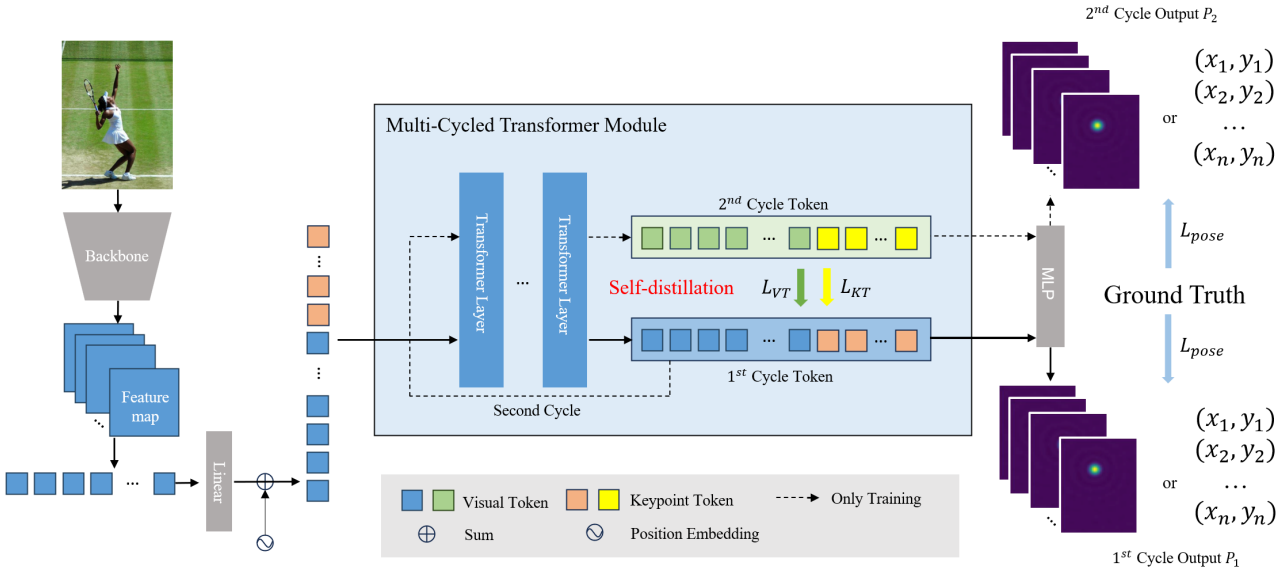

Figure 3. Overall architecture of SDPose used twice cycles. During training, The visual tokens and keypoint tokens will pass through the transformer layers twice. The tokens and heatmaps obtained during the second time will serve as the teacher to distill the tokens and heatmaps obtained during the first time.

lizes transformer encoder to further encode the feature map produced by a convolutional neural network into keypoint representations. Xu *et al.* [5], on the other hand, designs a pure-transformer architecture for initial image feature extraction as well as feature processing. Li *et al.* [3] designs token representation of keypoint information and feeds the learnable keypoint tokens as input to the transformer modules. While accurate, these transformer-based models tend to be complex.

Several works have proposed more lightweight transformer-based HPE models. Ye *et al.* [8] designed DistilPose with a novel simulated heatmap loss to enable knowledge transfer from a heatmap-based teacher network to a regression-based student network. While it achieved SOTA performance, DistilPose requires training a teacher network for the separate distillation process. In comparison, Ma *et al.* proposed PPT [14] that finds and discards the less attended image token in TokenPose to reduce computational complexity. Although PPT reduced computation without extra hassles, it comes with the cost of performance drops compared to TokenPose. Previous works failed to reduce computation complexity while achieving SOTA results through an integral process, and our work is to our knowledge the first success towards this goal for HPE.

2.2. Knowledge Distillation

To reduce the cost of training and deploying deep learning models, several techniques have been proposed, among which knowledge distillation is the most relevant to our method.

Originally proposed by Hinton et.al [15] as a model compression technique, knowledge distillation transfers knowledge from a teacher model to a student model. Recent works explored knowledge distillation within one model, namely self-distillation. Be Your Own Teacher [16] distill knowledge in deeper layers into shallower layers within one model. Born-Again Neural Network [17] applies self-distillation along the temporal dimension, distilling knowledge from the model in previous iterations to supervise model learning in the current iteration. Our work furthered this line of work by making the first effort to apply self-distillation on transformer-based HPE models.

3. Methods

In this section, we propose the Multi-Cycled transformer(MCT) module for our cyclic forwarding scheme. Further, we propose a self-distillation human pose estimation framework SDPose for our MCT module. During training, the model passes tokens through the MCT module for several cycles, where the output from the previous cycle is used as the input to the next cycle. Then, we use the output of each cycle in the MCT module to distill the output of the previous cycle, thus extracting the knowledge from the complete inference of the MCT module into one single pass. During inference, the model maintains its original inference pipeline, without incurring additional computation but achieving stronger performance. The overall framework is shown in Fig. 3.

3.1. Multi-Cycled Transformer Module

To better improve small transformer-based models, we first investigated how to go about increasing the *latent depth*

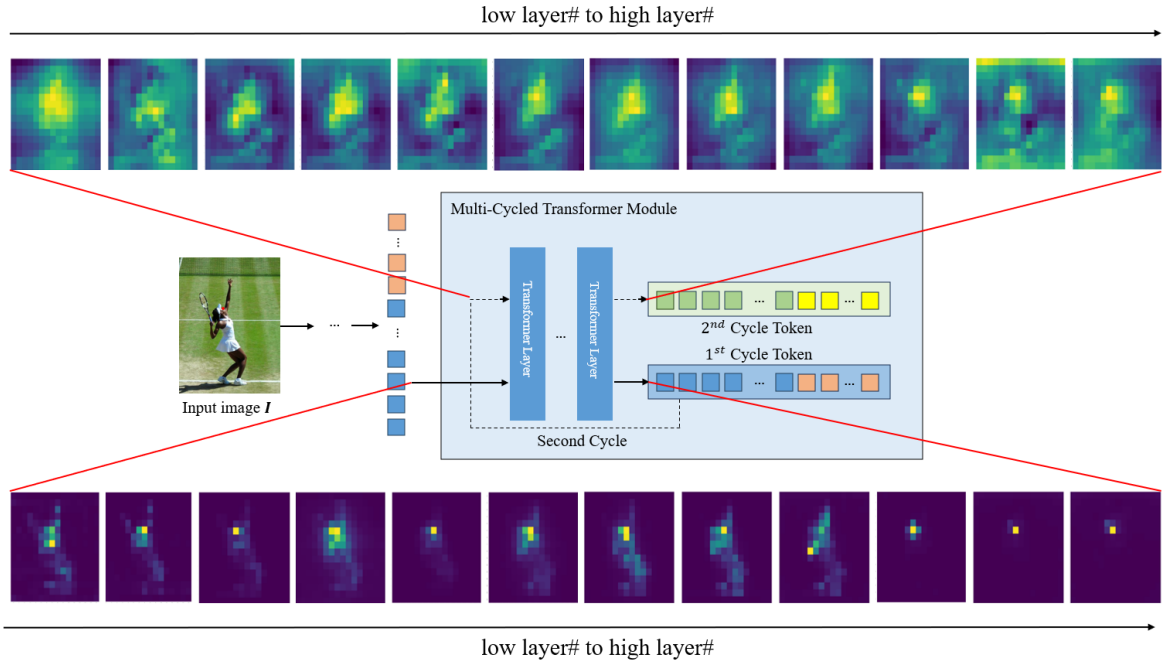

Figure 4. Visualization of the attention maps between nose keypoint token and visual tokens in different layers of MCT module. The lower one is from the first cycle, The top one is from the second cycle.

of small transformer-based models. We propose the MCT module, which loops tokenized features multiple cycles through the transformer layers and makes the performance of the transformer network equivalent to that of a deeper transformer network.

Specifically, we followed the scheme of TokenPose[3]. for an input image *I*, we extract feature *F* from the backbone and then divide *F* into a grid of patches. Then we flatten them and use a linear projection function to form them as visual tokens *VT*. And we use extra *K* learnable tokens *KT* to represent *K* keypoints. Then we concatenated keypoint tokens with visual tokens and sent them to transformer encoder layers. For a MCT module that of cycled *N* times, we denote the output keypoint tokens and visual tokens of each cycle as $VT_1, KT_1 ... VT_N, KT_N$, respectively. For the i^{th} cycle, we take VT_{i-1} and KT_{i-1} as inputs and the outputs is VT_i and KT_i. At last, we use VT_N and KT_N as the transformer layers' output to make the prediction. The MCT module we designed has higher *latent depth* compared with transformer model which has the same number of transformer layers, and allows the parameters to be learned more fully for better performance. We further interpret this conclusion in Sec. 4.3.

3.2. Self-Distillation On MCT Module

The MCT module incurs additional computation, which we want to avoid without sacrificing model performance.

A seemingly promising approach is to use the result of the first cycle in the MCT directly, but this will drastically reduce performance. As shown in Fig. 4, during the first cycle through the transformer layers, the attention of the keypoint tokens is consistently focused on a smaller area and gradually contracts to a single location. However, during the second cycle through the transformer layers, the attention of the keypoint tokens expands to a larger area across all layers. This demonstrates that each cycle carries rich information, and naively ignoring outputs from latter cycles lead to information loss and thus performance degradation.

Inspired by works in self-distillation, we use the complete inference in the MCT module as a teacher and extract the knowledge of it into one single pass in the MCT module, which we used as a student. Since the input and output tokens are in the same vector space in the transformer layers, we can distill between output tokens from different cycles with minimized information loss.

Specifically, for an MCT module cycled *N* times, we use the output VT_i and KT_i to distill the the output VT_{i-1} and KT_{i-1} in the previous cycle. During training, with each inference we distill all the cycles, thus gradually distilling the knowledge to the first cycle.

Meanwhile, in order to constrain the correctness of the tokens, we send all the output tokens $VT_1, KT_1 ... VT_N, KT_N$ through the same prediction head to get predicted result $P_1, P_2 ... P_N$ respectively and constrain the predictions with the ground-truth.

Method	Params(M)	GFLOPs	*AP*	AP^{50}	AP^{75}	AP^{M}	AP^{L}	*AR*
SimpleBaseline-Res50[18]	34.0	8.9	70.4	88.6	78.3	67.1	77.2	76.3
SimpleBaseline-Res101[18]	53.0	12.4	71.4	89.3	79.3	68.1	78.1	77.1
SimpleBaseline-Res152[18]	68.6	15.7	72.0	89.3	79.8	68.7	78.9	77.8
TokenPose-S-V1*[3]	6.6†	2.4†	69.5†	87.7	77.1	65.7	76.6	74.9
TokenPose-S-V2*[3]	6.2	4.7	71.8‡	88.7	79.0	68.3	78.5	77.0
TokenPose-B*[3]	13.2	5.2	73.2§	89.5	80.2	70.1	79.8	78.7
OKDHP-2HG[7]	13.0	25.5	72.8	91.5	79.5	69.9	77.1	75.6
OKDHP-4HG[7]	24.0	47.0	74.8	92.5	81.6	72.1	78.5	77.4
PPT-S*[14]	6.6	2.0	69.8	87.7	76.8	66.1	76.7	75.1
PPT-B*[14]	13.2	4.7	73.4	89.5	80.8	70.3	79.8	78.8
SDPose-T(Ours)	4.4†(↓ 33.3%)	1.8†(↓ 25.0%)	**69.7**†(↑ 0.2%)	**88.1**	**77.3**	**66.1**	**76.6**	**75.2**
SDPose-S-V1(Ours)	6.6	2.4	**72.3**†(↑ 2.8%)	**89.2**	**79.6**	**68.8**	**79.1**	**77.7**
SDPose-S-V2(Ours)	6.2	4.7	**73.5**‡(↑ 1.7%)	**89.5**	**80.4**	**70.1**	**80.3**	**78.7**
SDPose-B(Ours)	13.2	5.2	**73.7**§(↑ 0.5%)	**89.6**	**80.4**	**70.3**	**80.5**	**79.1**

Table 1. Results of heatmap-based methods on MSCOCO validation dataset. the input size is 256×192. * means we re-train and evaluate the models on mmpose[19]. †, ‡ and § represents the data pair for comparison.

Methods	Backbone	Input Size	Params(M)	GFLOPs	FPS	*AP*
PRTR[20]	ResNet-50	384×288	41.5	11.0	33.8	68.2
PRTR[20]	ResNet-50	512×384	41.5	18.8	32.7	71.0
Poseur[11]	MobileNetV2	256×192	11.4	0.5	12.1	71.9
Poseur[11]	ResNet-50	256×192	33.3	4.6	12.0	75.4
RLE[12]	ResNet-50	256×192	23.6	4.0	51.5	70.5
DistilPose-S[8]	Stemnet	256×192	5.4	2.4	39.2	71.6
SDPose-Reg(Ours)	Stement	256×192	5.4	2.4	38.3	**73.0**

Table 2. Results of regression-based methods on MSCOCO validation dataset.

3.3. Loss Function

For the tokens of each cycle, we use the output tokens as teachers to distill the input tokens, respectively. Specifically, our loss is designed as:

$$L_{kt} = \sum_{i=1}^{N-1} MSE\left(KT_i, KT_{i+1}\right) \tag{1}$$

$$L_{vt} = \sum_{i=1}^{N-1} MSE\left(VT_i, VT_{i+1}\right) \tag{2}$$

where MSE refers to the Mean Squared Error Loss which has been shown to be effective in measuring differences between tokens.

Meanwhile, in order to ensure that the results of both cycle predictions are correct, we compute the loss of both predicted heatmaps with the ground truth:

$$L_{pose} = \sum_{i=1}^{N} MSE\left(P_i, GT\right) \tag{3}$$

where GT represents the ground truth.

In summary, The overall loss function of our self-distillation framework is as follows:

$$L = L_{pose} + \alpha_1 L_{kt} + \alpha_2 L_{vt}, \tag{4}$$

where α_1, α_2 are hyper-parameters.

4. Experiments

We evaluated the proposed SDPose models and performed abundant ablation studies on MSCOCO as well as Crowdpose dataset[21, 22]. For fairness, all our experiments are conducted using MMPose[19] framework.

4.1. Implementation Details

4.1.1 Datasets

We conducted experiments on 2 datasets, the MSCOCO dataset[21] and the Crowdpose dataset [22]. MSCOCO contains over 200K human body images, with each human body having 17 pre-annotated keypoints. We use MSCOCO train2017 with 57K images to train our models and compare methods. We evaluated them on both MSCOCO val2017

Methods	Input Size	Params(M)	GFLOPs	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
heatmap based methods								
TokenPose-S-V1*[3]	256 × 192	6.6	2.4	68.6	89.9	76.1	65.1	74.5
TokenPose-S-V2*[3]	256 × 192	6.6	4.7	71.1	90.4	78.7	67.7	77.1
PPT-S*[14]	256 × 192	6.6	2.0	69.2	90.1	76.8	65.8	75.2
SDPose-T(Ours)	256 × 192	4.4	1.8	69.2	90.2	76.8	65.7	75.2
SDPose-S-V1(Ours)	256 × 192	6.6	2.4	71.7	91.1	79.5	68.3	77.5
SDPose-S-V2(Ours)	256 × 192	6.2	4.7	**72.7**	**91.2**	**80.3**	**69.3**	**78.5**
regression based methods								
PRTR-Res101[20]	512 × 384	60.4	33.4	72.0	89.3	79.4	67.3	79.7
RLE-Res50*[12]	256 × 192	23.6	4.0	69.8	90.1	77.5	67.2	74.3
DistilPose-S*[8]	256 × 192	5.4	2.4	71.0	91.0	78.9	67.5	76.8
SDPose-Reg(Ours)	256 × 192	5.4	2.4	**72.1**	**91.2**	**79.5**	**68.6**	**78.0**

Table 3. Result on MSCOCO test-dev dataset. * means we re-train and evaluate the models on MMPose[19].

Methods	AP	AR
TokenPose-S-V1[3]	55.7†	65.2
TokenPose-S-V2[3]	62.3‡	71.3
SDPose-S-V1(Ours)	**57.3**† (↑ 1.6%)	**66.8**
SDPose-S-V2(Ours)	**64.5**‡ (↑ 2.2%)	**73.7**

Table 4. Results of heatmap-based methods on Crowdpose test dataset. † and ‡ represents the data pair for comparison.

with 5K images and dev2017 with 20K images, individually. We also evaluated our methods in the more challenging crowded scene using Crowdpose. This dataset consists of 20K human body images containing about 80K persons with overlaps of body parts, with 14 pre-annotated keypoints per person.

The bounding box and evaluation metrics used for our evaluations are consistent with previous works[3, 12, 22].

4.1.2 Settings And Training

We applied our method to TokenPose-S-V1, TokenPose-S-V2, and TokenPose-B. We substitute the naive transformer module with our MCT module and all other model configurations remain consistent with those in TokenPose[3] paper. We name the MCT-based models SDPose-S-V1, SDPose-S-V2, and SDPose-B. To demonstrate that our method can improve the *latent depth*, we set up a smaller model SDPose-T which changes TokenPose-S-V1 to six transformer layers and inference three cycles during training, using the latter cycle to distill the former. Meanwhile, we also designed a regression-based model SDPose-Reg, which uses the regression-based head named TokenReg in Distilpose[8] with an RLE loss[12], and used our method to train. For our method, we train the models on a machine with 8 NVIDIA Tesla V100 GPUs, allocating 64 samples per GPU. We use the Adam optimizer for 300 epochs of training. The initial learning rate was set to 1e-3 and decayed by a factor of ten at epochs 200 and 260, respectively. For our loss function, we set the hyper-parameters to $\alpha_1 = \alpha_2 = 5e-6$.

4.2. Main Results

Compared with heatmap-based methods. As Tab. 1 shown, our proposed SDPose achieves competitive performance compared with the other small-scale models. We mainly compare our methods with TokenPose[3], OKDHP[7], and PPT[14]. Specifically, SDPose-S-V1 achieves 72.3% AP with 6.6M params and 2.4 GFLOPs, Which under the same parameter and computational complexity as TokenPose-S-V1, and makes a 2.8% AP improvement. Similarly, SDPose-S-V2 and SDPose-B achieve 1.7% and 0.5% AP improvement with the same parameter and GFLOPs as TokenPose-S-V2 and TokenPose-B, respectively. Furthermore, SDPose-T slightly improves performance(↑ 0.2%) with a significantly lower number of parameters(↓ 33.3%) and GFLOPs(↓ 25.0%) compared to TokenPose-S-V1. Compared to other lightweight methods, our approach achieved higher performance with fewer parameters in most cases. Specifically, SDPose-T reduces more number of parameters and computation without degrading the performance compared to PPT-S. Also, Tab. 3 shows the results of our method and those of the other small models on the MSCOCO test-dev set. We see that SDPose-S-V2 achieved SOTA performance among the small models. In addition, our method can be applied to PPT to get better performance. Detail results are presented in Sec. 4.5.
Compared with regression-based methods. As shown in Tab. 2 and Tab. 3, our proposed SDPose achieves competitive performance compared with the other regression models. Compared with PRTR[20], which is also a transformer-based model, our method achieved a 1.5% AP improvement with a significant reduction in the number of parame-

L_{pose}	Distillation		AP	Improv.
	L_{kt}	L_{vt}		
			55.4%	-
✓			70.2%	↑ 14.8%
✓	✓		69.6%	↑ 14.2%
✓		✓	71.7%	↑ 16.3%
	✓	✓	58.2%	↑ 2.8%
✓	✓	✓	72.3%	↑ 16.9%

Table 5. Ablation studies for different distillation types. All ablation experiments are based on SDPose-S-V1, The combination of all distillation loss brings the best performance, which is our method. L_{pose} Means use the results predicted by each cycle to calculate the loss or use only the last cycle. Improv. = Improvement.

ters and GFLOPs. Compared with the smaller Poseur[11], our method has a 0.6% AP improvement with a significant FPS improvement(↑ 26.2). Compared with DistilPose-S[8], 1.4% AP improvement was also obtained using our method. Also, as Tab. 3 shown, SDPose-Reg makes a 1.1% AP improvement compared with DistilPose-S[8] on the MSCOCO test-dev set.

Evaluation on Crowdpose dataset. To verify our model's generalizability and to challenge our models to a harder scenario, we trained and evaluated our models on the Crowdpose dataset. As shown in Tab. 4, all of our MCT-based models outperform their corresponding naive transformer-based baselines.

4.3. Visualization

To explore the reasons for the performance improvement of our method, we first visualized the attention map between keypoint tokens and visual tokens in different transformer layers of the MCT module. As shown in Fig. 4, each cycle carries a lot of information with it. We also visualized the attention maps between keypoint tokens in different transformer layers. As shown in Fig. 5, during the first cycle through the transformer layers, similar to the ordinary baseline method, the attention of the keypoint tokens gradually concentrates on themselves. However, during the second cycle through the transformer layers, the attention of the keypoint tokens is redistributed to all other keypoint tokens, which we think represented obtaining more global information. Overall, this global information in the MCT module enables better learning of the parameters.

Furthermore, we visualized the distribution of transformer parameters. As shown in Fig. 2, TokenPose[3] has more near-zero parameters in each transformer layer, which are commonly considered to be insufficiently trained parameters. Our approach has significantly fewer near-zero parameters than TokenPose[3], indicating that our network

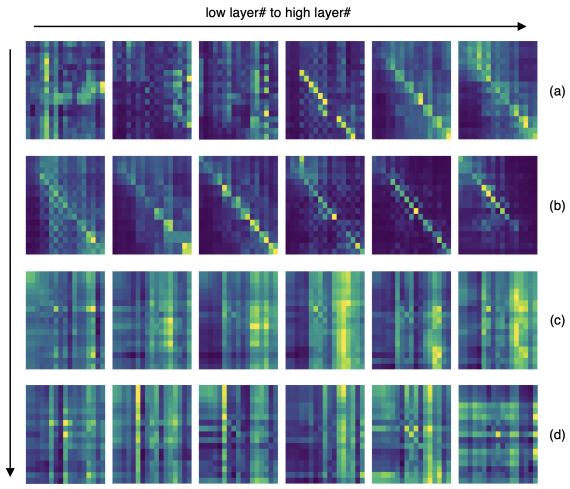

Figure 5. Visualization of the attention maps between nose keypoint token and visual tokens in different layers of MCT module. (a) (c) refer to layers #1-#6, (b) (d) refer to layers #7-#12 and (a) (b) refer to the first time, (c) (d) refer to the second time.

Backbone	Layers	Cycle	AP
Stemnet	12	1	69.5
Stemnet	12	2	73.3(↑ 3.8%)
Stemnet	12	3	72.4(↑ 2.9%)
Stemnet	4	2	69.4(↓ 0.1%)
Stemnet	4	3	70.8(↑ 1.3%)

Table 6. Ablation studies for different cycle networks without distillation. Layers means the transformer layer number. Cycle means the number of times that token through transformer layers.

Backbone	Layers	Cycle	Distil.	AP
Stemnet	12	1	-	69.5
Stemnet	12	2	2→1	72.3(↑ 2.8%)
Stemnet	12	3	3→2,2→1	71.7(↑ 2.2%)

Table 7. Ablation studies for different distillation settings. Layers means the transformer layer number. Cycle means the number of times that token through transformer layers. Distil. = distillation method, which means the type of distillation method. 2→1 means distillation second times result to first times result.

is more adequately trained. This also demonstrates that the more global information contained in the MCT module allows for more adequate parameter learning.

4.4. Ablation Studies

4.4.1 Losses

In this section, we investigated the contribution of distillation losses from different parts to the performance of our method. As shown in Tab. 5, we set different distillation

losses in various experiments. When not using distillation and directly predicting results from the tokens obtained in the first cycle, the network only has the constraint from the second cycle's final output. In this case, the tokens output from the first cycle are equivalent to intermediate results. The direct use of its predictions loses the information embedded in the later cycles, so the performance is poor. When we use distillation loss, the model performance can be improved, which proves that the keypoint tokens learn more information through the MCT module and thus interact better with the model parameters. As we gradually add distillation losses, the performance of the first cycle prediction gradually improves. When all three parts of the distillation loss are added, the performance reaches its best.

4.4.2 Network Scale

In this section, We evaluated the effectiveness of our method under different network configurations and sizes. As shown in Tab. 6, We first investigated the effectiveness of our network augment method. We set different MCT module cycle numbers based on TokenPose-S-V1 and trained without using self-distillation, predicting the output from the last cycle. When we increase the number of cycles, the performance is improved compared to using only a single pass. This fully demonstrates that our augment method enables the transformer layers to learn more information, and effectively augment the original network to a deeper transformer network. To demonstrate the importance of introducing global information, we also designed a network with fewer layers but using three cycles. It is equivalent to going through 12 layers of transformer layers but gets better performance than the baseline. Furthermore, we noticed that the performance of the 12 transformer layers network with three cycles is lower than that of the network with two cycles. We believe that although multiple cycles can help tokens pay attention to more global information, too many cycles may cause the network to forget the more critical local information of keypoints. As shown in Tab. 7, We designed a distillation experiment with a three-cycled MCT module. The test performance was lower than that of the distillation experiment with a two-cycled MCT module. This also proves that excessive augment of the network structure to learn global knowledge is not entirely beneficial.

4.5. Extensibility Study

In this section, we further investigate the extensibility of our approach. As shown in Tab. 8, we show the performance of our method when combined with PPT[14]. For training, we pruned the tokens using PPT[14] in the first cycle of SDPose. Ultimately, the performance maintained the level of SDPose-S-V1 while reducing the computation to the same of PPT-S. This proves that our method works well in conjunction with other lightweight methods.

Methods	Params(M)	GFLOPs	AP
TokenPose-S-V1*[3]	6.6	2.4	69.5
PPT-S*[14]	6.6	2.0	69.8
SDPose-S-V1+PPT[14]	6.6	2.0	72.3

Table 8. Result of combined method on MSCOCO validation dataset. *means we re-train and evaluate the models on mmpose[19].

Deit-Tiny[23]	AP
	72.2
+ 2 Cycle	73.4
+ SDPose (Ours)	72.7(↑ 0.5%)

Table 9. Results on Imagenet 1K dataset. Cycle means we apply different cycle networks without distillation on the Deit-Tiny.

We also migrated SDPose to the classification task. We used Deit-Tiny[23] as our baseline. As shown in Tab. 9, when we applied the MCT module on Deit-Tiny[23], the performance of the network was significantly improved. When we trained the baseline using our SDPose, the performance of the network was also slightly improved without additional parameters and computations. This demonstrates the ability of our method to be extended to various tasks of the transformer-based model.

5. Conclusion

In this work, we proposed a novel human pose estimation framework, termed SDPose, which includes a Multi-Cycled Transformer(MCT) module and a self-distillation paradigm. Through our design, we have enabled the small transformer-based model to be dramatically improved without increasing the amount of computation and the number of parameters, and achieved new state-of-the-art in the same scale models. Meanwhile, we also extend our method to other models, proving the generality of our method.

In short, SDPose achieved state-of-the-art performance among the same scale models.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62302297, No.72192821), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Science and Technology Commission (21511101200), Shanghai Sailing Program (22YF1420300), Young Elite Scientists Sponsorship Program by CAST (2022QNRC001) and YuCaiKe[2023] Project Number: 14105167-2023.

References

[1] Arpita Vats and David C. Anastasiu. Key point-based driver activity recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun 2022. 1

[2] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, page 90–126, Nov 2006. 1

[3] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. *CoRR*, abs/2104.03516, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[4] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *CoRR*, abs/2012.14214, 2020. 1, 2

[5] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation, 2022. 1, 3

[6] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *CoRR*, abs/1902.09212, 2019. 1

[7] Zheng Li, Jingwen Ye, Mingli Song, Ying Huang, and Zhigeng Pan. Online knowledge distillation for efficient pose estimation. *CoRR*, abs/2108.02092, 2021. 1, 5, 6

[8] Suhang Ye, Yingyi Zhang, Jie Hu, Liujuan Cao, Shengchuan Zhang, Lei Shen, Jun Wang, Shouhong Ding, and Rongrong Ji. Distilpose: Tokenized pose regression with heatmap distillation, 2023. 1, 2, 3, 5, 6, 7

[9] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Practical guidelines for vit feature knowledge distillation, 2022. 1

[10] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013. 2

[11] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, Zhibin Wang, and Anton van den Hengel. Poseur: Direct human pose regression with transformers. *CoRR*, abs/2201.07412, 2022. 2, 5, 7

[12] Jiefeng Li, Siyuan Bian, Ailing Zeng, Can Wang, Bo Pang, Wentao Liu, and Cewu Lu. Human pose regression with residual log-likelihood estimation. *CoRR*, abs/2107.11291, 2021. 2, 5, 6

[13] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. *CoRR*, abs/1406.2984, 2014. 2

[14] Haoyu Ma, Zhe Wang, Yifei Chen, Deying Kong, Liangjian Chen, Xingwei Liu, Xiangyi Yan, Hao Tang, and Xiaohui Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation, 2022. 3, 5, 6, 8

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 3

[16] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *CoRR*, abs/1905.08094, 2019. 3

[17] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks, 2018. 3

[18] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *CoRR*, abs/1804.06208, 2018. 5

[19] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020. 5, 6, 8

[20] Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. *CoRR*, abs/2104.06976, 2021. 5, 6

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. *Microsoft COCO: Common Objects in Context*, page 740–755. Jan 2014. 5

[22] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10863–10872, 2019. 5, 6

[23] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. 8