Breaking the Group Size Barrier: Parameter-Efficient Group Dance Generation with Chain-of-Dancers

Jing Xu **Cunjian Chen*** **Qiuhong Ke**
Department of Data Science and Artificial Intelligence
Faculty of Information Technology
Monash University

Abstract

Group dance generation aims to synthesize coordinated multi-dancer choreography from music, with broad applications in animation and interactive content creation. This task requires modeling dense inter-person dependencies to ensure spatial coordination, while naturally preserving individual dancer identities. Existing approaches model all dancers jointly with end-to-end transformers, which tie the architecture to a fixed group size and entangle per-dancer identities across frames. We propose **ChainDance**, a scalable framework that *reformulates* group dance generation as a *Chain-of-Dancers*: a sequential decomposition over per-dancer conditional distributions, allowing a single model to scale across variable group sizes without retraining and naturally preserving per-dancer identity. Built on a frozen single-dancer diffusion backbone, ChainDance introduces two lightweight modules: a **Role-Aware Text Encoder** (RATE) for per-dancer semantic conditioning, and a **Group-Aware Motion Encoder** (GAME) that aggregates previously generated dancers via a distance-weighted graph convolutional network, and incorporates a training-free noise optimization procedure at inference time to enforce global spatial coherence. Experiments on AIOZ-GDance demonstrate that ChainDance achieves state-of-the-art motion quality and group coordination while structurally preserving per-dancer identity, with 3-4$\times$ fewer parameters and requiring 3-6$\times$ less training time compared to prior approaches.

1 Introduction

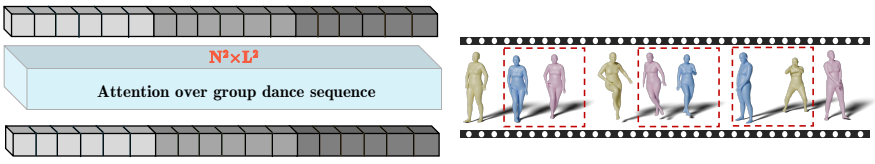
Group dance generation aims to synthesize coordinated multi-person motion from music while preserving individual identities and interactions [28, 40]. Despite recent progress, existing approaches [16, 39, 3] are fundamentally limited by how they formulate the problem. Most methods model all dancers jointly within a fixed-size representation, coupling group size to model architecture. This leads to two inherent constraints (Figure 1): (1) **group-size rigidity**, requiring retraining to handle different numbers of dancers, and (2) **identity entanglement**, where individual dancer characteristics are obscured by shared representations.

These limitations are not merely architectural inefficiencies. They stem from a deeper assumption that group choreography must be generated as a single joint process. In contrast, we ask: can group choreography be constructed as a composition of individual motion processes while still capturing inter-person coordination? This question is grounded in real choreographic practice, where professional choreographers first match music-aligned motions to performers, then design spatial formations.

In this work, we answer this question affirmatively by introducing a new formulation of group dance generation as a sequential conditional decomposition over dancers, which we term a *Chain-of-*

*Corresponding author: cunjian.chen@monash.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Intensive attention computation when retraining for different group sizes.

(b) Identity entanglement problem.

Figure 1: **Key challenges in group dance generation.** (a) Group-size rigidity: adapting to a new group size N requires retraining from scratch, but self-attention over the concatenated sequence scales as $O((NL)^2)$, making per-size retraining prohibitively expensive. As a result, existing models are locked to a fixed group size. (b) Identity entanglement: joint representations obscure per-dancer identity, causing frequent swaps across frames. We visualize three consecutive frames from CoDancers [39]: between Frame 1 and Frame 2, the pink and blue dancers abruptly swap positions; in Frame 3, the blue dancer suddenly relocates to the leftmost position. When rendered as video, these inconsistencies appear as visible flickering and identity-switching artifacts.

Dancers. Rather than treating dancers as simultaneously generated tokens, we model each dancer as conditioned on previously generated ones, enabling flexible scaling to arbitrary group sizes while preserving consistent identity structure. This perspective shifts the problem from joint generation to structured composition, decoupling group size from model capacity.

This reformulation enables a single pretrained single-dancer model [9, 6, 25] to generalize to multi-dancer scenarios without retraining, while still capturing global coordination through explicit inter-dancer conditioning. Crucially, it allows identity to be preserved by construction, rather than enforced through architectural constraints.

Building on this formulation, we introduce a lightweight instantiation that injects role-specific semantics and inter-dancer interactions into a frozen backbone. We further incorporate a training-free inference procedure [24] to enforce global spatial coherence. Together, these design choices operationalize the proposed decomposition while maintaining parameter efficiency.

Experiments demonstrate that this reformulation achieves strong motion quality and coordination across varying group sizes, while significantly reducing training cost. More importantly, our results suggest that structured decomposition is a viable alternative to joint modeling for multi-agent motion generation, opening a new direction for scalable and controllable choreography synthesis. Our contributions can be summarized as follows:

- **Reformulating group dance generation.** We propose a *Chain-of-Dancers* formulation that models group choreography as a sequential conditional composition of individual dancers, shifting from joint generation to structured decomposition and addressing group-size rigidity and identity entanglement.
- **Scalable and identity-preserving generation.** Our approach decouples group size from model architecture, enabling a single pretrained model to generalize to variable numbers of dancers without retraining, while preserving each dancer's identity by construction.
- **Efficient coordination with minimal overhead.** We instantiate the formulation with a Role-Aware Text Encoder, a Group-Aware Motion Encoder, and a training-free inference strategy for global coherence, achieving strong results with significantly reduced training cost.

2 Related Work

2.1 Music-Driven Single Dance Generation

Music-to-dance generation targets rhythmic and stylistically consistent motion synthesis [12, 30, 31]. Transitioning from rule-based heuristics [15, 13, 27] to deep learning on large datasets [18, 22, 20, 29], various architectures have emerged: RNNs and autoregressive models capture transitions but struggle with error accumulation [1, 37], while Transformers and GANs/VAEs improve long-range coherence and realism [32, 5, 18, 11, 8]. Currently, diffusion models define the state-of-the-art via audio-aligned denoising [34, 42]. However, most methods focus on music-only conditioning, limiting controllability, semantic expressiveness, and global coherence.

2.2 Group Dance Generation

Group dance generation requires spatial coordination, identity preservation, and semantic diversity across multiple performers [40, 34]. Many works adopt a joint modeling strategy that concatenates all dancer motions and processes them with transformer backbones, e.g., GDanceR [17], GCD [16], TCDiff [3], and CoDancers [39]. While capturing global synchronization, this design fixes the dancer dimension and requires retraining for each group size, blurs per-dancer identity through uniform tokenization, and typically offers only music-only conditioning, widening the semantic gap between music and motion [35, 18]. These limitations motivate frameworks that scale across group sizes without retraining, support joint music–text control, and explicitly preserve per-dancer identity. Variable group-size modeling has also been explored in multi-person motion anticipation. Social Diffusion [33] uses a fully convolutional diffusion model with order-invariant aggregation and allows the number of people to differ between training and inference. It predicts future motion from observed histories, whereas our work generates music- and text-conditioned group choreography by adapting a single-dancer generative prior. A complementary line of work models pairwise leader-follower interactions [29] but is restricted to fixed two-person generation. In contrast, we generalize this paradigm to variable group sizes via a shared bidirectional pair-wise model with graph-based group conditioning.

2.3 Multimodal Conditioning with Text and Music

Controlling human motion with semantic and rhythmic cues has been widely studied [35, 2]. Text-to-motion methods map language to coherent movements [41, 43], while music-conditioned models learn beat- and phase-aligned trajectories [45, 19, 22]. Recent works fuse text and audio via cross-modal encoders to couple semantic intent with temporal structure [38, 7], improving controllability over unimodal conditioning. However, simultaneous text-and-music control for single dance is still limited. Most models use music alone or coarse style tags rather than free-form language [34, 10]. The gap widens in *group dance*, where role-aware conditioning is rare due to scarce multi-person annotations [44]. Moreover, many multimodal pipelines emphasize global synchronization or average style, causing identity permutation and weak role differentiation in group dance settings. We address this gap by enabling coordinated group generation under *both* text and music while explicitly maintaining dancer-specific identity and role consistency.

3 Method

Given an input music clip M and an optional dance instruction T (style, dancer count, interaction type), our goal is to synthesize a spatiotemporal group dance $\{\mathbf{X}_{1:L}^{(i)}\}_{i=1}^{N}$, where $\mathbf{X}_l^{(i)}$ denotes the motion of dancer i at time l.

Rather than modeling all dancers jointly, we introduce a *Chain-of-Dancers* formulation, where group choreography is constructed as a sequence of conditional single-dancer generations. This shifts group generation from joint modeling to structured composition, enabling scalable and identity-consistent synthesis.

This formulation introduces three key challenges: (a) how to inject inter-dancer dependencies into a single-dancer backbone, (b) how to support variable group sizes without retraining, and (c) how to enforce global coordination beyond local interactions.

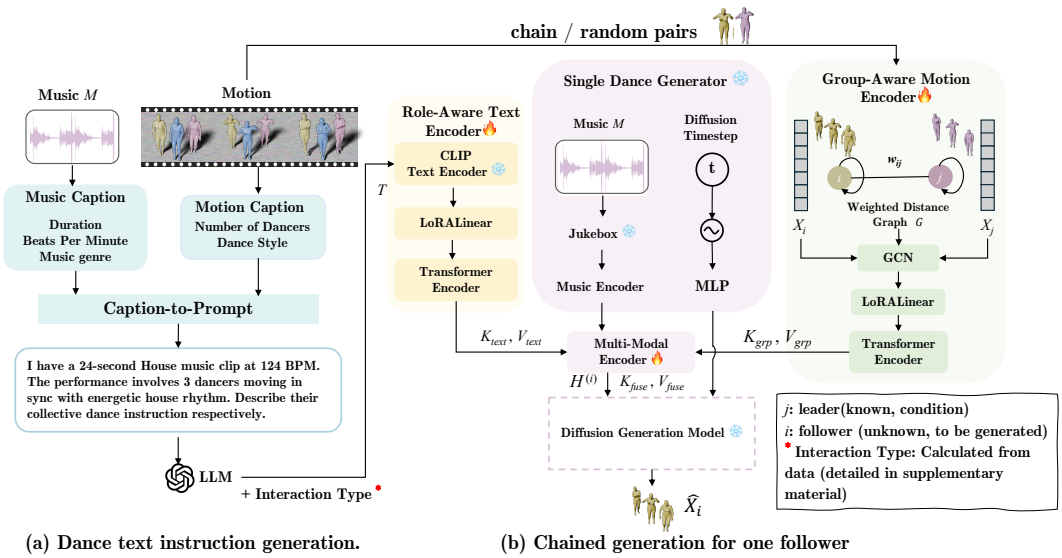
We address these challenges through **ChainDance** (Figure 2), a unified framework that combines conditional generation (§3.1), pairwise decomposition (§3.2), and inference-time optimization (§3.3).

3.1 Multimodal Single-Dancer Generation and Inter-Dancer Interaction Modeling

To realize the proposed decomposition, each dancer must be generated conditionally on both global instructions and previously generated dancers. We achieve this by augmenting a pretrained single-dancer diffusion model with two types of conditioning: (a) semantic conditioning from text (roles, interactions, style), and (b) relational conditioning from previously generated dancers.

Rather than modifying the backbone architecture, we inject these signals through lightweight conditioning pathways, allowing the model to remain fixed while adapting to group scenarios. This design ensures that each dancer retains individual identity, and coordination emerges through explicit conditioning, not shared latent entanglement.

Textual Instruction Generation. As shown in Figure 3, we first merge structured multimodal metadata (music captions such as tempo and genre, motion captions such as style and interaction type)



(a) Dance text instruction generation. (b) Chained generation for one follower

Figure 2: **Overall framework of ChainDance.** The model builds on a frozen single dance diffusion backbone and introduces lightweight multimodal encoders for text, music, and other dancers. Through sequential chain generation and inference-time global optimization, ChainDance produces coherent and coordinated group choreographies.

into a global choreography prompt. We then use an LLM (ChatGPT-4o) to generate a textual dance instruction T that aligns with the music M. This global instruction is decomposed into individualized instructions based on the pairing strategy. Each dancer pair is assigned an interaction keyword (e.g., *approach*, *keep distance*, *face to*, *lead follow*, *swap*, *passby*) to encourage diverse yet coherent dancer-aware behaviors. Existing group dance datasets provide high-quality kinematic trajectories but lack fine-grained textual annotations of inter-dancer interactions. We bridge this gap by automatically deriving interaction keywords from motion samples via geometrically-grounded rule-based detection (Appendix C), serving as approximate yet effective weak supervision (Table 3, w/o RATE).

Lightweight Multimodal Encoders. While the decomposition reduces complexity, coordination still requires modeling interactions between dancers. We capture these interactions through two complementary mechanisms: (a) *textual interaction cues*, which specify roles and interaction types (e.g., approach, follow, keep distance), and (b) *motion-based relational context*, derived from previously generated dancers. These signals provide structured guidance for coordination, and importantly, interaction modeling is explicit and compositional, rather than implicitly entangled in a joint representation.

To realize these two mechanisms, two parameter-efficient encoders are inserted into the cross-attention layers of the frozen single-dancer diffusion backbone $\mathcal{B}$. Pretrained single-dancer models condition only on a global instruction, whereas group choreography requires each dancer to carry role-specific information (e.g., who leads, who interacts). To bridge this gap, the *Role-Aware Text Encoder* (RATE) encodes dancer-aware instructions using a CLIP text encoder [26], projects them through a low-rank

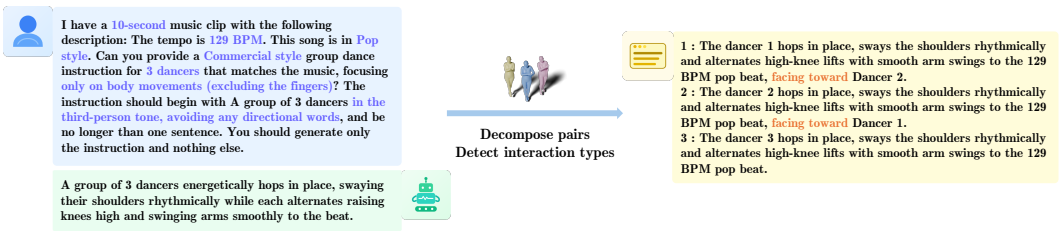


Figure 3: **Prompt generation and role-specific conditioning.** The system converts structured multimodal captions into individualized text prompts for each dancer. Based on a global choreography description, dancer pairs are randomly sampled and assigned interaction keywords (e.g., *approach*, *keep distance*, *face to*, *lead follow*, *swap*, or *passby*), providing diverse yet coherent textual conditions for ChainDance.

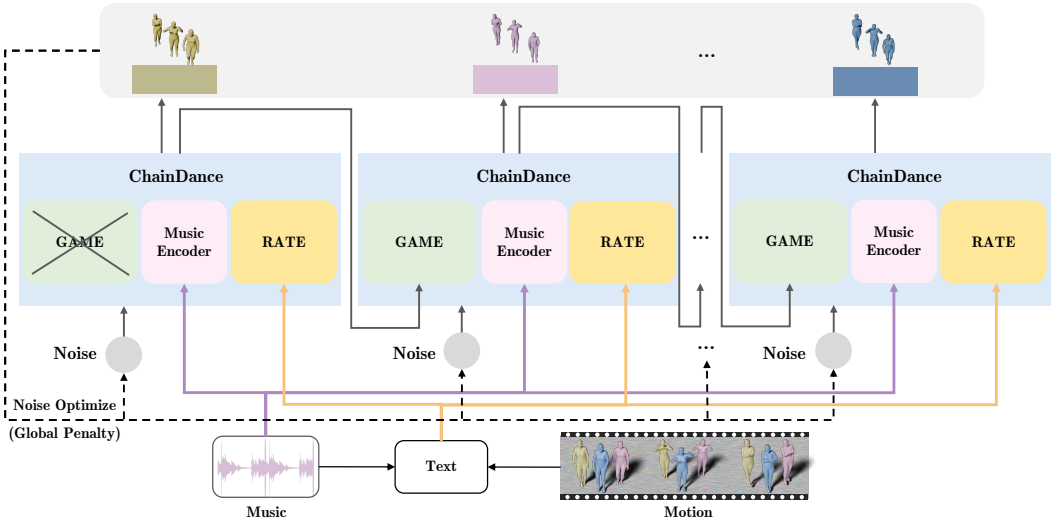


Figure 4: **Chain-of-Dancers generation design.** At inference, ChainDance generates dancers sequentially and refines global consistency via diffusion-based optimization. Each dancer is conditioned on the previously generated one, and this process repeats until the whole group is complete. After that, optimize the shared initial noise via gradient descent until the global penalty L converges below the target threshold. All pairs share the weights of the text and motion encoders.

LoRA module and a lightweight Transformer, and provides them as (K_{text}, V_{text}) in a cross-attention computation:

$$\mathrm{Attn}\big(\mathbf{H}^{(i)}\big) = \mathrm{softmax}\left(\frac{\mathbf{H}^{(i)} K_{text}^{\top}}{\sqrt{d}}\right) V_{text}, \tag{1}$$

where $\mathbf{H}^{(i)}$ denotes the hidden state of dancer i. In this way, style and role information are injected into text embeddings, which are then incorporated into the encoder output (Figure 2) to enable fine-grained semantic control.

The *Group-Aware Motion Encoder* (GAME) conditions on previously generated dancers by constructing a distance-weighted graph G over dancers, where edge weight w_{ij} encodes spatial proximity or interaction strength. A GCN [14, 23, 4] aggregates neighbor features, refined by a Transformer with LoRA updates, to produce motion-aware (K_{grp}, V_{grp}) fused with $\mathbf{H}^{(i)}$ through cross-attention (see Appendix B.1 and B.2). This captures *local* pairwise dependencies. The Role-Aware Text Encoder provides semantic cues via (K_{text}, V_{text}) from textual prompts, while the Group-Aware Motion Encoder introduces spatial context through (K_{grp}, V_{grp}) from previously generated dancer X_j.

All these (K, V) pairs are fed into the multimodal encoders together with the music condition. The encoder outputs are fused (through concatenation and projection) into key-value pairs (K_{fuse}, V_{fuse}) that are applied to the cross-attention layers in the frozen generation model. Through this mechanism, the hidden state of dancer i integrates textual semantics, music, and prior motions for coherent and coordinated choreography. Together, these encoders transform the single-dancer diffusion model into a multimodal generator capable of synchronized and identity-consistent group choreography.

3.2 Pair-wise Decomposition for Scalable Generation

A key consequence of the Chain-of-Dancers formulation is that modeling full joint dependencies is unnecessary. Instead, we approximate group choreography through pairwise conditional generation. Concretely, we decompose the generation process into leader–follower pairs, where each dancer is conditioned on another dancer. A single shared model is trained on such pairs and reused across all positions in the chain. At inference time, group choreography is constructed sequentially: each subsequent dancer is conditioned on previously generated ones, and the same model is reused for all steps. This enables: (a) scalability to varying group sizes and (b) no retraining when adding new dancers.

Pair-wise training. Given a group of N dancers, we decompose the joint distribution into pairwise conditionals via the chain rule (see Appendix A):

$$P(X_1, \ldots, X_N) = P(X_1) \prod_{i=2}^{N} P(X_i \mid X_{1:i-1}). \tag{2}$$

We approximate each conditional $P(X_i \mid X_{1:i-1})$ with a shared pair-wise model: each training sample is a pair (X_j, X_i), where X_j is the conditioning leader and X_i is the follower to be denoised. The same RATE and GAME modules are shared across all pairs, so adding a new dancer at inference adds no learnable parameters.

To avoid directional bias, we adopt **bidirectional pairing**, allowing the model to learn symmetric interactions rather than fixed roles: both (i, j) and (j, i) samples are included, allowing the model to learn mutual conditioning rather than a fixed leader–follower direction. Two pair sampling strategies are explored:

i. **Chain pairing**: pairs follow consecutive dataset indices, e.g., $(D_1, D_2), (D_2, D_3)$. Indices reflect data order, not spatial position.

ii. **Random pairing**: two dancers are uniformly sampled from the same group.

Both yield comparable results in our experiments, and we use chain pairing as the default. The training loss combines a reconstruction term with four auxiliaries:

$$\mathcal{L}_{\text{local}} = \mathcal{L}_{\text{simple}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{contact}}\mathcal{L}_{\text{contact}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}. \tag{3}$$

$\mathcal{L}_{\text{simple}}$ is the reconstruction loss; $\mathcal{L}_{\text{pos}}$, $\mathcal{L}_{\text{vel}}$, $\mathcal{L}_{\text{contact}}$ ensure pose accuracy, smooth transitions, and physical foot–ground contact; $\mathcal{L}_{\text{align}}$ aligns text–music–motion embeddings via cosine similarity. Only RATE and GAME are trainable; the diffusion backbone $\mathcal{B}$ remains frozen.

Inference: from pair to chain. At test time, the same model generates groups of variable size N through sequential conditioning (Figure 4). Starting from X_1 produced by the frozen single-dancer backbone, we iteratively generate dancer X_i by feeding the most recent leader X_j into GAME, which constructs a distance-weighted graph over previously generated dancers to compute spatial weights, then aggregates this context onto the leader feature to produce (K_{grp}, V_{grp}); RATE simultaneously injects the role-specific text prompt for dancer i. The pair-wise model naturally generalizes to chained generation, since each step reuses the same shared encoders and the group-aware motion graph grows incrementally as new dancers are added (see Appendix B.1). Algorithm details see Appendix B.2.

3.3 Global Consistency via Inference-Time Optimization

Sequential generation captures local dependencies but does not guarantee global coherence, such as formation structure or collision avoidance among non-adjacent dancers. Instead of increasing model complexity, we enforce global consistency through a training-free inference-time optimization. Specifically, we refine the shared latent variables to minimize a global objective encoding spatial formation constraints, inter-dancer distance relationships, and collision avoidance. This optimization complements the sequential generation process, enabling global coordination without retraining or architectural changes.

Following the principle of PINO [24], we keep all network parameters frozen and instead optimize the shared diffusion noise via gradient descent, backpropagating a global penalty through the diffusion sampling process. Crucially, because all dancers are generated from a shared latent variable, this optimization provides a unified control signal that simultaneously adjusts all dancers, enabling group-level coordination that cannot be achieved by per-dancer generation alone (see Table 9). The test-time objective decomposes into global controls and overlap penalties,

$$L \;=\; L_{\text{global}} \;+\; L_{\text{overlap}}, \tag{4}$$

where L_{global} encodes formation and orientation constraints, and L_{overlap} discourages inter-person penetration. The global part is a weighted sum of differentiable controls:

$$L_{\text{global}} = \sum_{k \in \mathcal{S}} \lambda_k \, L_k, \tag{5}$$

where $\mathcal{S}$ is the subset of enabled controls (e.g., root position, movement region, orientation, relative distance), chosen to reflect group choreography constraints. The overlap term is implemented as a

hinge on pairwise root distances:

$$L_{\text{overlap}} = \frac{1}{T} \sum_{t} \sum_{i<j} \max\Big(0,\ \delta - \big\|p^{(i)}_{\text{root}}(t) - p^{(j)}_{\text{root}}(t)\big\|_2\Big). \tag{6}$$

At inference, we update only the shared latent to reduce Eqn. (4), which adds a small latency overhead while improving interaction quality.

4 Experimental Results

We evaluate ChainDance in standard single- and group-dance settings. Unless stated otherwise, we freeze the backbone and train only RATE and GAME. We report quality, physical plausibility, group coherence, and efficiency, along with ablation studies.

4.1 Experimental Setup

Baseline Methods. We compare ChainDance with other group dance methods that generate all dancers jointly as high-capacity baselines: **GCD** [16], which integrates group-based contrastive learning with diffusion; **CoDancers** [39], which employs music-aligned choreographic units; **TCDiff** [3], which incorporates trajectory control for temporal modeling; and **ST-GDance** [36], which factorizes temporal and spatial modeling to preserve layouts while avoiding parameter blow-up at scale. All methods share the same music inputs, beat extraction, and evaluation protocols. Each baseline is retrained per group size, while our model is trained once and generalizes to varying group sizes.

Datasets. We pretrain our single-dancer backbones on two standard single dance datasets: AIST++ [22] and FineDance [20]. AIST++ contains 1,408 paired 3D music–dance sequences across 10 genres, performed by 60 professional dancers. FineDance builds upon AIST++ with more diverse styles and longer clips, offering broader coverage of musical and motion patterns. All evaluations are conducted on AIOZ-GDance [17], the only publicly available large-scale group dance benchmark comprising 16.7 hours of music-aligned 3D motion data from over 4,000 dancers, spanning 7 dance styles and 16 music genres. We follow the official train/test split protocol from [17].

Evaluation Metrics. Following standard practice [3, 36], we evaluate both single- and group-level performance, along with model efficiency:

- **Single dance metrics.** Frechet Inception Distance (FID, ↓) assesses distribution fidelity, Generation Diversity (Div, ↑) quantifies sample diversity, and Physical Foot Contact (PFC, ↓) measures foot sliding artifacts.
- **Group dance metrics.** Group Motion Realism (GMR, ↓) measures feature similarity via Frechet Inception Distance; Group Motion Correlation (GMC, ↑) evaluates coherence through cross-correlation between generated dancers; Trajectory Intersection Frequency (TIF, ↓) counts the frequency of inter-person body-part collisions.
- **Efficiency.** We report FLOPs, parameter count, training time per epoch, and inference time.

Repeated Evaluation. For the main comparisons and ablations, we repeat generation and evaluation 20 times with the trained models and test split fixed. The first dancer is resampled in each ChainDance run. We report the mean and sample standard deviation of each metric across the 20 run-level results in Tables 1 and 3.

Training Strategy. We freeze the pretrained backbone and train the two lightweight LoRA encoders. We adopt a *teacher forcing* strategy during training, where each follower conditions on the ground-truth motion of previously generated dancers, perturbed with noise to reduce the train-inference gap. At inference, the model operates auto-regressively, conditioning on its own predictions. We optimize using the Adam optimizer with cosine learning rate decay.

Implementation Details. We train our model on a dual RTX 4090 GPU setup. ChainDance uses the Lodge [21] backbone pretrained on FineDance and employs a chain pairing strategy with bidirectional pairs during training for the main results. Following previous works [3, 35], the loss function includes weighted terms for position ($\lambda_{\text{pos}} = 0.646$), velocity ($\lambda_{\text{vel}} = 2.964$), and contact ($\lambda_{\text{contact}} = 10.942$) to model motion dynamics, along with a cross-modal alignment loss ($\lambda_{\text{align}} = 1.0$) to ensure consistency between the generated motion and the input text/music. For all group sizes, the sequence length is set to 120 frames. Our model was trained for 1000 epochs using the Adam optimizer with an initial learning rate of 0.001 and a cosine learning rate scheduler. At inference, we adopt a conservative KV cache that reuses final-projection key–value states across dancers and maintains a causal temporal cache to avoid recomputing past frames. Code will be released upon acceptance.

Table 1: **Single- and group-dance performance on AIOZ-GDance.** Results are mean ± standard deviation over 20 evaluation runs. Best means are **bold**; second-best means are underlined. * denotes the model without global noise optimization.

Method	1 Dancer			2 Dancers		
	FID↓	Div↑	PFC↓	GMR↓	GMC↑	TIF↓
GCD [16]	39.24 ± 1.02	9.62 ± 0.11	2.56 ± 0.14	34.43 ± 0.41	80.35 ± 0.36	0.16 ± 0.08
CoDancers [39]	23.98 ± 0.33	9.44 ± 0.14	3.54 ± 0.32	24.54 ± 0.20	72.51 ± 0.40	$\mathbf{0.08 \pm 0.04}$
TCDiff [3]	37.33 ± 1.14	$\mathbf{13.95 \pm 0.12}$	0.51 ± 0.06	15.74 ± 0.24	$\mathbf{81.92 \pm 0.44}$	0.12 ± 0.05
ST-GDance [36]	28.88 ± 0.62	12.81 ± 0.10	0.97 ± 0.06	19.44 ± 0.17	80.52 ± 0.66	0.12 ± 0.05
ChainDance*	$\mathbf{23.11 \pm 0.23}$	13.83 ± 0.11	$\mathbf{0.50 \pm 0.04}$	14.11 ± 0.31	81.54 ± 0.45	0.11 ± 0.04
ChainDance	$\mathbf{23.11 \pm 0.23}$	13.83 ± 0.11	$\mathbf{0.50 \pm 0.04}$	$\mathbf{13.63 \pm 0.29}$	81.32 ± 0.50	0.12 ± 0.03

Method	3 Dancers			4 Dancers		
	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓
GCD [16]	30.22 ± 0.42	80.22 ± 0.72	0.19 ± 0.07	36.28 ± 0.32	81.82 ± 0.61	0.13 ± 0.10
CoDancers [39]	26.34 ± 0.29	74.22 ± 1.05	$\mathbf{0.08 \pm 0.04}$	26.44 ± 0.35	75.34 ± 0.97	$\mathbf{0.10 \pm 0.06}$
TCDiff [3]	15.36 ± 0.34	82.77 ± 0.59	0.11 ± 0.02	13.44 ± 0.53	81.70 ± 1.03	0.15 ± 0.05
ST-GDance [36]	14.76 ± 0.27	81.24 ± 0.55	0.11 ± 0.04	14.02 ± 0.46	81.42 ± 0.52	$\mathbf{0.10 \pm 0.07}$
ChainDance*	12.44 ± 0.34	82.32 ± 0.36	0.11 ± 0.02	13.28 ± 0.42	82.13 ± 0.57	0.12 ± 0.05
ChainDance	$\mathbf{11.63 \pm 0.32}$	$\mathbf{83.74 \pm 0.52}$	0.10 ± 0.03	$\mathbf{12.59 \pm 0.38}$	$\mathbf{82.16 \pm 0.66}$	0.13 ± 0.06

4.2 Quantitative Comparisons Against Baseline Models

Single Dance (1 Dancer). As shown in Table 1, our model achieves the best overall performance in single-dancer generation, reaching the lowest FID (23.11) and PFC (0.50) and the second-highest diversity (13.83). This indicates superior motion quality and stronger music alignment compared to ST-GDance [36] and TCDiff [3]. These results also confirm that the proposed multimodal encoders effectively enhance motion quality and controllability under text–music supervision, without compromising the fidelity of the pretrained single-dancer backbone.

Group Dance (2–4 Dancers). For multi-dancer choreography, ChainDance achieves the best GMR↓, indicating the highest motion realism through improved feature distribution matching. It also achieves the best GMC↑ at three and four dancers, demonstrating well-balanced group dance quality. Specifically, for three dancers, our model reduces GMR to **11.63** and achieves the highest GMC (**83.74**), indicating superior coordination and spatial harmony. Even as the number of dancers increases to four, ChainDance maintains more coherent and rhythmic stability, outperforming all baselines with lower GMR and higher GMC values. Moreover, the variant without the global optimization penalty still outperforms other approaches on most metrics, demonstrating that the core chain-of-dancers design and pairwise training strategy are sufficient to ensure strong inter-dancer coordination. These results validate the model's ability to generalize to variable group sizes, supporting efficient and coherent choreography generation from single to multi-person scenarios. To mitigate data confounds, Appendix D.6 reports ChainDance pretrained solely on AIOZ-GDance, which remains competitive (Table 10). Visualizations for larger groups are available on the supplementary webpage.

Computational Efficiency. In Table 2, our model remains compact and trainable, evidenced by the second lowest FLOPs (16.92G), the fewest parameters (15.27M), and the fastest training time per epoch (0:14). Without the optional test-time refinement, inference reduces from 0:15 to 0:06, comparable to the fastest baselines while preserving generation quality. This overhead is adjustable (fewer optimization steps trade a small amount of accuracy for speed; see appendix) and requires no retraining. In summary, ChainDance has a small model size, fast training, and produces high-quality group dances with controllable inference. A breakdown of sampling and optimization time across group sizes is provided in Appendix 15.

Table 2: **Efficiency comparison (3 dancers group).** Best results are **bold**; second-best are underlined. * denotes models without the global noise optimization penalty.

Method	FLOPs (G)↓	Params (M)↓	Train/epoch (min:sec)↓	Inf Time (min:sec)↓
GCD [16]	27.69	62.16	1:04	0:05
Codancers [39]	58.95	59.32	0:42	**0:02**
TCDiff [3]	22.04	62.48	1:29	0:08
ST-GDance [36]	**6.75**	50.21	0:37	0:03
ChainDance*	16.92	**15.27**	**0:14**	0:06
ChainDance	16.92	**15.27**	**0:14**	0:15

Controllability and User Study. We report quantitative controllability evaluation across 10 music genres in Appendix D.9 and a 30-participant user study in Appendix D.5.

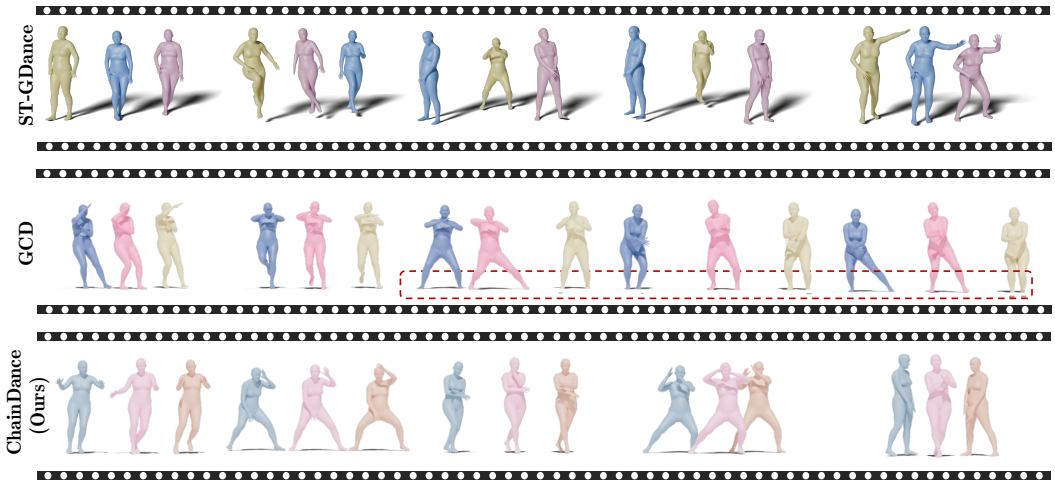


Figure 5: **Qualitative results of generated 3-dancers group sequence compared with baselines.**

4.3 Qualitative Results

Figure 5 compares dance sequences generated by our method, GCD, and ST-GDance under the same clip length and frame stride. ST-GDance exhibits frequent *apparent* "position swaps" between dancers, which do not reflect true choreographic exchanges but arise from the *identity-shift* issue, where global motion remains plausible while per-dancer temporal continuity is lost. Meanwhile, GCD shows interpenetration artifacts (red boxes), where the rightmost dancer's foot penetrates the ground plane, indicating weak physical plausibility. In contrast, ChainDance maintains stable identities and consistent spatial relations, producing collision-free trajectories and realistic ground contact throughout the sequence. This directly supports the identity preservation claim: while joint modeling architecturally permits dancer-feature swaps, ChainDance's sequential decomposition structurally precludes them, since each dancer is an independent conditional sample with no shared feature dimensions. More qualitative videos are on the supplementary webpage.

4.4 Ablation Study

Ablation Settings. To better understand the contribution of each component in our framework, we conduct ablation studies on the group dance generation task using the Lodge backbone pretrained on FineDance. We progressively remove or disable key modules to isolate their impact.

w/o GAME: The follower dancers are generated independently without conditioning on the motion of previously generated dancers. This removes cross-person interaction modeling.

w/o RATE: The system operates without semantic dance-style guidance, relying solely on music and motion context.

w/o Global Penalty: We remove the global motion-level consistency penalty at inference, which is designed to prevent inter-person collisions and encourage coherent global formation.

The sequence length is set to 120 frames. All models are trained and evaluated following the same protocol, and group-level metrics are computed on group dance sequences from AIOZ-GDance.

Table 3: **Ablation study for group dance generation (3 dancers).** Mean $\pm$ standard deviation over 20 evaluation runs. Bold indicates the best mean.

Method	GMR↓	GMC↑	TIF↓
w/o GAME	17.46 ± 0.23	79.14 ± 0.47	0.19 ± 0.04
w/o RATE	14.32 ± 0.42	81.98 ± 0.58	0.11 ± 0.03
w/o Global Penalty	12.44 ± 0.34	82.32 ± 0.36	0.11 ± 0.02
ChainDance	$\mathbf{11.63 \pm 0.32}$	$\mathbf{83.74 \pm 0.52}$	$\mathbf{0.10 \pm 0.03}$

Results. As summarized in Table 3, removing the group-aware motion encoder (w/o GAME) leads to the most pronounced degradation (GMR 17.46, GMC 79.14), highlighting the importance of modeling motion dependencies among dancers. Disabling the role-aware text prompt (w/o RATE) substantially degrades both GMR (from 11.63 to 14.32) and GMC (from 83.74 to 81.98), suggesting textual guidance improves both stylistic synchronization and spatial coordination; beyond raw metrics, RATE also provides a human-interpretable interface for per-dancer control (Appendix D.9). Removing the global penalty (*w/o Global Penalty*) leads to moderate degradation in GMR and GMC, with a small change in TIF,

Table 4: **Comparison across backbones and pretraining datasets.** Best results are **bold**, and second-best are underlined.

Method	w/o Global Optimization						with Global Optimization					
	Single-dance			Group-dance			Single-dance			Group-dance		
	FID↓	Div↑	PFC↓	GMR↓	GMC↑	TIF↓	FID↓	Div↑	PFC↓	GMR↓	GMC↑	TIF↓
EDGE (AIST++)	24.74	12.98	1.04	15.23	80.21	0.13	24.74	12.98	1.04	12.32	82.49	0.11
EDGE (FineDance)	26.12	13.34	0.62	14.93	80.73	0.12	26.12	13.34	0.62	12.25	82.65	**0.10**
Lodge (AIST++)	27.31	14.41	1.01	14.63	80.99	0.12	27.31	14.41	1.01	11.69	83.30	0.11
Lodge (FineDance)	**23.12**	**13.85**	**0.50**	**12.44**	**82.32**	**0.11**	**23.12**	**13.85**	**0.50**	**11.63**	**83.74**	**0.10**

confirming its role in inter-person coherence. The full model achieves the best mean GMR (11.63), GMC (83.74), and TIF (0.10), demonstrating complementary component effects.

A further ablation shows that conditioning the follower on the leader's motion alone cannot replace direct music conditioning (Appendix D.9.1).

4.5 Different Backbones and Pretraining Datasets

Table 4 summarizes results across different backbones and pretraining datasets. The global noise optimization defines penalty rules over inter-dancer interactions and therefore does not affect single-dance generation. Overall, adding this global penalty improves group dance (3 dancers) quality in most settings, particularly for GMR and GMC. For example, with the Lodge backbone on FineDance, GMR decreases from 12.44 to 11.63 and GMC increases from 82.32 to 83.74, showing better spatial coordination and rhythm alignment. The EDGE backbone is more efficient but produces slightly lower quality than Lodge. Models trained on FineDance outperform those trained on AIST++, likely due to FineDance's broader style coverage and richer motion variety. These results show that ChainDance successfully adapts single-dancer models for stable multi-person generation across different architectures and datasets.

5 Conclusion

We presented **ChainDance**, a parameter-efficient framework that scales group dance generation by reformulating the task as a *Chain-of-Dancers*. By adapting a frozen backbone with two lightweight modules, a **Role-Aware Text Encoder** and a **Group-Aware Motion Encoder**, our model synthesizes variable group sizes through pair-wise conditioning and inference-time noise optimization. On AIOZ-GDance, ChainDance achieves state-of-the-art motion quality while structurally preserving each dancer's identity, using 3-4$\times$ fewer parameters and 3-6$\times$ less training time. More broadly, our results suggest that structured decomposition is a viable alternative to joint modeling for multi-agent motion generation.

Acknowledgments and Disclosure of Funding

Funding. This research was supported by the Australian Government through the Australian Research Council Discovery Early Career Researcher Award (DE250100030) and Discovery Project (DP260100218). Qiuhong Ke is the recipient of the Discovery Early Career Researcher Award (DE250100030).

Competing interests. The authors declare no competing interests.

References

[1] Omid Alemi, Jules Françoise, and Philippe Pasquier. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. In *Proceedings of the 4th International Workshop on Machine Learning for Creativity (ML4Creativity), ACM SIGKDD*, Halifax, Nova Scotia, Canada, 2017.

[2] Lele Chen, Guofeng Cui, Celong Liu, Zhong Li, Ziyi Kou, Yi Xu, and Chenliang Xu. Talking-head generation with rhythmic head motion. In *European conference on computer vision*, pages 35–51. Springer, 2020.

[3] Yuqin Dai, Wanlu Zhu, Ronghui Li, Zeping Ren, Xiangzheng Zhou, Xiu Li, Jun Li, and Jian Yang. Harmonious group choreography with trajectory-controllable diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[4] Sihao Ding, Fuli Feng, Xiangnan He, Yong Liao, Jun Shi, and Yongdong Zhang. Causal incremental graph convolution for recommender system retraining. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4718–4728, 2022.

[5] Joao P Ferreira, Thiago M Coutinho, Thiago L Gomes, José F Neto, Rafael Azevedo, Renato Martins, and Erickson R Nascimento. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computers & Graphics*, 94:11–21, 2021.

[6] Anindita Ghosh, Bing Zhou, Rishabh Dabral, Jian Wang, Vladislav Golyanik, Christian Theobalt, Philipp Slusallek, and Chuan Guo. Duetgen: Music driven two-person dance generation via hierarchical masked modeling. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–11, 2025.

[7] Kehong Gong, Dongze Lian, Heng Chang, Chuan Guo, Zihang Jiang, Xinxin Zuo, Michael Bi Mi, and Xinchao Wang. TM2D: Bimodality driven 3d dance generation via music-text integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9942–9952, 2023.

[8] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. AvatarCLIP: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)*, 2022.

[9] Dongjin Huang, Yue Zhang, Zhenyan Li, and Jinhua Liu. TG-Dance: Transgan-based intelligent dance generation with music. In *International Conference on Multimedia Modeling*, pages 243–254. Springer, 2023.

[10] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. In *International conference on learning representations*, 2020.

[11] Yin-Fu Huang and Wei-De Liu. Choreography cgan: generating dances with music beats using conditional generative adversarial networks. *Neural Computing and Applications*, 33(16): 9817–9833, 2021.

[12] Manish Joshi and Sangeeta Chakrabarty. An extensive review of computational dance automation techniques and applications. *Proceedings of the Royal Society A*, 477(2251):20210071, 2021.

[13] Tae-hoon Kim, Sang Il Park, and Sung Yong Shin. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics (TOG)*, 22(3):392–401, 2003.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *Proceedings of SIGGRAPH*, 2002.

[16] Nhat Le, Tuong Do, Khoa Do, Hien Nguyen, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Controllable group choreography using contrastive diffusion. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023.

[17] Nhat Le, Thang Pham, Tuong Do, Erman Tjiputra, Quang D Tran, and Anh Nguyen. Music-driven group choreography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8673–8682, 2023.

[18] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[19] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1272–1279, 2022.

[20] Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. Finedance: A fine-grained choreography dataset for 3d full body dance generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10234–10243, 2023.

[21] Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1524–1534, 2024.

[22] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13401–13412, 2021.

[23] Weizhi Nie, Rihao Chang, Minjie Ren, Yuting Su, and Anan Liu. I-GCN: Incremental graph convolution network for conversation emotion detection. *IEEE Transactions on Multimedia*, 24: 4471–4481, 2021.

[24] Sakuya Ota, Qing Yu, Kent Fujiwara, Satoshi Ikehata, and Ikuro Sato. Pino: Person-interaction noise optimization for long-duration and customizable motion generation of arbitrary-sized groups. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10676–10685, 2025.

[25] Qiaosong Qi, Le Zhuo, Aixi Zhang, Yue Liao, Fei Fang, Si Liu, and Shuicheng Yan. Diffdance: Cascaded human motion diffusion model for dance generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1374–1382, 2023.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[27] Alla Safonova and Jessica K Hodgins. Construction and optimal search of interpolated motion graphs. In *Proceedings of ACM SIGGRAPH*, pages 106–es, 2007.

[28] Judith L Schwartz. The passacaille in Lully's "armide": Phrase structure in the choreography and the music. *Early Music*, 26(2):301–320, 1998.

[29] Li Siyao, Tianpei Gu, Zhitao Yang, Zhengyu Lin, Ziwei Liu, Henghui Ding, Lei Yang, and Chen Change Loy. Duolando: Follower gpt with off-policy reinforcement learning for dance accompaniment. In *The Twelfth International Conference on Learning Representations*.

[30] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050–11059, 2022.

[31] Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando++: 3d dance gpt with choreographic memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14192–14207, 2023. doi: 10.1109/TPAMI.2023.3319435.

[32] Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S Kankanhalli, Weidong Geng, and Xiangdong Li. Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia*, 23:497–509, 2020.

[33] Julian Tanke, Linguang Zhang, Amy Zhao, Chengcheng Tang, Yujun Cai, Lezi Wang, Po-Chen Wu, Juergen Gall, and Cem Keskin. Social diffusion: Long-term multiple human motion anticipation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9567–9577. IEEE, 2023.

[34] Jonathan Tseng, Rodrigo Castellon, and Karen Liu. Edge: Editable dance generation from music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 448–458, 2023.

[35] Qing Wang, Xiaohang Yang, Yilan Dong, Naveen Raj Govindaraj, Gregory Slabaugh, and Shanxin Yuan. Dancechat: Large language model-guided music-to-dance generation. *arXiv preprint arXiv:2506.10574*, 2025.

[36] Jing Xu, Weiqiang Wang, Cunjian Chen, Jun Liu, and Qiuhong Ke. St-gdance: Long-term and collision-free group choreography from music. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2025.

[37] Nelson Yalta, Shinji Watanabe, Kazuhiro Nakadai, and Tetsuya Ogata. Weakly-supervised deep recurrent neural networks for basic dance step generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[38] Han Yang, Kun Su, Yutong Zhang, Jiaben Chen, Kaizhi Qian, Gaowen Liu, and Chuang Gan. Unimumo: Unified text, music, and motion generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25615–25623, 2025.

[39] Kaixing Yang, Xulong Tang, Ran Diao, Hongyan Liu, Jun He, and Zhaoxin Fan. Codancers: Music-driven coherent group dance generation with choreographic unit. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pages 675–683, 2024.

[40] Siyue Yao, Mingjie Sun, Bingliang Li, Fengyu Yang, Junle Wang, and Ruimao Zhang. Dance with you: The diversity controllable dancer generation via diffusion models. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8504–8514, 2023.

[41] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8476, 2024.

[42] Wenjie Yin, Hang Yin, Kim Baraka, Danica Kragic, and Mårten Björkman. Dance style transfer with cross-modal transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5058–5067, 2023.

[43] Ling-An Zeng, Guohong Huang, Gaojie Wu, and Wei-Shi Zheng. Light-t2m: A lightweight and fast model for text-to-motion generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9797–9805, 2025.

[44] Zixiang Zhou, Yu Wan, and Baoyuan Wang. AvatarGPT: All-in-one framework for motion understanding planning generation and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2024.

[45] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2):1–21, 2022.

A Prior Assumptions: Joint Distribution Perspective on Group Dance Generation

From a probabilistic modeling perspective, generative models fundamentally learn conditional probability distributions over data. For the group dance generation task considered in this work, let $\mathcal{X}_{1:N} = \{X_1, X_2, \ldots, X_N\}$ denote a set of dance motion sequences performed by N dancers, where each X_i represents the full temporal motion trajectory of the i-th dancer. The learning objective of group dance generation can thus be formulated as modeling the joint distribution $P(\mathcal{X}_{1:N})$.

According to the chain rule of probability, the joint distribution can be exactly factorized as:

$$P(\mathcal{X}_{1:N}) = P(X_1) \prod_{i=2}^{N} P(X_i \mid \mathcal{X}_{1:i-1}), \tag{7}$$

which is a mathematically equivalent decomposition and does not introduce any approximation.

Our approach is directly motivated by this factorization. We first leverage a well-trained and stable single-dancer diffusion model to capture the marginal distribution $P(X_1)$, providing strong temporal dynamics and motion priors. For each subsequent dancer X_i, we model the conditional distribution $P(X_i \mid \mathcal{X}_{1:i-1})$ via conditional generation, where the conditioning signals consist of the motions of previously generated dancers together with multimodal control inputs.

This formulation offers two key advantages. First, temporal continuity, rhythm, and motion realism—already well captured by the single-dance generator—do not need to be relearned in the group setting. Second, inter-dancer interactions and spatial constraints are explicitly introduced through conditional dependencies during generation, avoiding the need to directly model the full high-dimensional joint distribution in a single step.

Therefore, from the perspective of joint probability modeling, the proposed sequential conditional generation strategy preserves the expressiveness of the target group dance distribution while providing a structured and scalable approach that improves training stability and generation robustness.

B Implementation Details

This section provides the detailed formulations of graph construction, all loss terms and encoder details related to ChainDance. Each loss term corresponds to either the local reconstruction stage or the global refinement stage described in Section 3 of the main paper.

B.1 Group-Aware Motion Graph

The Group-Aware Motion Encoder (Section 3) is implemented as a graph convolutional network operating on a *distance-weighted dynamic graph* that grows incrementally as new dancers are generated.

Distance-weighted edges. For the i-th dancer (node) being generated, we define raw interaction weights w_{ij} with all previously generated dancers j based on their Euclidean distance:

$$w_{ij} = \frac{1}{|p_i - p_j| + \epsilon}, \tag{8}$$

where p_i and p_j denote the global root positions of dancer i and j, respectively, and $\epsilon = 10^{-6}$ is a small constant added to avoid division by zero. This formulation encodes a natural inductive bias: *closer dancers exert a stronger influence on the generation of the current dancer*, mirroring how real choreography prioritizes nearby partners over distant ones.

Graph construction. Following the chain-of-dancers formulation, the graph is built incrementally rather than over a fully connected static topology. When a new dancer is added to the chain, edges are dynamically constructed only between the new node and the previously generated dancers, while existing edges are preserved. This design has been shown effective in recent graph-based generation studies, including I-GCN [23] and IGC [4], and naturally fits our sequential generation scheme. During diffusion sampling, the follower's root position is taken from the intermediate prediction $\hat{X}_i^{(t)}$ at each timestep t; the graph is recomputed dynamically as denoising progresses.

Top-k edge sparsification. To focus on local coordination and improve computational efficiency, we sparsify the graph by retaining only the top 60% nearest edges (i.e., highest w_{ij} values) for each newly added dancer. This avoids over-smoothing from far-away dancers whose contributions to local coordination are minimal, while preserving sufficient connectivity to model meaningful inter-dancer relationships. The sparsified weights are then row-normalized via softmax to obtain the final adjacency matrix used in the GCN layer:

$$\tilde{w}_{ij} = \frac{\exp(w_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(w_{ik})}, \tag{9}$$

where $\mathcal{N}_i$ denotes the set of retained neighbors of dancer i after top-k sparsification.

B.2 Algorithm Details

Due to space limitations in the main paper, we provide a detailed description of the Group-Aware Motion Encoder in Algorithm 1. This section explicitly specifies the input and output representations, intermediate feature dimensionalities, and the sequence of modules used to generate inter-dancer motion-aware key/value pairs. The goal is to clarify how motion interactions are encoded and injected into the frozen single-dancer diffusion backbone, complementing the high-level description in the main text.

Algorithm 1 Group-Aware Motion Encoder

Input: Immediate leader motion $\mathbf{X}_j$ (where $j = i-1$), all previously generated motions $\{\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}\}$
Output: Inter-dancer key/value $(\mathbf{K}_{grp}, \mathbf{V}_{grp})$
$\mathcal{G} \leftarrow \text{BuildGraph}(\{\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}\})$ // Distance-weighted graph over all previously generated dancers
$\mathbf{Z}_j \leftarrow f_{\text{motion}}(\mathbf{X}_j)$ // Leader Motion Encoding, $\mathbf{X}_j \in \mathbb{R}^{B\times T\times D_m}, \mathbf{Z}_j \in \mathbb{R}^{B\times T\times d_z}$
$\tilde{\mathbf{Z}}_j \leftarrow \text{GCN}(\mathbf{Z}_j, \mathcal{G})$ // Aggregate group-wide spatial context onto leader feature, temporal length L preserved
$\mathbf{H}_j \leftarrow \text{Transformer}_{\text{LoRA}}(\tilde{\mathbf{Z}}_j)$ // LoRA-Adapted Transformer Encoding, which applies low-rank updates to linear projections
$\mathbf{K}_{grp} \leftarrow \mathbf{H}_j\mathbf{W}_K$ // Key / Value Projection, $\mathbf{K}_{grp}, \mathbf{V}_{grp} \in \mathbb{R}^{B\times H\times L\times d_h}$
$\mathbf{V}_{grp} \leftarrow \mathbf{H}_j\mathbf{W}_V$
Inject $(\mathbf{K}_{grp}, \mathbf{V}_{grp})$ into frozen backbone // Cross-Attention Injection

B.3 Training Loss

The detailed definitions of the loss terms used in Equation 3 in Section 3.2 are given as follows.

Pose Reconstruction Loss. To ensure accurate recovery of dancer poses, we use an ℓ_1 loss between predicted $\hat{\mathbf{X}}_l^{(n)}$ and ground-truth $\mathbf{X}_l^{(n)}$ joint representations:

$$\mathcal{L}_{\text{pose}} = \frac{1}{NL}\sum_{n=1}^{N}\sum_{l=1}^{L}\left\|\hat{\mathbf{X}}_l^{(n)} - \mathbf{X}_l^{(n)}\right\|_1. \tag{10}$$

This term drives the model to reproduce spatially precise skeleton structures at each frame.

Velocity Regularization Loss. To suppress temporal jitter and enforce smooth motion transitions, we apply a velocity consistency constraint:

$$\mathcal{L}_{\text{vel}} = \frac{1}{NL}\sum_{n,l}\left\|(\hat{\mathbf{X}}_l^{(n)} - \hat{\mathbf{X}}_{l-1}^{(n)}) - (\mathbf{X}_l^{(n)} - \mathbf{X}_{l-1}^{(n)})\right\|_1. \tag{11}$$

It encourages the predicted motion derivatives to follow the ground-truth temporal dynamics.

Foot-Ground Contact Loss. To reduce foot sliding and floating artifacts, we constrain the contact joints (e.g., ankles and toes) relative to the ground plane:

$$\mathcal{L}_{\text{contact}} = \frac{1}{L}\sum_{l}\sum_{k\in\Omega_{\text{foot}}}\left(|\hat{z}_{l,k}| + \alpha\,|\hat{v}_{l,k}|\right), \tag{12}$$

where $\hat{z}_{l,k}$ and $\hat{v}_{l,k}$ denote the height and velocity of foot joint k, and α balances positional and velocity constraints.

Multimodal Alignment Loss. We follow the text-pivoted alignment formulation proposed in DanceChat [35], which encourages consistency between modalities by aligning both text–music and text–motion embeddings via cosine similarity. At each timestep i, the alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = \frac{1}{N}\sum_{i=1}^{N}\Big[(1-\text{sim}(E_{t_i}, E_{m_i})) + (1-\text{sim}(E_{t_i}, E_{x_i}))\Big], \tag{13}$$

Relation (REL)	Decision rule in window $\mathcal{W}$	Artifacts (for prompt)
`approach`	Monotonic decrease in distance: $\Delta d/W < -0.1$ and $d_{ij}(w_0 + W - 1) \in [d_{\text{near}},\, d_{\text{mid}}]$.	WHEN, target distance $d^\star \approx d_{\text{near}}$
`keep_distance`	Low variance around mid distance: $\mathrm{Var}_{w\in\mathcal{W}}(d_{ij}(w)) < 0.05$ and $\mathrm{mean}(d_{ij}) \in [d_{\text{mid}} - 0.2,\, d_{\text{mid}} + 0.2]$.	WHEN, target distance $d^\star \approx d_{\text{mid}}$
`face_to`	Mutual facing with proximity: $\cos\big(\mathbf{f}_i, \mathbf{p}_j - \mathbf{p}_i\big) > 0.7$ and $\cos\big(\mathbf{f}_j, \mathbf{p}_i - \mathbf{p}_j\big) > 0.7$ and $d_{ij} < d_{\text{far}}$.	WHEN
`lead_follow`	Cross-correlation of motion energies peaks at a positive lag: $\arg\max_{\Delta w\in[2,8]} \mathrm{xcorr}(E_i, E_j) = \Delta w > 0$. (Here i leads, j follows.)	WHEN, lag Δw; leader/follower IDs
`swap`	Start/end positions swap across the window: $\|\mathbf{p}_i(w_0) - \mathbf{p}_j(w_0+W-1)\|_2 < \epsilon$ and $\|\mathbf{p}_j(w_0) - \mathbf{p}_i(w_0 + W - 1)\|_2 < \epsilon$, while trajectories do not collide.	WHEN
`passby`	Trajectories intersect near mid/near distance with opposite normal velocities: $\exists\, w \in \mathcal{W}$ s.t. $d_{ij}(w) \in [d_{\text{near}},\, d_{\text{mid}}]$, $\langle \dot{\mathbf{p}}_i^{\perp}, \dot{\mathbf{p}}_j^{\perp} \rangle < 0$, and distance increases rapidly after intersection.	WHEN

Table 5: Window-based relation detection rules for prompt mining. *WHEN* records the time span(s) in $\mathcal{W}$ where the rule is satisfied; artifacts (e.g., $d^\star$, contacted parts, lag Δt) are injected into prompt templates.

where $\mathrm{sim}(\mathbf{a}, \mathbf{b}) = \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|\,\|\mathbf{b}\|}$ denotes cosine similarity and E_{t_i}, E_{m_i}, E_{x_i} represent the encoded embeddings of text, music, and motion, respectively, at timestep i. While we adopt the same loss structure as DanceChat, we re-implement the encoders E_m and E_x using our own music and motion backbones to ensure compatibility with the encoder-based architecture.

B.4 Inference Global Penalties

We used GPU parallel acceleration in different sample batch dimensions to minimize latency, where multiple noise samples are optimized in parallel within each iteration of gradient descent. In Section 3.3, we introduce a global penalty at inference time. The loss terms of L_{global} in Equation 5 are listed as follows:

Root Position Penalty. To ensure that an individual reaches a specific location at a particular time, we define the root position penalty L_{root} as follows:

$$L_{\text{root}} = \sum_{l \in L} \max\left(0, \|p^p_{\text{root}}(l) - p_{\text{target}}(l)\|^2 - \delta\right), \tag{14}$$

where $p^p_{\text{root}}(l)$ represents the root position of individual p at frame l, and $p_{\text{target}}(l)$ is the target position at the same frame. The hyperparameter δ acts as a threshold distance, below which no penalty is applied. This formulation relaxes the loss function by penalizing deviations from the target position only when the distance exceeds δ. Such a design avoids overly rigid constraints on close positions and encourages more natural and flexible motion synthesis, consistent with prior work PINO [24].

Movement Region Penalty. To restrict movement within a defined area, we introduce the movement region penalty L_{region} for a chosen individual p:

$$L_{\text{region}} = \frac{1}{LJ} \sum_{n} \sum_{j} \phi\left(p^p_j(n)\right), \tag{15}$$

where L is the total number of frames, J is the total number of joints, and $\phi(\cdot)$ penalizes positions outside the desired region. Here, $p^p_j(n)$ denotes the position of joint j of individual p at frame n. The penalty function $\phi(\cdot)$ can be flexibly defined according to task requirements. For example, restricting movement within a rectangular cuboid region with lower bounds $l = (l_x, l_y, l_z)$ and upper bounds

$u = (u_x, u_y, u_z)$ yields:

$$\phi(p) = \sum_{k \in \{x,y,z\}} \max(0,\, l_k - p_k) + \max(0,\, p_k - u_k), \tag{16}$$

where p_k is the k-th coordinate of p.

Orientation Penalty. To control facing directions at specific frames, we define the orientation penalty L_{orient} with threshold δ as:

$$L_{\text{orient}} = \sum_{l \in L} \max\left(0,\, 1 - d^p(l) \cdot d_{\text{target}}(l) - \delta\right), \tag{17}$$

where $d^p(l)$ is the normalized facing direction of individual p at frame l, and $d_{\text{target}}(l)$ is the desired direction. The threshold δ determines the tolerance margin within which no penalty is applied. By penalizing deviations only when direction similarity falls below $1 - \delta$, this formulation promotes natural orientation control and avoids abrupt directional changes, similar to the soft-direction constraints adopted in PINO [24].

Relative Position Penalty. To ensure that the root positions of two individuals remain within a desired distance range, we define the relative position penalty L_{relative} as:

$$\begin{aligned} L_{\text{relative}} = \sum_{l \in L} \Big[& \max\left(0,\, d_{\min} - \|p^1_{\text{root}}(l) - p^2_{\text{root}}(l)\|\right) \\ & + \max\left(0,\, \|p^1_{\text{root}}(l) - p^2_{\text{root}}(l)\| - d_{\max}\right) \Big], \end{aligned} \tag{18}$$

where $p^1_{\text{root}}(l)$ and $p^2_{\text{root}}(l)$ denote the root positions of two individuals at frame l, and $d_{\min}$ and $d_{\max}$ represent the lower and upper bounds of the allowable interpersonal distance. This constraint maintains natural interaction spacing and prevents unrealistic crowding or separation, an important requirement also emphasized by PINO [24].

To accelerate the optimization of global penalty over initial noise at the inference stage, in the main experiment, we did not use the penalty of the entire long-time sequence as the optimization target, but instead selected the 70 frames with the largest penalty values in the entire sequence as the optimization target. Meanwhile, we used GPU parallel acceleration in different sample batch dimensions to minimize latency in the inference stage.

C From Rules to Interaction Types

As described in Section 3.1 and illustrated in Figure 3, the mined pairwise relations from each window $\mathcal{W}$ are verbalized into dancer-specific textual instructions that serve as the conditioning input for ChainDance. We use a sliding temporal window $\mathcal{W}$ to define these relations: for each window, we look at the locations of two dancers within this short time segment and decide whether an interaction occurs and, if so, what type it is. This design uses windows instead of the whole sequence because interactions do not happen at every frame and are usually completed within a short time range. We adopt minimal, compositional templates that consume the artifacts produced by the decision rules (e.g., estimated distance $d^\star$, lag Δt, and time span *WHEN*), while strictly following the prompt format constraints in Section 3.1. The detailed computation of these rules is summarized in Table 5.

Templates. Detected relations are mapped to short phrases as follows:

- `approach`: "*Dancer-i approaches Dancer-j to about $d^\star$.*"
- `keep_distance`: "*i and j keep a steady distance $\approx d^\star$.*"
- `face_to`: "*i and j face each other.*"
- `lead_follow`: "*i leads j (lag Δw).*"
- `swap`: "*i and j swap positions at.*"
- `passby`: "*i and j pass by.*"

These phrases are then injected into per-dancer prompts (e.g., "*The dancer i ..., facing toward Dancer j.*") on top of the global description, yielding individualized conditioning as specified in Section 3.1.

De-duplication and Priority. If multiple relations fire in the same window, we keep the one with the highest confidence (e.g., larger margin to threshold) and merge contiguous *WHEN* intervals if gaps are < 0.5 s. The final per-dancer prompt is constructed by (i) preserving the global motif and (ii) appending at most one relation phrase per active pair within $\mathcal{W}$.

D More Quantitative Results

D.1 Sensitivity to Heuristic Settings

We evaluate the sensitivity of the rule-based interaction mining and GAME graph construction to their heuristic settings. For each row in Table 6, we change the listed parameter by -20% or $+20\%$ while keeping the other settings fixed; the distance thresholds d_{near} and d_{mid} are varied together. The maximum changes relative to the default setting are 0.11 in GMR and 0.25 in GMC. These results indicate limited sensitivity within the tested range, although they do not assess interaction types that the rule-based labels may miss.

Table 6: **Sensitivity to heuristic settings for three-dancer generation.** Results for each perturbation are shown as -20% / $+20\%$; all other settings remain fixed.

Perturbed setting	GMR↓ ($-\Delta$ / $+\Delta$)	GMC↑ ($-\Delta$ / $+\Delta$)	Max. $\lvert\Delta\text{GMR}\rvert$
Default	11.63	83.74	–
d_{near}, d_{mid}	11.71 / 11.68	83.55 / 83.60	0.08
Window length W	11.66 / 11.70	83.68 / 83.51	0.07
Graph top-k ratio	11.74 / 11.65	83.49 / 83.71	0.11
Gaze cosine threshold	11.69 / 11.67	83.60 / 83.66	0.06

D.2 Comparison of 5 Dancers

Table 7 summarizes the results across the 1–5 dancer settings. We report 5-dancer results here rather than in the main paper because the number of 5-dancer samples in the AIOZ-GDance test set is limited, making the comparison less statistically robust. At $N{=}5$, ChainDance achieves the best GMC (**81.56**) and the second-best GMR (14.73 vs. TCDiff's 14.62), with the small GMR gap reflecting a fundamental trade-off of our design: while joint-modeling baselines can be specifically optimized for each fixed group size, ChainDance's unified pair-wise training transfers across all sizes from a single model. Across the full 1–5 dancer range, ChainDance remains consistently competitive with one unified model, an advantage no baseline can match.

D.3 Comparison on DD100 dataset

Although AIOZ-GDance is the only publicly available large-scale group dance benchmark, several duet datasets exist. We therefore evaluate ChainDance's generalization on DD100 [29], a duet dance dataset with 117 minutes of professional motion-capture performances across 10 genres. Following our main setup, we adopt the Lodge [21] backbone pretrained on FineDance [20] and post-train the group-aware modules on DD100. Since DD100 does not provide text annotations, we disable the Role-Aware Text Encoder for this experiment and condition only on music, while retaining the Group-Aware Motion Encoder and chain-of-dancers formulation. We follow the official train/test split and compare with Duolando [29] and GCD [16] under the same evaluation protocol. Results are reported in Table 8.

As shown in Table 8, ChainDance achieves the best performance across all metrics on DD100, demonstrating that our framework generalizes effectively beyond AIOZ-GDance. In particular, ChainDance reduces FID from 9.73 (GCD) to **2.97**, achieves the closest Div to GT (15.52 vs. GT's 15.67), and outperforms both Duolando and GCD on group-level metrics. These results confirm that even with text conditioning disabled, the chain-of-dancers formulation, combined with the Group-Aware Motion Encoder, generalizes effectively to the duet setting.

Table 7: **Unified comparison across 1–5 dancers on the AIOZ-GDance dataset.** For all baselines, *separate models* are trained and evaluated for each group size. In contrast, **ChainDance uses a single unified model for all group sizes (1–5 dancers)**. Best results are **bold**; second-best are underlined. Results with * denote models without the global noise optimization penalty. Results in this table are from a single evaluation run.

Method	1 dancer			2 dancers			3 dancers			4 dancers			5 dancers		
	FID↓	Div↑	PFC↓	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓
GCD [16]	39.24	9.64	2.53	34.39	80.32	0.17	30.22	80.22	0.19	36.34	81.54	0.14	38.43	81.44	0.17
CoDancers [39]	23.98	9.48	3.53	24.55	72.52	**0.08**	26.34	74.22	**0.08**	26.53	75.42	**0.10**	27.27	74.34	**0.11**
TCDiff [3]	37.31	**14.01**	0.51	15.77	**81.92**	0.12	15.36	82.77	0.11	13.44	81.70	0.15	**14.62**	81.40	**0.11**
ST-GDance [36]	28.87	12.82	0.97	19.42	80.52	0.12	14.76	81.24	0.11	14.02	81.42	**0.10**	23.22	80.76	0.12
ChainDance*	**23.12**	13.85	**0.50**	14.10	81.52	0.11	12.01	82.68	0.11	13.28	82.13	0.12	15.34	80.93	0.13
ChainDance	**23.12**	13.85	**0.50**	**13.67**	81.28	0.12	**11.63**	**83.74**	0.10	**12.59**	**82.16**	0.13	14.73	**81.56**	0.12

Table 8: **Two-dancer generation results on the DD100 dataset.** Best results are **bold**. "→" indicates that closer to ground-truth (GT) is better.

Method	FID↓	Div→	PFC↓	GMR↓	GMC↑	TIF↓
GT	–	15.67	0.36	–	–	–
Duolando [29]	12.42	14.35	16.22	14.52	81.32	0.15
GCD [16]	9.73	14.62	5.12	14.31	81.21	0.13
ChainDance (w/o RATE)	**2.97**	**15.52**	**4.41**	**12.21**	**82.42**	**0.12**

D.4 Comparison with PINO

A potential concern is whether the gains of ChainDance over end-to-end baselines simply come from the test-time noise optimization adopted from PINO [24]. To address this, we directly compare ChainDance with PINO using the same single-dancer backbone for group sizes 2–4, isolating the contribution of our proposed encoders and chain-of-dancers formulation.

As shown in Table 9, ChainDance consistently outperforms PINO across all group sizes. Notably, the performance gap *widens* as the group size increases (e.g., GMR improves by 0.43 for 2 dancers but by 1.24 for 4 dancers), indicating that relying solely on global physical constraints becomes increasingly insufficient as inter-dancer interactions grow more complex. While PINO refines spatial consistency at inference time, it lacks the capacity to model fine-grained inter-dancer relationships during generation. Our encoders address this gap by injecting relational information directly into the latent space, providing the structural foundation that PINO's optimization alone cannot establish. This isolation experiment confirms that ChainDance's contribution goes beyond inference-time refinement: the proposed encoders and chain-of-dancers formulation are the key drivers of the observed performance gains.

D.5 User Study

Beyond automatic metrics, we conduct a user study to evaluate the perceptual quality of generated group dances. Following the protocol of our main baseline TCDiff [3], we invite **30 participants** (ages 18–55, aligning with the target user group of our application scenario) to participate in a blind ranking test. Outputs from all methods are anonymized and shuffled before presentation. For each comparison, participants are shown four videos (one per method) and asked to rank them from best to worst along four perceptual dimensions:

Table 9: **Comparison with PINO across different group sizes.** Best results are in **bold**.

Method	2			3			4		
	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓	GMR↓	GMC↑	TIF↓
PINO [24]	14.10	80.83	0.12	12.69	82.92	0.11	13.83	80.93	0.14
ChainDance	**13.67**	**81.28**	**0.12**	**11.63**	**83.74**	**0.10**	**12.59**	**82.16**	**0.13**

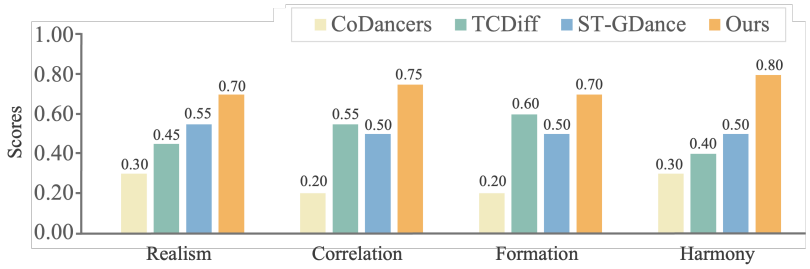


Figure 6: **User study results across four perceptual dimensions.** ChainDance is consistently preferred over baselines on Realism, Music Correlation, Formation Aesthetics, and Dancer Harmony. The largest improvement is observed on Dancer Harmony (0.80 vs. 0.50 of the best baseline), which jointly captures inter-dancer coordination and per-dancer identity stability—both structurally improved by our sequential decomposition.

- **Realism**: How natural and human-like are the generated motions?
- **Music Correlation**: How well do the dance movements align with the music's rhythm and style?
- **Formation Aesthetics**: How visually pleasing and well-coordinated are the group formations?
- **Dancer Harmony**: How harmoniously do the dancers interact and complement each other, and how consistently does each dancer maintain a stable identity throughout the sequence (i.e., absence of identity swaps or sudden position jumps)?

Rankings are converted into scores from 4 (best) to 1 (worst), and the averaged scores are further normalized to $[0, 1]$ for fair comparison. Results are reported in Figure 6.

As shown in Figure 6, ChainDance is preferred across all four dimensions, achieving the highest normalized scores on Realism (0.70), Music Correlation (0.75), Formation Aesthetics (0.70), and Dancer Harmony (0.80). The largest gap appears on *Dancer Harmony*, where ChainDance scores 0.80 compared to 0.50 for the strongest baseline on this dimension (ST-GDance). This dimension captures both inter-dancer coordination and per-dancer identity stability over time—two aspects that joint-modeling baselines struggle with simultaneously, as joint tokenization tends to produce identity swaps and flickering artifacts (Figure 1b). ChainDance's sequential decomposition structurally avoids identity entanglement, while RATE and GAME ensure coherent inter-dancer interaction. This provides direct perceptual evidence supporting the identity preservation claim, complementing the objective metrics in Table 1 and Table 9, jointly demonstrating that ChainDance produces group choreography that is both quantitatively superior *and* perceptually preferred.

D.6 Other Ablation Settings

To comprehensively evaluate ChainDance, we compare results under different configurations: single-dancer generators pretrained on AIOZ-GDance, models pretrained jointly on FineDance and AIST++, and variants trained without motion captions in text descriptions.

Table 10 summarizes several additional ablation settings. When we pretrain the single-dancer generator on the AIOZ-GDance subset and then finetune with multi-dancer clips, the performance is slightly lower than the main results in the paper. This is understandable because AIOZ-GDance is primarily a multi-person dataset, and its single-dance samples are limited, which makes the single-dancer pretraining less effective.

Table 10: **Comparison of 3 dancers performance under different ablation settings on the AIOZ-GDance dataset.** Best results are **bold**.

Method	Group Dance		
	GMR↓	GMC↑	TIF↓
ChainDance	**11.63**	**83.74**	**0.10**
Ours-Lodge(AIOZ-GDance)	12.08	82.32	0.11
Ours-Lodge (FineDance&AIST++)	11.97	82.41	0.12
ChainDance (w/o Motion Caption)	11.89	82.52	0.11
ChainDance (w/o bidirectional pairing)	12.93	79.42	0.13
ChainDance (with random pairing strategy)	11.58	83.80	0.10

Using a backbone pretrained jointly on FineDance and AIST++ achieves similar performance to the AIOZ-GDance-based model, but remains below the version pretrained only on FineDance in the main paper. A possible explanation is that the mixed-domain training increases the diversity of styles and camera motions, which benefits generality but introduces additional variance that is not fully aligned with AIOZ-GDance.

Removing motion captions from the text descriptions leads to a small drop in performance, but the results remain stable and still compare favorably with end-to-end multi-dancer baselines that do not use such information. This suggests that motion captions are helpful, but they are not the dominant factor. The improvements mainly come from the full text inputs (music, motion caption, beat structure) together with the ChainDance formulation, which simultaneously provide the model with strong guidance and allow efficient multi-dancer generation. Training pairs without bidirectional leader relationship between dancers may cause the model to learn only some fixed local interaction patterns, thus greatly reducing the effectiveness of group dance generation. Comparing the two pairing strategies, both random pairing and chain pairing yield similar group-level performance, with only marginal differences across metrics. For clarity and reproducibility, we use chain pairing as the default setting in the main experiments.

D.7 Error Accumulation Analysis Across Group Sizes

A natural concern with sequential chain generation is whether errors accumulate as group size grows. Our inference-time global optimization is designed to mitigate such error propagation. Table 11 compares generation quality with and without the global penalty across group sizes 2–5.

Table 11: **Comparison of results with / without the global penalty across group sizes.** Format: *with penalty* / *without penalty*.

N	GMR↓		GMC↑		TIF↓	
2	13.67	14.52	81.28	80.92	0.12	0.13
3	11.63	12.44	83.74	82.12	0.12	0.13
4	12.59	14.30	82.16	81.47	0.13	0.13
5	14.73	15.52	81.56	80.64	0.12	0.13

Several observations are noteworthy. First, metrics do not degrade monotonically with group size—for instance, TIF for 5 dancers (0.12) is lower than for 4 dancers (0.13), and GMC for 3 dancers (83.74) outperforms 2 dancers (81.28), suggesting the chain formulation benefits from richer conditioning context as more dancers are added. Second, the global penalty consistently improves performance across all group sizes, confirming that it effectively mitigates error propagation inherent in sequential generation rather than simply applying post-hoc corrections.

Per-position analysis for ten dancers. We further examine whether dancers generated later in the chain show progressive degradation at $N = 10$. Table 12 reports physical foot contact (PFC) and trajectory intersection frequency (TIF) at each chain position k; TIF counts collisions involving the k-th dancer. Without the global penalty, PFC rises from 0.52 to 0.63 and TIF from 0.11 to 0.15 between the first and last positions. With the penalty, the corresponding changes are smaller (0.51 to 0.54 for PFC and 0.11 to 0.13 for TIF). These ground-truth-free metrics suggest that the global

penalty reduces the measured degradation along the chain. They do not establish distributional realism or formation quality for ten-dancer groups.

Table 12: **Per-position analysis for ten-dancer generation.** TIF counts collisions involving the dancer at chain position k. Lower values are better for both metrics.

Chain position k	Without global penalty		With global penalty	
	TIF↓	PFC↓	TIF↓	PFC↓
1	0.11	0.52	0.11	0.51
2	0.11	0.54	0.11	0.52
3	0.12	0.55	0.11	0.52
4	0.12	0.56	0.12	0.52
5	0.12	0.55	0.11	0.53
6	0.13	0.56	0.12	0.53
7	0.13	0.57	0.12	0.53
8	0.13	0.60	0.12	0.53
9	0.14	0.61	0.13	0.54
10	0.15	0.63	0.13	0.54
Mean $\pm$ Std	0.13 ± 0.01	0.57 ± 0.03	0.12 ± 0.01	0.53 ± 0.01

D.8 Analysis on Different Optimization Steps and More Number of Dancers

Table 13: **In-depth analysis of the trade-off between inference time and dance quality on the AIOZ-GDance dataset (3 dancers).** Best results are **bold**; second-best are underlined.

Optimization Steps	**Group Dance**		**Efficiency**
	GMR↓	GMC↑	Inf Time (min:sec)↓
0×	12.01	<u>82.68</u>	**0:06**
0.1×	11.83	82.04	<u>0:08</u>
0.25×	11.78	81.72	0:10
0.5×	**11.52**	82.35	0:11
0.75×	11.72	82.59	0:13
ChainDance (80 steps)	<u>11.63</u>	**83.74**	0:15

Table 13 compares generation quality and inference efficiency under different penalty threshold settings. The ChainDance row shows the configuration that achieves the best performance (consistent with the main experiments in the paper). We then examine the trade-offs between generation quality and efficiency across different multiplication factors.

Table 13 shows that fewer optimization steps reduce computation but also weaken the global refinement, leading to lower GMC. As the multiplier increases, the global penalty constraints take effect more strongly, and GMC improve, but at the cost of longer inference time. The ChainDance configuration achieves the best overall balance, giving the highest GMC and the second-lowest GMR while keeping the inference cost within a reasonable range. This confirms that the chosen penalty provides the most stable refinement behavior for multi-dancer generation. We omit TIF metric in this comparison because the variations across settings are minimal.

Table 14 reports the trade-off between inference time and group-level dance quality as the number of dancers increases. As the group size increases from 2 to 6, inference time increases gradually. Importantly, this growth is accompanied by relatively stable group dance quality, indicating that scalability does not come at the cost of coordination or realism. In particular, GMR and GMC remain competitive across different group sizes, with the best overall balance observed at three dancers, while TIF shows only a mild increase for larger groups, indicating that collision artifacts do not escalate dramatically with scale.

These results suggest that although sequential generation incurs additional inference cost for larger groups, the proposed method maintains consistent coordination and motion realism as the number

of dancers increases. This highlights a practical trade-off between scalability and efficiency: higher group sizes naturally require longer generation time, but the quality degradation remains limited and gradual, demonstrating the robustness of the chain-based formulation in multi-dancer scenarios. While not real-time, it is effective for high-quality offline content of arbitrary-sized group motion, whose training data is scarce. Accelerating optimization is future work.

Table 14: **In-depth analysis of the trade-off between inference time and dance quality on the AIOZ-GDance dataset for more dancers.** Best results are **bold**; second-best are underlined.

Number of Dancers	Group Dance			Efficiency
	GMR↓	GMC↑	TIF↓	Inf Time (min:sec)↓
2	13.67	81.28	0.12	**0:07**
3	**11.63**	**83.74**	**0.10**	0:15
4	12.59	82.16	0.13	0:19
5	14.73	81.56	0.12	0:22
6	14.85	80.87	0.13	0:29

Table 15 further separates ChainDance's diffusion sampling time from the optional global optimization time. For three dancers, sampling takes 6 seconds and optimization takes 9 seconds. The optimization share increases from 42.86% at two dancers to 65.52% at six dancers. Without this optional stage, ChainDance* takes 10 seconds for six dancers, compared with 8 seconds for GCD and 16 seconds for TCDiff.

Table 15: **Inference-time breakdown across group sizes (seconds).** Diffusion and optimization are the two components of ChainDance's total inference time. ChainDance* uses diffusion sampling without global optimization. GCD and TCDiff use models trained separately for each group size.

N	Diffusion (ChainDance*)	Global optimization	Optimization share (%)	ChainDance total	GCD	TCDiff
2	4	3	42.86	7	4	7
3	6	9	60.00	15	5	8
4	7	12	63.16	19	6	10
5	8	14	63.64	22	7	13
6	10	19	65.52	29	8	16

D.9 Quantitative Evidence for Music and Text Controllability

To rigorously validate the responsiveness of ChainDance to its multimodal conditioning signals, we provide quantitative measurements using **Beat Alignment Score (BAS)** for music alignment and **R-precision** for text alignment. Since no prior baseline reports such metrics for group dance generation, we report ground-truth (GT) values as reference values.

D.9.1 Follower Audio Ablation

To examine whether the leader's motion alone provides sufficient musical information for a follower, we remove the follower's direct music conditioning while retaining its conditioning on the leader's motion. As shown in Table 16, removing direct audio increases FID from 23.11 to 25.35 and GMR from 11.63 to 15.53, while decreasing GMC from 83.74 to 79.32. BAS also moves further from the ground-truth value of 0.2463. These results show that the leader's motion does not fully substitute for direct music conditioning of each follower.

Table 16: **Effect of direct music conditioning for follower dancers (3 dancers).** The ground-truth BAS is 0.2463.

Setting	FID↓	BAS	GMR↓	GMC↑
w/o follower audio	25.35	0.3332	15.53	79.32
ChainDance	**23.11**	**0.2247**	**11.63**	**83.74**

D.9.2 Matched vs. Mismatched Inputs

We measure model responsiveness by comparing performance under *matched* inputs (correctly paired music/text and motion) versus *mismatched* inputs (randomly swapped from other categories). A large gap between the two indicates that the model genuinely conditions on the input rather than ignoring it.

Table 17: **Quantitative results for music and text alignment.** *Matched* uses correctly aligned inputs; *Mismatched* uses randomly swapped inputs from other categories. A large gap indicates strong responsiveness to the conditioning signal.

	Matched	Mismatched	Gap↑
GT (BAS)	0.2463	–	–
Ours (BAS)	0.2247	0.0624	0.1623
GT (R-precision)	0.5106	–	–
Ours (R-precision)	0.4152	0.1093	0.3059

As shown in Table 17, ChainDance achieves a substantial gap between matched and mismatched conditions ($+0.1623$ for music, $+0.3059$ for text), providing quantitative evidence that the model is genuinely responsive to both music and text inputs. The matched-condition scores remain close to GT (0.2247 vs. 0.2463 for BAS; 0.4152 vs. 0.5106 for R-precision), indicating that ChainDance generates motions well-aligned with the conditioning signals.

D.9.3 Genre-wise Statistical Analysis

To further demonstrate controllability and generation quality across diverse music genres, we conduct a genre-wise analysis over all 10 music genres in the AIOZ-GDance test set, reporting both alignment metrics (BAS, R-Precision) and group dance quality metrics (GMR, GMC, TIF).

Table 18 shows that BAS scores remain consistently close to GT across all genres (gap typically below 0.03), with no notable degradation in R-precision. Group dance quality metrics (GMR, GMC, TIF) remain stable across all genres, with no genre showing significant performance drops. These results confirm that ChainDance maintains *robust dual-condition control and group coordination across a broad range of musical styles*, rather than being biased toward a particular genre. Together with Table 17, these analyses provide rigorous quantitative evidence supporting the controllability claim made in the main paper.

Table 18: **Quantitative results across different music genres.** ChainDance maintains stable alignment scores (BAS, R-precision) and consistent group dance quality (GMR, GMC, TIF) across all 10 genres, demonstrating robust dual-condition control.

Music genre	BAS		R-precision		Group Dance Quality		
	GT	Ours	GT	Ours	GMR↓	GMC↑	TIF↓
Disco	0.2871	0.2661	0.5234	0.4194	13.52	83.03	0.12
Electronic	0.2712	0.2562	0.5076	0.4212	13.21	82.83	0.13
Folk	0.2154	0.1961	0.5112	0.4287	13.84	81.97	0.13
Funk	0.2176	0.1923	0.5154	0.4431	12.87	82.42	0.12
Indian	0.2234	0.2001	0.5223	0.4076	13.48	83.13	0.12
Latin	0.2715	0.2504	0.4989	0.3821	13.24	81.32	0.13
Pop	0.2203	0.1983	0.5187	0.4102	12.76	80.98	0.12
R&B	0.2728	0.2553	0.4923	0.3954	13.49	82.84	0.13
Rap	0.2156	0.1929	0.5067	0.3998	11.98	83.13	0.12
Reggae	0.2681	0.2393	0.5095	0.4445	12.57	82.76	0.12

D.10 Order Sensitivity Analysis

A natural concern with sequential generation is whether ChainDance is sensitive to the choice of generation order or leader selection. Although Section 3.2 of the main paper shows that chain pairing and random pairing yield comparable results, that experiment evaluates each strategy only once. To rigorously assess the model's sensitivity to generation order, we conduct two additional analyses on the AIOZ-GDance dataset.

We note that the AIOZ-GDance dataset *lacks leader annotations*, and its dancer indexing does not correspond to any physical position. ChainDance therefore does not depend on a predefined order or fixed leader choice—instead, the model is encouraged to focus on *who participates*, *their motion characteristics*, and *their relative spatial relations*, rather than the absolute feature dimension assigned to each dancer.

D.10.1 All Possible Orders for 3-Dancer Generation

We randomly sample 100 sequences with 3 dancers and evaluate ChainDance under all $3! = 6$ possible generation orders. Results are reported in Table 19.

Table 19: **Evaluation results for 3-dancer generation under all possible generation orders.** The model is evaluated on 100 randomly sampled sequences. Standard deviations across orders are negligible compared to the gap between methods, indicating order-insensitivity.

Order	GMR↓	GMC↑	TIF↓
123	11.42	83.53	0.11
132	11.54	82.67	0.10
213	11.14	83.76	0.10
231	11.61	83.21	0.11
312	11.63	84.03	0.11
321	11.73	84.52	0.11
Mean ± Std	11.51 ± 0.22	83.62 ± 0.67	0.107 ± 0.005

As shown in Table 19, varying the generation order causes *negligible* performance variance, with standard deviations of only 0.22 (GMR), 0.67 (GMC), and 0.005 (TIF). These variations are far smaller than the gaps between ChainDance and prior baselines (e.g., GMR gap of 3.13 against TCDiff in Table 1), indicating that the model is insensitive to the choice of generation order.

This robustness arises from two design choices: (i) the dancer indexing in AIOZ-GDance is not aligned with physical positions, so the model cannot exploit fixed-order priors during training; and (ii) the GCN-based Group-Aware Motion Encoder captures coordination through *relative* positions and connectivity, rather than absolute feature dimensions assigned to each dancer.

D.10.2 Random Orders for 5-Dancer Generation

To verify that order-insensitivity also holds for larger groups, we further evaluate ChainDance on 5-dancer generation under 4 different random orders.

Table 20: **Results for 5-dancer generation under 4 different random orders.** ChainDance maintains stable performance across orders, confirming that order-insensitivity extends to larger group sizes.

Run	GMR↓	GMC↑	TIF↓
1	14.87	81.21	0.13
2	14.63	81.63	0.13
3	14.32	81.83	0.13
4	14.21	82.02	0.12
Mean ± Std	14.51 ± 0.31	81.67 ± 0.35	0.128 ± 0.005

Table 20 confirms that the model remains order-insensitive at larger group sizes, with standard deviations on the same order of magnitude as the 3-dancer case. The observed variations are again much smaller than the performance gaps between competing methods, demonstrating that ChainDance produces *consistent group choreography regardless of the chosen generation order*, even as group size increases.

E Limitations and Future Work

While ChainDance addresses group-size rigidity and structurally preserves dancer identity by construction, several limitations remain that point to future research directions.

Inference Latency. While our approach avoids the prohibitively expensive $\mathcal{O}((NL)^2)$ training cost associated with joint modeling, its inference latency is relatively high. This is a common bottleneck for diffusion-based models due to the iterative denoising process, which currently limits strictly real-time interactive deployment. However, this latency (measured in seconds rather than minutes) is acceptable for our primary target applications, such as offline animation and video production. Accelerating inference toward real-time interactivity is a natural next step, e.g., via diffusion distillation or consistency-model variants.

Scaling to Massive Crowds. While ChainDance addresses flexible group-size modeling, generating highly coordinated motions for extremely large crowds (e.g., $N > 10$) remains a distinct research problem. Although our dynamic growing graph and test-time penalty optimization effectively mitigate spatial inconsistencies for typical group sizes, maintaining uniform global coherence at massive scale becomes increasingly difficult. Our current capacity nonetheless covers the majority of real-world applications, such as generating pop-group (e.g., boy or girl band) choreographies for music videos and gaming, where group sizes rarely exceed ten. To scale seamlessly to massive crowds, future work will explore parallel-friendly mechanisms or hierarchical generation strategies.

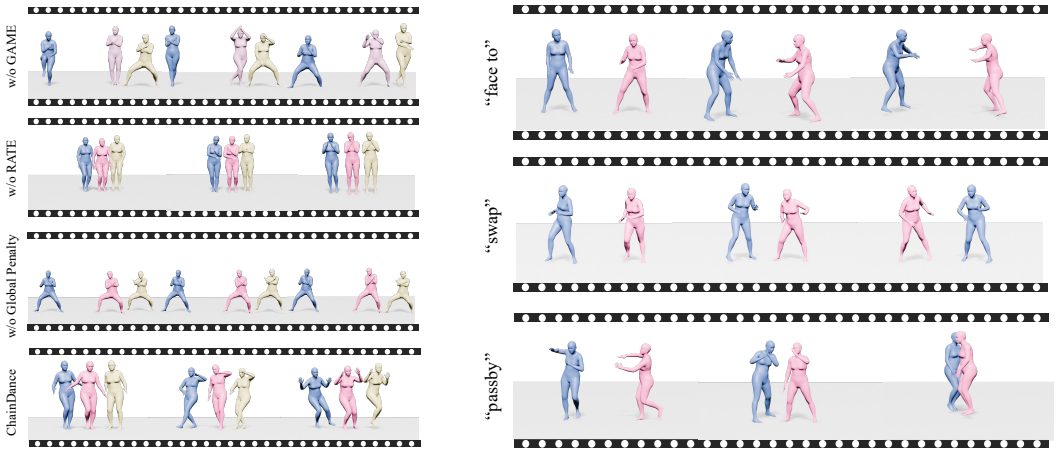
F More Qualitative Results

To visually demonstrate the effect of individual components in ChainDance, we present the qualitative results of the ablation study in Figure 7a. Without GAME, each dancer moves almost independently, and the group looks uncoordinated. Without text conditioning, the motions of different dancers become too similar and lack variation. Without the global penalty, errors accumulate over time, and dancers drift to unreasonable positions with overly large distances. These examples show that all three parts are important for coordinated, diverse, and reasonable group choreography.

Then, we provide additional frame-by-frame visualizations of the three types of interaction control in Figure 7b, and show ChainDance results under different group sizes in Figure 9. Figure 8 further visualizes the process of progressively adding dancers to the same music, showing that ChainDance integrates new dancers smoothly and remains effective for large groups of 6–8 dancers. All examples

follow the same visualization settings as in the main paper, where we use the simple SMPL built-in mesh to convert PKL sequences into FBX files.

For the supplementary video demos included in the **index.html** file, we further adopt the refined YBot model to render the motions with higher visual quality. The videos are rendered in Blender using scripts adapted from the open-source TCDiff [3] code, to which we are grateful.



(a) Ablation study (b) Interaction control via text

Figure 7: **Qualitative visualizations.** (a) Ablation study showing the effect of each component. (b) Interaction control under different text prompts.

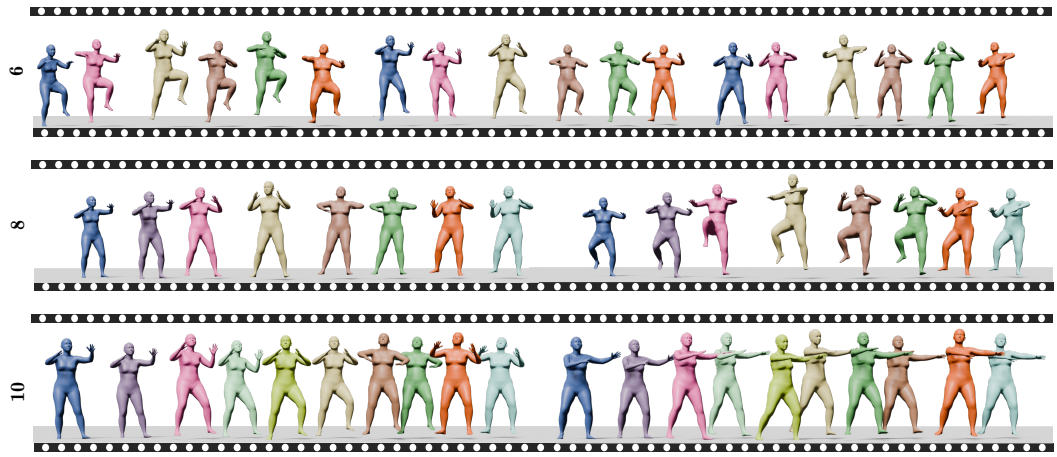


Figure 8: **ChainDance results on the same music under varying group sizes (from 6 to 10 dancers).** The generated choreography remains coordinated and diverse as the group size scales up.

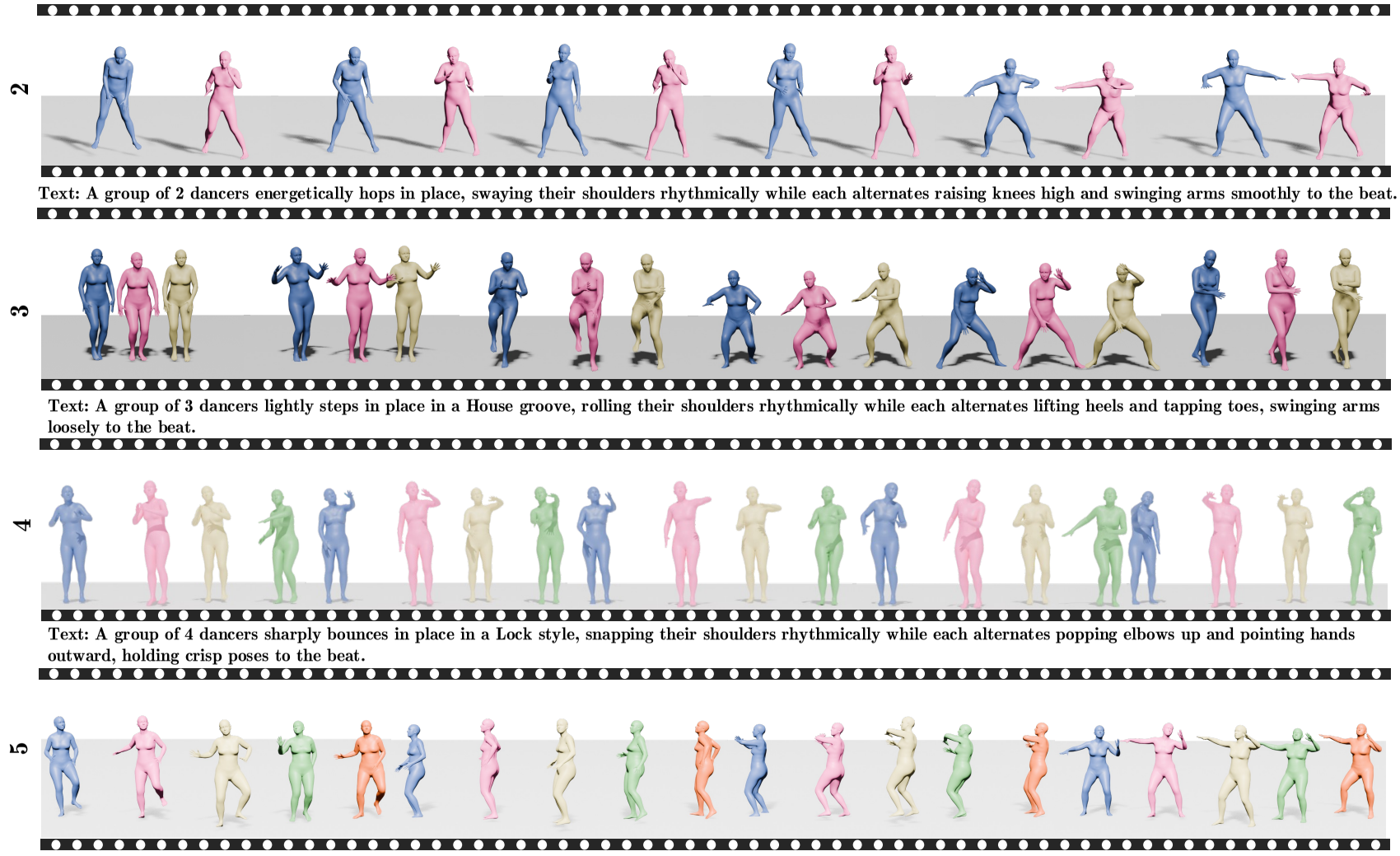


Figure 9: Qualitative results of ChainDance under different group sizes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our main contributions: (i) reformulating group dance generation as a chain of conditional single-dancer generations, (ii) a unified model that generalizes across group sizes, and (iii) training-free global optimization at inference. These claims are directly supported by experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The probabilistic factorization underlying our chain-of-dancers formulation is presented in Appendix A, where we derive the conditional decomposition of the joint group dance distribution from the chain rule of probability. The derivation is exact and does not introduce any approximation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed implementation information in Section 4.1 and Appendix B, including backbone architecture, adapter design, graph construction, loss functions and weights, training hyperparameters (optimizer, learning rate, batch size, epochs), inference penalty configurations, and dataset preprocessing details. All datasets used (AIST++, FineDance, AIOZ-GDance) are publicly available, and we follow standard train/test splits.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets used (AIST++, FineDance, AIOZ-GDance) are publicly available. Source code is not released at submission time to preserve double-blind review, and we plan to release it upon acceptance. The paper and appendix provide sufficient implementation details (architecture, hyperparameters, training and inference procedures) to enable independent reproduction.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 reports the experimental setup, including datasets, train/test splits, evaluation metrics, training strategy, and key hyperparameters (loss weights, sequence length, optimizer, learning rate, number of epochs). Additional implementation details for the loss functions, graph construction, and inference penalties are provided in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables 1 and 3 report the mean and sample standard deviation of each metric over 20 repeated generation and evaluation runs. Each run includes resampling the first dancer for ChainDance. The standard deviation is computed across the 20 run-level metric values and captures inference-time sampling variability with the trained models and test split held fixed; it does not measure variation across independently trained models. We do not interpret these values as confidence intervals or assume a normal distribution.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As reported in Section 4.1, all experiments are conducted on a dual NVIDIA RTX 4090 GPU setup. Training takes approximately 14 seconds per epoch for 500 epochs. Inference time across different group sizes (2–6 dancers) and different optimization-step settings is reported in Tables 2, 13, and 14.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research follows the NeurIPS Code of Ethics. All datasets used are publicly available with established licenses. The user study (reported in Appendix D.5) involves anonymous participants providing perceptual rankings of generated motion sequences, with no collection of personally identifiable information.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: ChainDance can lower the barrier for group choreography production in entertainment, education, and virtual content creation, enabling smaller teams or individuals to create coordinated group performances. Potential negative impacts include misuse of generated motion in deepfake-style content; however, our method outputs only skeleton-level motion sequences (not photorealistic video), which substantially reduces such risks compared to image- or video-level generative models. We encourage downstream users to combine our outputs with appropriate watermarking when rendered into final visual content.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our model outputs only skeleton-level motion sequences rather than photorealistic visual content, posing limited risk for direct misuse. We do not release new datasets or pre-trained large generative models that pose high misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring

that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets (AIST++ [22], FineDance [20], AIOZ-GDance [17]) and pre-trained backbones (Lodge [21], EDGE [34]) are properly cited. We use these assets in accordance with their respective licenses for academic research.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new datasets or models at submission time. Code and trained model weights will be released upon acceptance with appropriate documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

Justification: Our user study (Appendix D.5) describes participant recruitment (30 participants), the four perceptual evaluation dimensions (Realism, Music Correlation, Formation Aesthetics, Dancer Harmony), and the blind ranking protocol on randomized videos. The study was a brief perceptual evaluation (approximately 10 minutes per participant), conducted as voluntary academic participation rather than as a paid labor task.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our user study only involves participants viewing publicly displayable animated dance videos and providing subjective rankings, with no personal data collection or sensitive content. The study poses minimal risk and falls outside the scope of activities typically requiring IRB review under our institutional guidelines.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: ChatGPT-4o is used in the data preparation pipeline to verbalize structured multimodal metadata into natural language prompts for training (Section 3.1). This usage is limited to pseudo-label generation and does not constitute an important, original, or non-standard component of the proposed method. The core contributions are independent of the specific LLM used and would remain valid with any alternative text generation approach.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.