Learning Higher-order Object Interactions for Keypoint-based video understanding

Yi Huang, Asim Kadav, Farley Lai, Deep Patel, Hans Peter Graf
Machine Learning Group
NEC Labs America


NEC
Machine Learning

VIDEO UNDERSTANDING BENEFITS FROM STRUCTURE

Segmentation

Skeleton

Detection/Tracking

Classification

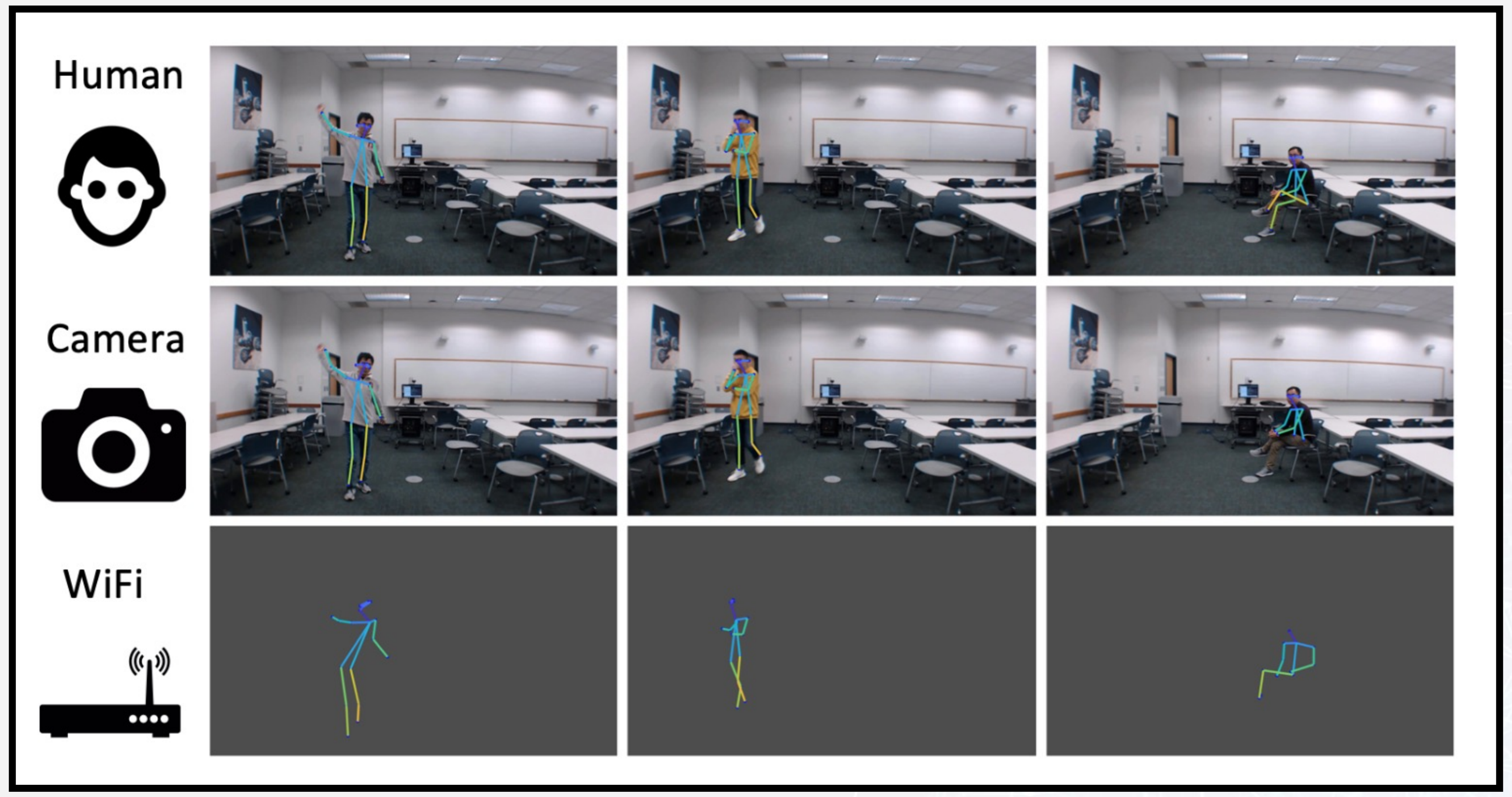
Localization

Language

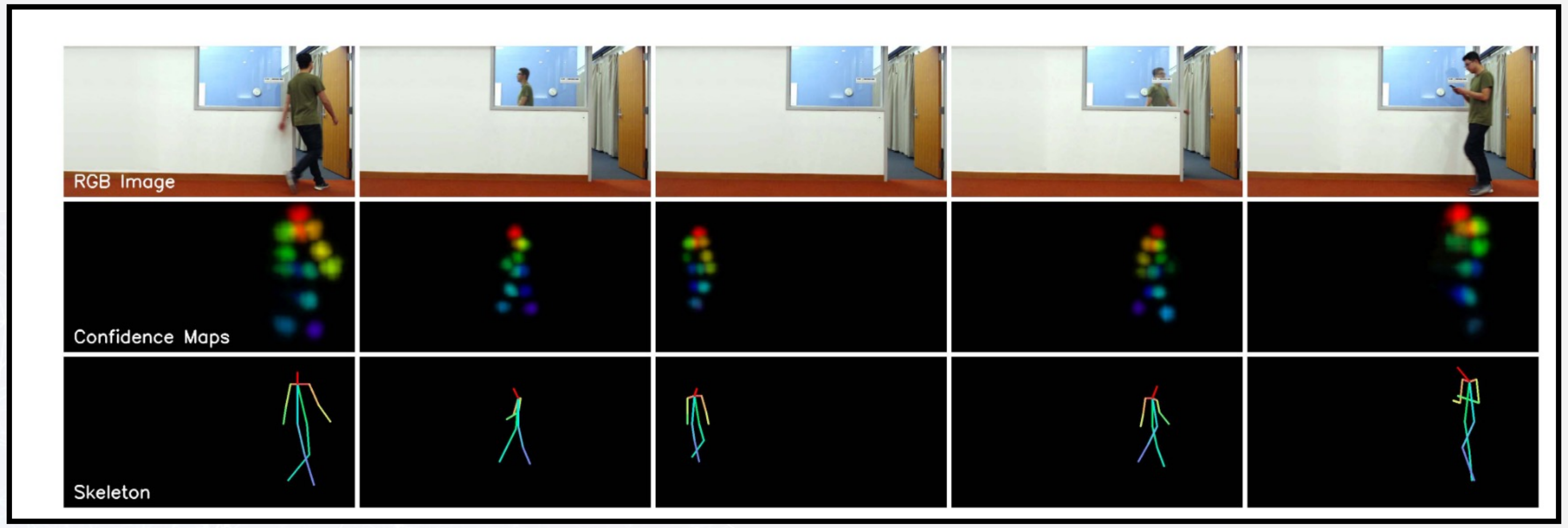
- Structure makes it easier for model to learn
- Improves learning and inference efficiency

This talk => Understanding videos from large amounts of keypoint data
Keypoint based tracking (CVPR '20) extended to Keypoint based action recognition (SVU Workshop'21)

WHY KEYPOINTS? MANY NOVEL HARDWARE DEVICES FOR POSE ESTIMATION

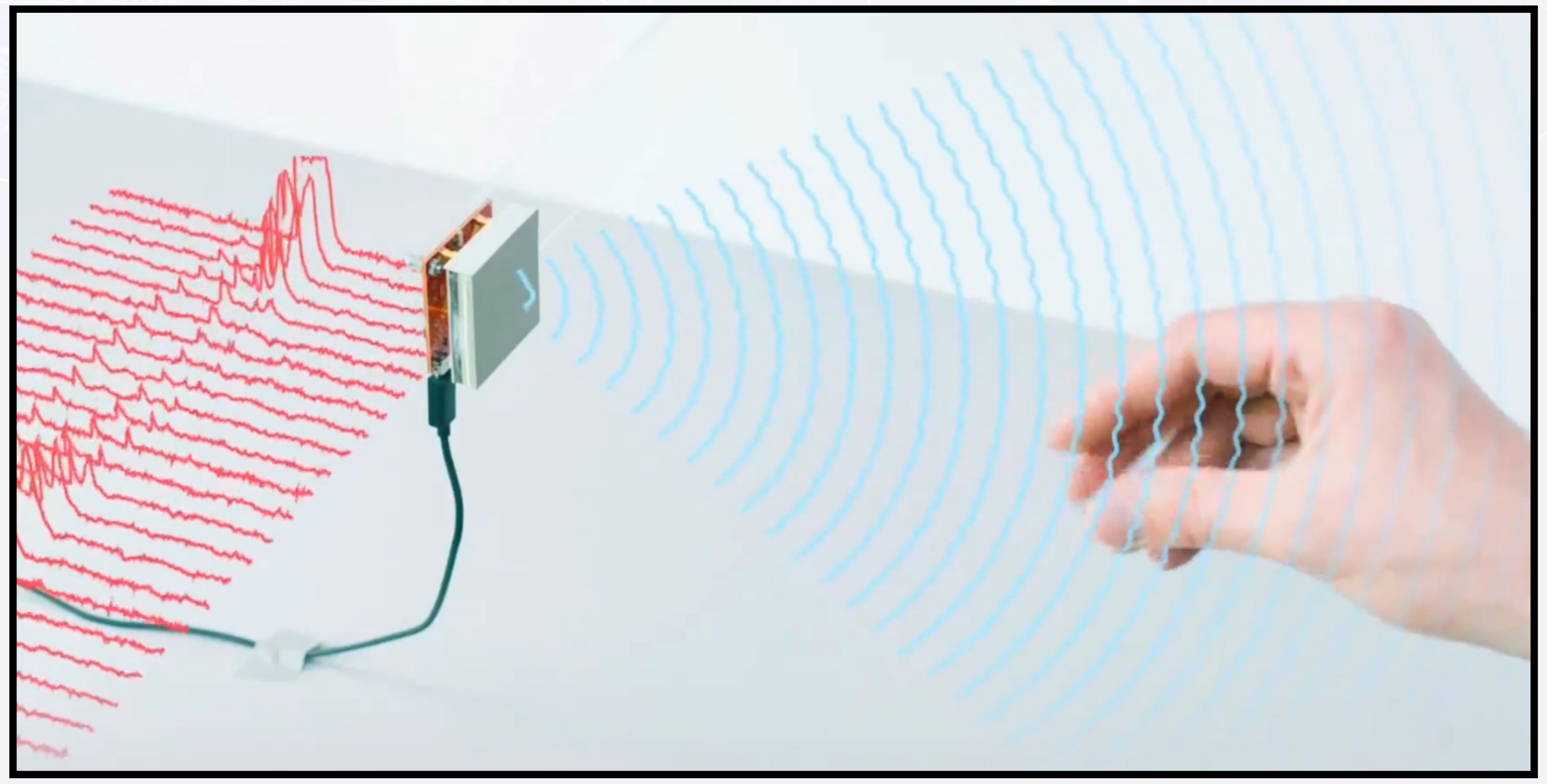


Can WiFI Estimate Person Pose?, arXiv 2019



Through-Wall Human Pose Estimation Using Radio Signals, CVPR 2018

WiFi based



Soli

Hand pose estimation, Google, 2019

RF based

TRACKING MULTIPLE PERSONS THROUGH TIME

Tracking persons through time is important for long term video understanding

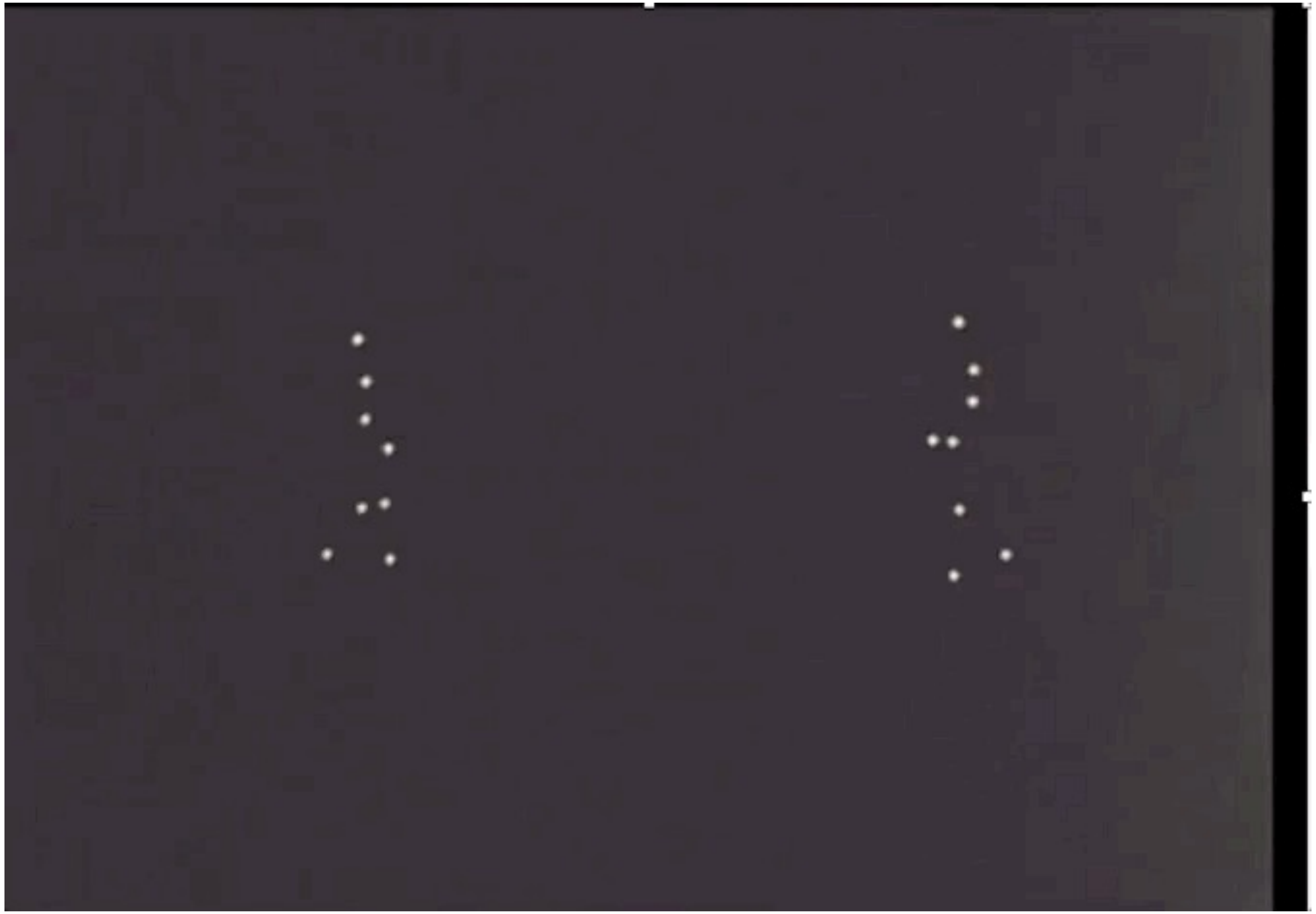
Crucial for building any applications over video

In this work, we focus on pose-based tracking (track 15 human joints)

Approaches must interpret occluded poses and complex motion while being efficient

Annotations from the PoseTrack Dataset, currently the largest benchmark for human pose tracking.

KEYPOINTS ARE ALL YOU NEED FOR TRACKING



Study of 2D motion perception, Gunnar Johansson, 1971.

- Can we use keypoints as our sole modality for tracking ?
- Why? This is 100s of times more efficient than Optical Flow based tracking, which must parse RGB information

Our approach: 15 keypoints is all you need, CVPR 2020

OVERVIEW OF MULTI-PERSON POSE TRACKING

KEYPOINT ESTIMATION

Top-Down

- Detect bboxes -> Estimate keypoints

Bottom-up

- Estimate keypoints for all poses at once
- Faster than top-down

TEMPORAL MATCHING

- IoU: fast, but prone to error
- Optical Flow: more accurate than IoU, but slow
- Graph Convolution Networks: more efficient and accurate than the previous two; but use convolutions and thus are dependent on spatial resolution

ID ASSIGNMENT

- Match scores from temporal matching step to track IDs
- Usually a greedy algorithm or Hungarian algorithm is used

OUR APPROACH: 15 KEYPOINTS IS ALL YOU NEED

KEYPOINT ESTIMATION

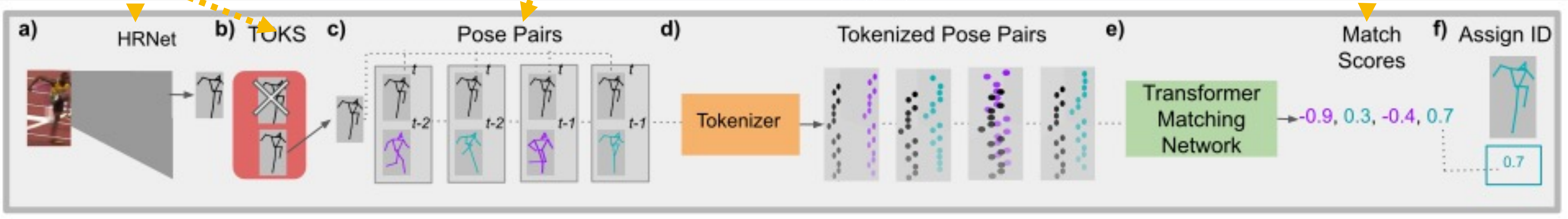
- Use HTC-Cascade for person bbox detection
- Use HRNet to detect keypoints in bounding boxes
- Use temporal information to augment missed/poor quality detections using TOKS*

TEMPORAL MATCHING

- Propose transformer based "pose entailment" network*
  - Tokenize pose-pairs at time-step t, t-d
  - Predict if the pairs temporally follow one another
- Simple binary classification with 0.43M parameters, achieved SOTA and #1 in PoseTrack Leaderboard 2019-20

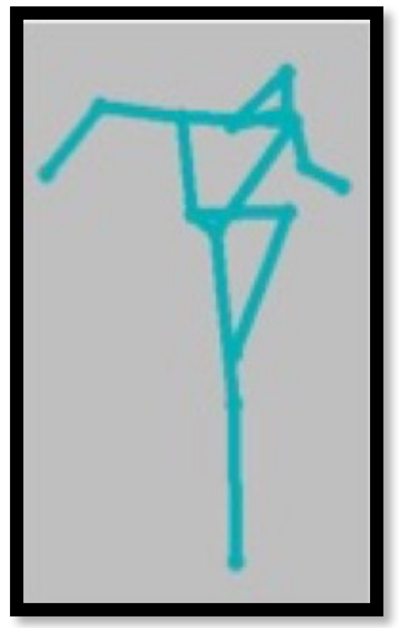
ID ASSIGNMENT

- Greedily maximize assignments using the match scores from matching step

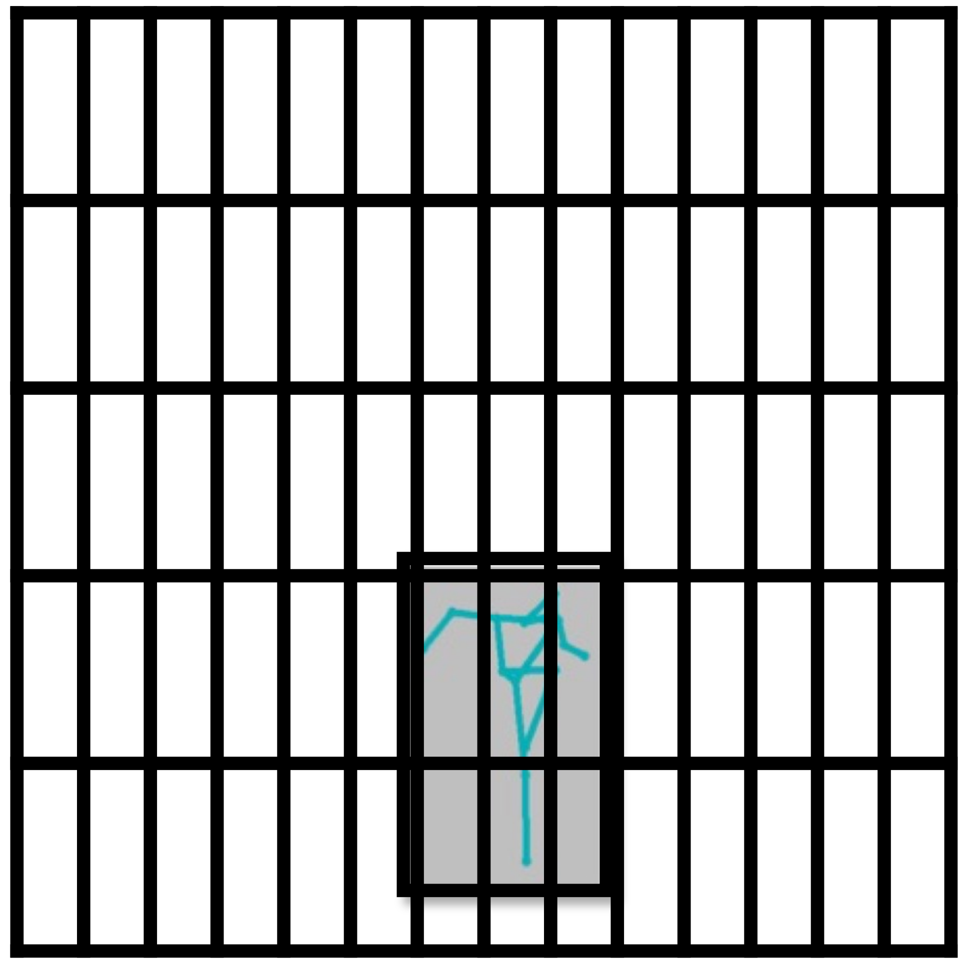


OUR APPROACH: ONLY USE KEYPOINT DATA AS TRANSFORMER INPUT

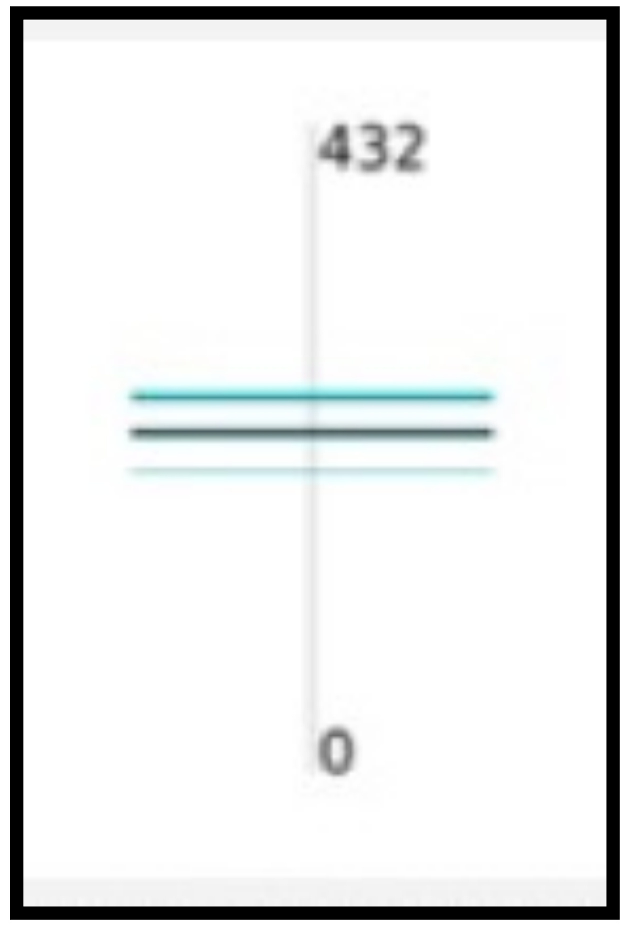
Original image
(e.g. 336x336)



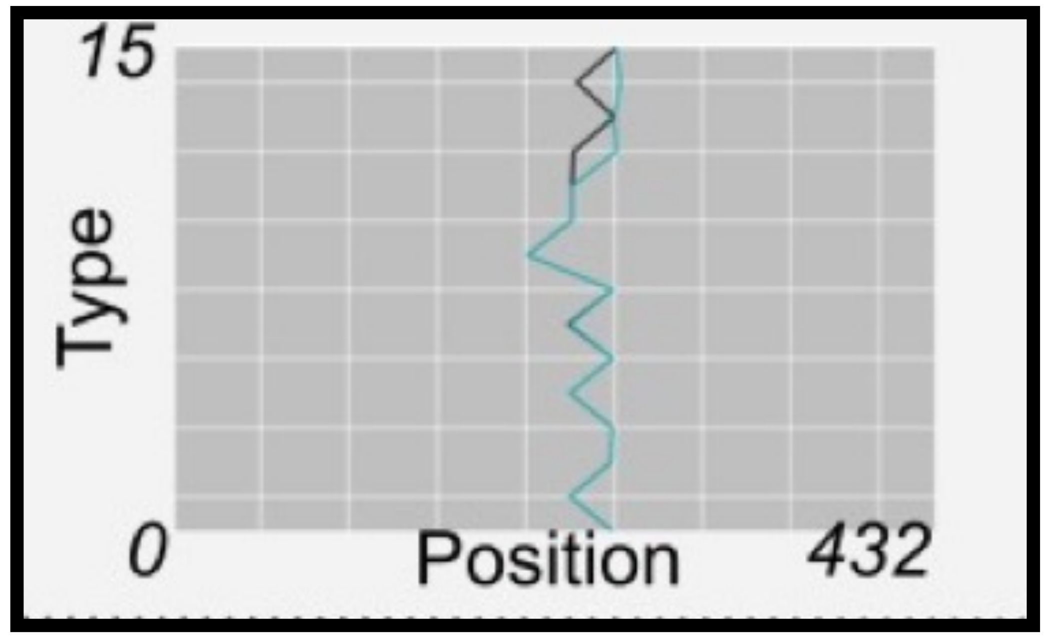
Extract pose for
every person



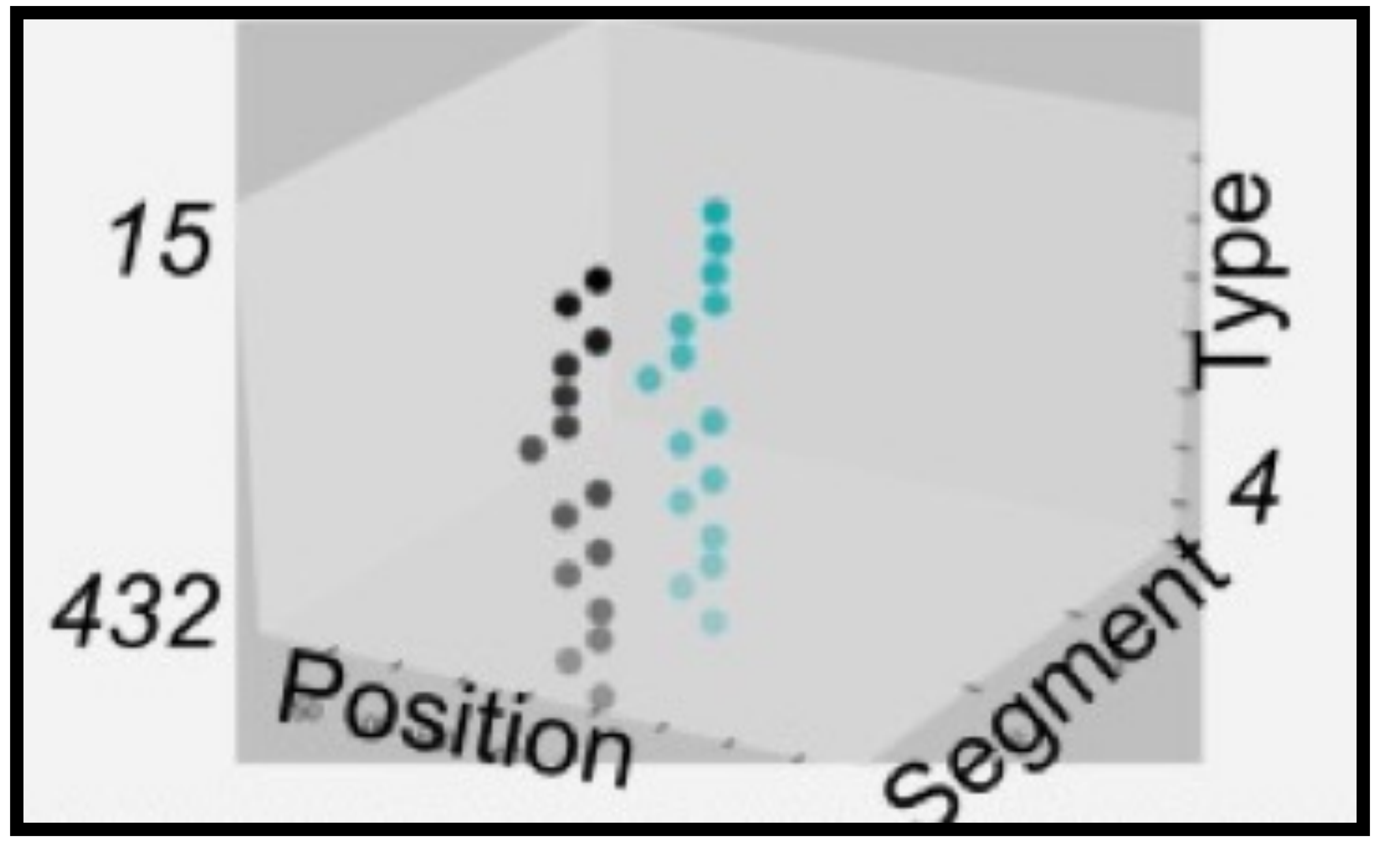
Downsample coordinates into 24x18



Flatten to 1D



Type (Joint) embeddings



Type (Joint) + Segment
(Time) embedding

TOKENIZING KEYPOINT SEQUENCES

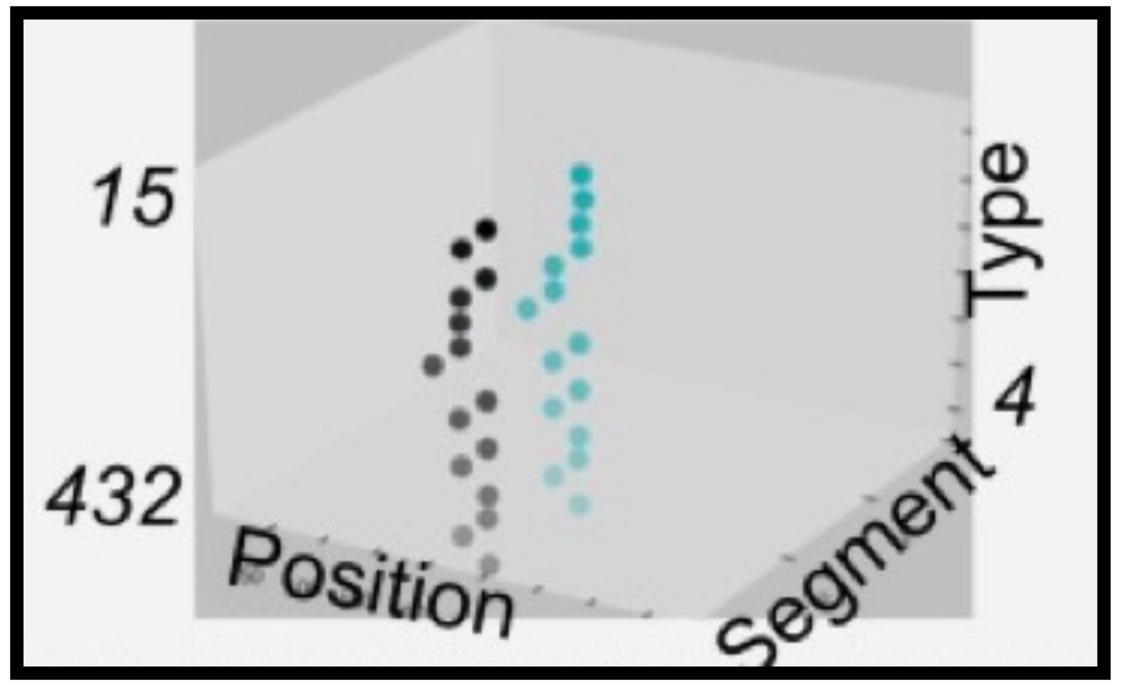
We tokenize a pose pair as follows (domain expression on left, range on right):

Position: a linear projection of a keypoint's cartesian coordinates

$\{\rho_1^{p^t}, \rho_2^{p^t}, \dots \rho_{|\mathcal{K}|}^{p^t}, \rho_1^{p^{t-\delta}}, \rho_2^{p^{t-\delta}}, \dots \rho_{|\mathcal{K}|}^{p^{t-\delta}}\}$ $\quad [1, w^{\mathcal{F}} h^{\mathcal{F}}]$

Type: Name of joint: e.g. the head, left shoulder, right ankle etc…

$\{1^{p^t}, 2^{p^t}, \dots |\mathcal{K}|^{p^t}, 1^{p^{t-\delta}}, 2^{p^{t-\delta}}, \dots |\mathcal{K}|^{p^{t-\delta}}\}$ $\quad [1, |\mathcal{K}|]$



Segment: Temporal distance from current frame. (We set this to 4)

$\{1^{p^t}, 1^{p^t}, \dots 1^{p^t}, \delta^{p^{t-\delta}}, \delta^{p^{t-\delta}}, \dots \delta^{p^{t-\delta}}\}$ $\quad [1, \delta]$

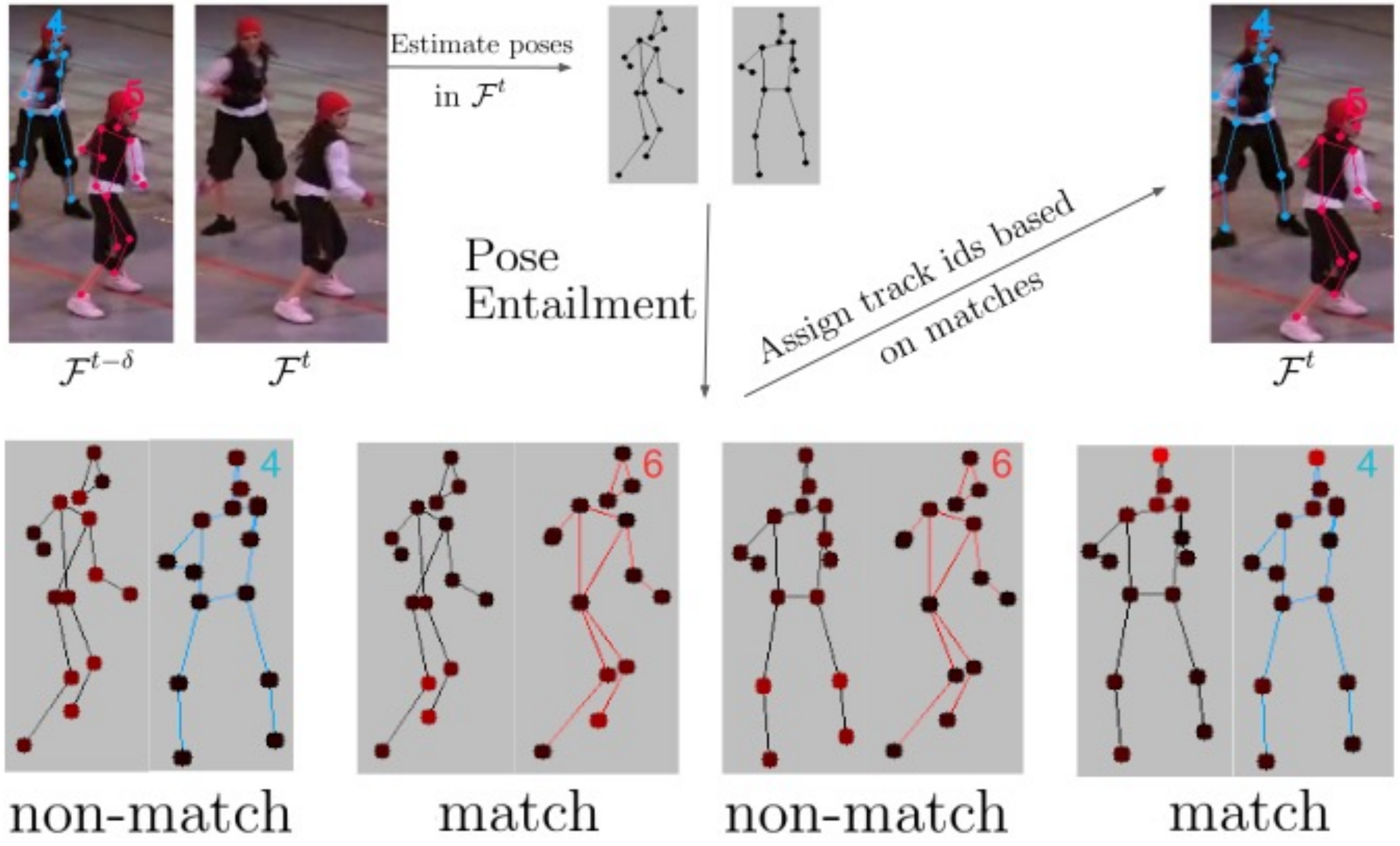
p^t pose from current frame	$p^{t-\delta}$ pose from previous frame	$w^{\mathcal{F}}$ frame width	$h^{\mathcal{F}}$ frame height	$\lvert\mathcal{K}\rvert$ num. keypoints per pose	ρ (x coord) * (y coord)

Tokens are then projected to the transformer hidden size, H, via a learned lookup table. The sum of the embeddings is input to a transformer matching network which classifies whether the pose pair is a match (i.e. the same person)

INPUT TOKENIZED POSES TO TRANSFORMER FOR ENTAILMENT

Inspired by Textual Entailment[1, 2], we propose *Pose Entailment*, where a transformer-based model learns to make a binary classification as to whether two poses temporally entail each other.

- ✓ **The input to our model is a sequence of keypoints representing two poses, making it more efficient than a model that slides convolutional filters over high-res keypoint images**
- ✓ **Transformers, not limited by receptive field, are able to learn higher order interactions over poses**

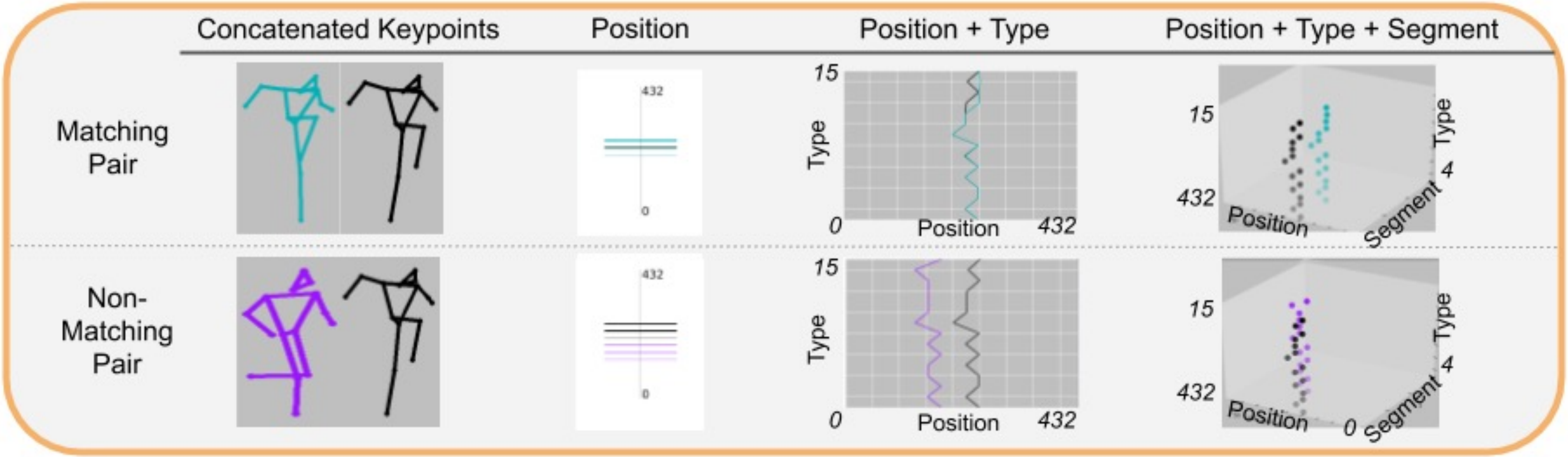


Our model assigns a match likelihood to pose pairs. Attention heat maps are visualized with bright red corresponding to high attention. In matching pairs, attention is evenly divided between the poses, whereas in non-matching pairs, it is focused on one pose.

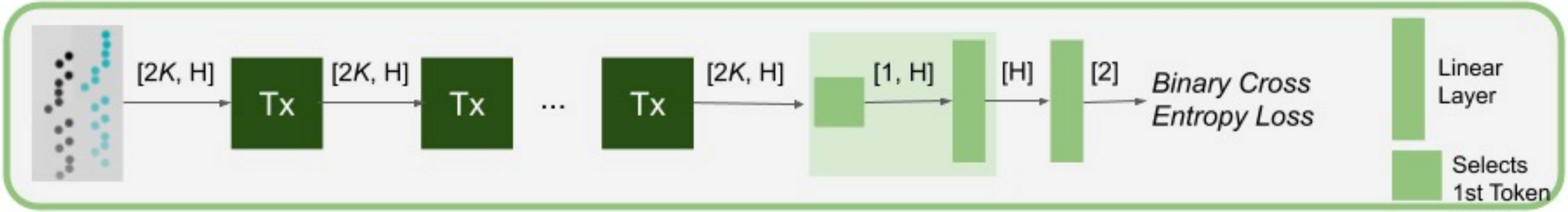
[1] https://demo.allennlp.org/textual-entailment

[2] Visual Entailment: A Novel Task for Fine-Grained Image Understanding, Neuripsw 2018,19

ENTAILMENT VISUAL ILLUSTRATION AND MODEL



Parameters = 0.41M, 6.2M FLOPs, Optical flow: Params: 38.7M, 52.7G FLOPs



Transformer matching network: K = |Keypoints| = 15, H = hidden size (128)

IMPROVING POSE ESTIMATION OUTPUTS WITH TEMPORAL OKS

- Pose estimation methods suffer from:
  - Missed bounding boxes
  - Imperfect bounding boxes
- Use bboxes from previous time steps
- Use OKS instead of NMS to determine the pose to keep

```
def  temporal_oks (p_t-1, p_t, F_t):

B = retrieve_bbox(p_t-1)
p_hat_t = new_pose_estimate (F_t,
B, alpha)
pose_to_keep = oks (p_hat_t, _pt)
return pose_to_keep
```

#1 ON POSETRACK LEADERBOARD (NOV 2019 - APR 2020)

PoseTrack 2018 ECCV Challenge Val Set

No.	Method	Extra Data	AP^T	AP	FPS	MOTA
1.	**KeyTrack (ours)**	✗	74.3	**81.6**	1.0	**66.6**
2.	MIPAL	✗	**74.6**	-	-	65.7
3.	LightTrack (offline)	✗	71.2	77.3	E	64.9
4.	LightTrack (online)	✗	72.4	77.2	0.7	64.6
5.	Miracle	✓	-	80.9	E	64.0
6.	OpenSVAI	✗	69.7	76.3	-	62.4
7.	STAF	✓	70.4	-	**3**	60.9
8.	MDPN	✓	71.7	75.0	E	50.6

PoseTrack 2017 Test Set Leaderboard

No.	Method	Extra Data	AP^T	FPS	MOTA
1.	**KeyTrack (ours)**	✗	74.0	1.0	**61.2**
2.	POINet	✗	72.5	-	58.4
3.	LightTrack	✗	66.7	E	58.0
4.	HRNet	✗	**75.0**	0.2	57.9
5.	FlowTrack	✗	74.6	0.2	57.8
6.	MIPAL	✗	68.8	-	54.5
7.	STAF	✓	70.3	**2**	53.8
8.	JointFlow	✗	63.6	0.2	53.1

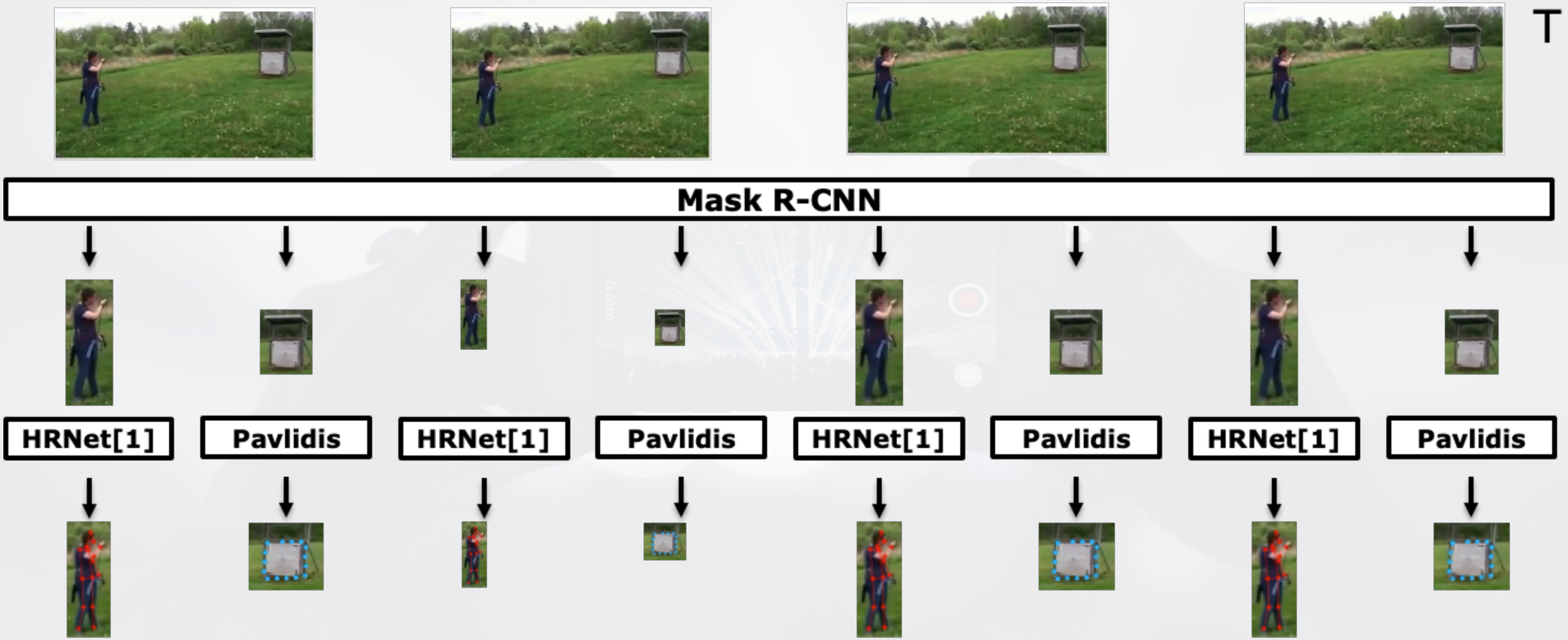
Achieves 61.2% tracking accuracy on the PoseTrack'17 Test Set and 66.6% on the Pose-Track'18 Val.

Tracking step has only 0.43M parameters and is 500X more efficient than the leading optical flow method

EXTENDING TRACKS TO PERFORM ACTION RECOGNITION

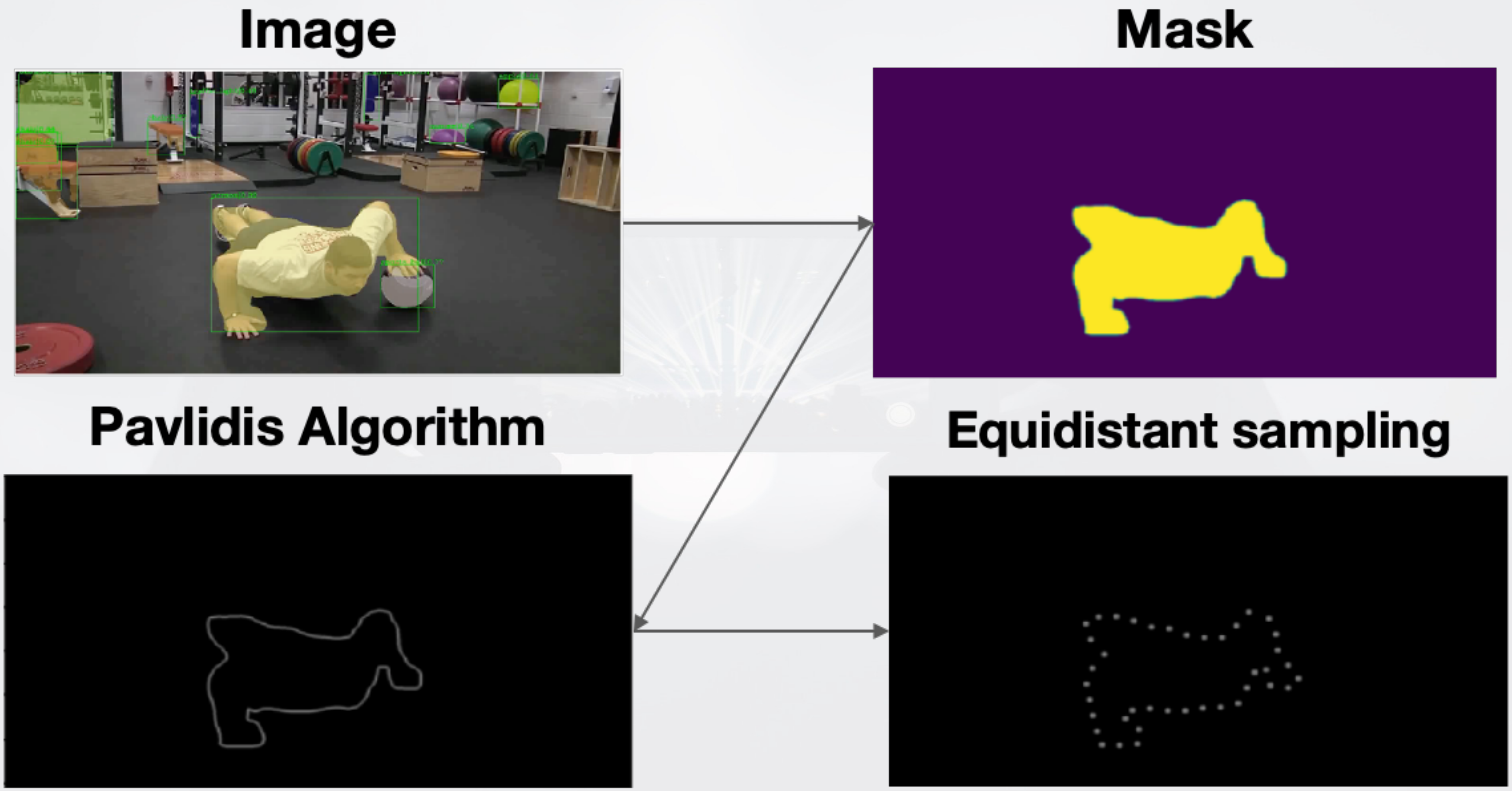
- **Hypothesis:**
  - Using only keypoint information for action recognition
- **Key idea:**
  - Use key-points for humans and objects, learn to connect them through space and time, object key points provide additional context
- **Advantages:**
  - New hardware developments to obtain keypoints
  - Lower cost than RGB pipelines
- **Open Questions:**
  - How to get sparse keypoint representations?
  - How to structure the embeddings for human and scene objects?
  - How to model the embedding for video understanding task?

HOW TO GET SPARSE KEYPOINT REPRESENTATION?



[1] Deep High-Resolution Representation Learning for Visual Recognition

CLASS AGNOSTIC OBJECT KEYPOINTS



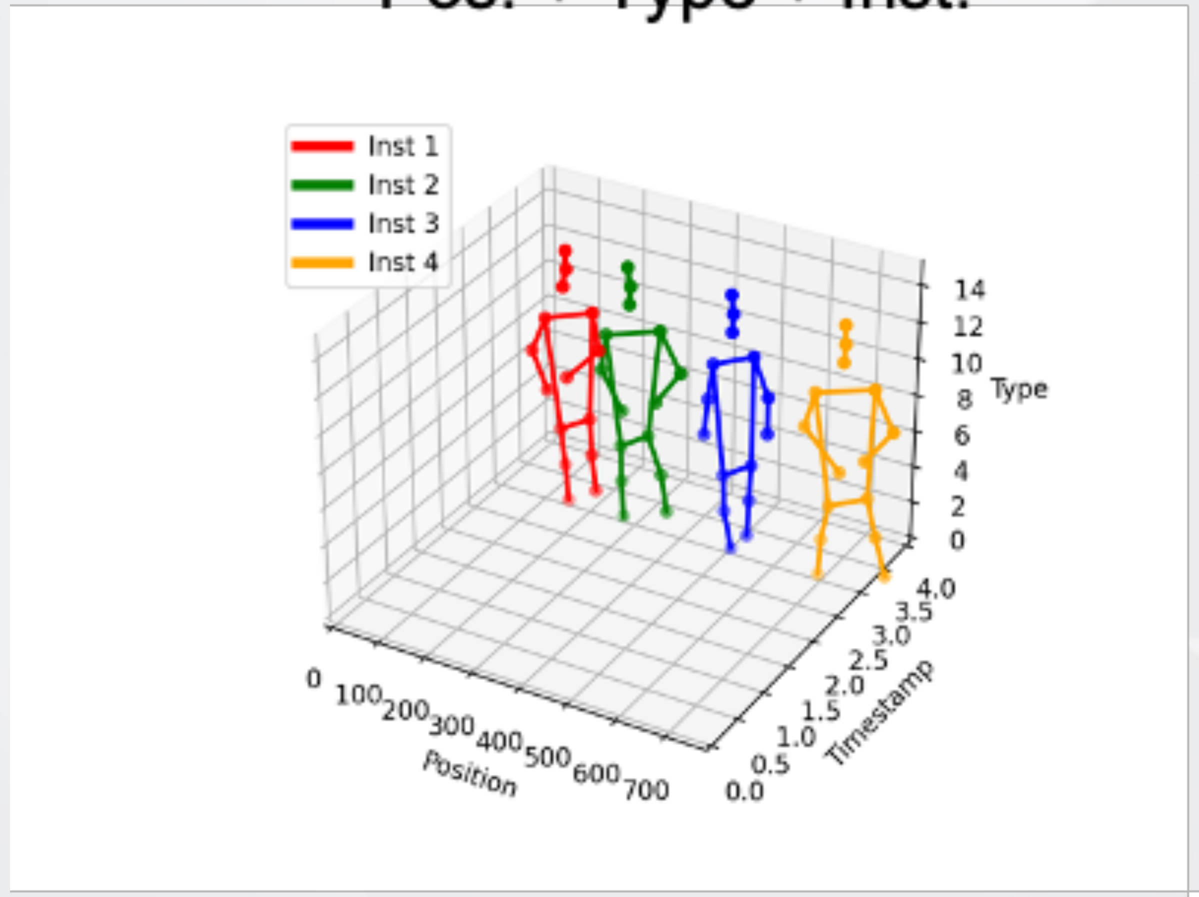
EMBEDDING TOKEN INFORMATION IN TRANSFORMERS

Input Keypoints	Head	Shou	Elb	Wri	2nd	Person	Key	Points	Next	Frame	Key	Points
Position Token	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}	E_{12}
Timestamp Token	T_1	T_1	T_1	T_1	T_1	T_1	T_1	T_1	T_2	T_2	T_2	T_2
Instance Token	P_1	P_1	P_1	P_1	P_2	P_2	P_2	P_2	P_1	P_1	P_1	P_1
Type Token	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4	S_1	S_2	S_3	S_4

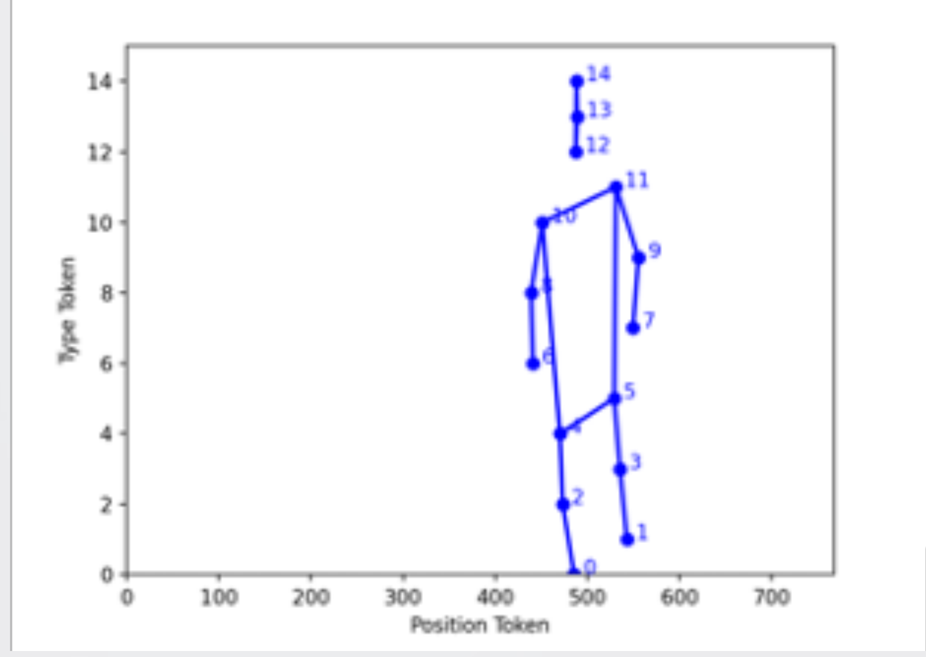
Type Token → Human body part information.

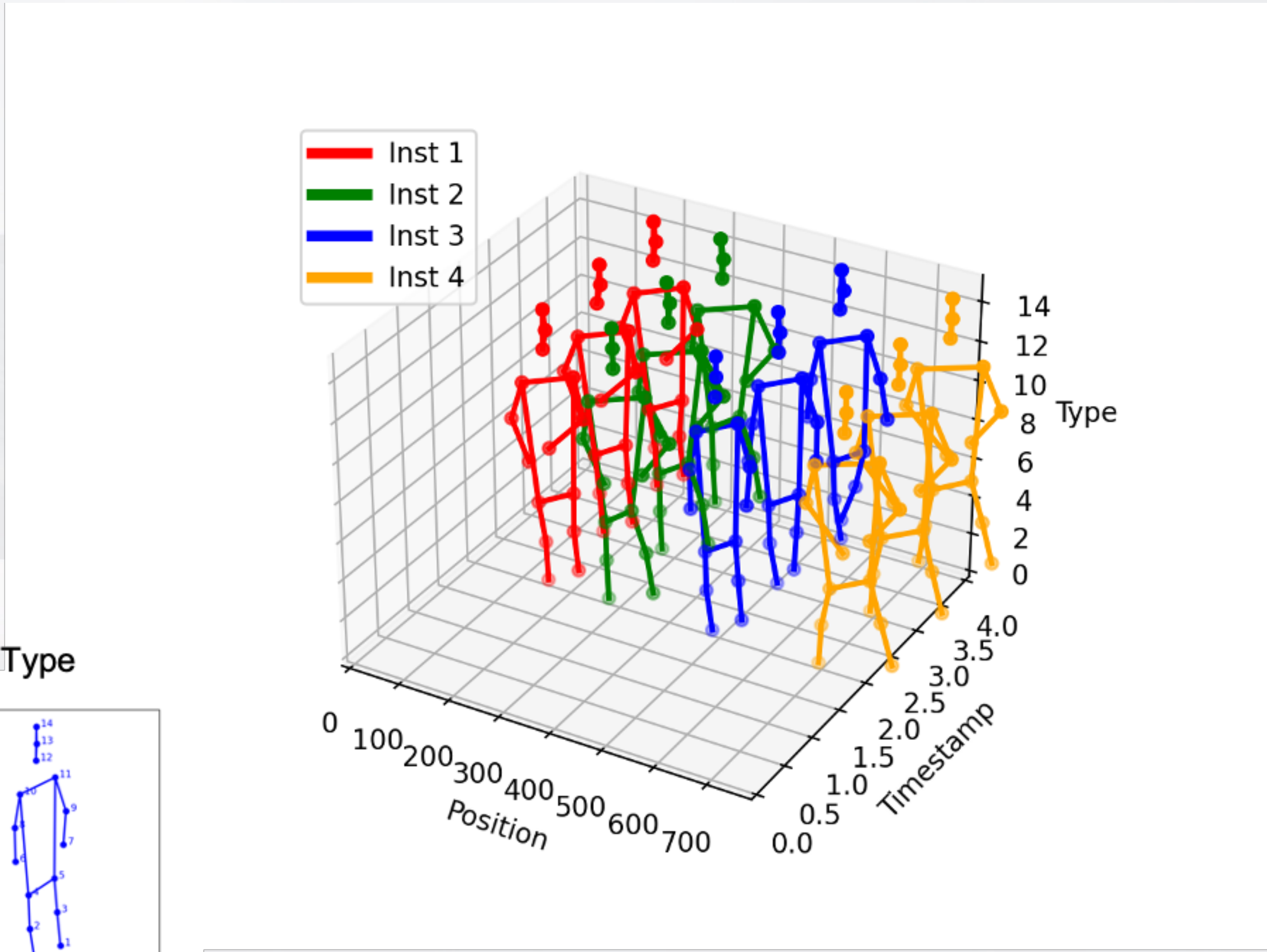
VISUALIZATION OF PROPOSED KEYPOINT TOKENS

Pos. + Type + Inst.

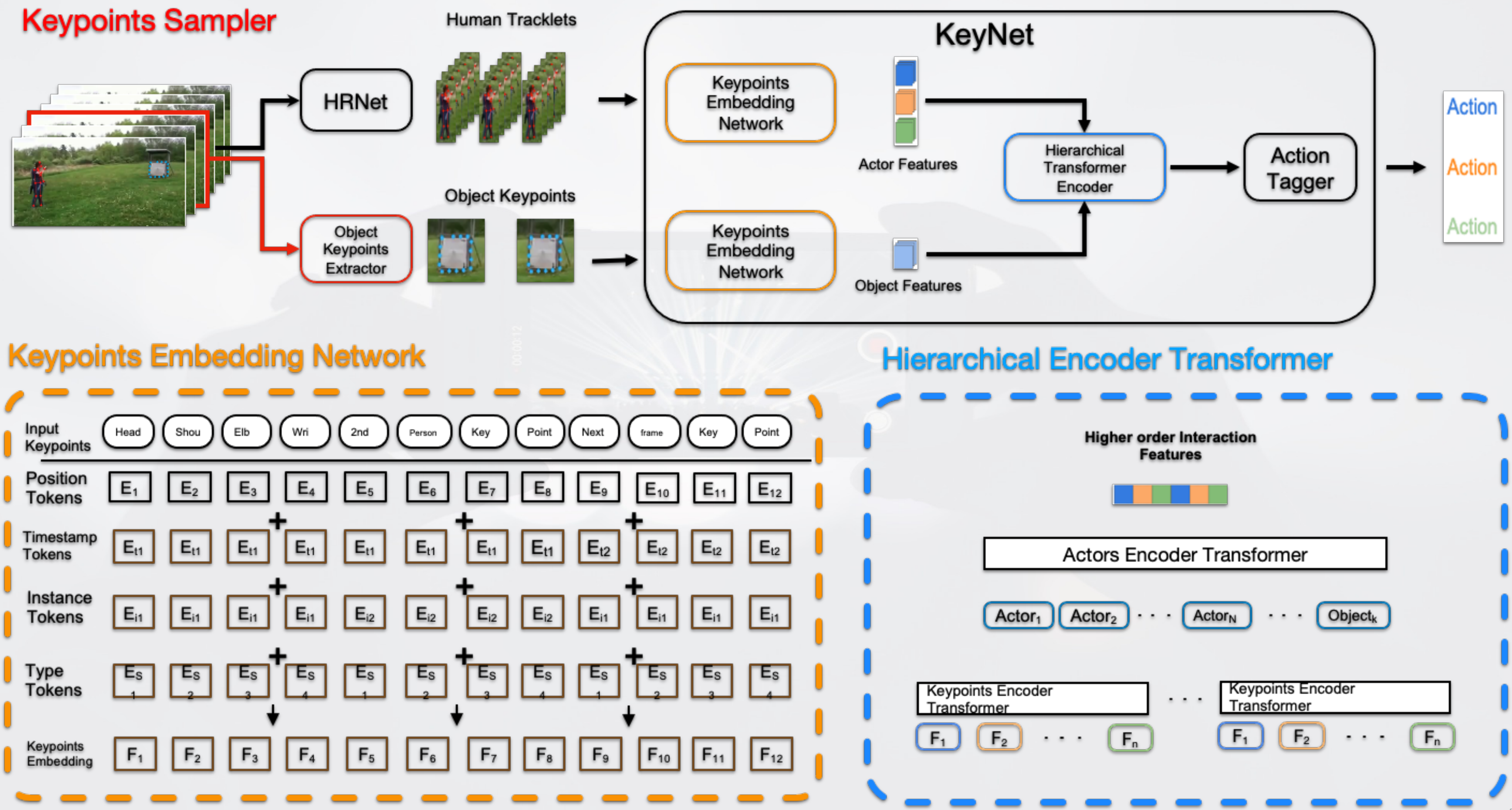


Pos. + Type





OVERALL ARCHITECTURE



RESULTS OVER AVA DATASET

Atomic Visual Action(AVA)

- Dataset
  - 430, 15-minute movie clips
  - 1 fps annotation.
  - 1.62 M action labels
- Categories
  - P : Person Movement (14 classes)
  - PP : Person-Person interaction (16 classes)
  - PO: Person-Object Interaction (50 classes)
- To avoid the highly imbalance nature in AVA, we only select 20 categories with more than 2000 samples, including 8 person movement actions, 4 person-person interaction actions and 4 person-object interaction actions.

Action Type	Data Aug.	Weighted Sampler	mAP
P			14.25
P	✓		20.28
P		✓	16.42
P	✓	✓	**31.41**

Object Keypoints	Action Type	mAP
✗	P + PP + PO	11.23
✓	P + PP + PO	11.45

SUMMARY

- Driven by hardware developments, keypoints are an excellent modality for video understanding
- Structuring the intermediate space with a focus of attention allows us to learn semantic video concepts
- KeyNet uses object keypoints to recover from loss of context in keypoints
- KeyNet achieves competitive results in multi-person tracking and action recognition using only keypoint information

QUESTIONS