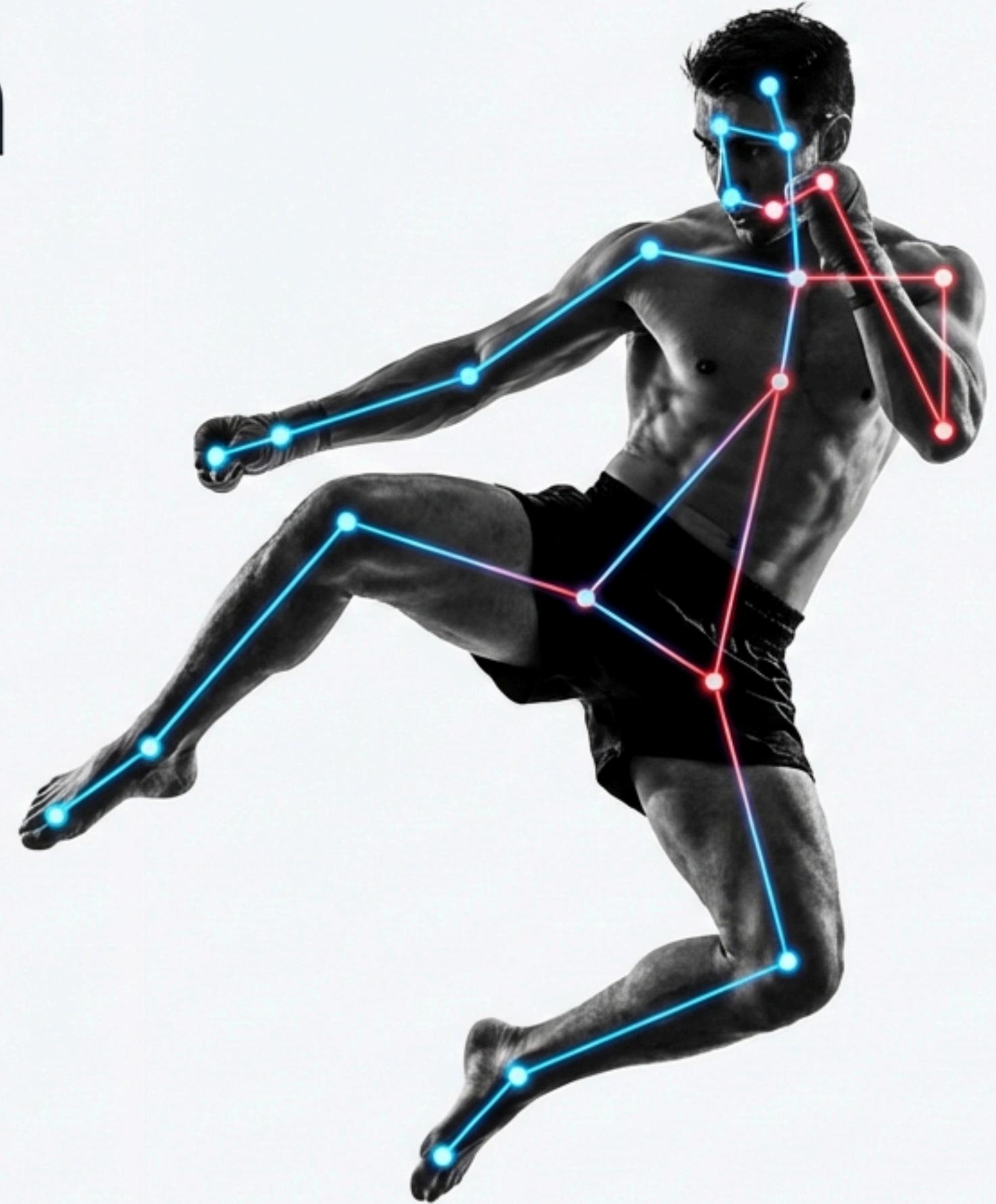


Muay Thai Action Recognition: An **ST-GCN Pipeline**



MUAY THAI STANCES



Mek Khara Lor Kaew



Kum Pa Gun
Poong Hork



Narai Kwang Jug



Hoang Hearn



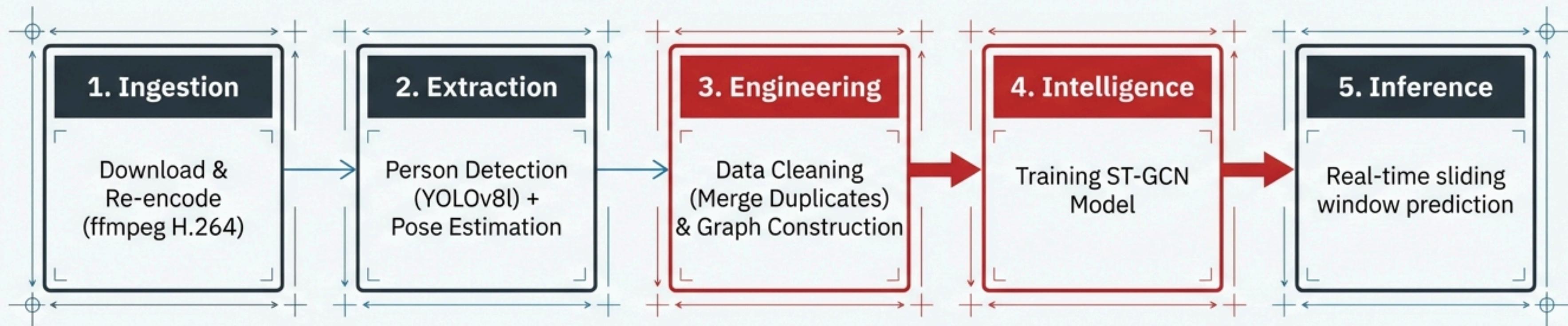
Phra Ram Phang Sorn



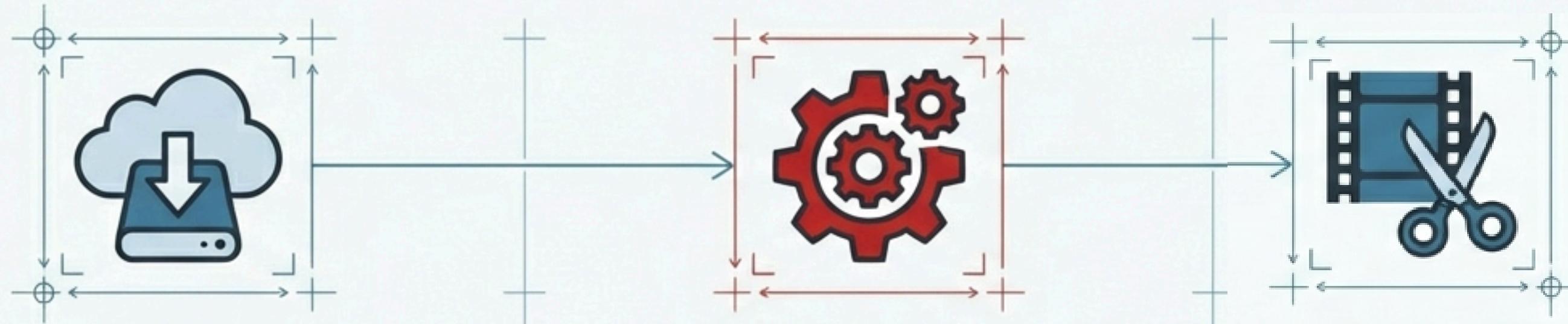
Tad Mai Khom Nam

จำแนกท่ามวยไทย 6 ท่า: Pose Classification

สรุปภาพรวมระบบ (System Architecture)



การเตรียมข้อมูล: จาก YouTube สู่ Standardized Clips



Tools: pytube fix, ffmpeg

Source: High Resolution
(1080p)

Re-encoding: แปลงไฟล์เป็น format มาตรฐาน H.264/AAC เพื่อความเสถียรในการประมวลผลบน OpenCV/Colab

```
segments = [  
    ("Mek_Khara_Lor_Kaew", "00:00:00", "00:00:15"),  
    ("Kum_Pa_Gun_Poong_Hork", "00:00:15", "00:00:41"),  
    ("Narai_Kwang_Jug", "00:00:41", "00:01:08"),  
    ("Hoang_Hearn", "00:01:08", "00:01:32"),  
    ("Phra_Ram_Pheng_Sorn", "00:01:32", "00:01:51"),  
    ("Tad_Mai_Khom_Nam", "00:01:51", "00:02:02"),  
]
```

Segmentation

Action Splitting: ตัดแบ่งคลิปตาม Timestamp ของท่ามวย 6 คลาส

Classes: Mek_Khara_Lor_Kaew, Tad_Mai_Khom_Nam, etc.

Goal: Video-level labeling

Key Insight

Why it matters: ลดความแปรปรวนของ Encoding และตัดส่วนที่ไม่เกี่ยวข้องออก (Noise Reduction at source)

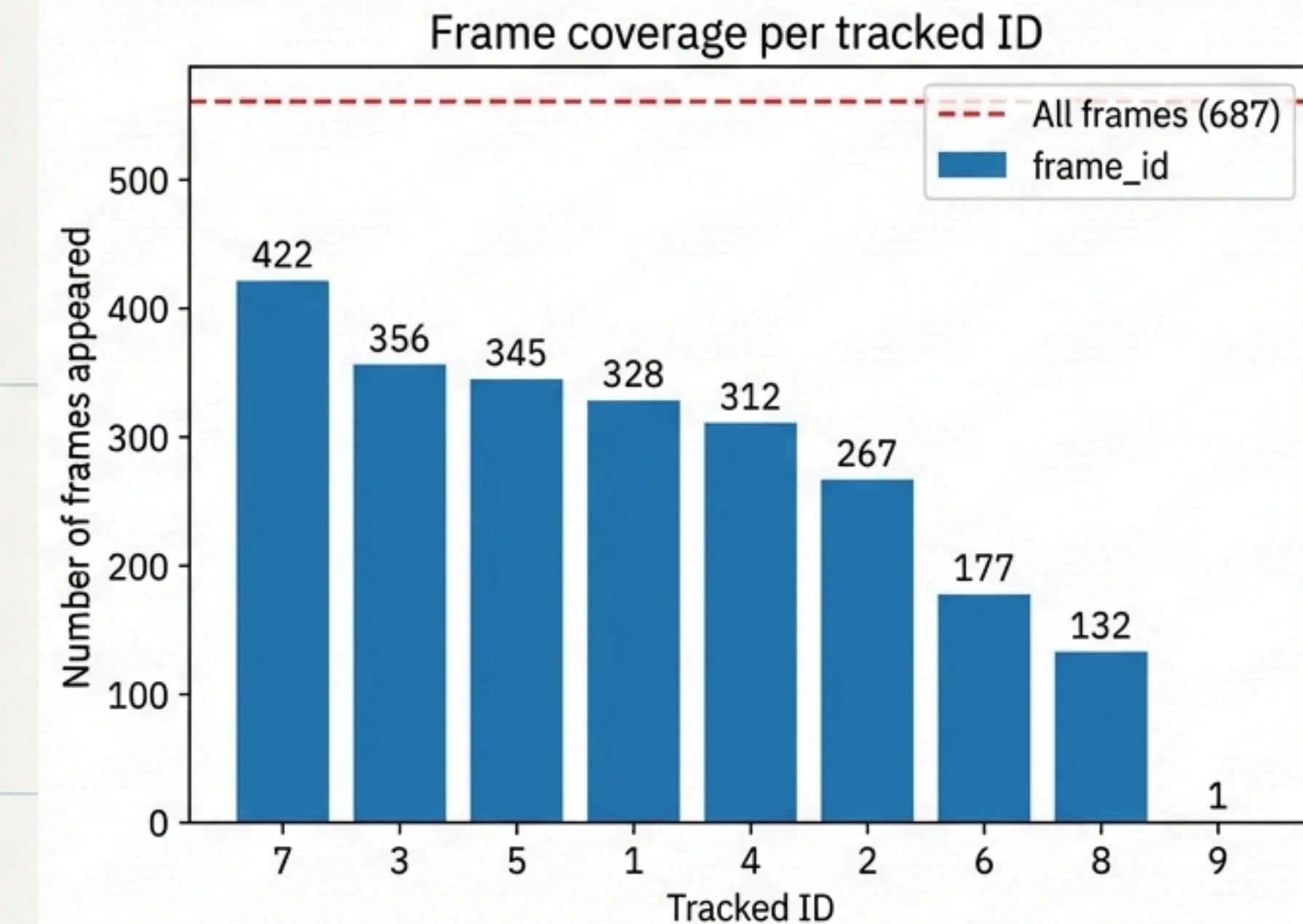
การระบุตัวตนและติดตามบุคคล (Detection & Tracking)

Model: YOLOv8I (Detect `person`)

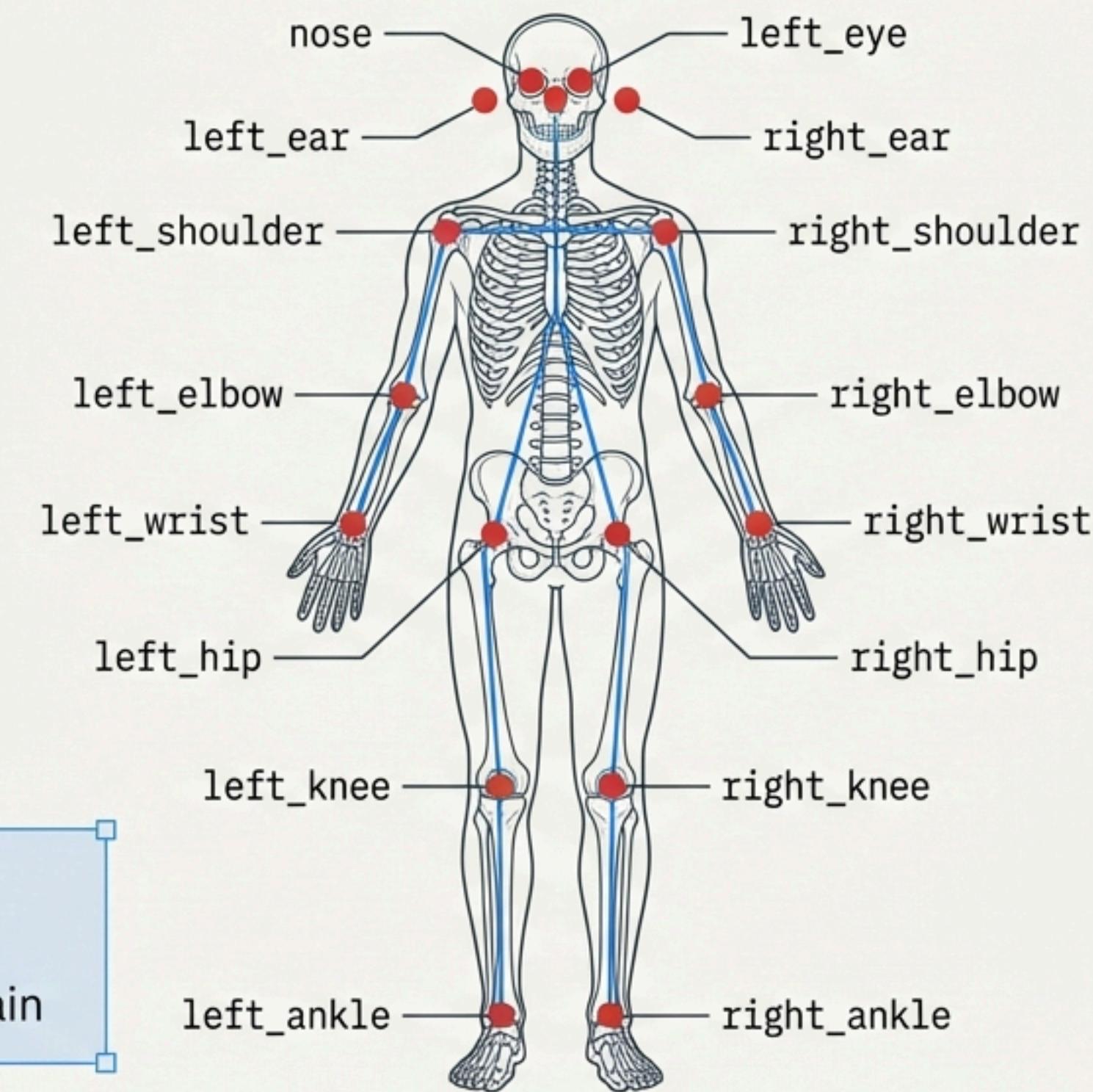
Tracker: ByteTrack
(`persist=True`)

Goal: Identity Consistency
(รู้ว่าคนไหนคือคนเดิม)

Output: `track_data.csv`



การแปลงร่างกายเป็นข้อมูลพิกัด (Pose Estimation)



Observation:

Head Keypoints (Eyes/Ears)

มักloyตัวแยกอิสระจาก Body Chain

Model: YOLOv8I-pose
(COCO 17 Keypoints)

Format:
(x , y , confidence_score)

ปัญหาที่พบ: Duplicate Detections & Noise

ไม่ใช่ ID Switching แต่เป็น Duplicate Full-Body Detection

The Symptom

Person id per each frame:

- ไม่เก่ากับ 9 คน = 146 frames
- ครบ 9 คน = 665 frames
- ไม่มี id หลัก = 0 frames

merge →

Person id per each frame:

- ไม่เก่ากับ 9 คน = 8 frames
- ครบ 9 คน = 803 frames
- ไม่มี id หลัก = 0 frames

detect lifespan →

Person ID	Frames
0	881
1	881
2	881
3	881
4	881
5	881
6	881
7	881
8	881
9	7
10	1

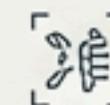
merge →

Person id per each frame:

- ไม่เก่ากับ 9 คน = 0 frames
- ครบ 9 คน = 811 frames
- ไม่มี id หลัก = 0 frames

The Cause

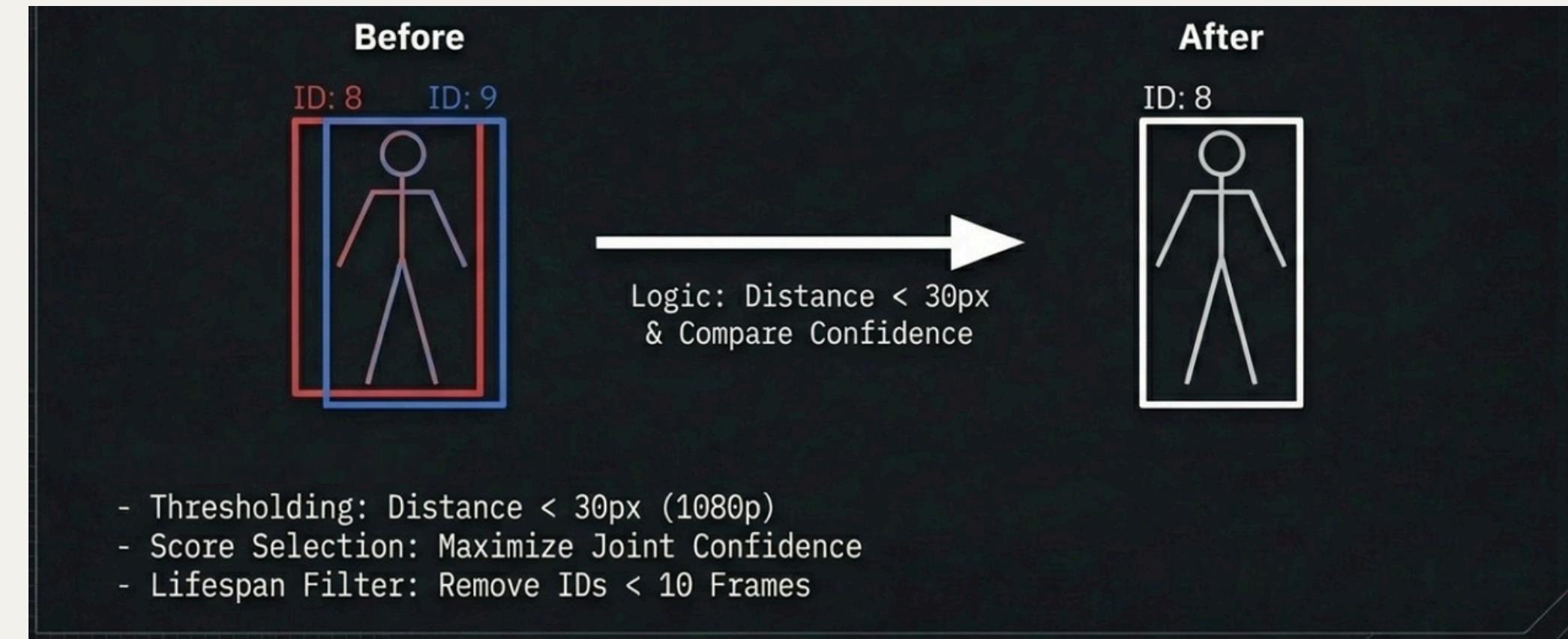
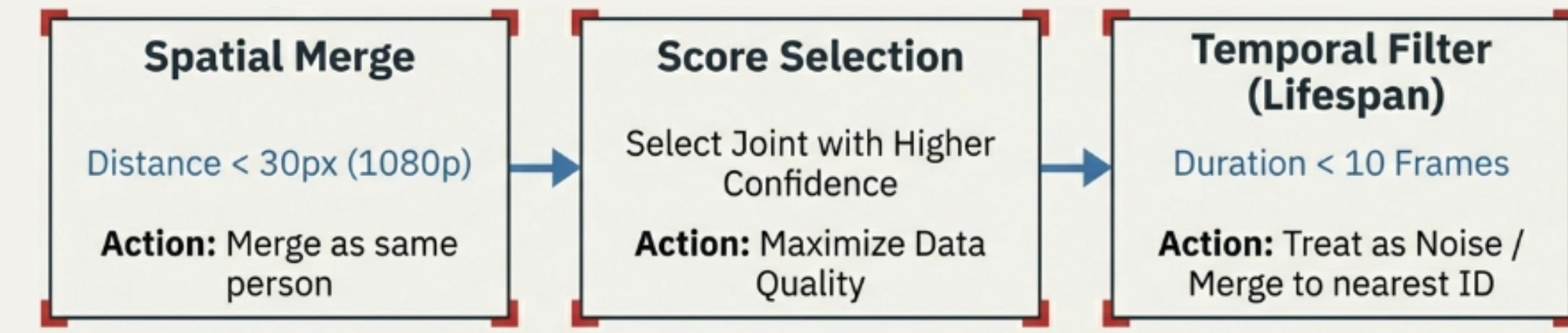
Scenario: โมเดลตรวจจับคนเดิมซ้ำเป็น 2 Bounding Boxes ที่ซ้อนกับกัน ทำให้เกิด Track ID เกินมา



Impact: หากนำข้อมูลนี้ไป tren กับที่ โมเดลจะสับสนและเรียนรู้ Pattern ที่ผิดพลาด (**Noise**)

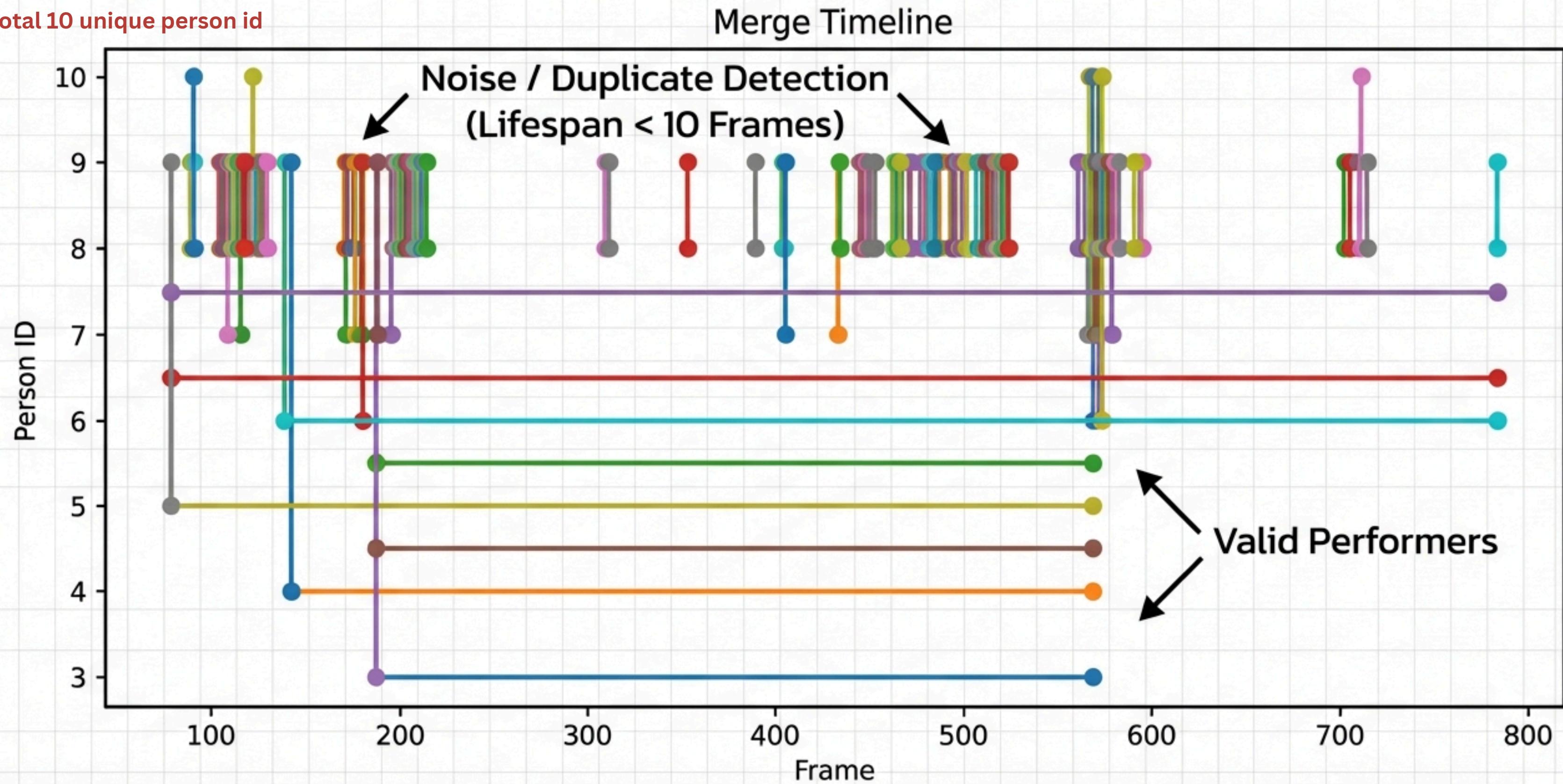


กลยุทธ์การทำความสะอาดข้อมูล (Data Cleaning Strategy)

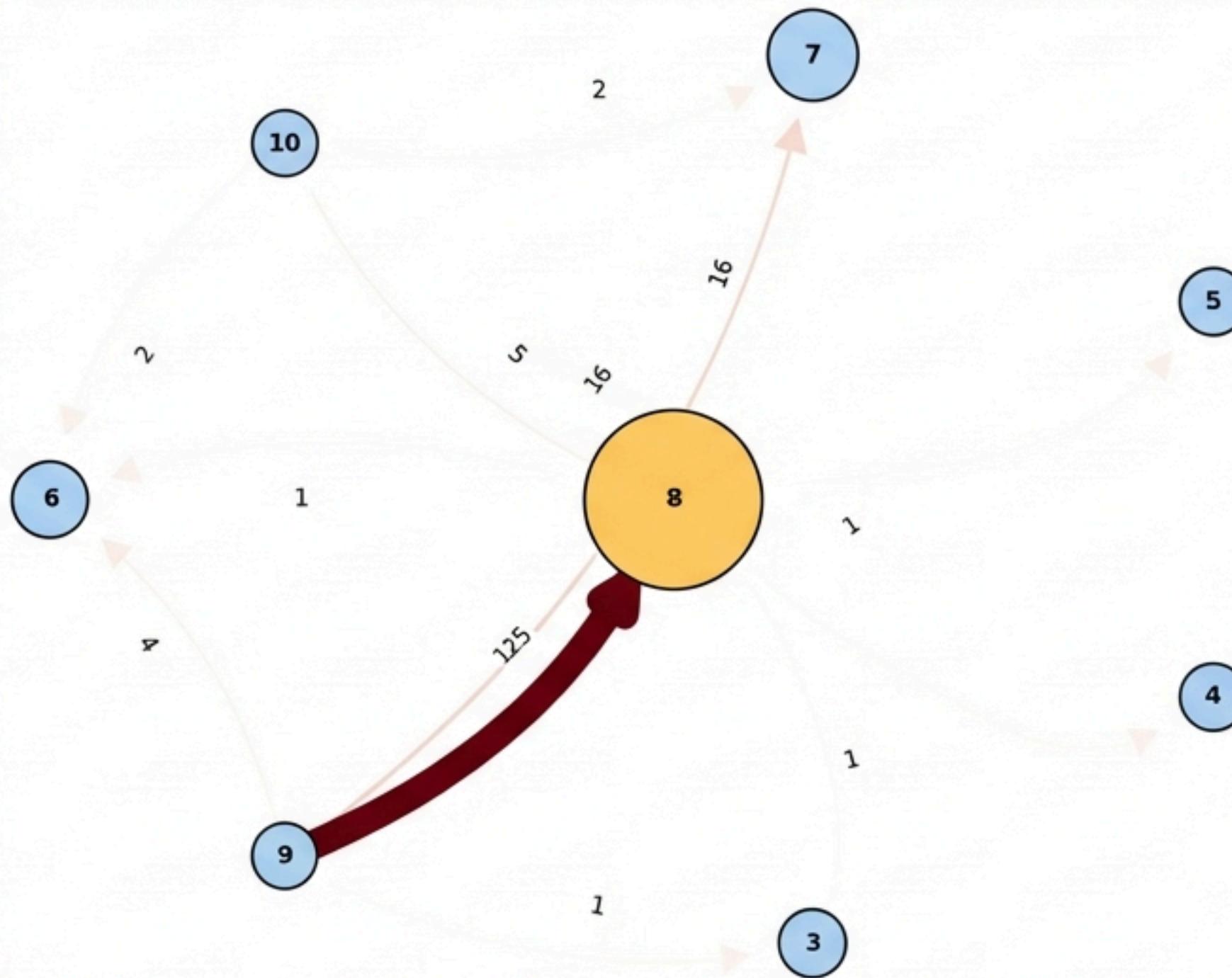


ພາລັບຮ່ວມການຮັບ ID (ID Merging Logic)

total 10 unique person id



ผลลัพธ์การรวม ID (ID Merging Logic)



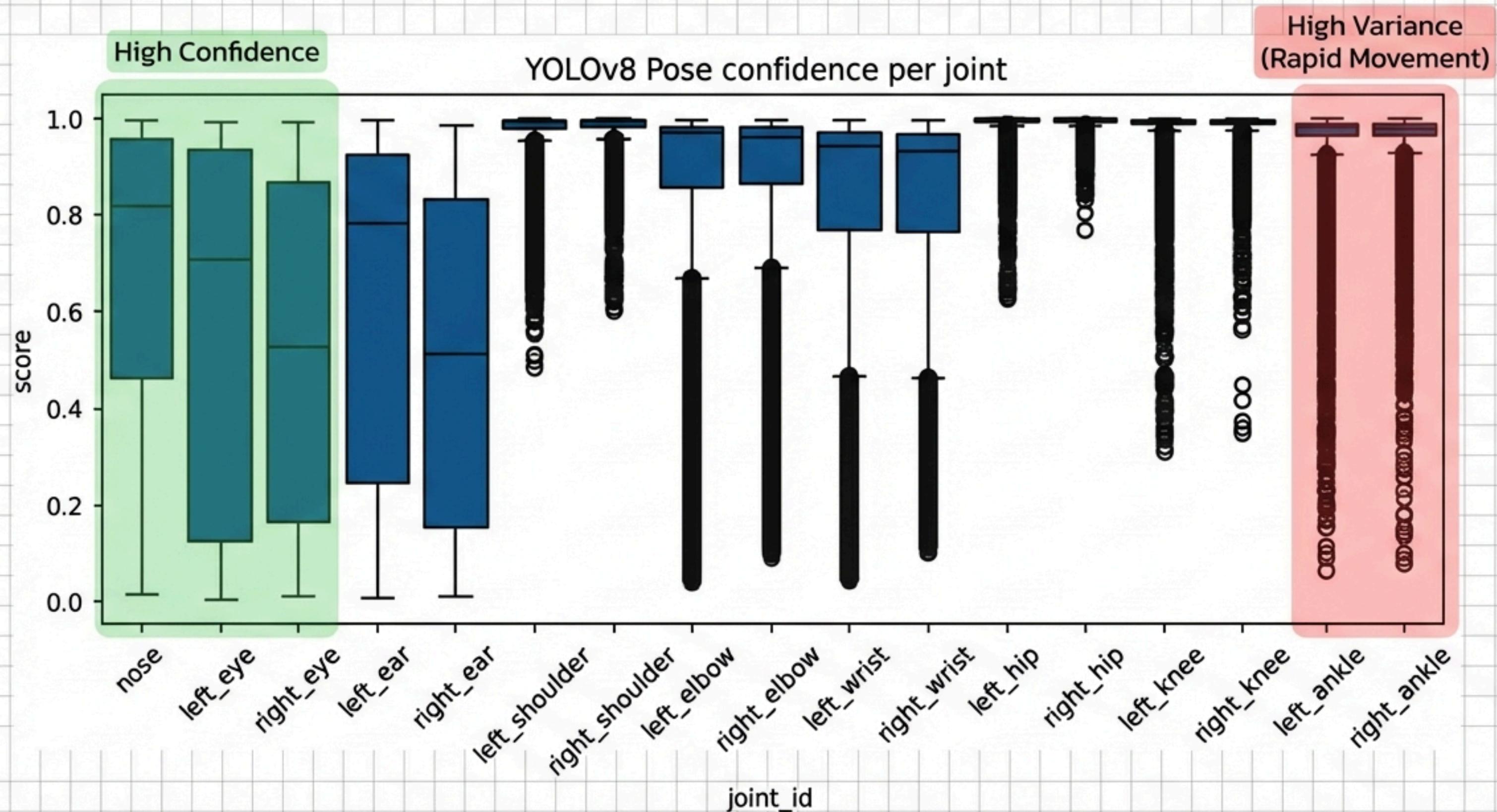
Logic:

IF ID Lifespan < 10 Frames →
Merge to Nearest Stable ID

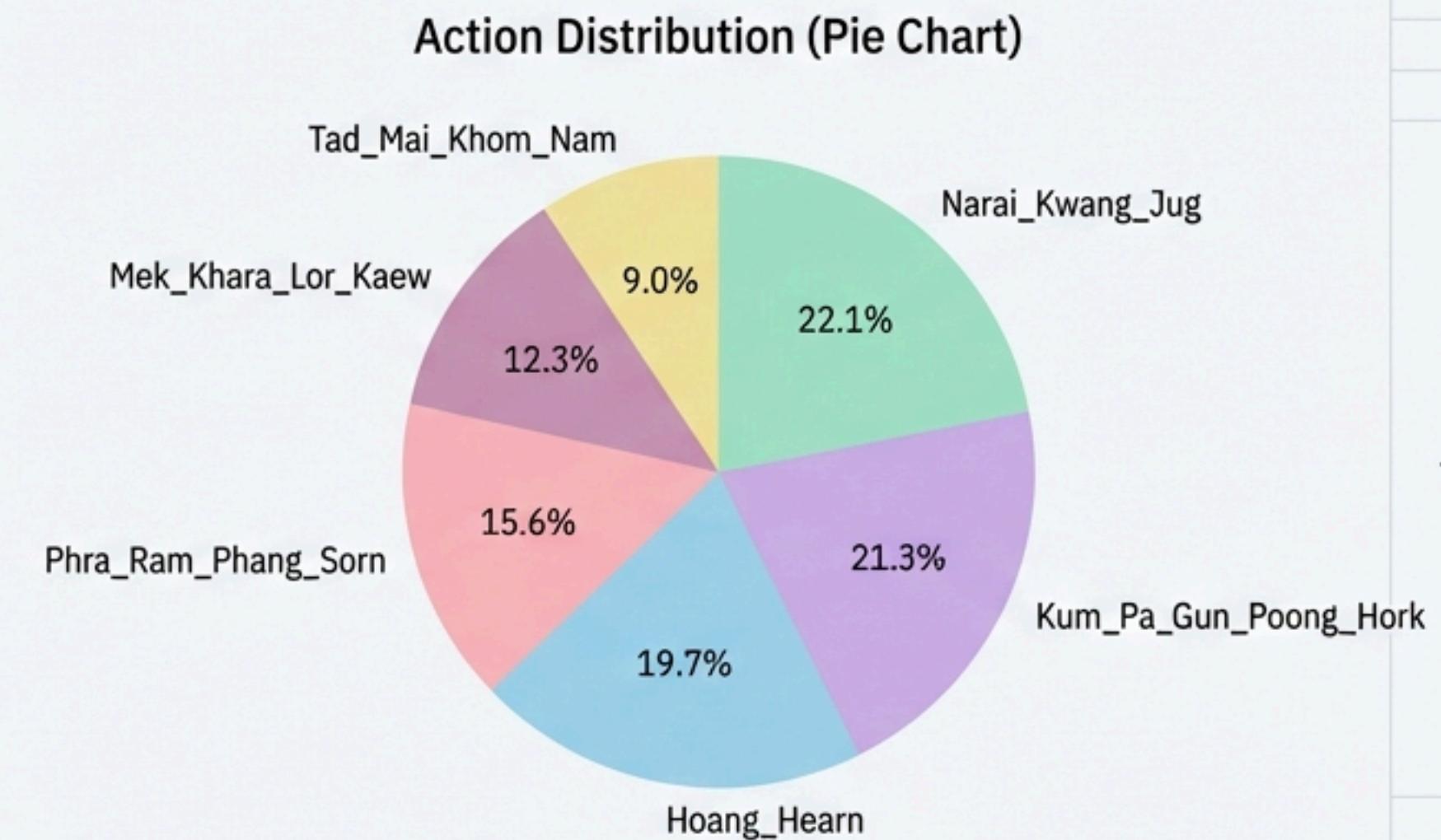
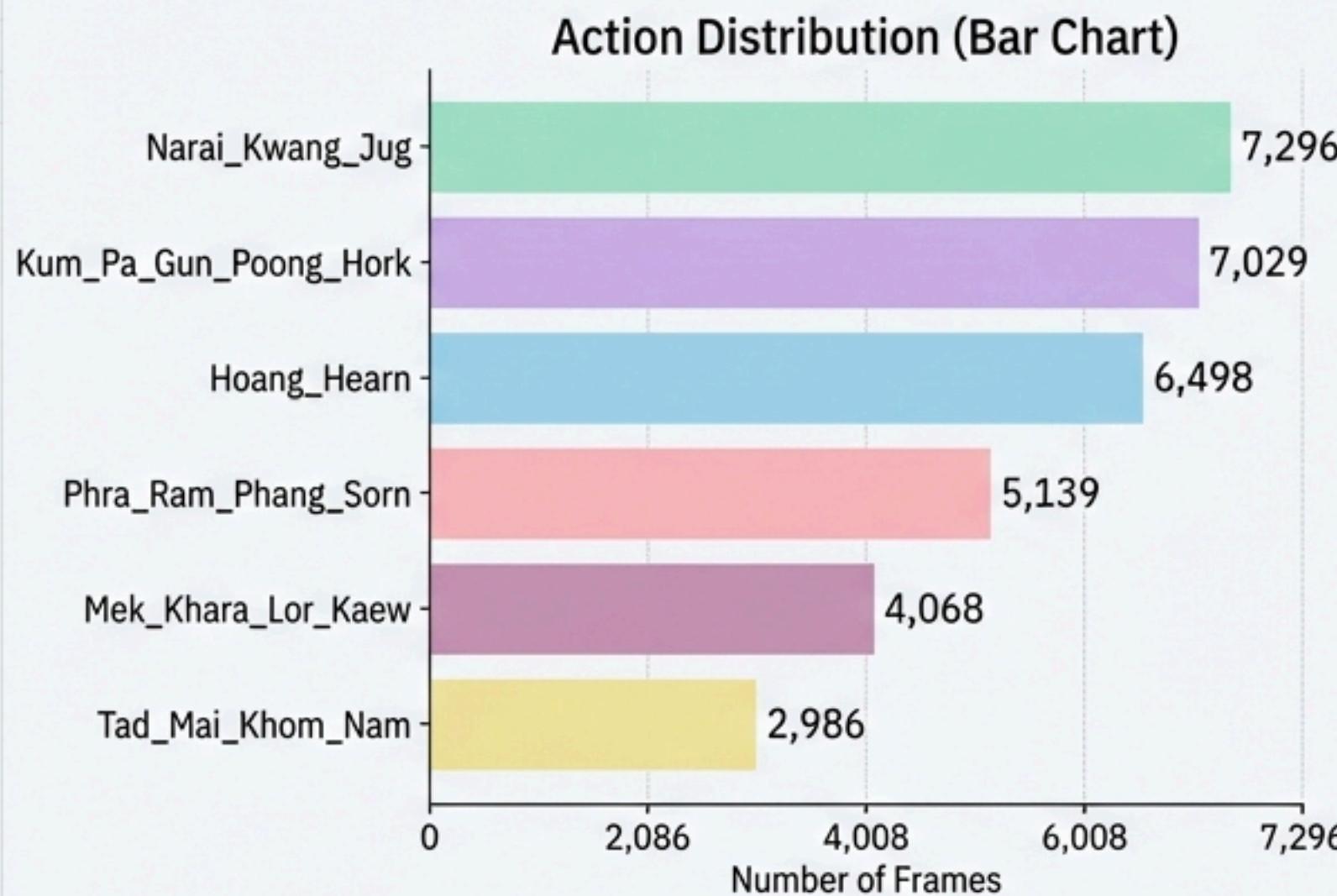
Result:

63 IDs → 9 True IDs
Temporal Consistency Achieved

การวิเคราะห์ความเชื่อมั่นของ Keypoints (Pose Confidence Analysis)



การกระจายตัวของข้อมูล (Data Distribution)

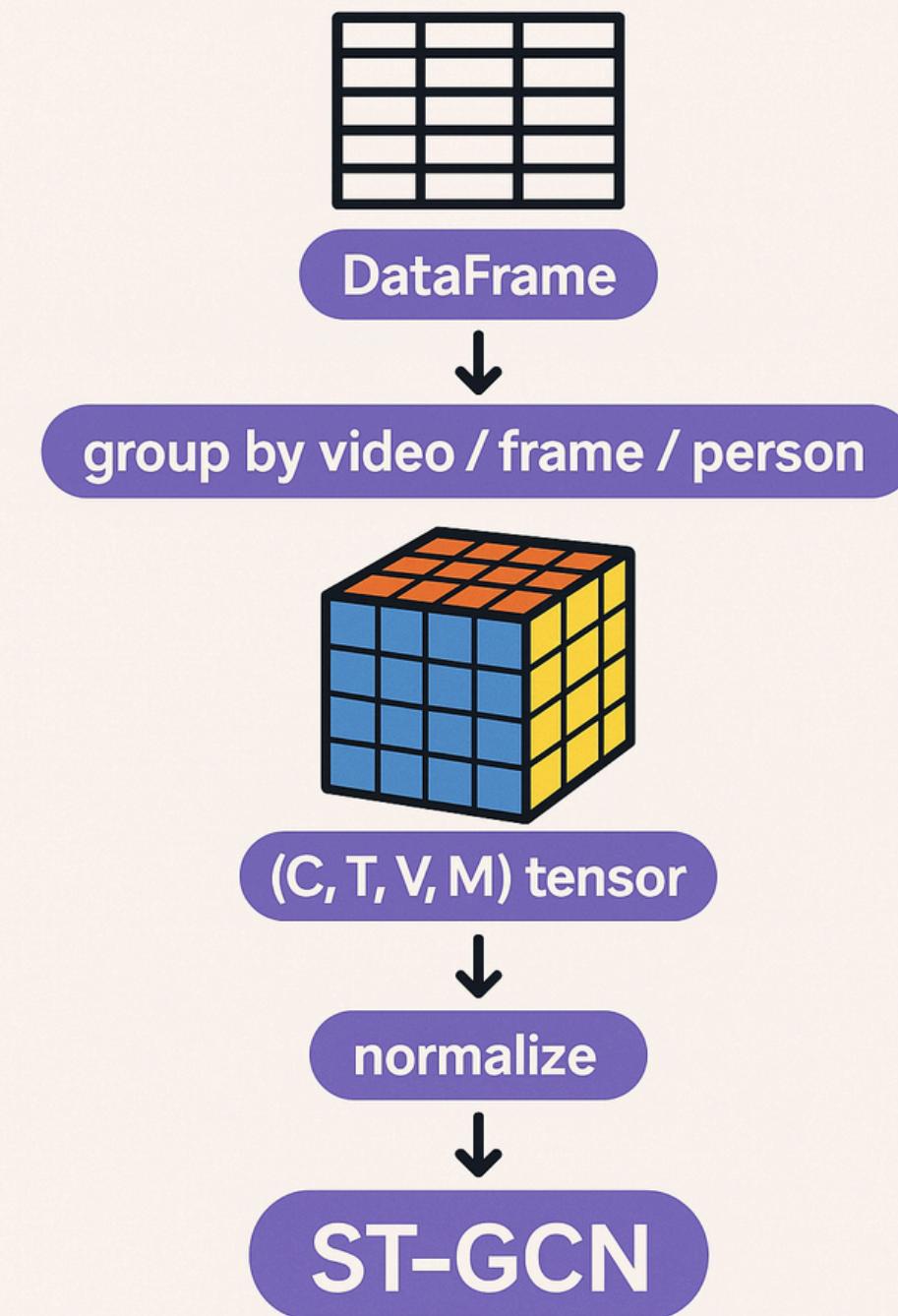
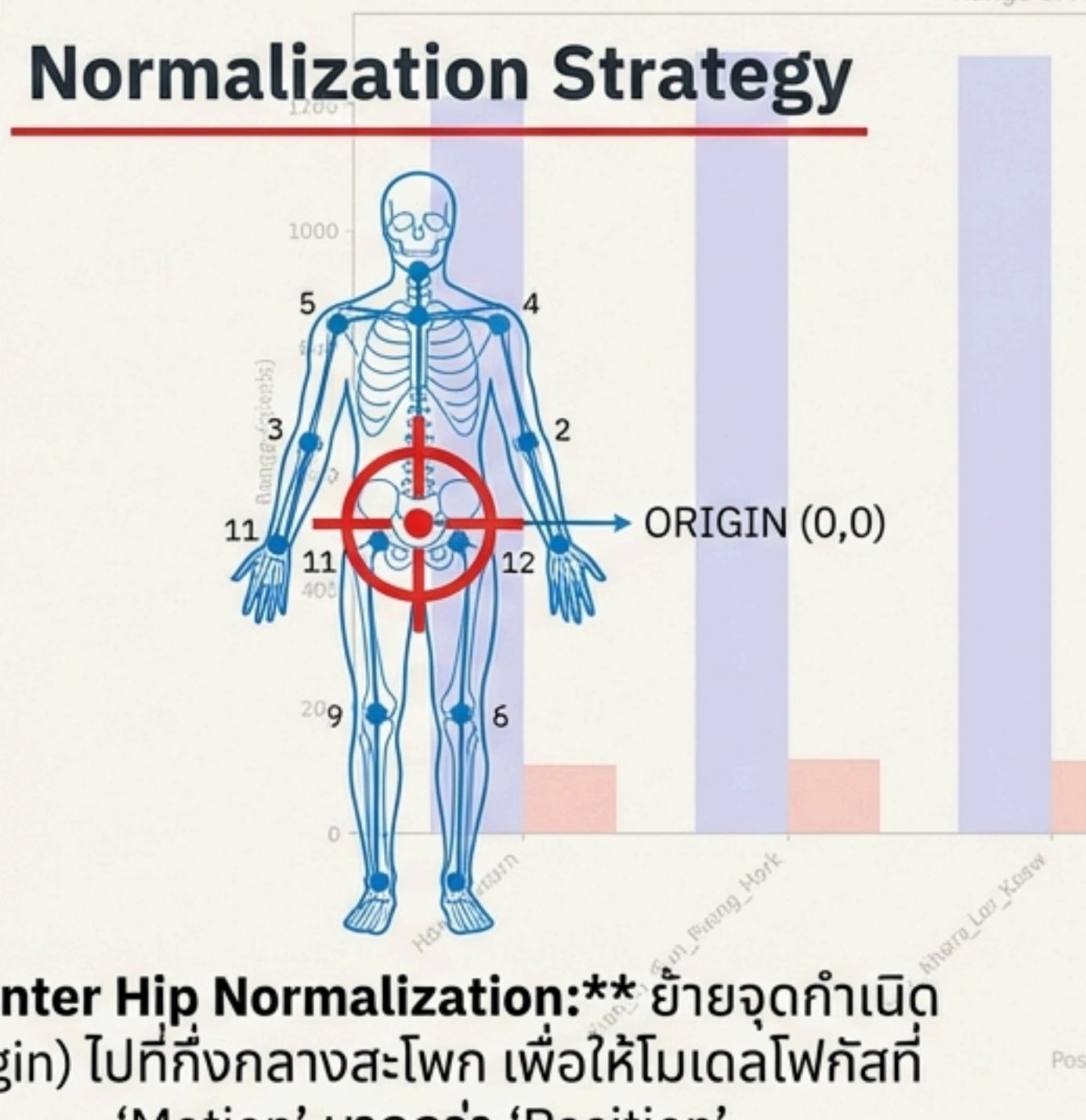


▶ **Imbalanced Dataset:**
ข้อมูลไม่ได้สมดุล 100%

▶ **Solution:** ใช้เทคนิค Label Smoothing
และ Weighted Loss ป้องกัน Bias

การสกัดข้อมูลและ Normalization (Feature Engineering)

Precision Engineering meets Cultural Heritage



การสร้างกราฟ Spatial-Temporal (Graph Construction)

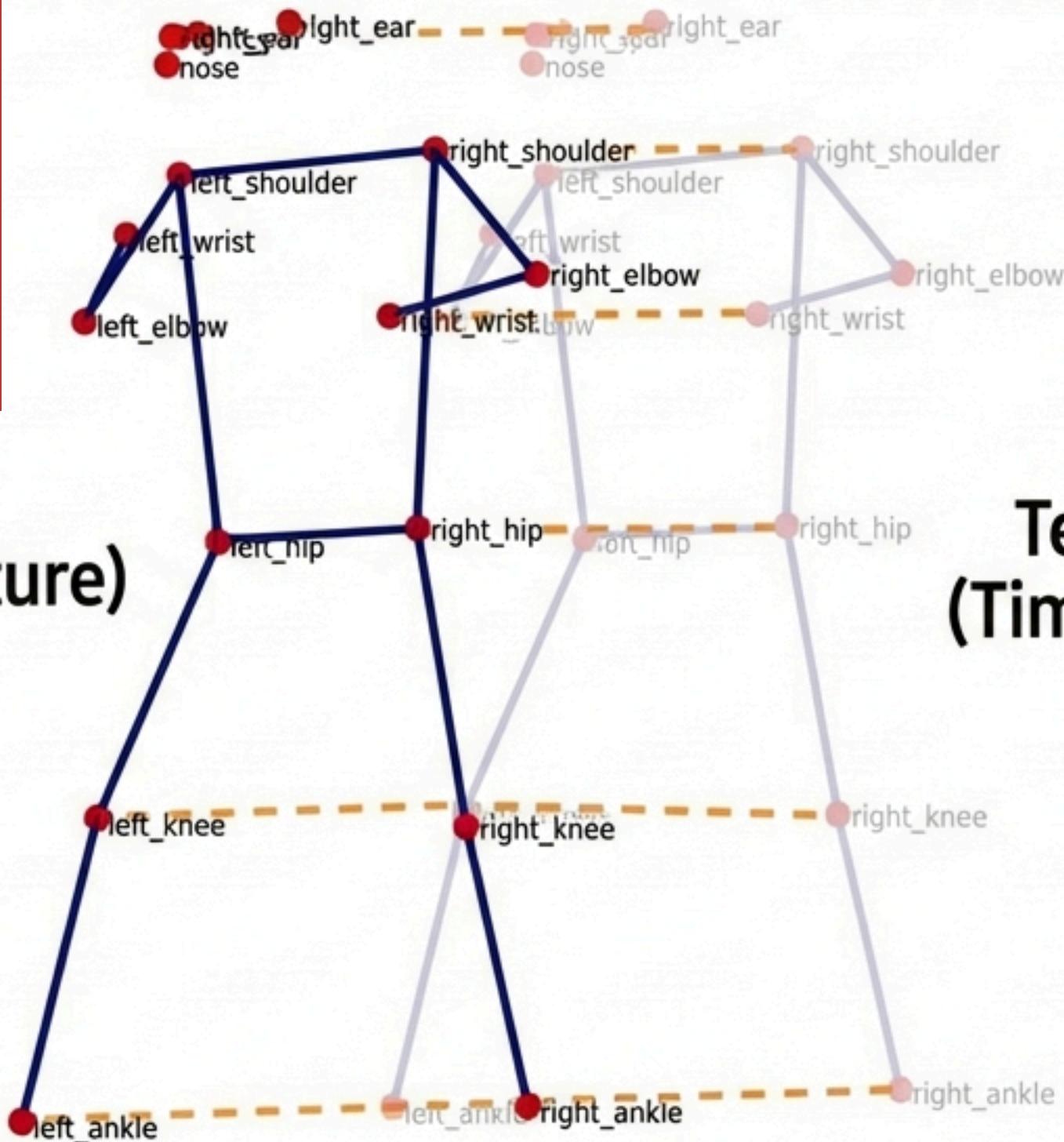
◆ Spatial (Body Structure)

ในแต่ละเฟรม ร่างกายถูกแทนเป็น กราฟของข้อต่อ (joints)

- จุด (node) = joint เช่น shoulder, elbow, knee
- เส้น (edge) = ความเชื่อมโยงตามโครงสร้างร่างกายจริง

Graph Convolution จะกระจายข้อมูลระหว่าง joint ที่เชื่อมกัน
→ ทำให้ไม่เดลへ้าใจ “ก่าก้าง ณ ขณะหนึ่ง” โดยอิงโครงสร้าง
ร่างกาย

Spatial (Body Structure)



Temporal (Time t to t+1)

◆ Temporal (Time t → t+1)

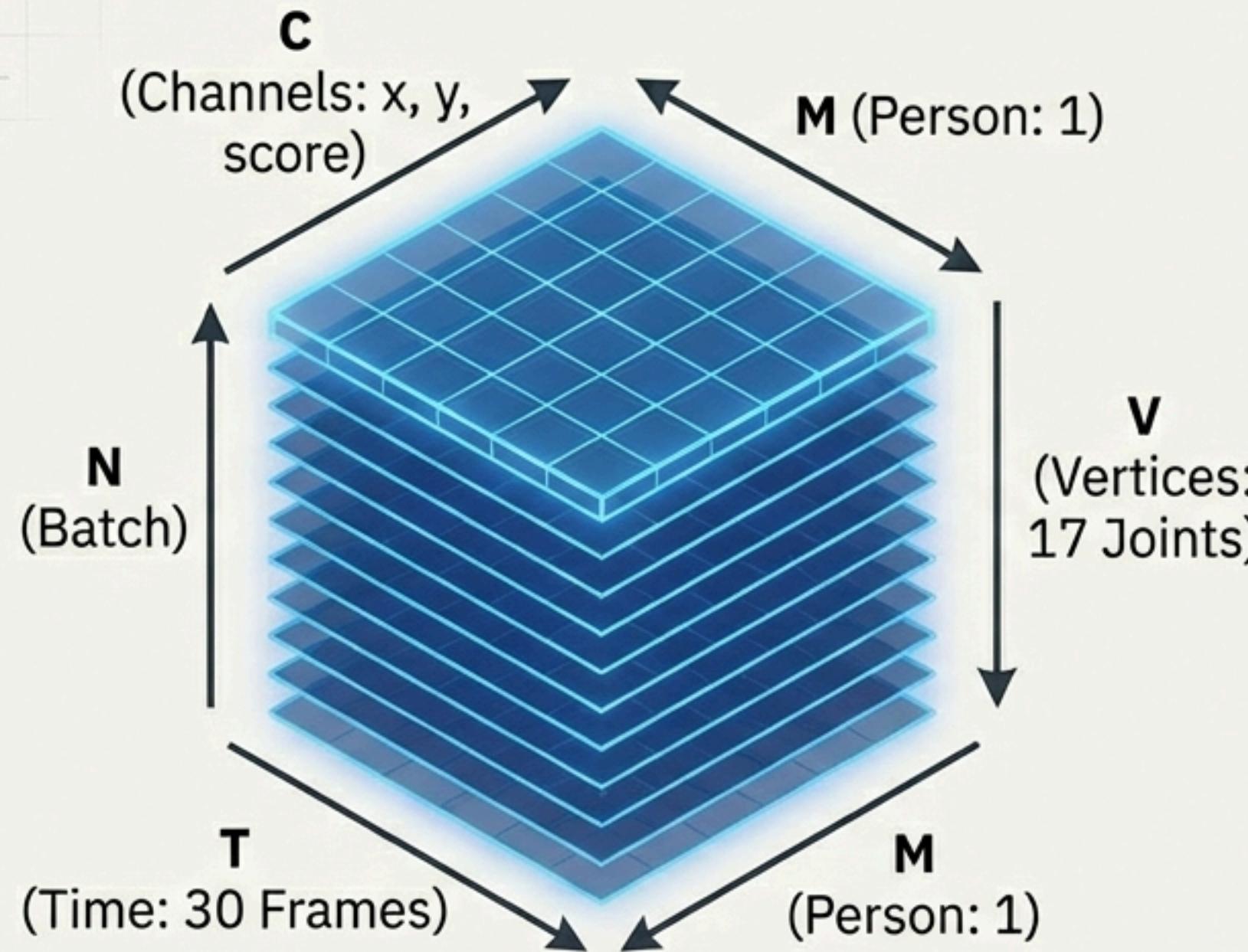
ในมิติของเวลา joint เดิมจะถูกเชื่อมต่อข้ามเฟรม

- joint เดียวกันที่เวลา t และ t+1 ถูกเชื่อมด้วย temporal edge

Temporal Convolution จะมองลำดับเฟรมต่อเนื่อง
→ เรียนรู้ “การเคลื่อนไหว” ไม่ใช่แค่ pose เดียว ๆ

ST-GCN เรียนรู้โครงสร้างร่างกาย
ผ่าน Spatial Graph และเรียนรู้การ
เคลื่อนไหวผ่าน Temporal
Convolution โดยใช้ลำดับของเฟรม
เป็นตัวแทนของเวลา

การสร้างกราฟสำหรับ ST-GCN (Graph Construction)

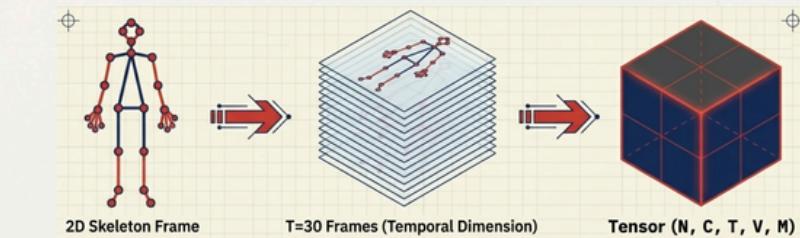


ดึงข้อมูลตั้งแต่
start_frame → start_frame + T - 1 (T = 30)
start, start+1, ..., start+29 (รวม 30 เฟรมติดกัน)

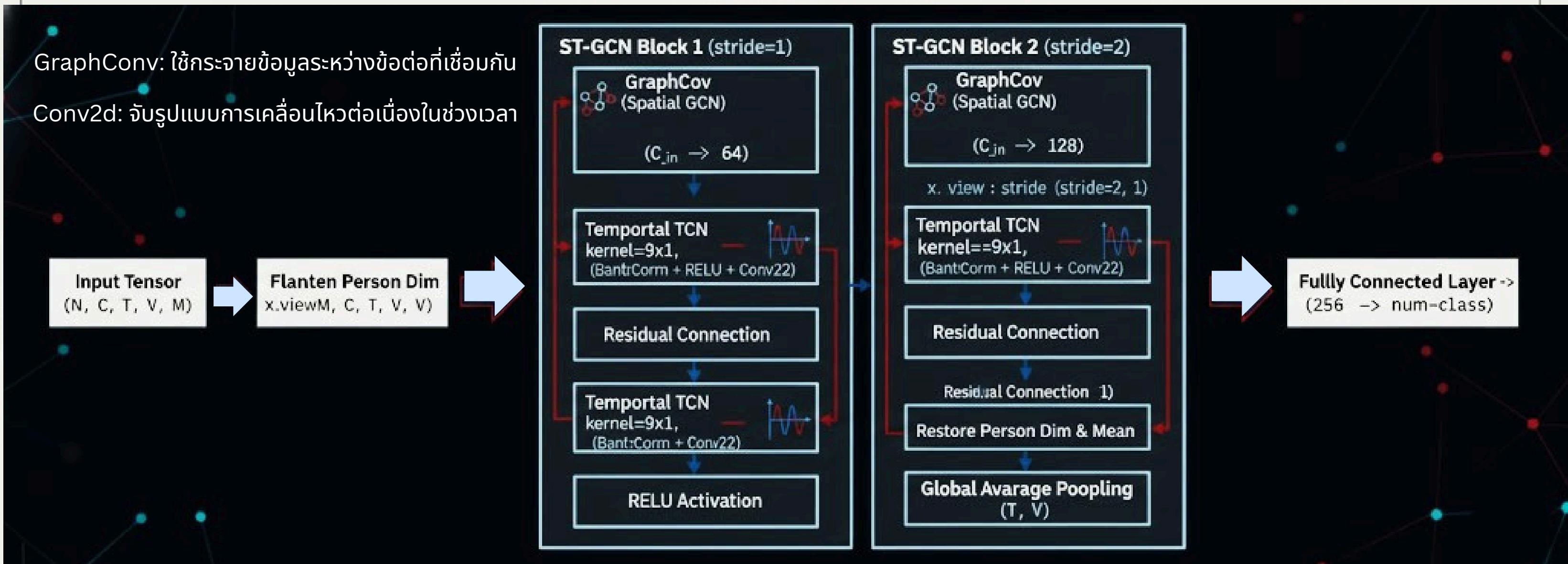
Graph Components

- **Nodes:** 17 Body Joints
- **Edges:** Physical Bones + Self-loops
- **Adjacency Matrix ($A\$$):**
 A_{self} (Root),
 A_{in} (Centripetal),
 A_{out} (Centrifugal)

IBM Power Monit



สถาปัตยกรรมโมเดล (ST-GCN Architecture)



- **Spatial:** GraphConv ใช้กระจายข้อมูลระหว่างข้อต่อที่เชื่อมกัน (โครงสร้างเชิงพื้นที่)
- **Temporal:** Conv2d (Kernel 7x1) จับรูปแบบการเคลื่อนไหวต่อเนื่อง (ความต่อเนื่องตามเวลา)
- **Structure:** 3 Layers หลัก ($64 \rightarrow 128 \rightarrow 256$ channels)
- **Residual:** มีการเชื่อมต่อแบบ Residual Link เพื่อช่วยการไหลเวียนของ Gradient

เทคนิคการเทรนโมเดล (Training Strategy)



Configuration

Loss: CrossEntropy +
Label Smoothing (0.1)

Optimizer: Adam
(LR=2e-4)



Controls

Gradient Clipping:
`norm=1.0` (Prevent explosion)

Scheduler:
`ReduceLROnPlateau`



Result

Early Stopping:
Patience = 10

Best Model: Epoch 24
Prevents Overfitting
effectively

Remark:

STGCNDataset

- ✓ `__getitem__` คืน 1 sequence ต่อ 1 sample
- ✓ ไม่มีการยุงกับ time dimension (T)
- ✓ ลำดับเฟรมภายใน sample ถูกเก็บไว้ครบ
- ✓ `DataLoader(shuffle=True)` จะ shuffle แค่ idx → สลับระดับ sample เท่านั้น

การวิเคราะห์การเปลี่ยนแปลงของโมเดล ST-GCN (Version 1-5)

สรุปผลลัพธ์

Version	Accuracy	F1-Score	Acc Δ	F1 Δ	การเปลี่ยนแปลงหลัก
V1	66.39%	64.71%	-	-	Baseline (No regularization)
V2	79.83% ★	78.48%	+20.25%	+21.28%	+ Dropout + Weight Decay
V3	75.63%	73.88%	+13.92%	+14.18%	ปรับ Dropout rates สูงขึ้น
V4	79.55%	78.51% ★	+19.83%	+21.33%	Fine-tune normalization & hyperparameters
V5	77.03%	75.53%	+16.03%	+16.73%	+ Noise Aug, ลด weight decay

🏆 Version 4 คือ Best Model เพราะ:

- ✓ F1-score สูงสุด: 78.51% (+21.33% จาก baseline)
- ✓ Balanced performance: ดีกับทุก class (สำคัญสำหรับ imbalanced data)
- ✓ Accuracy ใกล้เคียง V2: แต่ F1 ดีกว่า
- ✓ เหมาะกับ multi-class classification ที่มี class imbalance

สิ่งที่ใช้ได้ผลดี ✓

1. Dropout Regularization (V2, V4)

- ช่วยลด overfitting อย่างมีนัยสำคัญ
- Balance ระหว่าง regularization และ capacity

2. Weight Decay (1e-4)

python

```
optimizer = Adam(lr=1e-3, weight_decay=1e-4)
```

- L2 regularization ช่วยปรับปรุง generalization
- ค่า 1e-4 เหมาะสมกับ dataset นี้

3. Early Stopping (patience=5)

- V4 ใช้ patience=5 → หยุดก่อน overfit
- V2 ใช้ patience=10 → accuracy สูงกว่าแต่ F1 ต่ำกว่า
- Patience ต่ำ → better balance

4. Learning Rate Scheduling

python

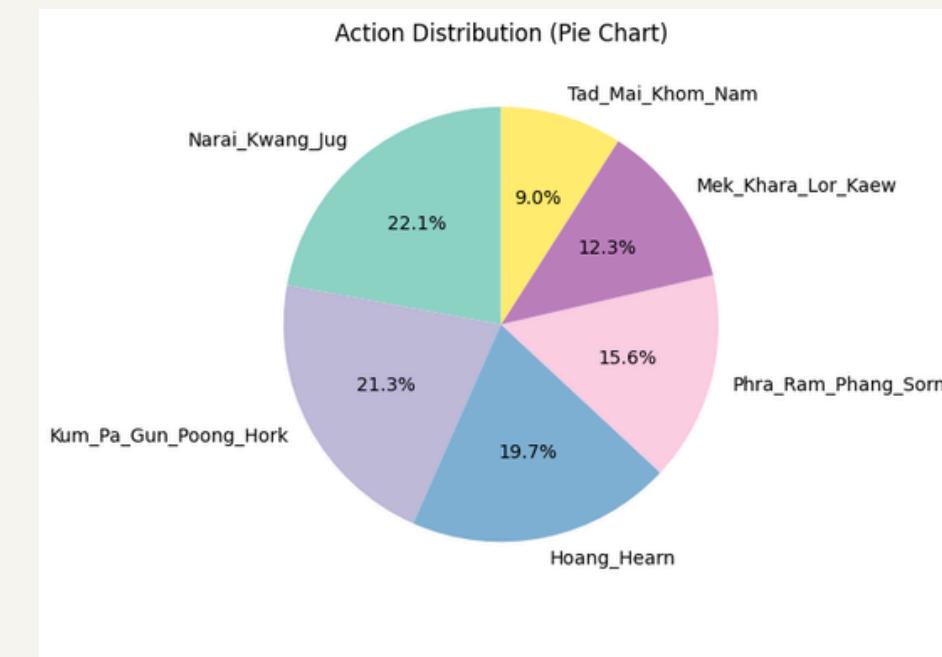
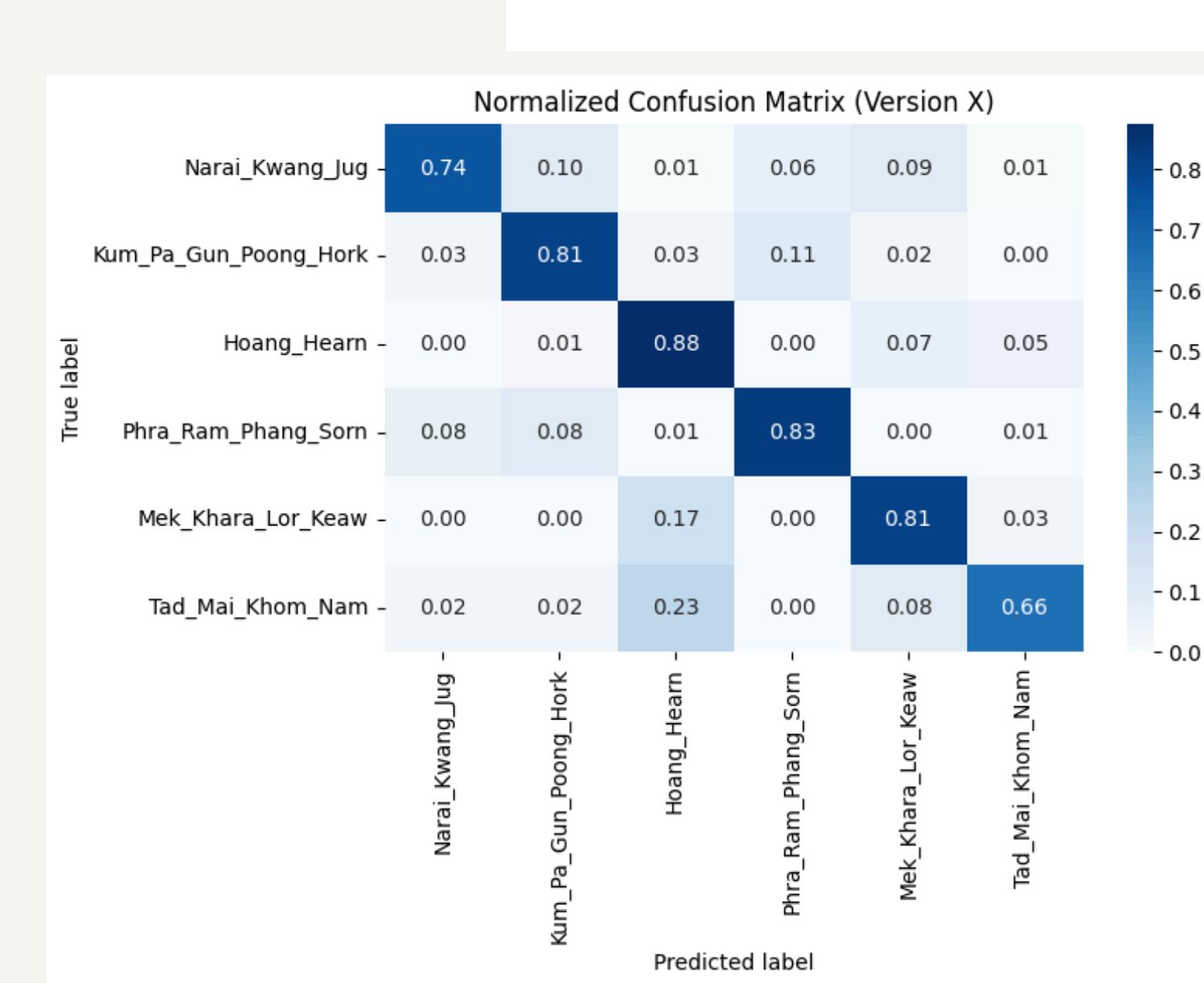
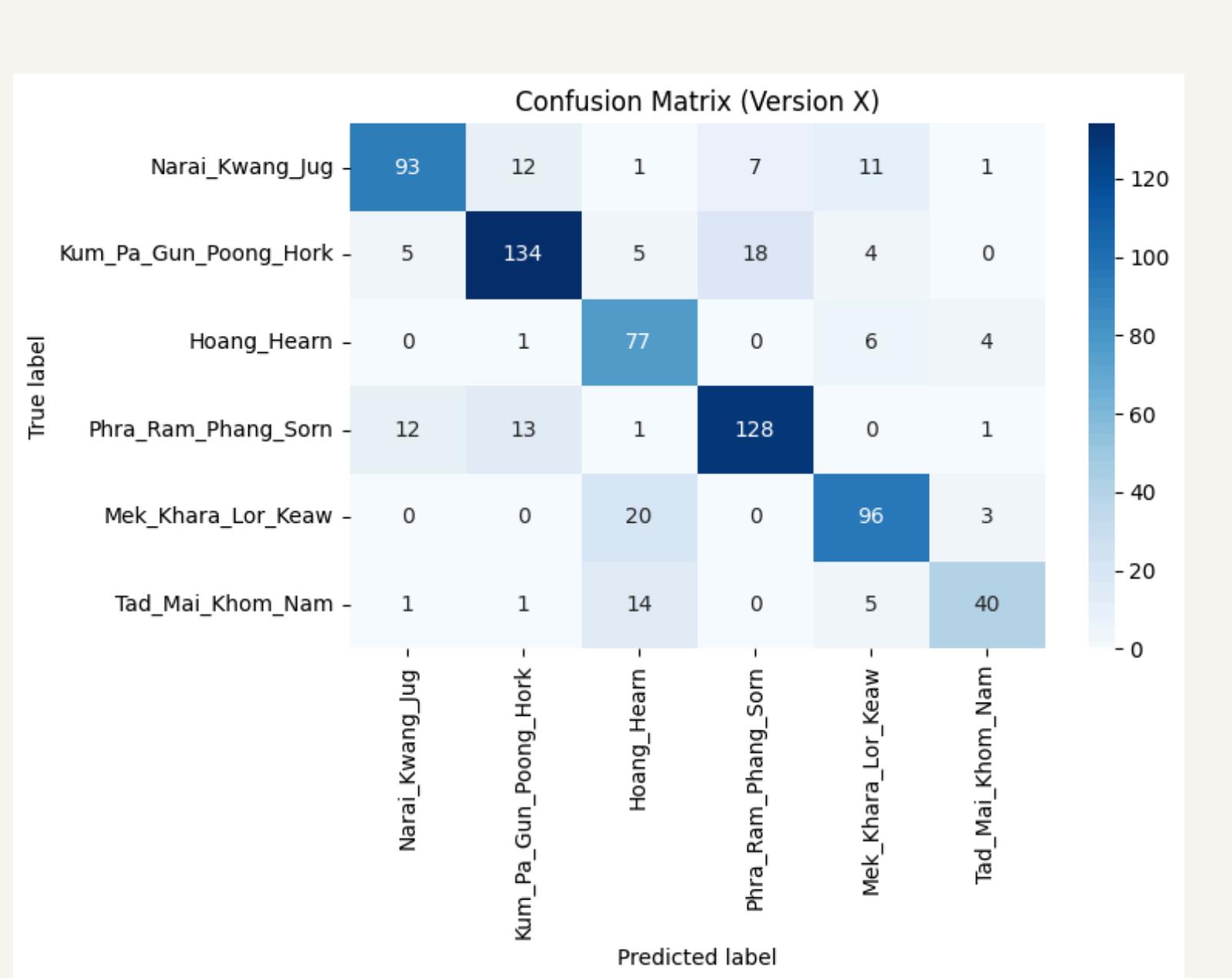
```
lr_scheduler = ReduceLROnPlateau(patience=5)
```

- ช่วยให้ training มีเสถียรภาพ
- Fine-tune ได้ดีในช่วงท้าย

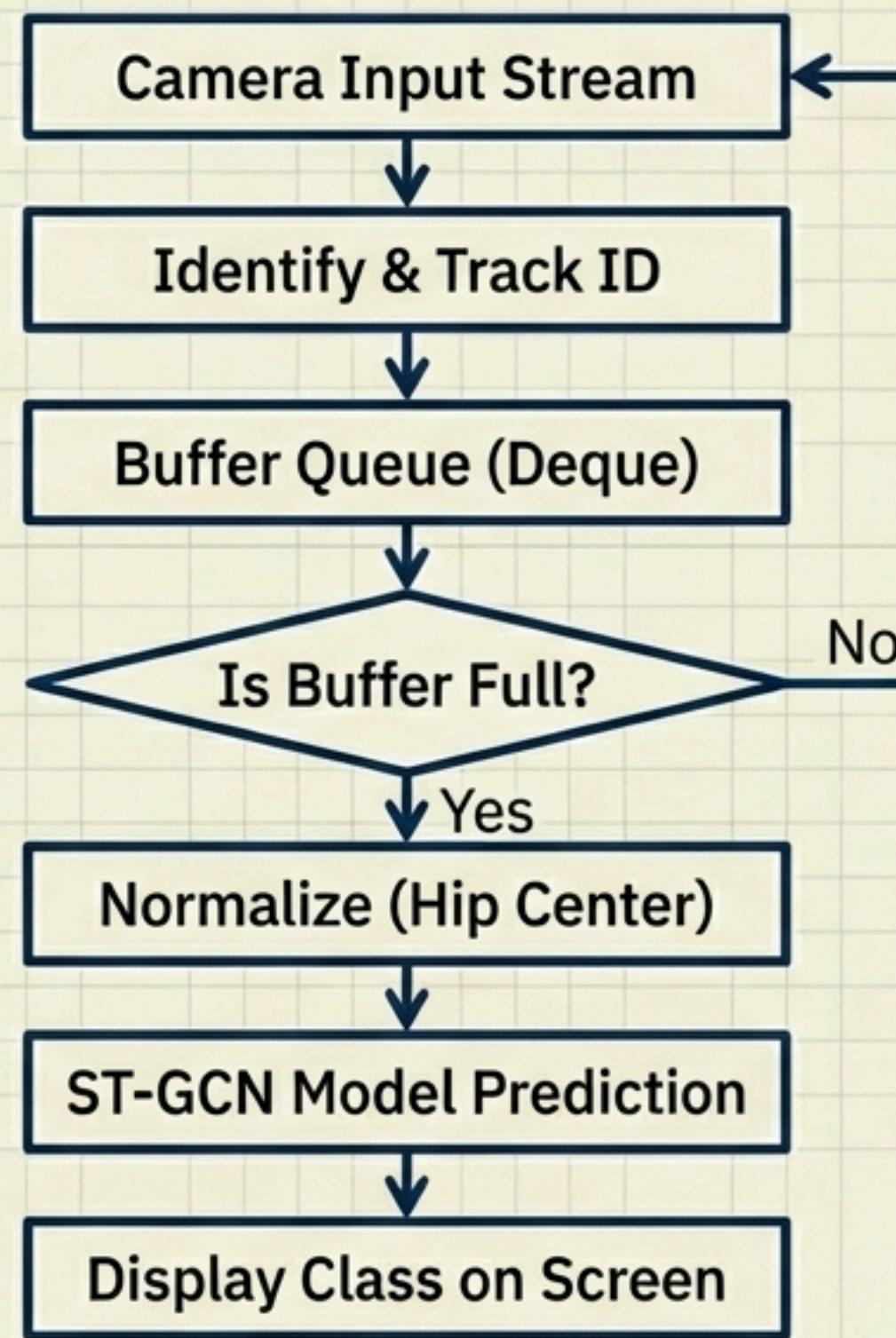
5. Normalization & Gradient Clipping (V4)

- Batch normalization
- Gradient clipping
- Label smoothing
- ช่วยให้ training เสถียรและ balanced

Confusion Matrix: Best Model (V4)



សេចក្តីថ្លែងការណ៍ Real-Time Inference



Must collect $T=30$ frames per ID

Real-Time Result

ก่าเริ่มต้น แยกจ่าแบนก



Real-Time Result

ការងារទូរសព្ទអំពីការបង្កើតរបស់ខ្លួន



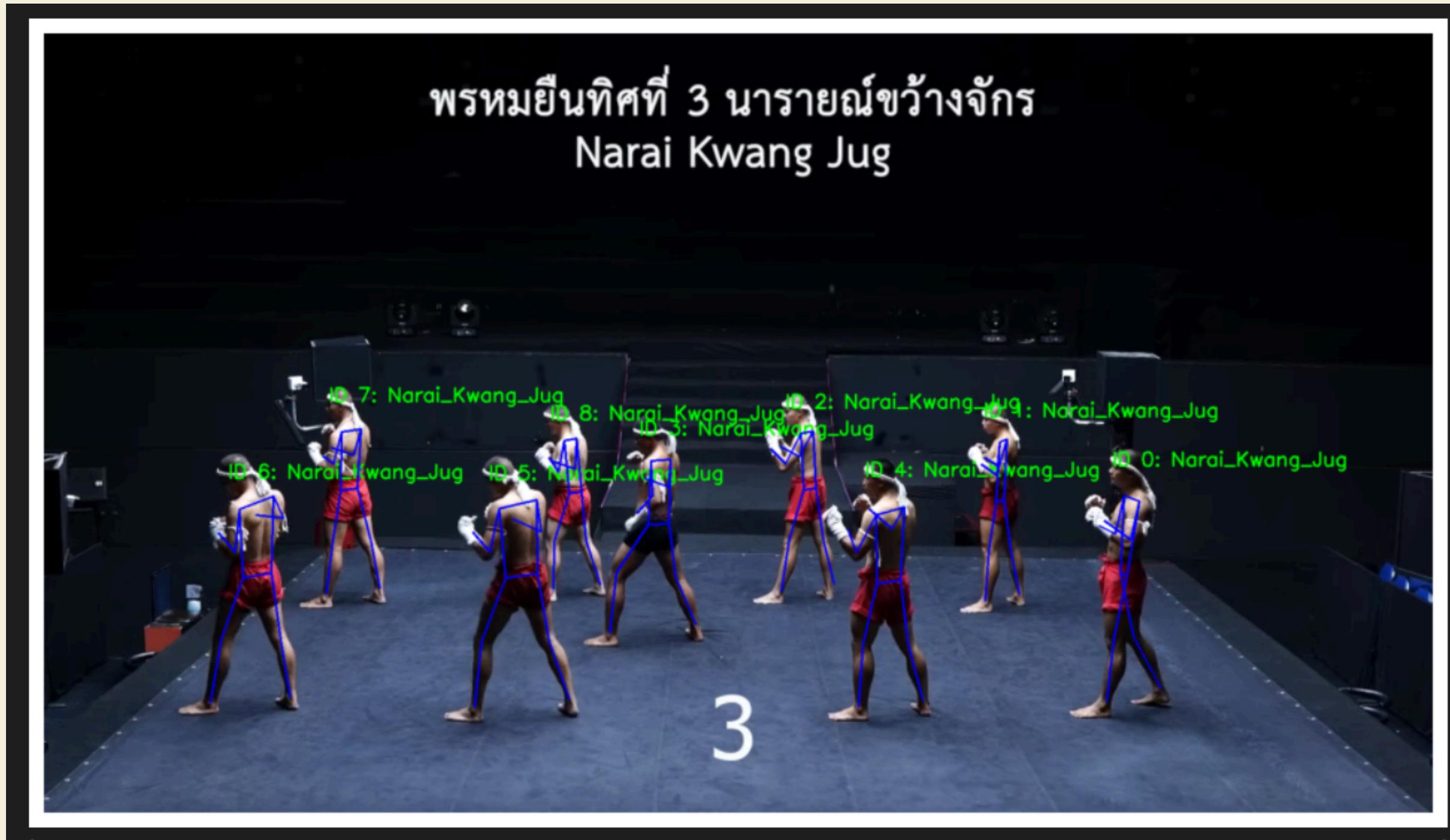
Real-Time Result

ท่าคล้ายกัน แต่ละคนกำองศาไม่เท่ากัน ทำให้กำหนดได้ท่าต่างกัน

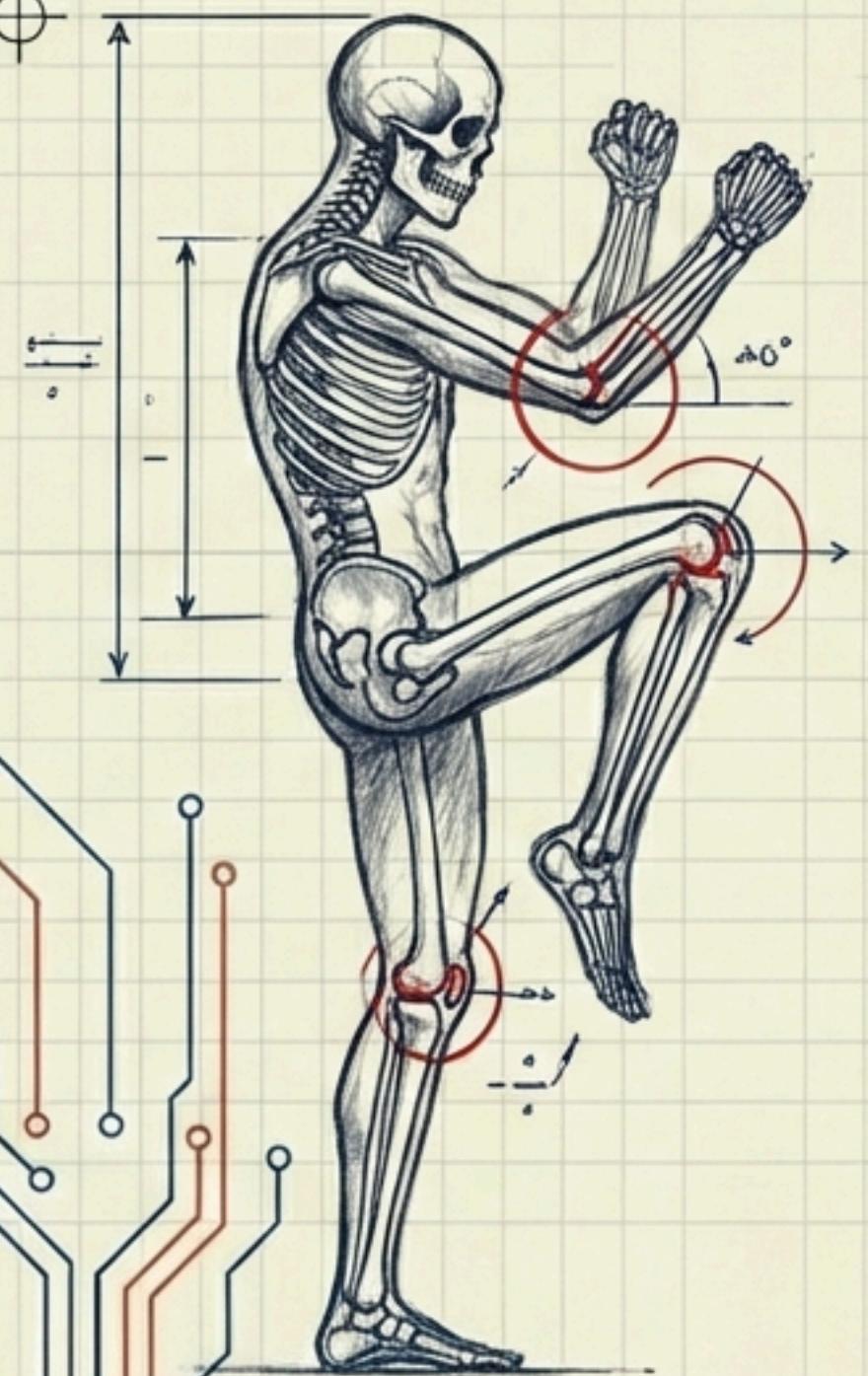


Real-Time Result

ID3: ตำแหน่งแขนข้างนึ่งต่างจากคนอื่น แต่ก็ยังสามารถกำหนดได้ถูก



การตรวจสอดความผิดปกติ (Anomaly Detection)



Concept

วัดระยะห่างระหว่าง ‘ก่าทีกำ’ กับ ‘ก่ามาตรฐาน’ ใน Training Set เพื่อประเมินความถูกต้องของฟอร์ม

Methodology

Feature: Joint Angles (มุมข้อต่อ)

Algorithm: Mahalanobis Distance

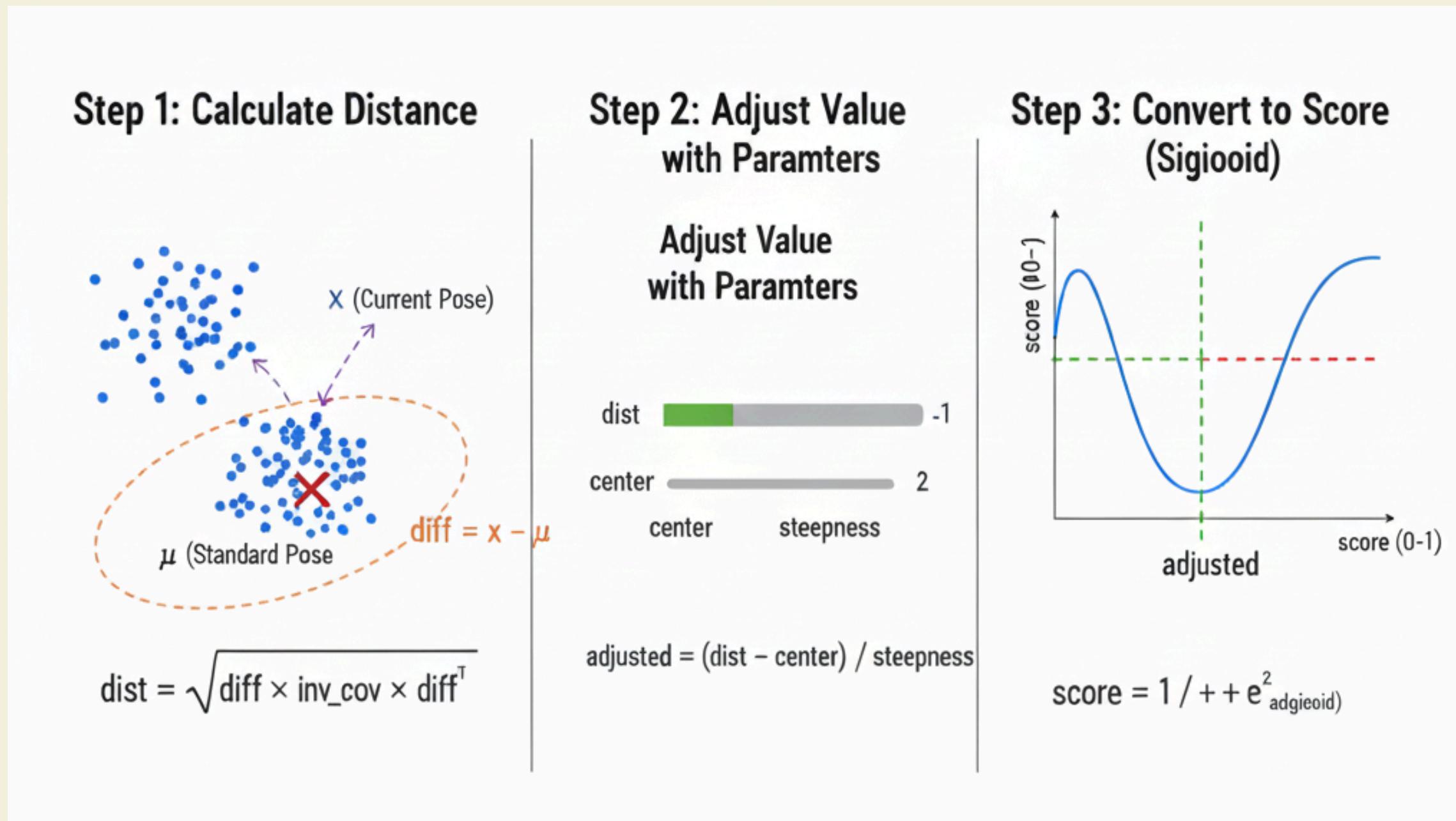
Interpretation

Similarity Score:

ความหมาย: ความคล้ายคลึงกับก่ามาตรฐาน

100% = ปกติมาก, 10% = ผิดปกติมาก

การตรวจสอดความผิดปกติ (Anomaly Detection)



Real-Time Result



Real-Time Result



Real-Time Result



สรุปและแนวทางการพัฒนา (Conclusion & Future Work)



Key Success

Data Quality: การ Merge Duplicate ช่วยแก้ปัญหา Noise ได้อย่างมีนัยสำคัญ

Robust Pipeline: รักษา Consistency ตั้งแต่ Ingestion ถึง Inference



Future Work

Augmentation: Rotate/Scale for camera invariance

Normalization: Add Image Size Normalization

Backbone Upgrade: Test YOL0v11 or MMPose

