

## A Appendix Overview

In this appendix, we present further analyses, implementation details, and additional experimental results. Specifically: (1) We provide a detailed video demonstration in the supplementary materials, with a corresponding explanation provided in Sec. [B](#). (2) We delve deeper into our key insight – global trajectory serves as the foundation for both local motion and interaction realism – and further validate this claim with quantitative experimental evidence in Sec. [C](#). (3) Additional information regarding the implementation of MoReact and the baseline models is detailed in Sec. [D](#). (4) Sec. [E](#) illustrates extra ablation studies to show the efficacy of MoReact’s framework. (5) We discuss MoReact’s limitations and social impacts in Sec. [F](#).

## B Visualization Demo

In the supplementary materials, we include a demo video that shows the visualization results associated with figures in the main paper. The video features: (1) visualizations of our main insights; (2) qualitative comparisons with baseline models; (3) showcases of MoReact’s controllability on both text and motion; and (4) qualitative results of MoReact in action-driven reaction generation task on CHI3D dataset. Please watch the video for further results and details.

## C A Further Investigation on Our Key Insight: The Central Role of Global Trajectory

As discussed and qualitatively validated in the main paper as well as in the demo video, a crucial insight of our work is that *the global trajectory serves as the foundation for both local motion and interaction realism. Incorrect global trajectory makes it difficult for the local motion to align with the action and text description, having a more detrimental impact on interaction realism than incorrect local motion.* Here to further validate this insight in a *quantitative* manner, we perform an experiment examining the impact of equivalent levels of noise on both global trajectory and local motion and their effects on the overall motion’s realism.

In detail, for a reaction  $\mathbf{x}_0$ , we use a diffusion style forward process to apply a sequence of Gaussian noise additions to  $\mathbf{x}_0$  and obtain the noised full-body reactions  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T$ , where  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Based on the noised full-body reactions  $\{\mathbf{x}_t\}_{t=1}^T$ , we can obtain reactions with noised global trajectory  $\{\mathbf{x}_t^g\}_{t=1}^T$  and reactions with noised local motion  $\{\mathbf{x}_t^l\}_{t=1}^T$  with the following equations:

$$\mathbf{x}_t^g = (1 - M^g) \odot \mathbf{x}_0 + M^g \odot \mathbf{x}_t \quad (9)$$

$$\mathbf{x}_t^l = (1 - M^l) \odot \mathbf{x}_0 + M^l \odot \mathbf{x}_t, \quad (10)$$

where  $M^g, M^l$  represent the masks for the dimensions that describe the global trajectory information and local motion information in the motion feature  $\mathbf{x}$  respectively, and  $\odot$  is the Hadamard product. For a set of time steps  $\{t\}$ , we compute  $\mathbf{x}_t, \mathbf{x}_t^g$  and  $\mathbf{x}_t^l$  for every reaction  $\mathbf{x}$  in the test dataset. Subsequently, we evaluate the realism of interaction between the actor’s motion and the noised reaction by calculating the Fréchet Inception Distance (**FID**) across the entire test dataset. The **FID** is computed by the MotionClip [\(Tevet et al., 2022a\)](#) provided by InterGen [\(Liang et al., 2024\)](#). The results, depicted in Fig. [C.1](#) and consistent with the demo video, show that adding noise to the global trajectory has a more detrimental effect on the realism of interactions compared with adding noise to the local motion, thus motivating the design of our MoReact framework.

## D Implementation Details

**Formulation of Velocity Interaction Loss  $L_1^v$ .** Beyond the position interaction loss introduced in the main paper, we also employ a velocity interaction loss to enhance the model’s ability to generate realistic close interactions. Similar to the computation of  $L_1^p$ , for  $L_1^v$ , we first compute  $\tilde{\mathbf{V}}_x$  and  $\mathbf{V}_y \in \mathbb{R}^{J \times (T-1) \times 3}$ , representing the joint velocities of the reactor and the actor. We then calculate the velocity interaction graph  $\tilde{\mathbf{M}}_v \in \mathbb{R}^{J \times J \times (T-1) \times 3}$ , where  $\tilde{\mathbf{M}}_v[i, j] = \mathbf{V}_y[j] - \tilde{\mathbf{V}}_x[i]$ . We also calculate  $\mathbf{M}_v$  for the ground truth reactor

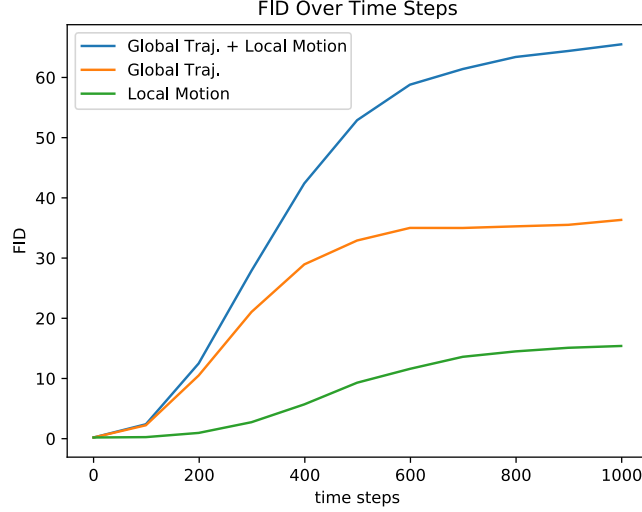


Figure C.1: **Change of FID for different noising modes and diffusion steps.** Adding noise to the global trajectory has a more detrimental effect on the realism of interactions compared with adding noise to the local motion.

and the actor. The velocity interaction loss  $L_I^v$  can be formulated as:

$$L_I^v = \frac{1}{|S'|} \sum_{(i,j,k) \in S'} \mathbf{W}_v[i, j, k] \|\tilde{\mathbf{M}}_v[i, j, k] - \mathbf{M}_v[i, j, k]\|_2^2, \quad (11)$$

where  $S' = \{(i, j, k) | \mathbf{D}_p[i, j, k] \leq c, k < T\}$  is a set of index pairs and  $\mathbf{W}_v = \sigma(\tilde{\mathbf{D}}_p) + \sigma(\mathbf{D}_p)$  is the weighted term. The definition of  $\tilde{\mathbf{D}}_p$ ,  $\mathbf{D}_p$ , and  $\sigma$  are consistent with those in the main paper.

**Formulation of Kinematic Loss  $L_K$ .** As shown in Sec. 3.3 of the main paper and building upon (Tevet et al., 2022b; Liang et al., 2024; Ghosh et al., 2023), we use a kinematic loss term,  $L_K$ , to prevent artifacts like foot sliding or jittering. Moreover, we aim to utilize the kinematic loss  $L_K$  to make our model focus more on the generation of global trajectory. This focus is crucial because, despite the greater importance of global trajectory compared with local motion, the motion representation allocates only 4 values for the global trajectory versus 259 for the local motion. As shown in GMD (Karunratanakul et al., 2023), such a disparity could lead the model to prioritize local motion generation. Therefore, we want to use the kinematic loss  $L_K$  to eliminate such a bias.

Specifically,  $L_K$  consists of 4 subterms, which can be formulated as:

$$L_K = \lambda_{\text{foot}} L_K^{\text{foot}} + \lambda_{\text{vel}} L_K^{\text{vel}} + \lambda_{\text{rot}} L_K^{\text{rot}} + \lambda_{\text{traj}} L_K^{\text{traj}}. \quad (12)$$

Here,  $L_K^{\text{foot}}$ ,  $L_K^{\text{vel}}$ ,  $L_K^{\text{rot}}$ , and  $L_K^{\text{traj}}$  correspond to the foot skating loss, velocity loss, global rotation loss, and global position loss, respectively. The coefficients  $\lambda_{\text{foot}}$ ,  $\lambda_{\text{vel}}$ ,  $\lambda_{\text{rot}}$ , and  $\lambda_{\text{traj}}$  denote the weights assigned to these four loss terms. To compute these losses, we first compute joint positions  $\tilde{\mathbf{P}}_x, \mathbf{P}_x \in \mathbb{R}^{J \times 3T}$  and joint velocities  $\tilde{\mathbf{V}}_x, \mathbf{V}_x \in \mathbb{R}^{J \times 3(T-1)}$  of the generated reaction and ground truth reaction. We further use  $\tilde{\mathbf{R}}_x, \mathbf{R}_x \in \mathbb{R}^T$  to denote the global rotation of the reactor along the y-axis. For clarity, we omit the subscript  $x$  in subsequent formulations.

To compute the foot skating loss  $L_K^{\text{foot}}$ , we first calculate  $\tilde{\mathbf{H}} \in \mathbb{R}^{J \times (T-1)}$ , which signifies the height of each joint across the previous  $T - 1$  frames. The formulation of  $L_K^{\text{foot}}$  is then articulated as follows:

$$C[i] = I(\|\tilde{\mathbf{V}}[i]\|_2 \leq \gamma_v) * I(\tilde{\mathbf{H}}[i] \leq \gamma_h) \quad (13)$$

$$L_K^{\text{foot}} = \frac{1}{\sum_{i \in \text{FootJoints}} C[i]} \sum_{i \in \text{FootJoints}} \|\tilde{\mathbf{V}}[i]\|_2^2 * C[i]. \quad (14)$$

Here,  $\gamma_v$  and  $\gamma_h$  serve as thresholds for calculating  $C$ , where  $C \in \{0, 1\}^{J \times (T-1)}$  indicates the contact between each joint and the ground in each frame.  $\text{FootJoints} \subset \{1, \dots, J\}$  represents the subset of indices corresponding to foot joints.

We use similar equations to compute velocity loss  $L_K^{\text{vel}}$ , global rotation loss  $L_K^{\text{rot}}$ , and global position loss  $L_K^{\text{traj}}$ , which are expressed as follows:

$$L_K^{\text{vel}} = \frac{1}{J * (T-1)} \sum_i \|\tilde{\mathbf{V}}[i] - \mathbf{V}[i]\|_2^2 \quad (15)$$

$$L_K^{\text{rot}} = \frac{1}{T} \|\tilde{\mathbf{R}} - \mathbf{R}\|_2^2 \quad (16)$$

$$L_K^{\text{traj}} = \frac{1}{T} \|\tilde{\mathbf{P}}[\text{root}] - \mathbf{P}[\text{root}]\|_2^2, \quad (17)$$

where ‘root’ denotes the index of the root joint of the reactor.

**Experimental Setup of InterGen.** Originally designed for text-driven human interaction generation, InterGen (Liang et al., 2024) processes a text prompt  $w$  to generate interactions  $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$  between two people with respect to  $w$ . However, it is *not* directly applicable to our task of text-driven human reaction generation. To adapt InterGen for this new task, we integrate an inpainting mechanism into the inference process of InterGen, which is similar to the method described in Sec. 3.4 of the main paper. At each time step  $t$  of the denoising process of InterGen, after estimating the clean interaction  $\tilde{\mathbf{z}}_0 = [\tilde{\mathbf{x}}_0, \tilde{\mathbf{y}}_0]$ , we embed the known actor’s motion into  $\tilde{\mathbf{z}}_0$  to obtain  $\hat{\mathbf{z}}_0$ . This operation can be expressed as:

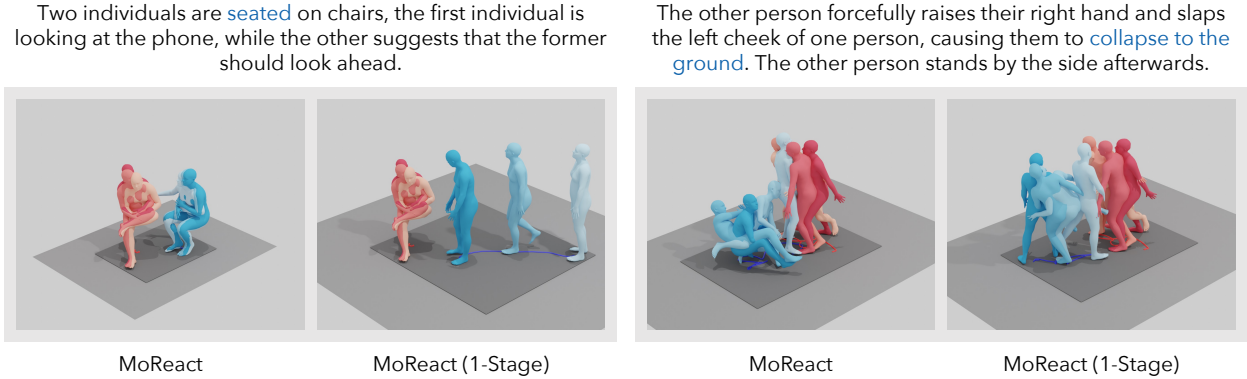
$$\hat{\mathbf{z}}_0 = [\mathbf{1}, \mathbf{0}] \odot \tilde{\mathbf{z}}_0 + [\mathbf{0}, \mathbf{1}] \odot [\mathbf{0}, \mathbf{y}] = [\tilde{\mathbf{x}}_0, \mathbf{y}]. \quad (18)$$

Here,  $\odot$  denotes the Hadamard product. The modified result,  $\hat{\mathbf{z}}_0 = [\hat{\mathbf{x}}_0, \hat{\mathbf{y}}_0]$ , is subsequently utilized to calculate  $\boldsymbol{\mu}_t$  and to sample  $\mathbf{z}_{t-1}$  from  $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ . By employing this inpainting mechanism, we continuously integrate the known actor’s motion  $\mathbf{y}$  throughout the denoising process, guaranteeing that the resulting interaction  $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$  accurately conforms to  $\mathbf{y}$ . Consequently,  $\tilde{\mathbf{x}}_0$  in the ultimate denoising outcome  $\mathbf{z}_0 = [\mathbf{x}_0, \mathbf{y}_0]$  represents the reaction generated with respect to both the textual prompt  $w$  and the actor’s motion  $\mathbf{y}$ .

Through communication with the authors of InterGen (Liang et al., 2024), we discovered that the publicly released checkpoint (Liang et al., 2023) of InterGen was trained using both the training and test sets to produce best demonstrations. Therefore, for a fair comparison, we train InterGen from scratch using their codebase, strictly following the experimental setup presented in their paper.

**Experimental Setup of MDM.** We adapted the official code of MDM to suit the text-driven reaction generation task. Specifically, by concatenating the action and reaction features before feeding them into the model, we enable the model to be aware of the interaction between two people instead of focusing on just one person. We experimented with two backbones for MDM: a transformer encoder-only backbone and a GRU backbone. For the transformer encoder-only backbone, we utilized N=8 blocks, each with a latent dimension of 1,024, and equipped each attention layer with 8 heads. For the GRU backbone, we set N=8 GRU layers with a latent dimension of 1,024. Both models were trained for 2,000 epochs using the AdamW optimizer, consistent with the training settings of MoReact.

**Detailed Model Configurations.** In the transformer-style architecture of our full-body motion diffusion model, we utilize N=8 blocks, each with a latent dimension of 1,024, and we equip each attention layer with 8 heads, consistent with the setup in InterGen (Liang et al., 2024). Before inputting the noised reaction vector  $\mathbf{x}_t$  into the transformer layers, we use a linear layer to adjust its dimension to match the transformer’s input dimension. Similarly, the output from the transformer layers is processed by another linear layer to match the motion feature’s dimension. For text processing, we utilize a frozen CLIP-ViT-L-14 model to encode the text prompt into text features for cross-attention. Moreover, following InterGen, we extract the most salient text feature embedding, combine it with the diffusion timestep feature, and employ this composite feature within the adaptive layer norms of the transformer blocks. To encode the actor’s motion  $\mathbf{y}$ , a transformer encoder layer comprising 2 blocks, a latent dimension of 1,024, and 8 heads per attention

Figure E.1: **Ablation study** on the design choice within MoReact.

layer is utilized prior to incorporating  $\mathbf{y}$  for cross-attention. Except for the absence of a cross-attention layer, the architecture of the trajectory diffusion model mirrors that of the full-body diffusion model.

During training, we use a 1,000-step diffusion process and adopt a classifier-free technique (Ho & Salimans, 2022) that randomly masks 10% of the text conditions, 10% of the actor’s motion conditions, and 10% of the global trajectory condition independently. During inference, we use the DDIM (Song et al., 2020) sampling strategy with 50 time steps and  $\eta = 0$ , and set the classifier-free guidance coefficient  $s = 3.5$ . For the hyperparameters used in the training of the revised model, we set  $(\lambda_R, \lambda_K, \lambda_I, \lambda_K^{\text{foot}}, \lambda_K^{\text{vel}}, \lambda_K^{\text{rot}}, \lambda_K^{\text{traj}}, L_I^p, L_I^v)$  to  $(7.0, 1.0, 1.0, 300.0, 110.0, 1.5, 10, 5.0, 25.0)$ , respectively. In addition, we set the threshold  $\bar{t}$  for applying the kinematic loss  $L_K$  and the interaction loss  $L_I$  as 700.

**Details for experiments on CHI3D dataset.** To demonstrate the generalization ability of MoReact, we adapted it to suit the action-driven reaction generation task and evaluated it on the CHI3D (Fieraru et al., 2020) dataset. Specifically, instead of using CLIP to extract features from text as in the text-driven reaction generation task, we employed a learnable action embedding to encode the action features. Additionally, compared to the architecture shown in Fig. 2(b) of the main paper, we eliminated the cross-attention layer that fuses the textual features into the denoising process. We reduced the latent dimension to 512 and the batch size to 16. The model was trained for 1,000 epochs using the AdamW optimizer. We also made corresponding adjustments to the baseline MDM model (reducing the latent dimension, adjusting the batch size, and training settings) to ensure a fair comparison. We follow the official implementation of ST-GCN (Yan et al., 2018) to build our evaluator, an interaction classifier trained on CHI3D.

## E Additional Ablation Studies

**Two-Stage vs. Single-Stage.** Beyond the quantitative analysis of the design choice in Sec. 4.4 of the main paper, we present some visual results generated by both the two-stage and single-stage frameworks. As illustrated in Fig. E.1 and supplementary video, our two-stage framework generates more natural and text-aligned reactions compared to the single-stage baseline, validating the effectiveness of our two-stage approach.

**Predicted Term of Trajectory Diffusion Model.** As mentioned in Sec. 3.2 of the main paper, diffusion models can employ two kinds of strategies during the denoising process to derive  $\mathbf{x}_{t-1}$  from the noised data  $\mathbf{x}_t$ : predicting the noise  $\epsilon$ , or predicting the clean data  $\mathbf{x}_0$ . Here, we conduct experiments to determine which approach is more effective for the trajectory diffusion model. The results, displayed in Table E.1, indicate that the variant focusing on noise prediction  $\epsilon$  outperforms, aligning with the conclusions drawn by GMD (Karunratanakul et al., 2023).

Table E.1: Ablation studies on predicted term of trajectory diffusion model. The trajectory model that predicts  $\epsilon$  achieves better performance in R-precision, FID and Multi-Modality Distance.

Methods	Traj. Model	3-Precision <sup>↑</sup>	FID <sup>↓</sup>	MM Dist <sup>↓</sup>	Diversity <sup>→</sup>
Real	-	0.704 $\pm$ 0.005	0.206 $\pm$ 0.009	3.784 $\pm$ 0.001	7.799 $\pm$ 0.031
MoReact	predict $\mathbf{x}_0$	0.568 $\pm$ 0.006	2.959 $\pm$ 0.030	3.826 $\pm$ 0.001	<b>7.808</b> $\pm$ 0.030
MoReact	predict $\epsilon$	<b>0.615</b> $\pm$ 0.007	<b>2.412</b> $\pm$ 0.050	<b>3.813</b> $\pm$ 0.002	7.775 $\pm$ 0.046

**Interaction Loss.** While some existing work also employed interaction loss to facilitate interaction generation, their implementations differ from ours in some important aspects. For example, ReMoS (Ghosh et al., 2023) only considers corresponding joints of the interacting individuals in its interaction loss, thus failing to capture diverse joint interaction patterns present in real-world scenarios. InterGen (Liang et al., 2024), on the other hand, does not incorporate a weighting mechanism, preventing it from effectively penalizing unrealistic close interactions or appropriately de-emphasizing irrelevant distant ones. In contrast, our interaction loss introduces a novel weighting mechanism that dynamically adjusts the importance of joint pairs based on both ground-truth and generated interactions, thereby enabling more realistic reaction generation. Additionally, we re-implemented the interaction losses employed by InterGen and ReMoS within MoReact and conducted quantitative comparisons with our method. As demonstrated in Table E.2, our approach consistently achieves superior performance in terms of R-precision, FID, and MM Dist, highlighting the effectiveness of our weighted interaction loss.

Table E.2: Quantitative comparison of different interaction loss designs. Our weighted interaction loss consistently outperforms InterGen (Liang et al., 2024) and ReMoS (Ghosh et al., 2023) losses on R-precision, FID, and MM Dist, demonstrating its superior effectiveness in generating realistic reactions.

Methods		3-Precision <sup>↑</sup>	FID <sup>↓</sup>	MM Dist <sup>↓</sup>	Diversity <sup>→</sup>
Real		0.704 $\pm$ 0.005	0.206 $\pm$ 0.009	3.784 $\pm$ 0.001	7.799 $\pm$ 0.031
InterGen	(Liang et al., 2024) Loss	0.596 $\pm$ 0.009	3.436 $\pm$ 0.075	3.826 $\pm$ 0.002	7.887 $\pm$ 0.039
ReMoS	(Ghosh et al., 2023) Loss	0.608 $\pm$ 0.007	2.817 $\pm$ 0.070	3.819 $\pm$ 0.002	<b>7.792</b> $\pm$ 0.035
MoReact		<b>0.615</b> $\pm$ 0.007	<b>2.412</b> $\pm$ 0.050	<b>3.813</b> $\pm$ 0.002	7.775 $\pm$ 0.046

## F Limitations and Social Impacts

**Limitations and Future Work.** MoReact is designed to generate reactions by considering both textual descriptions and the motion of another individual. Future research will aim to generalize our method to broader contexts, for example, generating reactions based on text and the motions of multiple people.

**Potential Social Impact.** We recognize the potential application of reaction synthesis in military training contexts. With our model, the military might generate a virtual soldier who can dodge and counteract in response to a real soldier’s movements, thereby simulating authentic battlefield scenarios to train soldiers.