RAMP: RHYTHM-ADAPTIVE MODELING WITH PROGRESSIVE TEACHER REGULARIZATION FOR DANCE-TO-MUSIC GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dance-to-music (D2M) generation requires fine-grained motion-rhythm control while maintaining coherent and semantically controllable music. We introduce **RAMP**, a framework for incorporating structured dance rhythm into a pretrained text-to-music (T2M) model. Its core component, the Style-Adaptive Rhythm Tokenizer (SART), performs global-to-local routing over shared rhythm-biased experts: a style-conditioned routing prototype captures clip-level rhythmic preferences, while motion-conditioned residual routing follows frame-level variation. To learn from these new conditions, RAMP uses teacher-regularized multimodal adaptation: it first adapts to the target text and lyric distribution, then introduces dance-aware conditioning while regularizing text-only velocity predictions toward a frozen Stage-1 teacher. The backbone's native text and lyric pathways remain explicit controls for high-level musical intent. We further construct **Dance100K** and a fixed 3K evaluation benchmark. Experiments show strong rhythmic alignment together with competitive semantic consistency and perceptual quality, supporting structured rhythm routing and teacher-regularized adaptation as complementary designs for controllable D2M generation. Anonymous project page.

1 INTRODUCTION

Dance-to-Music (D2M) generation aims to synthesize music that follows the rhythm and expressive dynamics of dance while maintaining coherent and style-consistent musical structure Ji et al. (2026); Aggarwal & Parikh (2021); Yang et al. (2025a). Existing D2M methods commonly represent dance using either global motion embeddings Zhu et al. (2022a); Sun et al. (2025); Liang et al. (2024); Han et al. (2024) or explicit rhythmic descriptors Su et al. (2021); Yu et al. (2023); Wang et al. (2025b). Global embeddings capture movement semantics but encode rhythmic structure only implicitly, whereas explicit descriptors provide more direct temporal cues. GACA-DiT Wang et al. (2025a) advances this direction with genre-adaptive rhythm extraction and context-aware temporal alignment. Complementary evidence that motion exhibits both shared patterns and dance-genre-specific variations Yang et al. (2025b) motivates the representation question: **how to model persistent dance-style rhythmic preferences together with frame-level motion variation**. Dance rhythm alone does not fully determine higher-level musical intent, such as genre, mood, instrumentation, vocals, or lyrical content. We therefore retain the pretrained backbone's native text and lyric pathways as explicit semantic controls rather than treating them as a separate method contribution. The adaptation question is how to incorporate the structured rhythm representation into a pretrained T2M model while constraining drift from a text-conditioned behavioral reference. Thus, RAMP addresses two coupled questions: how to represent dance rhythm at both clip and frame levels, and how to learn from that representation during multimodal adaptation without unconstrained changes to existing text-conditioned behavior.

We propose **RAMP**, a framework for introducing structured dance-rhythm control into a pretrained text-to-music model. Its core component, the **S**tyle-**A**daptive **R**hythm **T**okenizer (**SART**), combines a style-conditioned routing prototype with motion-conditioned residual routing to capture clip-level rhythmic preferences and frame-level motion variation. To incorporate these rhythm representations into the generator, we develop a two-stage, teacher-regularized adaptation scheme: the model first adapts to the target text and lyric distribution, then learns dance-aware conditioning while its

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

text/lyric-only velocity predictions are regularized toward a frozen Stage-1 teacher. Text and lyrics remain explicit controls for musical semantics through the backbone's native conditioning pathways.

To support training and evaluation, we construct **Dance100K**, a large-scale, temporally aligned dance–music corpus integrating motion, audio, dance style, textual descriptions, and lyric annotations. We further establish **Dance100K-Benchmark** with no source-item overlap between training and evaluation, enabling reliable assessment of rhythmic alignment, semantic consistency, and perceptual quality. Experiments show that RAMP achieves stronger rhythmic alignment while maintaining competitive semantic consistency and perceptual quality.

Our main contributions are summarized as follows:

- **Global-to-local rhythm routing.** We introduce SART, a structured dance-rhythm tokenizer that combines a shared bank of kinematically biased experts with a style-conditioned routing prototype and motion-conditioned residual routing. This design captures clip-level rhythmic preferences while adapting expert weights to frame-level motion dynamics.
- **Teacher-regularized multimodal adaptation.** We develop a two-stage adaptation scheme that first adapts a pretrained text-to-music model to the target text and lyric distribution, then introduces dance-derived conditioning. During multimodal adaptation, a frozen Stage-1 teacher regularizes the student's text/lyric-only velocity predictions, providing a behavioral reference while the student learns dance-aware control.
- **Dance100K and a fixed evaluation benchmark.** We construct Dance100K, a large-scale aligned dance–music dataset with motion and textual annotations, and establish a fixed 3,000-sample evaluation benchmark with source-item-level disjointness. We evaluate rhythmic alignment, text–music consistency, and perceptual quality through automatic metrics and a human study.

2 Related Work

2.1 Dance-to-Music Generation

Existing Dance-to-Music (D2M) methods primarily learn representations that capture the correspondence between dance motion and music. One line conditions music generation on global motion representations extracted from pose or video encoders, such as ST-GCN and I3D Zhu et al. (2022a); Sun et al. (2025); Liang et al. (2024); Han et al. (2024). These representations capture high-level movement semantics but encode rhythmic structure only implicitly. Another line extracts explicit motion-derived cues, including velocity, energy, and motion beats Su et al. (2021); Yu et al. (2023); Wang et al. (2025b). GACA-DiT Wang et al. (2025a) further introduces genre-adaptive rhythm extraction and context-aware temporal alignment. Beyond motion representation, recent works incorporate dance genre or transfer dance-derived rhythm and semantic representations into pretrained text-to-music (T2M) models Zhang & Hua (2024); Li et al. (2024). CMA-OT Wang et al. (2026) further enriches D2M supervision with hierarchical representations from an external music expert. While these approaches demonstrate the importance of rhythmic and semantic guidance, they do not explicitly factorize dance and text into complementary conditioning roles. RAMP instead models dance style as a global rhythmic prior, uses local motion dynamics for fine-grained rhythmic control, and employs text to specify high-level musical semantics.

2.2 Music Generation Foundation Models

Recent advances in generative modeling have established text-to-music (T2M) foundation models as powerful backbones for controllable music generation. Models such as MusicLM Agostinelli et al. (2023) and MusicGen Copet et al. (2023) demonstrate strong language-conditioned music generation, while diffusion-based models such as Stable Audio Evans et al. (2025) and DiffRhythm Ning et al. (2025); Jiang et al. (2025) further improve audio quality and long-form synthesis. More recent models, including YuE Yuan et al. (2026) and SongGen Liu et al. (2025), extend these capabilities toward scalable long-form song generation. Among open-source T2M models, ACE-Step Gong et al. (2025) and ACE-Step 1.5 Gong et al. (2026) offer strong musical priors together with flexible text-based semantic control, making them particularly suitable as backbones for multimodal music generation. However, adapting such text-native models to additional motion-derived conditions

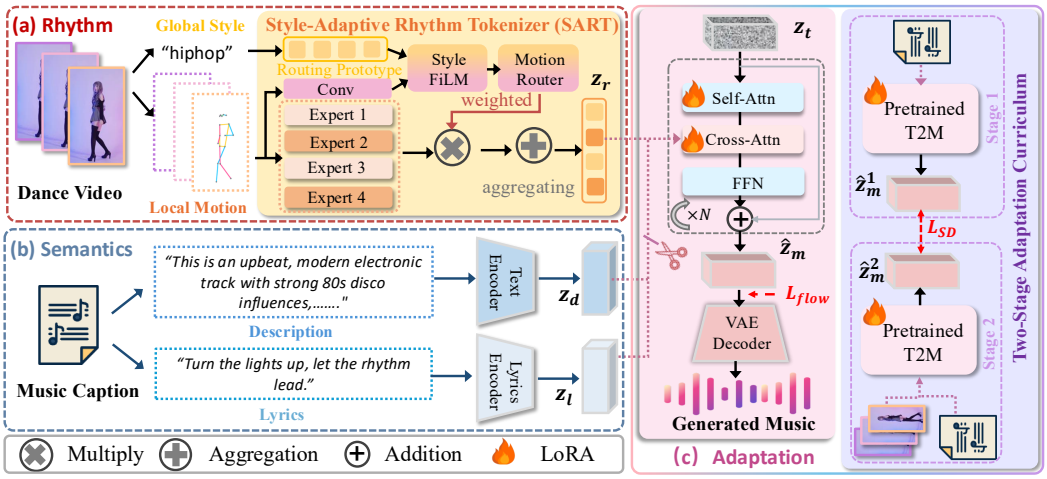


Figure 1: Overview of RAMP. (a) **Rhythm representation:** SART uses a style-conditioned routing prototype and motion-conditioned residual routing over shared rhythm-biased experts to capture clip-level preferences and frame-level variation. (b) **Retained native semantic conditions:** the pretrained backbone's text and lyric pathways provide explicit controls for high-level musical intent. (c) **Teacher-regularized adaptation:** RAMP progresses from text/lyric-only to joint dance-aware conditioning, with $\mathcal{L}_{\text{SD}}$ constraining drift from the frozen Stage-1 teacher's text/lyric-conditioned velocity predictions.

remains non-trivial, as the new modalities may introduce a modality gap and induce drift in the backbone's text-conditioned behavior.

3 Method

3.1 Overview

Given a dance video V, an associated dance-style tag s, a music description D, and lyrics L, **RAMP** introduces structured dance-rhythm control into a pretrained T2M model. Its core representation component, the **Style-Adaptive Rhythm Tokenizer (SART)**, produces structured rhythm tokens $\mathbf{Z}_r^{\text{SART}}$ through global-to-local routing: the style-conditioned routing prototype captures clip-level rhythmic preferences, while motion-conditioned residual routing follows frame-level dynamics. A multimodal conditioning interface combines these rhythm tokens and general dance tokens with the backbone's retained text and lyric conditions. The generator then learns to use the added dance-derived conditions through teacher-regularized multimodal adaptation, which constrains its text/lyric-only velocity predictions toward a frozen Stage-1 teacher.

3.2 Style-Adaptive Rhythm Tokenizer

Dance rhythm varies at two coupled scales: style creates persistent rhythmic preferences across a clip, while individual movements introduce local changes. A single motion representation tends to entangle these effects. SART addresses this problem with a shared, kinematically biased expert bank and global-to-local routing, separating clip-level style guidance from frame-level motion adaptation.

Explicit Rhythm Representation. Raw pose sequences contain fine-grained spatial configurations and motion details that do not all directly characterize rhythm. Given a pose sequence $\mathbf{P}$, we therefore construct the explicit rhythm sequence $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T]$, where

$$\mathbf{x}_t = [e_t, a_t, p_t, \sin(2\pi\phi_t), \cos(2\pi\phi_t)] \in \mathbb{R}^5, \tag{1}$$

where e_t, a_t, p_t, and ϕ_t denote motion energy, acceleration magnitude, a motion-peak indicator, and peak-relative phase at frame t, respectively. These deterministic, style-agnostic descriptors

summarize motion intensity, transients, salient peaks, and local phase while reducing sensitivity to pose variations less relevant to rhythm. They provide low-level analytic rhythm cues without encoding dance-style information or imposing expert assignments; style-conditioned preferences and motion-conditioned framewise routing are learned by subsequent SART components.

Kinematically Biased Rhythm Experts. Dance contains qualitatively different temporal patterns, including abrupt impacts, periodic grooves, sustained holds, and smooth phrases. A single encoder may average them into an overly generic representation. SART instead uses $K = 4$ independently parameterized experts:

$$\mathbf{Z}^{(k)} = E_k(\mathrm{View}_k(\mathbf{X})), \qquad k \in \{1, \dots, K\}, \tag{2}$$

where View_k emphasizes the *sharp-hit*, *groove*, *freeze/hold*, or *smooth-phrase* cues, and E_k is the corresponding expert encoder. The resulting sequences $\mathbf{Z}^{(k)}$ are complementary rhythm candidates shared across styles.

Global-to-Local Style-Adaptive Routing. Different dance styles favor different rhythmic patterns, but their timing and strength vary within a clip. Local motion alone may yield inconsistent style, whereas a fixed mixture cannot follow frame-level changes. We therefore derive a style-conditioned routing prototype from training-split motion statistics:

$$u_{s,k} = \frac{1}{|\mathcal{D}_s|} \sum_{i \in \mathcal{D}_s} r_k(\mathbf{X}_i), \qquad \mathbf{p}_s^{\mathrm{proto}} = \mathrm{Softmax}(\mathbf{u}_s / T_p), \tag{3}$$

where $\mathcal{D}_s$ is the set of training clips with style s, r_k is the kinematic score aligned with expert k, and T_p is the prototype temperature. The prototype is uniformly smoothed to $\widetilde{\mathbf{p}}_s^{\mathrm{proto}}$ and then frozen, avoiding manually assigned style-to-expert weights.

At each frame, the prototype supplies a global logit bias and FiLM-conditions a motion router that predicts a bounded residual $\boldsymbol{\delta}_t$. Routing and expert aggregation are

$$\begin{aligned} \mathbf{q}_t &= \mathrm{Softmax}\left(\frac{\lambda_{\mathrm{style}} \log \widetilde{\mathbf{p}}_s^{\mathrm{proto}} + \mathbf{b} + \alpha_{\mathrm{route}} \boldsymbol{\delta}_t}{\tau} \right), \\ \mathbf{z}_t &= \sum_{k=1}^{K} q_{t,k} \mathbf{Z}_t^{(k)}, \end{aligned} \tag{4}$$

where $\mathbf{q}_t$ contains the expert weights, $\mathbf{b}$ is a learned expert bias, λ_{style} controls prototype strength, α_{route} scales local adaptation, and τ is the routing temperature. Together, the prototype and residual router preserve clip-level rhythmic coherence while adapting to momentary motion events. The resulting raw SART sequence $\mathbf{Z}_r^{\mathrm{SART}} = [\mathbf{z}_1, \dots, \mathbf{z}_T] \in \mathbb{R}^{T \times 512}$ is subsequently projected as the rhythm-conditioning stream.

3.3 Multimodal Conditioning Interface

Dance-derived conditions provide temporal and rhythmic guidance, but they do not fully determine musical genre, instrumentation, mood, or vocal content. We therefore retain the native text and lyric pathways of ACE-Step Gong et al. (2026) as complementary controls. Given a music description D and lyrics L, these pathways produce $\mathbf{Z}_d = E_{\mathrm{text}}(D)$ and $\mathbf{Z}_l = E_{\mathrm{lyric}}(L)$, which form the semantic condition $\mathbf{C}_{\mathrm{sem}}$. RAMP augments this semantic condition with general dance-conditioning tokens $\mathbf{Z}_{\mathrm{dance}}$ and rhythm-specific SART tokens $\bar{\mathbf{Z}}_r$:

$$\mathbf{C}_{\mathrm{full}} = [\mathbf{Z}_{\mathrm{dance}}; \bar{\mathbf{Z}}_r; \mathbf{C}_{\mathrm{sem}}]. \tag{5}$$

This complementary design lets dance-derived conditions primarily organize temporal structure while text and lyrics steer high-level musical semantics, without assuming complete separation between the modalities.

3.4 Teacher-Regularized Multimodal Adaptation

We progressively adapt the pretrained T2M model in two stages. Stage 1 adapts the backbone to the text and lyric distribution, establishing a task-compatible text-conditioned initialization. Stage 2 then introduces dance-derived rhythmic guidance and uses stage-wise self-distillation to stabilize the Stage-1 text-conditioned behavior.

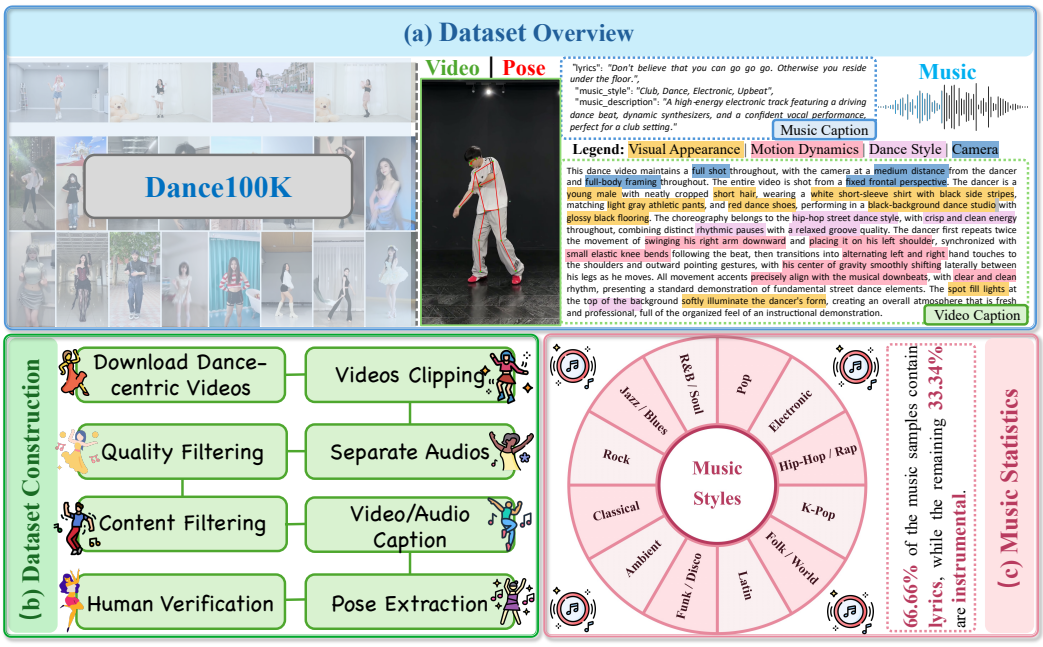


Figure 2: Overview of Dance100K. (a) Each sample is a multimodal dance clip paired with video, pose, music, and textual annotations, including detailed music and video descriptions. (b) Dataset construction pipeline, from video collection and filtering to captioning, verification, and pose extraction. (c) Statistics of music styles and lyric availability, illustrating the diversity of the collected musical content.

Stage 1: Text-Space Adaptation. We first adapt the model to the Dance100K text and lyric distribution while remaining within its native text-conditioned space, using only the layout-compatible text/lyric-only condition $\mathbf{C}_{\text{text}}$, whose 640-position non-text prefix is zeroed and masked. Let $\mathbf{x}_0$ be the target audio latent and $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be Gaussian noise. At flow time t, we form $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$ with target velocity $\mathbf{v}^\star = \mathbf{x}_1 - \mathbf{x}_0$. The Stage-1 flow-matching objective is

$$\mathcal{L}_{\text{FM}}^{(1)} = \|\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{C}_{\text{text}}) - \mathbf{v}^\star\|_2^2 . \tag{6}$$

This stage produces a task-compatible text-conditioned initialization; a frozen copy of it serves as the teacher when dance-derived conditions are introduced.

Stage 2: Multimodal Adaptation with Self-Distillation. We then expand the conditioning space to $\mathbf{C}_{\text{full}} = [\mathbf{Z}_{\text{dance}}; \bar{\mathbf{Z}}_r; \mathbf{C}_{\text{sem}}]$ to incorporate dance-derived temporal guidance. The corresponding flow-matching objective is

$$\mathcal{L}_{\text{FM}}^{(2)} = \|\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{C}_{\text{full}}) - \mathbf{v}^\star\|_2^2 . \tag{7}$$

To prevent this additional condition from disrupting the musical behavior established in Stage 1, we freeze the Stage-1 model as a teacher and use its text/lyric-only prediction as a behavioral reference. Specifically, for the same $\mathbf{x}_t$ and t, we define

$$\mathcal{L}_{\text{SD}} = \|\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{C}_{\text{text}}) - \mathbf{v}_{\text{T}}(\mathbf{x}_t, t, \mathbf{C}_{\text{text}})\|_2^2 , \tag{8}$$

where $\mathbf{v}_\theta$ and $\mathbf{v}_{\text{T}}$ denote the Stage-2 student and frozen Stage-1 teacher, respectively. Thus the Stage-2 student has two forward branches: the flow-matching branch receives the full multimodal condition $\mathbf{C}_{\text{full}}$, whereas the distillation branch evaluates both student and teacher with the same text/lyric-only condition $\mathbf{C}_{\text{text}}$; the teacher never receives dance or rhythm tokens. The student is optimized with

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{FM}}^{(2)} + \lambda_{\text{SD}}\mathcal{L}_{\text{SD}}, \tag{9}$$

where λ_{SD} controls the strength of the self-distillation constraint. The resulting curriculum enables the model to acquire dance-conditioned temporal control while stabilizing the text-conditioned music-generation behavior inherited from the Stage-1 teacher.

Table 1: System-level comparison on rhythmic alignment, quality, musicality, and semantic alignment. Methods marked † do not natively accept lyric conditions; their L-WER values are diagnostic and excluded from lyric-control comparisons.

	Rhythmic Alignment			Quality & Coherence			Text Alignment	
Methods	**ABHS ↑**	**MDA$_{\text{phys}}$ ↑**	**F1 ↑**	**Coh. ↑**	**PQ ↑**	**Mus. ↑**	**L-WER ↓**	**CLAP ↑**
D2M-GAN†	0.0677	0.1799	0.3659	1.8544	5.9301	1.9861	1.4611	-0.0910
CDCD†	0.0980	0.1341	0.3955	1.7667	7.1156	1.8421	1.3871	-0.0056
LORIS†	**0.1169**	0.2467	0.5101	2.3146	5.5068	2.4381	1.7004	0.0484
Text-Inv†	0.1049	0.2703	0.5998	2.4590	6.3657	2.3609	1.7122	0.0541
MotionComposer†	0.1091	0.2814	0.6049	1.7861	4.2374	1.7735	1.6755	-0.0309
CMA-OT†	0.1116	0.2840	0.5599	2.6297	6.5559	2.5806	1.2967	0.1578
VIBE†	0.1058	0.2794	0.5772	2.4117	6.7853	2.3829	1.5921	0.2071
RAMP (Ours)	0.1117	**0.2978**	**0.6224**	**2.9947**	**7.6686**	**2.9804**	**1.0543**	**0.2923**

4 EXPERIMENTS

4.1 EXPERIMENTAL DETAILS

Datasets. We introduce **Dance100K**, a large-scale multimodal dataset for D2M generation. It contains approximately 100K in-the-wild dance clips collected from social media platforms, covering diverse dance styles, performers, viewpoints, and real-world recording conditions. As shown in Figure 2, each clip is associated with five complementary modalities and annotations: dance video, pose sequence, music waveform, video caption, and music caption. Together, they provide motion, visual, and musical semantic information for training and evaluating D2M models. We reserve 3,000 samples as a fixed evaluation benchmark, **Dance100K-Benchmark**, with source-item-level disjointness from the training set. This design prevents source-item overlap between training and evaluation. Model weights are not optimized on this benchmark; however, we use it both to select λ_{SD} and the Stage-1 training ratio and to report final results, as no separate validation split is available. The remaining samples are used for training. Dance100K also covers diverse musical styles and varying lyric availability. Details of data collection, filtering, annotation, quality control, and data splitting are provided in Appendix B.

Evaluation Metrics. We evaluate generated music from four perspectives. For dance–music rhythmic alignment, we use the physical track of MDAlign from TMD-Bench Yang et al. (2026), including Video Beat Consistency Score (VBCS) and Audio Beat Hit Score (ABHS), whose arithmetic mean is denoted as MDA_{phys}. Following the established protocol Yu et al. (2023), we additionally report Beat Coverage Score (BCS), Beat Hit Score (BHS), and Beat F1 for onset-level alignment. For text–music consistency, we use CLAP Wu et al. (2023). For perceptual quality, we report Meta AudioBox Aesthetics Tjandra et al. (2025), including Content Enjoyment (CE), Content Usefulness (CU), Production Complexity (PC), and Production Quality (PQ), as well as SongEval Yao et al. (2025), using its released predictor for our 5-second clips to measure Coherence (Coh.), Memorability (Mem.), Clarity (Clar.), Naturalness (Nat.), and Musicality (Mus.). For samples with reference lyrics, we additionally report Lyrics Word Error Rate (L-WER) based on Whisper large-v3 transcriptions Radford et al. (2023).

Implementation Details. We fine-tune ACE-Step-1.5 Base for 12 epochs using rank-64 LoRA adapters on the attention projections. The first 20% of optimization steps form Stage 1 at a learning rate of 5×10^{-5}; the remaining steps form Stage 2 at 10^{-4} with $\lambda_{\text{SD}} = 0.15$. The pretrained SART checkpoint remains frozen. Inference uses 50-step explicit Euler integration and Adaptive Projected Guidance with scale 7.0. Full inference settings are provided in Appendix D.1.

Table 2: Ablation study of RAMP. Each block tests structured rhythm modeling, the components of multimodal adaptation, or teacher design.

	Rhythmic Alignment			Music Quality		Text Alignment	
Variants	**ABHS ↑**	**MDA$_{\text{phys}}$ ↑**	**F1 ↑**	**PQ ↑**	**Mus. ↑**	**L-WER ↓**	**CLAP ↑**
Rhythm Representation							
ST-GCN	**0.1128**	0.2869	0.5740	6.2215	2.3562	1.3811	0.1656
Binary Rhythm	0.1113	0.2763	0.5935	5.9997	2.4942	1.2848	0.2042
SART	0.1117	**0.2978**	**0.6224**	**7.6686**	**2.9804**	**1.0543**	**0.2923**
Adaptation Components							
ACE-Step	0.1055	0.2244	0.5086	7.4975	2.7710	1.1981	0.2036
One-stage w/o SD	0.1005	0.2627	0.5715	5.7981	2.4110	1.4004	0.1598
Two-stage w/o SD	**0.1129**	**0.2983**	0.5955	6.8418	2.5957	1.0812	0.2567
Two-stage + SD	0.1117	0.2978	**0.6224**	**7.6686**	**2.9804**	**1.0543**	**0.2923**
Teacher Design							
Base Teacher	0.1092	0.2921	0.5767	6.5718	2.5341	1.1667	0.2387
Dual Teacher	0.1101	0.2919	0.5820	6.6131	2.4926	1.1611	0.2288
Stage-1 Teacher	**0.1117**	**0.2978**	**0.6224**	**7.6686**	**2.9804**	**1.0543**	**0.2923**

4.2 Main Results

4.2.1 Comparison with SOTA Methods

We compare RAMP with representative D2M methods on Dance100K-Benchmark, including D2M-GAN Zhu et al. (2022a), CDCD Zhu et al. (2022b), LORIS Yu et al. (2023), Text-Inv Li et al. (2024), MotionComposer Wang et al. (2025b), and CMA-OT Wang et al. (2026), as well as the multimodal music generation method VIBE Bhosale et al. (2026).

As shown in Table 1, RAMP achieves the best beat-level F1 score among the evaluated systems, obtaining 0.6224 and improving over the strongest competing system by 2.9%. It also achieves the best MDA$_{\text{phys}}$ while remaining competitive on ABHS, establishing strong full-system rhythmic alignment. RAMP further obtains the best Coherence, PQ, Musicality, and CLAP scores. Because the compared systems use different pretrained backbones and natively supported conditions, these semantic and perceptual results are full-system outcomes of the complete RAMP design: adopting a pretrained T2M backbone, retaining its native text/lyric semantic conditioning, and integrating the proposed rhythm and adaptation mechanisms. They should therefore not be attributed to any single component. L-WER is interpreted as lyric controllability only for methods with native lyric conditioning; values for methods marked † are diagnostic because those systems do not accept lyrics. We rely on the same-ACE-Step ablations in Table 2 for component-level attribution.

4.2.2 Ablation Study

We organize controlled ablations around the two method contributions. For global-to-local rhythm routing, we compare rhythm representations and intervene on the expert bank, style-conditioned prototype, and motion-conditioned routing. For teacher-regularized multimodal adaptation, we compare self-distillation, adaptation schedules, and teacher designs. All trained variants use the same ACE-Step-1.5 Base initialization, LoRA configuration, training split, and evaluation protocol; only the stated component is changed.

Rhythm Representation. We replace SART with either ST-GCN motion features or binary rhythm cues while keeping the downstream generator unchanged. SART improves F1, while also improving PQ. These results indicate that structured rhythm representations provide more informative temporal control than generic motion features or sparse beat cues, improving rhythmic alignment while also enhancing perceptual quality.

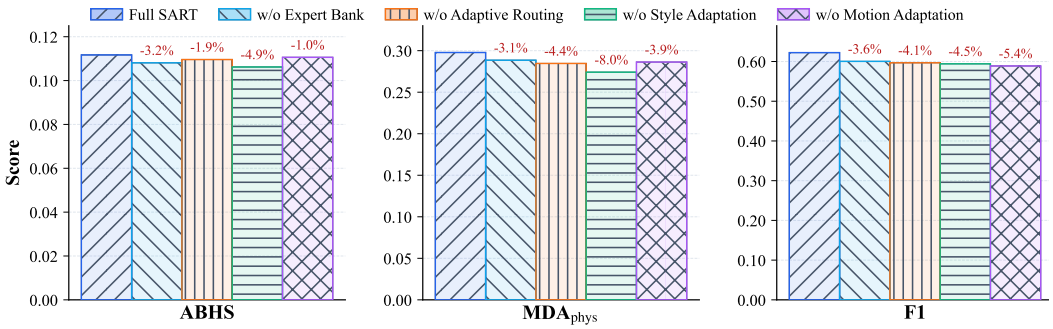


Figure 3: Component-level SART analysis. Bar heights report metric values, while red annotations indicate relative drops from Full SART. The interventions replace the expert bank with a single style-agnostic projection (w/o expert bank), assign equal expert weights (w/o adaptive routing), retain only the learned expert bias (w/o style adaptation), or retain the style-conditioned prototype and expert bias while removing motion-conditioned residuals (w/o motion adaptation).

Adaptation Components. We compare matched-budget one- and two-stage variants to distinguish the contributions of staged adaptation and self-distillation. The one-stage variant uses the full dance–rhythm–text condition from its first update without self-distillation. The two-stage variants instead use only text/lyric conditions during the first 20% of optimization before introducing the full multimodal condition; the SD variant additionally uses the frozen, text-adapted Stage-1 model as its teacher. All three variants use the same total number of optimization steps, rank-64 LoRA configuration, available conditioning inputs, and inference settings. Without SD, moving from one-stage to two-stage adaptation improves F1 from 0.5715 to 0.5955, PQ from 5.7981 to 6.8418, and CLAP from 0.1598 to 0.2567. Adding SD within the two-stage setup further improves these metrics to 0.6224, 7.6686, and 0.2923, respectively. Together with the teacher-consistency analysis in Table 6, these results show that staged adaptation improves generation even without SD, while teacher regularization provides additional gains and constrains text-conditioned prediction drift within the two-stage setup.

Teacher Design. We further isolate the source of the distillation target by comparing the Stage-1 teacher with the base and dual-teacher alternatives. The Stage-1 teacher achieves higher F1, PQ, and CLAP than both alternatives. These results support the stage-wise knowledge-transfer mechanism: the text-adapted teacher provides a task-compatible text-conditioned target, whereas the base teacher is not adapted to the target space and the dual-teacher design introduces competing supervision.

SART Component Analysis. We evaluate SART through nested routing interventions and a simplified representation baseline. The routing variants reuse the same pretrained SART expert checkpoint and fixed, deterministic kinematic input construction while modifying the indicated routing rule. For each variant, we independently retrain the downstream LoRA under the same optimization and inference settings. *w/o adaptive routing* assigns equal weight to every expert; *w/o style adaptation* uses bias-only static routing, omitting both the style-conditioned prototype and motion-conditioned residual; and *w/o motion adaptation* uses static routing determined by the style-conditioned prototype and learned expert bias. These routing variants form a nested sequence rather than independent leave-one-out removals: comparing the two static variants probes the clip-level style-conditioned preference, whereas comparing the style-static variant with Full SART probes frame-level motion residuals. The *w/o adaptive routing* variant provides the closest controlled comparison for adaptive expert selection. The *w/o expert bank* configuration instead replaces the expert bank with a single style-agnostic projection and is therefore interpreted as a simplified representation baseline because it changes both representation and capacity. As shown in Figure 3, all simplified configurations degrade the reported rhythmic-alignment metrics. These results support the contributions of style-conditioned global rhythmic preferences and motion-conditioned local routing within the pretrained SART representation.

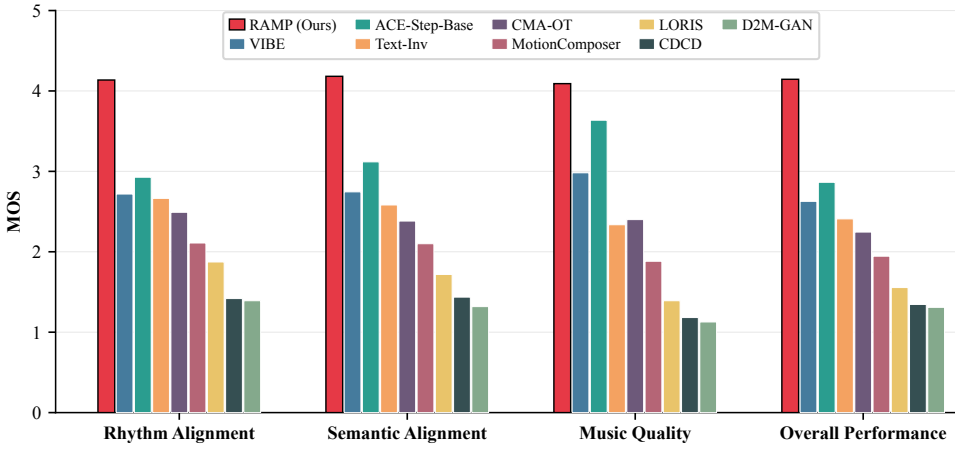


Figure 4: User study results of generated music.

4.2.3 User Study

We conducted a user study to evaluate the perceptual quality of the generated music. We randomly selected 30 dance–music samples, and 25 audio experts independently rated the corresponding generations. To reduce presentation bias, outputs from different methods were presented in randomized order with anonymized method labels. Ratings were collected along four dimensions: (1) *rhythm alignment*, measuring whether musical beats and accents coincide with motion timing; (2) *dance–music semantic alignment*, assessing the consistency of musical style, emotion, and energy with the dance; (3) *music quality*, evaluating naturalness, clarity, coherence, and generation artifacts; and (4) *overall suitability*, reflecting the appropriateness of the generated music as accompaniment for the dance. All dimensions were rated on a five-point Likert scale, with 1 indicating very poor and 5 indicating excellent. As shown in Figure 4, RAMP achieves the highest MOS among the compared methods on all four dimensions. The consistent gains in both alignment and perceptual quality suggest that RAMP generates music that better follows dance motion while maintaining coherent musical semantics and high perceptual quality.

4.3 Case Study

We present two representative case studies to qualitatively examine the behavior of RAMP. As shown in Figure 7, RAMP more closely follows the spectro-temporal patterns of the ground-truth music. Its beat and energy peaks are better aligned with salient motion events, while the resulting spectrogram shows fewer apparent transient misalignments and abrupt spectral discontinuities than the baselines. Figure 8 further demonstrates relative semantic steering under fixed dance and rhythm conditions. When the text prompt is modified while the dance and rhythm conditions remain fixed, the resulting music exhibits clear changes in text-related semantic characteristics, whereas the rhythm-related temporal peaks remain largely unchanged. Together, these cases illustrate RAMP's ability to steer musical semantics while largely retaining dance-driven rhythmic structure.

5 Conclusion

We presented **RAMP**, a framework for introducing structured dance-rhythm control into a pre-trained text-to-music model. Its core mechanism, SART, combines a style-conditioned routing prototype with motion-conditioned residual routing to model clip-level rhythmic preferences and frame-level variation. A two-stage, teacher-regularized adaptation scheme then teaches the generator to use dance-derived conditions while constraining drift from the frozen Stage-1 teacher's text/lyric-conditioned velocity predictions. The backbone's native text and lyric pathways remain explicit semantic controls. We also introduced **Dance100K** and a fixed 3K evaluation benchmark. Experiments and user studies support the effectiveness of the rhythm representation and adaptation designs across rhythmic alignment, semantic consistency, and perceptual quality.

6 AI USE STATEMENT

We used pretrained multimodal AI models during dataset construction: Seed-2.0-Lite to generate video captions, Gemini 2.5 Pro to generate structured music captions and lyric transcripts, Qwen3-VL-8B-Instruct for clip-quality filtering, and a frozen Seed-Lite video annotator to infer weak dance-style tags. These annotations were processed through consistency filtering and human verification as detailed in Appendix B. Generative AI assistants were also used to provide feedback on methodology and experimental design, assist with code development, debugging, and preliminary analysis and interpretation of experimental results, and support manuscript drafting and polishing. The authors reviewed and verified all AI-assisted outputs, made all final methodological and experimental decisions, and take full responsibility for the final manuscript.

ETHICS STATEMENT

Dance100K is constructed from publicly accessible online dance videos that can contain identifiable performers. Public accessibility does not waive the rights or privacy interests of creators, performers, or copyright holders. We therefore plan to release metadata, annotations, data splits, and processing scripts, while redistributing raw media only when permitted by the applicable license, platform policy, or explicit authorization. Our user study involved 25 adult audio experts who participated voluntarily after providing informed consent. We collected only anonymous ratings, collected no personal or sensitive information, and provided no compensation. Further details on privacy, copyright, and data release are provided in Appendix B.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the RAMP architecture, teacher-regularized training objectives, optimization and inference settings, baseline reproduction protocol, Dance100K construction and split procedure, and SART implementation details in the main paper and appendices. We report the training configuration and random seed used for the main model, and provide supplementary demonstrations on the anonymous project page. We plan to release code, configurations, data splits, processing scripts, and releasable metadata and annotations in accordance with the data-release conditions described in Appendix B.

REFERENCES

Gunjan Aggarwal and Devi Parikh. Dance2music: Automatic dance-driven music generation. *arXiv preprint arXiv:2107.06252*, 2021.

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Marco Verzetti, Antoine Caillon, Qing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Caroline Frank. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Lu, Dunjie Lu, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Aryan Vijay Bhosale, Vaibhavi Lokegaonkar, Vishnu Raj, Gouthaman KV, Sreyan Ghosh, Ramani Duraiswami, Lie Lu, and Dinesh Manocha. Vibe: Video instruction-aligned background music generation. *arXiv preprint arXiv:2608.30125*, 2026.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, Gabriel Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Defossez. Simple and controllable music generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 47704–47720, 2023.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025.

Zach Evans, J. D. Parker, C. J. Carr, et al. Stable audio open. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical report, Google DeepMind, July 2025.

Junmin Gong, Sean Zhao, Sen Wang, Shengyuan Xu, and Joe Guo. Ace-step: A step towards music generation foundation model. *arXiv preprint arXiv:2506.00045*, 2025.

Junmin Gong, Yulin Song, Wenxiao Zhao, Sen Wang, Shengyuan Xu, Jing Guo, and Xuerui Yang. Ace-step 1.5: Pushing the boundaries of open-source music generation. *arXiv preprint arXiv:2602.00744*, 2026.

Bo Han, Yuheng Li, Yixuan Shen, Yi Ren, and Feilin Han. Dance2midi: Dance-driven multi-instrument music generation. *Computational Visual Media*, 10(4):791–802, 2024.

Shulei Ji, Songruoyao Wu, Zihao Wang, Shuyu Li, and Kejun Zhang. A comprehensive survey on generative ai for video-to-music generation. *ACM Computing Surveys*, 58(13):1–36, 2026.

Yixiao Jiang, Hao Chen, Zhi Ning, et al. Diffrhythm 2: Efficient and high fidelity song generation via block flow matching. *arXiv preprint arXiv:2510.22950*, 2025.

Sifei Li, Weiming Dong, Yuxin Zhang, Fan Tang, Chongyang Ma, Oliver Deussen, Tong-Yee Lee, and Changsheng Xu. Dance-to-music generation with encoder-based textual inversion. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 135:1–135:11. Association for Computing Machinery, 2024. doi: 10.1145/3680528.3687562.

Xiao Liang, Wensheng Li, Lifeng Huang, and Chengying Gao. Dancecomposer: Dance-to-music generation using a progressive conditional music generator. *IEEE Transactions on Multimedia*, pp. 1–14, 2024. doi: 10.1109/TMM.2024.3405734.

Z. Liu, S. Ding, Z. Zhang, et al. Songgen: A single stage auto-regressive transformer for text-to-song generation. *arXiv preprint arXiv:2502.13128*, 2025.

Zhi Ning, Hao Chen, Yixiao Jiang, et al. Diffrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. *arXiv preprint arXiv:2503.01183*, 2025.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 28492–28518. PMLR, 2023.

Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 29258–29273, 2021.

Changchang Sun, Gaowen Liu, Charles Fleming, and Yan Yan. Enhancing dance-to-music generation via negative conditioning latent diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8321–8330, 2025.

Andros Tjandra, Yi-Chiao Wu, Baishan Guo, John Hoffman, Brian Ellis, Apoorv Vyas, Bowen Shi, Sanyuan Chen, Matt Le, Nick Zacharov, Carleigh Wood, Ann Lee, and Wei-Ning Hsu. Meta audiobox aesthetics: Unified automatic quality assessment for speech, music, and sound. *arXiv preprint arXiv:2502.05139*, 2025. URL `https://arxiv.org/abs/2502.05139`.

Jinting Wang, Chenxing Li, and Li Liu. Gaca-dit: Diffusion-based dance-to-music generation with genre-adaptive rhythm and context-aware alignment. *arXiv preprint arXiv:2510.26818*, 2025a.

Jinting Wang, Li Liu, and Jun Wang. Motioncomposer: Enhancing rhythmic music generation with adaptive retrieval reference. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025b.

Jinting Wang, Chenxing Li, Dong Yu, and Li Liu. Cma-ot: Hierarchical expert supervision for dance-to-music generation. *arXiv preprint arXiv:2609.13118*, 2026.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Yu, Xionghui Xu, Haoran Lin, and C.-C. Jay Kuo. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *International Conference on Machine Learning (ICML)*, 2023.

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, pp. 25615–25623, 2025a.

Kaixing Yang, Xulong Tang, Ziqiao Peng, Yuxuan Hu, Jun He, and Hongyan Liu. Megadance: Mixture-of-experts architecture for genre-aware 3d dance generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 10947–10969. Curran Associates, Inc., 2025b. doi: 10.52202/085713-0366. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/102bd3c1076dfe98043b19340ac44e08-Paper-Conference.pdf.

Xiaoda Yang, Majun Zhang, Changhao Pan, Nick Huang, Yang Yuguang, Fan Zhuo, Pengfei Zhou, Jin Zhou, Sizhe Shan, Shan Yang, Miles Yang, Yang You, and Zhou Zhao. Tmd-bench: A multi-level evaluation paradigm for music-dance co-generation. In *Proceedings of the Forty-Third International Conference on Machine Learning*, 2026.

Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 4210–4220, 2023.

Jixun Yao, Guobin Ma, Huixin Xue, Huakang Chen, Chunbo Hao, Yuepeng Jiang, Haohe Liu, Ruibin Yuan, Jin Xu, Wei Xue, et al. Songeval: A benchmark dataset for song aesthetics evaluation. *arXiv preprint arXiv:2505.10793*, 2025.

Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

R. Yuan, H. Lin, S. Guo, et al. Yue: Scaling open foundation models for long-form music generation. In *International Conference on Learning Representations*, pp. 101737–101778, 2026.

Chaoyang Zhang and Yan Hua. Dance2music-diffusion: Leveraging latent diffusion models for music generation from dance videos. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024(1):48, 2024. doi: 10.1186/s13636-024-00370-6.

Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized gan for complex music generation from dance videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 182–199, 2022a.

Ye Zhu, Yan Wu, Kyle Olszewski, Yu Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. *arXiv preprint arXiv:2206.07771*, 2022b.

A LIMITATIONS

RAMP is evaluated on five-second clips, so its ability to preserve choreography–music coherence over longer sequences remains untested. Because Dance100K-Benchmark is used both for hyperparameter selection and final reporting, future evaluation with a separate validation split or an independent benchmark would provide an additional assessment of generalization. Dance100K is not restricted to Western dance styles, but its Internet-video composition and coarse style vocabulary yield uneven geographic and cultural coverage; regional and traditional forms may be underrepresented. Dance-style tags are weak labels inferred from video-side evidence by a frozen Seed-Lite annotator and filtered through multi-view consistency checks and human auditing. The style-conditioned routing prototypes therefore inherit residual annotation noise and model bias, and should not be interpreted as ground-truth identification of dance styles. Finally, RAMP is built on ACE-Step-1.5. Our

controlled routing ablations evaluate global and local conditioning within a shared pretrained SART representation; further capacity- and pretraining-matched studies would enable finer attribution of gains over alternative rhythm encoders. Its portability to other pretrained text-to-music models also remains unverified. Longer-sequence evaluation, broader cultural coverage, larger-scale expert annotation, label-noise robustness analysis, and cross-backbone validation are important directions for future work.

B Dataset Construction Details

B.1 Data Processing

As illustrated in Figure 2, we construct Dance100K through a multi-stage processing pipeline. We first collect dance-centric videos from social media platforms and divide them into short clips.

Source-item-level split. We define a source item as one collected original video before temporal segmentation. All five-second clips derived from the same source item are assigned to a single partition, preventing adjacent or overlapping segments from appearing across training and benchmark partitions.

Quality filtering. Each candidate clip is then processed by a multi-stage quality- and content-control pipeline. We use Whisper-based speech detection Radford et al. (2023) to remove clips containing substantial dialogue or commentary, and employ PaddleOCR Cui et al. (2025) to detect persistent subtitles and overlaid text. Clips dominated by subtitles are rejected or processed with subtitle removal, while small logos and natural scene text may be retained.

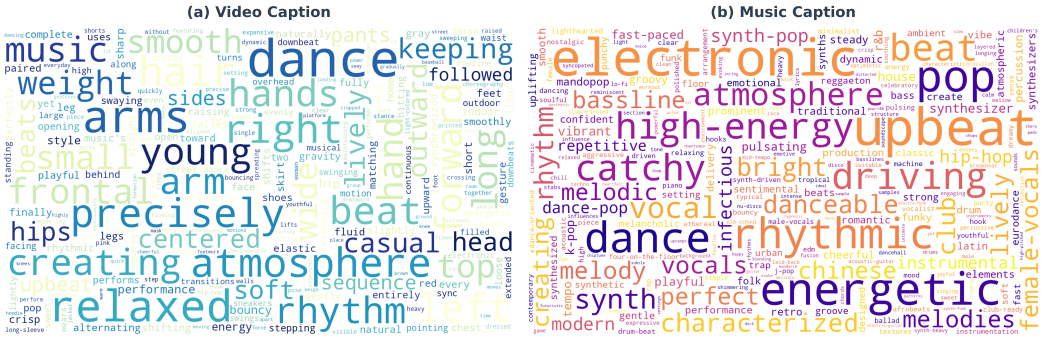


Figure 5: Word clouds of the generated captions. (a) Video-caption vocabulary describing dance movements and visual content. (b) Music-caption vocabulary describing genre, rhythm, mood, and instrumentation. Word size indicates frequency across the dataset.

We further use a vision-language model (Qwen3-VL-8B-Instruct Bai et al. (2025)) to assess whether each clip contains meaningful dance motion, a clearly visible dancer, sufficient motion richness, and acceptable temporal consistency. Clips containing non-dance activities, fitness or aerobic exercises, speech-dominated content, or severe visual degradation are discarded. The retained clips are standardized to approximately 5 seconds at 25 FPS, with their corresponding audio tracks extracted separately.

For the retained samples, we construct multimodal semantic and motion annotations. Specifically, we use **Seed-2.0-Lite**[1] to generate detailed video captions and **Gemini 2.5 Pro** to generate music captions covering fine-grained musical semantics. We further employ **DWPose** Yang et al. (2023) to extract human pose sequences as an explicit representation of dance motion. Finally, **human verification** is conducted to assess sample quality and cross-modal consistency. The resulting dataset consists of temporally aligned dance video, pose sequence, music waveform, video caption, and music caption.

[1] https://seed.bytedance.com/en/seed2

B.2 AUDIO CAPTION GENERATION

We use Gemini 2.5 Pro Gemini Team (2025) to generate structured audio captions that capture both lyrical content and perceptible musical characteristics. The model is instructed to act as a multimodal audio-analysis expert and produce three fields:

- `transcript`: the detected lyrics, or an empty string for instrumental music;
- `music_style`: multiple concise phrases describing the musical style; and
- `music_description`: a sentence summarizing the overall musical characteristics.

The core audio prompt is:

```
Analyze the audio and identify its lyrics and musical style.  If
lyrics are present, transcribe them; otherwise, return an empty
transcript.  Provide multiple short phrases describing the music
style and a sentence describing its overall musical characteristics.
Return only a JSON object containing transcript, music_style, and
music_description.
```

The structured JSON output enables reliable automatic parsing and provides consistent semantic annotations for subsequent analysis of music styles, lyrical content, and vocal characteristics.

B.3 VIDEO CAPTION GENERATION

Video captions are designed to provide fine-grained semantic annotations of dance content. The model is instructed to describe:

- the number, position, orientation, and interactions of dancers;
- observable appearance and clothing;
- dance style and characteristic movements;
- movements of the arms, legs, hips, head, and torso;
- motion speed, rhythm, strength, and movement quality;
- synchronization between movement and music;
- spatial movement, weight transfer, and formation changes;
- shot scale, camera distance, body framing, and camera motion; and
- background, lighting, atmosphere, and performance intensity.

The essential video prompt is:

```
Act as a professional dance-video and choreography analyst.  Generate
a detailed caption for text-to-dance learning using only directly
observable information.  Describe the dancers, dance style, major
body movements, characteristic actions, motion rhythm and quality,
synchronization with music, and spatial changes.  Explicitly
distinguish shot scale, camera distance, body framing, camera motion,
and camera viewpoint.  Briefly describe the environment and lighting.
Output one fluent paragraph of five to seven sentences without
unsupported inference or subjective judgment.
```

B.4 DATA SOURCES AND COVERAGE

Our approximately 100K music–dance pairs are collected from publicly available online platforms, such as YouTube and similar video-sharing websites, covering diverse dance styles, performer settings, and music genres. In addition, 66.66% of the music samples contain lyrics, while the remaining 33.34% are instrumental. The videos exhibit substantial variation in performers, camera viewpoints, environments, and lighting conditions, providing broad multimodal coverage for studying the relationships between dance motion, music, and semantics.

B.5 Ethics, Privacy, and Data Release

Dance100K is constructed from publicly accessible online dance videos for research purposes. Public accessibility does not imply unrestricted ownership: the videos, performances, and accompanying music remain subject to the rights of their original creators, performers, and copyright holders. We plan to release metadata, annotations, data splits, and data-processing scripts to facilitate reproducible research. Raw video and audio will be redistributed only when permitted by the applicable license, platform policy, or explicit authorization; otherwise, where permitted, we will provide source identifiers or links together with derived annotations.

B.6 Caption vocabulary analysis

Figure 5 presents word clouds constructed from the video and music captions of approximately 100K samples. For video captions, we remove common stop words and template-related filming terms, such as 'camera," 'frame," and "shot," to reduce captioning artifacts. For music captions, we retain only music-style keywords and semantic descriptions, excluding lyric transcripts. Multi-word expressions are normalized before frequency counting, and word size is proportional to corpus-level frequency. The video captions predominantly describe rhythmic characteristics and body movements, whereas the music captions are dominated by terms related to energetic and upbeat qualities, as well as pop and electronic styles.

C Additional Experimental Results

C.1 Comparison with SOTA Methods

Baseline reproduction protocol. We retrain D2M-GAN Zhu et al. (2022a), CDCD Zhu et al. (2022b), LORIS Yu et al. (2023), Text-Inv Li et al. (2024), MotionComposer Wang et al. (2025b), CMA-OT Wang et al. (2026), and VIBE Bhosale et al. (2026) on the Dance100K training split using their released official implementations. We retain each method's officially supported input modalities, preprocessing, architecture, optimization hyperparameters, and inference configuration. Text and lyric conditions are used when supported by the official interface and omitted otherwise. All methods are evaluated on the same fixed Dance100K-Benchmark split using the common evaluation pipeline described in the main text.

Interpretation and fairness. We distinguish system-level comparison from component-level attribution. Cross-method results evaluate complete systems under their natively supported modalities and official operating configurations; architectures, parameter counts, and samplers are therefore not strictly matched. The shared training split, benchmark set, output duration, and evaluation pipeline support an end-to-end comparison, but cross-method CLAP, L-WER, and quality differences should not be attributed solely to RAMP's architecture. In particular, L-WER values for systems without native lyric conditioning are diagnostic rather than comparisons of lyric controllability. We assess the contributions of rhythm representation, Teacher Consistency, and staged adaptation through controlled ablations sharing the same ACE-Step-1.5 Base initialization, rank-64 LoRA configuration, conditioning inputs, total optimization budget, and inference settings. The additional results in Table 3 further support the effectiveness of RAMP. For complementary rhythmic alignment, MotionComposer achieves the highest BHS, RAMP achieves the highest VBCS, and CMA-OT achieves the highest BCS; RAMP remains competitive on BCS, differing from CMA-OT by only 0.0016. More importantly, RAMP achieves the strongest audio-aesthetics scores (CE, CU, and PC) and the highest reported SongEval proxy scores (Memorability, Clarity, and Naturalness), yielding favorable performance across the reported quality-related metrics.

C.2 Detailed Ablation Results

The complementary results in Table 4 further support the conclusions drawn from the main ablation study. SART consistently provides stronger rhythmic alignment and audio quality than alternative rhythm representations. Under the matched training budget, two-stage adaptation without SD improves over one-stage adaptation without SD across rhythmic, aesthetic, and SongEval metrics;

Table 3: Additional evaluation results on complementary rhythmic alignment, audio aesthetics, and automatic SongEval proxy scores.

	Rhythmic Alignment			Aesthetics Score			SongEval		
Methods	**VBCS ↑**	**BHS ↑**	**BCS ↑**	**CE ↑**	**CU ↑**	**PC ↑**	**Mem. ↑**	**Clar. ↑**	**Nat. ↑**
D2M-GAN	0.2922	0.3037	0.5419	2.7550	6.0690	2.3587	1.7720	1.8329	1.6859
CDCD	0.1702	0.3424	0.4895	3.4616	6.2949	2.7226	1.7946	1.6439	1.6903
LORIS	0.3766	0.5144	0.5332	3.4544	5.1483	5.1558	2.4322	2.3015	2.3780
Text-Inv	0.4357	0.6592	0.5694	5.2439	6.5559	6.2492	2.3539	2.3239	2.3513
MotionComposer	0.4537	**0.7736**	0.5108	1.9469	3.1934	2.5887	1.8183	1.6784	1.7078
CMA-OT	0.4564	0.5464	**0.5977**	5.2804	6.2788	6.3579	2.5447	2.4549	2.3155
VIBE	0.4529	0.6114	0.5952	5.8501	7.1108	5.3690	2.3625	2.2318	2.2645
RAMP (Ours)	**0.4839**	0.6884	0.5961	**6.8843**	**7.6375**	**6.5430**	**2.9647**	**2.8909**	**2.9377**

Table 4: Complementary ablation results on rhythmic, aesthetic, and automatic SongEval proxy metrics. The Adaptation Components block uses the same matched-budget configurations as Table 2.

	Rhythmic Alignment			Aesthetics Score			SongEval			
Methods	**VBCS ↑**	**BHS ↑**	**BCS ↑**	**CE ↑**	**CU ↑**	**PC ↑**	**Coh. ↑**	**Mem. ↑**	**Clar. ↑**	**Nat. ↑**
Rhythm Representation										
ST-GCN	0.4611	0.6216	0.5595	5.0036	5.7568	6.2796	2.3797	2.3433	2.2530	2.3463
Binary Rhythm	0.4414	0.6840	0.5411	4.9609	5.2891	6.4792	2.5439	2.5095	2.3771	2.4929
SART	**0.4839**	**0.6884**	**0.5961**	**6.8843**	**7.6375**	**6.5430**	**2.9947**	**2.9647**	**2.8909**	**2.9377**
Adaptation Components										
ACE-Step	0.3434	0.5079	0.5484	5.6644	7.3595	3.3084	2.6173	2.7618	2.4990	2.5992
One-stage w/o SD	0.4248	0.6438	0.5359	4.2445	4.9448	5.7533	2.4452	2.4110	2.3223	2.3898
Two-stage w/o SD	0.4836	0.6548	0.5703	5.9404	6.5370	6.3866	2.6625	2.6088	2.4774	2.5711
Two-stage + SD	**0.4839**	**0.6884**	**0.5961**	**6.8843**	**7.6375**	**6.5430**	**2.9947**	**2.9647**	**2.8909**	**2.9377**
Teacher Design										
Base Teacher	0.4750	0.6222	0.5626	5.4058	6.0145	6.2008	2.5559	2.5334	2.4223	2.5156
Dual Teacher	0.4736	0.6325	0.5656	5.6812	6.2005	6.4334	2.5591	2.5216	2.3914	2.4741
Stage-1 Teacher	**0.4839**	**0.6884**	**0.5961**	**6.8843**	**7.6375**	**6.5430**	**2.9947**	**2.9647**	**2.8909**	**2.9377**

adding SD within the two-stage setup provides further gains. Among the teacher designs, the Stage-1 teacher achieves the most favorable overall performance, supporting the use of the text-adapted model as a behavioral reference during dance-conditioned adaptation.

Sensitivity Analysis. We investigate the sensitivity of RAMP to the self-distillation weight λ_{SD} and the Stage-1 training ratio, with results shown in Figure 6. For λ_{SD}, both rhythm and semantic alignment vary non-monotonically, with $\lambda_{\text{SD}} = 0.15$ achieving the best overall performance, obtaining an F1 score of 0.6224, ABHS of 0.1117, and CLAP of 0.2923. A smaller weight ($\lambda_{\text{SD}} = 0.05$) provides relatively strong semantic alignment but weaker rhythmic alignment, whereas increasing the weight beyond 0.15 does not yield further improvements. This suggests that moderate self-distillation provides a useful balance between adapting to dance-conditioned rhythm and stabilizing the text-conditioned music-generation behavior inherited from the Stage-1 teacher during multi-modal adaptation.

We further examine the Stage-1 training ratio within the two-stage schedule. A ratio of 0.2 provides the best overall trade-off, achieving the highest F1 and CLAP while remaining close to the best ABHS, which occurs at 0.1. Allocating either less or more training to Stage 1 reduces F1 and CLAP. This indicates that the two-stage curriculum benefits from an appropriate allocation: Stage 1 receives sufficient optimization to establish text-conditioned musical semantics, while the majority

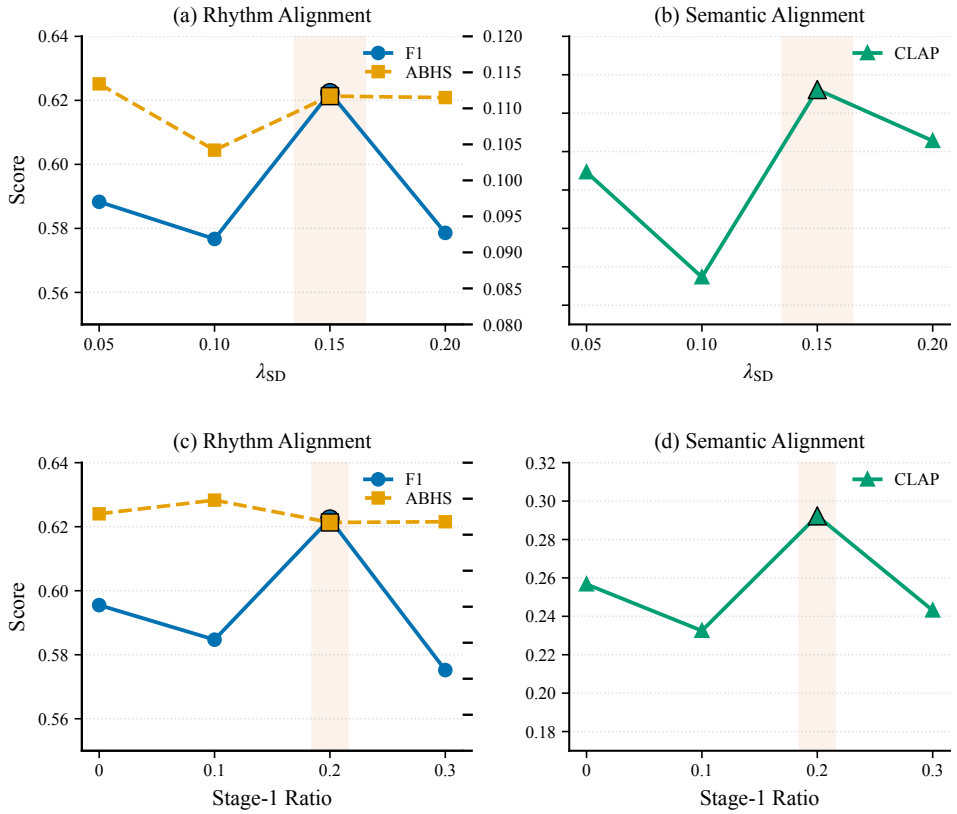


Figure 6: Sensitivity analysis of RAMP. We vary the self-distillation weight λ_{SD} (top) and the Stage-1 training ratio (bottom), and evaluate rhythm alignment (F1 and ABHS) and semantic alignment (CLAP). The shaded regions mark the settings selected using Dance100K-Benchmark, $\lambda_{\text{SD}} = 0.15$ and a Stage-1 training ratio of 0.2. The same benchmark is used for final reporting; no separate validation split is used for this selection.

of training remains available for Stage 2 to introduce dance-specific rhythmic adaptation without excessively disrupting the Stage-1 text-conditioned behavior. Based on this balance, we use $\lambda_{\text{SD}} = 0.15$ and a Stage-1 training ratio of 0.2 throughout the remaining experiments.

C.3 Qualitative Comparison

Figure 7 provides a qualitative comparison of the generated music using Log-Mel spectrograms. The ground-truth music exhibits temporally varying harmonic structures and broadband transients that often coincide with the extracted motion beats. RAMP exhibits similar time–frequency continuity, with rhythmic transients and harmonic energy evolving coherently throughout the sequence. In contrast, several baselines show more stationary spectral patterns, limited frequency activity, abrupt energy variations, or low-energy intervals, suggesting weaker temporal adaptation and less consistent spectral dynamics. Overall, the visualization highlights the ability of RAMP to maintain temporally coherent rhythmic and acoustic structures over the generated sequence.

C.4 Text-Guided Semantic Steering under Fixed Rhythm

We examine to what extent text can steer musical semantics while the motion-derived rhythm conditions remain fixed. As shown in Figure 8, we generate paired samples while keeping the dance, rhythm tokens, diffusion seed, original lyrics condition, and inference settings fixed, varying only the music-style component of the text prompt. Specifically, we change *Eurodance* to *Funk* in the first example and *Pop dance* to *Rock* in the second, while preserving the same lyrics within each pair. The resulting spectrograms show clear differences in spectral envelope, harmonic structure,

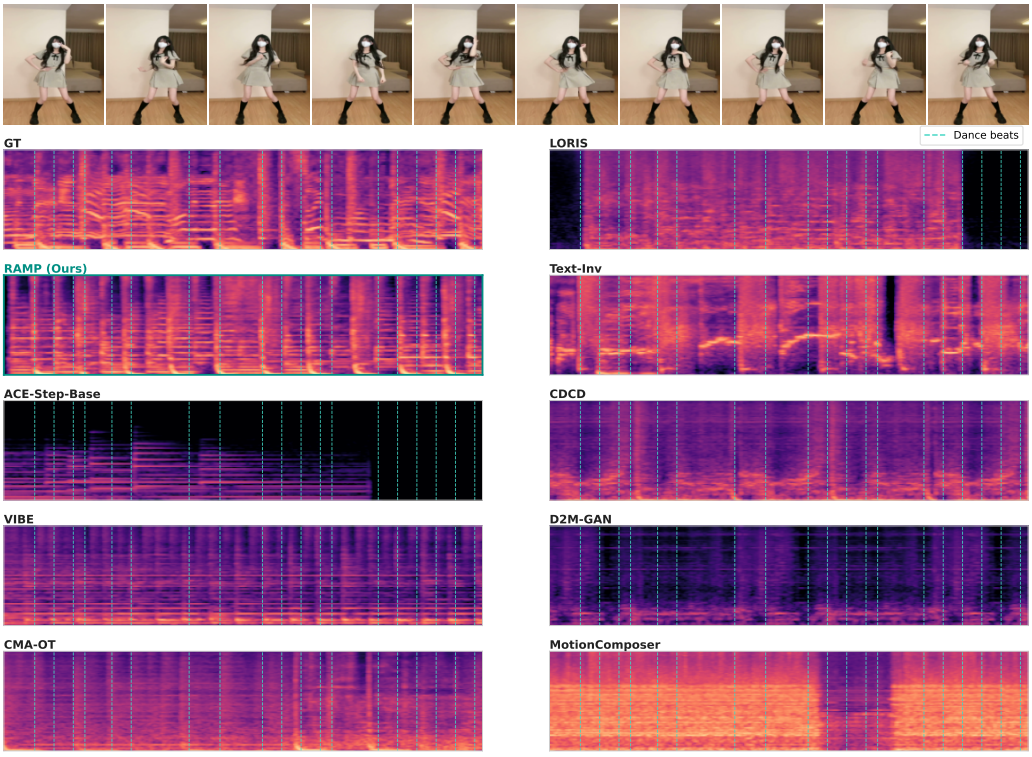


Figure 7: Qualitative comparison of music generation results. The top row shows uniformly sampled frames from the input dance video. The bottom panels present the Log-Mel spectrograms of the ground-truth music and generation results.

and frequency-energy distribution while maintaining similar temporal organization within each pair. This controlled comparison provides qualitative evidence that RAMP can modify high-level musical semantics while retaining dance-conditioned rhythmic structure.

For the quantitative intervention metrics, let $c(a, p)$ denote CLAP similarity between audio a and prompt p, and let (a_o, a_m) and (p_o, p_m) denote the audio and prompt before and after the intervention, respectively. Target CLAP is $c(a, p_m)$. We define

$$\begin{aligned} A_{\text{target}} &= c(a_m, p_m) - c(a_m, p_o), \\ \Delta_{\text{pref}} &= \big[c(a_m, p_m) - c(a_m, p_o)\big] - \big[c(a_o, p_m) - c(a_o, p_o)\big]. \end{aligned} \tag{10}$$

Here A_{target} is the modified audio's target advantage, while Δ_{pref} measures how much the intervention shifts relative prompt preference toward the modified target.

Table 5: Prompt intervention on 50 paired samples with fixed dance and rhythm conditions. Changing the text prompt largely preserves rhythmic alignment while steering semantic alignment. Δ denotes Modified $-$ Original.

	Rhythm		Change
Metric	**Original**	**Modified**	Δ
BCS ↑	0.5772	0.5849	+0.0077
BHS ↑	0.6114	0.6147	+0.0033
F1 ↑	0.5809	0.5858	+0.0049
Semantic response			
Target CLAP ↑	0.1916	0.2350	**+0.0434**
CLAP preference shift ↑		–	**+0.0499**

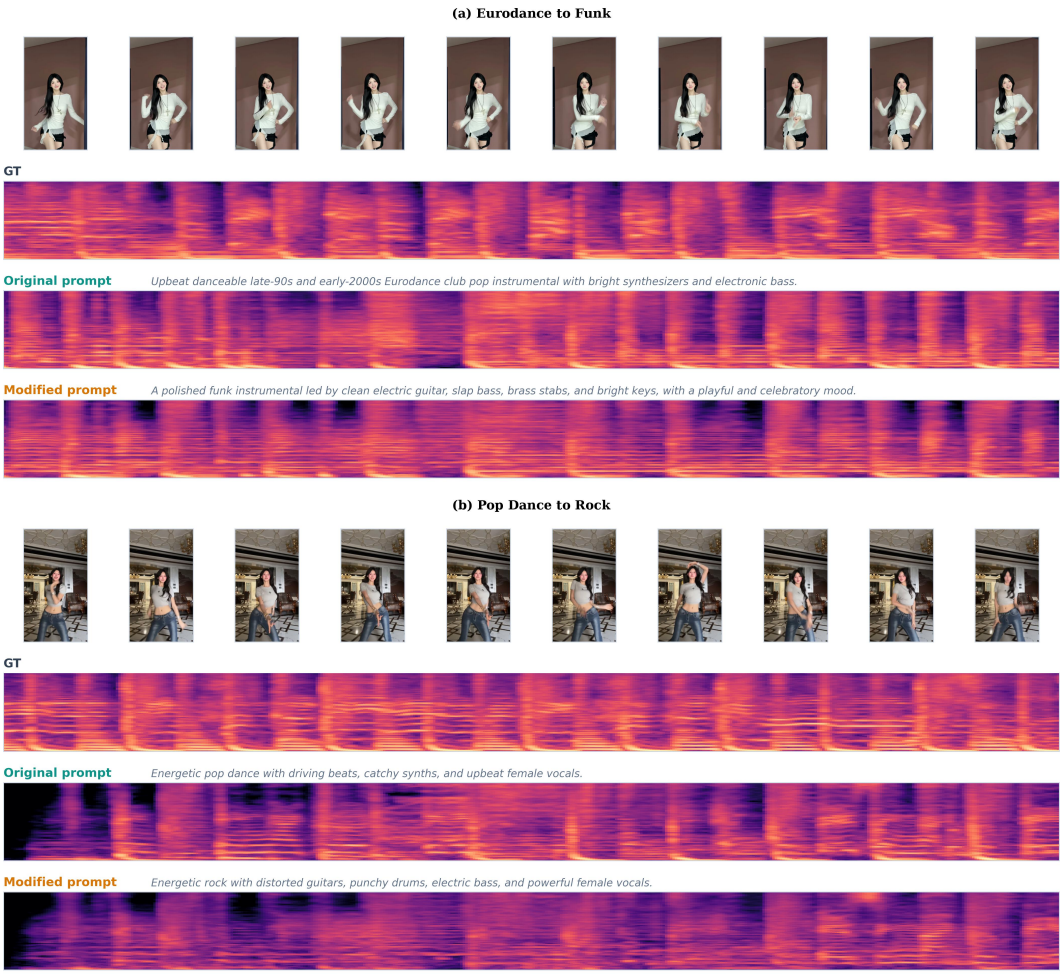


Figure 8: Qualitative semantic steering under fixed dance–rhythm conditions. We present two examples: (a) Eurodance to Funk and (b) Pop dance to Rock. Each panel shows sampled dance frames followed by log-Mel spectrograms of the ground-truth audio and generations conditioned on the original and modified prompts.

We further evaluate this behavior quantitatively using 50 paired interventions, with the dance input, rhythm tokens, lyrics, diffusion seed, and inference settings fixed. We report 95% percentile-bootstrap confidence intervals from 10,000 resamples of the paired examples. As shown in Table 5, changing the style prompt produces no systematic degradation in beat alignment: the paired changes in BCS, BHS, and Beat F1 are $+0.0077$ (CI $[-0.0087, 0.0246]$), $+0.0033$ (CI $[-0.0299, 0.0335]$), and $+0.0049$ (CI $[-0.0173, 0.0257]$), respectively. In contrast, CLAP similarity to the target-style prompt increases by 0.0434 (CI $[0.0209, 0.0695]$), and the cross-prompt preference shifts toward the target by 0.0499 (CI $[0.0230, 0.0846]$). The modified audio nevertheless remains, on average, more similar to the original prompt than to the target prompt, with a target advantage of -0.0513 (CI $[-0.0852, -0.0155]$). These results therefore support relative semantic steering with preserved rhythmic organization, rather than complete separation or full style reversal.

C.5 Teacher Consistency Analysis

We evaluate text/lyric-only inference on the full benchmark after Stage 2. We also measure the text/lyric-only velocity-field discrepancy to the frozen Stage-1 teacher for Stage 2 models trained with and without self-distillation (SD). Table 6 jointly reports semantic alignment, audio quality,

Table 6: Text/lyric-only generation and teacher–student velocity-field discrepancy. PQ is the AudioBox production-quality score; Musicality and Naturalness are automatic proxy scores.

Model	CLAP ↑	PQ ↑	Mus. ↑	Nat. ↑	Discrep. ↓
Two-stage w/o SD	**0.2887**	7.6047	2.8431	2.8083	0.1676
Two-stage + SD	0.2862	**7.6126**	**2.8689**	**2.8349**	**0.0446**

automatic proxy scores, and teacher–student discrepancy. CLAP is computed on 2,997 valid samples; all other columns use 2,999 samples.

Let $\mathcal{M}_i$ be the valid audio-latent positions selected by the attention mask for sample i, and let D_v be the velocity-channel dimension. For a Stage-2 model m, write $\mathbf{v}_i^{(m)}$ and $\mathbf{v}_i^{(\mathrm{T})}$ for the student and teacher predictions under matched inputs, and $e_{i,\ell,d}^{(m)} = v_{i,\ell,d}^{(m)} - v_{i,\ell,d}^{(\mathrm{T})}$ for their elementwise difference. We compute

$$d_i^{(m)} = \frac{1}{|\mathcal{M}_i| D_v} \sum_{\ell \in \mathcal{M}_i} \sum_{d=1}^{D_v} \left(e_{i,\ell,d}^{(m)} \right)^2,$$
$$\text{Discrep.}^{(m)} = \frac{1}{N} \sum_{i=1}^{N} d_i^{(m)}. \tag{11}$$

Student and teacher use the same noisy latent, flow timestep, text/lyric-only condition, context latent, and attention mask. The discrepancy is an elementwise MSE over valid positions and channels, computed in float32; beyond division by the number of valid elements, it is not normalized by the teacher norm or variance.

Self-distillation reduces the mean velocity-field discrepancy from 0.1676 to 0.0446, a 73.4% relative reduction. The paired difference is -0.1230, with a 95% bootstrap confidence interval of $[-0.1268, -0.1193]$, and discrepancy is lower for 99.97% of the 2,999 samples. Meanwhile, text/lyric-only generation metrics remain comparable: CLAP changes from 0.2887 to 0.2862, while PQ, Musicality, and Naturalness improve slightly. These results show that SD strongly constrains drift in the local text/lyric-only velocity field without materially degrading the measured generation metrics. We therefore interpret SD as teacher-consistency regularization that stabilizes the text-conditioned music-generation behavior inherited from the Stage-1 teacher during multimodal adaptation.

C.6 Routing Prototype Analysis

Table 7 compares the default style-conditioned routing prototype with a learned routing-prototype variant on the full benchmark. The learned-prototype variant is initialized from the smoothed routing prototype with a zero-initialized residual output head and is regularized by $\mathcal{L}_{\text{sem}}$ with $\lambda_{\text{sem}} = 0.03$. Its residual prior parameters are optimized jointly with the SART experts and router only during tokenizer pretraining; the resulting SART checkpoint is then frozen during downstream generator adaptation and inference. The learned routing prototype increases MDA_{phys} from 0.2978 to 0.3158, indicating stronger motion-conditioned alignment on this metric. However, the frozen style-conditioned routing prototype performs better on ABHS and beat F1 and substantially outperforms the learned variant in semantic alignment and perceptual quality, as reflected by CLAP, PQ, Musicality, and L-WER. We therefore use the frozen style-conditioned routing prototype as the default because it provides a more balanced rhythm–semantics–quality trade-off rather than optimizing a single alignment metric. This comparison is between the corresponding fully trained variants and thus characterizes their end-to-end trade-off.

Why learning can hurt. A plausible explanation is optimization coupling during SART pretraining rather than insufficient capacity. Once the prototype is learnable, it can co-adapt with the expert bias, FiLM modulation, and motion-dependent residual router under the tokenizer pretraining objectives. This added freedom may let the prototype absorb training-set shortcuts or correlations in noisy and imbalanced style labels, weakening its role as a stable global routing anchor. Freezing the corpus-derived prototype instead constrains global style preference and leaves frame-level adap-

Table 7: Full-benchmark comparison of the frozen style-conditioned routing prototype and learned routing prototype. Musicality is an automatic proxy score. Best values are bolded.

Routing prototype	ABHS ↑	MDA_{phys} ↑	F1 ↑	PQ ↑	Mus. ↑	L-WER ↓	CLAP ↑
Frozen prototype	**0.1117**	0.2978	**0.6224**	**7.6686**	**2.9804**	**1.0543**	**0.2923**
Learned prototype	0.1098	**0.3158**	0.6059	7.4368	2.6823	1.2431	0.2068

tation to the motion residuals, yielding a better-regularized global-to-local decomposition. These mechanisms are consistent with the observed trade-off but do not establish that learned prototypes are universally inferior; stronger regularization or larger, cleaner style annotations may reduce the gap.

D Additional Details of Two-Stage Adaptation

D.1 Stage Transition and Trainable Adapters

Stage 2 continues from the same student optimized in Stage 1; the student LoRA is neither reinitialized nor replaced at the stage boundary. Let θ_1 denote the student parameters immediately after Stage 1. The first Stage 2 update therefore starts from $\theta \leftarrow \theta_1$. When the Stage-1 teacher is enabled, a frozen copy of this adapter is captured immediately before that first Stage 2 update. Subsequent Stage 2 optimization changes only the active student copy, while the captured teacher remains fixed.

The trainable student adapter is applied to the q, k, v, and output projections of both self- and cross-attention, with LoRA rank 64, scale $\alpha_{\text{LoRA}} = 128$, and dropout 0.05. The original ACE-Step decoder weights remain frozen. The full model enables separate dance and rhythm conditioning projectors and type embeddings as trainable conditioning modules; they are not LoRA parameters. Both projectors are initialized to the identity. Because Stage 1 masks all non-text positions, these motion-conditioning modules become effective during Stage 2. We linearly warm the dance and rhythm conditions from scale 0.25 to 1.0 over the first 10% of Stage 2. During Stage 2, joint-motion, dance-only, and rhythm-only dropout probabilities are 0.10, 0.05, and 0.10, respectively. We clip the global gradient norm to 1.0 and use training seed 42.

Optimization and inference details. Stage 1 occupies the first 20% of optimization steps at 5×10^{-5}, and Stage 2 uses 10^{-4}. We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$), weight decay 0.01, a 100-step Stage-1 warmup, and a 200-step stage-transition warmup followed by cosine decay to 10% of the Stage-2 peak rate. The non-text prefix contains 512 dance tokens and 128 rhythm tokens; the pretrained SART checkpoint remains frozen. Modality-type embeddings use scale 0.02, and per-sample norm alignment matches the mean ℓ_2 norms of dance and rhythm tokens to that of the ACE text–lyrics condition. At inference, we use 50-step explicit Euler integration on a linear flow-time grid from $t = 1$ to $t = 0$ with timestep shift 1.0. Adaptive Projected Guidance is applied to the conditional–unconditional velocity difference with scale 7.0, while the text, dance, and rhythm modality scales are each 1.0.

The three relevant models differ only in adapter state:

- **Student:** the active trainable LoRA, initialized by and continuously updated from Stage 1;
- **Base teacher:** the frozen original ACE-Step decoder with all LoRA adapters disabled;
- **Stage-1 teacher:** a frozen LoRA snapshot θ_1 captured at the Stage-1/Stage-2 boundary.

Dual Teacher uses both frozen teacher predictions; it does not introduce an additional trainable LoRA.

D.2 Exact Teacher Objectives

Let $\mathbf{x}_0$ be the target audio latent, $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$, and $\mathbf{v}^\star = \mathbf{x}_1 - \mathbf{x}_0$ the flow-matching target. Denote the full dance–rhythm–text condition by $\mathbf{C}_{\text{full}}$ and the layout-preserving text/lyric-only condition, obtained by zeroing and masking the 640 non-text prefix positions, by

$\mathbf{C}_{\text{text}}$. Stage 1 optimizes

$$\mathcal{L}_{\text{S1}} = \text{MSE}(v_\theta(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}), \mathbf{v}^\star)\,. \tag{12}$$

In Stage 2, the primary dance-to-music objective is

$$\mathcal{L}_{\text{D2M}} = \text{MSE}(v_\theta(\mathbf{x}_t, t; \mathbf{C}_{\text{full}}), \mathbf{v}^\star)\,. \tag{13}$$

The Stage-2 student is evaluated twice for each training sample: $\mathcal{L}_{\text{D2M}}$ uses the full multimodal condition, while the teacher-consistency branch evaluates both the current student and the frozen teacher on the same $\mathbf{x}_t$, timestep, context latent, and text/lyric-only condition. The teacher never receives dance or rhythm tokens. For Base Teacher, the objective used in the teacher-design ablation is

$$\mathcal{L}_{\text{Base}} = \mathcal{L}_{\text{D2M}} + 0.15\ \text{MSE}(v_\theta(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}), v_0(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}))\,, \tag{14}$$

where v_0 is the original base decoder with adapters disabled. For Dual Teacher, the exact expanded objective is

$$\begin{aligned}\mathcal{L}_{\text{Dual}} = \mathcal{L}_{\text{D2M}} &\\ &+ 0.10\ \text{MSE}(v_\theta(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}), v_0(\mathbf{x}_t, t; \mathbf{C}_{\text{text}})) \\ &+ 0.05\ \text{MSE}(v_\theta(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}), v_{\theta_1}(\mathbf{x}_t, t; \mathbf{C}_{\text{text}}))\,.\end{aligned} \tag{15}$$

The implementation computes a normalized weighted average of the two Dual Teacher errors and then multiplies it by $0.10 + 0.05 = 0.15$; after expansion, the absolute coefficients are therefore exactly 0.10 and 0.05, rather than 0.15 for each teacher. The Stage-1 Teacher variant used by the full model replaces the base target with v_{θ_1} and uses weight 0.15. In all variants, teacher outputs are stop-gradient targets and the coefficient of $\mathcal{L}_{\text{D2M}}$ is one.

E Additional Details of SART

This section specifies the implemented SART tokenizer, including the exact expert views, routing, architecture, pretraining losses, and downstream token layout. Unless otherwise stated, the values below refer to the released large checkpoint at epoch 30.

E.1 Expert-Specific Rhythm Views

Given a dance pose sequence $\mathbf{P} \in \mathbb{R}^{T_0 \times D_p}$, we extract motion energy e_t, acceleration magnitude a_t, a binary motion-peak indicator p_t, and peak-relative phase ϕ_t. After rounded-index temporal resampling to $T = 128$, the explicit rhythm representation is

$$\mathbf{x}_t = [e_t, a_t, p_t, \rho_t^{\sin}, \rho_t^{\cos}] \in \mathbb{R}^5, \qquad \rho_t^{\sin} = \sin(2\pi\phi_t), \quad \rho_t^{\cos} = \cos(2\pi\phi_t). \tag{16}$$

Stacking the framewise vectors gives $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{T\times 5}$. We use a peak threshold of 0.2 and a minimum peak distance of 6 frames. These are global preprocessing constants applied uniformly to every style and data split, rather than style-specific parameters. Let $\Delta u_t = u_t - u_{t-1}$ with $\Delta u_1 = 0$, let $M_w(u)$ denote a centered moving average of width w, let $\sigma(\cdot)$ be the logistic sigmoid, and let $N(\cdot)$ denote channel-wise z-score normalization over the flattened batch and time axes. The four implemented expert inputs are

$$\begin{aligned}\text{View}_1(\mathbf{X}) &= N([a, p, |\Delta e|, |\Delta a|, M_3(e)])\,,\\ \text{View}_2(\mathbf{X}) &= N\big([M_7(e), M_7(a), \rho^{\sin}, \rho^{\cos}, M_9(p)]\big)\,,\\ \text{View}_3(\mathbf{X}) &= N([\sigma(-e), M_9(\sigma(-e)), \sigma(-e_{t-1})p_t, -e, a])\,,\\ \text{View}_4(\mathbf{X}) &= N\big([M_{15}(e), M_{15}(a), \Delta M_{15}(e), M_{15}(\rho^{\sin}), M_{15}(\rho^{\cos})]\big)\,.\end{aligned} \tag{17}$$

For the first frame, the third component of View_3 is set to zero. These fixed, style-agnostic analytic views are applied identically to all clips and expose cues associated with sharp hits, periodic grooves, freeze/hold intervals, and smooth phrases. They provide an interpretable architectural bias but neither encode style labels nor determine expert usage, which is learned through the global-to-local router. We use $K = 4$ as a compact basis spanning four complementary temporal regimes—transient impacts, periodic grooves, sustained stillness, and long-range phrases—rather than claiming that these views are the unique or identifiable factors of dance rhythm. Each independently parameterized expert produces

$$\mathbf{Z}^{(k)} = E_k(\text{View}_k(\mathbf{X})), \qquad k \in \{1, 2, 3, 4\}. \tag{18}$$

Architecture and parameter count. For the large checkpoint, each expert uses three stride-one temporal convolutions, $5 \to 256$ with kernel 3 followed by two $256 \to 256$ layers with kernel 5, and a kernel-1 projection $256 \to 512$. All dilations are one and same padding preserves all 128 time steps; consequently, SART performs no internal token downsampling, and each expert has an 11-frame receptive field. GELU and dropout (0.05) follow each hidden convolution. The router uses a $5 \to 256$ kernel-11 convolution, style FiLM, and a $256 \to 4$ kernel-11 convolution, giving a 21-frame receptive field. The reconstruction MLP is $512 \to 256 \to 256 \to 256 \to 5$. The serialized pretrained module has $3{,}458{,}573$ parameters: $3{,}166{,}208$ in the four experts, $28{,}168$ in the router, and $264{,}197$ in the reconstruction head. The expert–router encoder accounts for $3{,}194{,}376$ parameters; the reconstruction head is retained in the checkpoint but serves only the pretraining objective.

Semantic anchoring. The expert-slot labels serve as architectural semantic anchors induced jointly by the fixed expert-specific views in Eq. (17), the style-conditioned routing prototype, and the routing-prior loss. Figure 9 shows differentiated style-dependent expert utilization, while these labels are not intended as claims of uniquely identifiable latent factors.

E.2 STYLE-CONDITIONED ROUTING PROTOTYPE

Dance-style tag construction. We treat the dance-style tag s as weak video-side metadata rather than a uniquely identifiable ground-truth category. Before prototype estimation, we fix a coarse vocabulary using only the training split: {*breaking*, *popping*, *locking*, *waacking*, *house*, *krump*, *hip-hop*, *jazz*, *contemporary*, *ballet*}, together with the special outcomes *other* and *uncertain*. Reliable choreography metadata is used when available and is converted by a fixed alias table; for example, `br`, `po`, `lo`, `wa`, `ho`, and `kr` map to breaking, popping, locking, waacking, house, and krump, while `mh`/`lh` and `js`/`jb` map to hip-hop and jazz, respectively. Dataset identity is never used as a default style label.

For clips without reliable metadata, we use a frozen Seed-Lite video annotator with the audio removed. DWPose detections define person-centered inputs while retaining limited scene context. The annotator independently processes three temporal views: 16 uniformly sampled frames from the full five-second clip, eight frames from the first half, and eight frames from the second half. We use greedy decoding (`do_sample=false`) with 128 maximum output tokens and constrain the output to the fixed vocabulary or the two special outcomes. The prompt instructs the annotator to rely only on observable body motion, posture, dynamics, and characteristic choreography, and to ignore clothing, background, performer identity, and inferred musical content.

We retain a canonical label only when at least two views agree and neither marks the clip as ambiguous. A clear dance outside the vocabulary is labeled *other*; conflicting predictions, insufficiently visible motion, or hybrid clips without a single stable label are assigned *uncertain*. No audio, lyrics, music caption, or target-side annotation is exposed to this procedure. Two dance-knowledge annotators independently audit a source-stratified subset containing at least 100 accepted clips per canonical class and 300 clips from each special outcome; disagreements are adjudicated. Before the final vocabulary is frozen, a category with fewer than 500 accepted training clips or unreliable audit performance is merged into its coarser parent or *other*. The vocabulary, alias table, annotator checkpoint and prompt, sampled-frame indices, per-view predictions, final canonical labels, label provenance, and class counts are stored with the dataset. The resulting labels are frozen before any prototype statistics are computed.

Prototype estimation. Only training-split style labels are used to group clips and estimate the prototypes; benchmark labels never enter prototype construction. We estimate each routing prototype exclusively from training-split motion statistics. For motion sequence $\mathbf{X}_i$, let $S(\cdot)$ denote scalar standardization using training-clip statistics. For the freeze score, we first standardize framewise motion energy using statistics computed over all training frames:

$$\widehat{e}_{i,t} = \frac{e_{i,t} - \mu_e^{\text{train}}}{\sigma_e^{\text{train}} + 10^{-6}}. \tag{19}$$

The same frozen μ_e^{train} and σ_e^{train} are used for benchmark clips. Let $\mathrm{ACF}(e_i; \ell)$ denote normalized motion-energy autocorrelation, and let $\mathcal{F}(e_i)_f$ denote the DFT coefficient at physical frequency f.

We compute four expert-aligned scores:

$$\begin{aligned} r_{i,\text{hit}} &= \tfrac{1}{2}[S(\mathbb{E}_t[p_t]) + S(\mathbb{E}_t[|\Delta a_t|])]\,, \\ r_{i,\text{groove}} &= S\left(\max_{\ell\in\mathcal{L}} \text{ACF}(e_i;\ell)\right), \\ r_{i,\text{freeze}} &= S(\mathbb{E}_t[\mathbb{I}(\widehat{e}_{i,t} < \eta_e)])\,, \\ r_{i,\text{phrase}} &= S\left(\frac{\sum_{0<f\le f_c} |\mathcal{F}(e_i)_f|^2}{\sum_{f>0} |\mathcal{F}(e_i)_f|^2 + 10^{-6}}\right). \end{aligned} \tag{20}$$

We use $\mathcal{L} = \{6, \ldots, 26\}$, $\eta_e = -0.5$ (one half standard deviation below the training-set mean energy), and $f_c = 1.5$ Hz; for 128 samples over approximately five seconds, this lag range covers roughly 59–256 BPM. For style s, we aggregate its training examples $\mathcal{D}_s$ and normalize across experts:

$$u_{s,k} = \frac{1}{|\mathcal{D}_s|}\sum_{i\in\mathcal{D}_s} r_{i,k}, \qquad p_{s,k}^{\text{proto}} = \frac{\exp(u_{s,k}/T_p)}{\sum_{j=1}^{K}\exp(u_{s,j}/T_p)}, \qquad T_p = 1. \tag{21}$$

We apply uniform smoothing,

$$\widetilde{\mathbf{p}}_s^{\text{proto}} = (1-\epsilon)\mathbf{p}_s^{\text{proto}} + \epsilon\frac{\mathbf{1}}{K}, \qquad K = 4, \quad \epsilon = 0.08. \tag{22}$$

All statistics are estimated exclusively from the training split, serialized with SART, and frozen during tokenizer pretraining, downstream adaptation, and evaluation. During training and benchmark evaluation, s is the frozen canonical video-side label constructed above; it is never inferred from paired audio or music-side text. For an external video, a valid user-supplied tag takes precedence; otherwise, we apply the same frozen video annotator. Clips labeled *other* or *uncertain*, as well as omitted or out-of-vocabulary tags, use a marginal prototype obtained by averaging $\mathbf{r}_i$ over all training clips and applying the same temperature-normalized softmax. Neither label assignment nor prototype estimation uses benchmark-set statistics.

The router first computes $\mathbf{H} = \text{Conv}_{11}(\mathbf{X})$ and applies style FiLM,

$$[\boldsymbol{\gamma}_s, \boldsymbol{\beta}_s] = \text{Linear}(\widetilde{\mathbf{p}}_s^{\text{proto}}), \qquad \widetilde{\mathbf{H}}_t = \mathbf{H}_t \odot (1 + \tanh\boldsymbol{\gamma}_s) + \boldsymbol{\beta}_s. \tag{23}$$

A second kernel-11 convolution predicts a bounded local residual,

$$\boldsymbol{\delta}_t = \delta_{\max}\tanh(\text{Conv}_{11}(\widetilde{\mathbf{H}})_t), \qquad \delta_{\max} = 1.0, \tag{24}$$

and the routing weights are

$$\mathbf{q}_t = \text{Softmax}\left(\frac{\lambda_{\text{style}}\log\widetilde{\mathbf{p}}_s^{\text{proto}} + \mathbf{b} + \alpha_{\text{route}}\boldsymbol{\delta}_t}{\tau}\right), \qquad \lambda_{\text{style}} = 1.0, \quad \alpha_{\text{route}} = 0.3, \quad \tau = 1.3. \tag{25}$$

The output token is $\mathbf{z}_t = \sum_{k=1}^{4} q_{t,k}\mathbf{Z}_t^{(k)}$.

E.3 SART Pretraining Objective

SART is pretrained independently of the music generator for 30 epochs using AdamW, a batch size of 64, a learning rate of 10^{-4}, and weight decay 10^{-4}. We perturb the input as

$$\widetilde{\mathbf{X}} = \mathbf{X} + 0.05\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{26}$$

and reconstruct the clean representation using an MLP R:

$$\mathcal{L}_{\text{rec}} = \text{MSE}(R(\mathbf{Z}_r^{\text{SART}}), \mathbf{X}). \tag{27}$$

For the batched SART output $\mathbf{Z}_r^{\text{SART}} \in \mathbb{R}^{B\times T\times D_r}$ and routing probabilities $\mathbf{q} \in \mathbb{R}^{B\times T\times K}$, where B, T, $D_r = 512$, and K denote the batch size, number of frames, rhythm-token width, and number

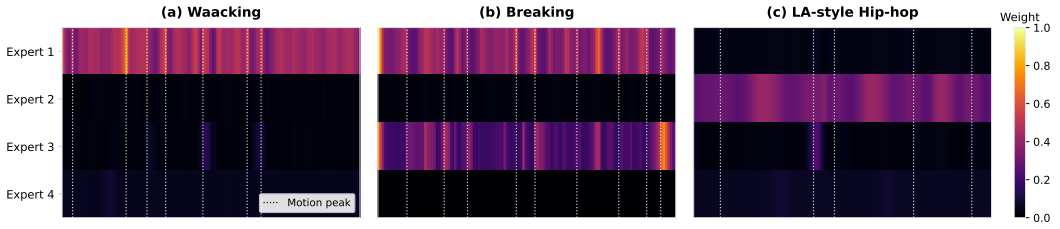


Figure 9: Time-varying contributions of SART rhythm experts. We visualize router-weighted expert contributions for (a) Waacking, (b) Breaking, and (c) LA-style Hip-hop. Color intensity denotes the globally normalized contribution while white dotted lines indicate detected motion peaks. Experts 1–4 receive the sharp-hit, groove, freeze/hold, and smooth-phrase views, respectively. These patterns show differentiated expert usage, while the semantic labels remain architectural anchors rather than uniquely identifiable latent factors.

of experts, respectively, all implemented auxiliary terms are

$$\mathcal{L}_{\text{tok}} = \mathbb{E}_{b,\, t=2,\ldots,T} \|\mathbf{z}_{b,t} - \mathbf{z}_{b,t-1}\|_2^2, \tag{28}$$

$$\mathcal{L}_{\text{route}} = \mathbb{E}_{b,\, t=2,\ldots,T} \|\mathbf{q}_{b,t} - \mathbf{q}_{b,t-1}\|_2^2, \tag{29}$$

$$\overline{q}_k = \mathbb{E}_{b,t}[q_{b,t,k}], \qquad \mathcal{L}_{\text{bal}} = \sum_{k=1}^{K} \overline{q}_k \log(K \overline{q}_k), \tag{30}$$

$$H(\mathbf{q}_{b,t}) = -\sum_{k} q_{b,t,k} \log q_{b,t,k}, \qquad \mathcal{L}_{\text{ent}} = \left(\mathbb{E}_{b,t}[H(\mathbf{q}_{b,t})] - 0.9\right)^2, \tag{31}$$

$$\mathcal{L}_{\text{prior}} = \mathbb{E}_{b,t}\, \mathrm{KL}(\mathbf{q}_{b,t} \| \widetilde{\mathbf{p}}^{\text{proto}}_{s_b}). \tag{32}$$

The frame-level $\mathcal{L}_{\text{prior}}$ anchors each routing distribution $\mathbf{q}_{b,t}$ to the style-conditioned prototype $\widetilde{\mathbf{p}}^{\text{proto}}_{s_b}$, preserving the global style preference while allowing the motion residual $\boldsymbol{\delta}_{b,t}$ to express local variation. The released checkpoint uses this frame-level prior KL; the implementation also supports applying KL to the clip-average routing distribution. For decorrelation, define the flattened expert-norm curve $\mathbf{c}_k = \mathrm{vec}_{b,t}(\|\mathbf{Z}^{(k)}_{b,t}\|_2)$ and standardize each curve to zero mean and unit variance. Then

$$\mathcal{L}_{\text{decorr}} = \frac{1}{K(K-1)} \sum_{i \neq j} \mathrm{Corr}(\mathbf{c}_i, \mathbf{c}_j)^2. \tag{33}$$

The learned-prototype variant replaces the frozen prototype in routing with a style-indexed distribution

$$\boldsymbol{\Delta}_{\theta,s} = g_\theta(\mathrm{Emb}_\theta(s)), \qquad \mathbf{p}_{\theta,s} = \mathrm{Softmax}\left(\lambda_{\text{proto}} \log \widetilde{\mathbf{p}}^{\text{proto}}_s + \boldsymbol{\Delta}_{\theta,s}\right). \tag{34}$$

Here $\mathrm{Emb}_\theta(s) \in \mathbb{R}^{32}$ is a learned style embedding and g_θ is a $32 \to 64 \to K$ MLP with a GELU activation. Its final linear layer is zero-initialized; with $\lambda_{\text{proto}} = 1.0$, this gives $\mathbf{p}_{\theta,s} = \widetilde{\mathbf{p}}^{\text{proto}}_s$ at initialization. We regularize the learned distribution by

$$\mathcal{L}_{\text{sem}} = \mathrm{KL}(\mathbf{p}_{\theta,s} \| \widetilde{\mathbf{p}}^{\text{proto}}_s), \tag{35}$$

using $\lambda_{\text{sem}} = 0.03$ for the learned-prototype comparison. The style-conditioned routing prototype is frozen and contains no learnable prior parameters, so $\mathcal{L}_{\text{sem}}$ is omitted for the default checkpoint. The complete objective, with the prototype-anchoring term included only for the learned-prototype variant, is

$$\begin{aligned} \mathcal{L}_{\text{SART}} = \mathcal{L}_{\text{rec}} &+ 0.005\mathcal{L}_{\text{tok}} + 0.01\mathcal{L}_{\text{route}} + 0.01\mathcal{L}_{\text{bal}} \\ &+ 0.01\mathcal{L}_{\text{ent}} + 0.03\mathcal{L}_{\text{prior}} + 0.03\mathcal{L}_{\text{sem}} + 0.03\mathcal{L}_{\text{decorr}}. \end{aligned} \tag{36}$$

Thus, the default released-checkpoint objective omits the semantic-prior term, whereas the learned-prototype variant includes it with weight 0.03.

E.4 Dance-Conditioning Pathway

The dance tokens are constructed from complementary RGB-motion and pose streams rather than from the SART encoder. For each video, a frozen RGB I3D network produces visual motion features

$\mathbf{F}^{\mathrm{rgb}} \in \mathbb{R}^{T_v \times 1024}$, while DWPose provides 17 body joints with $(x, y, \mathrm{confidence})$ coordinates, yielding $\mathbf{F}^{\mathrm{pose}} \in \mathbb{R}^{T_p \times 51}$. Both streams are converted to a common length $N_d = 512$ by uniform temporal sampling when longer and zero padding when shorter. We then apply separate Xavier-initialized linear projections and concatenate their outputs:

$$\mathbf{Z}_{\mathrm{dance}} = \mathrm{Concat}_{\mathrm{feat}}\left(\widehat{\mathbf{F}}^{\mathrm{rgb}} W_{\mathrm{rgb}}^{\top}, \widehat{\mathbf{F}}^{\mathrm{pose}} W_{\mathrm{pose}}^{\top}\right) \in \mathbb{R}^{512 \times 2048}, \tag{37}$$

where $W_{\mathrm{rgb}} \in \mathbb{R}^{1024 \times 1024}$ and $W_{\mathrm{pose}} \in \mathbb{R}^{1024 \times 51}$ are preprocessing projections, and the hat denotes the length-aligned sequence. I3D, DWPose, and these preprocessing projections are not optimized during RAMP adaptation; their outputs are cached before generator training and shared by all compared variants.

During Stage 1, all dance-token positions are zeroed and masked. During Stage 2, the cached tokens pass through a trainable $2048 \to 2048$ condition projector initialized to the identity, together with a learned modality-type embedding; these conditioning modules are optimized jointly with the student LoRA. This pathway preserves broad visual-motion and pose context. In contrast, SART converts explicit kinematic rhythm cues into 128 compact tokens whose expert routing emphasizes event timing and style-dependent rhythmic structure. The two conditions are therefore complementary: $\mathbf{Z}_{\mathrm{dance}}$ supplies general choreographic context, whereas $\bar{\mathbf{Z}}_r$ supplies specialized rhythmic control.

Fixed preprocessing versus trainable conditioning projections. These projections have distinct roles and parameter states. The *fixed preprocessing projections* operate offline: W_{rgb} and W_{pose} construct the cached 2048-dimensional dance tokens, while the fixed $512 \to 2048$ SART projection matches rhythm tokens to the generator width. They are not updated by the RAMP objective. The *trainable conditioning projectors* operate inside the Stage-2 model: separate identity-initialized $2048 \to 2048$ maps adapt the cached dance and rhythm tokens to the generator conditioning space. These two maps, together with their modality-type embeddings, are optimized jointly with the student LoRA and saved with the downstream adapter. Thus, "fixed" describes offline dimensional alignment, whereas "trainable" describes online modality adaptation during Stage 2.

E.5 Downstream Token Layout

SART returns 128 raw rhythm tokens $\mathbf{Z}_r^{\mathrm{SART}} \in \mathbb{R}^{128 \times 512}$. A fixed preprocessing matrix $W_r \in \mathbb{R}^{2048 \times 512}$ maps them to generator-width conditioning tokens,

$$\bar{\mathbf{Z}}_r = \mathbf{Z}_r^{\mathrm{SART}} W_r^{\top} \in \mathbb{R}^{128 \times 2048}. \tag{38}$$

This $512 \to 2048$ projection is not learned as part of the SART checkpoint. Independently, the fixed dance-conditioning preprocessing produces 512 dance tokens $\mathbf{Z}_{\mathrm{dance}} \in \mathbb{R}^{512 \times 2048}$. ACE-Step encodes the music description D and lyrics L independently:

$$\mathbf{Z}_d = E_{\mathrm{text}}(D), \qquad \mathbf{Z}_l = E_{\mathrm{lyric}}(L), \tag{39}$$

where E_{text} is the pretrained text encoder and E_{lyric} is the dedicated lyric encoder. Their outputs, together with the fixed silence-timbre context used by ACE-Step, form the native variable-length semantic condition $\mathbf{C}_{\mathrm{sem}}$. RAMP prepends the independently produced dance and rhythm tokens, giving

$$\mathbf{C}_{\mathrm{full}} = \left[\mathbf{Z}_{\mathrm{dance}}; \bar{\mathbf{Z}}_r; \mathbf{C}_{\mathrm{sem}}\right]. \tag{40}$$

For layout-compatible text/lyric-only forwards, the same 640 positions are retained but zeroed and masked:

$$\mathbf{C}_{\mathrm{text}} = \left[\mathbf{0}_{640}; \mathbf{C}_{\mathrm{sem}}\right]. \tag{41}$$

Accordingly, the phrase *640 dance tokens* does not denote homogeneous dance tokens: it is the total non-text motion-prefix length, comprising 512 dance tokens and 128 SART rhythm tokens. We use *640 non-text motion-prefix tokens (512 dance + 128 rhythm)* to avoid this ambiguity.