Multi-agent Long-term 3D Human Pose Forecasting via Interaction-aware Trajectory Conditioning

Supplementary Material

1. More qualitative results

As shown in additional visualizations on CMU-Mocap (UMPM) dataset of Fig. 1, our t2p model generates more plausible motion for diverse behaviors. Indeed, our model was able to learn the specific physical mechanism of not only common motion such as walking forward (sequence 3, from top to bottom) but also unique behaviors such as punching (sequence 1) and walking backward (sequence 2). The competence of our model is more clearly exhibited at later timesteps. At timestep later than 1.5s, predictions of previous works more frequently show agents traversing in parallel feet, or manifesting physically unstable postures. On the other hand, our T2P model predicts physically natural pose sequences at these timesteps, depicting the effectiveness of our coarse-to-fine approach in making long-term predictions. As for the JRDB-GMP 2s/5s dataset, previous methods often predict motion that causes collision with other agents. Our method, on the other hand, effectively models agent interaction even in complex scenes and predicts plausible pose sequences as shown in Fig. 3

2. Detailed dataset specifications

2.1. Construction of JRDB-GMP dataset

Here, we elaborate on the details of constructing JRDB-GMP dataset. As explained in the main paper, we use SOTA method for monocular scale-aware 3D pose extraction [4] and both 2D pose annotation and 3D bounding boxes. Algorithm 1 outlines the overall process:

Where $p^{3D,X}_{t,i,n} \in P^{3D,X}_{t,i}$, $p^{2D,X}_{t,i,n} \in P^{2D,X}_{t,i}$ denotes extracted 3D pose and its corresponding 2D pose mapped to the camera coordinates. Also, $p^{2D,Y}_{t,i,n} \in P^{2D,Y}_{t,i}$ and $b^{Y}_{t,i,n} \in B^{Y}_{t,i}$ denote 2D pose annotation and 3D bounding box annotations. P^{3D}_{y} denotes 2D pose annotation projected in 3D world coordinates to represent viable proposals. τ denotes the 2D keypoint L2 distance threshold for detection filtering. $\phi_{closest}, \phi_{pose}, \phi_{center}$, and ϕ_{rotate} denotes algorithmic operations elaborates as below.

$\boldsymbol{\phi_{closest}}$ The extracted 3D poses inevitably contains noise in its samples, since they are extracted via an algorithm. We use 2D annotations to filter these outliers. Namely, if L2 distance between the annotation joints $p^{2D,X}_{t,i,n}$ and 2D-projected 3D pose is larger than τ in 2D camera coordinates, the detected pose $p^{3D,X}_{t,i,n}$ is not used.

$\boldsymbol{\phi_{pose}}$ Even after filtering the outliers, the dataset still contains some noise that influences the extracted 3D poses.

Algorithm 1 JRDB-GMP construction algorithm

for $t = 1, t{+}{+}$, while $t < T$ **do**
  for $i = 1, i{+}{+}$, while $i <= 6$ **do**
    Extract $P^{3D,X}_{t,i}$
    $N \leftarrow$ number of detected persons in$P^{3D,X}$
    for $n = 1, n{+}{+}$, while $n <= N$ **do**
      if $L2(p^{2D,X}_{t,i,n}, \phi_{closest}(p^{2D,X}_{t,i,n})) < \tau$ **then**
        $p^{3D,X}_{t,i,n} \leftarrow \phi_{pose}(p^{3D,X}_{t,i,n}, p^{3D,Y}_{t,i,n})$
        $p^{3D,X}_{t,i,n} \leftarrow \phi_{center}(p^{3D,X}_{t,i,n}, b^{Y}_{t,i,n})$
        $p^{3D,X}_{t,i,n} \leftarrow \phi_{rotate}(p^{3D,X}_{t,i,n})$
        if $p^{3D}_{t,i}$ is not registered **then**
          $\tilde{p}^{3D,X}_{t,i,n} \leftarrow$ register$(p^{3D,X}_{t,i,n})$
        end if
      end if
    end for
  end for
end for

Therefore, we use 2D pose annotations to minimize these temporal jitters. First, for each 2D pose of extracted 3D pose $p^{2D,X}_{t,i,n}$ of a person, we adjust the average x-axis value of its corresponding 2D annotation match $P^{2D,Y}_{t,i}$ to that of $p^{2D,X}_{t,i,n}$. In doing so, the subsequent process only refines the only the local pose while its global position remains stationary. The joint 3D position of an initially extracted 3D pose and its corresponding 2D camera coordinates of 2D pose annotation are given as (x, y, z) and (X, Y). Then, a 3D line is drawn from $(0, 0, z)$ to the 3D-projection of (X, Y), namely $inv(K) \cdot (X, Y, 1)$. This line represents the possible 3D coordinates of the joint, given a 2D annotation. The initially acquired 3D positions of the joint (x, y, z) are adjusted to its closest point to the previously drawn line. Such modification on each joint position corroborates the local accuracy of acquired 3D pose by minimizing the remnant noise via use of 2D pose annotation.

$\boldsymbol{\phi_{center}}$ The aforementioned process $\boldsymbol{\phi_{pose}}$ only adjusts local motion, requiring an additional step for adjusting its global position. For this purpose, the global x, y positions of $p^{2D,X}_{t,i,n}$ is adjusted into the central x_b, y_b positions of 3D bounding box annotation $b^{Y}_{t,i,n}$ of its match. This refinement puts the detected 3D joints in their accurate global position.

$\boldsymbol{\phi_{rotate}}$ 3D poses are acquired for each view of omnidirectional 5 cameras of the robot. Therefore, The acquired 3D poses of each view is rotated along the Z-axis to the

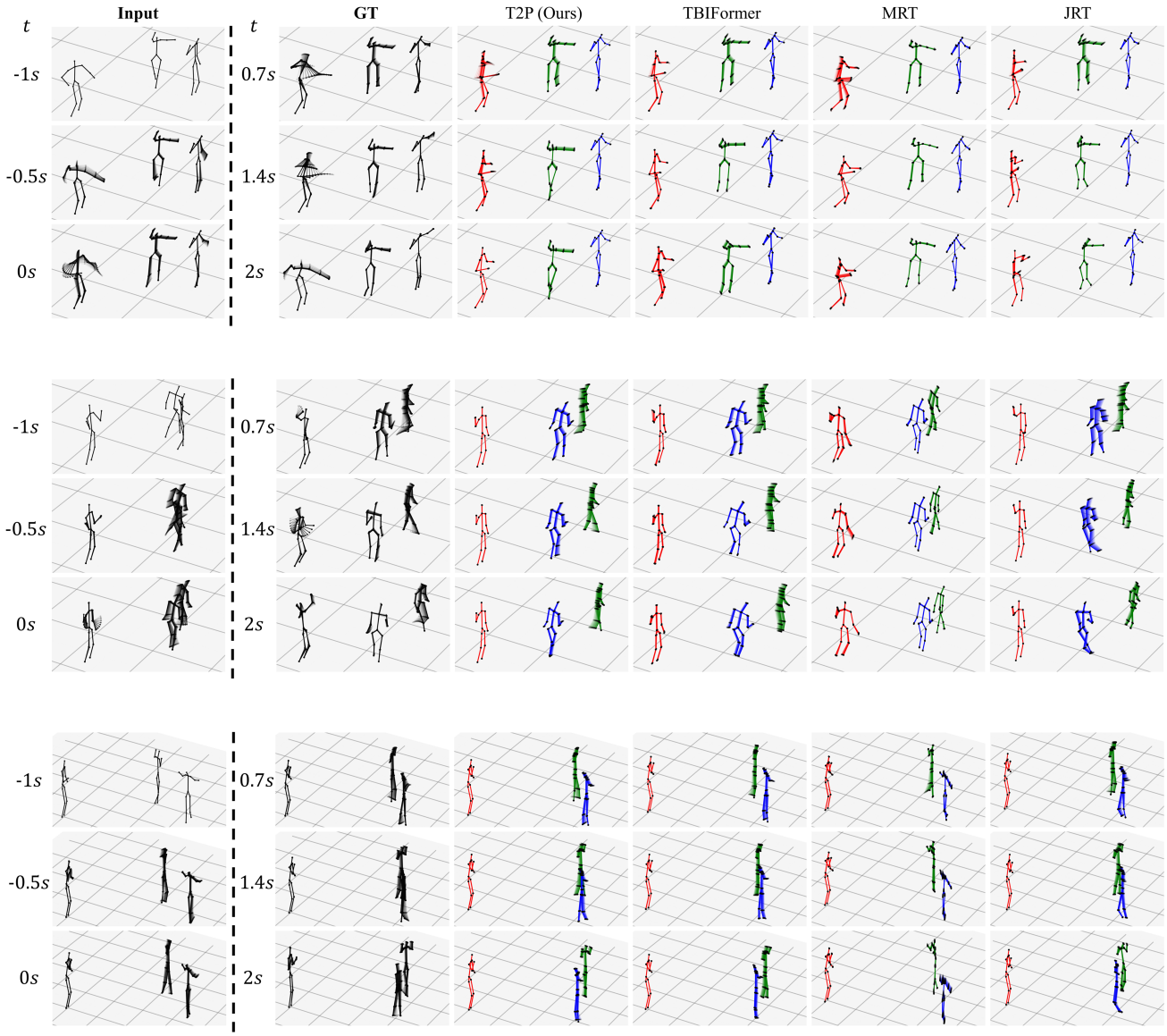

Figure 1. More visualization of predictions on CMU-Mocap (UMPM) dataset. Ours constantly show plausible and accurate predictions of human motion, especially during long-term motion. Best viewed when zoomed in.

front view, where Z-axis is where the robot is located. In doing so, the 3D poses of all visible agents in 360° of a given scene are accurately acquired in global coordinates.

2D pose annotation $P_{t,i}^{2D,Y}$ and 3D bounding box annotations $B_{t,i}^{Y}$ both include agent numbers, which we use to temporally track the acquired 3D poses. These agent IDs are used to register the refined pose to acquire the final 3D pose sequence $\tilde{p}_{t,i,n}^{3D,X} \in \tilde{P}^{3D,X}$. Only the agents within 4.5 meters from the robot are considered, as agents beyond this threshold contains more noise than closer agents. Also, we only consider sequences where at least 3 agents are present within 4.5 meters from the robot during the sequence of given scene. In order to ensure a sufficient amount of training data that prevents underfitting, we parse the 1s/2s dataset by every 15 frames, and 2s/5s by every 3 frames.

2.2. Visualization of JRDB-GMP dataset

Figure 2 visualizes example scenes from the JRDB dataset along with its extracted poses. JRDB scenes include both indoor and outdoor scenes, both of which include scenes with various number of agents as less and two and as many as 25. As depicted in the figure, 2D pose annotations are available for agents with slight occlusion, which we use to refine the extracted 3D pose. Agents with severe occlusion

Figure 2. Input images and 2D annotations (left column) used for extracting 3D poses (right column) on various indoor/outdoor scenes.

are not detected by the monocular 3D pose extraction algorithm [4]. As a result, the JRDB-GMP dataset contains real world sequences of 3D poses with minimal amount of noise, depicting a long-term multi-agent environment.

2.3. Limitation of JRDB-GMP dataset

Since JRDB dataset is collected in a monocular setting, JRDB-GMP inevitably incorporates noise due to a lack of geometric constraints when regressing the 3D poses. We have minimized these artifacts by refining the extracted 3D poses via 2D and 3D annotations as explained in the supplementary materials. Their GIF examples of supplementary materials demonstrate the resulting natural sequences. Despite such a pruning process, minimal noise remains with partially occluded bodies. Pose annotation was often not available for these occluded parts, limiting their refinement process. Nevertheless, our use of SoTA 3D pose extraction algorithm [4] and subsequent maximal refinement has constructed a reliable testbed for human motion forecasting at the current technology level. Certainly, we believe our acquisition strategy is well-suited and scalable in real-world scenes where IMU sensors are not available.

3. Elaboration on dataset scenes

JRDB dataset is comprised of 17 indoor (cafe, library, study room, etc) and 10 outdoor scenes (road, plaza, etc). JRDB-Act [2] scrutinizes the types and distribution of 26 actions included in the dataset. Its most common actions are standing, walking, sitting, and holding sth. In addition, specific actions like cycling, going upstairs, eating, skating, and greeting gestures are also included by a non-trivial amount. JRDB-GMP convers such varied representations of real-world behaviors. 3DPW is also collected in a real-world in-the-wild environment and includes diverse motions of two agents: dancing, riding bus, fencing, etc. CMU-Mocap (UMPM) dataset includes 140+ categories of motion: rolling, dancing, throwing, and more.

In terms of geological features, the venues in JRDB-GMP and 3DPW incorporate diverse geometrical constraints. Multi-agent interactions in such unique environments guide agents to traverse in diverse patterns. For example, desks and rails in cafe and library compose convoluted scenes that naturally guide disparate non-linear trajectories. CMU-Mocap (UMPM) also consists of varied locomotion modes since constructed to capture as diverse motion categories as possible. Therefore, the datasets account for diverse trajectories that incorporate real-world environments. Indeed, the trajectory-level human motion is already complex in a multi-agent environment, which is further convoluted by each local motion. Our coarse-to-fine approach competently handles such complex task.

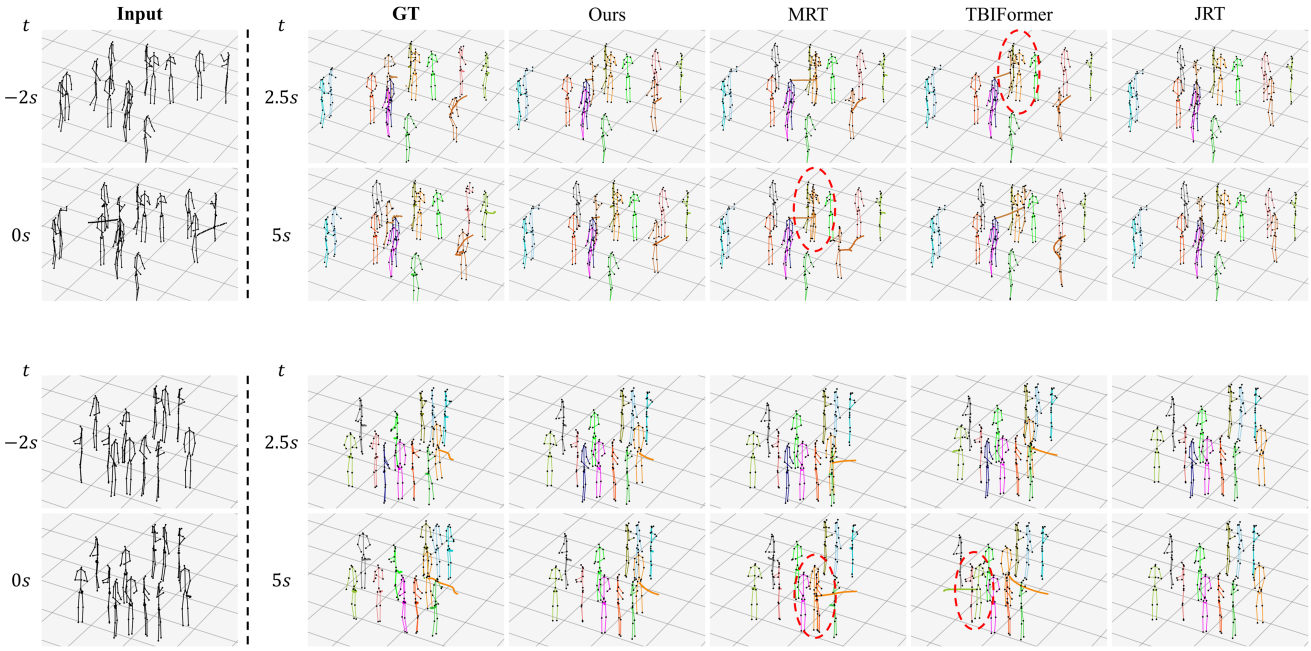

Figure 3. More visualization of predictions on JRDB-GMP 2s/5s. Unlike previous methods, our model predicts plausible future motion without collision with other agents. Best viewed when zoomed in.

4. Metrics

Out of F modes of prediction $\mathbf{Y}$, the mode with minimum JPE metric is selected as the mode to calculate all metrics.
APE: Aligned mean per joint Position Error is used as a metric to evaluate the forecasted local motion. $L2$ distance of each joint in the hip joint coordinate is averaged over all joints for a given timestep.

$$\text{APE}(Po_{\mathcal{F}}, \widehat{Po}_{\mathcal{F}}) = \frac{1}{N \times J} \sum_{i=1}^{N} \sum_{j=1}^{J} \parallel Po_{\mathcal{F}_{ji}} - \widehat{Po}_{\mathcal{F}_{ji}} \parallel^2 \tag{1}$$

FDE: Final Distance Error evaluates the forecasted global trajectory by calculating the hip joint $L2$ distance of a given timestep.

$$\text{FDE}(Tr_{\mathcal{F}}, \widehat{Tr}_{\mathcal{F}}) = \frac{1}{N} \sum_{i=1}^{N} \parallel Tr^i_{\mathcal{F}} - \widehat{Tr}^i_{\mathcal{F}} \parallel^2 \tag{2}$$

JPE: Joint Precision Error evaluates both global and local predictions by mean $L2$ distance of all joints for a timestep.

$$\text{JPE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{N \times J} \sum_{i=1}^{N} \sum_{j=1}^{J} \parallel \mathbf{Y}^i_j - \hat{\mathbf{Y}}^i_j \parallel^2 \tag{3}$$

Where N is the number of agents and J is the number of joints.

5. Model architecture

The model implemented with pytorch-lightning will be released publicly upon acceptance. We provide detailed description of the model as below:

5.1. Pose encoder

For DCT and MPBPConv layer, we adopt the same layer structure from TBIFormer [3] with final feature dimension of 128. Multi-head self-attention layer has 8 attention heads and input dimension of 128, with key/value/query dimension of 64. After the multi-head attention layer, feature is expanded to 128 with mlp and passes a dropout layer of dropout rate 0.2. The following FFN layer is composed of 2 linear layers of 128-1024-128, along with layer normalization with $\epsilon = 1e-6$, drop out layer of 0.2, and skip connection. The MSA and FFN layers are repeated $L = 2$ times.

5.2. Trajectory module

The trajectory module is composed of *aaencoder* from HiVT [7]. This module is composed of a message passing module which computes agent-wise interaction in a trans-rotation invariant manner at each time step. After passing the trajectory module, a feature is comprised of a latent with dimension 96 at each time step.

5.3. Traj-pose module

The pose embedding Z_{Po} of dimension 128 is reduced to dimension 96 with a single linear layer. Then, graph attention following Eq. 4 of the main manuscript is performed at every time step to get traj-pose embedding Z of dimension 96.

5.4. Trajectory decoder

Trajectory decoder is composed of temporal encoder, aggregator, and a MLP layer. The temporal encoder utilizes an extra learnable token of dimension 96 and temporal masks similar to BERT [1]. There are sequentially 4 transformer encoder layer in the temporal encoder. The aggregator is composed of a single message passing layer. During message passing, its node is 96-dimsensional traj-pose embedding, and its edge is a 96-dimensional edge feature which is the output of a single mlp layer that takes the concatenation of edge attributes such as relative position and relative rotation angle. The MLP layer is composed of two linear layers that gradually reduce the dimension to 192-96-future_timestep$\times$3. After first two linear layers, layer normalization and ReLU activation layers are followed.

5.5. Pose decoder

128-dimensional pose query is repeated num_mode times, and concatenated with 96-dimensional traj query. Then, a mlp layer reduces the concatenated query into 128 dimension. The decoder is composed of 2 layers of multi-head attention decoder layers and 3 fully connected linear layers. The multi-head attention decoder layers have the same structure as the multi-head attention encoder layers. After three fully connected linear layers, inverse DCT is performed to recover the output to the location.

6. Implementation details

For our T2P model, 2 layers of pose encoder transformer are stacked, followed by 2 layers of transformer in pose decoder. Embedding dimensions of 96 and 128 are used for trajectory and pose embeddings, respectively. The transformed key, value dimension of 64 is used for all transformer architectures. A learning rate of 0.003 is used with an AdamW optimizer with weight decay. We train our model on a single NVIDIA A6000 GPU. Further details are included in the released code.

7. More ablation study

7.1. Module ablation on CMU-Mocap (UMPM) dataset

We report further ablation of module components on CMU-Mocap (UMPM) dataset. Table. 1 reports the influence of core components of our trajectory encoder and pose decoder on CMU-Mocap (UMPM) dataset. We assess the significance of incorporating local pose embedding and modeling agent interaction in the trajectory encoder. Comparing experiments 1, 5, and 3, 4, all demonstrate enhancements in JPE and FDE metrics when utilizing local pose embedding. Our approach leverages detailed local pose cues to deduce an agent's global intention, with a more pronounced improvement in JPE and FDE observed when considering agent interaction (exp. 1, 5). Indeed, factoring in the interaction between local motion cues amplifies the advantages of predicting global motion intents for all agents. Regarding interaction modeling, its application proves advantageous for both global and local forecasts, as evident in experiments 4 and 6. These collective enhancements underscore the importance of accounting for local and global motion interactions in their respective forecasts. Turning to the pose decoder, comparisons between experiments 2, 4 and 5, 6 reveal consistent improvements in the APE metric. This consistent enhancement validates the efficacy of the trajectory-conditioned local motion forecast approach in generating plausible local motion from global intention.

Table 1. Ablation studies on core components of model structures. Experiments are done with CMU-Mocap (UMPM) dataset with 1s as given and 2s of forecasted horizon.

Exp. #	Trajectory encoder		Pose decoder	Metrics		
	Local pose embedding	Agent interaction	Trajectory -conditioning	JPE @5s	APE @5s	FDE @5s
-				290.9	160.8	228.5
1		✓		270.7	153.9	197.6
2	✓			275.2	154.9	200.8
3			✓	273.8	153.2	201.5
4	✓		✓	272.9	152.4	200.8
5	✓	✓		268.7	153.9	193.7
6	✓	✓	✓	**262.7**	**151.7**	**188.9**

7.2. Forecast on fine-grained motion

Our model also shows competence on fine-grained motion, which we characterize as action with minimal global displacement. Indeed, global and local intentions contain complementary hints even for fine-grained motion. For example, let's consider two stationary people shaking hands. Their stationary and adjacent trajectories capture the presence of coarse interaction. Then, their detailed local motion of shaking hands or greeting corroborates the modeled interaction. When generating forecasts, initial immobile trajectories represent the encoded static interaction. Fine-grained motion such as hand-shaking is then conditionally predicted from such coarse intention of stationary trajectories. Table 2 shows prediction results for fine-grained motion, CMU-Mocap (UMPM) dataset samples thresholded by trajectory displacement of past motion. Our method is superior for all metrics; specifically, optimal performance in APE metric highlights our competence in capturing and forecasting the fine-grained local motion. Qualitatively, for

the top row red person of Fig. 1 in supplementary materials, ours' predicts a more dynamic punching motion for the stationary agent.

Table 2. Comparison of performance on fine-grained motion on CMU-Mocap (UMPM) dataset.

Global displacement threshold (m)	<0.05			<0.2		
Metric @ 2s	APE	JPE	FDE	APE	JPE	FDE
MRT [5]	108.7	140.8	71.8	138.1	180.6	99.0
JRT [6]	92.9	104.6	44.8	129.7	157.8	94.9
TBIFormer [3]	93.2	156.9	105.8	127.8	189.6	128.1
Ours	**87.6**	**93.1**	**22.6**	**122.9**	**146.3**	**65.1**

8. Failure cases

Our method, along with previous works, failed when local and global motion did not share consistent intentions. One example could be found in the CMU-Mocap (UMPM) dataset. As it was constructed with diversity as a priority, a few random motions such as flapping arms while walking could be found as in Fig. 4. Such a sequence lacks consistency between the local motion of flapping and the global motion of walking. As the intentions are misaligned, complementary contributions towards each decoding were marginal. Aside from these outliers, our method improved on sequences comprised of natural motion for all datasets.

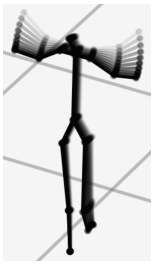

Figure 4. Unnatural motion GT data example.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[2] Mahsa Ehsanpour, Fatemeh Saleh, Silvio Savarese, Ian Reid, and Hamid Rezatofighi. Jrdb-act: A large-scale dataset for spatio-temporal action, social group and activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20983–20992, 2022. 3

[3] Xiaogang Peng, Siyuan Mao, and Zizhao Wu. Trajectory-aware body interaction transformer for multi-person pose forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17121–17130, 2023. 4, 6

[4] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13243–13252, 2022. 1, 3

[5] Jiashun Wang, Huazhe Xu, Medhini Narasimhan, and Xiaolong Wang. Multi-person 3d motion prediction with multi-range transformers. *Advances in Neural Information Processing Systems*, 34:6036–6049, 2021. 6

[6] Qingyao Xu, Weibo Mao, Jingze Gong, Chenxin Xu, Siheng Chen, Weidi Xie, Ya Zhang, and Yanfeng Wang. Joint-relation transformer for multi-person motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9816–9826, 2023. 6

[7] Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8823–8833, 2022. 4