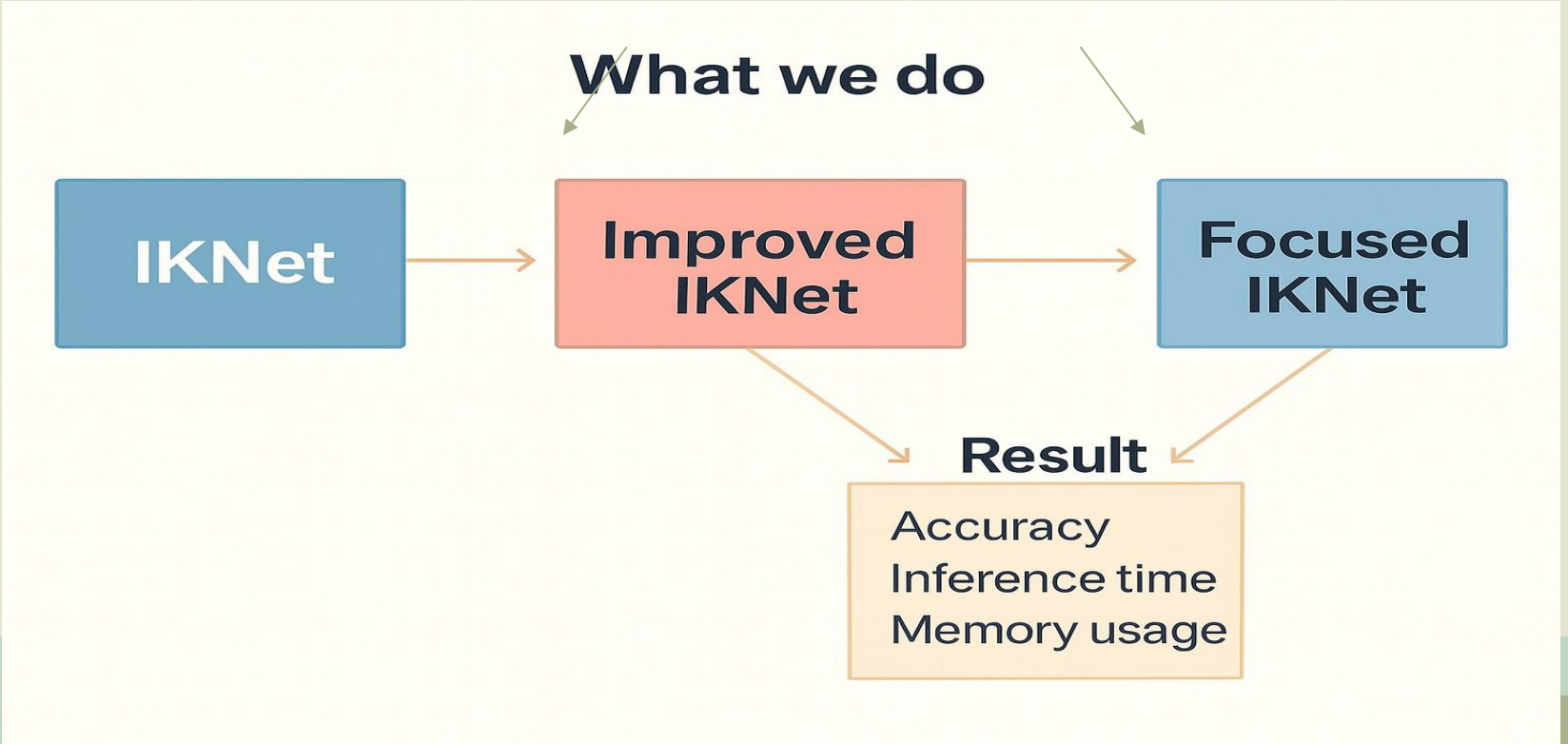
Inverse Kinematics Neural Network Models for Improving Inference Efficiency and Memory Usage

Sheng-Kai Chen | National Center for High-Performance Computing

Yi-Ling Tsai, Chun-Chih Chang | Electrical Engineering,Yuan Ze University

Yan-Chen Chen, Po-Chiang Lin

Outline


What we do
IKNet
Improved IKNet
Focused IKNet
Result
Accuracy
Inference time
Memory usage

Background

Traditional IK Methods

- Computationally expensive and inefficient
- Unstable for complex systems

Deep Learning Models

- Faster and flexible
- Need a GPU

Motivation

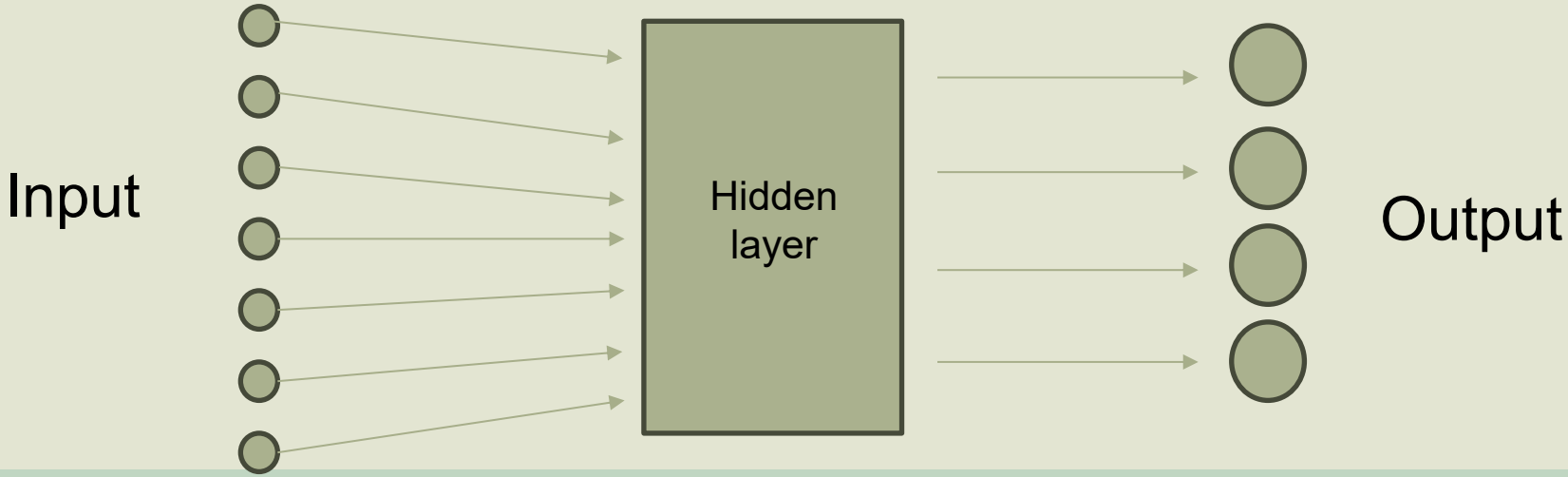
- CPU-only environments
- Reduce inference time and memory constraints

Original IKNet Overview

- Hurt performance
- Run slowly on the GPU and uses much memory

youtalk / iknet Public

NVIDIA open source



Improved IKNet

Differences Between Original IKNet and Improved IKNet

- Dropout **only after the input layer**
- **Kaiming Normal initialization** improves weight scaling
- **BatchNorm1d** improves convergence and stability
- **Residual Blocks** to avoid the vanishing gradient problem

Focused IKNet

Differences Between Original IKNet and Focused IKNet

- Separate position and orientation
- Each branch uses its layer
- Combine output to MLP

Position 3D (x, y, z)

Orientation 4D (qx, qy, qz, qw)

Experimental Setup

CPU

Intel Core i5-13500

GPU

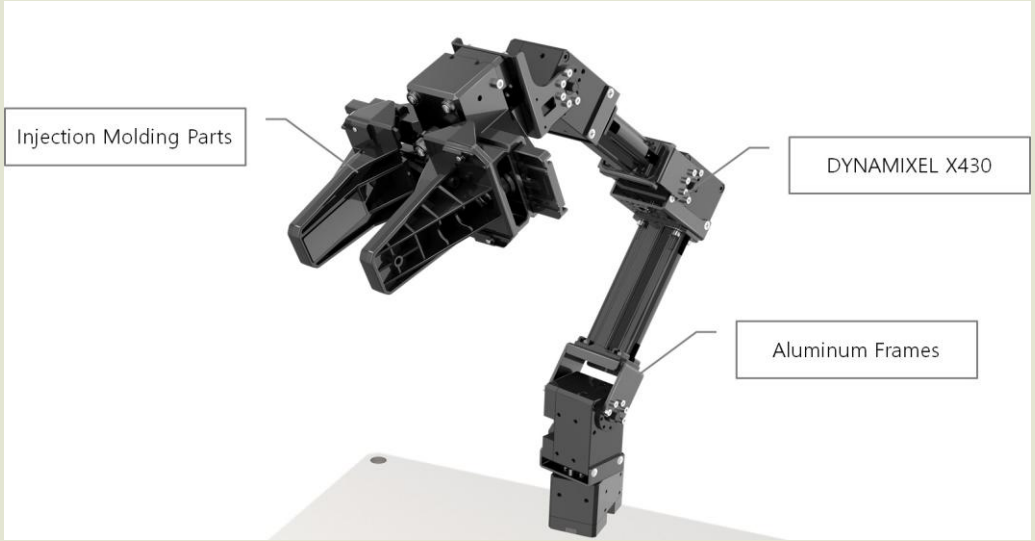
NVIDIA RTX 4060Ti

Memory

8 Gigabyte

Data collection

ROBOTIS Open Manipulator-X



Experimental Results
&
Analysis

Accuracy Comparison

Improved IKNet

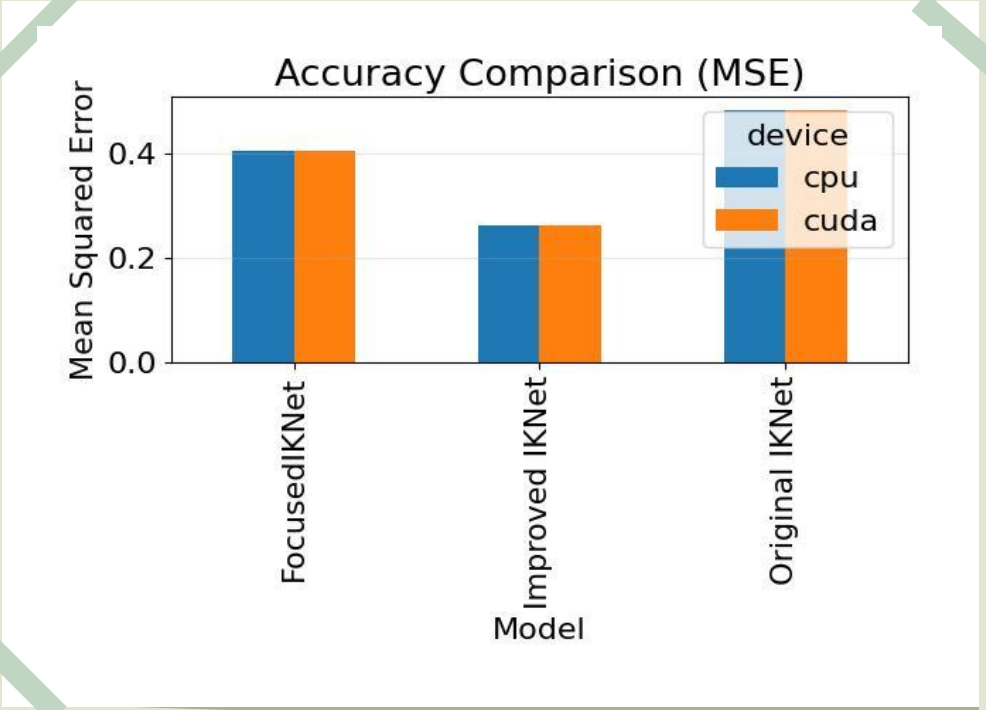
- Lowest MSE
- Predictive accuracy
- Model generalization

Focused IKNet

- Slightly higher MSE
- Dual-branch design offers flexibility
- Require further tuning

Original IKNet

- Worst performance on CPU



Inference Time Comparison

Improved IKNet

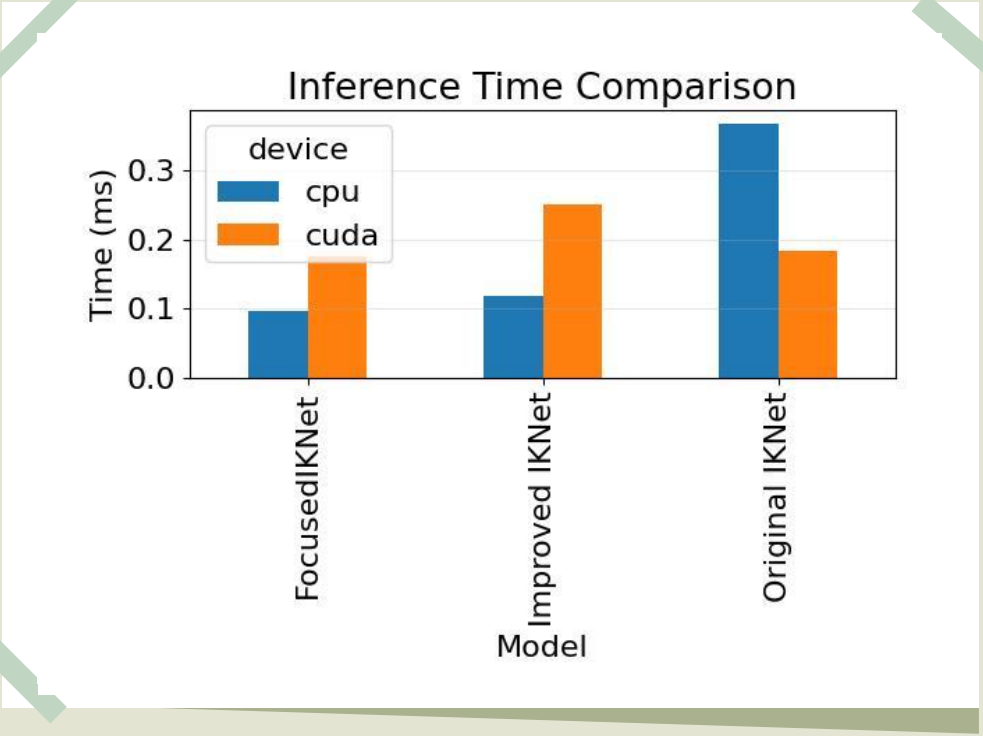
- Significantly faster on CPU
- Moderately faster on CUDA

Focused IKNet

- Shortest inference time
- Efficient execution speed

Original IKNet

- Slowest inference time on CPU
- Poor efficiency for real-time systems



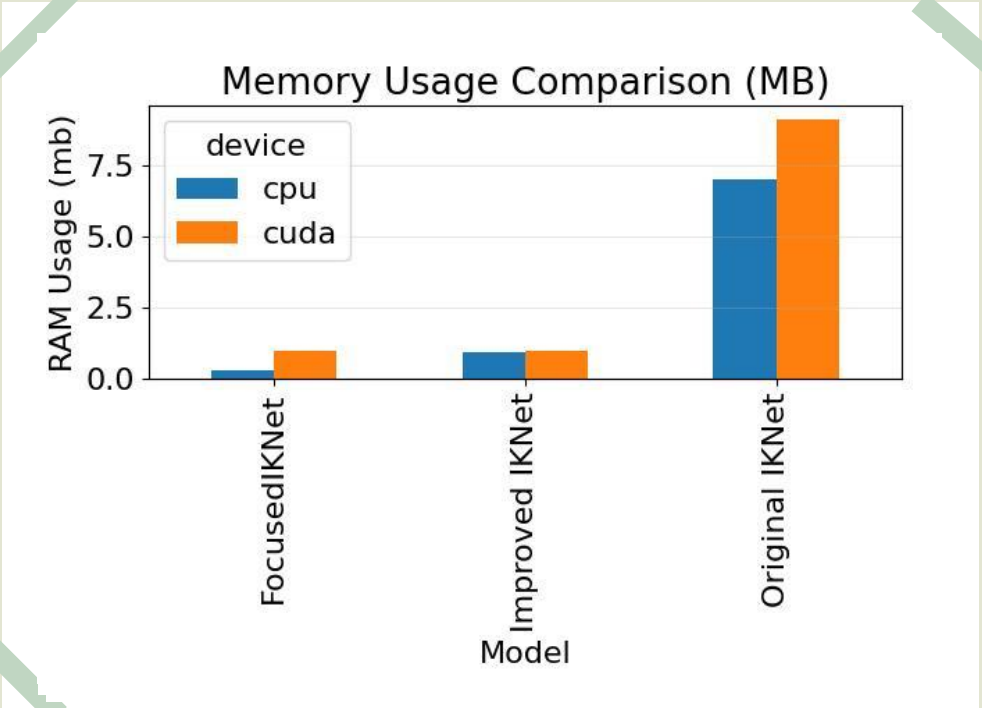
Memory Usage Comparison

Focused & Improved IKNet

- Less usage on both CPU and CUDA
- Highly optimized for efficient memory use
- Good accuracy and speed

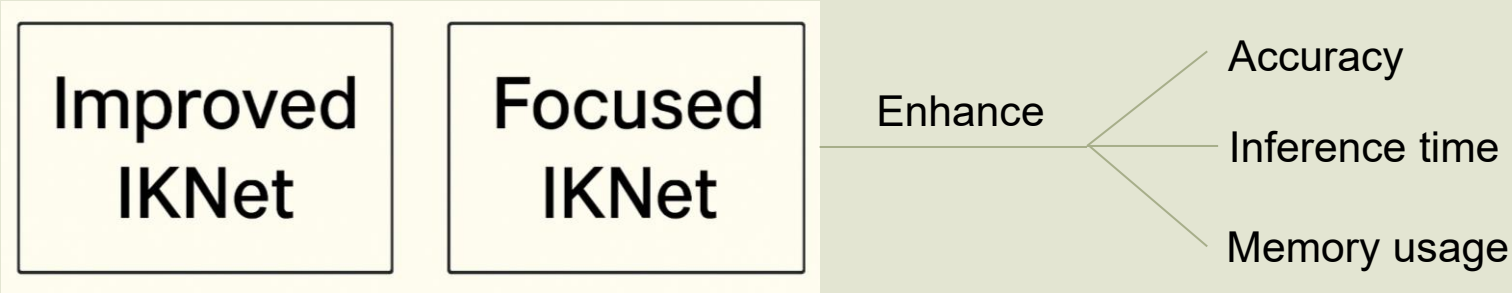
Original IKNet

- Larger model size
- Higher computational cost
- Unsuitable for memory-constrained embedded systems



Contributions

- CPU-only environments
- Enhance accuracy, inference time, and memory usage



CPU is better than GPU

Conclusion

Accuracy Comparison

- Most accurate model
- Well-suited for embedded CPU-only systems

Inference Time Comparison

- Ideal for low-latency applications **(Improved IKNet)**
- Offers a good trade-off between speed and accuracy **(Focus IKNet)**
- Not suitable for real-time CPU-based deployment **(Original IKNet)**

Memory Usage Comparison

- Low-memory environments **(Improved & Focused IKNet)**
- Resource-intensive **(Original IKNet)**

References

1.Deep-Neural-Network-for-Solving-the-Inverse-Kinematics Problem [Online] Available:
https://github.com/OmarJItani/Deep Neural-Network-for-Solving-the-Inverse-Kinematics Problem/blob/main/README.md. (accessed in Feb 2025)

2.IKNet: Inverse kinematics neural networks [Online] Available:
https://github.com/youtalk/IKNet?tab=readme-ov-file. (accessed in Feb 2025)

3.CUDA[Online] Available:
https://developer.nvidia.com/cuda-toolkit (ac cessed in Feb 2025)

4.Pytorch [Online] Available:
https://pytorch.org/ (accessed in Feb 2025)

Thank you for your listening!

Sheng-Kai Chen

2503100@niar.org.tw

Yi-Ling Tsai

yiling930910@gmail.com

Chen-Yan Chen

hannahchen0619@gmail.com

Chun-Chih Chang

Justin.chang93@gmail.com