# TEMPO: Efficient Multi-View Pose Estimation, Tracking, and Forecasting Supplementary Material

Rohan Choudhury     Kris M. Kitani     László A. Jeni
Robotics Institute, Carnegie Mellon University
{rchoudhu, kmkitani}@andrew.cmu.edu   laszlojeni@cmu.edu

## 1. Implementation Details

Our code was built on top of the MMPose[1] public repository , and we used their built-in implementations for the image backbone as well as the inference time analysis tools. We plan to release our code, model checkpoints and video results.

**Backbone** We use ResNet-50 [13, 3] as our backbone. On the Panoptic studio dataset, we use the checkpoint trained for 20 epochs on the Panoptic Studio dataset with 960 × 512 resolution images, introduced by the VoxelPose[11] codebase for accurate comparison with existing methods. Since we use synthetic heatmaps for Shelf and Campus, we use no backbone. On Human3.6M, we use the pre-trained ResNet backbone from the Learnable Triangulation [5] codebase. On all other datasets, we used HRNet[10] with $384 \times 384$ resolution, with no pre-training, following TesseTrack[9]. Following MvP [12], we use the pre-final layer of the backbone model's output head rather than the final per-joint heatmaps. This pre-final layer has 256 channels for ResNet and 32 for HRNet. **Detector** The person detector follows the design of [14]. We used a fixed voxel size of 10 $\mathrm{cm}^3$. For dataset-specific training, we follow previous papers and used a volume size of of $80 \times 80 \times 20$, on the Panoptic Studio dataset. We use the basic structure of V2V-Net [8], for the networks in this stage, but in 2D and 1D. The building block of this network consists of a convolutional block and a residual (skip-connection) block, with a ReLU connection and BatchNorm. We first feed the input to the network through a layer with a $7 \times 7$ kernel and then passed through three successive blocks, each with $3 \times 3$ kernels, with maxpooling between each block with kernel size 2. We then apply three transposed convolutional layers to obtain a feature map with the same spatial size as the input, to which we apply a $1 \times 1$ convolution to get the desired channel output size.

### 1.1. Cross-dataset Generalization

**Pose Estimation and Forecasting** The recurrent network we used was based on the SpatialGRU implementation used in FIERY [4] with a 2D LayerNorm based on the official ConvNexT implementation [7].

| Component | Time (ms) | GFLOPs | Params |
|---|---|---|---|
| Backbone | 11.72 | 29.3 | 23.51M |
| Detector (FV) | 17.3 | 1.204 | 1.51M |
| Detector(Ours) | 17.3 | 1.204 | 1.51M |
| Pose (FV) | 14.9 | 6.621 | 1.13M |
| Pose (Ours) | 16.5 | 7.331 | 1.926M |
| Total (FV) | 43.92 | 37.125 | 32.40M |
| **Total** | 45.52 | 38.831 | 33.19M |

Table 1. Runtime analysis of TEMPO compared with Faster VoxelPose (FV) [14]. Our model is competitive with Faster VoxelPose, which is the state-of-the-art in efficiency. Our model achieves significantly better pose estimation performance despite adding relatively few parameters and without adding significant overhead.

| Method | Panoptic | Human3.6M |
|---|---|---|
| MVPose | 55.6 | 83.4 |
| VoxelPose | 17.68 | 273.2 |
| Faster VoxelPose | 18.26 | 283.1 |
| TEMPO (Ours) | 14.18 | 63.4 |

Table 2. TEMPO significantly surpasses optimization-based methods on datasets it was not trained on, despite their dataset-agnostic design

At each timestep, the 2D projected features were fed into an encoder with the same structure as the encoder portion of the 2D CNN used in the detection stage. We then feed the encoded features through the RNN, and run a 2D CNN with the same structure as the detection network's decoder. on the hidden state output. The output of the decoder network was fed into a learned weight network with the exact same structure as in Faster VoxelPose [14].

We used 4 timesteps of input at training time, following the augmentation scheme of BEVFormer, and the forecasting output is 2 timesteps into the future, each 3 frames apart. At inference time, we only feed a single timestep of input

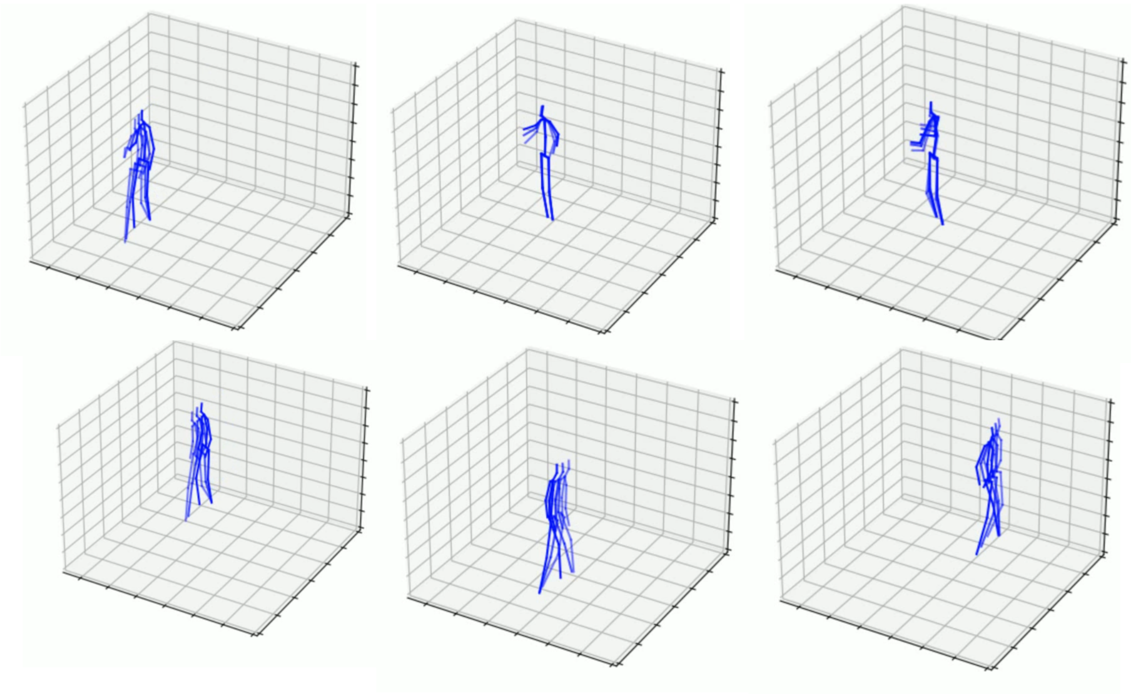

Figure 1. Sample forecasting outputs on the Human3.6M dataset. Our model produces feasible forecasts up to 0.33 seconds into the future, surpassing the accuracy of comparable works [15].

into the network, and TEMPO saves the previous embedding features, matching them to detections at each timestep with the tracker.

**Training Details** For the ResNet backbone, we trained the entire network jointly to convergence for 10 epochs with a batch size of 1. We used the Adam optimizer with weight decay 1e-4, learning rate 1e-4, and applied a linear decay schedule with $\gamma = 0.7$, updating every 2 epochs. We used a batch size of 2 and trained the network for 20 epochs, and used a learning rate of 5e-4, with all other parameters the same. For the Panoptic, Human3.6M, and DynAct datasets, we used images as input, while for the Shelf and Campus dataset we followed the scheme of [11, 14, 9, 12] and used synthetic joint heatmaps, produced by projecting ground-truth poses from the Panoptic dataset onto the cameras in the Campus and Shelf dataset.

# 2. Additional Ablation Details

## 2.1. Cross-dataset Generalization

Although TEMPO is not explicitly designed to provide strong generalization across multi-view datasets, we found that simply computing the space and volume dimensions from the camera configuration, it was able to transfer surprisingly well. In Table 2, we show that TEMPO exceeds both VoxelPose and Faster VoxelPose in this regard. Furthermore, TEMPO significantly exceeds the performance of MVPose [2], a method that is based on graph optimization and is dataset-agnostic by design, underscoring the strength of volumetric pose estimation methods.

## 2.2. Inference Time

We conducted a more detailed inference time analysis, comparing our work with Faster VoxelPose [14], the current fastest method. Our results are shown in Table 1. In the main text, we follow the convention of [14, 12, 6] and omit the runtime of the image backbone. We include it here for a full picture of our method's speed. Since the image backbone time is dependent on the number of views, we used 5 views for testing, in line with the Panoptic Studio dataset. We benchmarked all our models on a Nvidia A-100 with a AMD EPYC 7352 24-Core Processor @ 2.3GHz CPU.

For each module in our model, we provide the inference time, GFLOPs, and number of parameters. Since both ours and Faster VoxelPose are top-down methods, the GFLOPs and runtime vary with the number of detections. In this analysis, we used 3 detections for both methods.

# 3. Sample Visualizations

We provide both sample visualizations of TEMPO's forecasting output on the Human3.6M dataset. In Figure 1 we show representative visualizations of the model's forecasting output on the Walking sequence of the Human3.6M validation set.

## References

[1] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. https://github.com/open-mmlab/mmpose, 2020.

[2] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021.

[5] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *International Conference on Computer Vision (ICCV)*, 2019.

[6] Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11886–11895, 2021.

[7] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[8] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[9] N Dinesh Reddy, Laurent Guigues, Leonid Pischulin, Jayan Eledath, and Srinivasa G. Narasimhan. Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *European Conference on Computer Vision (ECCV)*, 2020.

[12] Tao Wang, Jianfeng Zhang, Yujun Cai, Shuicheng Yan, and Jiashi Feng. Direct multi-view multi-person 3d human pose estimation. *Advances in Neural Information Processing Systems*, 2021.

[13] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

[14] Hang Ye, Wentao Zhu, Chunyu Wang, Rujie Wu, and Yizhou Wang. Faster voxelpose: Real-time 3d human pose estimation by orthographic projection. In *European Conference on Computer Vision (ECCV)*, 2022.

[15] Shihao Zou, Yuanlu Xu, Chao Li, Lingni Ma, Li Cheng, and Minh Vo. Snipper: A spatiotemporal transformer for simultaneous multi-person 3d pose estimation tracking and forecasting on a video snippet. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.