UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

LAB 03

Deep Learning Research

REPORT: POSE ESTIMATION PROJECT

Supervisor: Võ Nhật Tân & Lê Nhựt Nam

Students:	**Full Name**	**ID**
	Nguyễn Lê Quang	23127109
	Phan Hoàng Quang Nghị	23127436
	Nguyễn Tấn Văn	23127515

Ho Chi Minh City, 2025

Mục lục

1 Proposed Method

1. Problem Statement

The paper addresses a critical gap in the field of 2D pose estimation: the disconnect between academic research and industrial application. State-of-the-art academic models (e.g., HRNet, ViTPose) achieve high accuracy but suffer from heavy parameters and high latency, making them unsuitable for edge devices. Conversely, existing industrial solutions prioritize speed but often fail to meet the accuracy requirements for critical applications.

2. Key Idea and Approach

The authors propose **RTMPose**, a high-performance real-time multi-person pose estimation framework. The key idea is to empirically optimize the pose estimation framework across five dimensions: paradigm, backbone, localization, training strategy, and deployment.

The specific technical approaches include:

- **Efficient Top-Down Paradigm:** RTMPose utilizes the Top-Down paradigm. Unlike traditional views that top-down methods are slow, the authors leverage modern real-time object detectors (e.g., RTMDet) to remove the detection bottleneck, allowing the system to achieve superior accuracy compared to bottom-up methods while maintaining real-time speed.
- **CSPNeXt Backbone:** Instead of using classification-based backbones (like ResNet), the model employs CSPNeXt, an architecture optimized for dense prediction tasks. This ensures a better balance between computational cost and accuracy.
- **SimCC-based Localization:** A core improvement is the adoption of Coordinate Classification (SimCC) instead of heatmap regression. By treating keypoint localization as a classification task for horizontal and vertical coordinates, the model eliminates expensive upsampling layers and reduces quantization errors, resulting in a lightweight architecture.
- **Inference Pipeline Optimization:** The framework implements a "skip-frame detection" strategy and employs a OneEuro smoothing filter during post-processing to enhance stability and reduce latency.

3. Version Comparison (v1 vs. v2)

The research has evolved from its initial version (v1) to the current version (v2) with notable improvements:

- **Model Variants:** While v1 introduced tiny, small, medium, and large (t/s/m/l) variants, v2 added an **extra-large (x)** variant to cater to high-precision scenarios.
- **Rigorous Evaluation:** Version 2 presents corrected evaluation results for the COCO-WholeBody dataset. For instance, the performance of RTMPose-l was adjusted from 67.0% AP in v1 to a more rigorous 64.8% AP in v2, reflecting improved evaluation integrity.

2 Related Work

The authors position RTMPose within the broader research landscape by comparing it against four major categories of existing methods:

1. Bottom-up Approaches

Methods like OpenPose and HigherHRNet detect all keypoints in an image first and then group them into individuals.

- *Limitation:* Although they offer stable computational costs in crowd scenarios, they require high-resolution inputs to handle scale variations, making it difficult to balance accuracy and inference speed.

2. Top-down Approaches

Approaches such as AlphaPose and HRNet crop individuals using a detector before estimating poses.

- *Positioning:* These methods dominate public benchmarks but are traditionally stereotyped as slow due to the detection step. RTMPose challenges this by proving that top-down methods can be real-time when paired with efficient detectors like RTMDet.

3. Coordinate Classification (SimCC)

This research builds upon SimCC, which formulates keypoint localization as a classification problem rather than regression.

- *Advantage:* Unlike heatmap-based methods that require costly upscaling layers, SimCC allows for a compact architecture. RTMPose further optimizes this by using fully-connected layers, avoiding spatial information loss and quantization errors associated with coordinate regression.

4. Vision Transformers

Transformer-based models (e.g., ViTPose, TransPose) achieve state-of-the-art accuracy but incur high computational costs.

- *Improvement:* RTMPose replaces heavy transformer blocks with a lightweight **Gated Attention Unit (GAU)** combined with the SimCC representation. This enables the model to capture global dependencies effectively without the heavy latency of traditional vision transformers.

3 Model / Theory

3.1 Theoretical Framework & Approach Paradigm

The paper explicitly identifies the gap between academic research (which focuses on accuracy but is computationally heavy) and industrial applications (which require real-time speed). To address this issue, the authors adopted the following approach:

Top-Down Approach:

- **Mechanism:** The system operates in two steps: (1) Using an Object Detector to identify the bounding box of the person; (2) Cropping the person's image and feeding it into the pose estimation model to locate keypoints.
- **Countering historical stereotypes:** Previously, Top-down methods were considered slow due to their dependence on the detection step and the fact that computational load increases linearly with the number of people. However, the paper argues that with the advent of high-performance real-time detectors (such as RTMDet), the detection step is no longer a bottleneck.
- **Advantages:** In standard scenarios (under 6 people/image), this method allows for real-time inference with higher accuracy compared to Bottom-up methods, as the pose model processes images at a standardized scale.

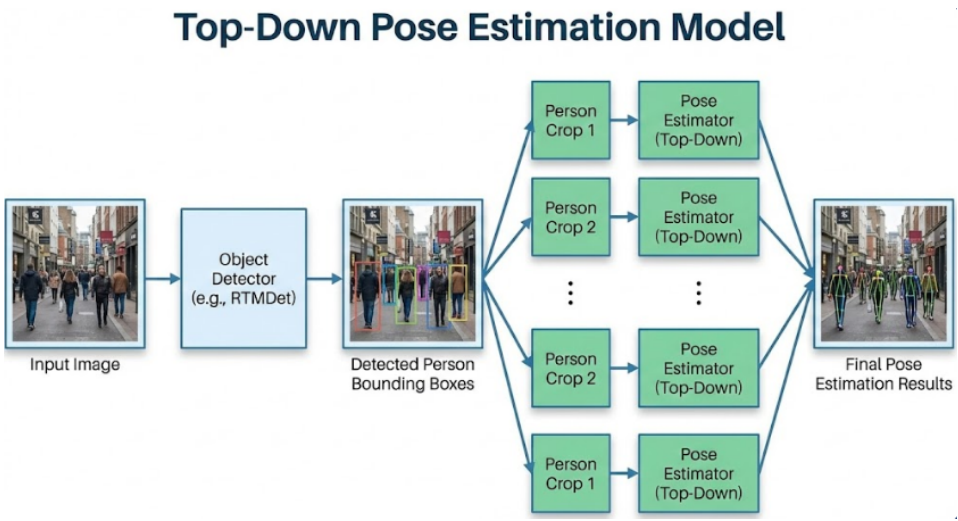

Figure 1: Illustration of the Top-Down Pose Estimation pipeline.

3.2 Model Architecture

3.2.1 Backbone: CSPNeXt (Large-kernel CSP Convolutional Network)

The Shift: Traditional backbones often face two issues: Image classification backbones (like ResNet) reduce spatial resolution too quickly, causing a loss of positional information necessary for dense prediction tasks like pose estimation. Conversely, high-resolution backbones (like HRNet) are too bulky and have high inference latency, making them difficult to deploy on edge devices.

Detailed Structure of CSPNeXt: RTMPose utilizes the backbone from RTMDet (a high-performance object detector by the same authors). This architecture features the following technical characteristics:

- **Cross Stage Partial (CSP) Mechanism:** This architecture splits the input feature flow into two parts: one part goes through a dense computation block, while the other goes through a skip connection and concatenates directly at the end of the block. This mechanism reduces computation and gradient duplication while maintaining strong feature extraction capabilities.
- **Large-kernel Depth-wise Convolution:** Instead of using the standard 3×3 kernel, CSPNeXt uses a large 5×5 kernel in depth-wise convolution layers. Increasing the kernel size expands the receptive field, allowing the model to capture global context better (e.g., seeing the relationship between distant body parts) without significantly increasing computational costs.

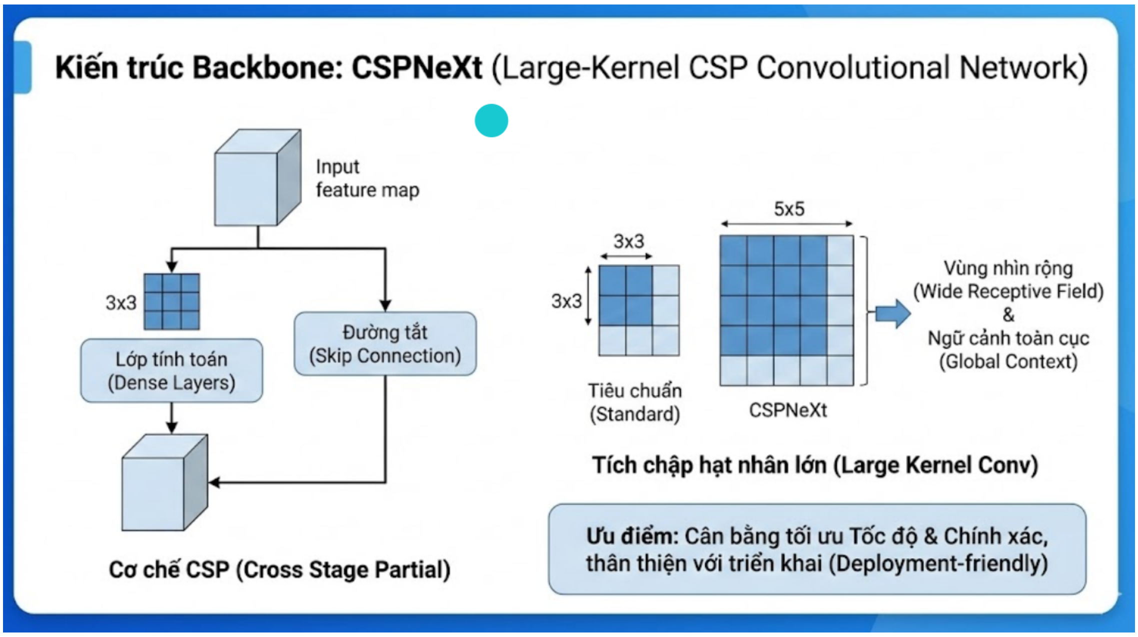

Figure 2: Structure of the CSPNeXt Backbone with Large-kernel Convolutions.

Benefits: CSPNeXt achieves an optimal balance between speed and accuracy. Its structure is highly "deployment-friendly" on hardware such as GPUs and NPUs, overcoming the weaknesses of Transformer or HRNet architectures.

3.2.2 Prediction Head: SimCC (Simple Coordinate Classification)

This is the core improvement replacing the traditional Heatmap method, which is heavy and resource-intensive.

Detailed Operating Principle: SimCC reshapes the keypoint localization problem from Regression to Classification through the following steps:

- **Decoupling:** Instead of predicting in a 2D space ($H \times W$), SimCC splits the problem into two independent classification tasks for the horizontal axis (x) and the vertical axis (y).
- **Discretization:** The input image is divided into grids or "bins". The horizontal axis is divided into W_x bins, and the vertical axis into W_y bins. Each bin represents a sub-pixel coordinate range (smaller than 1 pixel).
- **Prediction:** The model predicts which "bin" the keypoint falls into.

Soft Label Mechanism (Gaussian Label Smoothing / SORD): To help the model learn more effectively, SimCC avoids using hard labels (one-hot encoding).

- **Problem with Hard Labels:** A prediction error of 1 bin (very close) is penalized as heavily as an error of 100 bins (very far), making it difficult for the model to converge.
- **Solution:** Using a Gaussian distribution to create soft labels around the Ground Truth position. This helps the model learn the ordinal nature of the space: bins neighboring the correct position have higher probabilities than distant bins.

Architecture Optimization (Trimmed SimCC / SimCC*): RTMPose uses a streamlined version of SimCC (SimCC*):

- **Removing Upsampling:** Unlike heatmap methods that require Deconvolution layers to upscale feature maps (consuming high FLOPs), SimCC predicts directly from low-resolution features.
- **Experimental Results:** Removing this layer reduced the Head's FLOPs from 1.4G to just 0.002G and parameters from 13.2M to 0.079M, while accuracy dropped only slightly (from 72.1% to 71.3% AP). This minor drop was later compensated for by stronger training strategies and the backbone.

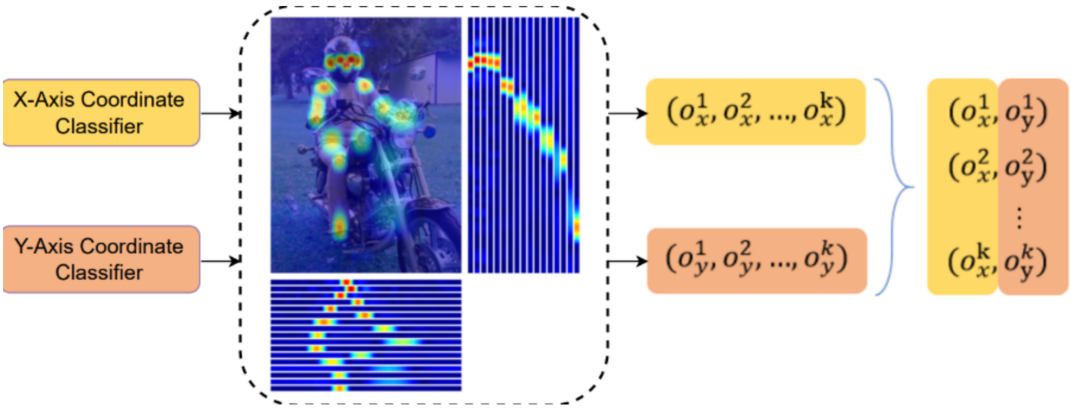

Figure 3: Illustration of SimCC: Decoupling 2D localization into two 1D classification tasks.

3.2.3 Self-Attention Module: Gated Attention Unit (GAU)

- **Purpose:** To refine keypoint representations and exploit both global and local spatial information.
- **Structure:** GAU is used instead of the standard Transformer. GAU improves the Feed-Forward Network (FFN) by using a Gated Linear Unit (GLU).
- **Formula:**

$$A = relu^2\left(\frac{Q(Z)K(Z)^\top}{\sqrt{s}}\right) \tag{1}$$

  Where $relu^2(\cdot)$ is the squared ReLU activation function.
- **Effectiveness:** GAU is faster, consumes less memory, and improves accuracy by an additional 0.5% AP (from 71.4% to 71.9%).

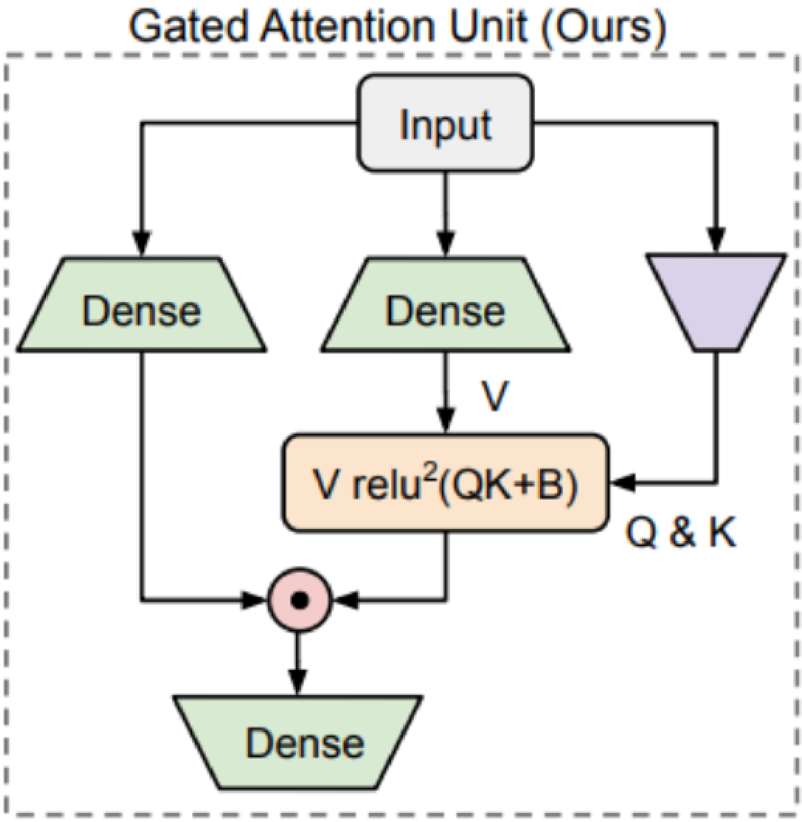

Figure 4: The architecture of Gated Attention Unit (GAU).

3.3 Training Strategies

The authors refer to these as "empirical training strategies" to help the model achieve maximum performance:

- **Pre-training:** The backbone is pre-trained using the UDP (Unbiased Data Processing) method based on heatmaps. This technique improves accuracy from 69.7% to 70.3% AP.
- **Optimization:**
  - Using EMA (Exponential Moving Average) to reduce overfitting.
  - Flat Cosine Annealing schedule for learning rate.
  - Disabling weight decay for normalization layers and biases.
- **Two-stage Augmentation:**
  - *Strong Stage (First 180 epochs):* Uses large random rotation (factor 80), wide scaling [0.6, 1.4], and specifically Cutout with 100% probability. The goal is to force the model to learn pose structure rather than relying on image texture.

- *Weak Stage (Last 30 epochs):* Disables random shift, reduces rotation amplitude, and reduces Cutout probability to 0.5. The goal is to fine-tune the model to match the real image distribution.

3.4 Optimization Techniques

These micro-adjustments contribute significantly to the final performance:

- **Feature Dimension Expansion:** Using a Fully Connected (FC) layer to expand keypoint representations to 256 dimensions, improving accuracy from 71.2% to 71.4% AP.
- **Loss Function & Soft Label:**
  - Using an unnormalized Gaussian distribution as the distance metric between classes.
  - The loss function is calculated based on Soft label encoding from SORD (Soft Labels for Ordinal Regression).
- **Softmax Temperature:** Adding a temperature parameter $\tau = 0.1$ to the Softmax function when calculating the probability distribution. This helps adjust the distribution shape, increasing accuracy to 72.7%.
- **Separate σ:** Instead of using a shared sigma value for both axes, RTMPose calculates separate sigma for x and y based on the number of bins for each axis (W_S). Formula: $\sigma = \sqrt{W_S/16}$.
- **Large Kernel Convolution:** The final convolutional layer uses a large 7×7 kernel instead of 1×1. Increasing this receptive field improves accuracy to 73.3% AP.
- **More Epochs and Multi-dataset Training:**
  - *Extended Training:* Prolonging the training process brings clear benefits. Specifically, increasing epochs to 270 and 420 results in 73.5% AP and 73.7% AP, respectively.
  - *Data Combination:* To maximize model potential, the researchers enriched training data by combining COCO and AI Challenger datasets. They used a balanced sampling ratio between datasets to ensure even feature learning.
- **Result:** This combined strategy helps the final model performance reach 75.3% AP.

Results		Input Size	GFLOPs	AP	CPU(ms)	GPU(ms)
COCO [37]	TinyPose	256 × 192	0.33	65.6	10.580	3.055
	LiteHRNet-30	256 × 192	0.42	66.3	22.750	6.561
	RTMPose-t	256 × 192	**0.36**	**67.1**	**3.204**	**1.064**
	RTMPose-s	256 × 192	**0.68**	**71.2**	**4.481**	**1.392**
	HRNet-w32+UDP	256 × 192	7.7	75.1	37.734	5.133
	RTMPose-m	256 × 192	**1.93**	**75.3**	**11.060**	**2.288**
	RTMPose-l	256 × 192	**4.16**	**76.3**	**18.847**	**3.459**
COCO-WholeBody [28]	HRNet-w32+DARK	256 × 192	7.72	57.8	39.051	5.154
	RTMPose-m	256 × 192	**2.22**	**59.1**	**13.496**	**4.000**
	RTMPose-l	256 × 192	**4.52**	**62.2**	**23.410**	**5.673**
	HRNet-w48+DARK	384 × 288	35.52	65.3	150.765	13.974
	RTMPose-l	384 × 288	**10.07**	**66.1**	**44.581**	**7.678**

Figure 5: Experimental results comparing RTMPose with other state-of-the-art methods.

3.5 Inference Pipeline Optimization

Beyond the pose estimation model, the research team optimized the entire inference pipeline for lower latency and better robustness:

- **Skip-frame Detection:** The Detector only runs every K frames. In intermediate frames, bounding boxes are generated from the pose estimation results of the preceding frame. This significantly reduces latency.
- **Post-processing:**
  - Using OKS-based Pose NMS (Non-Maximum Suppression) to eliminate duplicate predictions.
  - Using the OneEuro filter to smooth keypoint movements across frames, reducing jitter.

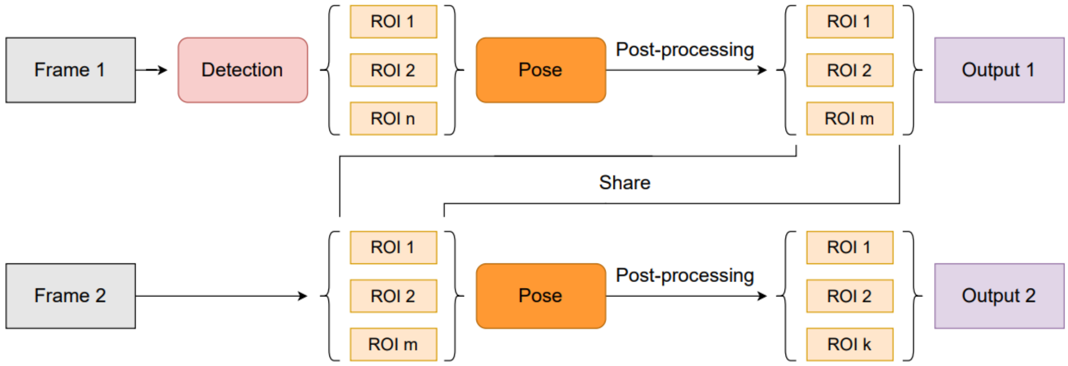

Figure 6: The inference pipeline with Skip-frame strategy. The detector runs only on key frames (Frame 1), while subsequent frames (Frame 2) reuse the tracking results.

4 Evaluation Metrics

To strictly evaluate the proposed RTMPose framework against both academic benchmarks and industrial requirements, the authors employ a comprehensive set of metrics covering accuracy, robustness, and efficiency.

1. Mean Average Precision (AP)
This is the primary evaluation metric used for the majority of datasets in the study, including COCO, COCO-SinglePerson, CrowdPose, and AP-10K.

- **Definition:** AP is calculated based on **Object Keypoint Similarity (OKS)**, which measures the proximity between predicted keypoints and ground-truth keypoints, normalized by the scale of the person. The paper reports mean AP (averaged over 10 OKS thresholds from 0.50 to 0.95) along with AP_{50}, AP_{75}, AP_M (medium objects), and AP_L (large objects).

- **Why it is appropriate:** AP is the standard metric for the COCO benchmark, enabling direct comparison with state-of-the-art top-down methods like HRNet and AlphaPose. It is robust to scale variations and provides a holistic view of the model's precision.

2. Average Recall (AR)
Although less emphasized in the text, Average Recall is explicitly reported for the **COCO-WholeBody** dataset evaluation (Table 6).

- **Definition:** AR measures the percentage of ground-truth keypoints that are successfully detected by the model, regardless of precision.

- **Why it is appropriate:** For whole-body estimation (which includes face, hands, and feet), avoiding false negatives is crucial. AR ensures that the model does not miss critical keypoints, providing a complementary perspective to AP.

3. Percentage of Correct Keypoints head (PCKh)
This metric is exclusively used for evaluating the **MPII Human Pose dataset** (Table 9).

- **Definition:** PCKh@0.5 calculates the percentage of keypoints that fall within a threshold of 50% of the head segment length from the ground truth.

- **Why it is appropriate:** PCKh is the designated standard for the MPII dataset. By normalizing error relative to head size, it remains invariant to the person's scale, ensuring fair comparison with baselines trained on this specific dataset.

4. Efficiency Metrics (GFLOPs, Latency, FPS)
Given the paper's title ("Real-Time") and its focus on industrial applications, efficiency metrics are as important as accuracy.

- **Metrics Used:**

  - **GFLOPs:** Quantifies the computational complexity (floating-point operations).
  - **Parameters (Params):** Measures the model size in millions.
  - **Inference Speed:** Measured in Latency (ms) and Frames Per Second (FPS) across different hardware: Intel i7-11700 CPU, NVIDIA GTX 1660 Ti GPU, and Snapdragon 865 mobile chip.

- **Why it is appropriate:** These metrics validate the core contribution of RTMPose: bridging the gap between high-accuracy academic models and real-time industrial needs. They demonstrate the model's feasibility for deployment on edge devices with limited computing power.

5 Datasets and Experiments

5.1 Datasets

The study employs a comprehensive set of datasets to ensure rigorous training, fair benchmarking, and evaluation of generalization capabilities.

1. **MS COCO (Common Objects in Context) – The Primary Dataset**
   - **Role:** Serves as the "backbone" of the study for main training and evaluation.
   - **Source & Characteristics:** COCO is the standard benchmark for 2D body pose estimation, featuring diverse data with multi-person scenarios, complex poses, and significant occlusion.
   - **Split:** The authors follow the standard split with *train2017* (118K images) for training and *val2017* (5K images) for validation.
   - **Usage:** Facilitates fair comparison with SOTA models (AlphaPose, HRNet, ViTPose) and evaluation of the full pipeline with detectors like YOLOv3 and RTMDet.

2. **COCO-SinglePerson – Custom Subset**
   - **Role:** Ensures fair comparison with lightweight, single-person specific models (e.g., BlazePose, MoveNet).
   - **Creation:** A subset filtered from COCO *val2017*, containing **1,045 images** with only a single person.
   - **Significance:** Demonstrates that RTMPose (a multi-person Top-down model) outperforms specialized single-person models in both speed and accuracy on their own turf.

3. **AI Challenger (AIC) – Augmented Dataset**
   - **Role:** Used for data enrichment to push performance boundaries.

- **Usage:** Combined with COCO for *pre-training* using a balanced sampling ratio. This strategy helped the model achieve competitive results (75.3% AP).

4. **COCO-WholeBody (V1.0)**
   - **Role:** Evaluates the model's scalability to fine-grained tasks.
   - **Characteristics:** Includes small and easily occluded keypoints: Hands, Face, and Feet.
   - **Usage:** Validates the SimCC architecture's ability to capture minute details.

5. **Cross-domain Datasets (MPII, AP-10K, CrowdPose)**
   - **Role:** Verifies generalization capabilities.
   - **Details:** MPII (Human activity, evaluated via PCKh), AP-10K (Animal pose), and CrowdPose (High crowding/occlusion scenes).

5.2 Experimental Setup

The experimental environment is rigorously designed to balance high accuracy with real-time inference speed.

5.2.1 Training Configuration

- **Hyperparameters:** The model is trained using the **AdamW** optimizer with a base learning rate of 0.004 and a **Flat-Cosine** decay schedule.
- **Batch Size:** A large batch size of **1,024** is used to ensure stable gradient descent.
- **Training Schedule:** The process consists of two phases:
  1. **Pre-training:** 210 epochs using the combined COCO and AIC datasets.
  2. **Fine-tuning:** 420 epochs on the target dataset.
- **Regularization:** Weight decay is set to 0.05 (for M/L models), and Exponential Moving Average (EMA) is applied to stabilize training.
- **Micro Design:** Softmax temperature $\tau = 0.1$, final convolution kernel size 7×7, and Gaussian Label Smoothing.

5.2.2 Augmentation Strategy

A **two-stage augmentation** strategy is employed to prevent overfitting while ensuring domain alignment:

- **Stage 1 (Strong - First 180 epochs):** Uses aggressive transformations including random scaling $[0.6, 1.4]$, large random rotation $[-80°, 80°]$, and Cutout to force the model to learn structural representations.

- **Stage 2 (Weak - Last 30 epochs):** Disables random shift and reduces rotation/Cutout to fine-tune the model on the realistic image distribution.

5.2.3 Hardware and Software Environments

- **Training Hardware:** All models are trained on a cluster of **8 × NVIDIA A100 GPUs**.
- **Inference Hardware:** To demonstrate real-time capabilities across devices, inference speed is tested on:
  - **Desktop CPU:** Intel Core i7-11700.
  - **Desktop GPU:** NVIDIA GeForce GTX 1660 Ti.
  - **Mobile Edge:** Snapdragon 865 chip.
- **Software Stack:** The framework is built upon **MMPose** and deployed using **MMDeploy** with support for ONNX Runtime, TensorRT, and ncnn backends.

6 Results and Discussion

6.1 Performance Comparison

The authors compared RTMPose against a wide range of formidable competitors in both segments: High Accuracy (SOTA) and High Speed (Real-time/Mobile).

- **Comparison with SOTA Models (HRNet):** Previous methods like HRNet (W32, W48) achieve high accuracy but suffer from heavy model parameters and high latency.
  - *Result:* RTMPose achieves equivalent or near-equivalent accuracy to HRNet but with superior speed and significantly lower computational costs.
  - *Evidence:* RTMPose-m achieves **75.8% AP** on the COCO validation set, outperforming competitors while maintaining **90+ FPS** on an Intel i7-11700 CPU.
- **Comparison with Mobile/Real-time Models (TinyPose, BlazePose, MoveNet):** These models are designed for devices with limited computing power. Libraries like BlazePose or MoveNet are often tailored for single-person scenarios.
  - *Result:* RTMPose-s achieves **72.2% AP**, outperforming existing open-source libraries.
  - On a Snapdragon 865 mobile chip, RTMPose runs faster and is more accurate than TinyPose and MoveNet. Specifically, RTMPose-s achieves **70+ FPS**.
- **Comparison on Whole-Body Task:** RTMPose-x achieves **65.3% AP** on the COCO-WholeBody dataset, surpassing methods like ZoomNet and OpenPose.

6.2 Strengths

The paper highlights four core strengths that contribute to the success of RTMPose:

1. **Excellent Speed-Accuracy Trade-off:** This is the most significant advantage. While RTMPose may not be the absolute most accurate model globally (compared to massive Transformer models), nor the lightest, it is the best model within the real-time region. As shown in Figure 1, RTMPose lies on the optimal Pareto frontier.

2. **Deployment-friendly:** The model employs the SimCC structure with simple Fully-Connected layers, avoiding complex operators (such as Deconvolution or Global Pooling) that complicate model conversion. It is easily deployable across various backends like ONNX Runtime, TensorRT, and ncnn, covering hardware from CPUs and GPUs to mobile chips.

3. **Lightweight and Efficient Architecture:** The utilization of the CSPNeXt Backbone and SimCC Head significantly reduces the number of Parameters and GFLOPs compared to heatmap-based methods.
    - *Example:* The optimized SimCC* (a trimmed version used in RTMPose) reduces the FLOPs of the prediction head from **1.4 G** to only **0.002 G**.

4. **Versatility:** RTMPose performs well across various tasks: standard Body Pose Estimation, Whole-body estimation (including face and hands), and Animal Pose estimation.

6.3 Limitations and Open Challenges

Although the paper focuses on the advantages, a technical analysis of the methodology and experiments reveals certain limitations and challenges acknowledged or implied by the authors:

- **Top-Down Dependency:** As a Top-down approach, RTMPose relies heavily on the quality of the accompanying object detector. Although RTMDet is highly effective, if the detector fails (i.e., fails to find the person), RTMPose receives no input to process. This is an inherent limitation compared to bottom-up methods.

- **Industrial Application Challenge:** The authors identify that the primary challenge is not merely achieving high scores on academic benchmarks, but performing robust and real-time multi-person pose estimation on devices with limited computing power. While RTMPose has made significant strides, maintaining high accuracy on extremely weak edge devices remains an ongoing open challenge for the field.