Best Practices for 2-Body Pose Forecasting
– Supplementary Material –

Muhammad Rameez Ur Rahman[*,1] Luca Scofano[*,2] Edoardo De Matteis[1]
Alessandro Flaborea[1] Alessio Sampieri[2]
Fabio Galasso[1]
Sapienza University of Rome, Italy
[1]{rahman, dematteis, flaborea, galasso}@di.uniroma1.it
[2]{scofano, sampieri}@diag.uniroma1.it

We supplement the main paper submission with an additional video, the source code for the proposed best practices, and the supplementary material in this document. The supplementary material is organized according to the following table of contents.

Contents

1. Proof on initialization

Here we provide more detailed proof for Eqs. 10-11, 13-14 of the main paper. At each layer l, we assume learnable matrices $W^l \in \mathbb{R}^{C\times C'}$, $A_s^l \in \mathbb{R}^{T\times 2J\times 2J}$ and $A_t^l \in \mathbb{R}^{2J\times T\times T}$ to be independent, have zero mean and be uniformly distributed. With T being the number of timeframes, J being the number of joints in one person, and C and C' being the number of input and output channels.

First, we review and demonstrate the proposed initialization for the forward (Sec. 1.1) and backward passes (Sec. 1.2). Then, in Sec. 1.3, we illustrate how the initialization results in better training robustness.

1.1. Forward propagation

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to encode the body kinematics, with all joints at all observed frames as the $2J \times T$ nodes defining the vertex set V, and edges $\epsilon \in \mathcal{E}$ connecting them.

Following up on Eq. (4) from the main paper, the response of a separable GCN [13] layer is

$$\begin{cases} Y^l = A_s^l A_t^l X^l W^l \\ X^l = \sigma\left(Y^{l-1}\right), \end{cases} \tag{1}$$

where $X \in \mathbb{R}^{T\times 2J\times C}$ is the C-dimensional embedding of each node. W may be interpreted as a fully connected layer acting on each of the graph node embeddings separately, i.e., on each of the joints from the two people at all times, for a total of $2J \cdot T$ connections. W may be assumed to have C' neurons, i.e. to output $n = C'$ neural activations per node. The matrices A_s and A_t act on the spatial and temporal number of connections of the graph, respectively (please also see [13] for more details). Specifically, A_s may be considered to model the interaction of each node with all $2J$ others at the same frame, by means of $n_v = 2J$ neurons. Correspondingly, we may consider A_t to model the interaction of each node with those of the same joint at all T times, by means of $n_t = T$ neurons.

The number of interactions corresponds to the number of terms that are summed. Assuming matrices to be i.i.d. [6], the variance of the sum yields the sum of variances, thus

$$Var\left[Y^l\right] = n^l n_v^l n_t^l Var\left[A_s^l A_t^l X^l W^l\right]. \tag{2}$$

Assuming A_s^l, A_t^l, and W^l to have zero mean [6], the variance of the product of independent variables is

$$Var\left[Y^l\right] = n^l n_v^l n_t^l Var\left[A_s^l\right] Var\left[A_t^l\right] \mathbb{E}\left[\left(X^l\right)^2\right] Var\left[W^l\right]. \tag{3}$$

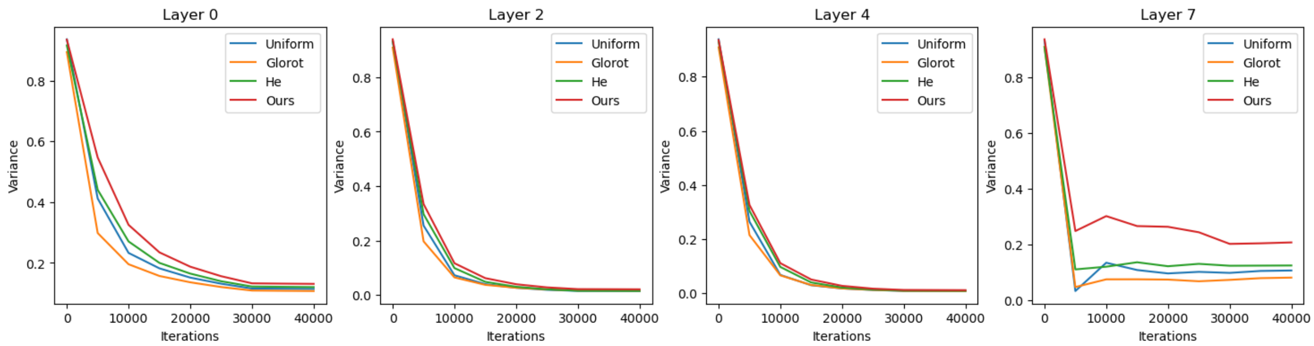

Figure 1. Comparison of feature activation variances, at layers 0, 2, 4 and 7, estimated during the model training, upon initialization with random “Uniform”, “Glorot” [3], “He” [6] against “Ours”, our proposed initialization technique.

We consider the PReLU as our activation function, i.e.

$$\sigma\left(X^{l}\right)=\max\left(0, Y^{l-1}\right)+a \min\left(0, Y^{l-1}\right), \tag{4}$$

with a being a learnable parameter that, when set to 0, reduces to the ReLU[1]. This means that for a generic a, $\mathbb{E}\left[X^{l}\right] \neq 0$. Let A_s^{l-1}, A_t^{l-1}, and W^{l-1} have symmetric zero-centered distributions [6]. This may then be also implied for Y^{l-1}, and we may write

$$\mathbb{E}\left[\left(X^{l}\right)^{2}\right]=\frac{1+a^{2}}{2} Var\left[Y^{l-1}\right]. \tag{5}$$

Substituting for Eq. (5) in Eq. (3) we get

$$\begin{aligned} Var\left[Y^{l}\right]=\frac{1+a^{2}}{2} n^{l} n_{v}^{l} n_{t}^{l} Var\left[A_{s}^{l}\right] Var\left[A_{t}^{l}\right] \\ Var\left[Y^{l-1}\right] Var\left[W^{l}\right]. \end{aligned} \tag{6}$$

Considering L layers, this yields the following variance formulation for the entire separable GCN model:

$$\begin{aligned} Var\left[Y^{L}\right]=Var\left[Y^{1}\right] \prod_{l=2}^{L} \frac{1+a^{2}}{2} n^{l} n_{v}^{l} n_{t}^{l} \\ Var\left[A_{s}^{l}\right] Var\left[A_{t}^{l}\right] Var\left[W^{l}\right]. \end{aligned} \tag{7}$$

In order to have the same input and output signal variance for the entire model, it suffices to assume that each layer l has the same input and output signal variances. This corresponds to setting the variance induced by the multiplicative parameters to be 1 i.e.,

$$\frac{1+a^{2}}{2} n^{l} n_{v}^{l} n_{t}^{l} Var\left[A_{s}^{l}\right] Var\left[A_{t}^{l}\right] Var\left[W^{l}\right]=1. \tag{8}$$

Towards this goal, it suffices to set each parameter initialization variance as follows

$$\frac{1+a^{2}}{2} n_{v}^{l} Var\left[A_{s}^{l}\right]=1 \tag{9}$$

$$\frac{1+a^{2}}{2} n_{t}^{l} Var\left[A_{t}^{l}\right]=1 \tag{10}$$

$$\frac{1+a^{2}}{2} n^{l} Var\left[W^{l}\right]=1 \tag{11}$$

[1] Also recall that a small a e.g., 0.01, is the LeakyRelu and $a=1$ is the linear case.

1.2. Backward propagation

The gradient of a separable GCN is

$$\begin{cases} \frac{\partial L}{\partial X^{l}}=A_{s}^{l} A_{t}^{l} \frac{\partial L}{\partial Y^{l}} \tilde{W}^{l} \\ \frac{\partial L}{\partial Y^{l}}=\frac{d \sigma}{d Y^{l}} \frac{\partial L}{\partial X^{l+1}}, \end{cases} \tag{12}$$

with $\tilde{W} \in \mathbb{R}^{C' \times C}$, while A_s and A_t have the same dimensionality as in the forward pass. Our backward response number is $\tilde{n}^l = C$ for $\tilde{W}$, and it is still n_v and n_t for A_s and A_t, respectively, thus

$$Var\left[\frac{\partial L}{\partial X^{l}}\right]=n_{v}^{l} n_{t}^{l} \tilde{n}^{l} Var\left[A_{s}^{l} A_{t}^{l} \frac{\partial L}{\partial Y^{l}} \tilde{W}^{l}\right]. \tag{13}$$

We let A_s^l, A_t^l, $\tilde{W}^l$ and $\frac{\partial L}{\partial Y^l}$ be independent. Let us assume A_s^l, A_t^l, and $\tilde{W}^l$'s to be zero-centered symmetric distributions, and $\frac{\partial L}{\partial X^l}$ to have zero mean [6]. Similarly to the forward pass, we have to consider the PReLU activation function. If we assume that $\frac{d\sigma}{dY^l}$ and $\frac{\partial L}{\partial X^{l+1}}$ are independent [6], we get

$$\mathbb{E}\left[\frac{\partial L}{\partial Y^{l}}\right]=\frac{1+a^{2}}{2} \mathbb{E}\left[\frac{\partial L}{\partial X^{l+1}}\right]=0, \tag{14}$$

$$\mathbb{E}\left[\left(\frac{\partial L}{\partial Y^{l}}\right)^{2}\right]=\frac{1+a^{2}}{2} Var\left[\frac{\partial L}{\partial X^{l+1}}\right]. \tag{15}$$

Considering Eq. (13) and the assumed independence, we

Action	A1				A2				A3				A4				A5				A6				A7				Average ↓			
Time (msec)	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*
LTD [10]	51	92	116	132	51	91	116	148	43	80	103	130	38	70	89	111	39	70	90	116	42	75	94	123	52	101	139	198	45	83	107	137
HisRep [9]	34	69	97	130	44	84	115	150	32	65	91	121	27	56	82	112	28	58	85	121	34	66	88	115	42	83	120	171	34	69	97	131
MSR-GCN [2]	41	75	99	126	54	96	129	180	41	74	98	135	34	61	82	106	33	59	79	109	42	71	93	124	57	103	146	210	43	77	104	141
MRT [14]	34	69	95	128	39	78	106	142	30	59	83	115	28	57	79	110	28	57	79	108	34	68	91	120	39	80	114	160	33	67	92	126
siMLPe [5]	32	69	94	115	44	93	122	160	33	73	102	138	26	61	87	114	28	60	84	112	32	69	93	123	45	94	127	171	34	74	101	133
XIA [4]	32	68	99	128	41	82	116	163	29	58	84	116	24	50	73	96	24	51	75	109	31	62	86	114	41	81	115	160	32	65	93	127
Ours	**24**	**51**	**76**	**114**	**31**	**66**	**93**	**132**	**23**	**49**	**70**	**103**	**19**	**41**	**60**	**85**	**21**	**44**	**64**	**93**	**24**	**52**	**73**	**100**	**29**	**64**	**95**	**143**	**24**	**52**	**76**	**110**

Table 1. Results in millimeters for ExPI Common actions split. Our model achieves state-of-the-art results in all actions considered, at each predicted time instant.

Action	A8			A9			A10			A11			A12			A13			A14			A15			A16			Average ↓		
Time (msec)	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*	*400*	*600*	*800*
LTD [10]	106	136	155	91	119	135	72	96	116	95	123	146	85	106	116	74	91	101	86	115	137	98	125	134	85	110	124	88	113	129
HisRep [9]	86	120	142	73	104	128	54	82	104	101	144	476	61	82	94	49	67	80	73	105	129	53	73	86	64	89	104	68	96	116
MSR-GCN [2]	88	118	142	90	113	136	90	122	148	103	134	155	101	135	160	74	98	121	103	143	173	87	111	132	84	106	122	91	120	143
MRT [14]	89	121	161	79	108	145	69	100	147	97	133	174	71	96	127	66	88	117	83	113	149	72	98	132	67	92	121	77	105	141
siMLPe [5]	95	125	141	82	114	134	63	93	115	124	174	212	61	80	92	50	67	79	83	116	138	59	81	90	72	99	116	77	106	124
XIA [4]	82	116	142	69	97	120	**52**	79	104	95	137	171	58	80	93	51	70	**84**	70	105	134	**53**	**73**	**88**	63	88	104	66	94	**116**
Ours	**68**	**95**	**115**	**66**	**95**	**116**	**52**	**78**	**103**	**86**	**124**	**150**	**54**	**76**	**91**	**47**	**68**	**84**	**59**	**86**	**108**	**53**	77	94	**53**	**77**	**94**	**60**	**86**	121

Table 2. Results in millimeters for ExPI Unseen actions split. On average, we outperform the baseline considered over short and long time horizons.

Action	A1				A2				A3				A4				A5				A6				A7			
Time (msec)	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*	*200*	*400*	*600*	*1000*
LTD [10]	51	99	129	163	61	110	150	229	53	96	131	188	46	81	106	142	44	79	106	147	53	100	162	176	70	133	163	198
HisRep [9]	51	93	114	127	51	91	116	162	43	80	100	126	38	70	88	118	39	70	90	125	42	75	93	123	52	101	137	188
MSR-GCN [2]	45	83	106	118	57	102	135	178	39	72	100	132	41	77	103	119	35	70	97	125	46	82	107	137	48	90	121	169
MRT [14]	36	69	93	123	44	81	106	138	41	76	96	114	30	61	81	105	33	64	88	121	34	64	83	104	42	83	114	157
siMLPe [5]	43	84	107	137	55	107	142	182	47	91	120	164	39	76	101	129	38	75	99	128	47	90	118	150	58	110	150	197
XIA [4]	43	84	115	131	53	99	136	185	35	68	98	140	37	74	106	128	29	59	86	125	39	72	94	119	43	82	112	**152**
Ours	**34**	**63**	**86**	**115**	**41**	**79**	**105**	**138**	**27**	**55**	**77**	**110**	**31**	**64**	**88**	**119**	**27**	**55**	**77**	**107**	**30**	**58**	**78**	**103**	**38**	**78**	**109**	154

Table 3. Results in millimeters for ExPI Single actions split. We outperform in 6 out of 7 stocks all baselines considered according to the MPJPE metric. For the other stocks our model is comparable with the current state of the art.

elaborate on Eq. (15) as follows

$$Var\left[\frac{\partial L}{\partial X^l}\right] = n_v^l n_t^l \tilde{n}^l Var\left[A_s^l\right] Var\left[A_t^l\right] \quad (16)$$
$$Var\left[\frac{\partial L}{\partial Y^l}\right] Var\left[\tilde{W}^l\right],$$
$$= \frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l Var\left[A_s^l\right] Var\left[A_t^l\right] \quad (17)$$
$$Var\left[\frac{\partial L}{\partial X^{l+1}}\right] Var\left[\tilde{W}^l\right].$$

For L layers, this yields

$$Var\left[X^2\right] = Var\left[\frac{\partial L}{\partial X^{L+1}}\right] \prod_{l=2}^{L} \frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l \quad (18)$$
$$Var\left[A_s^l\right] Var\left[A_t^l\right] Var\left[\tilde{W}^l\right].$$

In order to avoid exploding and vanishing gradients, a sufficient condition is to set the gradient of each layer to maintain the signal variance throughout the backpropagation

$$\frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l Var\left[A_s^l\right] Var\left[A_t^l\right] Var\left[\tilde{W}^l\right] = 1. \quad (19)$$

Finally, it suffices to set the variance initialization of each of the parameter matrices as follows:

$$\frac{1+a^2}{2} n_v^l Var\left[A_s^l\right] = 1 \quad (20)$$
$$\frac{1+a^2}{2} n_t^l Var\left[A_t^l\right] = 1 \quad (21)$$
$$\frac{1+a^2}{2} \tilde{n}^l Var\left[\tilde{W}^l\right] = 1 \quad (22)$$

Note that the result in Eq. (19) resonates with what was obtained for the forward pass, in Eq. (8). This ensures that the same initialization may be adopted to yield the signal and gradient requirements both in the forward and backward passes.

1.3. Training variance

The model output should maintain unit variance during training for the sake of activation functions and robust training, avoiding vanishing gradient [3, 6]. This becomes more challenging when adopting deeper networks, which regards our case, as we consider a separable-GCN twice as deep as the original one of [13] (8 Vs. 4 layers).

In Fig. 1, we consider the variances of the feature activations at sampled layers during training and compare the result of our proposed initialization against the "Uniform", "Glorot" [3], and "He" [6] techniques. Observe how "Ours" yields feature variances that are consistently closer to the desired unit variance. This is especially true for the last layer (layer 7), arguably the most challenging.

2. AME results

Following [4], we also evaluate our model using a different metric for the error between poses, the *Aligned Mean per joint position Error* (AME). Both poses are independently normalized in advance to avoid positional errors, to correct errors due to having a root joint as the origin, we use a rigid alignment transformation T.

$$L_{\text{AME}} = \frac{1}{V}\sum_{v=1}^{V} ||\hat{n}_{vt} - T(\hat{n}_{vt}, n_{vt})||_2, \tag{23}$$

where n is the coordinate x after normalization as defined above. The results are reported in Tab. 1, Tab. 3 and Tab. 2. Results are consistent with those reported in the main paper, expressed in terms of *Mean per joint position Error*(*MPJPE*), as it favors comparability with other works [1, 2, 5, 9–13]

3. Implementation details

In this section, we thoroughly describe the implementation procedures that we have used in training and testing. Furthermore, we describe the iterative approach used during testing.

3.1. Training and testing details

We use 10 frames as input and 10 frames as output during training. We use an iterative mechanism at test time to make a 1-second prediction(25 frames). We exclusively use our predictions as input for subsequent iterations. We extensively analyze the iterative mechanism impact in the supplementary materials. We adopt the ADAM [8] optimizer and a learning rate of 1×10^{-5}, decayed to 5×10^{-8} after 30K iterations. The model converges in 40K iterations, i.e., the training takes 23 min on a single Nvidia P6000 GPU. We also get an average prediction time on CPU[2] of 0.07 seconds compared with the fastest [4]'s 0.4. At each layer, we adopt batch normalization [7] and residual connections.

[2]An AMD Ryzen 5 3600 6-Core processor.

3.2. Iterative approach

We test different combinations of T input frames and N output frames (See Tab. 4) and notice that some perform better than others. Most notable works in pose forecasting [4, 5, 13] use $T = 50$ input frames and $N = 10$ output ones. Conversely, the best results are obtained using $T = 10$ a mechanism [4, 9, 10] that iteratively feeds both parts of the observed history and new predictions as an input.

			MPJPE (ms)↓			
	Input Frames	*Output Frames*	*200*	*400*	*600*	*1000*
	50	1	43	113	176	274
	50	5	52	114	162	243
	50	10	62	126	174	243
	50	50	68	131	177	244
	10	1	**36**	98	164	294
	10	5	37	89	141	238
Used	10	10	39	**86**	**129**	**202**

Table 4. Results in millimeters on the ExPI dataset, on average common actions split. We show the impact that different combinations of input-output frames have on performance. Using 10 input frames makes predictions in the short term more accurate, helping results to be more stable in the long term.

4. Complete list of actions

This section lists (ref. Tab. 5) each action A_i, with $i = 1, ..., 16$. Refer to the supplementary material in [4] for a more detailed explanation.

Action	Name
A_1	A-frame
A_2	Around the back
A_3	Coochie
A_4	Frog classic
A_5	Noser
A_6	Toss out
A_7	Cartwheel
A_8	Back flip
A_9	Big ben
A_{10}	Chandelle
A_{11}	Check the challenge
A_{12}	Frog-turn
A_{13}	Twisted toss
A_{14}	Crunch-toast
A_{15}	Frog-kick
A_{16}	Ninja-kick

Table 5. List of actions and their corresponding names

Actions $A_1, ..., A_7$ are performed by both couples $\mathcal{A}_1$ and $\mathcal{A}_2$. Actions $A_8, ..., A_{13}$ are exclusive to couple $\mathcal{A}_1$, and actions $A_{14}, ..., A_{16}$ to couple $\mathcal{A}_2$.

5. Sample videos

In addition, we include a video comparing the results of our and the current SoA's model [4] qualitatively. It is possible to see how our model is far more accurate when analyzing both basic and complex activities. For comparison, we use the pre-trained model provided by [4] and showcase only 10 of their 50 input frames to make it the same length as ours. Still, the number of output frames remains at 25 for both models. We release videos on our project page at https://www.pinlab.org/bestpractices2body.

References

[1] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. Motionmixer: Mlp-based 3d human body pose forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 2022. 4

[2] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3, 4

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9. PMLR, 2010. 2, 4

[4] Wen Guo, Xiaoyu Bie, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Multi-person extreme motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 4, 5

[5] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Alameda-Pineda Xavier, and Moreno-Noguer Francesc. Back to mlp: A simple baseline for human motion prediction. *arXiv preprint arXiv:2207.01567*, 2022. 3, 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015. 1, 2, 4

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015. 4

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015. 4

[9] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3, 4

[10] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 4

[11] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Multi-level motion attention for human motion prediction. *International Journal of Computer Vision*, 129(9):2513–2535, 2021. 4

[12] Alessio Sampieri, Guido Maria D'Amely di Melendugno, Andrea Avogaro, Federico Cunico, Francesco Setti, Geri Skenderi, Marco Cristani, and Fabio Galasso. Pose forecasting in industrial human-robot collaboration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 4

[13] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-time-separable graph convolutional network for pose forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 4

[14] Jiashun Wang, Huazhe Xu, Medhini Narasimhan, and Xiaolong Wang. Multi-person 3d motion prediction with multi-range transformers, 2021. 3