Human Motion Prediction under Unexpected Perturbation

Supplementary Material

Code and pre-processed data are available: https://github.com/realcrane/Human-Motion-Prediction-under-Unexpected-Perturbation

1. Additional Experiments

1.1. Single-person Results

1.1.1 More Comparison

We give the full visual comparison between our methods and the 7 baseline methods in Fig. 1. Overall, our method achieves the best results and is the closest to the ground-truth. Comparatively, MDM and EqM predict visually unreasonable motions with strange poses. A2M, ACTOR, SiMLPe generate visually reasonable snapshots but low-quality motions as well as inaccurate prediction. RMD and PhyVae give more aesthetically pleasing results, but again not high-quality motions and accurate prediction.

A more detailed numerical comparison is presented in Fig. 2. Note that the visual comparison is to some extent consistent with the numerical results. MDM and EqM give the worst quality metrics on MBLE and FSE (MBLE for MDM is 0 since it uses a joint-angle-based representation hence no bone-length error). Across all metrics, our model is the best.

Looking closely, at motion tracking errors, MDM and EqM are not the worst. It suggests that motion tracking metrics and quality metrics evaluate two aspects of the results. This is indeed the case. RMDDiffuse and PhyVae give good motion quality among the baselines and their quality metrics are also good but not necessarily the best. Meanwhile, their tracking metrics are also good but not necessarily the best. A2M can achieve better FSE and sometimes better MBLE than RMDDiffuse and PhyVae, but its motion quality is generally lower. This suggests that there might be a trade-off between motion quality and prediction accuracy among the baselines. But our method achieves the best on both kinds of metrics.

1.1.2 More Generalization

We show more generalization experiments in the single-person scenario. We mainly test out-of-distribution push forces in magnitude, timing and duration. In magnitude, we fix the duration of the force to be the same as a strong push but use an extra stronger push that is 37.36% higher than the strongest push in the dataset. The result is shown in Fig. 3. We can see that the motion pushed by the extra strong push is significantly different from the ground truth and the predicted motion under the strong push. The motion contains earlier foot movements since the initial push is extra strong and it generates a much larger acceleration in the beginning. Also, the upper body is stiffer and has less swing because the balance recovery under an extra strong push tends to require the body to stiffen quickly to prevent the character from falling down and recover balance ultimately.

Furthermore, we generalize the push in timing and duration. This time, we apply multiple pushes at different times, as opposed to one push in the beginning as in the data. Note there are not multiple pushes in the data at all. We first apply a weak push, then a medium push at the 15th frame, and finally a strong push at the 50th frame. We show the visual results of this three-phase push in Fig. 4. One can see that the motion is initially slow and sluggish due to the weak initial push, then gradually intensifies as more pushes are introduced. Under the weak push, the character does not even start to make a step, then it starts to take steps after the medium push at the 15th frame. In the end, large strides need to be made, after the strong push at the 50th frame, to recover balance and counter-balance the accumulated accelerations.

In theory, our model can generalize to other scenarios like slippery surfaces as the friction is learned (Sec 3.1.2 in the main paper). Overall, our model can generalize to out-of-distribution physical disturbances in magnitude, timing and duration.

Method	MPJPE	hipADE	hipFDE
PPR	0.623	0.455	0.602
PHC	0.488	0.409	0.662
Ours	0.097	0.086	0.171

Table 1. Comparison with Full-body Physics-based Baselines.

1.1.3 Comparison with Full-body Physics-based Models

In literature, there is work which also employs body physics for motion imitation under full-body physics-based models [2, 9, 11, 20]. Although they turn fully/partially observed/user-specified motions into physically valid ones which is different from our task, they could be adapted to our new task. However, they are still intrinsically incapable of learning force interactions in multi-people. So, we could only compare the performance on single-person. To this

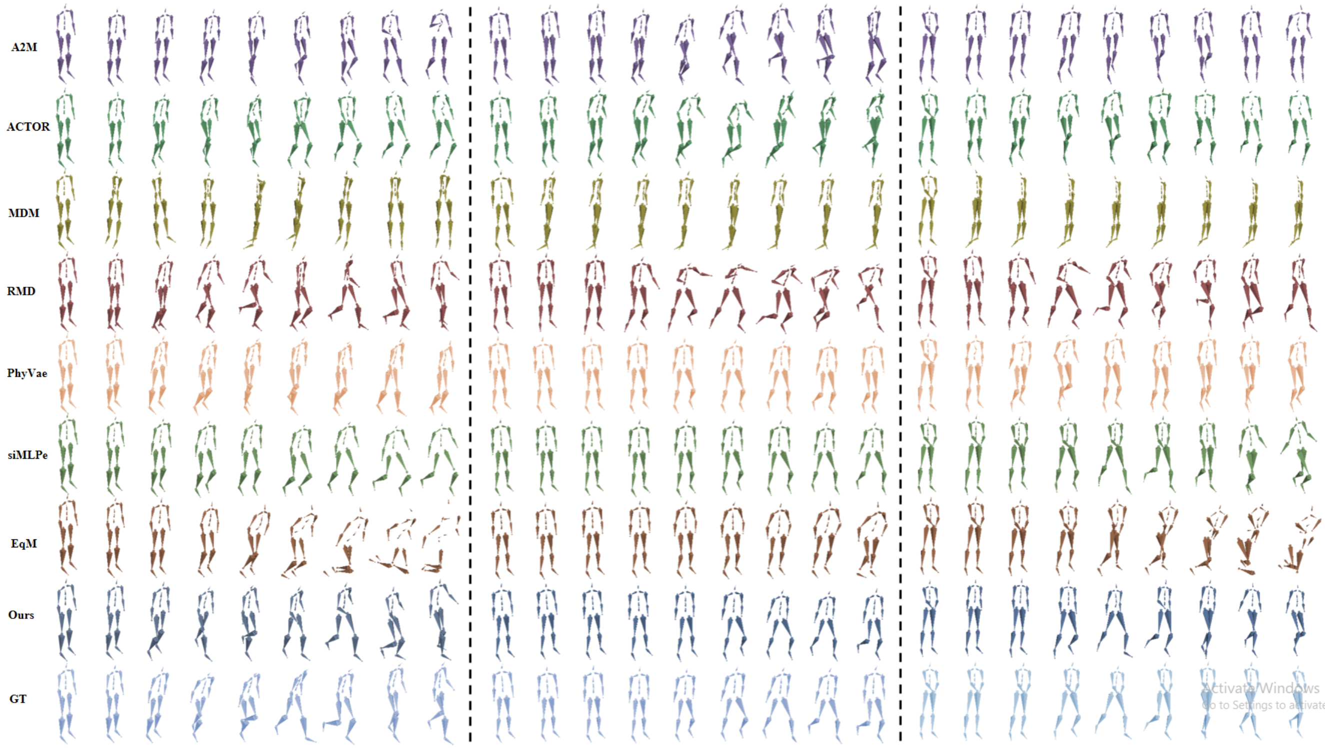

Figure 1. Visual comparison on pushes with different magnitudes. Left: strong, Middle: weak, Right: medium.

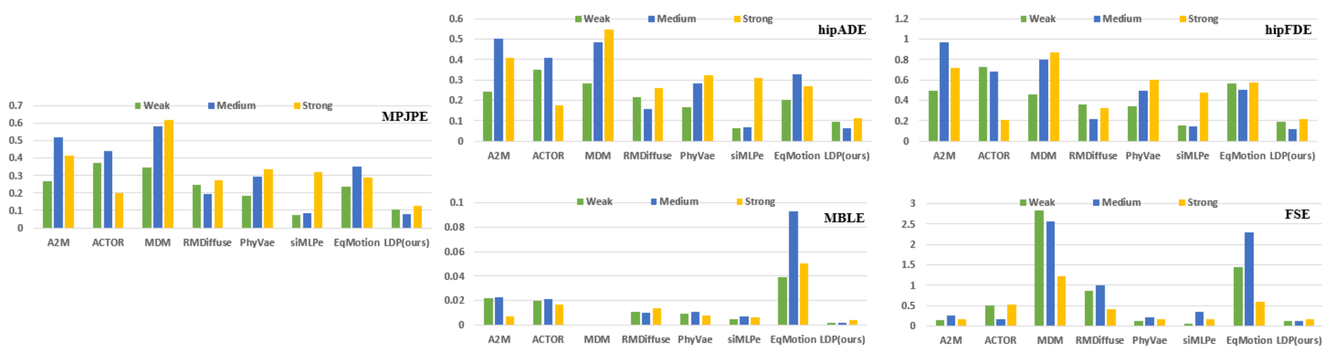

Figure 2. Perturbations with different magnitudes in single-person.

end, we adapted the latest physics-based models PPR [20] and PHC [9] and compared them with our model in the single-person scenario. Results are shown in Tab. 1. MBLE and FSE are not considered because these two baselines are joint-angle-based and simulation-based. Overall, Our model still performs best on all metrics. PPR and PHC can generate physically valid motions, but these motions are not necessary accurate prediction.

Compared with the full-body physics models, the Inverted Pendulum Model (IPM) is not fine-grained but has the right granularity for our problem. IPM is a compact yet flexible representation and therefore has been widely used for articulated bodies such as bipedal/quadrupedal robots including humanoids [5], especially in balance recovery. Further, simplification is crucial for scalable interaction learning. A full-body model contains 50-100 degrees of freedom (Dofs). Learning from a 4-people scene then involves 200-400 Dofs plus Dofs for interaction forces, which will be extremely unscalable/slow as the learning requires many iterations of forward simulation (for many time steps) and backward propagation. Also, the Dofs will quickly explode in simulation when the number of people increases, *e.g.* our 13-person example. In comparison, one IPM only has 4 Dofs and is much more scalable for both learning and simulation, whose representational capacity has been proven [6, 7]. Also, even with a small model, our model

Figure 3. The Generalization to an extra strong push. There are three motions (yellow, blue and green). Blue is the ground truth of a strong push. Yellow is our prediction on the same strong push. Green is an extra strong push.

Figure 4. The Generalization to a multi-push scenario. Yellow is the predicted motion under a strong push as in Fig. 3. Green is the extra strong push in Fig. 3. Red is the three-phase push motion. The numbers indicate the frames.

does not overfit, as evidenced by the superb testing results.

Another advantage of using IPMs instead of full-body physics models is the interaction modeling. We learn interaction forces as potential-energy based forces between two IPMs (Sec 3.1.3), which is flexible and easily learnable. This is because contact information (position, duration, etc.) is not in the data. Therefore the physics model cannot involve accurate contact modeling even with full-body models, especially when the contact can be frequently established and destroyed in push propagation.

1.2. Multi-people Results

1.2.1 More Comparison

We provide the complete visual comparison between our model with the 4 baseline methods in Fig. 5. Overall, our model obtains the best motion quality and is the closest to the ground truth. DuMMF cannot produce natural movements. JRFormer tends to predict merely subtle motions deviating from the ground truth. MRT and TBIFormer suffer from severe intersections between people for the group formation that is a line. MRT generates serious foot skating for the group formation where people stand in two lines, while TBIFormer performs as well as our model in this formation. Note that all baselines here are given much more information than our model. See the video for a more intuitive comparison.

Detailed numerical comparison can be found in Fig. 6. DuMMF employs the joint-angle-based representation, resulting in a zero MBLE. Overall, our model achieves or is close to the best performance across metrics and perturbation levels. For all tracking error metrics, our model is much better than baseline methods. This is because only our model can predict the onset and duration of interaction accurately. In motion quality metrics, our model outperforms all baselines across three perturbation levels, meaning that our motion has the best quality.

1.2.2 More Generalisation

Other than the 13-people in a diamond formation shown in the main paper, we conduct further generalization experiments. We employ a formation with ten people standing closely in a line, to test whether a strong push can be propagated. Since we explicitly set the distances between people to be very small, we expect a strong push on the first person to be propagated through people all the way to the front, like what is commonly observed in high-density crowds.

Our prediction results are shown in Fig. 7. Note this type of scenario is totally out of distribution, in terms of the number of people and the formation (a much longer line). One can see a clear push propagation starting at the back of the line and then being carried over all the way to the front. This shows not only are the individual motions captured by the model, but the interaction as well as the interaction propagation are also predicted well.

Furthermore, looking closely at the predicted interaction force between people, we find that the core reason for this push to be propagated, instead of dying down, is that it is intensified by interactions. This is also observed in high-density crowds where a small push can be intensified to cause the "butterfly effects" and finally even cause crushes. By turning the parameters in basic forces in the interaction module, it is possible to let the propagation dissipate more quickly. Overall, this shows the great flexibility of our model in capturing complex interactions and interaction propagation. This flexibility could be crucial in crowd simulation in high-density crowds where potential crushes can happen.

1.3. Data Efficiency

One core reason for our LDP design is the lack of data. So it is essential to test the data efficiency. Although the original data is already much smaller than existing datasets for human motion prediction, we further reduce the data to 25%

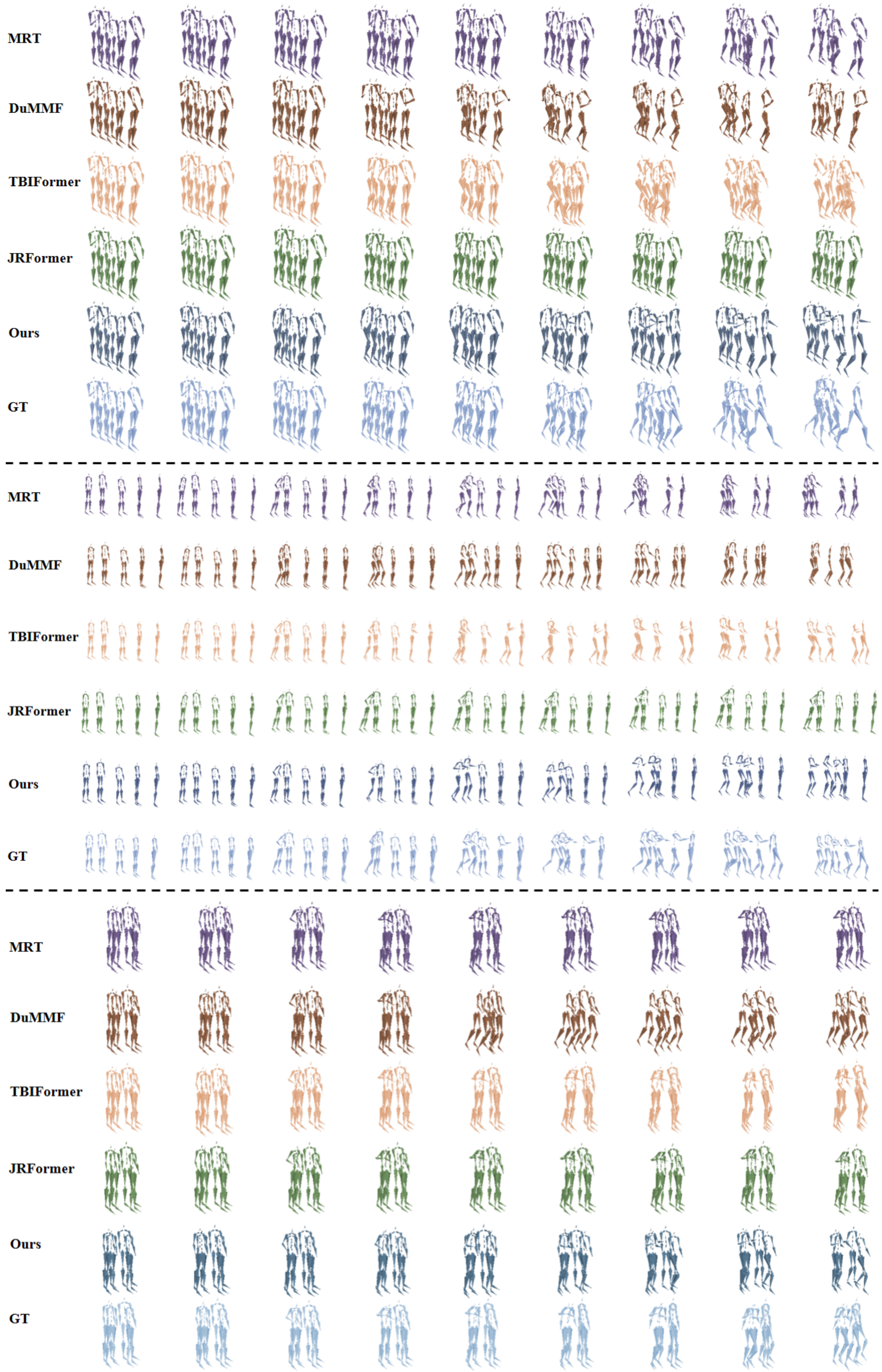

Figure 5. Visual comparison on pushes with different magnitudes and group formations. Top: medium, Middle: strong, Bottom: weak.

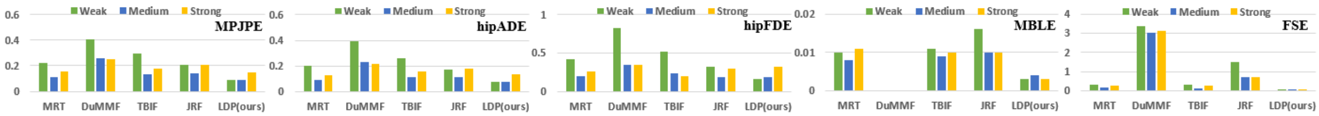

Figure 6. Perturbations with different magnitudes in multi-people.

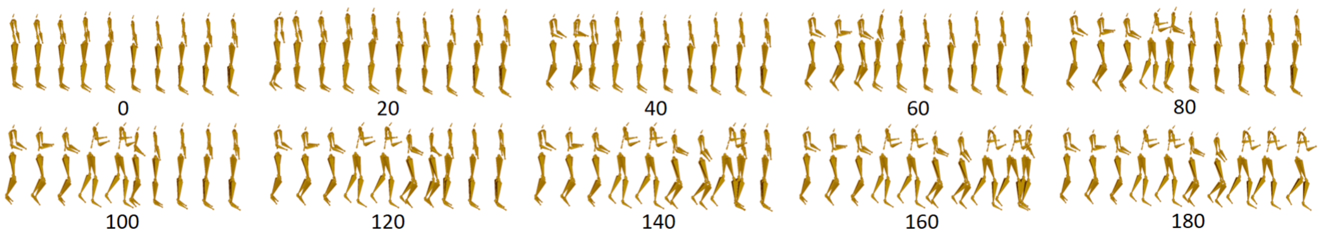

Figure 7. Generalization on Ten people in a line. The first person is pushed by a strong force and we can simulate the force propagation. The number denotes which frame.

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
A2M	0.403	0.386	0.730	0.019	0.200
ACTOR	0.362	0.338	0.591	0.020	0.434
MDM	0.500	0.424	0.686	0	2.567
RMDiffuse	0.228	0.202	0.299	0.011	0.790
PhyVae	0.260	0.249	0.460	0.009	0.170
siMLPe	0.130	0.117	0.226	0.006	0.182
EqMotion	0.296	0.270	0.543	0.064	1.552
Ours	0.097	0.086	0.171	0.002	0.131
siMLPe_25%	0.203	0.189	0.411	0.009	0.650
Ours_25%	0.207	0.190	0.267	0.009	0.211

Table 2. Metrics in complete (top) and 25% (bottom) training data for single-person.

of its original size and repeat the training on single-person and multi-people scenarios.

As shown in Tab. 2, our model trained on 25% training data still outperforms all baselines trained 100% data, except for siMLPe in the single-person scenario. Therefore, we also trained siMLPe on 25% training data and evaluated it on all metrics for comparison. siMLPe achieves good performance and is slightly better than our model on MPJPE and hipADE on 25% training data, while our model performs much better on hipFDE and FSE. It's notable that we gave much more information to siMLPe.

One possible reason for the good performance of siMLPe might be its lightweight, as aimed for by its authors. So we also compare the model sizes in Tab. 3. It is clear that the lightweight is not the only reason, as other baselines which are smaller than ours cannot achieve good results. We speculate that expressivity especially explicit physics is the key. Further, even siMLPe can achieve good numerical results, its predicted motions are of lower visual quality. More importantly, extending siMLPe to multi-people scenarios is challenging as it cannot learn interactions at all.

Next, we suspect that reduced training data brings more difficulty to the multi-people motion prediction. The results prove us correct, shown in Tab. 4. Our model is still better than all baselines trained on 100% data, even though the training data for our model is reduced to 25%.

The high data efficiency of our model is mainly because the physics model embedded in our model has a low number of learnable parameters, but largely dictates the motion trend. The governing differential equation (Eq. 3 in the main paper) restricts the overall input-output mapping of the whole model and therefore it requires little data to learn. Similar phenomena have been observed in other differentiable physics research [15, 21].

2. Additional Experiment Details

2.1. Dataset Details

The new dataset, FZJ Push, records human motions under expected physical perturbations. There are 45 single-person motions and 63 multi-people motions in the dataset. In both scenarios, repeated experiments were conducted on applying unexpected physical pushes with varying magnitude onto a person. In the single-person scenario, this is simply recording reactive motions to push and balance recovery; in multi-people scenario, one person is pushed and this person pushes other people to recover balance so that the push can be propagated among several people.

After discarding redundant frames such as those in waiting, we have 3104 frames and 5614 frames in the single-person and multi-people scenarios, respectively. All pushes are recorded via a pressure sensor Xsensor LX210:50.50.05

Method	A2M	ACTOR	MDM	RMD	PhyVae	siMLPe	EqMotion	Ours_S	MRT	DuMMF	TBIF	JRF	Ours_M
Parameters	0.45	14.78	18.10	40.96	2.72	0.02	0.64	2.67	6.98	6.54	10.26	3.70	2.94

Table 3. Model Size in Single-person (left) and multi-people (right). The unit is M (million). Our_S means our model for single-person which excludes the differential interaction model. Our_M is our complete model.

Method	MPJPE	hipADE	hipFDE	MBLE	FSE
MRT	0.162	0.140	0.282	0.010	0.256
DuMMF	0.312	0.285	0.480	0	3.194
TBIFormer	0.204	0.177	0.305	0.010	0.234
JRFormer	0.181	0.152	0.260	0.012	0.932
Ours	0.106	0.092	0.218	0.003	0.069
Ours_25%	0.139	0.115	0.270	0.011	0.117

Table 4. Metrics in complete (top) and 25% (bottom) training data for multi-people.

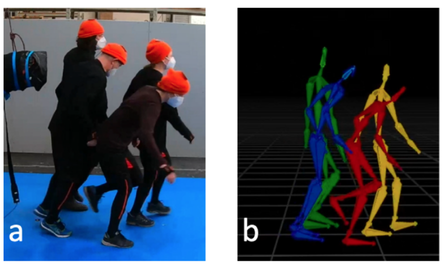

Figure 8. FZJ Push [1]. The blue agent was pushed by the punch bag and then he pushed other people.

on the punching bag. The punching bag was moved manually by the same operator in all experiments. In addition, the pushes are labelled as small, medium and strong.

In the single-person scenario, the dataset involves 4 subjects (S1-S4). Tab. 5 left shows the number of experiments on each subject under different push magnitudes. We select randomly about 30% of the data to construct the test set for every subject, while the remaining data is used for training. Finally, the test set and train set have 13 motions and 32 motions, respectively. In the multi-people scenario, the dataset involves 4 group settings. G1 has four people standing in two lines, shown in Fig. 8. G2 has a formation where four people are in a line. G3 and G4 contain 5 people in a line. We give the number of motions in every group setting under different push magnitudes in Tab. 5 right. We randomly select approximately 20% of the data in each group setting for the testing set, while the remaining is for training. Eventually, the test set and train set have 14 motions and 49 motions, respectively.

Single	Wk	Med	Str	Tot	Group	Wk	Med	Str	Tot
S1	3	4	3	10	G1	4	4	4	12
S2	5	3	4	12	G2	6	6	4	16
S3	5	4	4	13	G3	10	9	6	25
S4	3	4	3	10	G4	0	10	0	10
Tot	16	15	14	45	Tot	20	29	14	63

Table 5. Dataset Details. There are four subjects and four group settings in single-person and multi-people respectively in the dataset. Three push magnitudes (weak, medium and strong) are used.

2.2. Metrics

We adopt five metrics commonly used for evaluating motion prediction accuracy and quality as follows. MPJPE (Mean Per Joint Position Error), hipADE (Average Displacement Error at the hip) and hipFDE (Final Displacement Error at the hip) are metrics measuring the tracking errors. MPJPE is the most widely used metric in human motion prediction to evaluate prediction accuracy on every joint. hipADE focuses on the main motion trend, while hipFDE pays attention to the final position of the hip. Moreover, hipADE and hipFDE are strongly relevant to the hip joint which corresponds to the point mass in our IPM. In addition, another two metrics MBLE (Mean Bone Length Error) and FSE (Foot Skating Error) are used to measure the motion quality. We adopt these two metrics to check if our model can produce reasonable poses and motions.

- **MPJPE**: Mean Per Joint Position Error (MPJPE) is the average l_2 distance between predicted positions of joints and their ground truth:

$$\text{MPJPE} = \frac{1}{TNJ} \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{j=1}^{J} \| X_t^n[j] - \hat{X}_t^n[j] \|_2, \quad (1)$$

  where $X_t^n[j]$ is the position of the jth joint of the nth person at frame t and $\hat{X}_t^n[j]$ is its prediction. This metric is used most widely to measure the 3D pose errors.
- **hipADE**: Average Displacement Error at the hip (hipADE) is the average l_2 distance between predicted positions of hip joints and their ground truth:

$$\text{hipADE} = \frac{1}{TN} \sum_{t=1}^{T} \sum_{n=1}^{N} \| h_t^n - \hat{h}_t^n \|_2, \quad (2)$$

where h_t^n is the hip position of the nth person at frame t and $\hat{h}_t^n$ is its prediction. This metric focuses on global errors.

- **hipFDE**: Final Displacement Error at the hip (hipFDE) is the average l_2 distance between predicted positions of the hip joints at the last frame in each motion sequence and their ground truth:

$$\text{hipFDE} = \frac{1}{N}\sum_{n=1}^{N}\|h_T^n - \hat{h}_T^n\|_2. \quad (3)$$

- **MBLE**: Mean Bone Length Error (MBLE) is the average l_1 distance between lengths of predicted bones and their ground truth:

$$\text{MBLE} = \frac{1}{TNB}\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{b=1}^{B}\left|X_t^{nb} - \hat{X}_t^{nb}\right|, \quad (4)$$

where X_t^{nb} is the length of bth bone of the nth person at frame t and $\hat{X}_t^{nb}$ is the corresponding prediction.

- **FSE**: Foot Skating Error (FSE) is the average of weighted foot velocities for all feet with a height h within a threshold H. The weighted velocity is $v_f(2 - 2^{h/H})$.

2.3. Baseline Adaptation

The task proposed in the main paper is new, so there is no similar work to our best knowledge. For comparison, we adapted 11 state-of-the-art baseline methods in the most relevant areas: motion forecasting, motion generation and motion synthesis. One selection criterion is the availability of the code, to ensure their original implementation is used.

Specifically, we choose A2M [3], ACTOR [12], MDM [13], RMDiffuse [23], PhyVae [16], EqMotion [17], siMLPe [4], PPR [20] and PHC [9] for the single-person scenario, and MRT [14], DuMMF [19], TBIFormer [10] and JRFormer [18] for the multi-people scenario. We try our best to keep the best performance of these baselines when adapting. The adaptation details are as follows:

- **A2M.** Action2Motion (A2M) is the first work to generate human motions given an action type. We use the push magnitudes (weak, medium and strong) as the action labels (0, 1, 2). The initial pose is applied to kick-start the generation instead of a blank pose filled with 0 in the testing phase.
- **ACTOR.** Action-conditioned Transformer VAE (ACTOR) is another action-to-motion method following A2M. Similar to A2M, the push magnitudes are regarded as the action labels (0, 1, 2). In addition, the initial pose is given when decoding.
- **MDM.** Motion Diffusion Model (MDM) is one of the first papers employing diffusion models in motion generation. This model can achieve great performance for text-to-motion and action-to-motion. We replace the text input in MDM with the input forces under the text-to-motion setting. Then, the part corresponding to the initial frame in $\hat{x}_0$ is overwritten at each iteration as the MDM does in its motion editing. This is to minimize the change for adaptation. MDM handles motion editing, where if we fix the first frame, the task setting is almost the same as our task. Specifically, motion editing with the initial frame fixed is equivalent to letting the model generate the whole motion given the input signal.
- **RMDiffuse.** Retrieval-augmented Motion Diffusion model (RMDiffuse) is the state-of-the-art model in motion generation. We adopt its test-to-motion setting and replace the original text input with the input force. Similar to MDM, the part corresponding to the initial frame in $\hat{x}_0$ is overwritten at each iteration during evaluation to ensure the information of the first frame is given.
- **PhyVae**. Physics-based VAE (PhyVae) is the state-of-the-art motion synthesis model. At each step, PhyVae predicts current action a_t given the current input signal g_t and current state s_t. Then a_t is fed into a pre-trained network (that can be regarded as a decoder) to predict the next state s_{t+1}. The input force at each time step t is regarded as the input signal to synthesize the motion.
- **siMLPe.** This model is a lightweight network based on MLP but can achieve state-of-the-art performance in single-person motion prediction. For this forecasting approach, it requires as input M frames and predicts N frames. To ensure the comparison is as fair as possible, we provide as input complete information on the input force including magnitude and duration. Specifically, we set M to the maximum duration of the input forces in the single-person scenario. Then, we keep the original ratio between the past and the future frames in the long-term setting in the paper to set N as M/2. M and N values are shown in Tab. 6. During testing, given the first M frames, we predict autoregressively to get the complete motion.
- **PPR and PHC.** These two baselines are state-of-the-art physics-based character animation methods which deal with perturbations. PPR and PHC can synthesize physically valid motions given reference motions. However, reference motions are unavailable during prediction in our new task. Therefore, following the setup in PHC, we use the adapted MDM to generate the reference motions during the test phase. Then these two baselines can generate motions based on the reference motions generated from the adapted MDM.
- **EqMotion, MRT, DuMMF, TBIFormer, JRFormer.** These models fall into human motion forecasting. They have a similar adaptation to that in siMLPe, as they have similar input/output requirements. Details of their settings of input/output frames are shown in Tab. 6. EqMotion is the state-of-the-art motion forecasting model for single-person. MRT is a classical multi-people motion

Method	Original		Adaptation	
	Past	Future	Past	Future
siMLPe	50	25	12	6
EqMotion	25	25	12	12
MRT	15	45	20	60
DuMMF	10	25	20	50
TBIFormer	15	45	20	60
JRFormer	15	45	20	60

Table 6. Adaptation for Motion Prediction Methods. 12 and 20 are the maximum duration of the input forces in the single-person and multi-people scenarios, respectively.

prediction method. DuMMF, TBIFormer and JRFormer are state-of-the-art multi-people motion prediction models.

2.4. Additional Details of Ablation Study

Here, we provide more details of the ablation study in the main paper. We conducted the ablation study to evaluate the effectiveness of two important components in our model: the Differentiable IPM and the Skeleton Restoration Model. We have four combinations: with/without IPM, and Full (full-body restoration) / Low-up (first lower body then upper body).

Our complete model is with IPM and uses a Low-up setting. Without IPM, it means that we only use the Skeleton Restoration Model to predict the next frame, while the two samplers (Upper-sampler and Lower-sampler) have to be dropped as they require the IPM state as input. Therefore, to sample the latent space of the CVAE, we sample the latent variable from a standard Normal distribution during the evaluation phase. The Full/Low-up setting is only within the Skeleton Restoration Model. In Full, we use a Conditional Variational Autoencoder (CVAE) to generate full-body poses directly. Using the current frame as a condition, we sample the latent space three times and average it. Then both are fed into the decoder to generate the next frame. In Low-up, we have two CVAEs and we generate the next frame in exactly the same way as in the Full setting, except that we first generate the lower body then the upper body.

3. Additional Details of Methodology

3.1. Differentiable Inverted Pendulum Model

Given I_0 and $\dot{I}_0$, we can simulate the IPM motion in time by solving Eq. (5) repeatedly:

$$M(I_t, l_t)\ddot{I}_t + C(I_t, \dot{I}_t, l_t) + G(I_t, l_t) = F_t^{net} \tag{5}$$

where $M \in \mathbb{R}^{4\times4}$, $C \in \mathbb{R}^{4\times1}$ and $G \in \mathbb{R}^{4\times1}$ are the inertia matrix, the Centrifugal/Coriolis matrix, and the external force such as gravity:

$$M_t = \begin{bmatrix} m_c + m_p & 0 & m_p l_t c_{\theta_t} & 0 \\ 0 & m_c + m_p & m_p l_t s_{\theta_t} s_{\phi_t} & -m_p l_t c_{\theta_t} c_{\phi_t} \\ m_p l_t c_{\theta_t} & m_p l_t s_{\theta_t} s_{\phi_t} & m_p l_t^2 & 0 \\ 0 & -m_p l_t c_{\theta_t} c_{\phi_t} & 0 & m_p l_t^2 c_{\theta_t}^2 \end{bmatrix}$$

$$C_t = \begin{bmatrix} -m_p l_t s_{\theta_t} \dot{\theta}_t^2 \\ m_p l_t (2 s_{\theta_t} c_{\phi_t} \dot{\theta}_t \dot{\phi}_t + c_{\theta_t} s_{\phi_t} (\dot{\theta}_t^2 + \dot{\phi}_t^2)) \\ m_p l_t^2 s_{\theta_t} c_{\phi_t} \dot{\phi}_t^2 \\ -2 m_p l_t^2 s_{\theta_t} c_{\theta_t} \dot{\theta}_t \dot{\phi}_t \end{bmatrix} \quad G_t = \begin{bmatrix} 0 \\ 0 \\ -m_p g l_t s_{\theta_t} c_{\phi_t} \\ -m_p g l_t c_{\theta_t} s_{\phi_t} \end{bmatrix}.$$

Here, m_c and m_p are the mass of the cart and the pendulum, respectively. c_{θ_t} and s_{θ_t} denote $\cos\theta_t$ and $\sin\theta_t$, while c_{ϕ_t} and s_{ϕ_t} represent $\cos\phi_t$ and $\sin\phi_t$. We set m_c and m_p as $0.1M$ and $0.9M$ respectively where M is the total mass of a person. Unlike the standard IPM, we allow the rod length to change with time. Given the net force $F_t^{net} \in \mathbb{R}^4$ and the rod length l_t, we can solve Eq. (5) for the next state I_{t+1} via a semi-implicit scheme:

$$\dot{I}_{t+1} = \dot{I}_t + \triangle t \ddot{I}_t, \quad I_{t+1} = I_t + \triangle t \dot{I}_{t+1},$$

where $\triangle t$ is the time step. We have elaborated on the prediction of F_t^{net} and l_t in the main paper. Then we have the following equation:

$$I_T - I_0 = \int_0^T \dot{I}_t \mathrm{d}t = \int_0^T \int M_t^{-1}(F_t^{net} - C_t - G_t)\mathrm{d}t\mathrm{d}t,$$

given the initial condition I_0 and $\dot{I}_0$ and the final station I_T. The prediction of F_t^{net} is based on the neural networks and other differentiable operations such as PD control and repulsive potential energy. The prediction of the rod length l_t is from a neural network. Finally, the semi-implicit scheme for updating I_t only includes simple differentiable arithmetic. Therefore, our complete IPM is differentiable for both single-person and multi-people scenarios.

Single-person Prediction. The hyper-parameters K_p and K_d in the PD control are [30, 30, 1500, 1500] and [4, 4, 200, 200], respectively. We use an LSTM with the size 256 to predict $F_t^{self-nn}$. The MLP predicting the rod length has hidden size [128, 128].

Differential Interaction Model. if $|r_{t,nj}| < r_{neigh}$, the jth person is the neighbor of the nth person at time t *i.e.* $j \in \Omega_{t,n}$, where $r_{neigh} = 0.5$. We use an MLP with 2 hidden layers [512, 512] to predict $F_{t,nj}^{inta-nn}$. The hyper-parameters u and σ in the repulsive potential energy function for calculating F_{nj}^{bs-xy} are 150 and 0.5, respectively. Then, we elaborate the $F_{nj}^{bs-\theta\phi} = [F_{nj}^{bs-\theta}, F_{nj}^{bs-\phi}]^{\mathbf{T}}$. We give details for $F_{nj}^{bs-\theta}$, where the same principle also applies to $F_{nj}^{bs-\phi}$. The magnitude of $F_{nj}^{bs-\theta}$ is a constant $k_\theta = 100$ ($k_\phi = 50$), while its direction is based on the θ_n and θ_j of IPM states of n and j. We categorize θ into three groups: positive, zero and negative. For two IPMs, this produces a total of 9 possible situations. Then we need to decide their relative position. Taking the nth person as the person in interest, if its relative position with respect to a neighbor j along the x-axis is positive *i.e.* $x_{nj} = x_n - x_j > 0$,

θ_j / θ_n	Pos	Zero	Neg	ϕ_j / ϕ_n	Pos	Zero	Neg
Pos	1/-1	0/-1	0/-1	Pos	-1/1	-1/0	-1/0
Zero	1/0	0/0	0/-1	Zero	0/1	0/0	-1/0
Neg	1/0	1/0	1/-1	Neg	0/1	0/1	-1/1

Table 7. Basic Interaction Force on Angles. X/X is BE/BA.

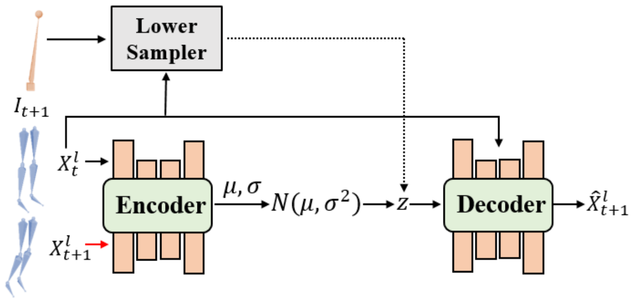

Figure 9. The Architecture of the CVAE-Lower. During training, the current lower-body pose X_t^l as the condition, and the next lower-body pose X_{t+1}^l are fed into the encoder to predict the distribution of the latent variable z. Then the decoder predicts the next pose $\hat{X}_{t+1}^l$ from the sampled variable z and the condition. The red connection is only used in training. During inference, we use the lower sampler to sample the latent variable z to predict motion.

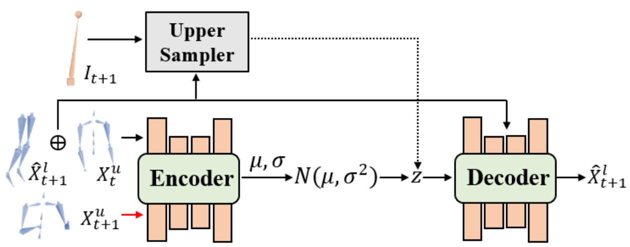

Figure 10. The Architecture of the CVAE-Upper. The condition X_t^u and $\hat{X}_{t+1}^l$, together with the next pose X_{t+1}^u are fed into the encoder to predict the distribution of the latent variable z. Then the decoder predicts the next pose $\hat{X}_{t+1}^u$ from the sampled variable z and the condition. The red connection is only used in training. During inference, we use the upper sampler to sample the latent variable z to predict motion.

we label it as BE, otherwise BA. We show the directions of the force for all 9 possible situations in Tab. 7, where 1 denotes the interaction force is positive, 0 denotes no interaction forces, and -1 denotes the negative direction.

3.2. Skeleton Restoration Model

Lower Body Restoration. We follow [8] to construct the CVAE-Lower as shown in Fig. 9. The encoder is an MLP with two hidden layers of 256 dimensions, with an ELU layer following each hidden layer. The dimension of the latent variable z is 64. A mixture-of-expert architecture is employed for the decoder, including 4 expert networks and a gating network. The input to the gate network and the expert networks are the latent variable z combined with the current lower-body pose X_t^l, while the output of the expert network is the next pose X_{t+1}^l.

Similar to the encoder, the gate network is an MLP with two 64D hidden layers followed by ELU activations. Each expert network has the same structure as the encoder except for the input layer and the output layer.

During testing, we use the Lower Sampler to sample the latent variable z given the current lower-body pose X_t^l and the predicted IPM state I_{t+1}. The Lower sampler has the same structure as that of the encoder except for the input layer.

Upper Body Restoration. The CVAE-Upper has the same architecture as the CVAE-Lower except for the condition X_t^u and $\hat{X}_{t+1}^l$ as shown in Fig. 10. Similarly, the upper sampler has the same structure as the encoder of CVAE-Upper except for the input layer. Although the upper body is not explicitly physically constrained, it is implicitly constrained by the IPM motion which is physically based.

State Representation. In the skeleton restoration model, we adopt pose representations [8, 22] for the full-body pose. Specifically, we use a vector including positions, rotations, and velocities to represent the pose X_t. X_t^l and X_t^u take the corresponding lower or upper part in the X_t. Furthermore, we use a 15D vector $[x_t, y_t, \theta_t, \phi_t, e_t, l_t, \dot{x}_t, \dot{y}_t, \dot{\theta}_t, \dot{\phi}_t, \dot{e}_t]$ for I_t and input the vector into the sampler, where e_t is the position of the end of the rod corresponding to the hip joint and l_t is the rod length.

3.3. Training

There are several components in our model. An end-to-end training but could lead to suboptimal local minima. Therefore, we employ pre-training to initialize individual components and also use auxiliary losses in addition to the main loss introduced in the main paper.

We train the IPM first. Then, we train the CVAE-Lower. Next, we train the lower sampler network based on the trained CVAE-Lower. Similarly, we train the CVAE-Upper first then the upper sampler network.

We train the differentiable IPM model by using the 0-order and 1-order information as shown in Eq. (6), where λ is a weight parameter. We minimize the angular velocity $\dot{\phi}$ instead of penalizing its l_1 norm as we do in other dimensions. This is because the angular velocity should always be smooth when recovering balances so that smoothing leads to better results than minimizing the l_1 norm.

$$\begin{aligned} L_{ipm} =& \frac{1}{T}\sum_{t=1}^{T}\{|\hat{x}_t - x_t| + |\hat{y}_t - y_t| + |\hat{\theta}_t - \theta_t| + |\hat{\phi}_t - \phi_t| \\ &+ |\hat{\dot{x}}_t - \dot{x}_t| + |\hat{\dot{y}}_t - \dot{y}_t| + |\hat{\dot{\theta}}_t - \dot{\theta}_t| + \lambda|\hat{\dot{\phi}}_t|\} \end{aligned} \tag{6}$$

We follow [8] to train the CVAE-Lower in our skeleton restoration model. Then we train the Lower sampler network based on the trained CVAE-Lower. The encoder of the CVAE-Lower and the lower Sampler both output the Gaussian distribution parameters $[\mu, \sigma]$ for the latent variables z. We train the lower sampler by using the loss function in Eq. (7) to let the outputs of the Lower Sampler be close to those of the encoder, and we ensure that the restored poses have low FSE.

$$L_{skel} = \|\hat{z}_\mu - z_\mu\|^2 + \|\hat{z}_\sigma - z_\sigma\|^2 + \text{FSE}(\hat{X}_t^l, X_t^l) \quad (7)$$

We train the CVAE-Upper and the Upper sampler network in the same way except that the FSE in the loss function Eq. (7) is ignored.

After initialization, the whole network can be trained as a whole. We use the Adam optimizer for all training. The learning rates for training the differentiable IPM and two samplers are 3e-4 and 1e-4, respectively. When training the CVAEs, a linear schedule is used to adjust the learning rate from 1e-4 to 1e-7, and we set the weight of the KL loss as 0.005 to encourage high reconstruction quality. The whole training takes about 15 hours on a single GeForce RTX 2080 Ti, but can be automated. The inference takes approximately 0.19 sec/frame in our 13-person experiment.

References

[1] Sina Feldmann and Juliane Adrian. Forward propagation of a push through a row of people. *Safety science*, 164:106173, 2023. 6

[2] Erik Gärtner, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu. Differentiable dynamics for articulated 3d human motion reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13190–13200, 2022. 1

[3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 7

[4] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4809–4819, 2023. 7

[5] Jaepyung Hwang, Jongmin Kim, Il Hong Suh, and Taesoo Kwon. Real-time locomotion controller using an inverted-pendulum-based abstract model. In *Computer Graphics Forum*, pages 287–296. Wiley Online Library, 2018. 2

[6] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kazuhito Yokoi, and Hirohisa Hirukawa. The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, pages 239–246. IEEE, 2001. 2

[7] Taesoo Kwon and Jessica K Hodgins. Momentum-mapped inverted pendulum models for controlling dynamic human motions. *ACM Transactions on Graphics (TOG)*, 36(1):1–14, 2017. 2

[8] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)*, 39(4):40–1, 2020. 9, 10

[9] Zhengyi Luo, Jinkun Cao, Kris Kitani, Weipeng Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10895–10904, 2023. 1, 2, 7

[10] 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. 7

[11] Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018. 1

[12] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 7

[13] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 7

[14] 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. 7

[15] Kun Wang, Mridul Aanjaneya, and Kostas Bekris. Sim2sim evaluation of a novel data-efficient differentiable physics engine for tensegrity robots. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1694–1701. IEEE, 2021. 5

[16] Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 7

[17] Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1410–1420, 2023. 7

[18] 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. 7

[19] Sirui Xu, Yu-Xiong Wang, and Liangyan Gui. Stochastic multi-person 3d motion forecasting. In *The Eleventh International Conference on Learning Representations*, 2022. 7

[20] Gengshan Yang, Shuo Yang, John Z Zhang, Zachary Manchester, and Deva Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3914–3924, 2023. 1, 2, 7

[21] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory forecasting with explainable behavioral uncertainty. *arXiv preprint arXiv:2307.01817*, 2023. 5

[22] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):1–11, 2018. 9

[23] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. *arXiv preprint arXiv:2304.01116*, 2023. 7