A APPENDIX

A.1 MORE BENCHMARK RESULTS

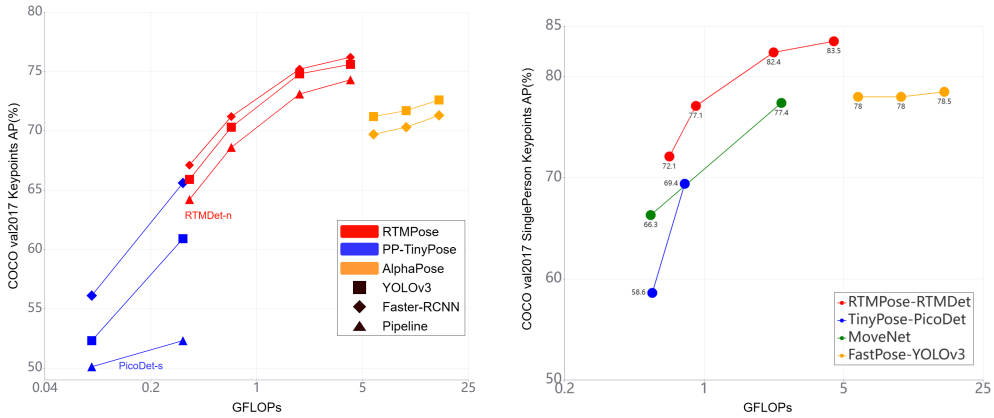

Figure 3: Comparison of GFLOPs and accuracy. Left: Comparison of RTMPose and other open-source pose estimation libraries on full COCO val set. Right: Comparison of RTMPose and other open-source pose estimation libraries on COCO-SinglePerson val set.

COCO-SinglePerson Popular pose estimation open-source libraries like BlazePose (Bazarevsky et al., 2020), MoveNet (Votel et al., 2023), and PaddleDetection (Authors) are designed primarily for single-person or sparse scenarios, which are practical in mobile applications and human-machine interactions. For a fair comparison, we construct a COCO-SinglePerson dataset that contains 1045 single-person images from the COCO `val2017` set to evaluate RTMPose as well as other approaches. For MoveNet (Votel et al., 2023), we follow the official inference pipeline to apply a cropping algorithm, namely using the coarse pose prediction of the first inference to crop the input image and performing a second inference for better pose estimation results. The evaluation results in Table 9 and Fig. 3 show that RTMPose archives superior performance and efficiency even compared to previous solutions tailored for the single-person scenario.

Table 9: Body pose estimation results on COCO-SinglePerson validation set. We sum up top-down methods' GFLOPs of detection and pose for a fair comparison with bottom-up methods. "*" denotes double inference. Flip test is not used.

Methods		Backbone	Detector	Det. Input Size	Pose Input Size	GFLOPs	AP	Extra Data
MediaPipe (Bazarevsky et al., 2020)	BlazePose-Lite	BlazePose	N/A	256 × 256	N/A	N/A	29.3	Internal(85K)
	BlazePose-Full	BlazePose	N/A	256 × 256	N/A	N/A	35.4	
MoveNet (Votel et al., 2023)	Lightning	MobileNetv2	N/A	192 × 192	N/A	0.54	53.6*	Internal(23.5K)
	Thunder	MobileNetv2 depth×1.75	N/A	256 × 256	N/A	2.44	64.8*	
PaddleDetection (Authors)	TinyPose	Wider NLiteHRNet	PicoDet-s	320 × 320	128 × 96	0.55	58.6	AIC(220K)
	TinyPose	Wider NLiteHRNet	PicoDet-s	320 × 320	256 × 192	0.80	69.4	+Internal(unknown)
MMPose (Contributors, 2020)	RTMPose-t	CSPNeXt-t	RTMDet-nano	320 × 320	256 × 192	0.67	72.1	AIC(220K)
	RTMPose-s	CSPNeXt-s	RTMDet-nano	320 × 320	256 × 192	0.91	77.1	
	RTMPose-m	CSPNeXt-m	RTMDet-nano	320 × 320	256 × 192	2.23	82.4	
	RTMPose-l	CSPNeXt-l	RTMDet-nano	320 × 320	256 × 192	4.47	83.5	

A.2 INFERENCE SPEED

In this appendix, we extend our experimentation to assess the inference speed of RTMPose on a mobile device using ncnn for deployment and testing. Table 10 demonstrates the comparison of inference speed on the mobile device, specifically the Snapdragon 865 chip with RTMPose models.

Furthermore, we maintained our evaluation of TensorRT inference latency on an NVIDIA GeForce GTX 1660 Ti GPU in the half-precision floating-point format (FP16) and ONNX latency on an Intel I7-11700 CPU with ONNXRuntime, using a single thread. The inference batch size remained

consistent at 1. All models underwent a rigorous testing regimen on the same devices, including 50 warm-up runs and 200 inference runs to ensure a fair comparison.

For a comprehensive evaluation, we also included TinyPose (Authors) in our tests, assessing it with both MMDeploy and FastDeploy. We observed that ONNXRuntime speed on MMDeploy was slightly faster (10.58 ms vs. 12.84 ms). The detailed results can be found in Table 10.

Table 10: Comparison of inference speed on Snapdragon 865. RTMPose models are deployed and tested using ncnn.

Methods		Input Size	GFLOPs	AP(GT)	FP32(ms)	FP16(ms)
PaddleDetection (Authors)	TinyPose	128×96	0.08	58.4	4.57	3.27
	TinyPose	256×192	0.33	68.3	14.07	8.33
MMPose (Contributors, 2020)	RTMPose-t	256×192	0.36	68.4	15.84	9.02
	RTMPose-s	256×192	0.68	72.8	25.01	13.89
	RTMPose-m	256×192	1.93	77.3	49.46	26.44
	RTMPose-l	256×192	4.16	78.3	85.75	45.37

Table 11 analyzes inference speeds across models and devices, revealing the balance between accuracy and speed. RTMPose performs well across sizes, while RTMDet-nano prioritizes efficiency. This data aids in selecting models for diverse real-time applications.

Table 11: Pipeline Inference speed on CPU, GPU and Mobile device.

Model	Input Size	GFLOPs	Pipeline AP	CPU(ms)	GPU(ms)	Mobile(ms)
RTMDet-nano RTMPose-t	320×320 256×192	0.31 0.36	64.4	12.403	2.467	18.780
RTMDet-nano RTMPose-s	320×320 256×192	0.31 0.42	68.5	16.658	2.730	21.683
RTMDet-nano RTMPose-m	320×320 256×192	0.31 1.93	73.2	26.613	4.312	32.122
RTMDet-nano RTMPose-l	320×320 256×192	0.31 4.16	74.2	36.311	4.644	47.642