# Supplementary Material for Motion Diversification Networks


## Abstract

*In this supplementary material, we first provide additional **implementation details**, including network architecture and training details, in Sec. 1. In Sec. 2, we present details regarding the **experimental setup** used for investigating how well realistic and in-the-wild 3D human motion can be captured by our model. In Sec. 3, we provide additional **ablation analysis** regarding underlying model components and **qualitative results** depicting the diverse and expressive motion produced by our final model. Additional visualizations with synthesized animations are shown in the **supplementary video**.*


## 1. Implementation

### 1.1. z-Transformer

In our work, we propose a simple and effective mechanism for learning to predict diverse 3D human motion in context. We leverage an attention-based architecture to transform a set of sampled random variables and guiding context information to a diverse set of latent codes, subsequently decoded into a predicted set of 3D human motions. We leverage **eight transformer layers** with positional encoding input [10, 40]. Each transformer layer updates the input query $\mathbf{Z}^0 = [\boldsymbol{\epsilon}_1; \boldsymbol{\epsilon}_2; \ldots; \boldsymbol{\epsilon}_K]$, where $\boldsymbol{\epsilon}_i \in \mathbb{R}^{128}$, and all encoded context and motion primitives (following summation) $\mathbf{E}^0 \in \mathbb{R}^{K \times 128}$, through multi-head self-attention (MHSA), multi-head cross-attention (MHCA), and a multi-layer perceptron (MLP):

$$\mathbf{E}^l = \text{MHSA}(\mathbf{E}^l, \mathbf{E}^l, \mathbf{E}^l) + \mathbf{E}^l \tag{1}$$

$$\mathbf{Z}^l = \text{MHCA}(\mathbf{Z}^l, \mathbf{E}^l, \mathbf{E}^l) + \mathbf{E}^l \tag{2}$$

$$\mathbf{Z}^{l+1} = \text{MLP}(\mathbf{Z}^l) + \mathbf{Z}^l \tag{3}$$

where MHSA(q, k, v) and MHCA(q, k, v) computes self-attentions and cross-attentions between queries $q$, keys $k$, and values $v$, respectively [10, 40].Our MLP comprises two linear layers and Layernorm [43]. We add the learnable 1-D positional encoding to the context-aware input embedding (summation with $\mathbf{E}^0$. We note that while this provides a general mechanism for fusing motion primitives, latent codes, and contextual information, our key insight is to apply the transformer-based module to a ***set of sampled random variables*** which enables effectively modeling correlations among the modes during the diversification process.

### 1.2. Encoder and Decoder Architecture

As shown by the overview figure in the main paper, our motion generation network assumes an encoder, for processing input observations and context, and a decoder, for mapping latent codes outputted by the transformer-based module to future motion, i.e., 3D pose and 2D path. The model is trained in two stages, first as a conditional variational autoencoder (CVAE) [18], then with the proposed diversity sampling function.

**Context Encoder:** Our encoder architecture is implemented similarly across the various types of contexts of human pose history, waypoints, image context, and social context. Each is processed using a dedicated GRU with 128 hidden units, followed by a two-layer MLP (with 300 and 200 hidden dimensions). The outputs are then summed and projected with a fully-connected (FC) layer to a final 128-dimensional vector which is taken as input by both the diversification module and the decoder (see Fig. 2 of the main paper). For ***image context encoding***, which captures interaction with surrounding context, we first process a padded person-centered window to extract a 1000-dimensional feature using a Swin Transformer [25] prior to inputting to the aforementioned dedicated GRU. Specifically, we leverage Swin-B [25] pre-trained on ImageNet-1K [9].

This backbone outperformed other backbones based on ResNet [16]. Leveraging other models, e.g., the larger SwinV2-B [26], did not result in further performance gains for our settings. We also note that our network architecture can be readily extended to integrate other types of contexts, e.g., 3D context with Bird's Eye View occupancy information [15, 28], yet as depth and occupancy information can be difficult to obtain in our dense monocular settings, we leave this for future work. Particularly for the in-the-wild YouTube dataset (also see our supplementary video), we find basic computer vision tasks, such as 2D pose estimation, to fail often. To facilitate capturing social relationships, in addition to the image-level context we also investigate the role of ***social context*** cues from nearby 3D skeletons. At each frame and for each pedestrian, we extract a social descriptor based on 3D poses of nearby people and whether they are in the same group or not (this variable obtained through analysis of tracked long-term trajectories, as will be discussed in Sec. 2). These per-frame descriptors are then inputted to a social context GRU. Fig. 1 depicts examples for the neighborhood over which social context is computed. Leveraging a scale-aware radius (0.6× height of the person's skeleton in the image) centered between the feet we are able to provide more explicit social cues as well as pool information over any nearby 3D pose interactions and groups. We also experimented with incorporating ego-motion cues (e.g., as in [49]), yet found no additional benefits (the aforementioned cues can already encode such spatio-temporal context).

**Branched Decoder:** For our full model (evaluated with image context in DenseCity), we find it beneficial to incorporate 2D path prediction task, i.e., feet center in image coordinates over time [4, 29], in addition to 3D pose in root-normalized 3D coordinates output [47, 50]. While prior work predicts these tasks sequentially, e.g., Cao et al. [4, 37] predicts a set of 2D paths and subsequently a single 3D pose along each path, we find it beneficial to holistically model the two tasks with a branched decoder architecture. Each decoder leverages a GRU followed by an MLP with three FC layers (hidden dimensions of 300, 200, and 192—the output dimensionality of the 3D pose). We find the branched decoder architecture to improve overall sample diversity as well as 3D pose prediction along each trajectory. The predicted 2D path can also be used to predict root translations, i.e., for animation, using a ground plane homography in evaluation [37] (depicted in our supplementary video). We note that directly predicting 3D pose in absolute coordinates resulted in poor performance on our benchmark. Predicting full joint rotations directly by the model beyond just 3D skeleton (e.g., [31, 51]) was found to be similarly difficult in our settings. Instead, in our qualitative animation results we leverage the swing-twist decomposition of HybrIK [21, 44] for estimating body-part rotations and leave direct prediction of such quantities future work.

### 1.3. Motion Generation Network Training Details

**Training Protocol:** We optimize the model in two stages using Adam [17], first the encoder and decoder structure using standard CVAE training, and then training of the latent space diversification network leveraging a joint reconstruction and diversity objective, as mentioned in the main paper. For each stage, the initial learning rate ($lr$) is set to $1 \times 10^{-3}$ and decayed after 100 epochs using $lr = 0.001 \times (1.0 - \frac{\max(0,\ \text{epoch}-100)}{400})$. The model is trained for 800 and 500 epochs during the first and second training stage, respectively. The batch size is set to 100. We leverage $K = 50$ samples which we empirically found to work well for our motion prediction dataset, and leverage a $J = 65$ skeleton representation on our dataset (the standard parameterization in Unreal Engine and CARLA). For 3DPW and HPS we follow the same parameterization, and for Human3.6M we follow the standard parameterization of 17 joints [50]. On Human3.6M, we leverage the additional loss terms from Xu et al. [30, 47]. We did not find the additional loss terms to be beneficial on the other datasets. When training the model over both synthetic and real-world YouTube data (see Sec. 2), we simply mix the motion trajectories obtained from both datasets, sampled at equal amounts within each batch.

## 2. Data

Our chosen experimental setup of diversifying 3D motion predictions focuses on a challenging but important use-case in robotics of navigation in urban settings. We emphasize that while there are ample benchmarks for context-aware 2D path motion forecasting [2, 3, 6, 13, 19, 29, 34, 46] few benchmarks exist for diverse 3D human motion generation in urban settings. Moreover, prior benchmarks for context-aware 3D human motion modeling (e.g., GTA-IM [4], JtA [11]) only incorporate limited motion diversity in simplified conditions (e.g., without social settings and intricate layouts). Other benchmarks, e.g., 3DPW [41] and HPS [14], only offer a handful of relevant scenes. Thus, to investigate how well our model captures natural 3D human motion in context, we leverage synthetic and real-world datasets comprising complex and crowded scenes. We specifically focus on 360° videos in both real-world and synthetic scenes that afford evaluation of longer-term context-aware prediction (as shown in Fig. 3 of the main paper, observed paths can span the entire view).

Figure 1. **Example Social Context Computation.** To effectively capture social interactions in our model, we input the model with social cues from nearby skeletons. To compute the social context, we leverage a simple neighborhood-based descriptor that is defined over spatial area based on the pedestrian height ($0.6\times$). Examples are shown for two scenes from $360^\circ$ videos (from YouTube in this case). We also overly the results of the 2D pose estimation results, to demonstrate our challenging settings.

## 2.1. Simulation Benchmark

Despite being more realistic in motion compared to related benchmarks, e.g., Habitat [35], Gibson [1], the current built-in controller in the CARLA simulation produces 3D motion with limited diversity. Moreover, the highly-tuned path and articulated kinematics controller may fail in dense conditions, such as around objects and crowds (shown in our supplementary video). To fully explore the ability of our proposed model to learn to generate diverse motion, we further analyze the benefits of realistic in-the-wild human motion from $360^\circ$ YouTube videos, as discussed next. When generating DenseCity, we specifically matched the simulation settings with the type of view that would be captured in the YouTube videos (and further perform perspective augmentation during training, as further discussed below).

## 2.2. YouTube Dataset Construction

In this section, we describe our processing of $360^\circ$ YouTube videos, used to fully explore the ability of our proposed model to learn to generate diverse motion. The panoramic view facilitates sufficiently long motion trajectories and complete sur-

Figure 2. **Example Scenes From the YouTube Dataset.** We visualize additional frames from our data from Boston (top left), Tokyo (top right), Quebec (bottom left), and Seoul (bottom right).

rounding context. Example visualizations of the videos in the dataset are shown in Fig. 2

**Extracting Long-Term Motion Sequences from Unconstrained Video:** To extract 3D motion from unconstrained videos with dense crowds, high pose variability, and diverse human-human and human-object interactions, we employ a modular approach. The modular approach follows several state-of-the-art techniques [5, 12, 21, 23, 32, 38, 44, 52, 54] (we note that two-step approaches are often found to outperform direct 3D pose estimation [22, 31, 33, 39]). We first detect and track the 2D joint keypoints in the image [8, 12]. We leverage the off-the-self multi-person 2D keypoint detection and tracking algorithm AlphaPose [12, 20, 45], trained on the Halpe dataset [12] with a ResNet-50 backbone [16]. We subsequently lift the 2D joints to the normalized 3D space $\mathbf{X}$. For consistency, we always set the coordinate frame of the target 3D poses in training to be in a fixed camera height from the ground and zero pitch and roll. We find state-of-the-art 2D keypoint estimation and tracking models to frequently fail in our settings, i.e., due to density of people, occlusion, and scale variability, an issue we address below using a robust lifting module.

**Geometry-guided Lifting of 2D Keypoint Sequences:** Our lifting module leverages a geometry-aware spatio-temporal transformer-based architecture [10, 40, 48, 53] to map 2D poses to a root-normalized 3D pose. Information across the joints of a person within a single frame is first encoded using a spatial transformer layer, and then aggregated over multiple frames with a temporal transformer layer. The approach is similar to Zheng et al. [53], with a key difference of an added consistency term. We incorporate extensive data augmentation strategies in training from the initial lifting dataset to improve robustness, e.g., various camera views and noisy or dropped 2D keypoints. Our lifting training objective is defined as a weighted sum over 3D pose and bone symmetry consistency,

$$\mathcal{L}_{3D} = \mathcal{L}_{pose} + \lambda_s \mathcal{L}_{symmetry} \tag{4}$$

where the pose term is computed as the Mean Per Joint Position Error (MPJPE) between ground truth $\mathbf{X}_j[t]$ and estimated position $\hat{\mathbf{X}}_j[t]$ of the $j$-th joint, i.e., $\mathcal{L}_{pose} = \frac{1}{J}\sum_{j\in\mathcal{J}}||\mathbf{X}_j[t] - \hat{\mathbf{X}}_j[t]||_2$, where we dropped the time index $t$ for clarity. The symmetry loss is an unsupervised term, $\mathcal{L}_{symmetry} = \sum_{(l,r)\in\mathcal{J}}||\text{BL}(\hat{\mathbf{X}}_l[t]) - \text{BL}(\hat{\mathbf{X}}_r[t])||_2$, penalizing bone length (BL) differences between all corresponding left ($l$) and right ($r$) side body bone pairs. The symmetry term regularizes 3D

Figure 3. **Example Scenes from the User Study.** Participants watched two side by side animation and selected their preferred video in term of realism.

lifting under noisy cases and partial occlusion ($\lambda_s$ is a scalar hyperparameter, set as $10^{-4}$). We did not find additional physical constraint terms, e.g., bone length [7, 24], to provide further performance gains in our settings. Moreover, standard re-projection consistency terms [42] cannot be utilized due to the unknown camera intrinsic parameters.

### 2.3. User Study Details

Accurately evaluating natural and realistically diverse motion is challenging. In order to holistically evaluate the predicted motion generated from our model, we sample trajectories from it and compared it to the CARLA build-in trajectories. We then perform a user study where participants watched two adjacent animations and (1) selected their preferred animation out of the two, and (2) ranked each on a realism likert scale from one to five. Examples of the scenes can be seen in our supplementary video and in Fig. 3.

## 3. Additional Analysis

In this section, we provide additional model ablations (Sec. 3.1) and more qualitative results (Sec. 3.2).

### 3.1. Model Ablation

**Impact of Motion Primitives Guidance:** We first analyze the role of the motion primitives input to the z-transformer in Table 1. As shown, the motion primitives help improve diversity (by 9%) while similar (slightly higher ADE) accuracy is observed. We note that even a small increase in APD is considered significant [44].

**Effectiveness of Separate Decoder:** In this experiment, we investigate the effectiveness of our joint training of pose and path with branched decoder across various architecture of decoder. Specifically, we train our model on DenseCity with three

Table 1. **Impact of Motion Primitives on Human3.6M**. We show the benefits of incorporating motion primitives into the transformer-based module on Human3.6M.

| Method | APD ↑ | ADE ↓ | FDE ↓ |
|---|---|---|---|
| MDN w/o Primitives | 15.992 | **0.352** | 0.446 |
| MDN | **17.450** | 0.355 | **0.442** |

Table 2. **Decoder Architecture Ablations on DenseCity.** Compared to the sequential pipeline, where first 2D path is predicted followed by 3D pose [4, 37], we efficiently model the two tasks jointly. We find a slight decrease in FDE (as the 2D path acts as a dedicated goal-prediction module [29]), but a higher diversity and better ADE performance using the joint decoder architecture. We select the ***branched decoder architecture due to its high diversity and overall balanced performance***.

| | 3D Pose | | | 2D Path | | |
|---|---|---|---|---|---|---|
| Method | APD ↑ | ADE ↓ | FDE ↓ | APD ↑ | ADE ↓ | FDE ↓ |
| Sequential [4] (Separate 2D path and 3D pose) | **17.915** | 0.595 | 0.939 | **1.257** | 0.692 | 0.646 |
| Joint Training (Shared Decoder) | 17.092 | 0.698 | 1.060 | 0.533 | 0.723 | 0.722 |
| Joint Training (Branched Decoder) | 16.799 | **0.584** | **0.879** | 1.065 | **0.666** | **0.621** |

Table 3. **Evaluation on DenseCity.** Results are shown in terms of diversity (APD) and accuracy (ADE, FDE) against several baselines for both 3D pose and 2D trajectory error over a **three seconds future prediction** task.

| | 3D Pose | | | 2D Path | | |
|---|---|---|---|---|---|---|
| Method | APD ↑ | ADE ↓ | FDE ↓ | APD ↑ | ADE ↓ | FDE ↓ |
| CVAE [18] | 11.790 | 0.794 | 1.309 | - | - | - |
| PoseGPT [27] | 14.227 | 1.011 | 1.343 | - | - | - |
| HuMoR [36] | 13.234 | 1.038 | 1.551 | - | - | - |
| DLow [50] | 14.778 | 0.755 | 1.254 | - | - | - |
| Cao et al. [4] | 8.157 | 1.197 | 1.643 | 1.240 | 0.681 | 0.625 |
| MDN | 15.332 | **0.749** | 1.253 | 1.750 | **0.675** | **0.614** |
| Cao et al. [4]+YouTube | 6.818 | 0.956 | 1.390 | 1.135 | 0.693 | 0.639 |
| MDN+YouTube | **17.010** | 0.752 | **1.248** | **1.868** | **0.675** | 0.623 |

different decoder structures, including a sequential method with separate 2D path and 3D pose decoders, and two joint training methods, with a shared decoder and a branched decoder, respectively. As shown in Table 2, we observe higher diversity due to the efficient joint training architecture compared to sequential training. While the sequential predictor performs well in terms of APD, the two architectures are complementary, e.g., goal-driven predictions can be used to further inform our jointly trained model in the future [29]. Notably, the accuracy in 3D pose and the 2D path is significantly improved with the adoption of the branched decoder in the MDN.

**Three Seconds Future Prediction:** The analysis in the main paper follows a standard two-second future prediction window [47, 50]. To validate our model beyond the standard window, Table 3 depicts longer three-second window results. As shown in the table, overall errors of all baselines increase due to the more challenging prediction task, particularly towards the end of the trajectory. We can see how reasoning over 3D pose and 2D path for the longer time window can be difficult, resulting in an increase for 3D Pose ADE and FDE. When considering our ***main metric of 3D future pose***, we find our model to provide the best prediction accuracy, with a 0.749 ADE and 1.253 FDE, and diversity (15.332) across all models. Incorporating YouTube video data also improves diversity and 3D pose prediction accuracy further. For further study, through integration into an open-source simulation environment, we hope our articulated human motion model can contribute towards more realistic validation scenarios for computer vision and robotics research in the future.

## 3.2. Additional Qualitative Results

In Fig. 4 we provide additional qualitative results by visualizing the end poses using a $K = 7$ samples model. We demonstrate increased 3D motion diversity compared to baselines, including the CVAE and DLow baselines. We also observed improvements in pose diversity by incorporating real-world motion.

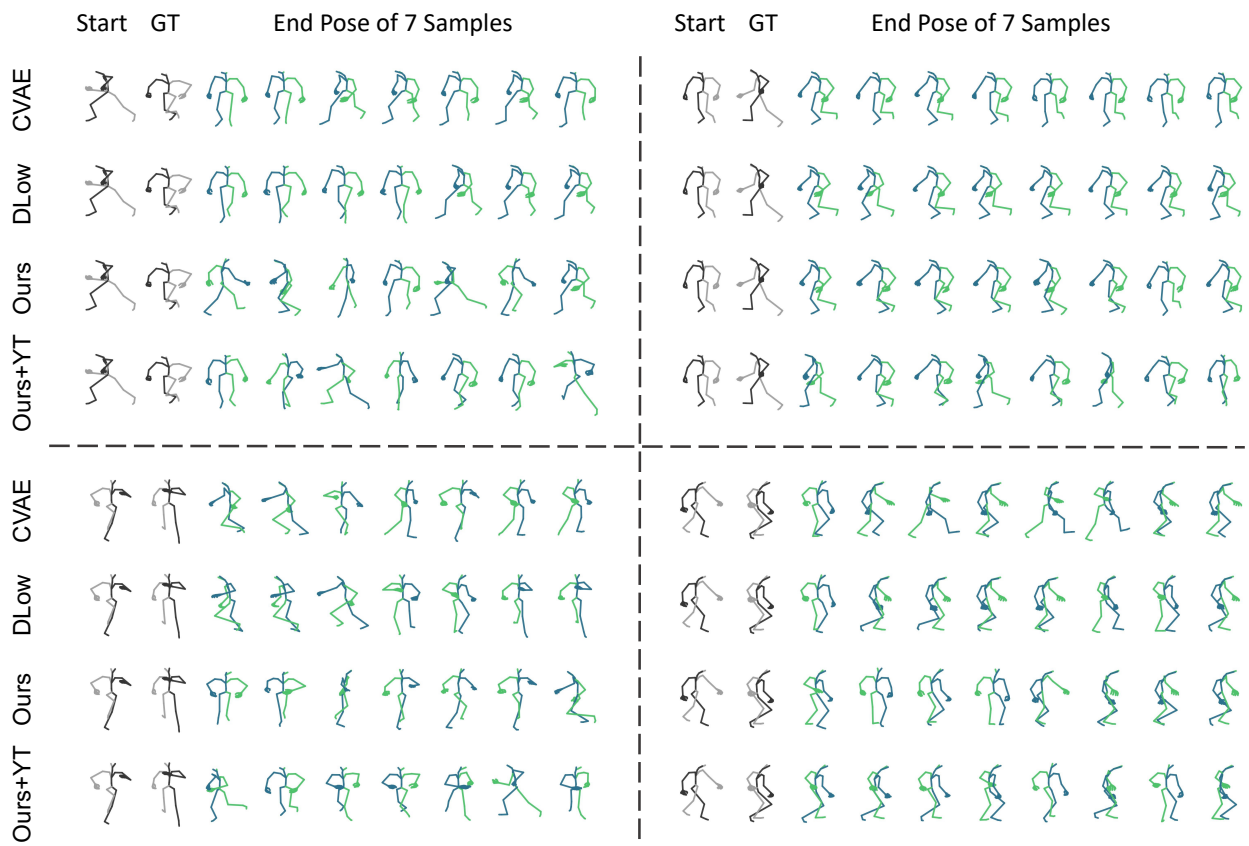
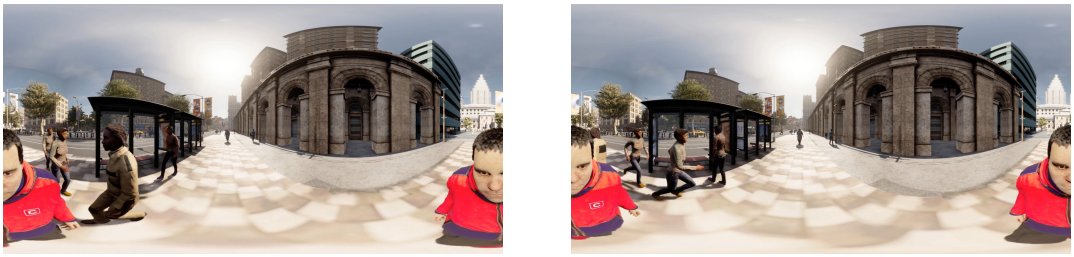



Figure 4. **Qualitative Comparisons on DenseCity.** We show diverse end-poses of our model compared to baselines. We further improve pose diversity by leveraging YouTube data (+ YT) during training.

**Failure Cases in CARLA:** To understand the limitations of current pedestrian controllers in simulation, Fig. 5 visualizes several failure cases in CARLA. As shown in the figure, while the kinematics controller performs reasonably over scenarios with little pedestrians, more dense settings can result in near-collisions, pedestrians getting stuck, and implausible scenarios. The failure cases can also be seen in our supplementary video, where our full MDN model is shown to resolve such cases.



(a) ***Unnatural turn*** shown next to the bus stop on the left due to multiple pedestrians at close proximity.

(b) **Collisions** may occur among pedestrians in dense scenes.

(c) **Physically implausible** scenarios due to two stranded pedestrians at close proximity.

Figure 5. **Qualitative Examples of Failure Cases with CARLA Built-in Pedestrian Controller.**

## References


[1] iGibson social navigation challenge. https://svl.stanford.edu/igibson/challenge.html, 2022. 3

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2

[4] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*, 2020. 2, 6

[5] Ching-Hang Chen and Deva Ramanan. 3D human pose estimation = 2D pose estimation + matching. In *CVPR*, 2017. 4

[6] Alexander Cui, Sergio Casas, Abbas Sadat, Renjie Liao, and Raquel Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *ICCV*, 2021. 2

[7] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *ECCV*, 2018. 5

[8] Qi Dang, Jianqin Yin, Bin Wang, and Wenqing Zheng. Deep learning based 2D human pose estimation: A survey. *Tsinghua Sci. Technol*, 2019. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1, 4

[11] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *ECCV*, 2018. 2

[12] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. AlphaPose: Whole-body regional multi-person pose estimation and tracking in real-time. *PAMI*, 2022. 4

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, 2018. 2

[14] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (HPS): 3D human pose estimation and self-localization in large scenes from body-mounted sensors. In *CVPR*, pages 4318–4329, 2021. 2

[15] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *ICCV*, 2019. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv.org, 1412.6980*, 2014. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv.org, 1312.6114*, 2013. 1, 6

[19] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer Graphics Forum*, 2007. 2

[20] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *CVPR*, 2019. 4

[21] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation. In *ICCV*, 2021. 2, 4

[22] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2015. 4

[23] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, and Pichao Wang. Lifting transformer for 3d human pose estimation in video. *arXiv preprint arXiv:2103.14304*, 2021. 4

[24] Zhi Li, Xuan Wang, Fei Wang, and Peilin Jiang. On boosting single-frame 3d human pose estimation via monocular videos. In *ICCV*, 2019. 5

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[26] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer V2: Scaling up capacity and resolution. In *CVPR*, 2022. 2

[27] Thomas Lucas, Fabien Baradel, Philippe Weinzaepfel, and Grégory Rogez. PoseGPT: quantization-based 3d human motion generation and forecasting. In *ECCV*, 2022. 6

[28] Zhengyi Luo, Shun Iwase, Ye Yuan, and Kris Kitani. Embodied scene-aware human pose estimation. *NeurIPS*, 2022. 2

[29] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *ECCV*, 2020. 2, 6

[30] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating smooth pose sequences for diverse human motion prediction. In *ICCV*, 2021. 2

[31] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 2, 4

[32] Qiang Nie, Ziwei Liu, and Yunhui Liu. Lifting 2D human pose to 3d with domain adapted 3D body concept. *IJCV*, 2023. 4

[33] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017. 4

[34] Stefano Pellegrini, Andreas Ess, and Luc Van Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010. 2

[35] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv.org, 2310.13724*, 2023. 3

[36] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. HuMoR: 3d human motion model for robust pose estimation. In *ICCV*, 2021. 6

[37] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *CVPR*, 2023. 2, 6

[38] István Sárándi, Alexander Hermans, and Bastian Leibe. Learning 3d human pose estimation from dozens of datasets using a geometry-aware autoencoder to bridge between skeleton formats. In *WACV*, 2023. 4

[39] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016. 4

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIIPS*, 2017. 1, 4

[41] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 2

[42] Bastian Wandt, James J Little, and Helge Rhodin. Elepose: Unsupervised 3d human pose estimation by predicting camera elevation and learning normalizing flows on 2d poses. In *ICCV*, 2022. 5

[43] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *ACL*, 2019. 1

[44] Kevin Xie, Tingwu Wang, Umar Iqbal, Yunrong Guo, Sanja Fidler, and Florian Shkurti. Physics-based human motion estimation and synthesis from videos. In *ICCV*, 2021. 2, 4, 5

[45] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose flow: Efficient online pose tracking. *CVPR*, 2018. 4

[46] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *ECCV*, 2022. 2

[47] Sirui Xu, Yu-Xiong Wang, and Liang-Yan Gui. Diverse human motion prediction guided by multi-level spatial-temporal anchors. In *ECCV*, 2022. 2, 6

[48] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *CVPR*, 2022. 4

[49] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *CVPR*, 2023. 2

[50] Ye Yuan and Kris Kitani. DLow: Diversifying latent flows for diverse human motion prediction. In *ECCV*, 2020. 2, 6

[51] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In *ICCV*, 2022. 2

[52] Andrei Zanfir, Mihai Zanfir, Alexander Gorban, Jingwei Ji, Yin Zhou, Dragomir Anguelov, and Cristian Sminchisescu. Hum3dil: Semi-supervised multi-modal 3d human pose estimation for autonomous driving. *CVPR*, 2022. 4

[53] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. In *ICCV*, 2021. 4

[54] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017. 4