Contact Matrix: Enhancing Dance Motion Synthesis with Precise Interaction Modeling

Supplementary Material

6. Network Architecture Details

6.1. VQ-VAEs

The encoder and decoder used in our VQ-VAEs follow the same network design as those in Duolando [46]. Specifically, the encoder uses a latent embedding width of 512 and contains two convolutional blocks, where each block begins with a 1D convolution with a stride 2 for temporal downsampling and is followed by a residual module with depth 3, a dilation growth rate of 3, and a convolution multiplier of 1. With this structure, the encoder reduces the temporal resolution by a factor of 4, allowing the features to capture contextual poses within the motion sequence. The encoded features are then passed into a vector quantization bottleneck with 512 codebook entries. The decoder mirrors the encoder, also with a latent embedding width of 512 and two blocks, where each block contains a residual module followed by a transposed 1D convolution that restores the temporal resolution. Reversed dilation is used in the decoder to match the encoder receptive field and support accurate reconstruction of the motion sequence.

As described in Sec. 3.2, based on this backbone, we implement three variants for our task: (1) PartFusion-VQ for modeling body poses, (2) a standard VQ-VAE for modeling the follower's global trajectory, and (3) another standard VQ-VAE for modeling the contact matrices between the follower and the leader. In PartFusion-VQ, the human body with 54 joints is divided into four parts: the upper body with $J_U = 15$ joints, the lower body with $J_D = 9$ joints, the left hand with $J_L = 15$ joints, and the right hand with $J_R = 15$ joints. Each part is encoded with its own encoder and codebook.

6.2. RCDiff

We adopt the transformer architecture used in ReMoS [15] to build our diffusion model. The input and condition tokens are first projected into a 512-dimensional latent space and then processed by a stack of 8 transformer blocks. Within each block, temporal self attention captures dependencies along the input sequences, followed by cross attention over the conditions to incorporate leader motion and music information. The resulting features are refined by a feed-forward network, whose output is adaptively modulated by scale and shift parameters derived from the timestep embedding to condition the representation on the current diffusion step. Residual connections are applied throughout to facilitate incremental updates.

7. Additional Ablation Studies

We further present several ablation studies to demonstrate the effectiveness of our method and to explain the rationale behind some of our design choices.

7.1. Ablation on Classifier-Free Guidance

Diffusion models typically employ classifier-free guidance (CFG) [19] to control the influence of conditional inputs on the generation process. We explore the effect of different CFG scales as well as varying the conditional dropout rate during training, where the condition is randomly replaced with a zero vector. As shown in Tab. 4, when the model is completely unconditional (*i.e.*, CFG scale = 0), the generated results are poor. Even if the drop rate is increased from 0.1 to 0.25, so that unconditional cases are more thoroughly trained, performance remains low without leader motion or music guidance. This is reflected not only in interaction metrics but also in solo metrics, as even individual motions in two-person dances strongly depend on the conditions, particularly the partner movements.

Furthermore, Tab. 4 reveals several trade-offs, where introducing CFG improves some metrics but leads to a decline in others. For example, when the drop rate is 0.25 and the CFG scale is 4.0, both FID_k and FID_{cd} decrease, but FID_g rises from 32.28 to 82.43. Overall, CFG provides no clear benefit for this task and is thus not used in our method.

7.2. Number of Diffusion Sampling Steps

Diffusion models require iterative denoising during inference, which makes sampling time-consuming. To speed up this process, DDIM [48] is often employed. Using this strategy, we investigate the effect of different numbers of sampling steps. We use 1000 diffusion steps during training. As shown in Tab. 5, unlike image and video generation tasks, reducing the number of sampling steps in motion generation significantly degrades performance, a phenomenon also observed in MDM [52]. When the number of steps is halved, both FIDk and FIDcd increase several times, and FID_g also rises slightly. With such poor FID scores, the slight improvement in diversity (Div) is insufficient to indicate good generation quality. Therefore, we ultimately adopt 1000 sampling steps.

7.3. Accuracy of Contact Matrix Prediction

We first evaluate the accuracy of our contact-matrix VQ-VAE using standard metrics, obtaining Precision = 0.748, Recall = 0.693, and F1 score = 0.704 at a 0.5 threshold.

Table 4. Quantitative results for different classifier-free guidance (CFG) scales and conditional dropout rates.

		Solo Metrics				Interactive Metrics				Rhythmic
Drop rate	CFG scale	$FID_k(\downarrow)$	$FID_g(\downarrow)$	$Div_k(\rightarrow)$	$Div_g(\rightarrow)$	$FID_{cd}(\downarrow)$	$Div_{cd}(\rightarrow)$	CF(%)	BED(↑)	BAS(↑)
Ground Truth		6.56	6.37	11.31	7.61	3.41	12.35	74.25	0.5308	0.1839
0.1	0.0	46.81	46.00	10.77	4.86	83.56	6.36	82.88	0.3696	0.2144
0.1	0.5	17.59	42.70	8.94	**7.44**	45.97	7.76	**71.14**	0.4205	0.2024
0.1	1.0	8.97	55.75	9.33	9.26	6.68	10.62	62.36	0.4704	0.2038
0.1	2.0	7.72	42.17	9.29	8.49	**3.86**	**10.96**	55.50	0.4490	0.1989
0.1	4.0	8.95	51.25	9.35	8.88	6.75	10.60	62.47	**0.4726**	0.1998
0.1	6.0	11.26	51.15	9.34	8.52	8.44	10.02	56.12	0.4353	0.2099
0.25	0.0	83.71	48.40	**11.44**	5.75	22.18	8.67	66.33	0.3326	0.2124
0.25	0.5	18.78	52.86	9.77	8.23	30.83	8.32	67.46	0.4136	0.2006
0.25	1.0	7.61	39.74	9.77	8.03	9.85	10.23	64.19	0.4530	0.2042
0.25	2.0	7.54	39.86	9.85	8.04	9.81	10.23	63.87	0.4518	0.2043
0.25	4.0	**7.11**	82.43	9.71	10.61	6.42	10.47	56.19	0.4198	0.2087
0.25	6.0	8.26	63.28	9.69	9.61	7.36	10.29	55.68	0.4216	**0.2179**
RCDiff		8.89	**32.28**	9.53	7.35	8.01	10.42	61.58	**0.4606**	0.2050

Table 5. Quantitative results for motion generation with different numbers of diffusion sampling steps.

	Solo Metrics				Interactive Metrics				Rhythmic
Steps	$FID_k(\downarrow)$	$FID_g(\downarrow)$	$Div_k(\rightarrow)$	$Div_g(\rightarrow)$	$FID_{cd}(\downarrow)$	$Div_{cd}(\rightarrow)$	CF(%)	BED(↑)	BAS(↑)
Ground Truth	6.56	6.37	11.31	7.61	3.41	12.35	74.25	0.5308	0.1839
100	44.72	83.58	9.90	5.16	276.96	4.34	100.00	0.3865	**0.2394**
200	44.54	59.69	10.05	5.78	277.97	4.97	99.88	0.4169	0.2374
500	28.13	36.39	**10.08**	6.25	188.53	5.61	89.48	**0.4675**	0.2120
1000	**8.89**	**32.28**	9.53	**7.35**	**8.01**	**10.42**	**61.58**	0.4606	0.2050

To examine the influence of contact information on motion generation, we also perform guided sampling using ground-truth contact matrices. As shown in Tab. 6, predicted contacts already provide strong guidance for solo, interactive, and rhythmic metrics, while using ground-truth contacts leads to further improvements. This demonstrates that our contact-matrix guidance is effective and the predicted contacts are sufficiently accurate for practical use.

7.4. Contact Guidance Injection Frequency

Since our diffusion model uses 1000 sampling steps during inference, we attempt to apply contact guidance to only a subset of these steps rather than all of them, in order to avoid excessive additional computation. Specifically, we experiment with applying guidance during the first 200 steps (Early), the last 200 steps (Late), and every 5 steps (Interval) throughout the sampling process.

As shown in Tab. 8, applying guidance during the first 200 sampling iterations (*i.e.*, timestep 1000–800) yields poor results, as the contact matrix is still highly unstable at this stage. Applying guidance during the last 200 iterations (*i.e.*, timestep 200–0), when the contact matrix is mostly determined, leads to significant improvements in interaction metrics such as FID_{cd} and Div_{cd}. However, at the late stage of sampling, the overall motion structure is already largely established, and forcing changes through contact guidance may harm the fidelity of the follower motion. For example, FID_g increases by 1.42. Therefore, we ultimately adopt an interval-based strategy, inserting guidance periodically throughout the sampling process, which achieves a good balance between interaction accuracy and motion fidelity.

7.5. Ablation on Contact Sparsity

Although we simplify all hand joints into a single representative joint, we still construct a dense contact matrix for the remaining 23 joints without further filtering (Section 3.1). To examine whether a dense contact representation is necessary, we conduct an ablation comparing this dense matrix with a sparse variant. Specifically, we construct sparse contact matrices by retaining only the top-20 most frequent joint pairs, as visualized in the contact-frequency heatmap in Fig. 6. Table 8 reports the quantitative results. On solo motion metrics, the sparse matrix slightly outperforms in FID_k (7.21 vs. 8.89) but underperforms in FID_g (36.35 vs. 32.28). For interactive metrics, the sparse matrix achieves lower FID_{cd} (6.98 vs. 8.01) and higher Div_{cd} (10.61 vs. 10.42), yet the dense matrix still achieves higher contact fidelity (CF: 61.58% vs. 60.34%) and better rhythmic-

Table 6. Effect of predicted versus ground-truth contact matrices on motion generation quality.

	Solo Metrics				Interactive Metrics				Rhythmic
Steps	$FID_k(\downarrow)$	$FID_g(\downarrow)$	$Div_k(\rightarrow)$	$Div_g(\rightarrow)$	$FID_{cd}(\downarrow)$	$Div_{cd}(\rightarrow)$	CF(%)	BED(↑)	BAS(↑)
Ground Truth	6.56	6.37	11.31	7.61	3.41	12.35	74.25	0.5308	0.1839
GT Matrix	9.57	34.79	**9.84**	7.04	**7.08**	10.11	58.23	**0.4713**	0.2031
Pred Matrix	**8.89**	**32.28**	9.53	**7.35**	8.01	**10.42**	**61.58**	0.4606	**0.2050**

Table 7. Quantitative results for different contact guidance injection frequencies during diffusion sampling.

	Solo Metrics				Interactive Metrics				Rhythmic
Steps	$FID_k(\downarrow)$	$FID_g(\downarrow)$	$Div_k(\rightarrow)$	$Div_g(\rightarrow)$	$FID_{cd}(\downarrow)$	$Div_{cd}(\rightarrow)$	CF(%)	BED(↑)	BAS(↑)
Ground Truth	6.56	6.37	11.31	7.61	3.41	12.35	74.25	0.5308	0.1839
Early	166.01	74.67	13.91	8.04	24.49	8.46	52.78	0.3222	**0.2264**
Late	**7.75**	33.70	9.35	**7.76**	**3.44**	**11.18**	59.68	0.4536	0.1995
Interval	8.89	**32.28**	**9.53**	7.35	8.01	10.42	**61.58**	**0.4606**	0.2050

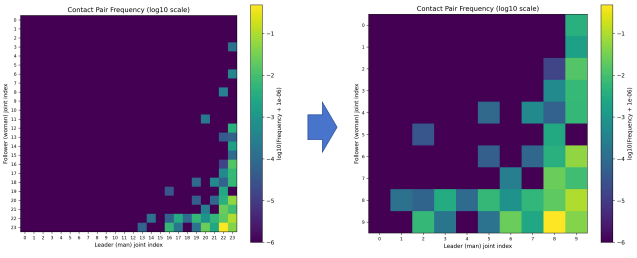

Figure 6. **Contact-frequency heatmap of DD100 [46].** Left: original heatmap showing all 23×23 pairs. Right: heatmap after retaining only the top-20 most frequent pairs.

ity (BED: 0.4606 vs. 0.4237, BAS: 0.2050 vs. 0.1998). Overall, the differences between the dense and sparse contact matrices are modest. While the dense matrix may incur slightly higher storage and computational costs, retaining all joint pairs allows our method to generalize beyond DD100-specific frequent contacts. Therefore, in our standard setting, we keep all joints in the contact matrix.

8. Further Qualitative Comparisons

In this section, we provide additional qualitative comparisons with Duolando [46] to demonstrate the effectiveness of our method further. As shown in Figs. 7 to 11, our method typically produces follower motion that coordinates well with the leader. For clearer illustration, we present a visualization every 5 frames so that larger and more meaningful motion changes can be observed.

In Fig. 7, the follower generated by our method moves toward the leader at the appropriate moment. Duolando performs a correct action, but fails to approach the leader in time, especially in the last four frames. It is also worth noting that the joint position produced by Duolando results in an implausible pose, as shown by the skeletons, where the upper body of the follower rotates by 180 degrees from frame 9 to frame 12.

In Fig. 8, our method correctly interprets the leader's intention and responds with suitable movements. Although the hands of the two dancers do not fully make contact between frames 6 and 11, they remain close, and the contact in the other frames is accurate. In comparison, Duolando also understands the leader's intention, but its inaccurate relative-position prediction makes the overall result appear unnatural. Fig. 9 shows similar situations. Both our method and Duolando can infer the leader's intended actions, yet our method generates more accurate relative positions, leading to motion that appears more coherent and natural.

In Fig. 10, because Duolando does not explicitly or implicitly model the detailed contact between the two dancers, the hands and arms of the follower and the leader fail to form reasonable contact. Moreover, even in Fig. 11, where the follower and leader have almost no physical contact, our method still produces reasonable results. However, because our contact modeling is performed at the skeleton level, mesh interpenetration occurs when the leader's hand is placed on the follower's shoulder.

Table 8. Quantitative comparison of dense and sparse contact matrices for motion generation.

	Solo Metrics				Interactive Metrics				Rhythmic
Steps	$FID_k(\downarrow)$	$FID_g(\downarrow)$	$Div_k(\rightarrow)$	$Div_g(\rightarrow)$	$FID_{cd}(\downarrow)$	$Div_{cd}(\rightarrow)$	CF(%)	BED(↑)	BAS(↑)
Ground Truth	6.56	6.37	11.31	7.61	3.41	12.35	74.25	0.5308	0.1839
sparse	**7.21**	36.35	9.23	7.84	**6.98**	**10.61**	60.34	0.4237	0.1998
dense	8.89	**32.28**	**9.53**	**7.35**	8.01	10.42	**61.58**	**0.4606**	**0.2050**

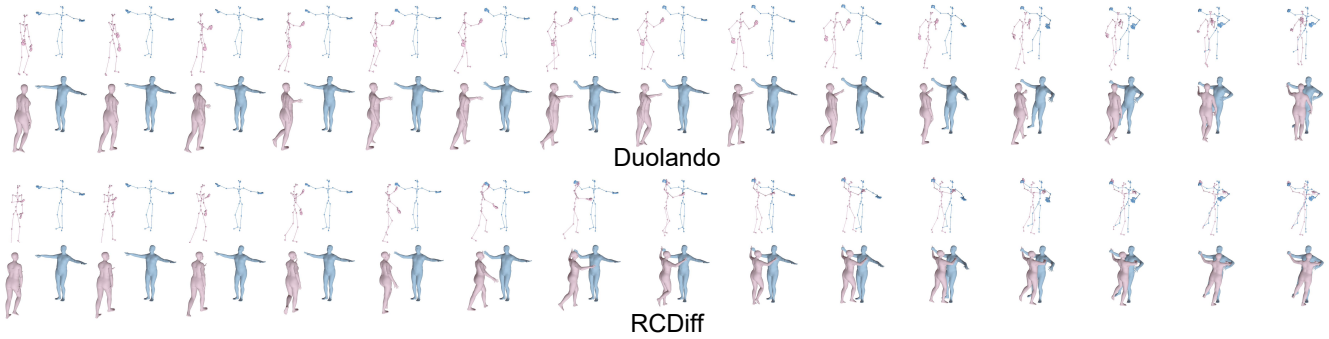

Figure 7. **Qualitative comparisons of Duolando [46] and our method RCDiff on Foxtrot.** The blue mesh represents the leader, while the pink mesh shows the generated follower, corresponding to the reactive motion produced by the models.

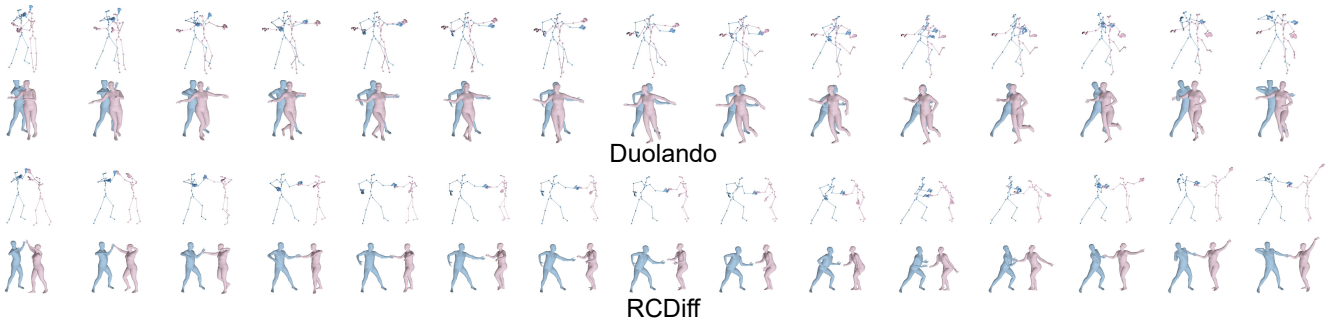

Figure 8. **Qualitative comparisons of Duolando [46] and our method RCDiff on Lumba.** The blue mesh represents the leader, while the pink mesh shows the generated follower, corresponding to the reactive motion produced by the models.

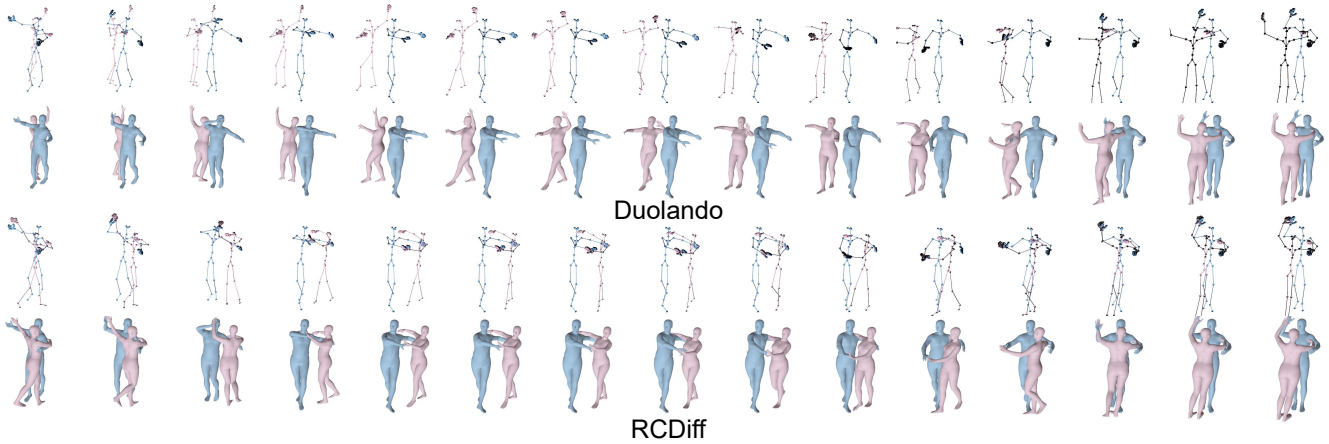

Figure 9. **Qualitative comparisons of Duolando [46] and our method RCDiff on PasoDable.** The blue mesh represents the leader, while the pink mesh shows the generated follower, corresponding to the reactive motion produced by the models.

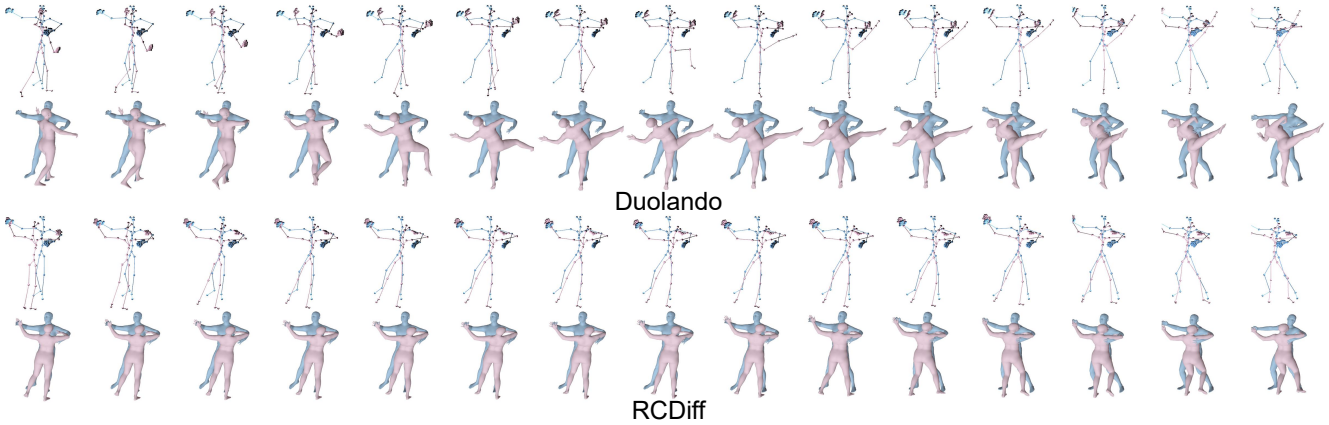

Figure 10. **Qualitative comparisons of Duolando [46] and our method RCDiff on Quickstep.** The blue mesh represents the leader, while the pink mesh shows the generated follower, corresponding to the reactive motion produced by the models.

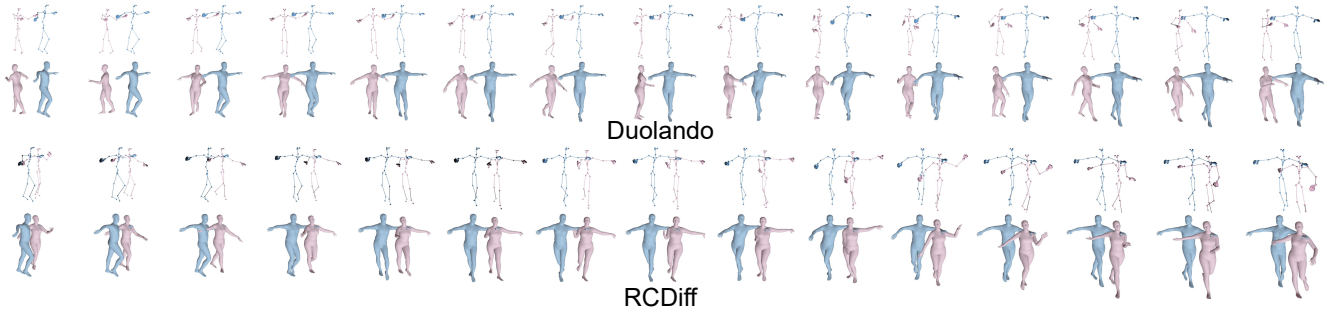

Figure 11. **Qualitative comparisons of Duolando [46] and our method RCDiff on Samba.** The blue mesh represents the leader, while the pink mesh shows the generated follower, corresponding to the reactive motion produced by the models.